Sun Java System Directory Server Enterprise Edition 6.2 관리 설명서

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

부품 번호: 820–3203
2007년 9월

071012@19028

목차

그림

표

코드 예

머리말

관리 설명서에서는 명령줄에서 디렉토리 서버 및 디렉토리 프록시 서버 기능을 구성하는 절차에 대해 설명합니다. 웹 기반 인터페이스(디렉토리 서비스 제어 센터)를 사용하여 이러한 기능을 구성하는 방법은 온라인 도움말을 참조하십시오.

대상

이 **관리 설명서**는 디렉토리 서버 및 디렉토리 프록시 서버 소프트웨어 관리자를 대상으로 합니다.

본 설명서를 읽기 전에

본 설명서에서는 소프트웨어 설치 관련 정보를 제공하지 않습니다. 설치에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide**를 참조하십시오.

디렉토리 서버 또는 디렉토리 프록시 서버의 이전 버전에서 마이그레이션하는 경우 서버 마이그레이션에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Migration Guide**를 참조하십시오. 이 버전의 새로운 기능에 익숙하지 않은 경우 **Sun Java System Directory Server Enterprise Edition 6.2 Evaluation Guide**에서 새로운 기능의 개요를 읽어 보십시오.

본 설명서의 구성

제1부에서는 디렉토리 서버 관리 절차에 대해 설명합니다.

제2부에서는 디렉토리 프록시 서버 관리 절차에 대해 설명합니다.

본 설명서에 사용된 예

일관성을 위해 본 설명서에서는 동일한 데이터 예가 계속 사용됩니다. 이 값을 사용자 시스템에 적합한 값으로 대체하십시오.

표 P–1 예에 사용된 기본값

변수	예에 사용된 값
접미어(SUFFIX_DN)	dc=example,dc=com
인스턴스 경로(INSTANCE_PATH)	디렉토리 서버: /local/ds/ 디렉토리 프록시 서버: /local/dps/
호스트 이름(HOST)	host1, host2, host3
포트(PORT)	LDAP: 루트 기본값: 389. 비루트 기본값: 1389 SSL 기본값: 루트 기본값: 636. 비루트 기본값: 1636

Directory Server Enterprise Edition 설명서 세트

이 Directory Server Enterprise Edition 설명서 세트는 Sun Java System Directory Server Enterprise Edition을 사용하여 디렉토리 서비스를 평가, 설계, 배포 및 관리하는 방법을 설명합니다. 또한 Directory Server Enterprise Edition용 클라이언트 응용 프로그램을 개발하는 방법도 보여줍니다. Directory Server Enterprise Edition 설명서 세트는 `http://docs.sun.com/coll/1224.3` 및 `http://docs.sun.com/coll/1586.3`에서 사용할 수 있습니다.

Directory Server Enterprise Edition에 대해 알아보려면 다음에 나열된 설명서를 순서대로 검토하십시오.

표 P–2 Directory Server Enterprise Edition Documentation

설명서 제목	내용
Sun Java System Directory Server Enterprise Edition 6.2 릴리스 노트	알려진 문제를 비롯하여 Directory Server Enterprise Edition에 대한 최신 정보가 포함되어 있습니다.
Sun Java System Directory Server Enterprise Edition 6.2 Documentation Center	설명서 세트의 주요 영역에 대한 링크가 포함되어 있습니다.
Sun Java System Directory Server Enterprise Edition 6.2 Evaluation Guide	이 릴리스의 주요 기능을 소개합니다. 이러한 기능이 작동하는 방식과 단일 시스템에서 구현할 수 있는 배포 상황을 가정하여 제공하는 기능을 설명합니다.

표 P–2 Directory Server Enterprise Edition Documentation *(계속)*

설명서 제목	내용
Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide	Directory Server Enterprise Edition을 기반으로 가용성과 확장성이 높은 디렉토리 서비스를 계획 및 설계하는 방법에 대해 설명합니다. 배포 계획 및 설계의 기본 개념과 원칙을 제공합니다. 솔루션 라이프사이클을 설명하고 Directory Server Enterprise Edition을 기반으로 솔루션을 계획할 때 사용할 높은 수준의 예와 전략을 제공합니다.
Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide	Directory Server Enterprise Edition 소프트웨어 설치 방법에 대해 설명합니다. 설치할 구성 요소를 선택하는 방법, 설치 후 이 구성 요소를 구성하는 방법 및 구성된 구성 요소가 제대로 작동하는지 확인하는 방법을 보여줍니다. 디렉토리 편집기 설치에 대한 자세한 내용을 보려면 `http://docs.sun.com/coll/DirEdit_05q1`로 이동하십시오. 디렉토리 편집기를 설치하기 전에 **Sun Java System Directory Server Enterprise Edition 6.2 릴리스 노트**에서 디렉토리 편집기 관련 정보를 읽어 보십시오.
Sun Java System Directory Server Enterprise Edition 6.2 Migration Guide	이전 버전의 디렉토리 서버, 디렉토리 프록시 서버 및 Identity Synchronization for Windows에서 구성 요소를 업그레이드하는 방법에 대해 설명합니다.
Sun Java System Directory Server Enterprise Edition 6.2 관리 설명서	Directory Server Enterprise Edition 관리를 위한 명령줄 지침을 제공합니다. 디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)를 사용하여 Directory Server Enterprise Edition을 관리하는 방법에 대한 힌트 및 지침은 DSCC의 온라인 도움말을 참조하십시오. 디렉토리 편집기 관리에 대한 자세한 내용을 보려면 `http://docs.sun.com/coll/DirEdit_05q1`로 이동하십시오 Identity Synchronization for Windows 설치 및 구성에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide**의 제II부, “Installing Identity Synchronization for Windows”를 참조하십시오.
Sun Java System Directory Server Enterprise Edition 6.2 Developer’s Guide	Directory Server Enterprise Edition의 일부로 제공된 도구 및 API와 함께 디렉토리 클라이언트 응용 프로그램을 개발하기 위한 지침을 제공합니다.
Sun Java System Directory Server Enterprise Edition 6.2 Reference	Directory Server Enterprise Edition의 기술적 기초 및 개념적 기초를 소개합니다. 구성 요소, 구조, 프로세스 및 기능에 대해 설명하며 개발자 API에 대한 참조도 제공합니다.
Sun Java System Directory Server Enterprise Edition 6.2 Man Page Reference	Directory Server Enterprise Edition에서 사용 가능한 명령줄 도구, 스키마 객체 및 기타 공개 인터페이스에 대해 설명합니다. 이 설명서의 각 절을 온라인 설명서 페이지로 설치할 수 있습니다.
Sun Java System Directory Server Enterprise Edition 6.2 Troubleshooting Guide	다양한 도구를 사용하여 문제 범위를 정의하고 데이터를 수집하며 문제 영역을 해결하기 위한 정보를 제공합니다.

표 P–2 Directory Server Enterprise Edition Documentation (계속)

설명서 제목	내용
Sun Java System Identity Synchronization for Windows 6.0 Deployment Planning Guide	Identity Synchronization for Windows의 계획 및 배포를 위한 일반 지침 및 유용한 정보를 제공합니다.

관련 자료

SLAMD Distributed Load Generation Engine은 네트워크 기반 응용 프로그램의 성능을 테스트 및 분석하기 위해 설계된 Java™ 응용 프로그램입니다. 원래 이 프로그램은 LDAP 디렉토리 서버의 성능을 벤치마킹하고 분석하기 위해 Sun Microsystems, Inc.에서 개발했습니다. SLAMD는 OSI 승인 오픈 소스 사용권인 Sun Public License에 의거하여 오픈 소스 응용 프로그램으로 사용할 수 있습니다. SLAMD에 대한 정보를 보려면 `http://www.slamd.com/`으로 이동하십시오. SLAMD를 java.net 프로젝트로 사용할 수도 있습니다. `https://slamd.dev.java.net/`을 참조하십시오.

JNDI(Java Naming and Directory Interface) 기술은 Java 응용 프로그램에서 LDAP 및 DSML v2를 사용하여 디렉토리 서버에 액세스하는 것을 지원합니다. JNDI에 대한 자세한 내용은 `http://java.sun.com/products/jndi/`를 참조하십시오. **JNDI Tutorial**에는 JNDI 사용법에 대한 자세한 설명과 예가 수록되어 있으며 `http://java.sun.com/products/jndi/tutorial/`에서 볼 수 있습니다.

Directory Server Enterprise Edition은 독립형 제품으로, Sun Java Enterprise System의 구성 요소로, Sun 제품군(예: Sun Java Identity Management Suite)의 일부로 또는 기타 Sun 소프트웨어 제품의 애드온 패키지로 그 사용이 허가됩니다. Java Enterprise System은 네트워크나 인터넷 환경에서 배포된 엔터프라이즈 응용 프로그램을 지원하는 소프트웨어 인프라입니다. Directory Server Enterprise Edition이 Java Enterprise System의 구성 요소로 사용권을 부여받은 경우 `http://docs.sun.com/coll/1286.3` 및 `http://docs.sun.com/coll/1397.2`에 있는 시스템 설명서를 잘 알고 있어야 합니다.

Identity Synchronization for Windows에서는 제한된 라이센스로 Message Queue를 사용합니다. Message Queue 설명서는 `http://docs.sun.com/coll/1307.2` 및 `http://docs.sun.com/coll/1406.2`에서 볼 수 있습니다.

Identity Synchronization for Windows는 Microsoft Windows 비밀번호 정책과 함께 작동합니다.

- Windows 2003의 비밀번호 정책에 대한 자세한 내용은 Microsoft 설명서에서 온라인으로 볼 수 있습니다.
- Microsoft Certificate Services Enterprise Root 인증 기관에 대한 자세한 내용은 Microsoft 지원 설명서에서 온라인으로 볼 수 있습니다.
- Microsoft 시스템에서 SSL을 통해 LDAP를 구성하는 방법에 대한 자세한 내용은 Microsoft 지원 설명서에서 온라인으로 볼 수 있습니다.

재배포 가능 파일

Directory Server Enterprise Edition에서는 재배포 가능 파일을 제공하지 않습니다.

기본 경로 및 명령 위치

이 절에서는 설명서에서 사용된 기본 경로에 대해 설명하고 다양한 운영 체제 및 배포 유형에 대한 명령 위치를 제공합니다.

기본 경로

이 절의 표에서는 이 설명서에서 사용된 기본 경로를 설명합니다. 설치된 파일에 대한 자세한 설명은 다음 제품 설명서를 참조하십시오.

- **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 14 장, "Directory Server File Reference"
- **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 25 장, "Directory Proxy Server File Reference"
- **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 부록 A, "Directory Server Resource Kit File Reference"

표 P–3 기본 경로

자리 표시자	설명	기본값
install-path	Directory Server Enterprise Edition 소프트웨어의 기본 설치 디렉토리를 나타냅니다. 소프트웨어가 이 기본 *install-path* 아래의 디렉토리에 설치됩니다. 예를 들어 디렉토리 서버 소프트웨어는 *install-path*`/ds6/`에 설치됩니다.	`dsee_deploy`(1M)를 사용하여 zip 배포에서 설치한 경우 기본 *install-path*가 현재 디렉토리입니다. `dsee_deploy` 명령의 `-i` 옵션을 사용하여 *install-path*를 설정할 수 있습니다. 예를 들어 Java Enterprise System 설치 프로그램을 사용하는 것과 같이 기본 패키지 배포에서 설치할 경우 기본 *install-path*는 다음 위치 중 하나입니다. ■ Solaris 시스템 - `/opt/SUNWdsee/` ■ Red Hat 시스템 - `/opt/sun/` ■ Windows 시스템 - `C:\Program Files\Sun\JavaES5\DSEE`

표 P–3 기본 경로 *(계속)*

자리 표시자	설명	기본값
instance-path	디렉토리 서버 또는 디렉토리 프록시 서버 인스턴스에 대한 전체 경로를 나타냅니다. 설명서에서는 디렉토리 서버에 대해 `/local/ds/`를, 디렉토리 프록시 서버에 대해 `/local/dps/`를 사용합니다.	기본 경로가 없습니다. 하지만 **로컬** 파일 시스템에 항상 인스턴스 경로가 있어야 합니다. 다음 디렉토리를 사용하는 것이 좋습니다. Solaris 시스템의 경우 `/var` Sun Cluster를 사용할 경우 `/global`
serverroot	Identity Synchronization for Windows 설치 위치의 부모 디렉토리를 나타냅니다.	설치에 따라 다릅니다. 디렉토리 서버의 경우 *serverroot* 의 개념이 더 이상 존재하지 않습니다.
`isw-`*hostname*	Identity Synchronization for Windows 인스턴스 디렉토리를 나타냅니다.	설치에 따라 다릅니다.
/path/to/`cert8.db`	Identity Synchronization for Windows에 대한 클라이언트 인증서 데이터베이스의 기본 경로 및 파일 이름을 나타냅니다.	*current-working-dir*`/cert8.db`
serverroot`/isw-`*hostname*`/logs/`	시스템 관리자, 각 커넥터 및 중앙 로거의 Identity Synchronization for Windows 로컬 로그에 대한 기본 경로를 나타냅니다.	설치에 따라 다릅니다.
serverroot`/isw-`*hostname*`/logs/central/`	Identity Synchronization for Windows 중앙 로그에 대한 기본 경로를 나타냅니다.	설치에 따라 다릅니다.

명령 위치

이 절의 표에서는 Directory Server Enterprise Edition 설명서에서 사용되는 명령에 대한 위치를 제공합니다. 각 명령에 대한 자세한 내용은 관련 설명서 페이지를 참조하십시오.

표 P–4 명령 위치

명령	Java ES, 기본 패키지 배포	Zip 배포
cacaoadm	Solaris - /usr/sbin/cacaoadm	Solaris - *install-path*/dsee6/cacao_2/usr/sbin/cacaoadm
	Red Hat - /opt/sun/cacao/bin/cacaoadm	Red Hat- *install-path*/dsee6/cacao_2/cacao/bin/cacaoadm
	Windows - *install-path*\share\cacao_2\bin\cacaoadm.bat	Windows - *install-path*\dsee6\cacao_2\bin\cacaoadm.bat
certutil	Solaris - /usr/sfw/bin/certutil	*install-path*/dsee6/bin/certutil
	Red Hat- /opt/sun/private/bin/certutil	
dpadm(1M)	*install-path*/dps6/bin/dpadm	*install-path*/dps6/bin/dpadm
dpconf(1M)	*install-path*/dps6/bin/dpconf	*install-path*/dps6/bin/dpconf
dsadm(1M)	*install-path*/ds6/bin/dsadm	*install-path*/ds6/bin/dsadm
dsccmon(1M)	*install-path*/dscc6/bin/dsccmon	*install-path*/dscc6/bin/dsccmon
dsccreg(1M)	*install-path*/dscc6/bin/dsccreg	*install-path*/dscc6/bin/dsccreg
dsccsetup(1M)	*install-path*/dscc6/bin/dsccsetup	*install-path*/dscc6/bin/dsccsetup
dsconf(1M)	*install-path*/ds6/bin/dsconf	*install-path*/ds6/bin/dsconf
dsee_deploy(1M)	제공되지 않음	*install-path*/dsee6/bin/dsee_deploy
dsmig(1M)	*install-path*/ds6/bin/dsmig	*install-path*/ds6/bin/dsmig
entrycmp(1)	*install-path*/ds6/bin/entrycmp	*install-path*/ds6/bin/entrycmp
fildif(1)	*install-path*/ds6/bin/fildif	*install-path*/ds6/bin/fildif
idsktune(1M)	제공되지 않음	압축되지 않은 Zip 배포의 루트에서
insync(1)	*install-path*/ds6/bin/insync	*install-path*/ds6/bin/insync
ns-accountstatus(1M)	*install-path*/ds6/bin/ns-accountstatus	*install-path*/ds6/bin/ns-accountstatus
ns-activate(1M)	*install-path*/ds6/bin/ns-activate	*install-path*/ds6/bin/ns-activate
ns-inactivate(1M)	*install-path*/ds6/bin/ns-inactivate	*install-path*/ds6/bin/ns-inactivate
repldisc(1)	*install-path*/ds6/bin/repldisc	*install-path*/ds6/bin/repldisc

표 P–4 명령 위치 *(계속)*

명령	Java ES, 기본 패키지 배포	Zip 배포
schema_push(1M)	*install-path*/ds6/bin/schema_push	*install-path*/ds6/bin/schema_push
smcwebserver	Solaris 및 Linux- /usr/sbin/smcwebserver	이 명령은 기본 패키지 배포를 사용하여 설치된 경우 DSCC에만 적용됩니다.
	Windows - *install-path*\share\webconsole\bin\smcwebserver	
wcadmin	Solaris 및 Linux- /usr/sbin/wcadmin	이 명령은 기본 패키지 배포를 사용하여 설치된 경우 DSCC에만 적용됩니다.
	Windows - *install-path*\share\webconsole\bin\wcadmin	

활자체 규칙

다음 표는 이 책에서 사용된 활자체 변경 사항에 대해 설명합니다.

표 P–5 활자체 규칙

서체	의미	예
AaBbCc123	명령, 파일 및 디렉토리의 이름, 그리고 컴퓨터 화면에 출력되는 내용입니다.	.login 파일을 편집하십시오. ls -a 명령을 사용하여 모든 파일을 나열하십시오. machine_name% you have mail.
AaBbCc123	사용자가 입력하는 내용으로 컴퓨터 화면의 출력 내용과 대조됩니다.	machine_name% **su** Password:
AaBbCc123	실제 이름이나 값으로 대체되는 자리 표시자입니다.	파일을 제거하는 명령은 rm *filename*입니다.
AaBbCc123	책 제목, 새로 나오는 용어, 강조 표시할 단어입니다. (일부 강조된 항목은 온라인상에서 볼드로 표시됩니다.)	**사용자 설명서**의 6장을 읽으십시오. **캐시**는 로컬로 저장된 복사본입니다. 파일을 저장하지 **마십시오**.

명령 예의 쉘 프롬프트

아래 표에는 기본 시스템 프롬프트와 수퍼유저 프롬프트가 설명되어 있습니다.

표 P–6 쉘 프롬프트

쉘	프롬프트
UNIX 및 Linux 시스템의 C 쉘	`machine_name%`
UNIX 및 Linux 시스템의 C 쉘 수퍼유저	`machine_name#`
UNIX 및 Linux 시스템의 Bourne 쉘 및 Korn 쉘	`$`
UNIX 및 Linux 시스템의 Bourne 쉘 및 Korn 쉘 수퍼유저	`#`
Microsoft Windows 명령줄	`C:\`

기호 규칙

아래 표에는 이 설명서에서 사용할 수 있는 기호가 설명되어 있습니다.

표 P–7 기호 규칙

기호	설명	예	의미
`[ ]`	선택적 인수와 명령 옵션을 포함합니다.	`ls [-l]`	`-l` 옵션은 필수가 아닙니다.
`{ | }`	필수 명령 옵션에 대한 일련의 선택 항목을 포함합니다.	`-d {y|n}`	`-d` 옵션에서는 `y` 인수 또는 `n` 인수를 사용해야 합니다.
`${ }`	변수 참조를 나타냅니다.	`${com.sun.javaRoot}`	`com.sun.javaRoot` 변수 값을 참조합니다.
-	동시에 입력하는 여러 키를 결합합니다.	Ctrl-A	Ctrl 키를 누른 채로 A 키를 누릅니다.
+	연속해서 입력하는 여러 키를 결합합니다.	Ctrl+A+N	Ctrl 키를 눌렀다가 놓은 다음 후속 키를 누릅니다.
→	그래픽 사용자 인터페이스의 메뉴 항목 선택을 나타냅니다.	파일 → 새로 만들기 → 템플리트	파일 메뉴에서 새로 만들기를 선택합니다. 새로 만들기 하위 메뉴에서 템플리트를 선택합니다.

설명서, 지원 및 교육

Sun 웹 사이트에서는 다음 추가 자원에 대한 정보를 제공합니다.

- 설명서 (http://www.sun.com/documentation/)
- 지원 (http://www.sun.com/support/)
- 교육 (http://www.sun.com/training/)

Sun 제품 설명서 검색

docs.sun.com℠ 웹 사이트에서 Sun 제품 설명서를 검색하는 것 외에, 검색 필드에 다음 구문을 입력하여 검색 엔진을 사용할 수 있습니다.

search-term `site:docs.sun.com`

예를 들어 "브로커"를 검색하려면 다음을 입력합니다.

```
broker site:docs.sun.com
```

검색에 다른 Sun 웹 사이트(예: java.sun.com, www.sun.com 및 developers.sun.com)를 포함하려면 검색 필드에서 `docs.sun.com` 대신 `sun.com`을 사용합니다.

타사 웹 사이트 참조 사항

이 설명서에서는 추가 관련 정보를 제공하기 위해 타사 URL을 참조하기도 합니다.

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

사용자 의견 환영

Sun은 설명서의 내용을 지속적으로 개선하고자 하며 사용자 여러분의 의견과 제안을 환영합니다. 사용자 의견을 보내시려면 `http://docs.sun.com`에서 의견 보내기를 누릅니다. 온라인 양식에 전체 설명서 제목과 부품 번호를 기입해 주시기 바랍니다. 부품 번호는 해당 설명서의 제목 페이지나 문서 URL에 있으며 일반적으로 7자리 또는 9자리 숫자입니다. 예를 들어 이 설명서의 부품 번호는 820-3203입니다.

1

디렉토리 서버 관리

◆ ◆ ◆ 1 장

1 디렉토리 서버 도구

Sun Java™ System Directory Server Enterprise Edition은 복제된 환경에서 여러 개의 서버, 인스턴스 및 접미어를 관리할 수 있는 브라우저 인터페이스 및 명령줄 도구를 제공합니다. 이 장에서는 디렉토리 서버 관리 도구에 대한 개요를 제공합니다.

이 장은 다음 내용으로 구성되어 있습니다.

디렉토리 서버 관리 개요

디렉토리 서버 관리 프레임워크에 대한 정보는 이 설명서 세트의 다른 설명서에서 제공합니다.

- 디렉토리 서버 관리 프레임워크에 대한 개요는 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 "Directory Server Enterprise Edition Administration Model"을 참조하십시오.
- 디렉토리 서버 관리 프레임워크에 대한 자세한 참조 정보는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 1 장, "Directory Server Overview"를 참조하십시오.

DSCC 및 명령줄 사용 시기 결정

Directory Server Enterprise Edition은 디렉토리 서버 및 디렉토리 프록시 서버의 이름을 지정하는 두 개의 사용자 인터페이스(브라우저 인터페이스인 디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC) 및 명령줄 인터페이스)를 제공합니다.

DSCC를 사용하여 절차 수행 가능 여부 결정

이 설명서에 있는 대부분의 절차는 명령줄 또는 DSCC를 사용하여 수행할 수 있습니다. 이 설명서의 절차에서는 명령줄을 사용하여 해당 절차를 수행하는 방법에 대해 설명하지만 대부분의 경우 DSCC를 사용하여 동일한 작업을 수행할 수 있습니다. 특정 절차에 DSCC를 사용할 수 있는 경우 이에 대한 설명이 절차 시작 부분에 명시됩니다.

DSCC 온라인 도움말에서는 DSCC를 사용하여 이 설명서의 절차를 수행하는 방법에 대한 자세한 지침을 제공합니다.

DSCC 사용이 바람직한 경우

DSCC를 사용하면 다음 절에 설명된 것처럼 일부 작업을 명령줄에서 수행하는 것보다 더 쉽게 수행할 수 있습니다. 일반적으로 여러 서버에 명령을 적용해야 할 때에는 DSCC를 사용하는 것이 가장 좋습니다.

서버 및 접미어 복제 상태 보기

DSCC에서는 DSCC에 등록된 모든 서버 인스턴스와 구성된 모든 접미어 및 각각의 상태가 테이블로 표시됩니다.

서버 테이블은 디렉토리 서버 탭에 있으며 서버의 작동 상태를 표시합니다. 나타낼 수 있는 서버 상태에 대한 전체 목록은 디렉토리 서버 온라인 도움말을 참조하십시오.

접미어 테이블은 접미어 탭에 있으며 항목 수와 누락된 변경 사항의 수, 경과 기간 등의 복제 상태 정보를 표시합니다. 이 테이블에 표시되는 정보에 대한 자세한 내용은 디렉토리 서버 온라인 도움말을 참조하십시오.

서버 그룹 관리

서버 그룹은 서버를 모니터링하고 구성하는 데 도움이 됩니다. 그룹을 만들고 이 그룹에 서버를 할당할 수 있습니다. 예를 들어 지리적 위치 또는 기능별로 서버를 그룹화할 수 있습니다. 서버가 여러 개 있는 경우 디렉토리 서버 탭에 표시된 서버를 필터링하여 그룹에 있는 서버만 보도록 할 수 있습니다. 또한, 한 서버의 서버 구성(예: 색인 또는 캐시 설정)을 그룹에 있는 다른 모든 서버에 복사할 수도 있습니다. 서버 그룹을 설정하고 사용하는 방법에 대한 지침은 디렉토리 서버 온라인 도움말을 참조하십시오.

구성 설정 복사

DSCC를 사용하여 기존 서버, 접미어 또는 복제 계약에 대한 구성 설정을 하나 이상의 다른 서버, 접미어 또는 복제 계약에 복사할 수 있습니다. 이러한 작업을 각각 수행하는 방법에 대한 자세한 내용은 디렉토리 서버 온라인 도움말을 참조하십시오.

복제 구성

DSCC를 사용하여 복제 토폴로지를 쉽고 빠르게 설정할 수 있습니다. 서버 인스턴스를 만든 다음 DSCC에서 제공하는 단계를 사용하여 각 서버의 역할을 지정하면 됩니다. DSCC에서는 자동으로 사용자를 위한 복제 계약을 만듭니다. DSCC를 사용하여 복제를 구성하는 방법에 대한 자세한 내용은 디렉토리 서버 온라인 도움말을 참조하십시오.

디렉토리 서비스 제어 센터 인터페이스

디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)는 브라우저를 사용하여 디렉토리 서버 및 디렉토리 프록시 서버를 관리할 수 있는 사용자 인터페이스입니다.

DSCC를 구성하려면 72 페이지 “DSCC 구성”을 참조하십시오. DSCC 사용에 대한 자세한 내용은 다음 절을 참조하십시오.

DSCC의 관리 사용자

DSCC에는 몇 가지 로그인 관리 권한이 있습니다.

- **OS 사용자.** 서버 인스턴스를 만들고 `dsadm` 명령을 사용하여 서버 인스턴스에서 운영 체제 명령을 실행할 권한이 있는 유일한 사용자입니다. 경우에 따라 DSCC에서 OS 사용자 비밀번호를 요청할 수도 있습니다. 이 사용자에게는 비밀번호가 있고 디렉토리 서버 인스턴스를 만들 수 있어야 합니다.
- **디렉토리 관리자.** 서버에 대한 LDAP 수퍼유저입니다. 기본 DN은 `cn=Directory Manager`입니다.
- **디렉토리 어드민 관리자.** 디렉토리 서버를 관리합니다. 이 사용자는 디렉토리 관리자와 동일한 권한을 갖지만 액세스 제어, 비밀번호 정책 및 인증 요구 사항을 따르며 디렉토리 어드민 관리자를 필요한 만큼 만들 수 있습니다.
- **디렉토리 서비스 관리자.** DSCC를 통해 여러 시스템의 서버 구성 및 데이터를 관리합니다. 이 사용자는 DSCC에 등록된 각 서버에 대해 디렉토리 관리자와 동일한 권한을 가지며 디렉토리 어드민 관리자 그룹의 구성원입니다.

▼ DSCC에 액세스하는 방법

DSCC에 액세스하는 데 어려움이 있는 경우 **Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide**의 "To Troubleshoot Directory Service Control Center Access"를 참조하십시오.

1 Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide의 "Software Installation"에 설명된 것처럼 DSCC가 올바르게 설치되었는지 확인합니다.

2 기본 패키지 설치로 DSCC를 설치한 경우 다음 단계를 수행합니다.

a. 브라우저를 열고 DSCC 호스트 URL을 다음 형식으로 입력합니다.

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

예를 들면 다음과 같습니다.

```
https://host1:6789
```

여기서 hostname은 DSCC 소프트웨어가 설치된 시스템입니다.

Sun Java Web Console 기본 포트는 6789입니다.

다음 그림은 Sun Java Web Console 로그인 창을 보여줍니다.

그림 1–1 Sun Java Web Console 로그인 창

b. Sun Java Web Console에 로그인합니다.

- Sun Java Web Console에 처음 로그인하는 경우에는 DSCC 소프트웨어가 설치된 시스템의 `root`로 로그인합니다.
- 다음 로그인 시에는 운영 체제 사용자 이름 및 비밀번호를 입력합니다. 이 사용자에게는 디렉토리 서버 인스턴스를 시작, 중지 및 관리할 수 있는 권한이 있어야 합니다.

로그인할 때 응용 프로그램 목록이 표시됩니다.

c. 디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)를 선택합니다.

DSCC 로그인 창이 표시됩니다.

3 Zip 설치로 DSCC를 설치한 경우 다음 단계를 수행합니다.

a. DSCC 호스트 URL을 입력하여 기본 응용 프로그램 서버에서 직접 DSCC에 액세스합니다. DSCC 호스트 URL은 응용 프로그램 서버의 구성에 따라 다음 중 하나일 수 있습니다.

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

또는

`http://`*hostname*`:6789`

b. 다음 명령을 사용하여 DSCC를 초기화합니다.

`$` *install path*`/dscc6/bin/dsccsetup ads-create`

4 DSCC에 로그인합니다.

DSCC에 처음 로그인하는 경우에는 디렉토리 서비스 관리자 비밀번호를 설정해야 합니다. 다음 로그인 시에는 처음 로그인했을 때 설정한 비밀번호를 사용합니다.

이제 DSCC에 로그인되고 일반 작업 탭이 표시됩니다.

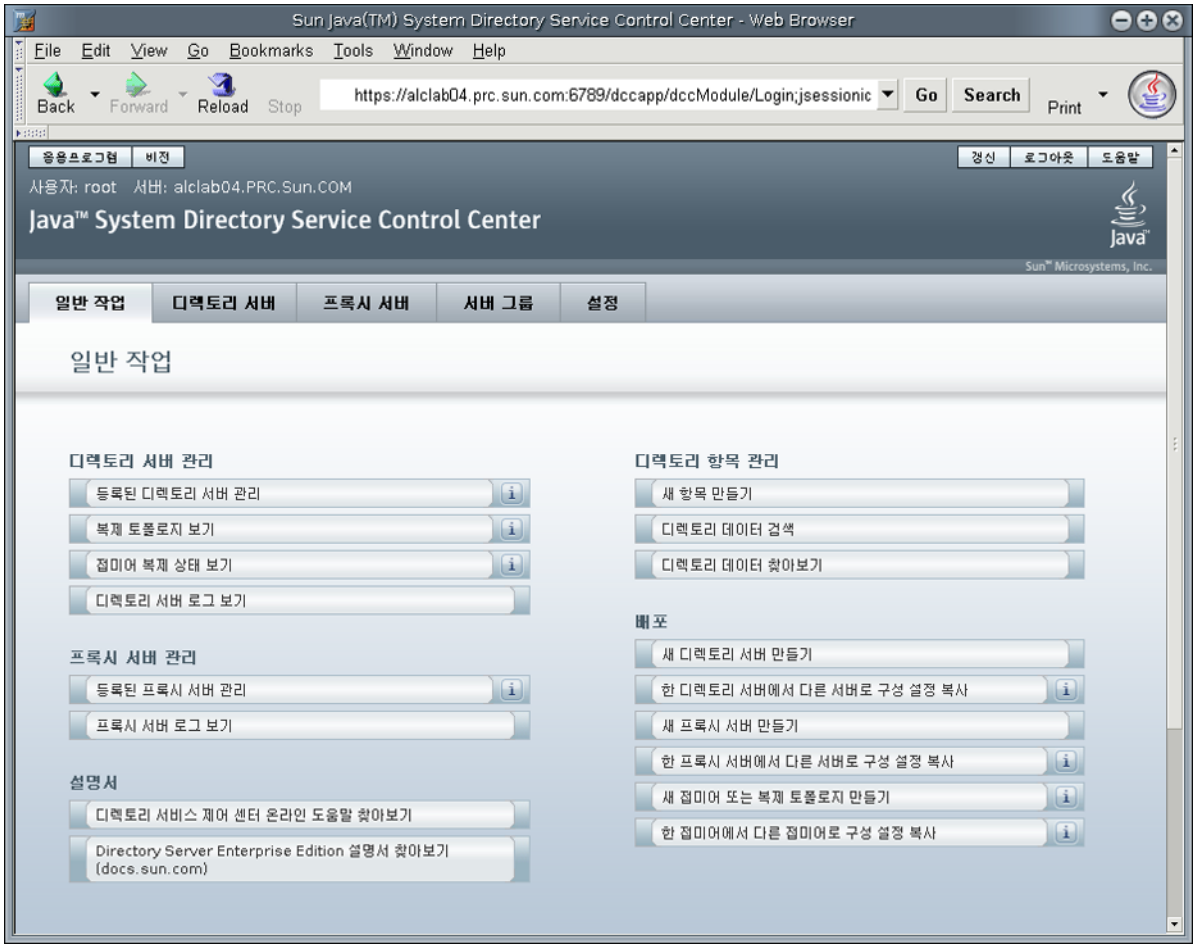


그림 1–2 DSCC 일반 작업 탭

5 이 탭을 사용하여 이동합니다.

- 일반 작업 탭에는 자주 사용되는 창과 마법사에 대한 바로 가기가 있습니다.
- 디렉토리 서버 탭에는 DSCC에서 관리되는 모든 디렉토리 서버가 표시됩니다. 특정 서버를 관리하고 구성하는 데 필요한 옵션에 대한 자세한 내용을 보려면 서버 이름을 누릅니다.
- 프록시 서버 탭에는 DSCC에서 관리되는 모든 디렉토리 프록시 서버가 표시됩니다. 특정 서버를 관리하고 구성하는 데 필요한 옵션에 대한 자세한 내용을 보려면 서버 이름을 누릅니다.

주 – DSCC를 사용하여 작업을 수행하는 방법에 대한 지침은 DSCC 온라인 도움말을 참조하십시오.

DSCC 탭 설명

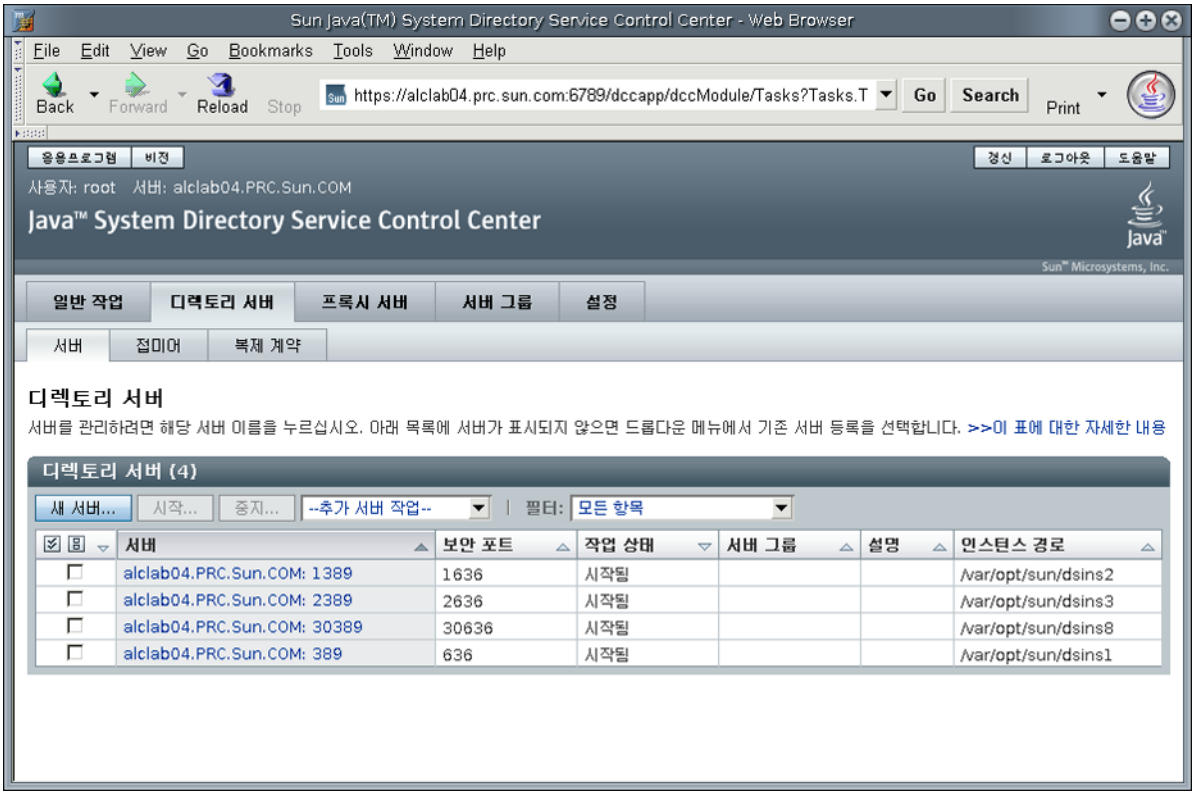


그림 1–3 서버 하위 탭의 디렉토리 서버 목록

인터페이스를 이동하려면 DSCC의 탭을 사용합니다.

일반 작업 탭

일반 작업 탭(그림 1–2 참조)은 DSCC를 열 때 표시되는 첫 인터페이스로, 디렉토리 데이터 검색, 로그 검사 및 서버 관리와 같은 자주 사용되는 관리 작업에 대한 링크가 포함되어 있습니다.

디렉토리 서버 탭

디렉토리 서버 탭(그림 1–3 참조)에는 DSCC에 등록된 모든 디렉토리 서버가 나열되며 각 서버에 대해 서버 상태 및 인스턴스 위치를 나타내는 인스턴스 경로를 볼 수 있습니다.

서버 이름을 누르면 해당 서버에만 관련된 다른 탭 집합이 별도의 창으로 표시됩니다.

프록시 서버 탭

프록시 서버 탭에는 DSCC에 등록된 모든 디렉토리 프록시 서버가 나열되며 각 서버에 대해 서버 상태 및 인스턴스 위치를 나타내는 서버 인스턴스 경로를 볼 수 있습니다.

서버 이름을 누르면 해당 서버에만 관련된 다른 탭 집합이 별도의 창으로 표시됩니다.

서버 그룹 탭

서버 그룹 탭을 사용하면 서버를 그룹에 할당함으로써 서버를 보다 쉽게 관리할 수 있습니다. 서버가 여러 개 있는 경우 필터를 사용하여 특정 그룹의 서버만 표시할 수 있습니다. 또한 한 서버의 서버 구성(예: 색인 또는 캐시 설정)을 그룹에 있는 다른 모든 서버에 복사할 수도 있습니다.

설정 탭

이 탭에는 DSCC 포트 번호가 표시되며, 이 탭을 사용하여 디렉토리 서비스 관리자를 만들고 삭제할 수 있습니다.

DSCC 온라인 도움말

온라인 도움말에서 제공되는 내용은 다음과 같습니다.

- 현재 사용 중인 페이지에서 상황에 맞는 도움말
- DSCC를 사용하여 관리 및 구성 절차를 수행하는 데 필요한 일반 도움말

대부분의 페이지에서 화면 오른쪽 상단의 도움말 버튼을 눌러 도움말에 액세스할 수 있습니다. 마법사 내에서는 도움말 탭을 눌러 도움말에 액세스할 수 있습니다. 일반 작업 탭에서 온라인 도움말에 액세스할 수도 있습니다.

디렉토리 서버 명령줄 도구

DSCC에서 수행하는 대부분의 작업은 명령줄 도구를 사용하여 수행할 수 있습니다. 이 도구를 사용하면 명령줄에서 직접 디렉토리 서버를 관리하고 스크립트를 사용하여 서버를 관리할 수 있습니다.

주요 디렉토리 서버 명령은 `dsadm` 및 `dsconf`입니다. 이 명령을 사용하여 백업, LDIF로 내보내기 및 인증서 관리 등의 작업을 수행할 수 있습니다. 이 명령에 대한 자세한 내용은 `dsadm`(1M) 및 `dsconf`(1M) 설명서 페이지를 참조하십시오.

이 절은 디렉토리 서버 명령줄 도구와 관련하여 다음 내용으로 구성되어 있습니다.

디렉토리 서버 명령 위치

디렉토리 서버 명령줄 도구는 기본 설치 디렉토리에 있습니다.

install-path/ds6/bin

설치 디렉토리는 운영 체제에 따라 다릅니다. 모든 운영 체제에 대한 설치 경로는 33 페이지 "기본 경로 및 명령 위치"에 나와 있습니다.

dsconf의 환경 변수 설정

dsconf 명령에는 환경 변수를 사용하여 미리 설정할 수 있는 몇 가지 옵션이 필요합니다. 이 명령을 사용할 때 옵션을 지정하지 않거나 환경 변수를 설정하지 않으면 기본 설정이 사용됩니다. 다음 옵션에 대한 환경 변수를 구성할 수 있습니다.

-D *user DN*	사용자 바인드 DN. 환경 변수: LDAP_ADMIN_USER. 기본값: cn=Directory Manager
-w *password-file*	사용자 바인드 DN에 대한 비밀번호 파일. 환경 변수: LDAP_ADMIN_PWF. 기본값: 비밀번호를 요청하는 메시지 표시
-h *host*	호스트 이름. 환경 변수: DIRSERV_HOST. 기본값: local host
-p *LDAP-port*	LDAP 포트 번호. 환경 변수: DIRSERV_PORT. 기본값: 389
-e, --unsecured	dsconf는 기본적으로 명확한 연결을 열도록 지정합니다. 환경 변수: DIRSERV_UNSECURED. 이 변수가 설정되지 않은 경우 dsconf는 기본적으로 보안 연결을 엽니다.

자세한 내용은 dsconf(1M) 설명서 페이지를 참조하십시오.

dsadm과 dsconf 비교

아래 표에는 dsadm과 dsconf 명령에 대한 비교 내용이 설명되어 있습니다.

표 1–1 dsadm과 dsconf 명령 비교

	dsadm 명령	dsconf 명령
설명	로컬 호스트에서 직접 실행해야 하는 관리 명령입니다. 예를 들면 다음과 같습니다. ■ 서버 시작 및 중지 ■ 서버 인스턴스 만들기	원격 호스트에서 실행할 수 있는 관리 명령입니다. 예를 들면 다음과 같습니다. ■ 복제 활성화 ■ 캐시 크기 설정
정보	서버를 중지해야 합니다(dsadm stop 및 dsadm info 명령 제외). 서버는 서버 인스턴스 경로(*instance-path*)로 식별합니다. 서버 인스턴스 경로에 대한 OS 액세스 권한이 있어야 합니다.	서버가 실행 중이어야 합니다. 서버는 호스트 이름(-h), 포트(-p) 또는 LDAPS 보안 포트(-P)로 식별합니다. 포트 번호를 지정하지 않으면 dsconf는 기본 포트(LDAP의 경우 389)를 사용합니다. cn=admin,cn=Administrators,cn=config 사용자와 같은 구성 데이터에 대한 LDAP 액세스 권한이 있어야 합니다.

dsadm 및 dsconf 사용에 대한 도움말 보기

dsadm 및 dsconf 명령을 사용하는 방법에 대한 자세한 내용은 dsadm(1M) 및 dsconf(1M) 설명서 페이지를 참조하십시오.

- 하위 명령 목록을 보려면 다음 명령을 입력합니다.

```
$ dsadm --help

$ dsconf --help
```

- 하위 명령을 사용하는 방법에 대한 자세한 내용을 보려면 다음 명령을 입력합니다.

```
$ dsadm subcommand --help

$ dsconf subcommand --help
```

dsconf를 사용하여 구성 등록 정보 수정

대부분의 dsconf 하위 명령을 사용하면 구성 등록 정보를 보고 수정할 수 있습니다.

- 디렉토리 서버에 사용된 구성 등록 정보를 나열하려면 다음을 입력합니다.

  ```
  $ dsconf help-properties
  ```

- 특정 등록 정보를 찾으려면 도움말 등록 정보의 출력 내용을 검색합니다.

  예를 들어 UNIX® 플랫폼을 사용하는 경우 참조와 관련된 모든 등록 정보를 검색하려면 다음 명령을 사용합니다.

  ```
  $ dsconf help-properties | grep -i referral
  SER  referral-url                      rw  M  LDAP_URL | undefined
       Referrals returned to clients requesting a DN not stored in this
       Directory Server (Default: undefined)
  SUF  referral-mode                     rw     disabled|enabled|only-on-write
       Specifies how referrals are used for requests involving the suffix
       (Default: disabled)
  SUF  referral-url                      rw  M  LDAP_URL | undefined
       Server(s) to which updates are referred (Default: undefined)
  SUF  repl-rewrite-referrals-enabled    rw     on|off
       Specifies whether automatic referrals are overwritten (Default: off)
  ```

  등록 정보가 접미어(SUF) 및 서버(SER)와 같은 대상 객체별로 그룹화됩니다. rw 키워드는 등록 정보가 읽기 및 쓰기 가능함을 나타냅니다. M 키워드는 등록 정보의 값이 여러 개임을 나타냅니다.

- 서버 속성을 보려면 세부 정보 표시 모드를 사용합니다. 예를 들어 UNIX 시스템에서 다음을 입력합니다.

  ```
  $ dsconf help-properties -v | grep -i referral-mode
  SUF  referral-mode    rw  disabled|enabled|only-on-write  nsslapd-state
    Specifies how referrals are used for requests involving the suffix
    (Default: disabled)
  ```

개별 등록 정보에 대한 자세한 내용은 해당 등록 정보에 대한 설명서 페이지를 참조하십시오. 설명서 페이지는 **Sun Java System Directory Server Enterprise Edition 6.2 Man Page Reference**에 있습니다.

dsconf로 여러 값을 갖는 등록 정보 설정

특정 디렉토리 서버 등록 정보의 값은 여러 개일 수 있습니다. 이 값을 지정하는 구문은 다음과 같습니다.

```
$ dsconf set-container-prop -h host -p port container-name \
 property:value1 property:value2
```

예를 들어 서버에 대한 암호화 암호를 여러 개 설정하려면 다음 명령을 사용합니다.

```
$ dsconf set-server-prop -h host1 -p 1389 ssl-cipher-family:SSL_RSA_WITH_RC4_128_MD5 \
 ssl-cipher-family:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
```

값이 이미 포함된 여러 값을 갖는 등록 정보에 값을 추가하려면 다음 구문을 사용합니다.

```
$ dsconf set-container-prop -h host -p port container-name property+:value
```

값이 이미 포함된 여러 값을 갖는 등록 정보에서 값을 제거하려면 다음 구문을 사용합니다.

```
$ dsconf set-container-prop -h host -p port container-name property-:value
```

예를 들어 이전에 설명된 시나리오에서 암호 목록에 SHA 암호화 암호를 추가하려면 이 명령을 실행합니다.

```
$ dsconf set-server-prop -h host1 -p 1389 \
 ssl-cipher-family+:TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
```

목록에서 MD5 암호를 제거하려면 이 명령을 실행합니다.

```
$ dsconf set-server-prop -h host1 -p 1389 ssl-cipher-family-:SSL_RSA_WITH_RC4_128_MD5
```

설명서 페이지

설명서 페이지는 디렉토리 서버에서 사용되는 모든 명령 및 속성에 대한 설명과 배포 시 명령 사용 방법에 대한 몇 가지 유용한 예를 제공합니다.

레거시 도구

레거시 도구는 이전 버전과의 호환성을 위해 일반 디렉토리 서버 도구에 포함되어 있습니다. 이 도구는 제공되어 있지만 더 이상 사용되지 않습니다.

◆ ◆ ◆ 2 장

2 디렉토리 서버 인스턴스 및 접미어

이 장에서는 디렉토리 서버 인스턴스 및 접미어를 만들고 관리하는 방법에 대해 설명합니다. 접미어 수준에서 구성되는 다양한 디렉토리 관리 작업에 대해서는 본 설명서의 다른 장에서 설명합니다.

이 장은 다음 내용으로 구성되어 있습니다.

서버 인스턴스 및 접미어 만들기에 대한 빠른 절차

이 장은 서버 인스턴스 및 접미어를 만드는 방법에 대한 자세한 정보로 구성되어 있습니다. 디렉토리 서버 인스턴스 및 접미어를 신속하게 만들고 일부 데이터 예를 가져와야 하는 경우에는 **Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide**의 "Server Instance Creation"을 참조하십시오.

디렉토리 서버 인스턴스 만들기 및 삭제

이 절에서는 디렉토리 서버 인스턴스를 만들고 삭제하는 방법에 대해 설명합니다.

▼ 디렉토리 서버 인스턴스를 만드는 방법

데이터를 관리하려면 명령줄 도구 또는 브라우저 인터페이스 디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)를 사용하여 디렉토리 서버 인스턴스를 만들어야 합니다. DSCC에서 디렉토리 서버 인스턴스는 간단하게 "디렉토리 서버"라고도 합니다.

디렉토리 서버 인스턴스를 만들 때 디렉토리 서버에 필요한 파일 및 디렉토리가 사용자가 지정한 *instance-path*에 만들어집니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

DSCC를 사용하여 새 서버 인스턴스를 만들면 기존 서버에서 서버 구성 설정의 일부 또는 전체를 복사할 수 있습니다.

1 새 디렉토리 서버 인스턴스를 만들고 인스턴스 경로를 설정합니다.

```
$ dsadm create instance-path
```

이 서버에 대한 디렉토리 관리자 비밀번호를 설정하라는 메시지가 표시됩니다.

서버 인스턴스에 대한 포트 번호를 기본값이 아닌 값으로 지정하거나, 다른 매개 변수를 지정하려면 `dsadm`(1M) 설명서 페이지를 참조하십시오.

예를 들어 `/local/ds` 디렉토리에 새 인스턴스를 만들려면 다음 명령을 사용합니다.

```
$ dsadm create /local/ds
Choose the Directory Manager password:
Confirm the Directory Manager password:
Use 'dsadm start /local/ds' to start the instance
```

2 서버 인스턴스가 올바르게 만들어졌는지 확인합니다.

```
$ dsadm info instance-path
```

예를 들면 다음과 같습니다.

```
$ dsadm info /local/ds1
Instance Path:          /local/ds1
Owner:                  user1(group1)
Non-secure port:        1389
Secure port:            1636
Bit format:             64-bit
State:                  Running
Server PID:             22555
DSCC url:               -
SMF application name:   -
Start at boot:          Disabled
Instance version:       D-A00
```

3 (옵션) Java Enterprise System 설치 프로그램이나 기본 패키지 설치를 사용하여 디렉토리 서버를 설치했고 OS에서 서비스 관리 솔루션을 제공하는 경우 다음 표와 같이 서버를 서비스로 관리할 수 있습니다.

운영 체제	명령
Solaris 10	Sun Cluster 환경에서 작업하는 경우에는 다음 명령을 사용합니다. `dsadm enable-service --type CLUSTER` *instance-path resource-group* 그렇지 않은 경우에는 다음 명령을 사용합니다. `dsadm enable-service --type SMF` *instance-path*
Solaris 9	Sun Cluster 환경에서 작업하는 경우에는 다음 명령을 사용합니다. `dsadm enable-service --type CLUSTER` *instance-path resource_group* 그렇지 않은 경우에는 다음 명령을 사용합니다. `dsadm autostart` *instance-path*
Linux, HP-UX	`dsadm autostart` *instance-path*
Windows	`dsadm enable-service --type WIN_SERVICE` *instance-path*

4 디렉토리 서버를 시작합니다.

```
$ dsadm start instance-path
```

주 – 서버가 실행 중이지만 데이터 또는 접미어가 없습니다. `dsconf`를 사용하여 접미어를 만듭니다.

---

5 (옵션) 다음 방법 중 하나를 사용하여 서버 인스턴스를 등록합니다.

- `https://`*host*`:6789` URL에 액세스하여 DSCC를 통해 서버를 등록합니다.
- `dsccreg add-server` 명령을 사용합니다.
  자세한 내용은 `dsccreg`(1M) 설명서 페이지를 참조하십시오.

6 비밀번호 정책을 사용하려 할 때 디렉토리 서버 인스턴스가 독립형인 경우 또는 인스턴스가 이미 `DS6-only` 비밀번호 정책 모드로 마이그레이션된 복제 토폴로지에 속한 경우 인스턴스를 해당 모드로 전환합니다.

```
$ dsconf pwd-compat -h host -p port to-DS6-migration-mode

## Beginning password policy compatibility changes .
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).
$ dsconf pwd-compat -h host -p port to-DS6-mode
```

```
## Beginning password policy compatibility changes .
## Password policy compatibility changes finished.

Task completed (slapd exit code: 0).
```

▼ 디렉토리 서버 인스턴스를 삭제하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 디렉토리 서버를 중지합니다.

```
$ dsadm stop instance-path
```

2 이전에 DSCC를 사용하여 서버를 관리했다면 명령줄을 사용하여 서버를 등록 취소합니다.

```
$ dsccreg remove-server /local/ds
Enter DSCC administrator's password:
/local/ds is an instance of DS
Enter password of "cn=Directory Manager" for /local/ds:
This operation will restart /local/ds.
Do you want to continue ? (y/n) y
Unregistering /local/ds from DSCC on localhost.
Connecting to /local/ds
Disabling DSCC access to /local/ds
Restarting /local/ds
```

자세한 내용은 dsccreg(1M) 설명서 페이지를 참조하십시오.

3 (옵션) 이전에 서비스 관리 솔루션에서 서버 인스턴스를 활성화했다면 서버가 서비스로 관리되지 않도록 설정합니다.

운영 체제	명령
Solaris 10	Sun Cluster 환경에서 작업하는 경우에는 다음 명령을 사용합니다. `dsadm disable-service --type CLUSTER` *instance-path* 그렇지 않은 경우에는 다음 명령을 사용합니다. `dsadm disable-service --type SMF` *instance-path*

운영 체제	명령
Solaris 9	Sun Cluster 환경에서 작업하는 경우에는 다음 명령을 사용합니다. `dsadm disable-service --type CLUSTER` *instance-path* 그렇지 않은 경우에는 다음 명령을 사용합니다. `dsadm autostart --off` *instance-path*
Linux, HP-UX	`dsadm autostart --off` *instance-path*
Windows	`dsadm disable-service --type WIN_SERVICE` *instance-path*

4 **서버 인스턴스를 삭제합니다.**

```
$ dsadm delete instance-path
```

주의 – 이 명령은 데이터베이스와 데이터를 포함한 모든 항목을 제거합니다.

인스턴스가 서비스로 활성화되었거나 인스턴스가 시스템 시작 시 자동으로 시작된 경우 `dsadm delete`에는 루트 액세스가 필요합니다.

디렉토리 서버 인스턴스 시작, 중지 및 다시 시작

명령줄에서 서버를 시작, 중지 또는 다시 시작하려면 각각 `dsadm start`, `dsadm stop` 및 `dsadm restart` 명령을 사용합니다.

주 – 디렉토리 서버 인스턴스를 중지 및 다시 시작할 때 항목이 유지되도록 메모리에 대규모의 캐시가 구성된 경우 캐시가 채워질 때까지 시간이 약간 걸립니다. 캐시가 채워지는 동안 인스턴스는 더 느리게 응답합니다.

이 명령은 디렉토리 서버를 만들었던 동일한 UID 및 GID로 실행되거나 루트로 실행되어야 합니다. 예를 들어 디렉토리 서버가 `user1`로 실행되면 `start`, `stop` 및 `restart` 유틸리티를 `user1`로 실행해야 합니다.

주 – Solaris에서는 역할 기반 액세스 제어를 통해 루트 이외의 사용자로 디렉토리 서버를 실행할 수 있습니다.

▼ 디렉토리 서버를 시작, 중지 및 다시 시작하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오. 그러나 서비스

관리 활성화 및 비활성화 단계에는 적용되지 않습니다. 서비스 관리 활성화 및 비활성화는 디렉토리 서버를 시작 및 중지할 때 명령줄에서 수행해야 합니다.

- **디렉토리 서버를 시작, 중지 또는 다시 시작하려면 다음 중 하나를 수행합니다.**
  - 서버를 시작하려면 다음을 입력합니다.

    ```
    $ dsadm start instance-path
    ```

    예를 들어 인스턴스 경로가 `/local/ds`인 서버를 시작하려면 다음 명령을 사용합니다.

    ```
    $ dsadm start /local/ds
    ```
  - 서버를 중지하려면 다음을 입력합니다.

    ```
    $ dsadm stop instance-path
    ```

    예를 들면 다음과 같습니다.

    ```
    $ dsadm stop /local/ds
    ```
  - 서버를 다시 시작하려면 다음을 입력합니다.

    ```
    $ dsadm restart instance-path
    ```

    예를 들면 다음과 같습니다.

    ```
    $ dsadm restart /local/ds
    ```

접미어 만들기

디렉토리 서버 인스턴스를 만든 후에는 서버 DIT(Directory Information Tree)에 대한 접미어를 하나 이상 만들어야 합니다. DIT는 고유 이름(DN)으로 식별되는 서버 내의 모든 항목으로 구성됩니다. DN의 계층 구조적 특성으로 인해 트리의 데이터를 구조화하는 분기와 리프가 만들어집니다. DIT는 관리상 접미어 및 하위 접미어로 정의 및 관리됩니다. DSCC에서는 이러한 모든 요소를 만들고 관리하기 위한 컨트롤을 제공하지만 명령줄 도구를 사용할 수도 있습니다.

일반적으로 디렉토리 데이터 구조화 및 접미어에 대한 개념 정보는 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**를 참조하십시오.

다음 절차에 설명된 것처럼 `dsconf create-suffix` 명령을 사용하여 디렉토리에서 접미어 구성을 만들 수 있습니다. 루트 접미어와 하위 접미어는 내부적으로 동일하게 관리되기 때문에 명령줄에서 만드는 절차도 거의 동일합니다. 이 절차는 필수 옵션만을 사용한 `dsconf create-suffix` 명령을 보여줍니다. 이 명령의 다른 옵션에 대한 자세한 내용은 `dsconf`(1M) 설명서 페이지를 참조하거나 다음 명령을 실행하십시오.

```
$ dsconf create-suffix --help
```

구성 항목은 관리 사용자가 만들 수 있습니다. 그러나, 접미어의 상위 항목은 **반드시** 디렉토리 관리자가 만들거나 `cn=admin,cn=Administrators,cn=config`와 같은 디렉토리 어드민 관리자로 만들어야 합니다.

▼ 접미어를 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

DSCC를 사용하여 새 접미어를 만드는 경우 기존 접미어에서 접미어 구성 설정의 일부 또는 전체를 복사할 수 있습니다.

1 루트 접미어를 만듭니다.

서버가 실행 중인지 확인한 후 다음 명령을 입력합니다.

```
$ dsconf create-suffix -h host -p port suffix-DN
```

여기서 *suffix-DN*은 새 접미어의 전체 DN입니다. 관례상, 루트 접미어는 도메인 구성 요소(`dc`) 이름 지정 속성을 사용합니다.

예를 들어 DN이 `dc=example,dc=com`인 접미어를 만들려면 다음 명령을 사용합니다.

```
$ dsconf create-suffix -h host1 -p 1389 dc=example,dc=com
```

이 명령을 실행하면 새 접미어가 다음과 같이 만들어집니다.

- 루트 접미어의 상위 수준(또는 기본) 항목이 만들어집니다.
- 접미어와 데이터베이스 모두에 대한 `cn=config`의 구성 항목이 만들어집니다.
- 기본 데이터베이스 이름은 접미어 DN을 기준으로 합니다.

새로 만든 접미어를 비롯하여 모든 접미어에 대한 자세한 내용을 보려면 다음 명령을 사용합니다.

```
$ dsconf list-suffixes -h host -p port -v
```

`-v` 옵션을 사용하면 접미어의 항목 수를 나타내는 세부 정보 표시 모드와 복제 정보가 표시됩니다.

주 – 디렉토리 서버 인스턴스가 둘 이상인 경우 `-h` *host name* 및 `-p`*port number* 옵션을 사용하여 접미어가 속해야 하는 서버 인스턴스를 지정합니다.

데이터베이스 파일에 대한 경로를 기본값 이외의 경로로 지정하려면 `-L` 옵션을 사용합니다. 접미어 데이터베이스 경로는 이후 단계에서 변경할 수 있습니다. 이 작업을 수행하려면 `dsconf set-suffix-prop` *suffix-DN* `db-path:` *new-db-path* 명령을 사용한 다음 서버를 중지하고, 데이터베이스 파일을 수동으로 이동한 후 서버를 다시 시작합니다.

접미어를 만들 때 사용할 수 있는 모든 옵션을 보려면 `dsconf`(1M) 설명서 페이지를 참조하십시오.

2 필요한 경우 하위 접미어를 만듭니다.

```
$ dsconf create-suffix -h host -p port subSuffix-DN
```

그런 다음 하위 접미어를 루트 접미어에 연결합니다.

```
$ dsconf set-suffix-prop -h host -p port subSuffix-DN parent-suffix-dn:parentSuffix-DN
```

여기서 *parentSuffix-DN*은 이전 단계의 *suffix-DN*과 동일한 값이어야 합니다. 하위 접미어의 *suffix-DN*에는 하위 접미어의 RDN(Relative Distinguished Name) 및 부모 접미어의 DN이 포함됩니다.

예를 들어 하위 접미어 `ou=Contractors,dc=example,dc=com`을 만들고 이 하위 접미어를 루트 접미어에 연결하려면 다음을 입력합니다.

```
$ dsconf create-suffix -h host1 -p 1389 ou=Contractors,dc=example,dc=com
$ dsconf set-suffix-prop -h host1 -p 1389 ou=Contractors,dc=example,dc=com \
 parent-suffix-dn:dc=example,dc=com
```

디렉토리에 이 항목이 추가되면 서버의 데이터베이스 모듈이 자동으로 다음 디렉토리에 데이터베이스 파일을 만듭니다.

instance-path`/db/`*database-name*

여기서 *database-name*은 접미어의 일부에서 자동으로 작성되는 이름입니다. 예를 들어 이전 예에서 *database-name*은 `Contractors`가 됩니다.

3 (옵션) 데이터를 사용하여 접미어를 초기화합니다. 203 페이지 "접미어 초기화"를 참조하십시오.

접미어 비활성화 또는 활성화

유지관리를 위해 접미어를 비활성화하거나 보안상의 이유로 해당 접미어 내용을 사용할 수 없도록 설정해야 하는 경우가 있습니다. 접미어를 비활성화하면 서버에서 클라이언트 작업에 대한 응답으로 접미어 내용을 읽거나 쓸 수 없습니다. 접미어를 비활성화하면 더 이상 해당 접미어에 액세스할 수 없고 참조 모드가 자동으로 비활성 상태로 설정됩니다.

▼ 접미어를 비활성화한 후 활성화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **접미어를 비활성화합니다.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN enabled:off
```

주 – 복제된 접미어에 대한 등록 정보는 대부분 복제 메커니즘에 의해 결정되기 때문에 복제가 활성화된 접미어는 비활성화할 수 없습니다.

---

2 **접미어를 활성화합니다.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN enabled:on
```

참조 설정 및 접미어 읽기 전용 만들기

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 참조를 다른 접미어에 추가할 수 있습니다.

접미어가 복제된 환경의 소비자인 경우 복제 메커니즘이 참조 설정 값을 결정합니다. 참조 설정은 수동으로 수정할 수 있지만 다음 복제 업데이트 시 참조를 덮어씁니다. 복제 참조 설정에 대한 자세한 내용은 235 페이지 "고급 사용자 구성을 수행하는 방법"을 참조하십시오.

참조는 레이블이 지정된 URL, 즉 뒤에 공백 문자 및 레이블이 올 수 있는 LDAP URL입니다. 예를 들면 다음과 같습니다.

```
ldap://phonebook.example.com:389/
```

또는

```
ldap://phonebook.example.com:389/ou=All%20People,dc=example,dc=com
```

공백 문자는 중요하기 때문에 참조의 URL 부분에 있는 모든 공백 문자는 `%20`을 사용하여 이스케이프시켜야 합니다.

▼ 참조를 설정하여 접미어를 읽기 전용으로 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 참조 URL을 설정합니다.

```
$ dsconf set-suffix-prop -h host -p port suffix-DN referral-url:LDAP-URL
```

여기서 *LDAP-URL*은 대상의 호스트 이름, 포트 번호 및 DN이 포함된 유효한 URL입니다.

예를 들면 다음과 같습니다.

```
$ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
 referral-url:ldap://phonebook.example.com:389/
```

LDAP URL의 수를 지정할 수 있습니다.

2 참조 모드를 설정하여 접미어를 읽기 전용으로 만듭니다.

```
$ dsconf set-suffix-prop -h host -p port suffix-DN referral-mode:only-on-write
```

접미어에 대한 읽기 및 쓰기 작업을 모두 비활성화하고 모든 요청에 대해 참조를 반환하려면 `referral-mode`를 `enabled`로 설정합니다.

3 명령이 성공하면 접미어는 읽기 전용 또는 액세스 불가능 상태가 되어 참조를 반환할 준비가 됩니다.

4 (옵션) 접미어를 사용할 수 있게 되면 참조를 비활성화하여 접미어를 다시 읽기/쓰기 상태로 만듭니다.

```
$ dsconf set-suffix-prop -h host -p port suffix-DN referral-mode:disabled
```

참조가 비활성화되면 접미어의 `enabled` 등록 정보를 `off`로 설정하여 접미어 자체를 비활성화하지 않는 한 접미어는 자동으로 읽기/쓰기 상태가 됩니다.

접미어 삭제

접미어를 삭제하면 DIT에서 전체 분기가 제거됩니다.

주 – 접미어를 삭제하면 디렉토리에서 모든 데이터 항목이 영구적으로 제거됩니다. 또한, 해당 복제 구성을 비롯한 모든 접미어 구성 정보도 제거됩니다.

부모 접미어는 삭제할 수 없으며 이에 따라 DIT에서 해당 하위 접미어를 새 루트 접미어로 유지할 수 없습니다. 하위 접미어가 포함된 분기 전체를 삭제하려면 삭제된 부모의 하위 접미어와 자신의 하위 접미어도 삭제해야 합니다.

▼ 접미어를 삭제하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **접미어 구성 항목을 제거합니다.**

  ```
  $ dsconf delete-suffix -h host -p port [subSuffix-DN] suffix-DN
  ```

  이 명령은 *suffix-DN*의 기본 항목에서 시작하여 접미어를 서버에서 제거합니다. 이 접미어는 더 이상 디렉토리에서 보거나 액세스할 수 없습니다.

접미어 압축

Directory Server 6.2는 접미어의 오프라인 압축을 지원합니다. 온라인 압축은 이 릴리스에서 지원되지 않습니다. 저장소 공간을 사용할 수 있는 경우 접미어를 압축하면 데이터베이스 키가 다시 구성되어 데이터베이스 크기가 줄어듭니다.

▼ 접미어를 오프라인으로 압축하는 방법

이 작업을 수행하기 전에 서버를 중지하고 데이터베이스를 백업합니다.

- **필요한 접미어를 압축합니다.**

  ```
  $ dsadm repack instance-path suffix-dn
  ```

  지정된 접미어와 관련된 모든 `.db3` 파일이 압축됩니다.

  `-b` 옵션과 함께 이 명령을 실행하는 경우 접미어 DN 대신 백엔드 데이터베이스 이름을 지정할 수 있습니다. 하나 이상의 접미어 또는 백엔드를 지정해야 합니다.

◆ ◆ ◆ 3 장

3 디렉토리 서버 구성

이 장에서는 디렉토리 서버 구성 방법에 대해 설명합니다. `dsconf` 명령을 사용할 수 있습니다(`dsconf`(1M) 설명서 페이지 참조).

기본 방법인 디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)를 사용할 수도 있습니다. DSCC는 구성 프로세스 중에 추가 검사 작업을 수행하여 오류를 최소화할 수 있습니다. 또한 DSCC를 사용하면 서버 인스턴스의 구성을 다른 서버 인스턴스로 복사할 수 있습니다. DSCC 사용에 대한 자세한 내용은 DSCC 온라인 도움말을 참조하십시오.

디렉토리 서버 인스턴스의 구성 표시

디렉토리 서버 인스턴스의 구성을 표시하려면 `dsconf info`를 실행합니다.

```
$ dsconf info -h host -p port
Instance path   :  instance path
Global State    :  read-write
Host Name       :  host
Port            :  port
Secure port     :  secure port
Total entries   :  20844

Suffixes        :  suffix-DN

Dest. Servers   :  host:port

On-Going Tasks  :  import
Finished Tasks  :  backup
```

위의 출력은 사용자가 대상 서버에서 접미어와 복제 계약을 만들었다고 가정합니다. 또한 진행 중인 가져오기 작업과 완료된 백업 작업이 표시됩니다.

DSCC를 사용하여 구성 수정

구성을 수정하려면 DSCC를 사용하는 것이 좋습니다. 이 브라우저 인터페이스는 구성을 신속하고 효율적으로 설정할 수 있도록 도와주는 작업 기반의 컨트롤을 제공합니다. DSCC를 사용하면 한 서버에서 구성 설정을 수정한 다음 해당 구성 설정을 다른 서버로 복사할 수 있습니다. 또한 DSCC 인터페이스는 사용자를 대신하여 복잡하고 상호 종속된 구성을 관리합니다. DSCC를 사용하여 구성을 수정하는 자세한 절차는 DSCC 온라인 도움말을 참조하십시오.

명령줄에서 구성 수정

명령줄 도구를 사용하는 스크립트를 작성하여 구성 작업을 자동화할 수 있습니다.

`dsconf` 명령을 사용하여 명령줄에서 구성을 수정합니다. 이 명령은 LDAP를 사용하여 `cn=config` 하위 트리를 수정합니다. `dsconf`에 대한 자세한 내용은 49 페이지 "디렉토리 서버 명령줄 도구"를 참조하십시오.

`dsconf`를 사용하여 수행할 수 없는 작업에서는 `ldapmodify` 명령을 사용합니다.

주 – `dsconf set-server-prop` 명령을 사용하여 서버 구성 등록 정보를 수정하려면 수정 가능한 등록 정보와 해당 기본값에 대해 알고 있어야 합니다. 모든 등록 정보에 대한 도움말을 표시하려면 다음 명령을 사용합니다.

```
$ dsconf help-properties -v
```

등록 정보 도움말에서 원하는 항목을 검색합니다. 예를 들어 UNIX 플랫폼에서 다음을 입력하여 메모리 캐시 등록 정보를 검색합니다.

```
$ dsconf help-properties -v | grep cache
```

`cn=config`의 구성 항목과 허용되는 값 범위를 포함한 모든 구성 항목 및 속성에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

dse.ldif 파일 수정

디렉토리 서버는 모든 구성 정보를 아래 파일에 저장합니다.

instance-path`/config/dse.ldif`

주의 – `dse.ldif` 파일의 내용을 직접 편집하여 구성을 수정할 경우 오류 발생 가능성이 커지므로 바람직하지 않습니다. 이 파일을 수동으로 편집하려면 파일을 편집하기 전에 서버를 중지하고 편집을 마친 후에 다시 시작합니다.

`dse.ldif` 파일은 LDIF(LDAP Data Interchange Format) 형식입니다. LDIF는 항목, 속성 및 해당 값을 텍스트로 표현하며 RFC 2849(`http://www.ietf.org/rfc/rfc2849`)에 설명된 표준 형식입니다.

`dse.ldif` 파일의 디렉토리 서버 구성은 다음과 같습니다.

- `cn=config` 항목의 속성과 값
- `cn=config` 아래의 하위 트리에 있는 모든 항목 및 해당 속성과 값
- 루트 항목("") 및 `cn=monitor` 항목의 객체 클래스와 액세스 제어 지침. 이러한 항목의 다른 속성은 서버에서 생성됩니다.

  디렉토리 서버 인스턴스를 소유한 시스템 사용자만 파일을 읽고 쓰는 권한이 있습니다.

  디렉토리 서버에서는 LDAP를 통해 모든 구성 설정을 읽고 쓸 수 있습니다. 기본적으로 디렉토리의 `cn=config` 분기는 인증된 모든 사용자가 읽을 수 있지만, 쓰기는 디렉토리 관리자(`cn=Directory Manager`)와 `cn=Administrators,cn=config` 아래의 관리 권한이 있는 사용자에게만 허용됩니다. 관리 사용자는 구성 항목을 다른 디렉토리 항목처럼 보고 수정할 수 있습니다.

  `cn=config` 항목 아래에 비구성 항목을 작성할 경우 일반 항목보다 확장성이 낮은 데이터베이스인 `dse.ldif` 파일에 항목이 저장되므로 바람직하지 않습니다. 따라서 많은 항목, 특히 자주 업데이트되는 항목을 `cn=config`에 저장하면 성능이 저하될 수 있습니다. 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.

관리 사용자 구성

디렉토리 서버에는 기본 관리 사용자, 디렉토리 관리자 및 `cn=admin,cn=Administrators,cn=config` 사용자가 있습니다. 이러한 모든 사용자는 동일한 액세스 권한을 갖지만 `cn=admin,cn=Administrators,cn=config`는 ACI를 따릅니다.

이 절에서는 루트 액세스 권한이 있는 관리 사용자를 만드는 방법과 디렉토리 관리자를 구성하는 방법에 대해 설명합니다.

▼ 루트 액세스 권한이 있는 관리 사용자를 만드는 방법

cn=admin,cn=Administrators,cn=config와 동일한 권한을 가진 새 관리 사용자를 만들려면 cn=Administrators,cn=config 그룹에서 새 사용자를 만듭니다. 이 그룹의 모든 사용자는 디렉토리 관리자와 동일한 액세스 권한이 허용되는 전역 ACI를 따릅니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **새 관리 사용자를 만듭니다.**

  예를 들어 cn=Admin24,cn=Administrators,cn=config 새 사용자를 만들려면 다음을 입력합니다.

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=admin24,cn=Administrators,cn=config
changetype: add
objectclass: top
objectclass: person
userPassword: password
description: Administration user with the same access rights as Directory Manager.
```

  -D 옵션과 -w 옵션은 각각 이 항목을 작성할 수 있는 권한이 있는 사용자의 바인드 DN과 비밀번호를 제공합니다.

▼ 디렉토리 관리자를 구성하는 방법

디렉토리 관리자는 UNIX 시스템의 root 사용자와 비슷한 권한이 있는 서버 관리자입니다. 액세스 제어는 디렉토리 관리자에게 적용되지 않습니다.

대부분의 관리 작업에서는 디렉토리 관리자를 사용할 필요가 없습니다. 대신 cn=admin,cn=Administrators,cn=config 사용자나 cn=Administrators,cn=config 아래에 만든 다른 사용자를 사용할 수 있습니다. 복제 복구, tombstone 검색 등과 같은 복제 문제 해결 작업과 루트 ACI 변경 작업에서만 디렉토리 관리자가 필요합니다.

디렉토리 관리자 DN과 비밀번호를 변경하고 비밀번호를 자동으로 읽어올 수 있는 파일을 만들 수도 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1. **기존 디렉토리 관리자 DN을 찾습니다.**

```
$ dsconf get-server-prop -h host -p port root-dn
root-dn:cn=Directory Manager
```

2. **필요한 경우 디렉토리 관리자 설정을 수정합니다.**

   - 디렉토리 관리자 DN을 수정하려면 다음을 입력합니다.

```
$ dsconf set-server-prop -h host -p port root-pwd-file:new-root-dn-password-file
```

디렉토리 관리자 DN에 공백이 있는 경우 따옴표를 사용합니다. 예를 들면 다음과 같습니다.

```
$ dsconf set-server-prop -h host1 -p 1389 root-dn:"cn=New Directory Manager"
```

- 디렉토리 관리자 비밀번호를 변경하려면 다음을 입력합니다.

```
$ dsconf set-server-prop -h host -p port root-pwd:new-root-dn-password
```

보안상 일반 텍스트 비밀번호를 명령줄 인수로 전달하지 않으려면 비밀번호 설정을 위한 임시 파일을 만듭니다.

```
$ echo password > /tmp/pwd.txt
```

이 파일을 읽고 나중에 사용하기 위해 비밀번호를 저장합니다. 서버 루트 비밀번호 파일 등록 정보를 설정합니다.

```
$ dsconf set-server-prop -h host -p port root-pwd-file:/tmp/pwd.txt
```

이 명령은 서버에 비밀번호 파일을 읽으라는 메시지를 표시합니다. 비밀번호 파일 등록 정보를 설정한 후 임시 비밀번호 파일을 제거합니다.

```
$ rm /tmp/pwd.txt
```

구성 정보 보호

루트 디렉토리 서버 항목(길이가 0인 DN ""을 사용한 기본 객체 검색에서 반환되는 항목)과 `cn=config`, `cn=monitor` 및 `cn=schema` 아래의 하위 트리에는 디렉토리 서버에서 자동으로 생성되는 액세스 제어 지침(ACI)이 포함되어 있습니다. 이러한 ACI는 디렉토리 항목에 대한 사용자 비밀번호를 결정하는 데 사용됩니다. 이 ACI는 평가용으로 충분합니다. 그러나 작업 환경 배포를 위해서는 액세스 제어 요구 사항을 평가하여 자체 액세스 제어를 설계해야 합니다.

보안상 하나 이상의 추가 하위 트리를 숨겨서 구성 정보를 보호하려면 DIT에 추가 ACI를 저장해야 합니다.

- 숨기려는 하위 트리의 맨 밑에 있는 항목에 ACI 속성을 저장합니다.
- `namingContexts` 속성의 루트 DSE 항목에 ACI를 저장합니다. `namingContexts`라는 루트 DSE 항목 속성에는 각 디렉토리 서버 데이터베이스에 대한 기본 DN 목록이 포함되어 있습니다.
- `cn=config` 및 `cn=monitor` 하위 트리에 ACI를 저장합니다. 하위 트리 DN은 `cn=config` 및 `cn=monitor` 아래의 매핑 트리 항목에도 저장됩니다.

ACI 만들기에 대한 자세한 내용은 6 장을 참조하십시오.

DSCC 구성

이 절에서는 DSCC 구성에 관한 다음 정보에 대해 설명합니다.

▼ 일반 에이전트 컨테이너 포트 번호를 변경하는 방법

일반 에이전트 컨테이너 포트 번호 기본값은 11162입니다. 일반 에이전트 컨테이너는 DSCC 에이전트 포트를 `jmxmp-connector-port`로 정의합니다. 관리상 DSCC 에이전트 및 일반 에이전트 컨테이너에 다른 포트 번호를 사용해야 하는 경우 다음 절차를 수행합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 루트로 `jmxmp-connector-port`에 대한 기존 포트 번호를 확인합니다.

```
$ su
Password:
# cacaoadm list-params
...
jmxmp-connector-port=11162
...
```

2 DSCC 에이전트 포트 번호를 변경합니다.

DSCC 에이전트 포트 번호를 변경할 때 일반 에이전트 컨테이너를 중지해야 합니다.

```
# cacaoadm stop
# cacaoadm set-param jmxmp-connector-port=new-port
# cacaoadm start
```

이 명령의 위치는 34 페이지 “명령 위치”를 참조하십시오.

3 DSCC에서 서버를 등록 취소한 다음 새 DSCC 에이전트 포트 번호를 사용하여 다시 등록합니다.

또한 새 서버를 만들 경우 기본값이 아닌 DSCC 에이전트 포트 번호를 지정해야 합니다.

▼ 디렉토리 서비스 관리자 비밀번호를 재설정하는 방법

디렉토리 서비스 관리자 비밀번호를 재설정하려면 이 절차에 설명된 것처럼 DSCC를 사용합니다.

1 **44 페이지 "DSCC에 액세스하는 방법"에 설명된 것처럼 DSCC에 액세스합니다.**

2 **설정 탭을 클릭한 다음 디렉토리 서비스 관리자를 선택합니다.**

3 **비밀번호를 변경할 디렉토리 서비스 관리자의 이름을 클릭합니다.**

4 **등록 정보 화면에 새 비밀번호를 입력합니다.**
비밀번호 확인 필드에 새 비밀번호를 다시 입력하여 확인합니다. 확인을 눌러 변경 사항을 저장합니다.

▼ DSCC 세션 자동 시간 초과 지연을 확장하는 방법

일정 시간 후에 DSCC 세션 시간이 초과되고 DSCC에서 로그아웃됩니다. 시간 초과 지연을 확장하려면 다음 절차를 수행합니다. 이 절차에서는 DSCC와 Sun Java Web Console의 모든 다른 응용 프로그램에 대한 시간 초과를 확장합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 **루트로 시간 초과 지연을 확장합니다.**

```
# wcadmin add -p -a ROOT session.timeout.value=mm
```

여기서 *mm*은 시간이 초과되기 전의 시간(분)입니다.

예를 들어 시간 초과를 2시간으로 설정하려면 다음을 입력합니다.

```
$ su
Password:
# wcadmin add -p -a ROOT session.timeout.value=120
Set 1 properties for the ROOT application.
# wcadmin list -p
Shared service properties (name, value):
    session.timeout.value  120
    ...
```

2 **Sun Java Web Console을 다시 시작합니다.**

```
# smcwebserver restart
Shutting down Sun Java(TM) Web Console Version 3.0.2 ...
```

```
Starting Sun Java(TM) Web Console Version 3.0.2 ...
The console is running.
```

이 명령의 위치는 34 페이지 “명령 위치”를 참조하십시오.

DSCC에 대한 페일오버 구성

DSCC는 DSCC에 등록된 서버를 표시합니다.

DSCC를 설치한 시스템이 실패할 경우 DSCC를 다른 시스템에 설치한 다음 서버를 다시 등록할 수 있습니다. 그러나 이 방법은 시간이 오래 걸릴 수 있습니다. DSCC를 통해 서버에 즉시 액세스하려면 DSCC 페일오버를 구성할 수 있습니다.

DSCC 페일오버를 구성하려면 다음을 고려하십시오.

- 등록된 서버에 대한 모든 정보가 DSCC 레지스트리에 저장됩니다. 이 레지스트리는 디렉토리 서버 인스턴스입니다. `dsadm` 및 `dsconf` 관리 명령을 사용하여 레지스트리를 관리할 수 있습니다.
- DSCC 레지스트리의 기본 특징은 다음과 같습니다.

서버 인스턴스	Solaris — `/var/opt/SUNWdsee/dscc6/dcc/ads` Linux 및 HP-UX — `/var/opt/sun/dscc6/dcc/ads` Windows — `C:\Program Files\Sun\DSEE\var\dscc6\dcc\ads`
접미어	`cn=dscc`
포트	LDAP `3998`, LDAPS `3999`

- DSCC를 두 대 이상의 컴퓨터에 설치한 후 DSCC 레지스트리 접미어 간의 복제를 설정할 수 있습니다. 10 장에 설명된 복제 명령줄 절차를 사용합니다. 간단한 복제 구성 설정에 대한 예는 `dsconf`(1M) 설명서 페이지를 참조하십시오.

  복제를 설정한 이후에는 DSCC에 등록된 서버를 다른 컴퓨터에서 액세스할 수 있습니다. 예를 들어 `host1`과 `host2` 사이에 DSCC 레지스트리 접미어 복제를 설정한 경우 `https://host1:6789` 또는 `https://host2:6789`에서 DSCC를 사용하여 동일한 서버를 관리할 수 있습니다. 호스트가 실패할 경우 다른 호스트에서 DSCC에 액세스합니다.

DSCC 문제 해결

DSCC 문제 해결에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide**의 “To Troubleshoot Directory Service Control Center Access”를 참조하십시오.

디렉토리 서버의 포트 번호 변경

DSCC 또는 `dsconf set-server-prop` 명령을 사용하여 사용자 디렉토리 서버의 LDAP 포트 또는 LDAPS 보안 포트 번호를 수정할 수 있습니다.

포트 번호를 변경하는 경우 다음에 주의해야 합니다.

- 권한이 없는 포트 번호를 설정하고, 디렉토리 서버를 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. 그러면 이 응용 프로그램에서 디렉토리 서버에 대한 요청을 처리할 수 있게 됩니다. 즉, 이 응용 프로그램을 사용하여 인증 프로세스에 사용된 비밀번호를 수집하거나, 서버 응답에 대한 클라이언트 요청을 변경하거나, 서비스 거부 공격을 생성할 수 있습니다. 이러한 보안 위험을 방지하려면 `listen-address` 또는 `secure-listen-address` 속성을 사용하여 디렉토리 서버가 수신하는 인터페이스(주소)를 지정합니다.

명령줄을 사용하여 포트 번호를 변경할 경우 다음에 주의해야 합니다.

- 디렉토리 서버가 다른 서버에 정의되어 있는 복제 계약에 언급되어 있는 경우 해당 복제 계약을 업데이트하여 새 포트 번호를 사용하도록 해야 합니다.
- 이전에 DSCC를 사용하여 서버를 관리한 경우 포트 번호를 변경하고 나면 서버를 일시적으로 볼 수 없습니다. 서버를 다시 보려면 서버를 등록 해제한 다음 새 포트 번호를 사용하여 DSCC에서 다시 등록해야 합니다.

▼ 포트 번호를 수정하고 포트를 사용 가능/사용 불가능하게 하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

주 – 수정한 후에는 서버를 다시 시작하여 변경 사항을 적용해야 합니다.

---

1 포트에 대한 기존 설정을 확인합니다.

```
$ dsconf get-server-prop -h host -p port port-type
```

여기서 *port-type*은 다음 중 하나입니다.

`ldap-port`	LDAP 기본 포트
`ldap-secure-port`	LDAPS 보안 포트
`dsml-port`	DSML 기본 포트

`dsml-secure-port` DSML 보안 포트

예를 들어 LDAPS 보안 포트를 표시하려면 다음을 입력합니다.

```
$ dsconf get-server-prop -h host1 -p 2501 ldap-secure-port
Enter "cn=Directory Manager" password:
ldap-secure-port : 2511
```

반환된 결과가 정수이면 포트가 사용 가능하게 됩니다. 반환된 결과가 `disabled`이면 포트가 사용 불가능하게 됩니다.

주 – `dsadm`을 사용하여 LDAP 기본 포트와 LDAPS 보안 포트를 나열할 수도 있습니다.

---

2 필요한 경우 포트 번호를 수정하거나 포트를 사용 가능하게 합니다.

```
$ dsconf set-server-prop -h host -p port port-type:new-port
```

예를 들어 LDAP 포트 번호를 1389에서 1390으로 변경하려면 다음 명령을 사용합니다.

```
$ dsconf set-server-prop -h host1 -p 1389 ldap-port:1390
```

DSML 보안 포트를 포트 번호 2250에서 사용 가능하게 하려면 다음 명령을 사용합니다.

```
$ dsconf set-server-prop -h host1 -p 1389 dsml-secure-port:2250
```

3 필요한 경우 포트를 사용 불가능하게 합니다.

```
$ dsconf set-server-prop -h host -p port port-type:disabled
```

예를 들어 DSML 보안 포트를 사용 불가능하게 하려면 다음 명령을 사용합니다.

```
$ dsconf set-server-prop -h host1 -p 1389 dsml-secure-port:disabled
```

DSML 구성

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

디렉토리 서버는 HTTP(Hypertext Transfer Protocol/1.1)를 통해 DSMLv2를 지원하며 SOAP(Simple Object Access Protocol) 버전 1.1을 DSML 내용 전송을 위한 프로그래밍

프로토콜로 사용합니다. 이러한 프로토콜에 대한 자세한 내용과 DSML 요청에 대한 예는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 10 장, "Directory Server DSMLv2"를 참조하십시오.

이 절은 다음 내용으로 구성되어 있습니다.

- 77 페이지 "DSML-over-HTTP 서비스를 사용 가능하게 하는 방법"
- 78 페이지 "DSML-over-HTTP 서비스를 사용 불가능하게 하는 방법"
- 79 페이지 "DSML 아이디 매핑"

▼ DSML-over-HTTP 서비스를 사용 가능하게 하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 DSML 모드를 on으로 설정합니다.

```
$ dsconf set-server-prop -h host -p port dsml-enabled:on
```

2 보안 DSML 포트를 설정합니다.

```
$ dsconf set-server-prop -h host -p port dsml-secure-port:port
```

3 비보안 DSML 포트를 설정합니다.

```
$ dsconf set-server-prop -h host -p port dsml-port:port
```

기본적으로 이 포트는 `disabled`로 설정됩니다.

4 서버를 다시 시작합니다.

```
$ dsadm restart instance-path
```

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

`http://`*host*`:`*DSML-port*`/ `*relative-URL*

`https://`*host*`:`*secure-DSML-port*` /`*relative-URL*

주 – *relative-URL*은 `dsml-relative-root-url` 등록 정보를 사용하여 읽고 설정할 수 있습니다.

▼ DSML-over-HTTP 서비스를 사용 불가능하게 하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 **DSML 모드를 `off`로 설정합니다.**

```
$ dsconf set-server-prop -h host -p port dsml-enabled:off
```

2 **보안 DSML 포트를 `disabled`로 설정합니다.**

```
$ dsconf set-server-prop -h host -p port dsml-secure-port:disabled
```

3 **서버를 다시 시작합니다.**

```
$ dsasm restart instance-path
```

▼ DSML 보안을 구성하는 방법

DSML 요청을 승인하는 데 필요한 보안 수준을 구성할 수 있습니다. 이 작업을 수행하려면 DSML 클라이언트 인증을 구성해야 합니다.

● **DSML 클라이언트 인증 모드를 설정합니다.**

```
$ dsconf set-server-prop -h host -p port dsml-client-auth-mode:dsml-mode
```

기본적으로 `dsml-client-auth-mode` 등록 정보는 `client-cert-first`로 설정됩니다.

*dsml-mode*는 다음 중 하나입니다.

- `http-basic-only` - 기본값입니다. 서버에서 HTTP 인증 헤더의 내용을 사용하여 디렉토리 항목에 매핑할 수 있는 사용자 이름을 찾습니다. 이 프로세스와 해당 구성은 SSL을 통해 암호화되지만 클라이언트 인증은 사용되지 않습니다. 자세한 내용은 79 페이지 “DSML 아이디 매핑”을 참조하십시오.
- `client-cert-only` - 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 5 장을 참조하십시오.
- `client-cert-first` - 서버에서 먼저 제공된 클라이언트 인증서를 사용하여 클라이언트 인증을 시도합니다. 클라이언트 인증서가 제공되지 않은 경우 서버는 인증 헤더의 내용을 사용하여 클라이언트를 인증합니다.

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

- `client-cert-only` 값을 지정할 때 클라이언트에서 인증 헤더만 제공하고 인증서는 제공하지 않는 경우
- `http-basic-only` 값을 지정할 때 클라이언트에서 인증서만 제공하고 인증 헤더는 제공하지 않는 경우

인증서가 제공되었지만 항목에 일치시킬 수 없는 경우나 HTTP 인증 헤더가 지정되었지만 사용자 항목으로 매핑할 수 없는 경우에는 클라이언트 인증 방법에 상관없이 오류 메시지 403: "금지됨"이 표시되며 DSML 요청이 거부됩니다.

DSML 아이디 매핑

인증서 없이 기본 인증을 수행하는 경우 디렉토리 서버는 **아이디 매핑**이라는 메커니즘을 사용하여 DSML 요청을 승인할 때 사용할 바인드 DN을 지정합니다. 이 기법은 HTTP 요청의 인증 헤더에서 필요한 정보를 추출하여 바인드에 사용할 아이디를 확인합니다.

DSML/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
dsSearchFilter: (uid=${Authorization})
```

이 구성에서 서버는 디렉토리 서버 접미어에 저장된 DN에 대한 `uid` 값으로 HTTP 사용자 아이디를 사용해야 합니다. 예를 들어 HTTP 사용자가 `bjensen`인 경우 서버는 `uid=bjensen,ou=people` DN을 사용하여 바인딩을 실행하려고 합니다.

매핑이 제대로 작동하려면 `dsSearchBaseDN` 값을 완료해야 합니다. 예를 들어 `dsSearchBaseDN` 값을 `ou=people,dc=example,dc=com`으로 변경할 수 있습니다. HTTP 사용자가 `bjensen`인 경우 서버는 `uid=bjensen,ou=people,dc=example,dc=com` DN을 사용하여 바인딩을 실행하려고 합니다.

```
dn: cn=default,cn=HTTP-BASIC,cn=identity mapping,cn=config
objectClass: top
objectClass: nsContainer
objectClass: dsIdentityMapping
cn: default
dsSearchBaseDN: ou=people,dc=example,dc=com
dsSearchFilter: (uid=${Authorization})
```

dsSearchFilter 매핑 항목 속성에 ${*header*} 형식의 자리 표시자를 사용할 수도 있습니다. 여기서 *header*는 HTTP 헤더의 이름입니다.

DSML 매핑에 사용되는 가장 일반적인 헤더는 다음과 같습니다.

${Authorization}	이 문자열은 HTTP 인증 헤더에 포함된 사용자 이름으로 바뀝니다. 인증 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 서버의 호스트 이름 및 포트 번호로 바뀝니다.

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

▼ HTTP 헤더에 대해 새 아이디 매핑을 정의하는 방법

1 **기본 DSML-over-HTTP 아이디 매핑을 편집하거나 이 프로토콜에 대한 사용자 정의 매핑을 작성합니다.**

매핑 항목은 cn=HTTP-BASIC,cn=identity mapping,cn=config 항목 아래에 있어야 합니다.

88 페이지 “ldapmodify를 사용한 항목 추가”에 설명된 것처럼 ldapmodify 명령을 사용하여 명령줄에서 이 항목을 추가합니다.

2 **새 매핑을 적용하려면 디렉토리 서버를 다시 시작합니다.**

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

서버를 읽기 전용으로 설정

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

서버 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 서버 읽기 전용 모드가 적용되지 않습니다.

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

이러한 분기는 읽기 전용으로 설정되어 있지 않더라도 관리자 이외의 사용자가 수정할 수 없도록 항상 ACI(Access Control Intstruction)를 사용하여 보호해야 합니다(6 장 참조). 전역 읽기 전용 모드는 디렉토리 관리자가 시작한 업데이트 작업을 포함하여 디렉토리의 다른 모든 접미사에 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 사용 불가능하게 할 때까지 소비자 복제본도 업데이트를 받지 못합니다. 다중 마스터 복제 환경에서는 마스터에 복제할 변경 사항이 추가되지 않을 뿐만 아니라 다른 마스터의 업데이트를 받을 수도 없습니다.

▼ 서버 읽기 전용 모드를 사용 가능/사용 불가능하게 하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 전역 읽기 전용 모드를 사용 가능하게 합니다.

```
$ dsconf set-server-prop -h host -p port read-write-mode:read-only
```

2 준비가 되면 읽기 전용 모드를 사용 불가능하게 합니다.

```
$ dsconf set-server-prop -h host -p port read-write-mode:read-write
```

메모리 구성

이 절에서는 다양한 유형의 메모리를 관리하는 방법에 대해 설명합니다. 다양한 유형의 캐시와 캐시 조정에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 5 장, “Directory Server Data Caching”을 참조하십시오.

캐시 초기화

캐시 초기화란 이후의 디렉토리 서버 동작이 램프 업(ramp up)이 아니라 정상적인 작동 성능을 반영하도록 캐시에 데이터를 채우는 것을 의미합니다. 캐시 초기화를 사용하면 벤치마킹에서 재생 가능한 결과에 도달하고 잠재적인 최적화를 측정 및 분석할 수 있습니다.

가능하면 캐시를 초기화하지 마십시오. 성능을 측정하기 전에 디렉토리 서버와의 일반적인 상호 작용을 통해 캐시가 초기화되도록 합니다.

데이터베이스 캐시 초기화 도구는 http://www.slamd.com에서 확인할 수 있습니다.

▼ 데이터베이스 캐시를 수정하는 방법

주의 – 캐시를 수정하면 서버 성능에 심각한 영향을 줄 수 있습니다. 따라서 캐시를 수정할 경우 주의하십시오.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **현재 데이터베이스 캐시 수준을 확인합니다.**

   ```
   $ dsconf get-server-prop -h host -p port db-cache-size
   ```

2. **데이터베이스 캐시 수준을 변경합니다.**

   ```
   $ dsconf set-server-prop -h host -p port db-cache-size:size
   ```

   여기서 *size*는 GB(G), MB(M), KB(k) 또는 바이트(b)로 표시할 수 있습니다. 사용자가 지정한 크기는 시스템에서 지원해야 합니다.

▼ 데이터베이스 캐시를 모니터하는 방법

설치 시 기본 캐시 수준은 작업 환경이 아니라 테스트 환경에 적합합니다. 조정을 위해 서버의 데이터베이스 캐시를 모니터할 수 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **데이터베이스 캐시를 모니터합니다.**

  ```
  $ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
   -b "cn=monitor,cn=ldbm database,cn=plugins,cn=config" "(objectclass=*)"
  ```

데이터베이스 캐시가 충분히 크고 캐시가 초기화되면 적중률(dbcachehitratio)이 높아야 합니다. 또한 읽은 페이지 수(dbcachepagein)와 기록되는 클린 페이지 수(dbcacheroevict)는 작아야 합니다. 여기서 "높음"과 "낮음"은 배포 제약 조건에 상대적인 의미입니다.

▼ 항목 캐시를 모니터하는 방법

조정을 위해 항목 캐시에서 하나 이상의 접미어를 확인할 수 있습니다. 항목 캐시 수준을 보려면 다음 절차를 수행합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **항목 캐시를 모니터합니다.**

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=monitor,cn=db-name,cn=ldbm database,cn=plugins,cn=config" "(objectclass=*)"
```

접미어를 위한 항목 캐시가 접미어에 있는 대부분의 항목을 보관하는 데 충분한 크기이고 캐시가 초기화되면 적중률(entrycachehitratio)이 높아야 합니다.

캐시를 초기화한 경우 이전에 비어 있던 항목 캐시가 채워질수록 항목 캐시 크기(currententrycachesize)는 최대 항목 캐시 크기(maxentrycachesize)에 근접하게 됩니다. 이상적인 항목의 크기(currententrycachecount)는 접미어에 있는 총 항목 수(ldapentrycachecount)와 같거나 비슷해야 합니다.

▼ 항목 캐시를 수정하는 방법

주의 – 캐시를 수정하면 서버 성능에 심각한 영향을 줄 수 있습니다. 따라서 캐시를 수정할 경우 주의하십시오.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1. **현재 항목 캐시 수준을 확인합니다.**

```
$ dsconf get-suffix-prop -h host -p port suffix-DN entry-cache-count entry-cache-size
```

2. **항목 캐시 횟수를 변경합니다.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN entry-cache-count:integer
```

여기서 *integer*는 캐시에 저장할 항목 수입니다.

3 **항목 캐시 크기를 변경합니다.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN entry-cache-size:size
```

여기서 *size*는 GB(G), MB(M), KB(k) 또는 바이트(b)로 표시되는 캐시 크기입니다. 사용자가 지정한 크기는 시스템에서 지원해야 합니다.

▼ 힙 메모리 임계값을 구성하는 방법

nsslapd 프로세스에서 사용되는 힙 메모리의 양을 제한하려면 동적 메모리 범위에 대한 임계값을 설정할 수 있습니다. 자원이 공유되거나 부족한 컴퓨터에서 디렉토리 서버를 실행하는 경우 이 임계값을 설정할 수 있습니다.

주 – 이 임계값은 Solaris 및 Linux 플랫폼에서만 설정할 수 있습니다.

메모리 크기 지정에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 "Directory Server and Memory"를 참조하십시오.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

주 – 기본적으로 heap-high-threshold-size 및 heap-low-threshold-size 등록 정보는 undefined입니다.

---

1 **최대 힙 메모리의 높은 임계값을 설정합니다.**

```
$ dsconf set-server-prop -h host -p port heap-high-threshold-size:value
```

여기서 *value*는 GB(G), MB(M), KB(k) 또는 바이트(b)로 표시되는 undefined 또는 메모리 크기입니다. 사용자가 지정한 크기는 시스템에서 지원해야 합니다.

heap-high-threshold-size에 대한 권장 사용 값은 server(5dsconf) 설명서 페이지를 참조하십시오.

2 **선택 사항으로 최대 힙 메모리의 낮은 임계값을 설정합니다.**

```
$ dsconf set-server-prop -h host -p port heap-low-threshold-size:value
```

여기서 *value*는 GB(G), MB(M), KB(k) 또는 바이트(b)로 표시되는 undefined 또는 메모리 크기입니다. 사용자가 지정한 크기는 시스템에서 지원해야 합니다.

heap-low-threshold-size에 대한 권장 사용 값은 server(5dsconf) 설명서 페이지를 참조하십시오.

각 클라이언트 계정에 대한 자원 제한 설정

각 클라이언트 계정에 대한 서버의 검색 작업 자원 제한을 제어할 수 있습니다. 계정의 작업 속성에서 이러한 제한을 설정하면 디렉토리 서버는 클라이언트에서 디렉토리에 바인드하는 데 사용하는 계정을 기반으로 해당 제한을 적용합니다.

다음과 같은 제한을 설정할 수 있습니다.

- 조회 제한은 검색 작업 시 조사할 최대 항목 수를 지정합니다.
- 크기 제한은 검색 작업에 대해 반환되는 최대 항목 수를 지정합니다.
- 시간 제한은 서버에서 검색 작업 처리에 사용할 수 있는 최대 시간을 지정합니다.
- 유휴 시간 초과는 연결이 끊기기 전에 클라이언트 연결이 유휴 상태로 유지될 수 있는 최대 시간을 지정합니다.

주 – 기본적으로 디렉토리 관리자가 사용할 수 있는 자원에는 제한이 없습니다.

특정 사용자 계정에 대해 설정한 자원 제한은 서버측 구성에 설정된 자원 제한보다 우선적으로 적용됩니다. 이 절에서는 각 계정에 대한 자원 제한 설정에 대해 설명합니다.

이 절에 제공된 예에서는 항목 속성에서 자원 제한을 직접 설정합니다. 서비스 클래스(CoS) 메커니즘을 사용하여 계정에 대한 자원 제한을 설정할 수도 있습니다. CoS 메커니즘은 클라이언트 응용 프로그램에 대한 항목이 검색될 때 계산된 속성을 생성합니다. CoS 정의에 대한 자세한 내용은 218 페이지 “서비스 클래스”를 참조하십시오.

▼ 힙 메모리 임계값을 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 `dsconf get-server-prop` **명령을 사용하여 자원 제한 서버 등록 정보를 읽습니다.**

```
$ dsconf get-server-prop -h host -p port look-through-limit search-size-limit \
 search-time-limit idle-timeout
look-through-limit  :  5000
search-size-limit   :  2000
search-time-limit   :  3600
idle-timeout        :  none
```

출력은 검색에서 최대 5000개의 항목을 조회하여 최대 2000개의 항목을 반환하며, 서버에서 검색을 처리하는 데 최대 1시간(3600초)이 소요됨을 나타냅니다.

2 조회 제한을 변경합니다.

```
$ dsconf set-server-prop -h host -p port look-through-limit:integer
```

여기서 *integer*는 검색 작업 시 조사할 최대 항목 수입니다.

3 검색 크기 제한을 변경합니다.

```
$ dsconf set-server-prop -h host -p port search-size-limit:integer
```

여기서 *integer*는 검색 작업에서 반환되는 최대 항목 수입니다.

4 검색 시간 제한을 변경합니다.

```
$ dsconf set-server-prop -h host -p port serach-time-limit:integer
```

여기서 *integer*는 검색 작업 처리에 사용할 수 있는 최대 시간입니다.

5 유휴 시간 초과를 변경합니다.

```
$ dsconf set-server-prop -h host -p port idle-timeout:integer
```

여기서 *integer*는 연결이 끊기기 전에 클라이언트 연결이 유휴 상태로 유지될 수 있는 최대 시간입니다.

◆ ◆ ◆ 4 장

4 디렉토리 서버 항목

이 장에서는 디렉토리의 데이터 항목을 관리하는 방법에 대해 설명합니다. 또한 참조를 설정하고 속성 값을 암호화하는 방법에 대해 설명합니다.

디렉토리 배포를 계획할 때 디렉토리에 저장할 데이터 유형을 결정해야 합니다. 항목을 만들고 기본 스키마를 수정하기 전에 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 관련 장을 읽어 보십시오.

디렉토리를 수정하려면 적절한 액세스 제어 지침(ACI)이 정의되어 있어야 합니다. 자세한 내용은 6 장를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

항목 관리

항목을 관리하는 가장 좋은 방법은 상황에 따라 다릅니다.

- 관리 작업에서 DSCC를 주로 사용하고, 일부 항목만 검색하거나 수정하려는 경우에는 DSCC를 사용합니다. DSCC에 대한 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스”를 참조하십시오.
- 디렉토리 서버에서 관리 작업을 수행하지 않고 일부 항목만 검색하거나 수정하려는 경우에는 디렉토리 편집기를 사용합니다. 디렉토리 편집기에 대한 자세한 내용은 **Sun Java System Directory Editor 1 2005Q1 Installation and Configuration Guide**를 참조하십시오.
- 많은 항목을 검색하거나 수정하려면 `ldapmodify` 및 `ldapdelete` 명령줄 유틸리티를 사용합니다.

DSCC를 사용한 항목 관리

DSCC을 사용하면 읽기 가능한 모든 항목 속성을 보고, 쓰기 가능한 해당 속성을 편집할 수 있습니다. 또한, 속성을 추가하거나 제거하고 여러 값을 갖는 속성을 설정하며 항목의 객체 클래스를 관리할 수 있습니다. DSCC를 사용하여 항목을 관리하는 방법에 대한 자세한 내용은 DSCC 온라인 도움말을 참조하십시오. 일반적으로 DSCC에 대한 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스”를 참조하십시오.

디렉토리 편집기를 사용한 항목 관리

디렉토리 편집기는 관리자와 최종 사용자가 데이터를 검색, 작성 및 편집하는 데 사용할 수 있는 간편한 디렉토리 편집 도구입니다. 이 데이터는 사용자, 그룹 및 컨테이너 형식입니다.

`ldapmodify` 및 `ldapdelete`를 사용한 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` 명령은 본 설명서의 절차에서 주로 사용하는 명령입니다. 다음 절에서는 절차를 수행하는 데 필요한 기본 작업에 대해 설명합니다. `ldapmodify` 및 `ldapdelete` 명령에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

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

올바른 LDIF 입력 서식 지정에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 “Guidelines for Providing LDIF Input”을 참조하십시오.

다음 절에서는 이러한 기본 작업에 대해 설명합니다.

ldapmodify를 사용한 항목 추가

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

ldapmodify의 -a 옵션을 사용하여 디렉토리에 하나 이상의 항목을 추가할 수 있습니다. 다음 예에서는 사용자가 포함될 구조적 항목을 작성한 후에 사용자 항목을 작성합니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind 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: secret
```

-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 다음 각 항목이 해당 DN과 속성 값으로 나열되고 항목 사이에는 빈 줄이 들어갑니다. ldapmodify 유틸리티는 입력된 항목을 작성하고 오류가 발생하면 이를 보고합니다.

관례상, 항목의 LDIF는 다음과 같은 속성을 열거합니다.

1. 항목의 DN
2. 객체 클래스 목록
3. 이름 지정 속성(하나 이상)이 속성은 DN에 사용되며, 반드시 필수 속성일 필요는 없습니다.
4. 모든 객체 클래스의 필수 속성 목록
5. 허용되는 속성 중에서 추가할 모든 속성

userPassword 속성 값을 입력할 때는 비밀번호를 일반 텍스트로 입력하십시오. 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 아래 예와 같이 항목 추가 명령문과 항목 수정 명령문을 결합할 수 있다는 이점이 있습니다.

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind 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: secret
```

`changetype: add` 키워드는 지정된 DN의 항목이 이후의 모든 속성을 사용하여 작성되어야 함을 나타냅니다. 모든 다른 옵션과 LDIF 규약은 이 절의 앞 부분에서 설명한 내용과 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.

`ldapmodify`를 사용한 항목 수정

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

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 지정할 수도 있습니다.

속성 값 추가

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

다음 예에서는 동일한 add LDIF 구문을 사용하여 여러 값을 갖는 기존 속성과 존재하지 않는 속성에 값을 추가할 수 있는 방법을 보여줍니다.

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: cn
cn: Babs Jensen
-
add: mobile
mobile: (408) 555-7844
```

다음 중 부합되는 조건이 있으면 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 속성 유형을 필요로 하지 않거나 허용하지 않는 경우
- 속성 유형이 여러 값을 갖지 않으며 이미 값이 있는 경우

이진 속성 하위 유형 사용

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

속성 ;binary 하위 유형은 실제 구문에 관계없이 속성 값이 이진 데이터로서 LDAP를 통해 전송됨을 나타냅니다. 이 하위 유형은 userCertificate와 같이 LDAP 문자열 표현이 없는 복잡한 구문에 사용되며, 이외의 용도로 사용해서는 안 됩니다.

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

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

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
version: 1
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: userCertificate;binary
userCertificate;binary:< file:///local/cert-file
```

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

언어 하위 유형이 지정된 속성 추가

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

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 다음과 같습니다.

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`을 추가해야 합니다.

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: modify
add: homePostalAddress;lang-fr
homePostalAddress;lang-fr: 34, rue de la Paix
```

주 – 속성 값에 비ASCII 문자가 있는 경우 UTF-8로 인코딩해야 합니다.

속성 값 수정

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

다음 예에서는 LDIF의 replace 구문을 사용하여 속성 값을 변경하는 방법을 보여줍니다.

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind 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 지정된 값이 모두 추가됩니다.

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

속성 값의 후행 공백

속성 값을 수정할 때 값의 끝에 실수로 후행 공백을 추가하지 마십시오. 후행 공백으로 인해 값이 base-64로 인코딩(예: 34xy57eg)으로 표시될 수도 있습니다.

속성 값이 후행 공백으로 끝나면 후행 공백이 속성 값의 일부로 인코딩됩니다. DSCC 또는 ldapsearch 명령을 사용하여 변경 사항을 확인할 때는 보이는 값이 일반 텍스트일 수도 있으나 base-64로 인코딩된 텍스트로 표시될 수도 있습니다. 이는 사용하는 디렉토리 서버 클라이언트에 따라 다릅니다.

속성 값 삭제

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

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

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind 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

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

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

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind 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

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

ldapdelete 명령줄 유틸리티를 사용하여 디렉토리에서 항목을 삭제합니다. 이 유틸리티는 디렉토리 서버에 바인드하여 해당 DN을 기반으로 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.

다음 예에서는 항목이 하나만 있는 조직 구성 단위를 보여줍니다. 이 항목을 먼저 삭제한 다음 부모 항목을 삭제할 수 있습니다.

```
$ ldapdelete -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
uid=bjensen,ou=People,dc=example,dc=com
ou=People,dc=example,dc=com
```

ldapmodify를 사용한 항목 삭제

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

ldapmodify 유틸리티를 사용하는 경우 changetype: delete 키워드를 함께 사용하여 항목을 삭제할 수도 있습니다. 이전 절에 설명된 것처럼 ldapdelete를 사용할 때와 동일한 제한이 모두 적용됩니다. LDIF 구문을 사용하여 항목을 삭제하면 한 개의 LDIF 파일로 혼합된 작업을 수행할 수 있다는 이점이 있습니다.

다음 예에서는 이전 예와 동일한 삭제 작업을 수행합니다.

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: delete

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

ldapsearch를 사용한 항목 검색

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

ldapsearch 명령줄 유틸리티를 사용하여 디렉토리 항목을 찾고 검색할 수 있습니다. ldapsearch 유틸리티는 Solaris 플랫폼과 함께 제공되는 유틸리티가 아닌, Directory Server Resource Kit의 일부입니다.

ldapsearch, 일반 ldapsearch 옵션, 허용되는 형식 및 예에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

▼ ldapmodify를 사용하여 항목을 이동하거나 이름을 바꾸는 방법

이 절차에서는 DN 수정 작업을 사용합니다. 이 작업을 시작하기 전에 97 페이지 "DN 수정 작업 사용에 대한 지침과 제한 사항" 절에 대해 잘 알고 있어야 합니다.

이 절차의 일부로, DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오. 해당 절차의 다른 부분은 명령줄에서만 수행할 수 있습니다.

주 – 그룹의 uniquemember인 항목의 DN을 수정할 경우 참조 무결성 플러그인이 활성화되어야 합니다. 참조 무결성은 항목이 이동될 때 그룹 구성원이 조정되도록 합니다. 참조 무결성 플러그인을 사용 가능하게 하고 구성하는 방법에 대해서는 230 페이지 "참조 무결성 플러그인을 구성하는 방법"을 참조하십시오.

---

1 부모에서 다른 부모로 항목을 이동하는 경우 부모 항목에 대한 ACI 권한을 확장합니다.

- 항목의 현재 부모 항목을 제거할 경우 ACI에서 allow (export ...) 구문을 사용하여 export 작업을 수행할 수 있어야 합니다.

- 항목의 이후 부모 항목을 제거할 경우 ACI에서 `allow (import ...)` 구문을 사용하여 `import` 작업을 수행할 수 있어야 합니다.

ACI 사용에 대한 자세한 내용은 6 장을 참조하십시오.

2 DN 수정 작업을 전역적으로 사용 가능하게 하거나, 적어도 이동 작업의 영향을 받는 접미어에 대해 사용 가능하게 합니다.

디렉토리 서버의 이전 릴리스와의 호환성을 위해 DN 수정 작업은 기본적으로 사용되지 않습니다.

이전에 DN 수정 작업을 이미 사용한 경우 다음 단계로 이동합니다.

서버에 대해 DN 수정 작업을 전역적으로 사용 가능하게 하려면 다음 명령을 사용합니다.

```
$ dsconf set-server-prop -h host -p port moddn-enabled:on
```

3 `ldapmodify` 명령을 실행합니다.

이 단계에서는 DN 수정 작업을 사용합니다. 다음 중 하나를 수행합니다.

- 항목을 이동합니다.

  예를 들어 다음 명령은 `uid=bjensen` 항목을 `ou=Contractors,dc=example,dc=com` 계약직원의 하위 트리에서 `ou=People,dc=example,dc=com` 직원의 하위 트리로 이동합니다.

  ```
  $ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
  Enter bind password:
  dn: uid=bjensen,ou=Contractors,dc=example,dc=com
  changetype: modrdn
  newrdn: uid=bjensen
  deleteoldrdn: 0
  newsuperior: ou=People,dc=example,dc=com
  ```

- 항목의 이름을 바꿉니다.

  예를 들어 다음 명령은 `uid=bbjensen` 항목의 이름을 `uid=bjensen`으로 바꿉니다.

  ```
  $ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
  Enter bind password:
  dn: uid=bbjensen,ou=People,dc=example,dc=com
  changetype: modrdn
  newrdn: uid=bjensen
  deleteoldrdn: 1
  ```

LDIF 명령문을 작성하는 경우 다음 속성에 주의하십시오.

- `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(`1`은 예, `0`은 아니요).

  이름 지정 속성이 항목 정의에서 의무적으로 사용되어야 할 속성일 경우 항목에서 이 속성을 제거할 수 없습니다.
- `newsuperior` - 항목의 새로운 상위 속성을 지정합니다.

`ldapmodify` 명령과 옵션에 대한 자세한 내용은 `ldapmodify`(1) 설명서 페이지를 참조하십시오.

4 **많은 항목을 포함하는 하위 트리를 이동하거나 이름을 바꿀 때 자원 제한 오류가 발생하면 데이터베이스에서 사용할 수 있는 잠금 수를 늘립니다.**

```
$ dsconf set-server-prop -h host -p port db-lock-count:value
```

이 등록 정보를 수정할 경우 변경 사항을 적용하려면 서버를 다시 시작해야 합니다.

DN 수정 작업 사용에 대한 지침과 제한 사항

이전 절에 설명된 것처럼 DN 수정 작업을 사용할 경우 다음 절의 지침을 수행합니다.

DN 수정 작업 사용에 대한 일반 지침

- 항목을 한 접미어에서 다른 접미어로 이동하거나 루트 접미어의 이름을 바꾸거나 이동하려면 DN 수정 작업을 사용하지 마십시오.
- Directory Server 5.2 2005Q1 버전 이상이 실행 중이어야 합니다. Directory Server 5.2 2005Q1보다 이전 버전의 디렉토리 서버에서는 DN 수정 작업을 사용할 수 없습니다.
- 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 작업은 서버에 있는 모든 접미어에 대해 전역적으로 사용 가능하게 하거나 작업을 수행할 각 접미어에 대해 개별적으로 사용 가능하게 합니다. 기본적으로 DN 수정 작업은 사용되지 않습니다.
- DN 수정 작업을 실행할 각 접미어에 대해 ACI 권한을 확장합니다. `Import` 액세스 권한을 사용하면 항목을 지정된 DN으로 가져올 수 있습니다. `Export` 액세스 권한을 사용하면 항목을 지정된 DN에서 내보낼 수 있습니다.
- DN 수정 작업을 수행하기 전에 이 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 이 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

주의 – 다음 요구 사항을 준수하지 않고 DN 수정 작업을 사용하면 복제가 손상되어 디렉토리 서비스가 종료될 수 있습니다.

- 복제 토폴로지의 모든 서버에서 Directory Server 5.2 버전 이상을 실행하고 있는지 확인합니다. Directory Server 5.2보다 이전 버전의 디렉토리 서버에서는 DN 수정 작업을 사용할 수 없습니다.
- 복제 토폴로지의 모든 서버에서 DN 수정 작업을 사용 가능하게 합니다. DN 수정 작업이 마스터 서버에서는 지원되지만 사용자 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 모든 서버에서** `DN` **수정 작업이 사용 가능하도록 복제 토폴로지를 다시 구성하고, 다음 방법 중 하나로 복제 세션을 시작합니다.**

  - 263 페이지 “복제를 강제로 업데이트하는 방법”의 지침을 수행합니다.
  - 공급자 서버에서 항목을 변경합니다. 이 변경 사항이 사용자 서버에 복제됩니다.
- 토폴로지의 모든 마스터 복제본에서 참조 무결성 플러그인을 사용 가능하게 하여 구성합니다. 이 작업으로 서버는 그룹과 역할에 대한 참조 무결성을 유지할 수 있습니다. 참조 무결성 플러그인을 사용 가능하게 하고 구성하는 방법에 대해서는 230 페이지 “참조 무결성 플러그인을 구성하는 방법”을 참조하십시오.

  DN 수정 작업을 수행한 후에 참조 무결성 플러그인이 이러한 변경 사항을 복제하는 데는 약간의 시간이 걸립니다.

참조 설정

참조를 사용하여 클라이언트 응용 프로그램에서 로컬에 없는 정보를 구하기 위해 연결할 서버를 알려줄 수 있습니다. 참조란 디렉토리 서버에서 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 요청하고, 서버가 기본 참조를 반환하도록 구성된 경우
- 유지관리 또는 보안상의 이유로 전체 접미어가 사용되지 않는 경우

  서버는 해당 접미어에 정의된 참조를 반환합니다. 접미어 수준 참조에 대한 자세한 내용은 63 페이지 "참조 설정 및 접미어 읽기 전용 만들기"를 참조하십시오. 또한 클라이언트에서 쓰기 작업을 요청하면 접미어의 읽기 전용 복제본은 마스터 서버에 대한 참조를 반환합니다.
- 클라이언트에서 명시적으로 스마트 참조에 액세스하는 경우

  스마트 참조는 사용자가 만든 항목입니다. 서버는 스마트 참조에 정의된 참조를 반환합니다.

참조는 항상 다른 서버의 호스트 이름, 포트 번호 및 DN(선택 사항)이 포함된 LDAP URL 형식으로 지정됩니다. 예를 들면 다음과 같습니다. `ldap://east.example.com:389` .

디렉토리 배포에서 참조를 사용하는 방법에 대한 개념 정보는 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**를 참조하십시오.

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

기본 참조 설정

기본 참조는 디렉토리 서버에서 유지관리하는 접미어에 포함되지 않은 DN에서 작업을 제출하는 클라이언트 응용 프로그램에 반환됩니다. 서버는 정의된 모든 참조를 반환하지만 반환 순서는 정의되어 있지 않습니다.

▼ 기본 참조를 설정하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **`dsconf` 명령줄 유틸리티를 사용하여 하나 이상의 기본 참조를 설정합니다.**

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN referral-url:referral-URL
  ```

  예를 들면 다음과 같습니다.

  ```
  $ dsconf set-server-prop -h host1 -p 1389 dc=example,dc=com \
   referral-url:ldap://east.example.com:1389
  ```

스마트 참조 설정

스마트 참조를 사용하여 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 스마트 참조로 정의할 수 있습니다.

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

스마트 참조는 `east.example.com` 서버에서 다른 항목을 가리킵니다.

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

디렉토리에서 스마트 참조를 사용하는 방법은 RFC 4511의 섹션 4.1.10(`http://www.ietf.org/rfc/rfc4511.txt`)절에 지정된 표준을 따릅니다.

▼ 스마트 참조를 만들거나 수정하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 스마트 참조를 만들려면 `referral` 및 `extensibleObject` 객체 클래스를 가진 항목을 만듭니다.

`referral` 객체 클래스는 LDAP URL이 포함될 것으로 예상되는 `ref` 속성을 허용합니다. `extensibleObject` 객체 클래스를 사용하면 이름 지정 속성과 같은 스키마 속성을 사용하여 대상 항목과 일치시킬 수 있습니다.

예를 들어 `uid=bjensen` 항목 대신 스마트 참조를 반환하도록 아래 항목을 정의하려면 다음 명령을 사용합니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind 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 **`%20`**을 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 수행합니다.

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=People,dc=example,dc=com
changetype: replace
replace: telephoneNumber
telephoneNumber: (408) 555-1234
```

2 (옵션) 스마트 참조 항목을 수정하려면 ldapmodify의 -M 옵션을 사용합니다.

```
$ ldapmodify -M -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind 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
```

유효한 속성 구문 검사

디렉토리 서버에서는 다음 작업을 수행할 때마다 속성의 무결성을 검사할 수 있습니다.

- dsadm import 또는 dsconf import를 사용한 데이터 가져오기
- LDAP 또는 DSML을 사용하여 항목을 추가 및 수정하거나 항목의 DN을 수정합니다.

구문 검사를 통해 속성 값이 IETF 권장 사항을 준수하는지 확인합니다. 준수하지 않는 모든 속성은 거부되어 오류 로그에 기록됩니다. 로그 메시지에는 연결 및 작업 아이디가 포함되어 있습니다(해당하는 경우).

기본적으로 서버는 앞에서 설명한 작업의 구문을 자동으로 검사합니다. 구문 검사를 해제하려면 다음 절차를 수행합니다.

주 – 구문 검사는 스키마 검사와 다릅니다. 스키마 검사에 대한 자세한 내용은 277 페이지 “스키마 검사 관리”를 참조하십시오.

▼ 자동 구문 검사를 해제하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

● **자동 구문 검사를 해제하려면 다음 명령을 사용합니다.**

```
$ dsconf set-server-prop -h host -p port check-syntax-enabled:off
```

디렉토리 항목에 대한 수정 추적

기본적으로 서버는 LDAP v3 사양에 지정된 것처럼 새로 생성되거나 수정된 항목에 대한 특수 속성을 유지관리합니다. 접미어의 항목에 저장되는 이러한 특수 속성은 다음과 같습니다.

- `creatorsName` — 항목을 처음 만든 사용자의 DN
- `createTimestamp` — 항목이 만들어진 시간에 대한 타임스탬프(GMT 형식)
- `modifiersName` — 항목을 마지막으로 수정한 사용자의 DN
- `modifyTimestamp` — 항목이 수정된 시간에 대한 타임스탬프(GMT 형식)

▼ 항목 수정 추적 기능을 해제하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

주의 – 항목 수정 추적 기능을 해제하면 호환되지 않는 데이터가 생성됩니다. 많은 응용 프로그램이 이러한 속성에 의존하고 있고, 이 기능을 사용 불가능하게 하면 성능 향상이 최소화되므로 항목 수정 추적 기능을 해제하지 않는 것이 좋습니다.

- **서버에 대한 항목 수정 추적 기능을 해제합니다.**

```
$ dsconf set-server-prop -h host -p port suffix-DN mod-tracking-enabled:off
```

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 디렉토리 서버 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 디렉토리 서버 간에 전송되는 속성 값을 암호화해야 합니다. 일반적인 데이터 보안의 구조적 개요와 특별한 속성 암호화에 대해서는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**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. DSCC를 사용하면 이러한 단계를 쉽게 수행할 수 있습니다. DSCC 사용에 대한 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스”를 참조하십시오.

보안상 속성에 대한 암호화를 설정할 경우 암호화되지 않은 값이 들어 있을 수 있는 데이터베이스 캐시 파일과 데이터베이스 로그 파일을 수동으로 삭제해야 합니다. 이러한 파일을 삭제하는 절차는 105 페이지 “속성 암호화를 구성하는 방법”을 참조하십시오.

새로운 접미어에 데이터를 로드하거나 작성하기 전에 암호화된 모든 속성을 사용 가능하게 해야 합니다.

일부 항목에서 이름 지정 속성으로 사용하는 속성을 암호화하도록 선택한 경우 DN에 표시되는 값은 암호화되지 않고 항목에 저장된 값은 암호화됩니다.

`userPassword` 속성은 암호화하도록 선택하더라도 비밀번호를 일반 텍스트로 저장해야 하기 때문에 보안상의 이점이 없습니다. DIGEST-MD5 SASL 인증의 경우도 마찬가지입니다. 비밀번호 정책에 정의된 암호화 메커니즘이 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}3hakc&jla+=snda%
```

온라인 상태에서 데이터를 가져와서 암호화할 때 키 데이터베이스 인증 비밀번호를 서버에 이미 제공했으므로 메시지는 다시 표시되지 않습니다. 데이터를 오프라인으로 가져오는 경우에는 디렉토리 서버에 비밀번호를 묻는 메시지가 표시됩니다. 가져올 데이터를 암호화하려면 비밀번호를 입력해야 합니다. 데이터 암호를 해독하면(보안상 중요한 작업) 내보내기 작업이 온라인 상태로 수행되는지, 아니면 오프라인 상태로 수행되는지에 관계없이 디렉토리 서버에서 키 데이터베이스 비밀번호를 묻는 메시지가 자동으로 표시됩니다. 따라서 추가 보안 계층이 제공됩니다.

주 – 인증서나 개인 키가 변경되지 않는 한 서버는 동일한 키를 계속 생성합니다. 따라서 동일한 인증서를 사용하는 두 서버 인스턴스 간에 데이터를 전송(내보낸 다음 가져오기)할 수 있습니다.

속성 암호화 및 성능

속성을 암호화하면 데이터 보안이 향상되지만 시스템 성능에 영향을 줄 수 있습니다. 따라서 암호화가 필요한 속성을 주의해서 결정하여 특히 중요하다고 생각되는 속성만 암호화하십시오.

색인 파일을 통해 중요한 데이터를 액세스할 수 있으므로 속성을 완전히 보호하려면 암호화된 속성에 해당하는 색인 키를 암호화해야 합니다. 색인 키 암호화를 수행하지 않은 상태에서 색인화만으로 디렉토리 서버 성능이 이미 저하된 경우 데이터를 데이터베이스로 가져오거나 추가하기 전에 **먼저** 속성 암호화를 구성합니다. 이 절차를 수행하면 암호화된 속성이 처음부터 색인화됩니다.

속성 암호화 사용 고려 사항

속성 암호화 기능을 구현할 때에는 다음을 고려하십시오.

- 속성 암호화 구성을 수정할 경우 데이터를 내보내고 구성을 변경한 다음 새로 구성된 데이터를 가져오는 것이 가장 일반적인 방법입니다.

  그러면 기능상의 손실 없이 모든 구성 변경이 전체적으로 적용됩니다. 이렇게 하지 않으면 일부 기능이 손실되어 데이터 보안이 손상될 수 있습니다.

- 기존 데이터베이스에서 속성 암호화 구성을 수정하면 시스템 성능에 중요한 영향을 미칠 수 있습니다.

  예를 들어 기존 데이터가 있는 데이터베이스 인스턴스가 있다고 가정합니다. 이 데이터베이스에는 `mySensitiveAttribute`라는 속성을 가진 항목이 이미 저장되어 있습니다. 이 속성의 값은 데이터베이스와 색인 파일에 일반 텍스트로 저장됩니다. 나중에 `mySensitiveAttribute` 속성을 암호화할 경우 서버에서 데이터베이스와 색인 파일을 속성 암호화 구성으로 업데이트하려면 데이터베이스 인스턴스의 모든 데이터를 내보낸 다음 데이터베이스로 다시 가져와야 합니다. 따라서 속성이 처음부터 암호화되어 성능 저하가 방지됩니다.

- 데이터를 암호화되지 않은 형식으로 내보낼 경우 비밀번호가 틀리면 내보내기가 거부됩니다.

  보안 조치의 일환으로, 사용자가 암호화되지 않은 형식으로 데이터를 내보내면 서버는 비밀번호를 묻는 메시지를 표시합니다. 사용자가 잘못된 비밀번호를 입력하면 서버는 암호화되지 않은 내보내기 작업을 거부합니다. 비밀번호를 직접 입력하거나 비밀번호가 들어 있는 파일 경로를 입력할 수 있습니다. 이 파일에는 SSL 비밀번호 파일과 동일한 구문이 있습니다. 115 페이지 “인증서 데이터베이스 비밀번호 구성”을 참조하십시오.

- 알고리즘 변경이 지원되지만 잘못 변경할 경우 색인화 기능이 손실될 수 있습니다.

  데이터를 암호화하는 데 사용되는 알고리즘을 변경하려면 데이터를 내보내고 속성 암호화 구성을 수정한 다음 해당 데이터를 가져옵니다. 이 절차를 수행하지 않으면 초기 암호화 알고리즘에 따라 작성된 색인이 더 이상 작동하지 않습니다.

암호화된 속성의 앞에는 사용된 암호화 알고리즘을 나타내는 암호 태그가 있으므로 내부 서버 작업에서 데이터를 가져옵니다. 따라서 디렉토리 서버를 사용하면 알고리즘을 변경하기 전에 데이터를 암호화된 형식으로 내보낼 수 있습니다.

- 서버의 SSL 인증서를 변경하면 암호화된 데이터를 해독할 수 없습니다.

  서버의 SSL 인증서는 속성 암호화 기능에서 암호화 및 암호 해독 작업 수행에 사용되는 자체 키를 생성하는 데 사용됩니다. 따라서 암호화된 데이터를 해독하려면 SSL 인증서가 필요합니다. 데이터를 해독하지 않고 인증서를 변경한 경우 해당 데이터의 암호를 해독할 수 없습니다. 이러한 문제를 방지하려면 데이터를 암호가 해독된 형식으로 내보내고, 인증서를 변경한 다음 데이터를 다시 가져옵니다.

- 데이터를 암호화된 형식으로 전송하려면 즉, 서버 인스턴스에서 다른 서버 인스턴스로 데이터를 내보내거나 가져오려면 두 서버 인스턴스가 동일한 인증서를 사용해야 합니다.

  자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 관리 설명서**의 "Encrypting Attribute Values"를 참조하십시오.

▼ 속성 암호화를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1. **속성 암호화를 구성할 접미어에 항목이 있으면 먼저 이 접미어의 내용을 LDIF 파일로 내보내야 합니다.**

   접미어에 암호화된 속성이 있는 경우 내보낸 LDIF 파일을 사용하여 접미어를 다시 초기화하려면 내보낸 LDIF에서 속성을 암호화된 상태로 유지할 수 있습니다.

2. **속성에 대한 암호화를 사용 가능하게 하려면 다음 명령을 사용합니다.**

   ```
   $ dsconf create-encrypted-attr -h host -p port suffix-DN attr-name cipher-name
   ```

   여기서 *cipher-name*은 다음 중 하나입니다.

   - `des` - DES 블록 암호
   - `des3` - Triple-DES 블록 암호
   - `rc2` - RC2 블록 암호
   - `rc4` - RC4 스트림 암호

   예를 들면 다음과 같습니다.

   ```
   $ dsconf create-encrypted-attr -h host1 -p 1389 dc=example,dc=com uid rc4
   ```

3. **암호화된 속성을 원본 상태로 되돌리려면 다음 명령을 사용합니다.**

   ```
   $ dsconf delete-encrypted-attr -h host -p port suffix-DN attr-name
   ```

4 **하나 이상의 속성을 암호화하도록 구성이 변경되었고 가져오기 작업을 수행하기 전에 해당 속성에 값이 있는 경우 데이터베이스 캐시를 지우고 로그를 제거합니다.**

암호화되지 않은 값은 데이터베이스 캐시 및 데이터베이스 로그에 표시되지 않습니다.

주 – 이러한 파일을 삭제하면 일부 추적 정보가 손실됩니다. 또한 이러한 파일을 삭제한 후에 서버가 복구 모드로 전환되므로 다시 시작하는 데 많은 시간이 소요될 수 있습니다.

데이터베이스 캐시를 지우고 로그를 제거하려면 다음을 수행합니다.

a. **59 페이지 "디렉토리 서버 인스턴스 시작, 중지 및 다시 시작"에 설명된 것처럼 디렉토리 서버를 중지합니다.**

b. **루트 또는 관리자 권한이 있는 사용자로 파일 시스템에서 데이터베이스 캐시 파일을 삭제합니다.**

```
# rm instance-path/db/__db.*
```

c. **파일 시스템에서 데이터베이스 로그 파일을 삭제합니다.**

```
# rm instance-path/db/log.0000000001
```

d. **디렉토리 서버를 다시 시작합니다.**

서버에서 자동으로 새 데이터베이스 캐시 파일을 작성합니다. 캐시가 다시 채워질 때까지 해당 접미어의 작업 성능이 다소 느려질 수도 있습니다.

5 **203 페이지 "접미어 초기화"에 설명된 것처럼 LDIF 파일이 있는 접미어를 초기화합니다.**

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

◆ ◆ ◆ 5 장

5 디렉토리 서버 보안

디렉토리 서버는 네트워크를 통해 안전하면서도 신뢰할 수 있는 통신을 제공하는 여러 메커니즘을 지원합니다. LDAPS는 SSL(Secure Sockets Layer)에 추가로 실행되는 표준 LDAP 프로토콜로, 데이터를 암호화하고 선택적으로 인증 시에 인증서를 사용합니다. 이 장에서 사용되는 SSL이라는 용어는 지원되는 프로토콜 SSL2, SSL3 및 TLS 1.0을 의미합니다.

디렉토리 서버는 Start TLS(Start Transport Layer Security) 확장 작업을 지원하므로 기존에 암호화되지 않은 LDAP 연결에서 TLS를 사용할 수 있습니다.

또한 디렉토리 서버에서는 SASL(Simple Authentication and Security Layer)을 통한 GSSAPI(Generic Security Service API)도 지원합니다. GSSAPI를 이용하면 Solaris 운영 체제(Solaris OS)에서 Kerberos 버전 5 보안 프로토콜을 사용할 수 있습니다. 이 경우 아이디 매핑 메커니즘이 Kerberos 사용자(Principal)를 디렉토리 아이디와 연결합니다.

추가 보안 정보에 대해서는 NSS 웹 사이트(http://www.mozilla.org/projects/security/pki/nss/ (`http://www.mozilla.org/projects/security/pki/nss/`))를 참조하십시오.

이 장에서는 SSL을 통해 보안을 구성하는 절차에 대해 설명합니다. ACI에 대한 자세한 내용은 6 장을 참조하십시오. 사용자 액세스 및 비밀번호에 대한 자세한 내용은 7 장을 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

디렉토리 서버에서 SSL 사용

SSL(Secure Sockets Layer)은 디렉토리 서버와 해당 클라이언트 간에 암호화된 통신과 선택적 인증을 제공합니다. SSL은 LDAP 또는 DSML-over-HTTP를 통해 사용할 수 있으며, 기본적으로 LDAP를 통해 활성화되지만 DSML-over-HTTP를 사용하면 SSL을 보다 쉽게 활성화할 수 있습니다. 또한, 복제 시 서버 간의 보안 통신에 SSL을 사용하도록 구성할 수 있습니다.

SSL을 단순 인증(바인드 DN 및 비밀번호)과 함께 사용하면 서버 간에 전송되는 모든 데이터가 암호화되므로 기밀성과 데이터 무결성이 보장됩니다. 선택 사항으로, 클라이언트는 인증서를 사용하여 디렉토리 서버에, 또는 SASL(Simple Authentication and Security Layer)을 통한 타사 보안 메커니즘에 인증할 수 있습니다. 인증서 기반 인증에서는 클라이언트나 서버를 사칭하지 못하도록 공개 키 암호화를 사용합니다.

디렉토리 서버는 별도의 포트에서 SSL 통신과 비SSL 통신을 동시에 지원합니다. 보안상의 이유로 모든 통신을 LDAP 보안 포트로 제한할 수도 있으며 클라이언트 인증을 구성할 수도 있습니다. 클라이언트 인증은 필수 또는 선택 사항으로 설정할 수 있으며 이 설정으로 적용할 보안 수준이 결정됩니다.

SSL을 사용하면 일반 LDAP 연결에 보안을 제공하는 Start TLS 확장 작업도 지원됩니다. 클라이언트는 표준 LDAP 포트에 바인드한 후에 TLS(Transport Layer Security) 프로토콜을 사용하여 연결을 안전하게 설정할 수 있습니다. Start TLS 작업은 클라이언트의 유연성을 높이며 포트 할당을 용이하게 합니다.

SSL에서 제공하는 암호화 메커니즘은 속성 암호화에도 사용됩니다. SSL을 사용할 경우 접미어에 속성 암호화를 구성하여 디렉토리에 저장된 데이터를 보호할 수 있습니다. 자세한 내용은 102 페이지 “속성 값 암호화”를 참조하십시오.

액세스 제어 지침(ACI)을 통해 디렉토리 내용에 대한 액세스 제어를 설정하여 보안을 강화할 수 있습니다. ACI에는 특정 인증 방법이 필요하며, 데이터가 보안 채널을 통해서만 전송될 수 있도록 합니다. SSL과 인증서의 사용이 상호 보충되도록 ACI를 설정합니다. 자세한 내용은 6 장을 참조하십시오.

SSL은 기본적으로 LDAP를 통해 활성화되지만 DSML-over-HTTP를 사용하면 SSL을 보다 쉽게 활성화할 수 있습니다. 또한, 다음 절에 설명된 것처럼 일부 SSL 구성을 수정할 수 있습니다.

인증서 관리

이 절에서는 디렉토리 서버에서 SSL 인증서를 관리하는 방법에 대해 설명합니다.

디렉토리 서버에서 SSL을 실행하려면 자체 서명된 인증서 또는 PKI(Public Key Infrastructure) 솔루션을 사용해야 합니다.

PKI 솔루션에는 외부 인증 기관(CA)이 필요합니다. 즉, PKI 솔루션에는 공개 키와 개인 키가 모두 있는 CA 서명된 서버 인증서가 필요합니다. 이 인증서는 하나의 디렉토리

서버에만 적용됩니다. 또한 공개 키가 포함된 신뢰할 수 있는 CA 인증서도 필요합니다. 신뢰할 수 있는 CA 인증서를 사용하면 CA의 모든 서버 인증서가 신뢰됩니다. 이 인증서를 CA 루트 키 또는 루트 인증서라고도 합니다.

주 – 인증서를 테스트용으로 사용하는 경우에는 자체 서명된 인증서를 사용할 수 있습니다. 그러나, 작업 환경에서 자체 서명된 인증서를 사용하면 보안상 안전하지 않을 수 있습니다. 작업 환경에서는 신뢰할 수 있는 인증 기관(CA) 인증서를 사용합니다.

이 절의 절차에서는 `dsadm` 및 `dsconf` 명령을 사용합니다. 이 명령에 대한 자세한 내용은 `dsadm`(1M) 및 `dsconf`(1M) 설명서 페이지를 참조하십시오.

이 절에서는 디렉토리 서버에서 인증서를 구성하는 방법과 관련하여 다음 정보에 대해 설명합니다.

▼ 자체 서명된 기본 인증서를 보는 방법

디렉토리 서버 인스턴스를 처음 만들 때 자체 서명된 기본 인증서가 포함됩니다. **자체 서명된 인증서**는 공개 키와 개인 키 쌍으로, 여기서 공개 키는 개인 키에 의해 서명됩니다. 자체 서명된 인증서는 3개월 동안 유효합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **자체 서명된 기본 인증서를 보려면 다음 명령을 사용합니다.**

```
$ dsadm show-cert instance-path defaultCert
```

▼ 자체 서명된 인증서를 관리하는 방법

디렉토리 서버 인스턴스를 만들 때 자체 서명된 기본 인증서가 자동으로 제공됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **기본값 이외의 설정으로 자체 서명된 인증서를 만들려면 다음 명령을 사용합니다.**

```
$ dsadm add-selfsign-cert instance-path cert-alias
```

여기서 *cert-alias*는 인증서를 식별하기 위해 입력하는 이름입니다.

이 명령에 대한 모든 옵션을 보려면 dsadm(1M) 설명서 페이지 또는 명령줄 도움말을 참조하십시오..

```
$ dsadm add-selfsign-cert --help
```

2 **자체 서명된 인증서가 만료되면 서버 인스턴스를 중지하고 인증서를 갱신합니다.**

```
$ dsadm stop instance-path
$ dsadm renew-selfsign-cert instance-path cert-alias
```

3 **서버 인스턴스를 다시 시작합니다.**

```
$ dsadm start instance-path
```

▼ CA 서명된 서버 인증서를 요청하는 방법

이 절차에서는 디렉토리 서버에 사용할 CA 서명된 서버 인증서를 요청하고 설치하는 방법에 대해 설명합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 **CA 서명된 서버 인증서 요청을 생성합니다.**

```
$ dsadm request-cert [-W cert-pwd-file] {-S DN | --name name [--org org] \
  [--org-unit org-unit] [--city city] [--state state] [--country country]} \
  [-o output-file] [-F format] instance-path
```

예를 들어 Example 회사에 대한 CA 서명된 서버 인증서를 요청하려면 다음 명령을 사용합니다.

```
$ dsadm request-cert --name host1 --org Example --org-unit Marketing \
 -o my_cert_request_file /local/ds
```

서버를 완벽하게 식별하기 위해 인증 기관(CA)에서는 이 예에 표시된 모든 속성이 필요할 수 있습니다. 각 속성에 대한 설명은 dsadm(1M) 설명서 페이지를 참조하십시오.

dsadm request-cert를 사용하여 인증서를 요청하는 경우 완성된 인증서 요청은 ASCII를 출력 형식으로 지정하지 않는 한 이진 인증서 요청이 됩니다. ASCII를 지정하면 완성된 인증서 요청은 PEM 형식의 PKCS #10 인증서 요청이 됩니다. PEM은 RFC 1421부터

1424(http://www.ietf.org/rfc/rfc1421.txt)까지 지정된 Privacy Enhanced Mail 형식으로, base64 인코딩된 인증서 요청을 US-ASCII 문자로 나타내는 데 사용됩니다. 요청 내용은 다음 예와 같이 표시됩니다.

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBrjCCARcCAQAwbjELMAkGA1UBhMCVXMxEzARBgNVBAgTCkNBElGT1JOSUExLD
AqBgVBAoTI25ldHNjYXBlIGNvb11bmljYXRpb25zIGNvcnBvcmF0aWuMRwwGgYDV
QQDExNtZWxsb24umV0c2NhcGUuY29tMIGfMA0GCSqGSIb3DQEBAUAA4GNADCBiQK
BgCwAbskGh6SKYOgHy+UCSLnm3ok3X3u83Us7u0EfgSLR0f+K41eNqqWRftGR83e
mqPLDOf0ZLTLjVGJaHJn4l1gG+JDf/n/zMyahxtV7+T8GOFFigFfuxJaxMjr2j7I
vELlxQ4IfZgwqCm4qQecv3G+N9YdbjveMVXW0v4XwIDAQABAAwDQYJKoZIhvcNAQ
EEBQADgYEAZyZAm8UmP9PQYwNy4Pmypk79t2nvzKbwKVb97G+MT/gw1pLRsuBoKi
nMfLgKp1Q38K5Py2VGW1E47/rhm3yVQrIiwV+Z8Lcc=
-----END NEW CERTIFICATE REQUEST-----
```

2 해당 절차에 따라 인증서 요청을 인증 기관에 전송합니다.

인증 기관 인증서를 얻기 위한 프로세스는 사용하는 인증 기관에 따라 다릅니다. 일부 상용 CA에서는 인증서를 자동으로 다운로드할 수 있는 웹 사이트를 제공합니다. 다른 CA에서는 요청한 인증서를 전자 메일로 보냅니다.

요청을 전송한 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.

3 인증 기관에서 받은 인증서를 저장합니다.

인증서를 안전한 위치에 백업합니다. 인증서가 손실될 경우 이 백업 파일을 사용하여 인증서를 다시 설치할 수 있습니다. 인증서는 텍스트 파일로 저장할 수 있습니다. PEM 형식의 PKCS #11 인증서는 다음 예와 같이 표시됩니다.

```
-----BEGIN CERTIFICATE-----
MIICjCCAZugAwIBAgICCEEwDQYJKoZIhKqvcNAQFBQAwfDELMAkGA1UEBhMCVVMx
IzAhBgNVBAoGlBhbG9a2FWaWxsZGwSBXaWRnZXRzLCBJbmMuMR0wGwYDVQQLExRX
aWRnZXQgTW3FrZXJzICdSJyBVczEpMCcGAx1UEAxgVGVzdCBUXN0IFRlc3QgVGVz
dCBUZXN0IFlc3QgQ0EswHhcNOTgwMzEyMDIzMzUWhcNOTgwMzI2MDIzMpzU3WjBP
MQswCYDDVQQGEwJVUzEoMCYGA1UEChMfTmV0c2NhcGUgRGlyZN0b3J5VIFB1Ymxp
Y2F0aW9uczEWMB4QGA1UEAxMNZHVgh49dq2tLNvbjTBaMA0GCSqGSIb3DQEBAQUA
A0kAMEYkCQCksMR/aLGdfp4m0OiGgijG5KgOsyRNvwGYW7kfW+8mmijDtZaRjYNj
jcgpF3VnlbxbclX9LVjjNLC5737XZdAgEDozYwpNDARBglghkgBhvhCEAQEEBAMC
APAwHkwYDVR0jBBgwFAU67URjwCaGqZHUpSpdLxlzwJKiMwDQYJKoZIhQvcNAQEF
BQADgYEAJ+BfVem3vBOPBveNdLGfjlb9hucgmaMcQa9FA/db8qimKT/ue9UGOJqL
bwbMKBBopsDn56p2yV3PLIsBgrcuSoBCuFFnxBnqSiTS7YiYgCWqWaUA0ExJFmD6
6hBLseqkSWulk+hXHN7L/NrViO+7zNtKcaZLlFPf7d7j2MgX4Bo=
-----END CERTIFICATE-----
```

▼ CA 서명된 서버 인증서 및 신뢰할 수 있는 CA 인증서를 추가하는 방법

이 절차에서는 디렉토리 서버에 사용할 CA 서명된 서버 인증서 및 신뢰할 수 있는 CA 인증서를 설치하는 방법에 대해 설명합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 CA 서명된 서버 인증서를 추가합니다.

```
$ dsadm add-cert --ca instance-path cert-alias cert-file
```

여기서 *cert-alias*는 인증서를 식별하기 위해 입력하는 이름이고 *cert-file*은 PEM 형식의 PKCS #11 인증서가 저장된 텍스트 파일입니다.

예를 들어 CA 서명된 서버 인증서를 설치할 경우 다음과 같은 명령을 사용할 수 있습니다.

```
$ dsadm add-cert --ca /local/ds server-cert /local/safeplace/serv-cert-file
```

이 인증서는 바로 설치되지만 신뢰되지는 않습니다. CA 서명된 서버 인증서를 신뢰하려면 인증 기관 인증서를 설치해야 합니다.

2 신뢰할 수 있는 인증 기관 인증서를 추가합니다.

```
$ dsadm add-cert --ca -C instance-path cert-alias cert-file
```

`-C` 옵션은 해당 인증서가 신뢰할 수 있는 인증 기관 인증서임을 나타냅니다.

예를 들어 신뢰할 수 있는 인증 기관 인증서를 설치할 경우 다음 명령을 사용할 수 있습니다.

```
$ dsadm add-cert --ca -C /local/ds CA-cert /local/safeplace/ca-cert-file
```

3 (옵션) 설치된 인증서를 확인합니다.

- 모든 서버 인증서를 나열하고 유효 날짜 및 별칭을 표시하려면 다음을 입력합니다.

  ```
  $ dsadm list-certs instance-path
  ```

  예를 들면 다음과 같습니다.

  ```
  $ dsadm list-certs /local/ds1
  Enter the certificate database password:
  Alias       Valid from Expires on Self-   Issued by          Issued to
                                    signed?
  ----------- ---------- ---------- ------- -----------------  -----------------
  serverCert  2000/11/10 2011/02/10 n       CN=CA-Signed Cert, CN=Test Cert,
              18:13      18:13              OU=CA,O=com        dc=example,dc=com
  ```

```
defaultCert 2006/05/18 2006/08/18 y       CN=host1,CN=DS,    Same as issuer
            16:28      16:28              dc=example,dc=com
2 certificates found
```

기본적으로 디렉토리 프록시 서버 인스턴스에는 이름이 `defaultCert`인 기본 서버 인증서가 포함되어 있습니다. `Same as issuer` 텍스트는 자체 서명된 서버 인증서가 기본 인증서임을 나타냅니다.

- 신뢰할 수 있는 CA 인증서를 나열하려면 다음을 입력합니다.

```
$ dsadm list-certs -C instance-path
```

예를 들면 다음과 같습니다.

```
$ dsadm list-certs -C /local/ds1
Enter the certificate database password:
Alias   Valid from Expires on Self-   Issued by           Issued to
                              signed?
------- ---------- ---------- ------- -----------------   --------------
CA-cert 2000/11/10 2011/02/10 y       CN=Trusted CA Cert, Same as issuer
        18:12      18:12              OU=CA,O=com
1 certificate found
```

- 인증서 만료 날짜를 비롯하여 인증서에 대한 자세한 내용을 보려면 다음을 입력합니다.

```
$ dsadm show-cert instance-path cert-alias
```

예를 들어 서버 인증서를 보려면 다음을 입력합니다.

```
$ dsadm show-cert /local/ds1 "Server-Cert"
Enter the certificate database password:
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 2 (0x2)
        Signature Algorithm: PKCS #1 MD5 With RSA Encryption
        Issuer:
            "CN=Server-Cert,O=Sun,C=US"
        Validity:
            Not Before: Fri Nov 10 18:12:20 2000
            Not After : Thu Feb 10 18:12:20 2011
        Subject:
            "CN=CA Server Cert,OU=ICNC,O=Sun,C=FR"
        Subject Public Key Info:
            Public Key Algorithm: PKCS #1 RSA Encryption
            RSA Public Key:
                Modulus:
                    bd:76:fc:29:ca:06:45:df:cd:1b:f1:ce:bb:cc:3a:f7:
```

```
                77:63:5a:82:69:56:5f:3d:3a:1c:02:98:72:44:36:e4:
                68:8c:22:2b:f0:a2:cb:15:7a:c4:c6:44:0d:97:2d:13:
                b7:e3:bf:4e:be:b5:6a:df:ce:c4:c3:a4:8a:1d:fa:cf:
                99:dc:4a:17:61:e0:37:2b:7f:90:cb:31:02:97:e4:30:
                93:5d:91:f7:ef:b0:5a:c7:d4:de:d8:0e:b8:06:06:23:
                ed:5f:33:f3:f8:7e:09:c5:de:a5:32:2a:1b:6a:75:c5:
                0b:e3:a5:f2:7a:df:3e:3d:93:bf:ca:1f:d9:8d:24:ed
            Exponent: 65537 (0x10001)
    Signature Algorithm: PKCS #1 MD5 With RSA Encryption
    Signature:
        85:92:42:1e:e3:04:4d:e5:a8:79:12:7d:72:c0:bf:45:
        ea:c8:f8:af:f5:95:f0:f5:83:23:15:0b:02:73:82:24:
        3d:de:1e:95:04:fb:b5:08:17:04:1c:9d:9c:9b:bd:c7:
        e6:57:6c:64:38:8b:df:a2:67:f0:39:f9:70:e9:07:1f:
        33:48:ea:2c:18:1d:f0:30:d8:ca:e1:29:ec:be:a3:43:
        6f:df:03:d5:43:94:8f:ec:ea:9a:02:82:99:5a:54:c9:
        e4:1f:8c:ae:e2:e8:3d:50:20:46:e2:c8:44:a6:32:4e:
        51:48:15:d6:44:8c:e6:d2:0d:5f:77:9b:62:80:1e:30
    Fingerprint (MD5):
        D9:FB:74:9F:C3:EC:5A:89:8F:2C:37:47:2F:1B:D8:8F
    Fingerprint (SHA1):
        2E:CA:B8:BE:B6:A0:8C:84:0D:62:57:85:C6:73:14:DE:67:4E:09:56

    Certificate Trust Flags:
        SSL Flags:
            Valid CA
            Trusted CA
            User
            Trusted Client CA
        Email Flags:
            User
        Object Signing Flags:
            User
```

▼ 만료된 CA 서명 서버 인증서를 갱신하는 방법

CA 서명된 서버 인증서(공개 키 및 개인 키)가 만료되면 이 절차를 사용하여 인증서를 갱신합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 해당 인증 기관에서 업데이트된 CA 서명된 서버 인증서를 얻습니다.

2 업데이트된 인증서를 받는 경우 서버 인스턴스를 중지하고 인증서를 설치합니다.

```
$ dsadm stop instance-path
$ dsadm renew-cert instance-path cert-alias cert-file
```

3 **서버 인스턴스를 다시 시작합니다.**

```
$ dsadm start instance-path
```

▼ CA 서명된 서버 인증서를 내보내고 가져오는 방법

나중에 인증서를 가져올 수 있도록 인증서의 공개 키와 개인 키를 내보내야 할 수 있습니다. 예를 들어 인증서를 다른 서버에 사용해야 할 수 있습니다.

이 절차에 있는 명령은 "cn=*,o=example"과 같은 와일드카드가 포함된 인증서에 사용할 수 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **인증서를 내보냅니다.**

```
$ dsadm export-cert [-o output-file] instance-path cert-alias
```

예를 들면 다음과 같습니다.

```
$ dsadm export-cert -o /tmp/first-certificate /local/ds1 "First Certificate"
$ dsadm export-cert -o /tmp/first-ca-server-certificate /local/ds1/ defaultCert
Choose the PKCS#12 file password:
Confirm the PKCS#12 file password:
$ ls /tmp
first-ca-server-certificate
```

2 **인증서를 가져옵니다.**

```
$ dsadm import-cert instance-path cert-file
```

예를 들어 인증서를 host1의 서버 인스턴스로 가져오려면 다음을 입력합니다.

```
$ dsadm import-cert -h host1 /local/ds2 /tmp/first-ca-server-certificate
Enter the PKCS#12 file password:
```

3 **(옵션) 인증서를 서버로 가져온 후 서버에서 가져온 인증서를 사용하도록 구성합니다.**

```
$ dsconf set-server-prop -e -h host -p port -w - ssl-rsa-cert-name:server-cert
```

인증서 데이터베이스 비밀번호 구성

기본적으로 디렉토리 서버에서는 저장된 비밀번호를 통해 내부적으로 SSL 인증서 데이터베이스 비밀번호를 관리합니다. 인증서를 관리할 때 사용자가 인증서 비밀번호를 입력하거나 비밀번호 파일을 지정할 필요는 없습니다. 이 옵션은 비밀번호가 숨겨지기만 하고 암호화되지는 않기 때문에 보안상 안전하지 않을 수 있습니다.

그러나, 인증서 사용을 좀 더 엄격하게 제어하려면 명령줄에서 비밀번호를 입력하라는 메시지가 표시되도록 서버를 구성할 수 있습니다. 이 경우 사용자는 `autostart`, `backup`, `disable-service`, `enable-service`, `info`, `reindex`, `restore` 및 `stop`을 제외한 모든 `dsadm` 하위 명령 실행 시에 인증서 데이터베이스 비밀번호를 입력해야 합니다. 인증서 데이터베이스는 *instance-path*`/alias` 디렉토리에 있습니다.

▼ 인증서 비밀번호를 입력하라는 메시지가 표시되도록 서버를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 서버를 중지합니다.

```
$ dsadm stop instance-path
```

2 비밀번호 프롬프트 플래그를 `on`으로 설정합니다.

```
$ dsadm set-flags instance-path cert-pwd-prompt=on
```

새 인증서 비밀번호를 선택하라는 메시지가 표시됩니다.

3 서버를 시작합니다.

```
$ dsadm start instance-path
```

디렉토리 서버의 인증서 데이터베이스 백업 및 복원

디렉토리 서버 인스턴스를 백업하는 경우 디렉토리 서버 구성 및 인증서를 백업합니다. 백업된 인증서는 *archive-path*`/alias` 디렉토리에 저장됩니다.

디렉토리 서버 백업 및 복원 방법에 대한 자세한 내용은 210 페이지 "재해 복구를 위한 백업을 만드는 방법"을 참조하십시오.

SSL 통신 구성

이 절은 SSL 활성화 및 비활성화와 관련된 절차로 구성되어 있습니다.

비보안 통신 비활성화

서버 인스턴스를 만들면 기본적으로 LDAP 일반 포트와 보안 LDAP 포트(LDAPS)가 모두 만들어집니다. 그러나, 서버가 SSL을 통해서만 통신하도록 비SSL 통신을 비활성화해야 할 수 있습니다.

자체 서명된 기본 인증서에 SSL 연결을 사용할 수 있습니다. 필요한 경우 사용자 고유의 인증서를 설치할 수 있습니다. 서버 시작 후 인증서를 관리하고 SSL을 비활성화하는 방법에 대한 지침은 5 장을 참조하십시오. 인증서, 인증서 데이터베이스 및 CA 서명된 서버 인증서를 얻는 방법에 대한 개요는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

▼ LDAP 일반 포트를 비활성화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 LDAP 일반 포트를 비활성화합니다.

비보안 포트를 비활성화하려면 LDAP 보안 포트에 바인드해야 합니다. 다음 예는 호스트 서버 `host1`에서 기본 LDAP 보안 포트 1636에 바인드하는 명령을 나타냅니다.

```
$ dsconf set-server-prop -h host1 -P 1636 ldap-port:disabled
```

2 변경 사항을 적용하려면 서버를 다시 시작합니다.

```
$ dsadm restart /local/ds
```

이제 비보안 포트 1389에서 더 이상 바인드할 수 없습니다.

암호화 암호 선택

암호는 데이터 암호화 및 암호 해독에 사용되는 알고리즘입니다. 일반적으로 암호화 중에 암호가 사용하는 비트 수가 많을수록 암호화는 더욱 **강력**해지거나 안전해집니다. SSL 암호는 사용된 메시지 인증 유형으로도 식별할 수 있습니다. 메시지 인증은 데이터 무결성을 보장하는 `checksum`을 계산하는 별개의 알고리즘입니다.

클라이언트가 서버와의 SSL 연결을 시작하려면 클라이언트 및 서버가 정보 암호화에 사용할 암호에 동의해야 합니다. 양방향 암호화 프로세스의 경우 클라이언트와 서버 모두 동일한 암호를 사용해야 합니다. 사용되는 암호는 서버에서 유지되는 암호 목록의 현재 순서에 따라 달라집니다. 서버는 클라이언트가 표시하는 항목 중 암호 목록 내의 암호와 일치하는 첫 번째 암호를 선택합니다. 디렉토리 서버의 기본 암호 값은 `all`이며, 이는 기본 SSL 라이브러리에서 지원하는 알려진 모든 보안 암호를 의미합니다. 그러나, 특정 암호만 허용하도록 이 값을 수정할 수 있습니다.

디렉토리 서버에 사용할 수 있는 암호에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

▼ 암호화 암호를 선택하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **SSL이 서버에 대해 활성화되어 있는지 확인합니다.**

116 페이지 “SSL 통신 구성”을 참조하십시오.

2 **사용 가능한 SSL 암호를 봅니다.**

```
$ dsconf get-server-prop -h host -p port ssl-supported-ciphers
ssl-supported-ciphers  :  TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_DHE_RSA_WITH_AES_256_CBC_SHA
ssl-supported-ciphers  :  TLS_DHE_DSS_WITH_AES_256_CBC_SHA
...
```

3 **(옵션) 암호화되지 않은 데이터의 복사본을 유지하려면 SSL 암호를 설정하기 전에 데이터를 내보냅니다.**

200 페이지 “LDIF로 내보내기”를 참조하십시오.

4 **SSL 암호를 설정합니다.**

```
$ dsconf set-server-prop -h host -p port ssl-cipher-family:cipher
```

예를 들어 암호 패밀리를 SSL_RSA_WITH_RC4_128_MD5 및 SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA로 설정하려면 다음을 입력합니다.

```
$ dsconf set-server-prop -h host1 -P 1636 ssl-cipher-family:SSL_RSA_WITH_RC4_128_MD5 \
 ssl-cipher-family:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
Enter "cn=Directory Manager" password:
Before setting SSL configuration, export Directory Server data.
Do you want to continue [y/n] ? y
Directory Server must be restarted for changes to take effect.
```

5 **(옵션) 기존 목록에 SSL 암호를 추가합니다.**

암호 목록이 이미 지정되어 있고 암호를 추가하려면 이 명령을 사용합니다.

```
$ dsconf set-server-prop -h host -p port ssl-cipher-family+:cipher
```

예를 들어 SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA 암호를 추가하려면 다음을 입력합니다.

```
$ dsconf set-server-prop -h host1 -P 1636 \
 ssl-cipher-family+:SSL_DHE_RSA_WITH_3DES_EDE_CBC_SHA
```

6 **변경 사항을 적용하려면 서버를 다시 시작합니다.**

```
$ dsadm restart /local/ds
```

자격 증명 수준 및 인증 방법 구성

클라이언트에 적용되는 보안 모델은 자격 증명 수준과 인증 방법의 조합으로 정의됩니다.

디렉토리 서버는 다음 자격 증명 수준을 지원합니다.

- **익명.**디렉토리의 특정 부분에 익명 액세스를 허용한다는 것은 디렉토리에 액세스하는 사람이면 누구나 읽기 액세스 권한이 있음을 의미합니다.

  익명 자격 증명 수준을 사용하는 경우 모든 LDAP 이름 지정 항목과 속성에 읽기 액세스를 허용해야 합니다.

주의 – 쓰기 액세스 권한이 있는 누구나 다른 사용자의 비밀번호 또는 사용자 고유의 아이디를 포함한 정보를 DIT에서 변경할 수 있으므로 디렉토리에 익명 쓰기 액세스를 허용하지 마십시오.

- **프록시.**클라이언트는 프록시 계정을 사용하여 디렉토리에 인증하거나 바인드합니다.

  이 프록시 계정은 디렉토리에 바인드하도록 허용된 모든 항목일 수 있습니다. 프록시 계정에서 디렉토리에 이름 지정 서비스 기능을 수행하려면 충분한 액세스 권한이 있어야 합니다. 프록시 자격 증명 수준을 사용하여 각 클라이언트에서 `proxyDN` 및 `proxyPassword`를 구성해야 합니다. 암호화된 `proxyPassword`는 클라이언트에서 로컬에 저장됩니다.
- **프록시 익명.**둘 이상의 자격 증명 수준이 정의된 여러 값을 갖는 항목입니다.

  클라이언트에 할당된 프록시 익명 수준은 먼저 프록시 아이디로 인증을 시도합니다. 클라이언트가 어떠한 이유로든(사용자 잠금, 비밀번호 만료 등) 프록시 사용자로 인증할 수 없는 경우 클라이언트는 익명 액세스를 사용합니다. 이렇게 되면 디렉토리가 구성된 방식에 따라 서비스 수준이 달라질 수 있습니다.

클라이언트 인증은 서버에서 클라이언트의 아이디를 확인하는 메커니즘입니다.

다음 방법 중 하나로 클라이언트 인증을 수행할 수 있습니다.

- DN 및 비밀번호 입력
- 클라이언트가 제공하는 인증서 사용

  인증서 기반의 인증 시에는 SSL 프로토콜을 통해 얻은 클라이언트 인증서를 사용하여 식별할 사용자 항목을 찾습니다. 인증서 기반의 인증 시 클라이언트는 외부 메커니즘을 지정하여 SASL 바인드 요청을 보냅니다. 바인드 요청은 이미 구성된 SSL 인증 메커니즘을 사용합니다.

- SASL 기반의 메커니즘 사용
  - 모든 운영 체제 - DIGEST-MD5를 통한 SASL
  - Solaris 운영 체제 - Kerberos V5를 사용하여 클라이언트 인증을 허용하는 GSSAPI 메커니즘을 통한 SASL

  두 SASL 메커니즘 중 하나를 사용하는 경우 서버도 아이디 매핑을 수행하도록 구성해야 합니다. SASL 자격 증명을 **사용자***(Principal)*라고 합니다. 각 메커니즘에는 사용자(Principal)의 내용에서 바인드 DN을 결정하는 특정 매핑이 있어야 합니다. 사용자(Principal)가 단일 사용자 항목에 매핑되고 SASL 메커니즘에서 해당 사용자의 아이디를 확인하는 경우 사용자의 DN은 연결을 위한 바인드 DN입니다.
- SSL 클라이언트 인증 모드 사용

  모든 클라이언트가 SSL 계층에서 승인되도록 하려면 SSL 클라이언트 인증을 사용합니다. 클라이언트 응용 프로그램에서 SSL 인증서를 서버로 보내서 인증합니다. `SSL-client-auth-mode` 플래그를 사용하여 서버에서 SSL 클라이언트 인증을 허용하거나 요구할지, 아니면 허용하지 않을지 여부를 지정합니다. 기본적으로 클라이언트 인증이 허용됩니다.

이 절에서는 이 두 SASL 메커니즘을 디렉토리 서버에 구성하는 방법과 관련하여 다음 정보에 대해 설명합니다.

- 120 페이지 “디렉토리 서버의 SASL 암호화 수준 설정”
- 122 페이지 “DIGEST-MD5를 통한 SASL 인증”
- 124 페이지 “GSSAPI를 통한 SASL 인증(Solaris OS에만 해당)”

보안 구성에 대한 자세한 내용은 127 페이지 “LDAP 클라이언트에서 보안을 사용하도록 구성”을 참조하십시오.

디렉토리 서버의 SASL 암호화 수준 설정

SASL 메커니즘을 구성하기 전에 암호화가 필요한지 여부를 지정해야 합니다. SASL 암호화 필요 여부는 최대 및 최소 SSF(Strength Security Factor)로 설정됩니다.

`dsSaslMinSSF`(5dsat) 및 `dsSaslMaxSSF`(5dsat) 속성은 암호화 키 길이를 나타내며 `cn=SASL, cn=security, cn=config`에 저장됩니다.

이 서버는 암호화 없음을 비롯하여 모든 수준의 암호화를 허용합니다. 이는 디렉토리 서버가 256보다 큰 `dsSaslMinSSF` 및 `dsSaslMaxSSF` 값을 허용한다는 의미입니다. 그러나, 현재 SASL 메커니즘은 128보다 큰 SSF를 지원하지 않습니다. 디렉토리 서버에서는 SSF 값을 SASL 메커니즘에서 지원하는 최대 값(128) 이하로 조정합니다. 따라서, 실제 최대 SSF 값은 사용 가능한 기본 메커니즘에 따라 구성된 최대 값보다 작을 수 있습니다.

SASL 보안 요소 인증은 서버 및 클라이언트 응용 프로그램에서 요청되는 최소 및 최대 요소와 기본 보안 구성 요소에서 제공되는 사용 가능한 암호화 메커니즘과 같은 두 가지

주요 항목에 따라 달라집니다. 즉, 서버 및 클라이언트는 둘 모두에 설정된 최대 요소보다 작거나 같은, 하지만 최소 요소보다는 크거나 같은 사용 가능한 최대 보안 요소를 사용하려고 시도합니다.

디렉토리 서버의 기본 최소 SASL 보안 요소인 dsSaslMinSSF는 0으로, 보호되지 않음을 의미합니다. 실제 최소 요소 값은 사용자가 디렉토리 서버의 최소 요소 값을 변경하지 않는 한 클라이언트 설정에 따라 달라집니다. 실제로 최소 요소 값은 서버와 클라이언트가 실제로 사용할 최저 수준으로 설정해야 합니다. 서버와 클라이언트 간에 최소 요소 값이 일치하도록 메커니즘이 조정되지 않으면 연결이 구성되지 않습니다.

▼ SASL 암호화를 허용하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **SASL 암호화를 허용하려면 dsSaslMinSSF 값을 요구되는 최소 암호화로 설정합니다.**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=SASL, cn=security, cn=config
changetype: modify
replace: dsSaslMinSSF
dsSaslMinSSF: 128
^D
```

▼ SASL 암호화를 허용하지 않는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **SASL 암호화를 허용하지 않으려면 dsSaslMinSSF 및 dsSaslMaxSSF 값을 모두 0으로 설정합니다.**

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=SASL, cn=security, cn=config
changetype: modify
replace: dsSaslMinSSF
dsSaslMinSSF: 0

replace: dsSaslMaxSSF
dsSaslMaxSSF: 0
^D
```

DIGEST-MD5를 통한 SASL 인증

DIGEST-MD5 메커니즘은 클라이언트가 전송한 해시된 값을 사용자 비밀번호의 해시와 비교하여 클라이언트를 인증합니다. 그러나 이 메커니즘은 사용자 비밀번호를 읽어야 하기 때문에 DIGEST-MD5를 통해 인증을 받으려는 사용자는 디렉토리에 {CLEAR} 비밀번호가 있어야 합니다. {CLEAR} 비밀번호를 디렉토리에 저장하는 경우 6 장에 설명된 것처럼 비밀번호 값에 대한 액세스가 ACI를 통해 적절하게 제한되도록 해야 합니다. 또한, 102 페이지 “속성 값 암호화”에 설명된 것처럼 접미어의 속성 암호화도 구성해야 합니다.

▼ DIGEST-MD5 메커니즘을 구성하는 방법

다음 절차에서는 디렉토리 서버에서 DIGEST-MD5를 사용하도록 구성하는 방법에 대해 설명합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 ldapsearch 명령을 사용하여 DIGEST-MD5가 루트 항목의 supportedSASLMechanisms 속성 값인지 확인합니다.

예를 들어 다음 명령을 실행하면 현재 활성화되어 있는 SASL 메커니즘이 표시됩니다.

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -s base -b "" "(objectclass=*)" supportedSASLMechanisms
Enter bind password:
dn:
supportedSASLMechanisms: EXTERNAL
supportedSASLMechanisms: DIGEST-MD5
supportedSASLMechanisms: GSSAPI
^D
```

2 DIGEST-MD5가 활성화되어 있지 않으면 다음 명령을 사용하여 활성화합니다.

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=SASL, cn=security, cn=config
changetype: modify
add: dsSaslPluginsEnable
dsSaslPluginsEnable: DIGEST-MD5
-
replace: dsSaslPluginsPath
dsSaslPluginsPath: SASL-library
^D
```

여기서 *SASL-library*는 다음 값 중 하나입니다.

JES 설치 `/usr/lib/mps/sasl2`

Zip 설치 *install-path*`/dsee6/private/lib`

3 **DIGEST-MD5에 기본 아이디 매핑을 사용하거나 새 아이디 매핑을 만듭니다.**
자세한 내용은 123 페이지 “DIGEST-MD5 아이디 매핑”을 참조하십시오.

4 **DIGEST-MD5를 사용하여 SSL을 통해 서버에 액세스하는 모든 사용자의 비밀번호가 {CLEAR}에 저장되어 있는지 확인합니다.**
비밀번호 저장소 체계에 대해서는 7 장을 참조하십시오.

5 **SASL 구성 항목 또는 DIGEST-MD5 아이디 매핑 항목 중 하나를 수정했다면 디렉토리 서버를 다시 시작합니다.**

DIGEST-MD5 아이디 매핑

SASL 메커니즘에 아이디를 매핑하면 디렉토리의 사용자 항목이 SASL 아이디의 자격 증명과 일치하는지 확인합니다. 매핑 중에 SASL 아이디에 해당하는 DN을 찾을 수 없으면 인증은 실패합니다. 이 메커니즘에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

SASL 아이디는 각 메커니즘의 고유 형식으로 사용자를 나타내는 **사용자***(Principal)*라고 불리는 문자열입니다. DIGEST-MD5의 경우, 클라이언트에서는 `dn:` 접두어와 LDAP DN 또는 `u:` 접두어 뒤에 클라이언트가 지정한 텍스트가 있는 사용자(Principal)를 만들어야 합니다. 클라이언트가 보낸 사용자(Principal)는 매핑 중에 `${Principal}` 자리 표시자에서 사용할 수 있습니다.

DIGEST-MD5에 대한 기본 아이디 매핑은 서버 구성의 다음 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(Principal) `dn` 필드에 기존 디렉토리 사용자의 DN이 포함되어 있다고 가정합니다.

▼ DIGEST-MD5에 대한 사용자 고유의 아이디 매핑을 정의하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1. **기본 매핑 항목을 편집하거나 `cn=DIGEST-MD5,cn=identity mapping,cn=config`에 새 매핑 항목을 만듭니다.**

   다음 명령은 이 매핑이 어떻게 정의되는지 보여줍니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind 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. **새 매핑을 적용하려면 디렉토리 서버를 다시 시작합니다.**

GSSAPI를 통한 SASL 인증(Solaris OS에만 해당)

SASL에서 GSSAPI(Generic Security Service API)를 통해 Kerberos V5와 같은 타사 보안 시스템을 사용함으로써 클라이언트를 인증할 수 있습니다. GSSAPI 라이브러리는 Solaris OS SPARC® 플랫폼에서만 사용할 수 있습니다. Sun Enterprise Authentication Mechanism™ 1.0.1 서버에 Kerberos V5 구현을 설치하는 것이 좋습니다.

서버는 GSSAPI를 사용하여 사용자 아이디를 확인합니다. 그런 다음 SASL 메커니즘에서 GSSAPI 매핑 규칙을 적용하여 이 연결이 유지되는 동안 모든 작업의 바인드 DN으로 지정될 DN을 얻습니다.

▼ Kerberos 시스템을 구성하는 방법

제조업체의 지침에 따라 Kerberos 소프트웨어를 구성합니다. Sun Enterprise Authentication Mechanism 1.0.1 서버를 사용하는 경우 이 절차를 사용합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1. **`/etc/krb5`에 있는 파일을 구성합니다.**

2. **사용자와 서비스를 저장할 Kerberos 데이터베이스를 만듭니다.**

3 **데이터베이스에서 LDAP 서비스에 대한 사용자(Principal)를 만듭니다.**

```
$ ldap/server-FQDN@realm
```

여기서 *server-FQDN*은 디렉토리 서버의 정규화된 도메인 이름입니다.

4 **Kerberos 데몬 프로세스를 시작합니다.**

주 – DNS는 호스트 시스템에 구성해야 합니다.

자세한 단계별 지침은 소프트웨어 설명서 및 130 페이지 “SASL에서 GSSAPI를 사용하는 Kerberos 인증 구성의 예”를 참조하십시오.

▼ GSSAPI 메커니즘을 구성하는 방법

다음 절차에서는 Solaris OS에서 디렉토리 서버가 GSSAPI를 사용하도록 구성하는 방법에 대해 설명합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 **125 페이지 “GSSAPI 아이디 매핑”에 설명된 것처럼 GSSAPI에 대한 기본 아이디 매핑 및 사용자 정의 매핑을 만듭니다.**

2 **서비스 키를 저장할 키 탭을 만듭니다.**

LDAP 서비스 키는 키 탭에 저장됩니다.

a. **디렉토리 서버 사용자가 키 탭에 대한 읽기만 가능한지 확인합니다.**

b. **파일 이름을 기본 `/etc/krb5/krb5.keytab`과 다르게 변경합니다.**

c. **스크립트에서 환경 변수 `KRB5_KTNAME`을 설정하여 기본 키 탭 대신 새 키 탭을 사용하도록 합니다.**

3 **SASL 구성 항목 또는 GSSAPI 아이디 매핑 항목 중 하나를 수정했다면 디렉토리 서버를 다시 시작합니다.**

DNS는 호스트 시스템에 구성해야 합니다.

GSSAPI 아이디 매핑

SASL 메커니즘에 아이디를 매핑하면 디렉토리의 사용자 항목이 SASL 아이디의 자격 증명과 일치하는지 확인합니다. 매핑 중에 SASL 아이디에 해당하는 DN을 찾을 수 없으면 인증은 실패합니다.

SASL 아이디는 각 메커니즘의 고유 형식으로 사용자를 나타내는 **사용자***(Principal)*라고 불리는 문자열입니다. GSSAPI를 사용하는 Kerberos에서 사용자(Principal)는 형식이 *uid* [*/instance*] [*@realm*]인 아이디로 나타납니다. 여기서 *uid*에는 *instance* 식별자가 선택적으로 포함될 수 있으며 이 식별자 뒤에 일반적으로 도메인 이름인 *realm*이 선택적으로 붙습니다. 예를 들어 다음 문자열은 모두 유효한 사용자입니다.

```
bjensen
bjensen/Sales
bjensen@EXAMPLE.COM
bjensen/Sales@EXAMPLE.COM
```

처음에는 디렉토리에 GSSAPI 매핑이 정의되어 있지 않습니다. 클라이언트에서 자신이 사용하는 사용자(Principal)를 정의하는 방법에 따라 기본 매핑 및 필요한 사용자 정의 매핑을 모두 정의해야 합니다.

▼ GSSAPI에 대한 아이디 매핑을 정의하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 **`cn=GSSAPI,cn=identity mapping, cn=config`에 새 매핑 항목을 만듭니다.**

아이디 매핑 항목의 속성 정의에 대한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오. GSSAPI 매핑에 대한 예는 *instance-path*`/ldif/identityMapping_Examples.ldif`에 있습니다.

이 파일의 기본 GSSAPI 매핑에서는 사용자(Principal)에 사용자 아이디만 포함되어 있다고 가정합니다. 이 매핑에서는 디렉토리의 고정 분기에 있는 사용자를 결정합니다.

```
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 사용자(Principal)에 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 새 매핑을 적용하려면 디렉토리 서버를 다시 시작합니다.

LDAP 클라이언트에서 보안을 사용하도록 구성

다음 절에서는 디렉토리 서버와 보안 연결을 구성하려는 LDAP 클라이언트에서 SSL을 구성 및 사용하는 방법에 대해 설명합니다. SSL 연결에서 서버는 해당 인증서를 클라이언트로 보냅니다. 클라이언트는 먼저 인증서를 신뢰하여 서버를 인증해야 합니다. 그런 다음, 클라이언트는 두 SASL 메커니즘 중 하나에 대한 정보 또는 자신의 인증서를 보내서 클라이언트 인증 메커니즘 중 하나를 선택적으로 시작할 수 있습니다. SASL 메커니즘은 DIGEST-MD5 및 Kerberos V5를 사용하는 GSSAPI입니다.

다음 절에서는 SSL을 사용하는 LDAP 클라이언트의 예로 `ldapsearch` 도구를 사용합니다.

다른 LDAP 클라이언트에서 SSL 연결을 구성하려면 응용 프로그램과 함께 제공된 설명서를 참조하십시오.

주 – 일부 클라이언트 응용 프로그램은 SSL만 구현하고 서버에 신뢰할 수 있는 인증서가 있는지 확인하지 않으므로 SSL 프로토콜을 사용하여 데이터 암호화는 제공하지만 기밀성이나 사칭에 대한 보호는 보장할 수 없습니다.

다음 절에서는 LDAP 클라이언트에서 보안을 사용하도록 구성하는 방법에 대해 설명합니다.

클라이언트에 SASL DIGEST-MD5 사용

클라이언트에 DIGEST-MD5 메커니즘을 사용하는 경우에는 사용자 인증서를 설치할 필요가 없습니다. 하지만 암호화된 SSL 연결을 사용하려면 108 페이지 “인증서 관리”에 설명된 것처럼 서버 인증서를 신뢰해야 합니다.

영역 지정

영역은 선택한 인증 아이디가 속하는 이름 공간을 정의합니다. DIGEST-MD5 인증에서는 특정 영역에 인증해야 합니다.

디렉토리 서버는 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 찾을 수 있도록 SASL_PATH 환경 변수를 설정해야 합니다. DIGEST-MD5 라이브러리는 SASL 플러그인에 의해 동적으로 로드되는 공유 라이브러리입니다. SASL_PATH 환경 변수를 다음과 같이 설정합니다.

```
export SASL_PATH=SASL-library
```

이 경로에서는 디렉토리 서버가 LDAP 도구를 호출하는 호스트와 동일한 호스트에 설치되어 있다고 가정합니다.

ldapsearch 명령 예

SSL을 사용하지 않고 DIGEST-MD5 클라이언트 인증을 수행할 수 있습니다. 다음 예에서는 기본 DIGEST-MD5 아이디 매핑을 사용하여 바인드 DN을 결정합니다.

```
$ ldapsearch -h host1 -p 1389 \
 -o mech=DIGEST-MD5 [ \
 -o realm="example.com"] \
 -o authid="dn:uid=bjensen,dc=example,dc=com" \
 -w - \
 -o authzid="dn:uid=bjensen,dc=example,dc=com" \
 -o secProp="minssf=56,maxssf=256,noplain" \
 -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 이름이어야 합니다. 프록시 작업용 authzid를 사용하지 않는 경우에도 authid와 authzid는 둘 다 있어야 하며 동일한 값을 가져야 합니다. -w 비밀번호 옵션은 authid에 적용됩니다.

authid 값은 아이디 매핑에 사용되는 사용자(Principal)입니다. authid는 dn: 접두어가 사용되고 뒤에 디렉토리의 유효한 사용자 DN이 오거나 u: 접두어가 사용되고 뒤에 클라이언트에서 결정되는 문자열이 와야 합니다. 이렇게 authid를 사용하면 123 페이지 “DIGEST-MD5 아이디 매핑”에 표시된 매핑을 사용할 수 있습니다.

일반적으로 SSL 연결을 사용하여 LDAPS 보안 포트를 통해 암호화를 제공하고 DIGEST-MD5를 사용하여 클라이언트 인증을 제공하도록 구성합니다. 다음 예에서는 SSL을 통해 동일한 작업을 수행합니다.

```
$ ldapsearch -h host1 -P 1636 \
 -Z -P .mozilla/bjensen/BJE6001.slt/cert8.db \
 -N "cert-example" -w - \
 -o mech=DIGEST-MD5 [-o realm="example.com"] \
 -o authid="dn:uid=bjensen,dc=example,dc=com" \
 -o authzid="dn:uid=bjensen,dc=example,dc=com" \
```

```
-o secProp="minssf=0,maxssf=0,noplain" \
-b "dc=example,dc=com" "(givenname=Richard)"
```

이 예에서는 작업이 SSL을 통해 수행되므로 `ldapsearch` 명령에 `-N` 및 `-w` 옵션이 필요합니다. 그러나 클라이언트 인증에는 이 옵션이 사용되지 않습니다. 대신, 서버는 `authid` 값의 사용자(Principal)에 대해 다른 DIGEST-MD5 아이디 매핑을 수행합니다.

클라이언트에 Kerberos SASL GSSAPI 사용

클라이언트에 GSSAPI 메커니즘을 사용하는 경우 사용자 인증서를 설치할 필요는 없지만, Kerberos V5 보안 시스템을 구성해야 합니다. 또한, 암호화된 SSL 연결을 사용하려면 108 페이지 “인증서 관리”에 설명된 것처럼 서버 인증서를 신뢰해야 합니다.

▼ 호스트에 Kerberos V5를 구성하는 방법

LDAP 클라이언트를 실행할 호스트 시스템에 Kerberos V5를 구성해야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 설치 지침에 따라 Kerberos V5를 설치합니다.

Sun Enterprise Authentication Mechanism 1.0.1 클라이언트 소프트웨어를 설치하는 것이 좋습니다.

2 Kerberos 소프트웨어를 구성합니다.

Sun Enterprise Authentication Mechanism 소프트웨어를 사용하여 `/etc/krb5`에 파일을 구성합니다. 이 구성 작업에서 `kdc` 서버를 설정하고 기본 영역 및 Kerberos 시스템에 필요한 다른 구성을 정의합니다.

3 필요한 경우 `/etc/gss/mech` 파일을 수정하여 `kerberos_v5`를 첫 번째 값으로 표시합니다.

▼ Kerberos 인증에 대한 SASL 옵션을 지정하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 GSSAPI 메커니즘을 사용하는 클라이언트 응용 프로그램을 사용하기 전에 먼저 Kerberos 보안 시스템을 사용자(Principal)로 초기화합니다.

```
$ kinit user-principal
```

여기서 *user-principal*은 사용자의 SASL 아이디입니다(예: `bjensen@example.com`).

2 **Kerberos를 사용하도록 SASL 옵션을 지정합니다.**

UNIX 환경에서 SASL_PATH 환경 변수를 SASL 라이브러리에 대한 올바른 경로로 설정해야 합니다. Korn 쉘의 예는 다음과 같습니다.

```
$ export SASL_PATH=SASL-library
```

이 경로에서는 디렉토리 서버가 LDAP 도구를 호출하는 호스트와 동일한 호스트에 설치되어 있다고 가정합니다.

아래의 ldapsearch 도구 예에서는 -o(소문자 o) 옵션을 사용하여 Kerberos 사용에 대한 SASL 옵션을 지정하는 방법을 보여줍니다.

```
$ ldapsearch -h www.host1.com -p 1389 -o mech=GSSAPI -o authid="bjensen@EXAMPLE.COM" \
 -o authzid="bjensen@EXAMPLE.COM" -b "dc=example,dc=com" "(givenname=Richard)"
```

authid는 kinit 명령으로 초기화된 Kerberos 캐시에 있기 때문에 생략할 수 있습니다. authid가 있는 경우, 프록시 작업에 대한 authzid를 사용하지 않는 경우에도 authid와 authzid 값은 동일한 값을 가져야 합니다. authid 값은 아이디 매핑에 사용되는 사용자(Principal)입니다. 사용자(Principal)는 영역을 포함하는 완전한 사용자여야 합니다. 125 페이지 “GSSAPI 아이디 매핑”을 참조하십시오.

SASL에서 GSSAPI를 사용하는 Kerberos 인증 구성의 예

디렉토리 서버에 대한 Kerberos 구성 작업은 복잡할 수 있습니다. 먼저 Kerberos 설명서를 참조하십시오.

다음의 절차 예를 사용하면 수행할 단계를 고려하는 데 도움이 됩니다. 그러나 이러한 절차는 예로 제시된 것이므로 자신의 구성과 환경에 맞게 수정해야 합니다.

Solaris OS에서 Kerberos를 구성하고 사용하는 방법에 대한 자세한 내용은 **System Administration Guide: Security Services**를 참조하십시오. 이 설명서는 Solaris 설명서 세트의 일부로, 설명서 페이지를 참조할 수도 있습니다.

이 예와 해당 단계에 대한 내용은 다음과 같습니다.

1. 131 페이지 “이 예에 대한 가정”
2. 132 페이지 “모든 컴퓨터: Kerberos 클라이언트 구성 파일 편집”
3. 133 페이지 “모든 컴퓨터: Administration Server ACL 구성 파일 편집”
4. 133 페이지 “KDC 컴퓨터: KDC 서버 구성 파일 편집”
5. 134 페이지 “KDC 컴퓨터: KDC 데이터베이스 만들기”
6. 134 페이지 “KDC 컴퓨터: 관리 사용자(Principal) 및 키 탭 만들기”
7. 135 페이지 “KDC 컴퓨터: Kerberos 데몬 시작”
8. 135 페이지 “KDC 시스템: KDC 및 디렉토리 서버 시스템에 호스트 사용자 추가”
9. 135 페이지 “KDC 컴퓨터: Directory Server에 대한 LDAP 사용자(Principal) 추가”
10. 136 페이지 “KDC 컴퓨터: KDC에 테스트 사용자 추가”
11. 136 페이지 “디렉토리 서버 시스템: 디렉토리 서버 설치”
12. 137 페이지 “디렉토리 서버 컴퓨터: GSSAPI를 활성화하도록 디렉토리 서버 구성”

13. 138 페이지 “디렉토리 서버 시스템: 디렉토리 서버 키 탭 만들기 ”
14. 138 페이지 “디렉토리 서버 시스템: 디렉토리 서버에 테스트 사용자 추가”
15. 139 페이지 “디렉토리 서버 컴퓨터: 테스트 사용자로 Kerberos 티켓 얻기”
16. 140 페이지 “클라이언트 컴퓨터: GSSAPI를 통해 디렉토리 서버에 인증”

이 예에 대한 가정

이 절차 예에서는 컴퓨터 한 대가 KDC(Key Distribution Center)로 작동하고 보조 컴퓨터가 디렉토리 서버를 실행하도록 구성하는 프로세스를 설명합니다. 이 절차를 사용하면 사용자가 GSSAPI를 통해 Kerberos 인증을 수행할 수 있습니다.

동일한 컴퓨터에서 KDC와 디렉토리 서버를 모두 실행할 수 있습니다. 동일한 컴퓨터에서 모두 실행되도록 선택한 경우, 같은 절차를 사용하지만 이미 디렉토리 서버 컴퓨터에서 KDC 컴퓨터에 대해 수행한 단계는 생략합니다.

이러한 절차는 사용하는 환경에 대한 여러 가지 가정을 전제로 합니다. 절차 예를 사용하는 경우 사용자의 환경에 맞게 값을 적절히 수정합니다. 다음과 같이 가정할 수 있습니다.

- 이 시스템에는 Solaris 9 소프트웨어와 최신 권장 패치 클러스터가 설치되어 있습니다. 적절한 Solaris 패치가 설치되어 있지 않으면 디렉토리 서버에 대한 Kerberos 인증이 실패할 수 있습니다.

  문서화된 절차는 대체로 Solaris 10의 절차와 같지만 구성 파일 형식은 약간 다르며, 일부 명령의 출력도 다를 수 있습니다.
- Kerberos 데몬을 실행하는 컴퓨터는 `kdc.example.com`이라는 정규화된 도메인 이름을 갖습니다. 이 컴퓨터는 이름 지정 서비스로 DNS를 사용하도록 구성해야 합니다. 이는 Kerberos의 요구 사항으로서 `file`과 같은 다른 이름 지정 서비스를 대신 사용하면 특정 작업이 실패할 수 있습니다.
- 디렉토리 서버를 실행하는 컴퓨터는 `directory.example.com`이라는 정규화된 도메인 이름을 갖습니다. 이 컴퓨터도 이름 지정 서비스로 DNS를 사용하도록 구성해야 합니다.
- 디렉토리 서버 컴퓨터는 Kerberos를 통해 디렉토리 서버에 인증하는 작업에 대해 클라이언트 시스템 역할을 수행합니다. 이러한 인증 작업은 디렉토리 서버 및 Kerberos 데몬과 통신할 수 있는 모든 시스템에서 수행할 수 있지만 이 예에 필요한 구성 요소가 모두 디렉토리 서버에 제공되므로 인증 작업은 이 시스템에서 수행됩니다.
- 디렉토리 서버의 사용자는 `uid=`*`username`*`,ou=People,dc=example,dc=com` 형식의 DN을 가지며 해당하는 Kerberos 사용자(Principal)는 *username*`@EXAMPLE.COM`입니다. 다른 이름 지정 체계를 사용하는 경우에는 다른 GSSAPI 아이디 매핑을 사용해야 합니다.

모든 컴퓨터: Kerberos 클라이언트 구성 파일 편집

/etc/krb5/krb5.conf 구성 파일은 KDC와 통신하기 위해 Kerberos 클라이언트에서 필요로 하는 정보를 제공합니다.

KDC 컴퓨터, 디렉토리 서버 컴퓨터 및 Kerberos를 사용하여 디렉토리 서버에 인증할 클라이언트 컴퓨터의 /etc/krb5/krb5.conf 구성 파일을 편집합니다.

- "___default_realm___"이 나올 때마다 "EXAMPLE.COM"으로 바꿉니다.
- "___master_kdc___"가 나올 때마다 "kdc.example.com"으로 바꿉니다.
- Kerberos 서버가 하나만 있게 되므로 "___slave_kdcs___"가 포함된 행을 제거합니다.
- "___domain_mapping___"을 ".example.com = EXAMPLE.COM"(.example.com의 처음 마침표에 주의)으로 바꿉니다.

업데이트된 /etc/krb5/krb5.conf 구성 파일은 다음 예와 같습니다.

예 5–1 편집된 Kerberos 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
```

예 5–1 편집된 Kerberos 클라이언트 구성 파일 /etc/krb5/krb5.conf *(계속)*

```
# 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.acl 구성 파일에서 "___default_realm___"을 "EXAMPLE.COM"으로 바꿉니다. 업데이트된 파일은 다음 예와 같습니다.

예 5–2 편집된 Administration Server ACL 구성 파일

```
#
# Copyright (c) 1998-2000 by Sun Microsystems, Inc.
# All rights reserved.
#
# pragma ident   "@(#)kadm5.acl  1.1     01/03/19 SMI"
*/admin@EXAMPLE.COM *
```

KDC 컴퓨터: KDC 서버 구성 파일 편집

/etc/krb5/kdc.conf 파일을 편집하여 "___default_realm___"을 "EXAMPLE.COM"으로 바꿉니다. 업데이트된 파일은 다음 예와 같습니다.

예 5–3 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
```

예 5–3 편집된 KDC 서버 구성 파일 /etc/krb5/kdc.conf *(계속)*

```
[realms]
        EXAMPLE.COM = {
                profile = /etc/krb5/krb5.conf
                database_name = /var/krb5/principal
                admin_keytab = /etc/krb5/kadm5.keytab
                acl_file = /etc/krb5/kadm5.acl
                kadmind_port = 749
                max_life = 8h 0m 0s
                max_renewable_life = 7d 0h 0m 0s
                default_principal_flags = +preauth
        }
```

KDC 컴퓨터: KDC 데이터베이스 만들기

```
$ /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: password
Re-enter KDC database master key to verify: password
$
```

KDC 컴퓨터: 관리 사용자(Principal) 및 키 탭 만들기

다음 명령을 사용하여 관리 데몬에서 사용할 kws/admin@EXAMPLE.COM 사용자(Principal) 및 서비스 키가 있는 관리 사용자를 만듭니다.

```
$ /usr/sbin/kadmin.local
kadmin.local:  add_principal kws/admin
Enter password for principal "kws/admin@EXAMPLE.COM": secret
Re-enter password for principal "kws/admin@EXAMPLE.COM": secret
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$
```

KDC 컴퓨터: Kerberos 데몬 시작

다음 명령을 실행하여 KDC 및 관리 데몬을 시작합니다.

```
$ /etc/init.d/kdc start
$ /etc/init.d/kdc.master start
$
```

KDC 프로세스는 프로세스 목록에서 `/usr/lib/krb5/krb5kdc`로 표시되고 관리 데몬은 `/usr/lib/krb5/kadmind`로 표시됩니다.

Solaris 10 OS의 경우 데몬은 SMF(Service Management Facility) 프레임워크에서 관리합니다. 따라서, Solaris 10 OS에서는 다음 명령을 사용하여 데몬을 시작합니다.

```
$ svcadm disable network/security/krb5kdc
$ svcadm enable network/security/krb5kdc
$ svcadm disable network/security/kadmin
$ svcadm enable network/security/kadmin
$
```

KDC 시스템: KDC 및 디렉토리 서버 시스템에 호스트 사용자 추가

다음 명령 시퀀스를 사용하여 KDC 및 디렉토리 서버 컴퓨터의 Kerberos 데이터베이스에 호스트 사용자를 추가합니다. 호스트 사용자는 `klist`와 같은 특정 Kerberos 유틸리티에서 사용합니다.

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
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
$
```

KDC 컴퓨터: Directory Server에 대한 LDAP 사용자(Principal) 추가

디렉토리 서버에서 인증 사용자가 보유한 Kerberos 티켓을 확인할 수 있으려면 디렉토리 서버에 자체 사용자(Principal)가 있어야 합니다. 현재 디렉토리 서버는 하드 코드되어 있으므로 `ldap/`*fqdn*`@`*realm*의 사용자(Principal)가 필요합니다. 여기서 *fqdn*은 디렉토리

서버의 정규화된 도메인 이름이며 *realm*은 Kerberos 영역입니다. *fqdn*은 디렉토리 서버를 설치할 때 제공된 정규화된 이름과 일치해야 합니다. 이 경우, 디렉토리 서버의 사용자(Principal)는 `ldap/directory.example.com@EXAMPLE.COM`이 됩니다.

다음 명령 시퀀스를 사용하여 디렉토리 서버에 대한 LDAP 사용자(Principal)를 만듭니다.

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin: add_principal -randkey ldap/directory.example.com
Principal "ldap/directory.example.com@EXAMPLE.COM" created.
kadmin: quit
$
```

KDC 컴퓨터: KDC에 테스트 사용자 추가

Kerberos 인증을 수행하려면 사용자 인증이 Kerberos 데이터베이스에 있어야 합니다. 이 예에서 사용자는 `kerberos-test`라는 사용자 이름을 갖게 되며, 이는 Kerberos 사용자(Principal)가 `kerberos-test@EXAMPLE.COM`이라는 의미입니다.

다음 예의 명령 시퀀스를 사용하여 사용자를 만듭니다.

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  add_principal kerberos-test
Enter password for principal "kerberos-test@EXAMPLE.COM": secret

Re-enter password for principal "kerberos-test@EXAMPLE.COM": secret

Principal "kerberos-test@EXAMPLE.COM" created.
kadmin:  quit
$
```

디렉토리 서버 시스템: 디렉토리 서버 설치

Directory Server 6.0과 최신 패치를 설치합니다. 설정 예는 다음과 같습니다.

변수 유형	예 값
정규화된 컴퓨터 이름	`directory.example.com`
설치 디렉토리	`/opt/SUNWdsee`
인스턴스 경로	`/local/ds`
서버 사용자	`unixuser`
서버 그룹	`unixgroup`

변수 유형	예 값
서버 포트	389
접미어	dc=example,dc=com

디렉토리 서버 컴퓨터: GSSAPI를 활성화하도록 디렉토리 서버 구성

먼저 /data/ds/shared/bin/gssapi.ldif 파일을 만들어 디렉토리 서버에서 사용할 매핑을 정의하고 사용자(Principal)를 기반으로 인증할 Kerberos 사용자를 식별합니다. 다음 예와 동일한 내용으로 파일을 만듭니다.

예 5–4 gssapi.ldif 파일 내용

```
dn: 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: /usr/lib/mps/sasl2/libsasl.so
```

다음으로, 아래 예와 같이 ldapmodify 명령을 사용하여 적절한 매핑으로 GSSAPI를 활성화하도록 디렉토리 서버를 업데이트합니다.

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -a -f /data/ds/shared/bin/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
$
```

디렉토리 서버 시스템: 디렉토리 서버 키 탭 만들기

앞서 언급한 바와 같이 디렉토리 서버는 GSSAPI를 통해 Kerberos 사용자를 인증하기 위해 KDC에 자체 사용자(Principal)가 있어야 합니다. 인증 작업이 올바로 작동하려면 이 사용자(Principal) 정보는 디렉토리 서버 컴퓨터의 Kerberos 키 탭에 있어야 합니다. 또한, 이 정보는 디렉토리 서버가 작동하는 사용자 계정에서 읽을 수 있는 파일 형식이어야 합니다.

다음 명령 시퀀스를 사용하여 올바른 등록 정보를 가진 키 탭 파일을 만듭니다.

```
$ /usr/sbin/kadmin -p kws/admin
Enter Password: secret
kadmin:  ktadd -k //local/ds/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
 WRFILE:/local/ds/config/ldap.keytab.
kadmin:  quit
$
```

이 사용자 정의 키 탭의 사용 권한과 소유권을 변경합니다. 키 탭은 디렉토리 서버를 실행하는 데 사용되는 사용자 계정에서 소유해야 하며 해당 사용자만 읽을 수 있어야 합니다.

```
$ chown unixuser:unixgroup /local/ds/config /ldap.keytab
$ chmod 600 /local/ds/config/ldap.keytab
$
```

기본적으로 디렉토리 서버는 `/etc/kerb5/krb5.keytab` 파일의 표준 Kerberos 키 탭을 사용하려고 합니다. 그러나 이 파일을 디렉토리 서버 사용자가 읽을 수 있도록 만들면 보안상 위험이 따르며, 이러한 이유로 디렉토리 서버에 사용자 정의 키 탭을 만들어야 합니다.

새 사용자 정의 키 탭을 사용하도록 디렉토리 서버를 구성합니다. `KRB5_KTNAME` 환경 변수를 설정하여 이 작업을 수행합니다.

마지막으로, 이러한 변경 사항이 적용되도록 디렉토리 서버를 다시 시작합니다.

```
$ KRB5_KTNAME=/etc/krb5/ldap.keytab dsadm restart /local/ds
```

디렉토리 서버 시스템: 디렉토리 서버에 테스트 사용자 추가

디렉토리 서버에 Kerberos 사용자를 인증하기 위해서는 해당 사용자의 Kerberos 사용자(Principal)에 해당하는 사용자에 대한 디렉토리 항목이 있어야 합니다.

이전 단계에서 `kerberos-test@EXAMPLE.COM` 사용자(Principal)가 있는 Kerberos 데이터베이스에 테스트 사용자가 추가되었습니다. 아이디 매핑 구성이 디렉토리에 추가되었기 때문에 그 사용자에 해당하는 디렉토리 항목에는 `uid=kerberos-test,ou=People,dc=example,dc=com`이라는 DN이 있어야 합니다.

사용자를 디렉토리에 추가하려면 다음 내용으로 testuser.ldif 파일을 만들어야 합니다.

예 5–5 새 testuser.ldif 파일

```
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, ldapmodify를 사용하여 이 항목을 서버에 추가합니다.

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f testuser.ldif
adding new entry uid=kerberos-test,ou=People,dc=example,dc=com
$
```

디렉토리 서버 컴퓨터: 테스트 사용자로 Kerberos 티켓 얻기

테스트 사용자는 Kerberos 데이터베이스와 디렉토리 서버 및 KDC에 있습니다. 따라서, 이제 GSSAPI를 사용하여 Kerberos를 통해 테스트 사용자로 디렉토리 서버에 인증할 수 있습니다.

먼저 아래 예와 같이 kinit 명령을 사용하여 이 사용자에 대한 Kerberos 티켓을 얻습니다.

```
$ kinit kerberos-test
Password for kerberos-test@EXAMPLE.COM: secret
$
```

그런 다음 klist 명령을 사용하여 이 티켓에 대한 정보를 확인합니다.

```
$ 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
$
```

클라이언트 컴퓨터: GSSAPI를 통해 디렉토리 서버에 인증

마지막 단계로 GSSAPI를 사용하여 디렉토리 서버에 인증합니다. 디렉토리 서버에 제공된 `ldapsearch` 유틸리티는 GSSAPI, DIGEST-MD5 및 EXTERNAL 메커니즘을 비롯하여 SASL 인증에 대한 지원을 제공합니다. 그러나, GSSAPI를 사용하여 바인드하려면 클라이언트에 SASL 라이브러리에 대한 경로를 제공해야 합니다. 이 경로를 제공하려면 `SASL_PATH` 환경 변수를 `lib/sasl` 디렉토리로 설정합니다.

```
$ SASL_PATH=SASL-library
$ export SASL_PATH
$
```

`ldapsearch`를 사용하여 디렉토리 서버에 대한 Kerberos 기반의 인증을 올바르게 수행하려면 `-o mech=GSSAPI` 및 `-o authzid=`*principal* 인수를 포함해야 합니다.

또한, 여기에 표시된 `-h directory.example.com`과 같이 정규화된 호스트 이름도 지정해야 합니다. 여기서 이 값은 서버의 `cn=config`에 대한 `nsslapd-localhost` 속성 값과 일치해야 합니다. GSSAPI 인증 프로세스 중 클라이언트가 제공한 호스트 이름이 서버에서 제공한 호스트 이름과 일치해야 하기 때문에 위와 같이 `-h` 옵션을 사용해야 합니다.

다음 예는 이전에 만든 Kerberos 테스트 사용자 계정으로 인증되는 동안 `dc=example,dc=com` 항목을 검색합니다.

```
$ ldapsearch -h directory.example.com -p 389 -o mech=GSSAPI \
 -o authzid="kerberos-test@EXAMPLE.COM" -b "dc=example,dc=com" -s base "(objectClass=*)"
version: 1
dn: dc=example,dc=com
dc: example
objectClass: top
objectClass: domain
$
```

디렉토리 서버 액세스 로그를 검사하여 인증이 예상대로 처리되었는지 확인합니다.

```
$ tail -12 /local/ds/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.
$
```

이 예에서 바인드는 3단계 프로세스로서, 처음 두 단계에서는 LDAP 결과 14(처리 중인 SASL 바인드)가 반환되었으며 세 번째 단계에서는 해당 바인드가 성공적으로 처리되었음을 보여줍니다. `method=sasl` 및 `mech=GSSAPI` 태그는 이 바인드에서 GSSAPI SASL 메커니즘을 사용했음을 나타내며, 성공적으로 처리된 바인드 응답의 끝 부분에 있는 `dn="uid=kerberos-test,ou=people,dc=example,dc=com"`은 바인드가 적합한 사용자로 수행되었음을 나타냅니다.

PTA(Pass-Through Authentication)

PTA(Pass-through authentication)는 바인드 요청이 바인드 DN으로 필터링되는 메커니즘입니다. 한 디렉토리 서버(위임자)가 바인드 요청을 수신하고 필터를 기반으로 다른 디렉토리 서버(대리자)가 바인드 요청을 인증합니다. 이 기능의 일부로, PTA 플러그인을 사용하면 위임자 디렉토리 서버가 해당 로컬 데이터베이스에 반드시 저장할 필요가 없는 항목에 대해 비밀번호 기반의 단순 바인드 작업을 허용할 수 있습니다.

PTA 플러그인은 서버와의 개인 통신을 위해 DSCC에서도 사용됩니다. 서버 인스턴스가 DSCC에 등록될 때 PTA 플러그인이 활성화되며 DSCC URL이 인수로 추가됩니다.

```
$ dsconf get-plugin-prop -h host -p port "Pass Through Authentication" enabled argument
argument  :  ldap://DSCC_URL:DSCC_PORT/cn=dscc
enabled   :  on
```

주 – 가능한 경우 사용자 고유 용도로 PTA 플러그인을 수정하지 마십시오. PTA 플러그인을 수정하면 DSCC에 대한 액세스 문제가 발생할 수 있습니다.

PTA 플러그인을 수정해야 하는 경우에는 다음 작업을 수행해야 합니다.

- `enabled` 등록 정보를 `on`으로 유지합니다.
- `argument` 등록 정보에 다른 값을 추가할 수 있는 경우에도 DSCC URL을 인수로 유지합니다.

PTA 플러그인이 비활성화되거나 DSCC URL이 인수에서 제거되면 서버 인스턴스는 DSCC에서 `inaccessible`로 표시됩니다. 이 경우, DSCC에서 PTA 플러그인을 재설정하라는 옵션이 자동으로 표시됩니다.

◆ ◆ ◆ 6 장

6 디렉토리 서버 액세스 제어

디렉토리에 대한 액세스 제어는 보안 디렉토리 생성에 있어 필요한 부분입니다. 이 장에서는 디렉토리에 액세스하는 사용자에게 부여되는 권한을 결정하는 액세스 제어 지침(ACI)에 대해 설명합니다.

전체 보안 정책을 제공하는 액세스 제어 전략은 디렉토리 배포의 계획 단계에서 정의합니다. 액세스 제어 전략을 계획하는 방법은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**를 참조하십시오.

ACI 구문 및 바인드 규칙을 포함하여 ACI에 대한 추가 정보는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

ACI 작성, 보기 및 수정

디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC) 또는 명령줄을 사용하여 ACI를 작성할 수 있습니다. 어떤 방법을 선택하든 처음부터 ACI를 새로 작성하는 것보다 기존 ACI 값을 보고 복사하는 것이 보다 간편합니다.

DSCC에서 `aci` 속성 값을 보고 수정할 수 있습니다. DSCC에서 ACI를 수정하는 방법에 대한 자세한 내용은 DSCC 온라인 도움말을 참조하십시오.

▼ ACI를 작성, 수정 및 삭제하는 방법

명령줄을 사용하여 ACI를 작성하려면 먼저 LDIF 명령문을 사용하여 ACI를 파일로 작성합니다. 그런 다음 `ldapmodify` 명령을 사용하여 ACI를 디렉토리 트리에 추가합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 ACI를 LDIF 파일로 작성합니다.

```
dn: dc=example,dc=com
changetype: modify
add: aci
aci: (target)(version 3.0; acl "name";permission bindrules;)
```

이 예에서는 ACI를 추가하는 방법을 보여줍니다. ACI를 수정하거나 삭제하려면 `add`를 `replace` 또는 `delete`로 대체하십시오.

일반적으로 사용되는 ACI의 예는 145 페이지 "액세스 제어 사용 예"를 참조하십시오.

2 LDIF 파일을 사용하여 변경합니다.

```
$ ldapmodify -h host -p port -D cn=admin,cn=Administrators,cn=config -w - -f ldif-file
```

▼ ACI 속성 값을 보는 방법

ACI는 하나 이상의 `aci` 속성 값으로 항목에 저장됩니다. `aci` 속성은 여러 값을 갖는 작동 가능 속성으로, 디렉토리 사용자가 읽고 수정할 수 있으므로 ACI에서 ACI 속성 자체를 보호해야 합니다. 일반적으로 관리 사용자는 `aci` 속성에 대한 전체 액세스 권한을 가집니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **다음 `ldapsearch` 명령을 실행하여 항목의 ACI 속성 값을 봅니다.**

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -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. LDIF 출력에 연속된 줄이 포함되지 않게 하려면 `-T` 옵션을 사용합니다. 출력 형식을 고려해서 LDIF 출력을 복사하여 붙여넣습니다.

주 – `aci` 값이 부여하거나 거부하는 권한을 보려면 158 페이지 “유효 권한 보기”를 참조하십시오.

▼ 루트 수준에서 ACI를 보는 방법

접미어를 작성할 때 일부 기본 ACI는 최상위 또는 루트 수준에서 작성됩니다. 이러한 ACI는 기본 관리 사용자 `cn=admin,cn=Administrators,cn=config`에게 디렉토리 관리자와 동일한 디렉토리 데이터 액세스 권한을 허용합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **기본 루트 수준 ACI를 봅니다.**

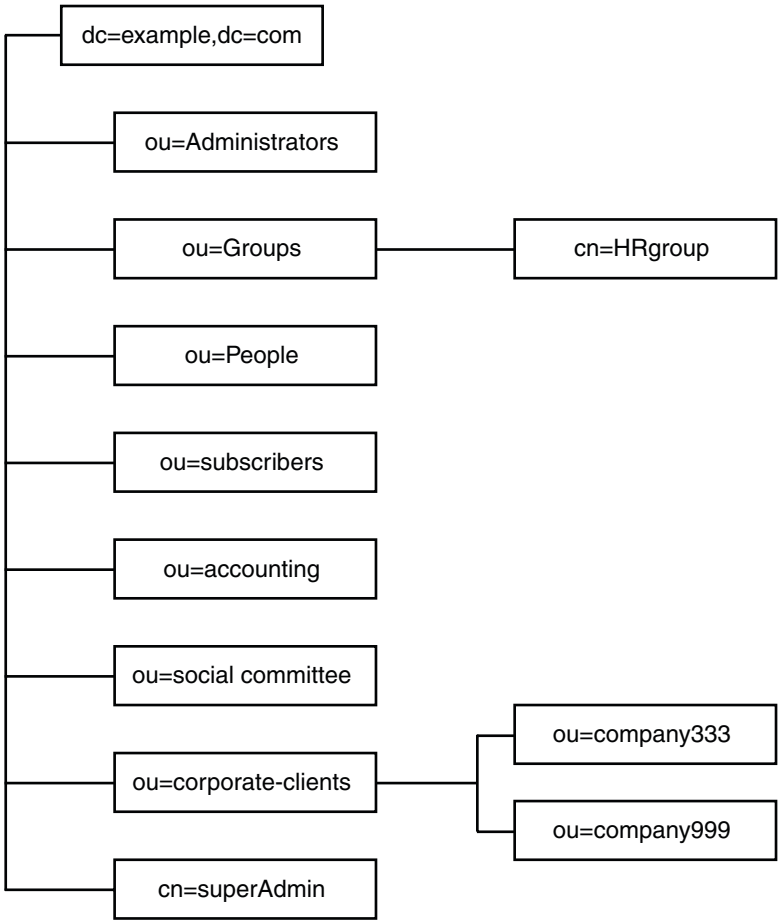
```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "" -s base "(objectclass=*)" aci
```

액세스 제어 사용 예

이 절에서는 가상의 ISP 업체인 Example.com에서 액세스 제어 정책을 구현하는 방법에 대한 예를 보여줍니다.

또한 *install_path*`/ds6/ldif/Example.ldif` 설치와 함께 제공된 LDIF 파일 예에서 ACI 예를 찾을 수 있습니다.

모든 예는 LDIF 파일을 사용하여 지정된 작업을 수행하는 방법에 대해 설명합니다. 다음 그림은 example.com 디렉토리 정보 트리를 그래픽 형식으로 보여줍니다.



Example.com은 웹 호스팅 서비스와 인터넷 액세스를 제공하며 웹 호스팅 서비스의 일부로 클라이언트 기업의 디렉토리를 호스팅합니다. Example.com은 실제로 두 개 중소 기업(Company333과 Company999)의 디렉토리를 호스팅하고 있으며 일부 관리 작업도 수행합니다. 또한 다수의 개인 가입자에게 인터넷 액세스를 제공합니다.

Example.com에서 적용하려는 액세스 규칙은 다음과 같습니다.

- Example.com 직원들에게 전체 Example.com 트리를 읽고, 검색 및 비교할 수 있는 익명 액세스 권한 부여. 147 페이지 "익명 액세스 권한 부여"를 참조하십시오.
- Example.com 직원들에게 `homeTelephoneNumber`, `homeAddress` 등의 개인 정보에 대한 쓰기 권한 부여. 148 페이지 "개인 항목에 대한 쓰기 액세스 권한 부여"를 참조하십시오.

- Example.com 가입자에게 회사 연락처 정보에서 `dc=example,dc=com` 항목에 대한 읽기 권한을 부여하고 그 아래 항목에 대한 읽기 권한 거부. 149 페이지 “특정 수준의 액세스 허용”을 참조하십시오.
- Example.com 직원들에게 중요한 특정 역할을 제외한 모든 역할을 자신의 항목에 추가할 수 있는 권한 부여. 150 페이지 “중요 역할에 대한 액세스 제한”을 참조하십시오.
- 특정 관리자에게 접미어에 대한 디렉토리 관리자와 동일한 권한 부여. 151 페이지 “모든 접미어에 대한 전체 액세스 권한 부여”를 참조하십시오.
- Example.com Human Resources 그룹에 People 분기의 항목에 대한 모든 권한 부여. 151 페이지 “그룹에 접미어에 대한 전체 액세스 권한 부여”를 참조하십시오.
- Example.com 직원들에게 디렉토리의 Social Committee 분기에 그룹 항목을 작성할 수 있는 권한과 소유한 그룹 항목을 삭제할 수 있는 권한 부여. 152 페이지 “그룹 항목 추가 및 삭제 권한 부여”를 참조하십시오.
- Example.com 직원들에게 디렉토리의 Social Committee 분기의 그룹 항목에 자신을 추가할 수 있는 권한 부여. 153 페이지 “사용자가 그룹에 자신을 추가 또는 제거할 수 있도록 허용”을 참조하십시오.
- 특정 조건을 사용하여 Company333과 Company999의 디렉토리 어드민 관리자(역할)에게 디렉토리 트리의 해당 분기에 대한 액세스 권한 부여. 이러한 조건에는 SSL 인증, 시간 및 날짜 제한, 지정된 위치 등이 포함됩니다. 154 페이지 “그룹이나 역할에 조건부 액세스 권한 부여”를 참조하십시오.
- 개인 가입자에게 자신의 항목에 대한 액세스 권한 부여. 148 페이지 “개인 항목에 대한 쓰기 액세스 권한 부여”를 참조하십시오.
- 개인 가입자가 자신의 항목에 있는 결제 정보에 액세스하지 못하도록 거부. 155 페이지 “액세스 거부”를 참조하십시오.
- 특별히 목록에 표시하지 않도록 요청한 가입자를 제외하고 개인 가입자 하위 트리에 대한 익명 액세스 권한 부여. 원하는 경우, 디렉토리에서 이 부분은 방화벽 외부의 읽기 전용 서버로 하루에 한 번 업데이트될 수 있습니다. 147 페이지 “익명 액세스 권한 부여” 및 157 페이지 “필터링을 사용한 대상 설정”을 참조하십시오.

익명 액세스 권한 부여

대부분의 디렉토리는 읽기, 검색 또는 비교를 위해 하나 이상의 접미어에 익명으로 액세스할 수 있도록 구성됩니다. 전화 번호부 등 직원들이 검색할 수 있는 인사 디렉토리를 실행하는 경우 이러한 권한을 설정할 수 있습니다. Example.com 내부의 경우가 이에 해당하며 148 페이지 “ACI "Anonymous Example.com""에서 자세히 설명합니다.

또한 Example.com은 ISP로서 누구든지 이용할 수 있는 공공 전화 번호부를 작성하여 모든 가입자의 연락처 정보를 제공하려고 합니다. 여기에 대해서는 148 페이지 “ACI "Anonymous World""에서 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; acl "Anonymous
 example"; allow (read, search, compare)
 userdn= "ldap:///anyone") ;)
```

이 예에서는 `dc=example,dc=com` `entry`에 `aci`를 추가한다고 가정합니다. `userPassword` 속성은 ACI 범위에서 제외됩니다.

주 – 비밀번호 속성을 보호하기 위한 예에서 동일한 구문을 사용하여 표시해서는 안되는 속성과 기밀 속성을 보호합니다(`targetattr !="`*attribute-name*`"`).

ACI "Anonymous World"

모든 사람에게 개인 가입자 하위 트리에 대한 읽기 및 검색 권한을 부여하는 동시에 목록에 표시하지 않을 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 `yes` 또는 `no`로 설정된 `unlistedSubscriber` 속성이 있다고 가정합니다. 대상 정의는 이 속성 값을 기준으로 목록에 없는 가입자를 필터링합니다. 필터 정의에 대한 자세한 내용은 157 페이지 “필터링을 사용한 대상 설정”을 참조하십시오.

개인 항목에 대한 쓰기 액세스 권한 부여

대부분의 디렉토리 어드민 관리자는 내부 사용자가 자신의 항목에 있는 일부 속성만 변경할 수 있도록 설정합니다. Example.com의 디렉토리 어드민 관리자도 사용자가 자신의 비밀번호, 집 전화 번호 및 집 주소만 변경할 수 있도록 설정하려고 합니다. 자세한 내용은 149 페이지 “ACI "Write Example.com""을 참조하십시오.

또한 Example.com에는 가입자가 디렉토리에 대한 SSL 연결을 설정할 경우 Example.com 트리에서 자신의 개인 정보를 업데이트하도록 허용하는 정책이 있습니다. 자세한 내용은 149 페이지 “ACI "Write Subscribers""를 참조하십시오.

ACI "Write Example.com"

주 – 이 권한을 설정하면 사용자에게 속성 값을 삭제할 수 있는 권한도 부여하게 됩니다.

Example.com 직원들에게 집 전화 번호 및 집 주소를 업데이트할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="homePhone ||
 homePostalAddress")(version 3.0; acl "Write Example.com";
 allow (write) userdn="ldap:///self" ;)
```

이 예에서는 `ou=People,dc=example,dc=com` 항목에 ACI를 추가한다고 가정합니다.

ACI "Write Subscribers"

주 – 이 권한을 설정하면 사용자에게 속성 값을 삭제할 수 있는 권한도 부여하게 됩니다.

Example.com 가입자에게 집 전화 번호를 업데이트할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="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 추가하고 사용자가 SSL을 사용하여 바인드해야 한다고 가정합니다.

Example.com 가입자는 해당 속성을 삭제할 수 있기 때문에 집 주소에 대한 쓰기 액세스 권한이 없습니다. 집 주소는 Example.com에서 청구하는 데 필요한 업무상 중요한 정보입니다.

특정 수준의 액세스 허용

디렉토리 트리 내에서 다른 수준에 영향을 미치는 ACI 범위를 설정하여 허용할 액세스 수준을 미세 조정할 수 있습니다. 대상 ACI 범위를 다음 중 하나로 설정할 수 있습니다.

`base` 항목 자체

`onelevel` 항목 자체와 한 수준 아래의 모든 항목

`subtree` 항목 자체와 해당 항목 아래의 모든 항목(제한 없음)

ACI "Read Example.com only"

Example.com 가입자에게 회사 연락처 정보의 dc=example,dc=com 항목에 대한 읽기 권한을 부여하고 그 아래 항목에 대한 액세스 권한은 허용하지 않으려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetscope="base") (targetattr="*")(version 3.0;
 acl "Read Example.com only";  allow (read,search,compare)
 userdn="ldap:///cn=*,ou=subscribers,dc=example,dc=com";)
```

이 예에서는 dc=example, dc=com 항목에 ACI를 추가한다고 가정합니다.

중요 역할에 대한 액세스 제한

디렉토리에서 역할 정의를 사용하여 네트워크 및 디렉토리 관리와 같은 업무상 중요한 기능을 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 특정 사이트에서 응급 조치 교육을 이수한 모든 직원 구성원을 포함하는 First Aid 역할을 작성할 수 있습니다. 역할 정의 작성에 대한 자세한 내용은 214 페이지 “역할 관리”를 참조하십시오.

중요한 기업 또는 비즈니스 기능에 대한 사용자 권한을 부여하는 역할이 있을 경우 이 역할에 대한 액세스를 제한해 보십시오. 예를 들어 Example.com의 직원들은 다음 예에 표시된 것처럼 superAdmin 역할을 제외한 모든 역할을 자신의 항목에 추가할 수 있습니다.

ACI "Roles"

Example.com 직원들에게 superAdmin 역할을 제외한 모든 역할을 자신의 항목에 추가할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="*") (targattrfilters="add=nsRoleDN:
 (nsRoleDN !="cn=superAdmin, dc=example, dc=com")")
 (version 3.0; acl "Roles"; allow (write)
 userdn= "ldap:///self" ;)
```

이 예에서는 ou=People,dc=example,dc=com 항목에 ACI를 추가한다고 가정합니다.

모든 접미어에 대한 전체 액세스 권한 부여

특정 사용자에게 접미어에 대해 디렉토리 관리자와 동일한 권한을 부여하는 것이 좋은 경우도 있습니다. Example.com에서 Kirsten Vaughan은 디렉토리 서버의 관리자로서 `superAdmin` 역할을 가지고 있습니다. 이 역할의 이점은 다음과 같습니다.

- 관리자가 직접 바인드하여 SSL과 같은 강력한 인증을 사용할 수 있기 때문에 보안 기능이 향상됩니다.
- 디렉토리 관리자 비밀번호가 사람들에게 거의 알려지지 않기 때문에 보안 기능이 향상됩니다.
- 로깅을 통한 추적 가능성이 향상됩니다.

주 – Kirsten Vaughan을 `cn=Administrators,cn=config` 그룹에 추가하면 디렉토리 관리자와 동일한 권한이 부여됩니다.

전체 서버에 대해 디렉토리 관리자와 동일한 권한을 부여하려면 70 페이지 “루트 액세스 권한이 있는 관리 사용자를 만드는 방법”의 절차를 수행합니다.

ACI "Full Access"

관리자 Kirsten Vaughan에게 디렉토리 관리자와 동일한 권한을 부여하려면 LDIF로 아래 명령문을 사용합니다.

```
aci: (targetattr="*") (version 3.0; acl "Full Access";
 allow (all) groupdn= "ldap:///cn=SuperAdmin,dc=example,dc=com"
 and authmethod="ssl" ;)
```

이 예에서는 루트 항목""(텍스트 없음)에 ACI를 추가한다고 가정합니다.

그룹에 접미어에 대한 전체 액세스 권한 부여

대부분의 디렉토리에는 기업의 특정 기능을 식별하는 그룹이 있습니다. 이러한 그룹에 디렉토리의 모두 또는 일부에 대한 액세스 권한을 부여할 수 있습니다. 그룹에 액세스 권한을 적용하면 각 구성원에 대해 개별적으로 액세스 권한을 설정할 필요가 없습니다. 대신, 사용자를 그룹에 추가하여 액세스 권한을 부여할 수 있습니다.

예를 들어 디렉토리 서버 인스턴스를 작성하는 경우 디렉토리에 대한 전체 액세스 권한이 있는 관리자 그룹 `cn=Administrators,cn=config`가 기본적으로 작성됩니다.

Example.com에서 Human Resources 그룹은 디렉토리의 `ou=People` 분기에 대한 전체 액세스 권한을 갖고 있으므로 152 페이지 “ACI "HR""에 표시된 것처럼 직원 디렉토리를 업데이트할 수 있습니다.

ACI "HR"

디렉토리의 employee 분기에 대한 모든 권한을 HR 그룹에 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="*") (version 3.0; acl "HR"; allow (all)
  groupdn= "ldap:///cn=HRgroup,ou=Groups,dc=example,dc=com";)
```

이 예에서는 다음 항목에 ACI를 추가한다고 가정합니다.

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

그룹 항목 추가 및 삭제 권한 부여

일부 조직에서 직원들은 효율성을 높이고 사기를 불어넣는 항목을 트리에 작성할 수 있습니다. 예를 들어 Example.com에는 테니스, 수영, 스키, 롤 플레잉 등 여러 클럽으로 구성된 사교 모임이 있습니다.

152 페이지 “ACI "Create Group"”에 표시된 것처럼 Example.com의 직원이라면 누구든지 새 클럽을 나타내는 그룹 항목을 작성할 수 있습니다.

153 페이지 “사용자가 그룹에 자신을 추가 또는 제거할 수 있도록 허용”에 표시된 것처럼 Example.com의 모든 직원은 이러한 그룹 중 하나의 구성원이 될 수 있습니다.

153 페이지 “ACI "Delete Group"”에 표시된 것처럼 그룹 소유자만 그룹 항목을 수정하거나 삭제할 수 있습니다.

ACI "Create Group"

Example.com 직원들에게 ou=Social Committee 분기에 그룹 항목을 작성할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="*") (targattrfilters="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";)
```

이 예에서는 ou=Social Committee,dc=example,dc=com 항목에 ACI를 추가한다고 가정합니다.

주 –

- 이 ACI는 쓰기 권한을 부여하지 않기 때문에 항목 작성자가 항목을 수정할 수 없습니다.
- 서버가 이면에서 `top` 값을 추가하므로 `targattrfilters` 키워드에 `objectClass=top`을 지정해야 합니다.
- 이 ACI는 클라이언트 시스템이 `example.com` 도메인에 있도록 제한합니다.

ACI "Delete Group"

Example.com 직원들에게 `ou=Social Committee` 분기에서 그들이 속하는 그룹의 항목을 수정하거나 삭제할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr = "*") (targattrfilters="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 `aci`를 추가한다고 가정합니다.

DSCC를 사용하여 이 ACI를 작성하려면 수동 편집 모드를 사용하여 대상 필터를 작성하고 그룹 소유권을 확인해야 하기 때문에 그다지 효율적이지 않습니다.

사용자가 그룹에 자신을 추가 또는 제거할 수 있도록 허용

대부분의 디렉토리는 사용자가 메일 목록과 같이 그룹에 자신을 추가하거나 제거할 수 있도록 허용하는 ACI를 설정합니다.

153 페이지 “ACI "Group Members""에 표시된 것처럼 Example.com의 직원들은 `ou=Social Committee` 하위 트리 아래의 모든 그룹 항목에 자신을 추가할 수 있습니다.

ACI "Group Members"

Example.com 직원들에게 그룹에 자신을 추가할 수 있는 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targettattr="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.

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.com은 자사가 호스팅 서비스를 제공하는 두 회사인 Company333과 Company999에 대해 각각 디렉토리 어드민 관리자 역할을 작성했습니다. Example.com은 두 회사가 자사 데이터를 관리하고 액세스 제어 규칙을 구현하는 동시에 침입자로부터 데이터를 보호할 수 있기를 원합니다.

이런 이유로 Company333과 Company999는 다음과 같은 조건에 부합될 경우 디렉토리 트리의 해당 분기에 대한 전체 권한을 갖습니다.

- 인증서를 사용하여 SSL을 통해 연결이 인증된 경우
- 월요일부터 목요일까지 오전 8시에서 오후 6시 사이에 액세스가 요청된 경우
- 각 회사에 지정된 IP 주소로부터 액세스가 요청된 경우

이러한 조건에 대해서는 각 회사별 ACI(ACI "Company333" 및 ACI "Company999")에서 설명합니다. 두 ACI의 내용이 동일하므로 다음 예에서는 "Company333" ACI만 사용합니다.

ACI "Company333"

이전에 명시된 조건을 전제로 Company333에 디렉토리의 분기에 대한 전체 액세스 권한을 부여하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (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.

액세스 거부

접미어 중 많은 부분에 대한 액세스 권한이 이미 있는 경우 기존 ACI 아래에 있는 더 적은 접미어 부분에 대한 액세스를 거부할 수 있습니다.

주 – 액세스 거부는 의외의 복잡한 액세스 제어 동작을 유발할 수 있으므로 가능하면 피해야 합니다. 범위, 속성 목록, 대상 필터 등을 조합하여 액세스를 제한합니다.

또한 거부 액세스 ACI를 삭제하여 권한을 제거하는 대신, 다른 ACI에서 설정한 권한을 확장합니다.

디렉토리 서버에서는 액세스 권한을 평가할 때 `deny` 권한을 먼저 읽은 다음 `allow` 권한을 읽습니다.

다음 예에서 Example.com은 모든 가입자가 자신의 항목에서 연결 시간이나 계좌 잔고와 같은 결제 정보를 읽을 수 있도록 허용합니다. 또한 Example.com은 이 정보에 대한 쓰기 액세스를 명시적으로 거부합니다. 읽기 액세스에 대한 자세한 내용은 155 페이지 “ACI "Billing Info Read"”를 참조하십시오. `deny` 액세스에 대한 자세한 내용은 155 페이지 “ACI "Billing Info Deny"”를 참조하십시오.

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.

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; acl "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.

프록시 인증

프록시 인증 방법은 특별한 인증 형식입니다. 자신의 아이디를 사용하여 디렉토리에 바인드하는 사용자에게 프록시 인증을 통해 다른 사용자의 권한을 부여합니다.

프록시 요청을 허용하도록 디렉토리 서버를 구성하려면 다음을 수행해야 합니다.

- 관리자에게 프록시에 대한 다른 사용자 권한을 부여합니다.
- 일반 사용자에게 액세스 제어 정책에 정의된 일반 액세스 권한을 부여합니다.

주 – 디렉토리 관리자를 제외한 모든 디렉토리 사용자에게 프록시 권한을 부여할 수 있습니다. 디렉토리 관리자의 DN을 프록시 DN으로 사용할 수는 없습니다. 프록시 권한을 부여하면 디렉토리 관리자 DN을 제외한 모든 DN을 프록시 DN으로 지정할 수 있는 권한을 부여하는 것이므로 특히 주의해야 합니다. 디렉토리 서버가 동일한 작업에서 프록시 인증 제어를 둘 이상 수신하면 클라이언트 응용 프로그램에 오류가 반환되고 작업 시도는 실패하게 됩니다.

프록시 인증 예

Example.com은 `MoneyWizAcctSoftware`로 바인드하는 클라이언트 응용 프로그램이 LDAP 데이터에 대해 계정 관리자와 동일한 액세스 권한을 갖도록 허용합니다.

적용되는 매개 변수는 다음과 같습니다.

- 클라이언트 응용 프로그램의 바인드 DN은 `uid=MoneyWizAcctSoftware, ou=Applications,dc=example,dc=com`입니다.
- 클라이언트 응용 프로그램이 액세스를 요청하는 대상 하위 트리는 `ou=Accounting,dc=example,dc=com`입니다.
- `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 계정 하위 트리에 액세스하려면 다음과 같은 조건에 부합해야 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: (targetattr="*") (version 3.0; acl "allowAll-AcctAdmin"; allow
 (all) userdn="ldap:///uid=AcctAdministrator,ou=Administrators,
 dc=example,dc=com";)
```

- 클라이언트 응용 프로그램에 프록시 권한을 부여하는 아래 ACI가 디렉토리에 있어야 합니다.

```
aci: (targetattr="*") (version 3.0; acl "allowproxy- accountingsoftware";
 allow (proxy) userdn= "ldap:///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 ldapmodify와 같은 LDAP 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 - \
 -y "uid=AcctAdministrator,ou=Administrators,dc=example,dc=com" ...
```

클라이언트는 자신으로 바인드하지만 프록시 항목의 권한이 부여되며프록시 항목의 비밀번호를 제공할 필요가 없습니다.

필터링을 사용한 대상 설정

디렉토리에 분산된 여러 항목에 대한 액세스를 허용하는 액세스 제어를 설정하려면 필터를 사용하여 대상을 설정할 수 있습니다.

필터를 사용하여 HR의 모든 사용자에게 직원 항목에 대한 액세스를 허용하려면 LDIF로 아래 명령문을 작성합니다.

```
aci: (targetattr="*") (targetfilter=(objectClass=employee))
 (version 3.0; acl "HR access to employees";
 allow (all) groupdn= "ldap:///cn=HRgroup,ou=People,dc=example,dc=com";)
```

이 예에서는 ou=People,dc=example,dc=com 항목에 ACI를 추가한다고 가정합니다.

주 – 검색 필터는 액세스를 관리할 객체를 직접 지정하지 않으므로 잘못된 객체에 대한 액세스를 허용하거나 거부하지 않도록 합니다. 잘못된 객체에 대한 액세스를 실수로 허용하거나 거부할 경우 디렉토리가 더 복잡해질 위험이 있습니다. 또한 필터를 사용할 경우 디렉토리 내의 액세스 제어 문제점을 해결하기 어렵다는 단점이 있습니다.

쉼표가 있는 DN에 대한 권한 정의

쉼표가 있는 DN은 LDIF ACI 명령문에서 특수 처리해야 합니다. ACI 명령문의 대상 및 바인드 규칙 부분에서 백슬래시(\)를 사용하여 쉼표를 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.";)
```

유효 권한 보기

디렉토리 항목에 대한 액세스 정책을 유지관리할 때 정의한 ACI의 보안에 미치는 영향을 알고 있어야 합니다. 디렉토리 서버에서는 ACI에서 지정된 항목의 특정 사용자에게 부여하는 유효 권한을 보고 기존 ACI를 평가할 수 있습니다.

디렉토리 서버는 검색 작업에 포함할 수 있는 "유효 권한 보기" 컨트롤에 응답합니다. 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 임의 DN에 대한 권한을 요청할 수 있으므로 관리자는 디렉토리 사용자의 권한을 테스트할 수 있습니다.

유효 권한 기능은 LDAP 컨트롤을 사용합니다. 원격 서버에 바인드할 때 사용한 프록시 아이디도 유효 권한 속성에 액세스할 수 있어야 합니다.

유효 권한 보기 컨트롤에 대한 액세스 제한

유효 권한 보기 작업은 보호 및 적절한 제한이 필요한 디렉토리 작업입니다.

유효 권한 정보에 대한 액세스를 제한하려면 `getEffectiveRights` 속성의 기본 ACI를 수정합니다. 그런 다음 `getEffectiveRightsInfo` 속성의 새 ACI를 작성합니다.

예를 들어 다음 ACI는 디렉토리 어드민 관리자 그룹의 구성원에게만 유효 권한 보기를 허용합니다.

```
aci: (targetattr != "aci")(version 3.0; acl
 "getEffectiveRights"; allow(all) groupdn =
 "ldap:///cn=Directory Administrators,ou=Groups,dc=example,dc=com";)
```

유효 권한 정보를 보려면 유효 권한 컨트롤 사용을 위한 액세스 제어 권한 **및** `aclRights` 속성에 대한 읽기 권한이 있어야 합니다. 이러한 이중 액세스 제어 계층은 필요에 따라 세부 조정할 수 있는 기본 보안을 제공합니다. 프록시와 마찬가지로 항목에 `aclRights` 속성에 대한 읽기 권한이 있는 경우 해당 항목과 속성에 대한 모든 사용자의 권한 정보를 요청할 수 있습니다. 즉, 자원을 관리하는 사용자는 해당 자원에 대한 권한이 있는 사용자를 결정할 수 있지만 이 사용자가 권한이 있는 사용자를 실제로 관리하는 것은 아닙니다.

권한 정보를 요청하는 사용자에게 유효 권한 컨트롤을 사용할 권한이 없는 경우 작업이 실패하고 오류 메시지가 반환됩니다. 그러나 권한 정보를 요청하는 사용자에게 컨트롤 사용 권한이 있지만 `aclRights` 속성에 대한 읽기 권한이 없는 경우에는 `aclRights` 속성이 반환된 항목에 표시되지 않습니다. 이 동작은 디렉토리 서버의 일반 검색 동작을 반영합니다.

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:` *bind DN* `"` — 지정된 *DN*을 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` 옵션 중 한 개 또는 두 항목 모두를 사용할 경우 "유효 권한 보기" 컨트롤의 OID에 `-J` 옵션이 암묵적으로 지정되므로 별도로 지정할 필요가 없습니다. 유효 권한 컨트롤에 NULL 값을 지정하면 현재 사용자에 대한 권한이 검색됩니다. 또한 현재 `ldapsearch` 작업에서 반환되는 속성과 항목에 대한 권한이 검색됩니다.

  그런 다음 보려는 정보 유형을 선택해야 합니다. 간단한 권한 정보를 선택하거나 해당 권한이 부여 또는 거부되는 방법을 설명하는 자세한 로깅 정보를 선택합니다. 각각 `aclRights` 또는 `aclRightsInfo`를 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 속성처럼 동작합니다. 이러한 속성은 디렉토리에 저장되지 않고 명시적인 요청이 있는 경우에만 반환되며 디렉토리 서버에서 "유효 권한 보기" 컨트롤에 대한 응답으로 생성됩니다.

따라서 필터 또는 검색 작업에서는 이러한 속성을 사용할 수 없습니다.

유효 권한 기능은 액세스 제어에 영향을 주는 다른 매개 변수를 상속합니다. 이러한 매개 변수에는 시간, 인증 방법, 시스템 주소, 이름 등이 포함됩니다.

아래 예에서는 사용자 Carla Fuente가 디렉토리에서 자신의 권한을 볼 수 있는 방법을 보여줍니다. 결과에서 `1`은 권한이 부여된 것을 의미하고 `0`은 권한이 거부된 것을 의미합니다.

```
$ ldapsearch -J "1.3.6.1.4.1.42.2.27.9.5.2 -h host1.Example.com -p 389 \
 -D "uid=cfuente,ou=People,dc=example,dc=com" -w - -b "dc=example,dc=com" \
 "(objectclass=*)" aclRights
Enter bind password:
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 관리자는 Carla Fuente에게 읽기 권한이 없는 항목을 볼 수 있습니다.

```
$ ldapsearch -h host1.Example.com -p 389 -D cn=admin,cn=Administrators,cn=config -w - \
 -c "dn: uid=cfuente,ou=People,dc=example,dc=com" -b "dc=example,dc=com" \
 "(objectclass=*)" aclRights
Enter bind password:
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` 및 `manager` 속성을 수정할 수 없다는 것을 확인할 수 있습니다.

```
$ ldapsearch -h host1.Example.com -p 389 -D cn=admin,cn=Administrators,cn=config -w - \
 -c "dn: uid=cfuente,ou=People,dc=example,dc=com" -b "dc=example,dc=com" \
 "(uid=cfuente)" aclRights "*"
Enter bind password:
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
```

고급 액세스 제어: 매크로 ACI 사용

반복 디렉토리 트리 구조를 사용하는 조직에서는 매크로를 사용하여 디렉토리에 사용되는 ACI 수를 최적화할 수 있습니다. 디렉토리 트리에서 ACI 수를 줄이면 액세스 제어 정책을 관리하기가 쉽고 ACI 메모리 사용 효율성도 향상됩니다.

매크로는 ACI에서 DN 또는 DN의 일부를 나타내는 자리 표시자입니다. 매크로를 사용하여 ACI의 대상 부분, 바인드 규칙 부분 또는 두 부분에서 모두 DN을 나타낼 수 있습니다. 실제로 LDAP 작업이 수신되면 디렉토리 서버는 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.

이 절은 매크로 ACI의 예와 매크로 ACI 구문에 대한 정보로 구성되어 있습니다.

매크로 ACI 예

예를 사용하면 매크로 ACI의 혜택과 작동 방식을 명확히 이해할 수 있습니다. 그림 6–1에서는 매크로 ACI를 사용하여 전체 ACI 수를 효과적으로 줄일 수 있는 디렉토리 트리를 보여줍니다.

이 그림에서는 하위 도메인의 반복 패턴이 동일한 트리 구조(`ou=groups,ou=people`)를 갖습니다. Example.com 디렉토리 트리에는 그림에 표시되지 않은 두 접미어(`dc=hostedCompany2,dc=example,dc=com` 및 `dc=hostedCompany3,dc=example,dc=com`)가 저장되어 있기 때문에 이 패턴도 트리 전체에 걸쳐 반복됩니다.

디렉토리 트리의 ACI에도 반복 패턴이 있습니다. 예를 들어 `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";)
```

이 ACI는 `domainAdmins` 그룹에 `dc=hostedCompany1,dc=example,dc=com` 트리의 항목에 대한 읽기 및 검색 권한을 부여합니다.

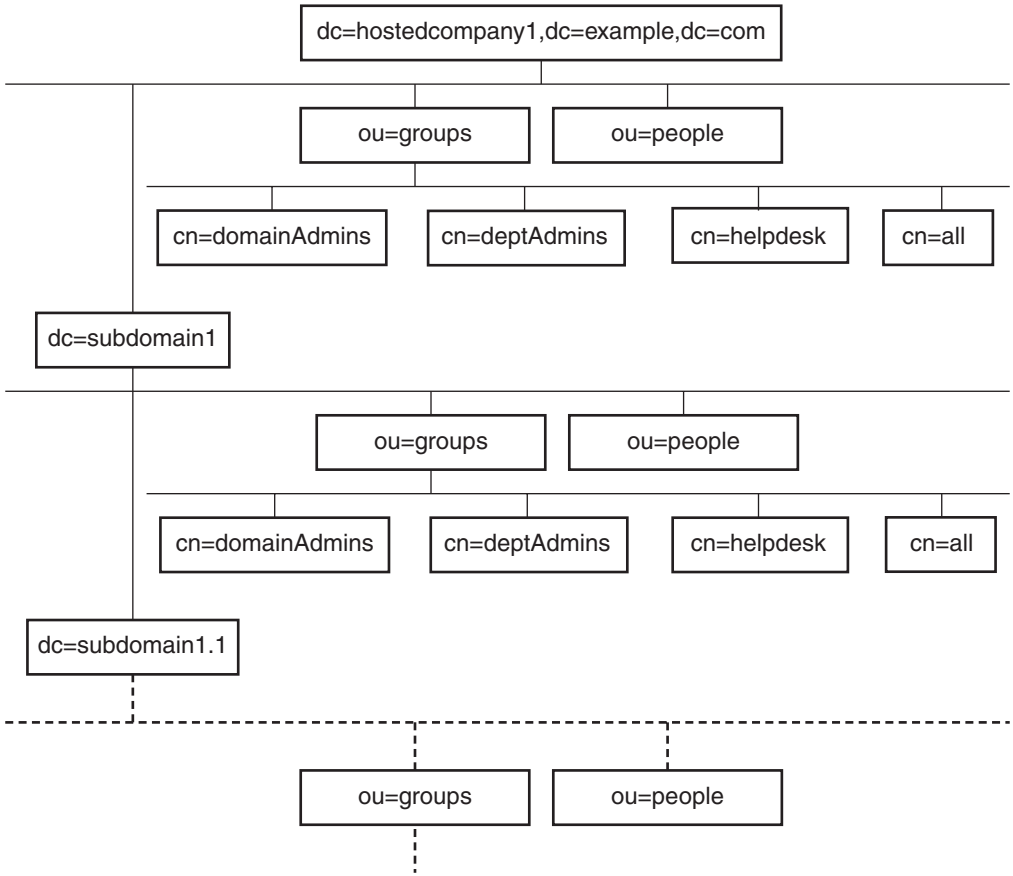


그림 6–1 매크로 ACI의 디렉토리 트리 예

`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";)
```

앞의 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 매크로 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 이전에 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 및 userattr 등 바인드 자격 증명을 제공하는 데 사용되는 ACI 키워드를 총체적으로 ACI의 **주제**라고 부릅니다. 주제는 ACI의 적용 대상을 결정합니다.

아래 표는 특정 ACI 키워드를 대체하는 데 사용할 수 있는 매크로를 보여줍니다.

표 6–1 매크로 ACI 키워드

매크로	설명	ACI 키워드
($dn)	대상 일치 및 주제의 직접 대체에 사용합니다.	target, targetfilter, userdn, roledn, groupdn, userattr
[$dn]	주제의 하위 트리에서 작동하는 여러 RDN을 대체하는 데 사용합니다.	targetfilter, userdn, roledn, groupdn, userattr
($attr.*attrName*)	대상 항목의 *attributeName* 속성 값을 주제로 대체하는 데 사용합니다.	userdn, roledn, groupdn, userattr

매크로 ACI 키워드에 적용되는 제한 사항은 다음과 같습니다.

- 주제에서 ($dn) 및 [$dn] 매크로를 사용할 경우 ($dn) 매크로가 포함된 대상을 **반드시** 정의해야 합니다.
- 주제에서 ($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 이 항목에서 대상으로 하는 LDAP 요청이 있습니다.

```
cn=all,ou=groups,dc=subdomain1, dc=hostedCompany1,dc=example,dc=com
```

또한, 다음과 같이 대상을 정의하는 ACI가 있습니다.

```
(target="ldap:///ou=Groups,($dn),dc=example,dc=com")
```

($dn) 매크로는 "dc=subdomain1, dc=hostedCompany1"과 일치하므로 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"
```

주제는 다음과 같습니다.

```
groupdn="ldap:///cn=DomainAdmins,ou=Groups,
 dc=subdomain1,dc=hostedCompany1,dc=example,dc=com"
```

매크로가 확장되면 디렉토리 서버는 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. 자식 DN이 대상이면 대체를 통해 주제 문자열에 유효한 DN을 작성할 수 없기 때문입니다.

주제의 [$dn] 대체

[$dn]의 대체 메커니즘은 ($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 있다고 가정합니다.

```
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. 서버는 대상의 ($dn)이 dc=subdomain1,dc=hostedCompany1과 일치하는지 확인합니다.
2. 주제의 [$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. 바인드 DN이 그룹의 구성원이 아니면 일치한 값의 마지막 RDN에서 매크로 확장이 중단되고 이 ACI에 대한 평가가 완료됩니다.

[$dn] 매크로를 사용할 경우 도메인 수준 관리자에게 디렉토리 트리의 **모든** 하위 도메인에 대한 액세스 권한을 유연성 있게 부여할 수 있다는 장점이 있습니다. 따라서 [$dn] 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";)
```

ACI는 cn=DomainAdmins,ou=Groups, dc=hostedCompany1,dc=example,dc=com 구성원에게 dc=hostedCompany1 아래의 모든 하위 도메인에 대한 액세스 권한을 부여합니다. 따라서 해당 그룹에 속하는 관리자는 ou=people,dc=subdomain1.1,dc=subdomain1과 같은 하위 트리에 액세스할 수 있습니다.

하지만 이와 동시에 cn=DomainAdmins,ou=Groups, dc=subdomain1.1의 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 가정합니다.

```
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 읽습니다. 그런 다음 주제에서 해당 값을 대체하여 매크로를 확장합니다. 이 예에서 roledn은 다음과 같이 확장됩니다.

```
roledn = "ldap:///cn=DomainAdmins,ou=Sales,dc=HostedCompany1,dc=example,dc=com"
```

그런 다음 디렉토리 서버는 일반 ACI 평가 알고리즘에 따라 ACI를 평가합니다.

매크로에 지정된 속성이 여러 값을 갖는 경우 각 값을 차례로 사용하여 매크로를 확장합니다. 처음 일치하는 항목을 제공한 값이 사용됩니다.

액세스 제어 로깅 정보

오류 로그에서 액세스 제어에 대한 정보를 얻으려면 적절한 로그 수준을 설정해야 합니다.

▼ ACI에 대한 로깅을 설정하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **ACI 처리에 사용할 로그 수준을 설정합니다.**

```
$ dsconf set-log-prop -h host -p port error level:err-acl
```

TCP 래핑을 통한 클라이언트-호스트 액세스 제어

TCP 래퍼를 사용하여 TCP 수준에서 연결이 허용되거나 거부된 호스트 또는 IP 주소를 제어할 수 있습니다. TCP 래핑을 통해 클라이언트-호스트 액세스를 제한할 수 있습니다. 이렇게 하면 디렉토리 서버에 대한 초기 TCP 연결을 호스트와 독립적으로 보호할 수 있습니다.

디렉토리 서버에 대한 TCP 래핑을 설정할 수는 있지만 TCP 래핑은 특히 서비스 거부(DoS) 공격 중에 상당한 성능 저하를 초래할 수 있습니다. 디렉토리 서버 외부에서 유지 관리되는 호스트 기반 방화벽이나 IP 포트 필터링을 사용하여 최상의 성능을 얻을 수 있습니다.

▼ TCP 래핑을 사용 가능하게 하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1. **인스턴스 경로 내의 다른 위치에서 `hosts.allow` 파일이나 `hosts.deny` 파일을 작성합니다.**

   예를 들어 *instance-path*`/config`에 파일을 작성합니다. 작성하는 파일의 서식이 `hosts_access`(4)를 준수해야 합니다.

2. **경로를 액세스 파일로 설정합니다.**

```
$ dsconf set-server-prop -h host -p port host-access-dir-path:path-to-file
```

   예를 들면 다음과 같습니다.

```
$ dsconf set-server-prop -h host -p port host-access-dir-path:/local/ds1/config
"host-access-dir-path" property has been set to "/local/ds1/config".
The "/local/ds1/config" directory on host1 must contain valid hosts.allow
and/or hosts.deny files.
Directory Server must be restarted for changes to take effect.
```

▼ TCP 래핑을 사용 불가능하게 하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **호스트 액세스 경로를 ""로 설정합니다.**

```
$ dsconf set-server-prop -h host -p port host-access-dir-path:""
```

◆ ◆ ◆ 7 장

7 디렉토리 서버 비밀번호 정책

디렉토리 서버에 연결하면 사용자가 인증됩니다. 디렉토리는 인증 중에 설정된 아이디에 따라 사용자에게 액세스 권한 및 자원 제한을 부여할 수 있습니다. 이 장에서 **계정**은 사용자 항목을 포괄적으로 나타냅니다. 또한 계정은 사용자가 디렉토리에서 작업을 수행할 수 있는 권한을 반영합니다. 비밀번호 정책 설명에서 모든 계정은 사용자 항목 및 비밀번호와 연관됩니다.

또한 이 장에서는 비밀번호 정책의 일환인 계정 활성화에 대해서도 설명합니다. 디렉토리 어드민 관리자는 비밀번호 정책과 상관없이 계정을 직접 잠그거나 잠금 해제할 수 있습니다.

인증 방법에 대해서는 이 장에서 설명하지 않습니다. SASL GSSAPI, 클라이언트 SSL 인증서 기반 인증 등과 같은 일부 인증 방법에서는 비밀번호를 사용하지 않습니다. 이 장에서 설명하는 비밀번호 정책 관련 정보는 해당 인증 방법에는 적용되지 않습니다. 인증 메커니즘을 구성하는 방법에 대한 자세한 내용은 5 장을 참조하십시오.

이 장에서는 Directory Server 6.2 및 이전 디렉토리 서버 버전 간에 비밀번호 정책의 호환성을 다루지 않습니다. Directory Server 6.2 인스턴스를 만들면 비밀번호 정책 구현이 이전 버전으로부터 업그레이드할 수 있도록 기본적으로 Directory Server 5 호환 모드로 설정됩니다. 이 장에서 설명된 비밀번호 정책 기능을 최대한 활용하기 위해 비밀번호 정책 호환성 모드를 변경해야 할 수 있습니다. 비밀번호 호환성 모드 설정에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Migration Guide** 의 “Password Policy Compatibility”를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

비밀번호 정책 및 워크시트

이 절에서는 비밀번호 정책 설정에 대해 설명하고 사용자 요구 사항에 맞는 비밀번호 정책을 정의하는 데 도움이 되는 워크시트를 제공합니다.

주 – 기본 비밀번호 정책을 사용하려면 177 페이지 "기본 비밀번호 정책 관리"를 참조하십시오.

비밀번호 정책 설정

디렉토리 서버에서 비밀번호 정책을 지정하는 경우 `pwdPolicy`(5dsoc) 객체 클래스를 포함하는 항목을 만들거나 수정합니다.

특정 유형의 사용자에 대한 비밀번호 정책을 정의할 경우 다음 사항을 고려해야 합니다.

- 침입자가 비밀번호를 해독하려고 시도할 때 계정을 잠그는 방법
  자세한 내용은 172 페이지 "계정 잠금 정책"을 참조하십시오.
- 비밀번호를 변경하는 방법
  자세한 내용은 173 페이지 "비밀번호 변경 정책"을 참조하십시오.
- 허용되는 비밀번호 값
  자세한 내용은 174 페이지 "비밀번호 내용 정책"을 참조하십시오.
- 비밀번호 만료 처리 방법
  자세한 내용은 175 페이지 "비밀번호 만료 정책"을 참조하십시오.
- 서버에서 성공한 마지막 인증 시간을 기록할지 여부
  175 페이지 "마지막 인증 시간 추적 정책"을 참조하십시오.

이 장의 이후에 나오는 절에서는 이러한 비밀번호 정책 영역을 처리하는 방법에 대해 설명합니다. 176 페이지 "비밀번호 정책 정의 워크시트"를 사용하여 구현할 각 비밀번호 정책을 명확하게 하십시오.

계정 잠금 정책

이 절에서는 계정 잠금을 제어하는 정책 속성에 대해 설명합니다.

디렉토리 서버 계정은 사용자가 디렉토리에서 작업을 수행하는 데 필요한 권한과 사용자 항목을 포괄적으로 나타냅니다. 각 계정은 바인드 DN 및 사용자 비밀번호와 연결되어 있습니다. 침입자가 비밀번호를 해독하려고 시도할 경우 디렉토리 서버에서 계정을 잠그도록 할 수 있습니다. 계정을 잠그면 침입자가 해당 계정을 사용하여 바인드할 수 없습니다. 또한 침입자가 공격을 계속할 수 없습니다.

관리자는 역할을 공유하는 모든 사용자 계정 또는 비활성 계정을 수동으로 렌더링할 수도 있습니다. 자세한 내용은 194 페이지 “수동으로 계정 잠금”을 참조하십시오. 비밀번호 정책의 핵심은 디렉토리 서버에서 사용자의 개입 **없이** 계정을 잠그는 환경을 만드는 것입니다.

먼저, 실패한 바인드가 너무 많이 발생할 경우 디렉토리 서버에서 pwdLockout(5dsat)를 사용하여 계정을 자동으로 잠그도록 지정해야 합니다. 디렉토리 서버는 계정에 대해 연속적으로 실패하는 바인드 시도를 추적합니다. pwdMaxFailure(5dsat)를 사용하여 디렉토리 서버에서 계정을 잠그기 전에 허용되는 연속 실패 횟수를 지정할 수 있습니다.

디렉토리 서버에서는 비밀번호 정책을 엄격하게 준수하여 계정을 잠급니다. 작업은 완전히 기계식입니다. 계정은 침입자가 계정에 대한 공격을 마운트하고 있기 때문이 아니라 사용자가 비밀번호를 잘못 입력했기 때문에 잠길 수 있습니다. 따라서 pwdFailureCountInterval(5dsat)를 사용하여 디렉토리 서버에서 실패한 시도의 레코드를 정리하기 전에 대기하는 시도 횟수를 지정할 수 있습니다. pwdLockoutDuration(5dsat)를 사용하여 디렉토리 서버에서 계정 잠금을 자동으로 해제하기 전의 잠금 지속 시간을 지정합니다. 관리자는 악의적인 의도 없이 법률에 위반되지 않는 실수를 한 사용자의 계정 잠금을 해제하기 위해 개입할 필요가 없습니다.

사용자 데이터가 복제 토폴로지를 통해 복제되는 경우 잠금 속성이 다른 항목 데이터와 함께 복제됩니다. pwdIsLockoutPrioritized(5dsat) 속성의 기본 설정은 TRUE이므로 잠금 속성에 대한 업데이트는 더 높은 우선 순위로 복제됩니다. 따라서 모든 계정 복제본이 잠기기 전에 pwdMaxFailure 연속 실패 바인드 시도 수가 제한되며, 이는 기타 복제본이 잠기기 전의 시도 수보다 적습니다. 사용자가 정확히 pwdMaxFailure번의 시도 후에 전체 복제된 토폴로지에서 잠기는지 확인하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 “Preventing Authentication by Using Global Account Lockout”을 참조하십시오.

비밀번호 변경 정책

이 절에서는 비밀번호 변경을 제어하는 정책 속성에 대해 설명합니다.

다양한 배포에서 디렉토리 서버는 아이디 데이터를 위한 저장소 역할을 합니다. 사용자는 pwdAllowUserChange(5dsat)에 지정된 대로 자신의 비밀번호를 변경할 수 있어야 하므로 사용자가 비밀번호를 변경할 필요가 없습니다.

사용자에게 자신의 비밀번호를 변경할 수 있도록 허용한 후 사용자가 비밀번호를 변경할 수 있는 경우를 제어할 수도 있습니다. pwdSafeModify(5dsat)를 사용하면 비밀번호를 변경할 사용자가 비밀번호를 바꾸려면 기존 비밀번호를 정확하게 입력해야 하도록 지정할 수 있습니다. 비밀번호를 수정하는 방법에 대한 예는 189 페이지 “pwdSafeModify가 TRUE인 경우 명령줄에서 비밀번호 수정”을 참조하십시오. pwdInHistory(5dsat)를 사용하여 디렉토리 서버에서 기억하는 비밀번호 수를 지정함으로써 사용자가 비밀번호를 다시 사용하지 못하도록 할 수 있습니다. 또한 pwdMinAge(5dsat)를 설정하여 사용자가 비밀번호를 너무 자주 변경하지 못하도록 할 수 있습니다.

대부분의 경우 관리자나 관리자가 관리하는 응용 프로그램에서 디렉토리에 사용자 항목을 만듭니다. 관리자는 사용자 비밀번호 값을 할당할 수 있으며, 이 비밀번호는 사용자가 새 계정에 처음으로 바인드할 때 변경합니다. 사용자 비밀번호를 재설정해야 할 수도 있습니다. 이 경우 다음에 사용자가 계정을 사용할 때 해당 비밀번호를 변경해야 합니다. 디렉토리 서버에는 다른 사용자가 비밀번호 값을 재설정한 후 사용자가 비밀번호를 변경해야 하는지 여부를 나타내는 데 사용할 수 있는 pwdMustChange(5dsat)라는 특수 속성이 있습니다.

passwordRootdnMayBypassModsChecks(5dsat)를 설정하여 디렉토리 어드민 관리자는 비밀번호를 변경할 때 정책의 적용을 받지 않도록 지정할 수도 있습니다.

비밀번호 내용 정책

이 절에서는 비밀번호 내용을 제어하는 정책 속성에 대해 설명합니다.

일반적으로 비밀번호 값은 디렉토리 검색에서 반환되지 않지만 공격자는 디렉토리 데이터베이스에 대한 액세스 권한을 얻을 수 있습니다. 따라서 비밀번호 값은 대부분 passwordStorageScheme(5dsat)를 사용하여 지정한 지원되는 해시 형식 중 하나로 저장됩니다.

또한 pwdCheckQuality(5dsat)를 설정하여 비밀번호가 최소 비밀번호 품질 정의를 충족하는지 확인하도록 할 수 있습니다. 그런 다음 서버는 비밀번호가 cn, givenName, mail, ou, sn 또는 uid 속성 값과 일치하지 않는지 확인합니다. 이러한 속성이 포함된 비밀번호의 비교는 대소문자를 구분하지 않습니다.

추가 검사는 pwdCheckQuality(5dsat) 집합에서 사용할 수 있습니다. pwdMinLength(5dsat)를 설정하여 비밀번호가 지정된 문자 수 이상이 되도록 할 수 있습니다. 또한, 강력한 비밀번호 확인 플러그인이 활성화되어 있는 경우 디렉토리 서버는 비밀번호에 플러그인에 사용되는 사전 파일의 문자열이 없는지를 확인합니다. 또한 비밀번호에 서로 다른 유형의 문자 조합이 있는지 확인합니다.

dsconf set-server-prop 명령을 사용하여 강력한 비밀번호 확인을 활성화할 수 있습니다. pwd-strong-check-enabled 등록 정보를 사용하여 플러그인을 설정한 다음 서버를 다시 시작하여 변경 사항을 적용합니다. pwd-strong-check-require-charset 등록 정보를 사용하여 비밀번호에 필요한 문자 집합을 지정합니다. pwd-strong-check-require-charset 등록 정보는 다음과 같은 값 마스크를 사용합니다.

lower	새 비밀번호에 소문자가 포함되어야 합니다.
upper	새 비밀번호에 대문자가 포함되어야 합니다.
digit	새 비밀번호에 숫자가 포함되어야 합니다.
special	새 비밀번호에 특수 문자가 포함되어야 합니다.
any-two	새 비밀번호에 위에 명시한 문자 집합 중 두 가지 이상에 해당하는 문자가 각각 하나 이상씩 포함되어야 합니다.
any-three	새 비밀번호에 위에 명시한 문자 집합 중 세 가지 이상에 해당하는 문자가 각각 하나 이상씩 포함되어야 합니다.

pwd-strong-check-require-charset 등록 정보의 기본 설정은 lower && upper && digit && special입니다.

비밀번호 만료 정책

이 절에서는 비밀번호 만료를 제어하는 정책 속성에 대해 설명합니다.

사용자가 비밀번호를 정기적으로 변경하도록 pwdMaxAge(5dsat)를 설정하여 비밀번호가 특정 사용 기간에 도달한 경우 디렉토리 서버에서 비밀번호를 만료시키도록 구성할 수 있습니다.

이 경우 비밀번호가 만료될 예정이라는 메시지를 사용자에게 보내야 합니다. 바인드에 사용된 비밀번호가 만료될 예정이라는 경로 메시지를 반환하도록 디렉토리 서버를 구성할 수 있습니다. pwdExpireWarning(5dsat)를 사용하여 클라이언트가 바인드할 때 만료되기 며칠 전에 경고 메시지를 표시할지를 정의합니다. **클라이언트 응용 프로그램에 경고 메시지가 표시됩니다.사용자에게 직접 경고 메시지가 표시되지는 않습니다.**클라이언트 응용 프로그램은 비밀번호가 만료될 예정이라는 경고 메시지를 받을 경우 최종 사용자에게 알려야 합니다.

pwdGraceAuthNLimit(5dsat)를 설정하여 사용자가 만료된 비밀번호를 사용하여 한 번 이상 바인드를 시도하도록 허용할 수 있습니다. 따라서 사용자가 기간 내에 비밀번호를 변경하지 못한 경우에도 바인드하여 비밀번호를 변경할 수 있습니다. 사용자는 정상 로그인을 사용하여 바인드한 후 모든 작업을 수행할 수 있습니다. 정상 로그인은 비밀번호가 만료되지 않은 것처럼 작동합니다.

디렉토리 서버는 항목의 비밀번호가 수정될 때마다 pwdChangedTime(5dsat) 작동 가능 속성을 업데이트합니다. 따라서 비밀번호 만료가 활성화될 때까지 기다린 후 비밀번호 만료를 활성화하면 이미 만료된 사용자 비밀번호가 즉시 만료됩니다. 이 동작을 원하지 않으면 경고 및 정상 로그인을 사용하십시오.

마지막 인증 시간 추적 정책

이 절에서는 pwdKeepLastAuthTime(5dsat) 비밀번호 정책 속성의 사용에 대해 설명합니다.

pwdKeepLastAuthTime을 설정하면 디렉토리 서버는 사용자가 인증될 때마다 성공한 마지막 바인드 시간을 추적합니다. 시간은 사용자 항목의 pwdLastAuthTime(5dsat) 작동 가능 속성에 기록됩니다.

이 동작은 성공한 바인드 작업마다 업데이트를 추가하기 때문에 pwdKeepLastAuthTime 기능은 기본적으로 활성화되지 않습니다. 이 기능을 배포에 사용하려면 해당 기능을 명시적으로 설정해야 합니다.

비밀번호 정책 정의 워크시트

이 워크시트를 사용하면 명령줄 인터페이스 또는 디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)를 사용하여 구현할 비밀번호 정책을 쉽게 정의할 수 있습니다. 비밀번호 정책별로 하나의 워크시트를 사용합니다.

비밀번호 정책 항목의 DN을 기록한 후 각 정책 영역에 속성 설정에 대한 결정 사항을 기록합니다. 또한 해당 설정에 대한 이론적 근거를 기록합니다.

비밀번호 정책 워크시트

비밀번호 정책 항목 고유 이름
dn: cn=

정책 영역	**속성**	**여기에 설정 기록**	**여기에 설정에 대한 이론적 근거 기록**
계정 잠금	pwdFailureCountInterval(5dsat)		
	pwdIsLockoutPrioritized(5dsat)		
	pwdLockout(5dsat)		
	pwdLockoutDuration(5dsat)		
	pwdMaxFailure(5dsat)		
비밀번호 변경	passwordRootdnMayBypassModsChecks(5dsat)		
	pwdAllowUserChange(5dsat)		
	pwdInHistory(5dsat)		
	pwdMinAge(5dsat)		
	pwdMustChange(5dsat)		
	pwdSafeModify(5dsat)		
비밀번호 내용	passwordStorageScheme(5dsat)		
	pwdCheckQuality(5dsat)		
	pwdMinLength(5dsat)		
비밀번호 만료	pwdExpireWarning(5dsat)		
	pwdGraceAuthNLimit(5dsat)		
	pwdMaxAge(5dsat)		

정책 영역	속성	여기에 설정 기록	여기에 설정에 대한 이론적 근거 기록
마지막 인증 시간 추적	pwdKeepLastAuthTime(5dsat)		

주 – pwdCheckQuality 속성이 2로 설정되어 있는 경우 서버는 추가 검사를 수행할 수 있습니다. 또한 비밀번호 확인 플러그인이 활성화되어 있는 경우 플러그인 설정은 새 비밀번호 값에 대해 수행하는 검사 유형에 영향을 미칩니다.

기본 비밀번호 정책 관리

기본 비밀번호 정책은 특별 정책이 정의되어 있지 않은 디렉토리 인스턴스 내의 모든 사용자에게 적용됩니다. 기본 비밀번호 정책은 디렉토리 관리자에게는 적용되지 않습니다. 정책 범위에 대한 자세한 내용은 180 페이지 “적용되는 비밀번호 정책”을 참조하십시오.

기본 비밀번호 정책은 dsconf 명령을 사용하여 구성할 수 있는 정책입니다. 또한 cn=Password Policy,cn=config를 실행하여 기본 비밀번호 정책을 볼 수도 있습니다.

이 절에서는 각 정책 영역에 대한 정책 속성과 관련 dsconf 서버 등록 정보를 보여줍니다. 또한 기본 비밀번호 정책 설정을 보거나 변경하는 방법에 대해 설명합니다.

비밀번호 정책 속성과 dsconf 서버 등록 정보 간의 상관 관계

다음 표에서는 각 비밀번호 정책 영역에 대한 비밀번호 정책 속성과 관련 dsconf 서버 등록 정보를 보여줍니다.

정책 영역	정책 속성	dsconf 서버 등록 정보
계정 잠금	pwdFailureCountInterval	pwd-failure-count-interval
	pwdLockout	pwd-lockout-enabled
	pwdLockoutDuration	pwd-lockout-duration
	pwdMaxFailure	pwd-max-failure-count

정책 영역	정책 속성	dsconf 서버 등록 정보
비밀번호 변경	passwordRootdnMayBypassModsChecks	pwd-root-dn-bypass-enabled
	pwdAllowUserChange	pwd-user-change-enabled
	pwdInHistory	pwd-max-history-count
	pwdMinAge	pwd-min-age
	pwdMustChange	pwd-must-change-enabled
	pwdSafeModify	pwd-safe-modify-enabled
비밀번호 내용	pwdCheckQuality	pwd-check-enabled, pwd-accept-hashed-password-enabled, pwd-strong-check-dictionary-path, pwd-strong-check-enabled, pwd-strong-check-require-charset
	pwdMinLength	pwd-min-length
	passwordStorageScheme	pwd-storage-scheme
비밀번호 만료	pwdExpireWarning	pwd-expire-warning-delay
	pwdGraceAuthNLimit	pwd-grace-login-limit
	pwdMaxAge	pwd-max-age
마지막 인증 시간 추적	pwdKeepLastAuthTime	pwd-keep-last-auth-time-enabled

주 – pwdCheckQuality와 상호 연관되는 등록 정보가 비밀번호 확인 플러그인을 구성합니다. 따라서 다섯 개의 등록 정보가 전체 서버 인스턴스에 적용됩니다. 다섯 개의 등록 정보는 다른 비밀번호 정책 pwdCheckQuality: 2에도 적용됩니다.

▼ 기본 비밀번호 정책 설정을 보는 방법

dsconf 명령을 사용하여 기본 비밀번호 정책 설정을 볼 수 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **기본 비밀번호 정책 구성을 읽습니다.**

```
$ dsconf get-server-prop -h host -p port | grep ^pwd-
pwd-accept-hashed-pwd-enabled      :  N/A
pwd-check-enabled                  :  off
pwd-compat-mode                    :  DS5-compatible-mode
pwd-expire-no-warning-enabled      :  on
```

```
pwd-expire-warning-delay           :  1d
pwd-failure-count-interval         :  10m
pwd-grace-login-limit              :  disabled
pwd-keep-last-auth-time-enabled    :  off
pwd-lockout-duration               :  1h
pwd-lockout-enabled                :  off
pwd-lockout-repl-priority-enabled  :  on
pwd-max-age                        :  disabled
pwd-max-failure-count              :  3
pwd-max-history-count              :  disabled
pwd-min-age                        :  disabled
pwd-min-length                     :  6
pwd-mod-gen-length                 :  6
pwd-must-change-enabled            :  off
pwd-root-dn-bypass-enabled         :  off
pwd-safe-modify-enabled            :  off
pwd-storage-scheme                 :  SSHA
pwd-strong-check-dictionary-path   :  /local/ds6/plugins/words-english-big.txt
pwd-strong-check-enabled           :  off
pwd-strong-check-require-charset   :  lower
pwd-strong-check-require-charset   :  upper
pwd-strong-check-require-charset   :  digit
pwd-strong-check-require-charset   :  special
pwd-supported-storage-scheme       :  CRYPT
pwd-supported-storage-scheme       :  SHA
pwd-supported-storage-scheme       :  SSHA
pwd-supported-storage-scheme       :  NS-MTA-MD5
pwd-supported-storage-scheme       :  CLEAR
pwd-user-change-enabled            :  on
```

▼ 기본 비밀번호 정책 설정을 변경하는 방법

`dsconf` 명령으로 서버 등록 정보를 설정하여 기본 비밀번호 정책을 변경할 수 있습니다.

주 – 이 절차를 완료하기 전에 176 페이지 “비밀번호 정책 정의 워크시트”를 읽고 완료하십시오.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 **워크시트 설정을 `dsconf` 명령 등록 정보 설정으로 변환합니다.**

2 **dsconf set-server-prop 명령을 사용하여 기본 비밀번호 정책 등록 정보를 적절하게 변경합니다.**

예를 들어 디렉토리 관리자는 다음 명령을 사용하여 비밀번호를 수정할 때 기본 정책을 위반할 수 있습니다.

```
$ dsconf set-server-prop -h host -p port pwd-root-dn-bypass-enabled:on
```

특수 비밀번호 정책 관리

특수 비밀번호 정책은 pwdPolicy(5dsoc) 항목에 정의됩니다. 정책은 디렉토리 트리 내의 임의의 위치(일반적으로 정책이 제어하는 계정으로 복제되는 하위 트리 내)에 정의될 수 있습니다. 정책은 cn=*policy name*,*subtree* 형태의 DN이 있습니다.

비밀번호 정책을 정의한 후 원하는 사용자 항목에서 pwdPolicySubentry(5dsat) 속성을 설정하여 비밀번호 정책을 할당합니다.

이 절은 다음 내용으로 구성되어 있습니다.

적용되는 비밀번호 정책

디렉토리 서버에서는 여러 비밀번호 정책을 구성할 수 있습니다. 이 절에서는 기본 비밀번호 정책과 특수 비밀번호 정책에 대해 설명합니다. 또한 지정된 계정에 여러 비밀번호 정책을 적용할 수 있는 경우 적용되는 정책에 대해 설명합니다.

디렉토리 서버 인스턴스를 처음 만들면 해당 인스턴스에 기본 비밀번호 정책이 적용됩니다. 기본 비밀번호 정책은 cn=PasswordPolicy,cn=config 구성 항목에 설명되어 있습니다. 기본 비밀번호 정책은 디렉토리 관리자를 제외한 디렉토리 내의 모든 계정에 적용됩니다.

모든 디렉토리 서버 비밀번호 정책과 마찬가지로 cn=PasswordPolicy,cn=config에는 pwdPolicy(5dsoc) 객체 클래스 및 sunPwdPolicy(5dsoc) 객체 클래스가 있습니다.

주 – 디렉토리 서버 인스턴스를 만들면 비밀번호 정책 속성이 이전 버전으로부터 업그레이드할 수 있도록 Directory Server 5 호환 모드 그대로 유지됩니다. 또한 Directory Server 5 호환 모드에서 디렉토리 서버는 `passwordPolicy`(5dsoc) 객체 클래스가 있는 비밀번호 정책 항목을 처리합니다.

업그레이드가 완료되면 **Sun Java System Directory Server Enterprise Edition 6.2 Migration Guide**에 설명된 것처럼 정식 모드에서 새 비밀번호 정책을 사용합니다. 관리 모드는 디렉토리 응용 프로그램에 투명하게 처리됩니다.

이 장에서는 새 비밀번호 정책 기능을 사용하는 비밀번호 정책 구성에 대해 설명합니다.

기본 비밀번호 정책을 변경하여 기본 설정을 무시할 수 있습니다. `dsconf`(1M) 명령을 사용하여 기본 비밀번호 정책에 대한 서버 등록 정보를 설정할 수 있습니다. 해당 서버 등록 정보의 이름은 일반적으로 `pwd-` 접두어로 시작합니다. 해당 등록 정보의 설정을 변경하면 인스턴스에 대한 기본 비밀번호 정책이 무시됩니다. 그러나 복제를 수행해도 복제본에 대한 변경 사항이 복사되지는 않습니다. 기본 비밀번호 정책에 대한 변경 사항은 디렉토리 데이터가 아닌 인스턴스 구성의 일부입니다.

기본 비밀번호 정책 구성 이외에 **특수 비밀번호 정책**을 구성할 수도 있습니다. 특수 비밀번호 정책은 디렉토리 트리의 항목에 정의됩니다. 특수 비밀번호 정책 항목은 기본 비밀번호 정책과 동일한 객체 클래스인 `pwdPolicy`(5dsoc)가 있으므로 정책 속성은 동일합니다. 특수 비밀번호 정책은 일반 디렉토리 항목이므로 정책 항목은 일반 디렉토리 항목과 동일한 방법으로 복제됩니다.

사용자 항목은 `pwdPolicySubentry`(5dsat) 작동 가능 속성 값을 통해 특수 비밀번호 정책을 참조합니다. 사용자 항목에서 참조하는 특수 비밀번호 정책은 인스턴스에 대한 기본 비밀번호 정책을 무시합니다. 사용자는 다양한 배포에서 사용자 역할을 할당합니다. `pwdPolicySubentry` 값을 설정하여 서비스 클래스(CoS)를 통해 사용자 계정에 적용되는 비밀번호 정책을 결정하도록 역할을 구성할 수 있습니다. 특정 역할에서 설정된 비밀번호 정책을 무시하려면 해당 사용자 항목에서 `pwdPolicySubentry` 값을 직접 변경합니다.

이 절을 요약하면 처음에는 기본 비밀번호 정책이 적용됩니다. 기본 비밀번호 정책을 변경하여 기본값을 무시할 수 있습니다. 그런 다음 특수 비밀번호 정책 항목을 만들어 기본 비밀번호 정책을 무시할 수 있습니다. 역할 및 CoS를 사용하여 비밀번호 정책을 할당할 경우 개별 항목에 대한 비밀번호 정책을 지정함으로써 CoS에서 할당된 정책을 무시할 수 있습니다.

▼ 비밀번호 정책을 만드는 방법

다른 디렉토리 항목의 경우와 동일한 방법으로 특수 비밀번호 정책을 만들고 수정합니다. 텍스트 편집기를 사용하여 LDIF에 비밀번호 정책 항목을 쓰는 절차는 다음과 같습니다. 그런 다음 `ldapmodify` 명령을 `-a` 옵션과 함께 사용하여 디렉토리에 비밀번호 정책 항목을 추가합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

시작하기 전에 별도로 명시하지 않는 한 여기에 표시된 데이터 예는 `Example.ldif`에서 가져온 것입니다.

1 만들려는 정책에 해당하는 비밀번호 정책 워크시트를 완료하십시오.

샘플은 176 페이지 “비밀번호 정책 정의 워크시트”를 참조하십시오.

2 워크시트를 기반으로 LDIF에 비밀번호 정책 항목을 작성합니다.

예를 들어 다음 정책 항목은 하위 트리 루트가 `dc=example,dc=com`인 Example.com의 임시 직원에 대한 비밀번호 정책을 지정합니다.

```
dn: cn=TempPolicy,dc=example,dc=com
objectClass: top
objectClass: pwdPolicy
objectClass: sunPwdPolicy
objectClass: LDAPsubentry
cn: TempPolicy
pwdAttribute: userPassword
pwdCheckQuality: 2
pwdLockout: TRUE
pwdLockoutDuration: 300
pwdMaxFailure: 3
pwdMustChange: TRUE
```

기본 비밀번호 정책 설정 이외에 여기에 표시된 정책은 추가 동작을 지정합니다. 비밀번호 품질 검사가 시행됩니다. 연속 세 번의 바인드 실패 후 5분(300초) 동안 계정을 잠급니다. 비밀번호를 재설정하고 나면 비밀번호를 변경해야 합니다. 정책을 사용자 계정에 할당하면 여기에 **명시적으로** 지정된 설정은 기본 비밀번호 정책을 무시합니다.

3 디렉토리에 비밀번호 정책 항목을 추가합니다.

예를 들어 다음 명령은 `dc=example,dc=com` 아래에 Example.com의 임시 직원에 대한 비밀번호 정책을 추가합니다. 비밀번호 정책이 `pwp.ldif`라는 파일로 저장되었습니다.

```
$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f pwp.ldif
Enter bind password:
adding new entry cn=TempPolicy,dc=example,dc=com

$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w --b dc=example,dc=com \
"(&(objectclass=ldapsubentry)(cn=temppolicy))"
Enter bind password:
version: 1
dn: cn=TempPolicy,dc=example,dc=com
objectClass: top
objectClass: pwdPolicy
objectClass: LDAPsubentry
```

```
cn: TempPolicy
pwdCheckQuality: 2
pwdLockout: TRUE
pwdLockoutDuration: 300
pwdMaxFailure: 3
pwdMustChange: TRUE
$
```

Example.ldif에 표시된 것처럼 kvaughan은 dc=example,dc=com 항목 수정 액세스 권한이 있는 Human Resources 관리자입니다. Example.ldif에 표시된 것처럼 Vaughan의 바인드 비밀번호는 bribery입니다.

참조 정의된 정책이 적용되는 사용자 계정을 정의하려면 183 페이지 "개인 계정에 비밀번호 정책을 할당하는 방법" 또는 184 페이지 "역할 및 CoS를 사용하여 비밀번호 정책을 할당하는 방법"을 참조하십시오.

▼ 개인 계정에 비밀번호 정책을 할당하는 방법

이 절차에서는 단일 사용자 계정에 기존 비밀번호 정책을 할당합니다.

주 – 이 절차를 완료하려면 할당할 특수 비밀번호 정책이 있어야 합니다. 181 페이지 "비밀번호 정책을 만드는 방법"을 참조하십시오.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

별도로 명시하지 않는 한 여기에 표시된 데이터 예는 Example.ldif에서 가져온 것입니다.

- **사용자 항목의 pwdPolicySubentry 속성 값에 비밀번호 정책 DN을 추가합니다.**
  예를 들어 다음 명령은 181 페이지 "비밀번호 정책을 만드는 방법"에 정의된 비밀번호 정책을 DN이 uid=dmiller,ou=people,dc=example,dc=com인 David Miller의 항목에 할당합니다.

```
$ cat pwp.ldif
dn: uid=dmiller,ou=people,dc=example,dc=com
changetype: modify
add: pwdPolicySubentry
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com

$ ldapmodify -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f pwp.ldif
Enter bind password:
modifying entry uid=dmiller,ou=people,dc=example,dc=com
```

```
$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w - -b dc=example,dc=com \
"(uid=dmiller)" pwdPolicySubentry
Enter bind password:
version: 1
dn: uid=dmiller, ou=People, dc=example,dc=com
pwdPolicySubentry: cn=TempPolicy,dc=example,dc=com
$
```

Example.ldif에 표시된 것처럼 kvaughan은 dc=example,dc=com 항목 수정 액세스 권한이 있는 Human Resources 관리자입니다. Example.ldif에 표시된 것처럼 Vaughan의 바인드 비밀번호는 bribery입니다.

▼ 역할 및 CoS를 사용하여 비밀번호 정책을 할당하는 방법

이 절차에서는 역할 및 서비스 클래스(CoS)를 적용하여 기존의 특수 비밀번호 정책을 사용자 집합에 할당합니다. 역할 및 CoS에 대한 자세한 내용은 9 장를 참조하십시오.

주 – 이 절차를 완료하려면 할당할 특수 비밀번호 정책이 있어야 합니다. 181 페이지 “비밀번호 정책을 만드는 방법”을 참조하십시오.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

별도로 명시하지 않는 한 여기에 표시된 데이터 예는 Example.ldif에서 가져온 것입니다.

1 비밀번호 정책을 적용할 항목에 대한 역할을 만듭니다.

예를 들어 다음 명령은 Example.com의 임시 직원에 대한 필터링된 역할을 만듭니다.

```
$ cat tmp.ldif
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

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f tmp.ldif
Enter bind password:
modifying entry cn=TempFilter,ou=people,dc=example,dc=com
```

```
$
```

Example.ldif에 표시된 것처럼 kvaughan은 dc=example,dc=com 항목 수정 액세스 권한이 있는 Human Resources 관리자입니다. Example.ldif에 표시된 것처럼 Vaughan의 바인드 비밀번호는 bribery입니다.

2 비밀번호 정책 항목의 DN을 생성할 서비스 클래스를 만듭니다.

DN은 사용자가 만든 역할이 있는 사용자의 pwdPolicySubentry 속성 값입니다.

예를 들어 다음 명령은 Example.com의 임시 직원에 대한 필터링된 역할을 만듭니다. 다음 명령은 cn=TempPolicy,dc=example,dc=com을 역할이 있는 사용자에게 할당합니다.

```
$ cat cos.ldif
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
pwdPolicySubentry: 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: pwdPolicySubentry operational

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f cos.ldif
Enter bind password:
modifying entry cn=TempFilter,ou=people,dc=example,dc=com

$
```

상태가 contractor인 사용자는 이제 cn=TempPolicy,dc=example,dc=com 비밀번호 정책에 적용됩니다.

▼ 첫 번째 로그인 비밀번호 정책을 설정하는 방법

대부분의 배포에서 새 계정에 적용할 비밀번호 정책은 설정된 계정에 적용할 비밀번호 정책과 다릅니다. 이 절에서는 첫 번째 로그인 비밀번호 정책에 대해 설명합니다. 이 정책은 사용자에게 새로 만든 계정을 3일 동안 사용하여 계정이 잠기기 전에 새 비밀번호를 설정할 수 있도록 허용합니다. 이 정책은 비밀번호를 재설정한 사용자의 경우와 동일하게 적용됩니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 새로 만든 계정에 대해 특수 비밀번호 정책을 만듭니다.

예를 들어 만료 시간을 3일(259,200초)로 설정하는 비밀번호 정책 항목을 추가합니다. 또한 이 비밀번호 정책에서는 pwdMustChange(5dsat)가 TRUE로 설정되어 있습니다. 즉, 사용자가 처음 바인드할 때 비밀번호를 변경해야 합니다.

```
$ cat firstLogin.ldif
dn: cn=First Login,dc=example,dc=com
objectClass: top
objectClass: LDAPsubentry
objectClass: pwdPolicy
objectClass: sunPwdPolicy
cn: First Login
passwordStorageScheme: SSHA
pwdAttribute: userPassword
pwdInHistory: 0
pwdExpireWarning: 86400
pwdLockout: TRUE
pwdMinLength: 6
pwdMaxFailure: 3
pwdMaxAge: 259200
pwdFailureCountInterval: 600
pwdAllowUserChange: TRUE
pwdLockoutDuration: 3600
pwdMinAge: 0
pwdCheckQuality: 2
pwdMustChange: TRUE

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f firstLogin.ldif
Enter bind password:
adding new entry cn=First Login,dc=example,dc=com

$
```

2 새로 만든 모든 계정을 포함하는 역할을 만듭니다.

이 역할을 만들 때 새로 만든 계정을 설정된 계정과 구분하는 몇 가지 방법을 설정합니다.

a. 새 계정의 pwdReset(5dsat) 속성을 TRUE로 설정합니다.

비밀번호 관리자와 같은 다른 사용자가 사용자의 비밀번호를 변경하면 pwdReset이 TRUE로 설정됩니다.

b. 새 계정을 식별하는 역할을 만듭니다.

예를 들어 다음 명령은 비밀번호를 재설정한 계정에 대한 역할을 만듭니다.

```
$ cat newRole.ldif
dn: cn=First Login Role,ou=people,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: nsRoleDefinition
objectclass: nsComplexRoleDefinition
objectclass: nsFilteredRoleDefinition
cn: First Login Role
nsRoleFilter: (pwdReset=TRUE)
description: Role to assign password policy for new and reset accounts

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f newRole.ldif
Enter bind password:
adding new entry cn=First Login Role,ou=people,dc=example,dc=com

$
```

3 서비스 클래스를 사용하여 새로 만든 계정에 대한 비밀번호 정책을 할당합니다.

```
$ cat newCoS.ldif
dn: cn=First Login Template,dc=example,dc=com
objectClass: top
objectClass: nsContainer

dn: cn="cn=First Login Role,ou=people,dc=example,dc=com",
 cn=First Login Template,dc=example,dc=com
objectClass: top
objectClass: extensibleObject
objectClass: LDAPSubEntry
objectClass: CoSTemplate
cosPriority: 1
pwdPolicySubentry: cn=First Login,dc=example,dc=com

dn: cn=First Login CoS,dc=example,dc=com
objectClass: top
objectClass: LDAPSubEntry
objectClass: CoSSuperDefinition
objectClass: CoSClassicDefinition
```

```
cosTemplateDN: cn=First Login Template,dc=example,dc=com
cosSpecifier: nsRole
cosAttribute: pwdPolicySubentry operational

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -f newCoS.ldif
Enter bind password:
adding new entry cn=First Login Template,dc=example,dc=com

adding new entry cn="cn=First Login Role,ou=people,dc=example,dc=com",
 cn=First Login Template,dc=example,dc=com

adding new entry cn=First Login CoS,dc=example,dc=com

$
```

예 7–1 비밀번호 정책 할당 검사

추가한 역할에 맞는 새 사용자를 추가합니다. 사용자를 추가하여 새 비밀번호 정책이 새 사용자에게는 적용되고 기존 사용자에게는 적용되지 않는지를 확인합니다.

```
$ cat quentin.ldif
dn: uid=qcubbins,ou=People,dc=example,dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
uid: qcubbins
givenName: Quentin
sn: Cubbins
cn: Quentin Cubbins
mail: quentin.cubbins@example.com
userPassword: ch4ngeM3!
description: New account

$ ldapmodify -a -D uid=kvaughan,ou=people,dc=example,dc=com -w - -f quentin.ldif
Enter bind password:
adding new entry uid=qcubbins,ou=People,dc=example,dc=com

$ ldapsearch -D uid=kvaughan,ou=people,dc=example,dc=com -w - \
-b dc=example,dc=com uid=qcubbins nsrole pwdPolicySubentry
Enter bind password:
version: 1
dn: uid=qcubbins,ou=People,dc=example,dc=com
nsrole: cn=first login role,ou=people,dc=example,dc=com
pwdPolicySubentry: cn=First Login,dc=example,dc=com
$ ldapsearch -b dc=example,dc=com uid=bjensen nsrole pwdPolicySubentry
version: 1
```

```
dn: uid=bjensen, ou=People, dc=example,dc=com
$
```

Barbara Jensen의 기존 계정에는 기본 비밀번호 정책이 적용됩니다. 그러나 Quentin Cubbins의 새 계정에는 정의된 비밀번호 정책이 적용됩니다.

pwdSafeModify가 TRUE인 경우 명령줄에서 비밀번호 수정

사용자의 비밀번호 정책에서 `pwdSafeModify`가 `TRUE`로 설정되어 있는 경우 비밀번호를 변경하려면 이전 비밀번호를 새 비밀번호와 함께 제공해야 합니다. `dsconf set-server-prop pwd-safe-modify-enabled:on` 명령은 기본 비밀번호 정책에 동일한 효과가 있습니다.

`ldappasswd`(1) 명령을 사용하여 비밀번호를 변경할 수 있습니다. 이 명령은 안전한 비밀번호 수정을 지원합니다. 이 명령은 RFC 3062, **LDAP Password Modify Extended Operation** (`http://www.ietf.org/rfc/rfc3062.txt`)을 구현합니다.

`ldapmodify`(1) 명령을 사용하여 비밀번호를 변경할 수 있습니다. 이 경우 `ldapmodify` 명령에 전달되는 LDIF는 다음과 같습니다.

```
dn: DN of user whose password you are changing
changetype: modify
delete: userPassword
userPassword: old password
-
add: userPassword
userPassword: new password
```

LDAP 비밀번호 수정 확장 작업을 사용할 수도 있습니다. 확장 작업 지원 설정에 대한 자세한 내용은 190 페이지 “비밀번호 수정 확장 작업을 사용하여 비밀번호를 재설정하는 방법”을 참조하십시오.

만료된 비밀번호 재설정

비밀번호 정책에서 비밀번호 만료를 시행할 때 사용자가 기간 내에 비밀번호를 변경하지 못하는 경우가 있습니다. 이 절에서는 만료된 비밀번호를 변경하는 방법에 대해 설명합니다.

주 – 디렉토리 서버는 항목의 비밀번호가 수정될 때마다 pwdChangedTime(5dsat) 작동 가능 속성을 업데이트합니다. 따라서 비밀번호 만료가 활성화될 때까지 기다린 후 비밀번호 만료를 활성화하면 이미 만료된 사용자 비밀번호가 즉시 만료됩니다. 이 동작을 원하지 않으면 경고 및 정상 로그인을 사용하십시오.

이 절에는 비밀번호 수정 확장 작업을 사용하여 비밀번호를 재설정하고 비밀번호가 만료된 경우에 정상 인증을 허용하는 절차가 포함되어 있습니다.

이 절에서 설명된 메커니즘은 실제 사용자의 디렉토리 상호 작용을 처리하는 응용 프로그램과 관리자를 위한 것입니다. 일반적으로 응용 프로그램을 사용하여 최종 사용자가 이 메커니즘을 의도된 방식으로 사용하고 있는지 확인합니다.

▼ 비밀번호 수정 확장 작업을 사용하여 비밀번호를 재설정하는 방법

비밀번호가 만료되면 사용자 계정이 잠깁니다. 비밀번호를 재설정하면 계정 잠금이 해제됩니다. 관리자와 같은 다른 사용자가 비밀번호를 재설정할 수 있습니다. 비밀번호를 재설정하면 디렉토리 서버에서 사용자 계정을 잠금 해제합니다. 디렉토리 서버는 RFC 3062, **LDAP Password Modify Extended Operation** (http://www.ietf.org/rfc/rfc3062.txt)을 지원합니다. 확장된 작업을 사용하면 디렉토리 어드민 관리자나 응용 프로그램에서 비밀번호 재설정을 통해 계정 잠금을 해제할 수 있습니다.

이 절차에 표시된 것처럼 비밀번호 수정 확장 작업을 사용하도록 허용할 경우에는 유의하십시오. 액세스 권한이 신뢰할 수 있는 관리자와 응용 프로그램으로 제한됩니다. 네트워크를 통해 비밀번호를 일반 텍스트로 전달할 수 없습니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1. **비밀번호 관리자 또는 비밀번호 관리 응용 프로그램에 사용자 액세스 권한을 부여합니다.**

2. **비밀번호 관리자 액세스를 통해 비밀번호 수정 확장 작업을 사용하도록 허용합니다.**

   다음 명령은 Password Managers 역할이 있는 구성원이 SSL을 통해 연결된 경우 비밀번호 수정 확장 작업을 사용하도록 허용하는 ACI를 설정합니다.

```
$ cat exop.ldif
dn: oid=1.3.6.1.4.1.4203.1.11.1,cn=features,cn=config
objectClass: top
objectClass: directoryServerFeature
oid: 1.3.6.1.4.1.4203.1.11.1
cn: Password Modify Extended Operation
aci: (targetattr != "aci")(version 3.0;
```

```
 acl "Password Modify Extended Operation
 "; allow( read, search, compare, proxy ) (roledn = "
 ldap:///cn=Password Managers,dc=example,dc=com" and authmethod = "SSL");)

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f exop.ldif
Enter bind password:
adding new entry oid=1.3.6.1.4.1.4203.1.11.1,cn=features,cn=config

$
```

`cn=features,cn=config` 아래의 항목을 사용하여 비밀번호 수정 확장 작업에 대한 액세스 권한을 관리할 수 있습니다.

3 **비밀번호 관리자가 사용자 비밀번호를 재설정합니다.**

이 단계는 사용자 계정의 잠금을 해제하고 `ldappasswd`(1) 명령을 사용하여 완료할 수 있습니다.

4 **(옵션) 사용자가 비밀번호를 변경해야 하는 경우 비밀번호 관리자가 사용자에게 알림 메시지를 보냅니다.**

사용자는 자신의 항목을 제어하는 비밀번호 정책에 `pwdMustChange: TRUE`가 포함되어 있는 경우 비밀번호를 재설정한 후 변경해야 합니다.

▼ 비밀번호가 만료된 경우 정상 인증을 허용하는 방법

이 절차에서는 사용자가 만료된 비밀번호를 변경할 수 있도록 정상 인증을 허용하는 방법에 대해 설명합니다.

정상 인증은 비밀번호 정책 요청 및 응답 컨트롤을 처리하는 응용 프로그램에서 관리하도록 설계되었습니다. 이 절차에서는 응용 프로그램에서 컨트롤을 사용하는 방법에 대한 간단한 예를 보여줍니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 **비밀번호 정책 요청 및 응답 컨트롤을 사용하는 응용 프로그램에 대한 액세스 권한이 사용자에게 있는지 확인합니다.**

응용 프로그램에서는 사용자가 정상 인증을 적절하게 처리하는지 확인해야 합니다.

2 **응용 프로그램에서 비밀번호 정책 컨트롤을 사용하도록 허용합니다.**

다음 명령은 `Password Managers` 역할을 가진 구성원에게 비밀번호 정책 컨트롤을 사용하도록 허용하는 ACI를 설정합니다.

```
$ cat ctrl.ldif
dn: oid=1.3.6.1.4.1.42.2.27.8.5.1,cn=features,cn=config
objectClass: top
```

```
objectClass: directoryServerFeature
oid: 1.3.6.1.4.1.42.2.27.8.5.1
cn: Password Policy Controls
aci: (targetattr != "aci")(version 3.0; acl "Password Policy Controls
 "; allow( read, search, compare, proxy ) roledn = "
 ldap:///cn=Password Managers,dc=example,dc=com";)

$ ldapmodify -a -D cn=admin,cn=Administrators,cn=config -w - -f ctrl.ldif
Enter bind password:
adding new entry oid=1.3.6.1.4.1.42.2.27.8.5.1,cn=features,cn=config

$
```

`cn=features,cn=config` 아래의 항목은 오로지 비밀번호 정책 요청 및 응답 컨트롤을 사용하는 작업에 대한 액세스를 관리하기 위한 것입니다.

3. **비밀번호 정책에서 `pwdGraceAuthNLimit`를 비밀번호가 만료된 이후 허용할 인증 횟수로 설정합니다.**

4. **응용 프로그램에서는 최종 사용자가 정상 인증 기간이 종료되기 전에 만료된 비밀번호를 변경하도록 안내해야 합니다.**

계정 등록 정보 설정

다음 절에서는 계정에 대한 조회 제한, 크기 제한, 시간 제한 및 유휴 시간 초과를 설정하는 방법에 대해 설명합니다.

▼ 계정에 대한 조회 제한을 설정하는 방법

- **`ldapmodify` 명령을 사용하여 `nsLookThroughLimit` 값을 설정합니다.**

  다음 명령은 Barbara Jensen에 대한 조회 제한을 제거합니다.

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsLookThroughLimit
nsLookThroughLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

▼ 계정에 대한 크기 제한을 설정하는 방법

- **ldapmodify 명령을 사용하여 nsSizeLimit 값을 설정합니다.**
  다음 명령은 Barbara Jensen에 대한 크기 제한을 제거합니다.

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsSizeLimit
nsSizeLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

▼ 계정에 대한 시간 제한을 설정하는 방법

- **ldapmodify 명령을 사용하여 nsTimeLimit 값을 설정합니다.**
  다음 명령은 Barbara Jensen에 대한 시간 제한을 제거합니다.

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsTimeLimit
nsTimeLimit: -1
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

▼ 계정에 대한 유휴 시간 초과를 설정하는 방법

- **ldapmodify 명령을 사용하여 nsIdleTimeout 값을 설정합니다.**
  다음 명령은 Barbara Jensen에 대한 유휴 시간 초과를 5분(300초)으로 설정합니다.

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bjensen,ou=people,dc=example,dc=com
changetype: modify
add: nsIdleTimeout
```

```
nsIdleTimeout: 300
^D
modifying entry uid=bjensen,ou=people,dc=example,dc=com

^D
$
```

수동으로 계정 잠금

디렉토리 서버에서는 지정된 실패한 바인드 시도 횟수 이후에 계정 잠금을 시행하는 비밀번호 정책을 구성할 수 있습니다. 자세한 내용은 172 페이지 "계정 잠금 정책"을 참조하십시오. 이 절에서는 디렉토리 관리자가 사용할 수 있는 수동 계정 잠금 및 활성화 도구에 대해 설명합니다.

디렉토리 관리자는 잠금 기간 타이머를 사용하지 않고 계정 잠금을 관리할 수 있습니다. 잠긴 계정은 비밀번호를 수동으로 재설정할 때까지 잠겨진 상태로 유지됩니다. 또한 디렉토리 관리자는 특정 계정을 무한 기간 동안 비활성 상태로 렌더링할 수 있습니다.

이 절에서는 계정 상태를 검사하고 계정을 비활성 상태로 렌더링하며 다시 활성화하는 방법에 대해 설명합니다.

▼ 계정 상태를 검사하는 방법

여기에 표시된 대로 계정 상태를 검사합니다.

주 – 디렉토리 관리자로 바인드해야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **`ns-accountstatus` 명령을 사용하여 계정 또는 역할 상태를 검사합니다.**

  다음 명령은 Barbara Jensen의 계정 상태를 검사합니다.

  ```
  $ ns-accountstatus -D "cn=Directory Manager" -j pwd.txt \
   -I uid=bjensen,ou=people,dc=example,dc=com
  uid=bjensen,ou=people,dc=example,dc=com activated.
  $
  ```

  자세한 내용은 `ns-accountstatus`(1M) 설명서 페이지를 참조하십시오.

▼ 계정을 비활성 상태로 렌더링하는 방법

여기에 표시된 대로 계정 또는 역할을 비활성 상태로 렌더링합니다.

주 – 디렉토리 관리자로 바인드해야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **ns-inactivate 명령을 사용하여 계정 또는 역할을 비활성 상태로 렌더링합니다.**
  다음 명령은 Barbara Jensen의 계정을 비활성 상태로 렌더링합니다.

```
$ ns-inactivate -D "cn=Directory Manager" -j pwd.txt \
-I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com inactivated.
$
```

  자세한 내용은 ns-inactivate(1M) 설명서 페이지를 참조하십시오.

▼ 계정을 다시 활성화하는 방법

여기에 표시된 대로 계정 또는 역할을 잠금 해제합니다.

주 – 디렉토리 관리자로 바인드해야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **ns-activate 명령을 사용하여 계정 또는 역할을 다시 활성화합니다.**
  다음 명령은 Barbara Jensen의 계정을 다시 활성 상태로 렌더링합니다.

```
$ ns-activate -D "cn=Directory Manager" -j pwd.txt \
-I uid=bjensen,ou=people,dc=example,dc=com
uid=bjensen,ou=people,dc=example,dc=com activated.
$
```

  자세한 내용은 ns-activate(1M) 설명서 페이지를 참조하십시오.

◆ ◆ ◆ 8 장

8 디렉토리 서버 백업 및 복원

디렉토리 서버에서 관리하는 데이터는 대량으로 가져오는 경우가 많습니다. Directory Server Enterprise Edition은 전체 접미어를 가져오거나 내보내기 위한 도구를 제공합니다. 모든 접미어를 동시에 백업하고, 이 백업을 사용하여 모든 데이터를 복원할 수 있는 도구도 제공됩니다.

백업 또는 복원 작업을 시작하기 전에 백업 및 복원 전략을 상황에 맞게 설계해야 합니다. 다른 백업 옵션, 고려해야 할 사항, 백업 및 복원 전략 지침에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 "Designing Backup and Restore Policies"를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

이진 백업

이 절에서는 디렉토리 데이터의 이진 백업 수행 방법에 대해 설명합니다. 이 절에서 설명하는 이진 백업 절차 외에도 복제 토폴로지에서 접미어를 초기화하는 데 사용할 이진 백업을 만들 수도 있습니다. 249 페이지 "이진 복사를 사용하여 복제된 접미어 초기화"를 참조하십시오.

디렉토리 데이터만 백업

이진 데이터 백업에서는 데이터베이스 파일이 나중에 손상되거나 삭제된 경우에 사용할 수 있는 디렉토리 데이터 복사본을 저장합니다. 구성 데이터는 이 작업을 통해 백업되지 않습니다. 재해 복구를 위해 전체 디렉토리 서버를 백업하려면 210 페이지 "재해 복구"를 참조하십시오.

주의 – 백업 작업 중에는 서버를 중지하지 마십시오.

백업은 **지연 제거**보다 더 자주 수행해야 합니다. `nsDS5ReplicaPurgeDelay` 속성을 통해 지정되는 지연 제거는 지정된 시간 이후에 변경 로그에서 내부 제거 작업이 수행되는 기간(초)입니다. 기본 지연 제거는 604800초(1주)입니다. 변경 로그에서는 업데이트 레코드를 복제 여부에 관계없이 유지관리합니다.

백업을 지연 제거보다 낮은 빈도로 수행하면 변경 로그가 백업되기 이전에 지워질 수 있습니다. 따라서 백업을 사용하여 데이터를 복원할 경우 변경 사항이 손실됩니다.

기본적으로 이 절에 설명된 모든 백업 절차는 서버 파일의 복사본을 동일한 호스트에 저장합니다. 보안을 강화하려면 이 백업을 복사하여 다른 시스템이나 파일 시스템에 저장해야 합니다.

▼ 디렉토리 데이터를 백업하는 방법

`dsadm backup` 명령을 실행하려면 디렉토리 서버를 중지해야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **디렉토리 데이터를 백업합니다.**

  ```
  $ dsadm backup instance-path archive-dir
  ```

  예를 들면 다음과 같습니다.

  ```
  $ dsadm backup /local/ds /local/tmp/20051205
  ```

주 – 서버가 실행 중일 때 `dsconf backup` 명령을 사용하여 디렉토리 데이터를 백업할 수 있습니다. 그러나 백업을 실행하는 동안 디렉토리 데이터가 변경되면 복구하기가 더 어려워집니다. `dsconf backup`을 사용할 때 이러한 문제를 방지하려면 복제 참조를 설정하거나 서버를 읽기 전용으로 지정합니다.

`dsadm` 및 `dsconf` 명령에 대한 자세한 내용은 `dsadm`(1M) 및 `dsconf`(1M) 설명서 페이지를 참조하십시오.

▼ dse.ldif 파일을 백업하는 방법

서버를 복원할 때 dse.ldif 구성 파일에는 해당 서버를 백업했을 당시와 동일한 구성 정보가 있어야 합니다.

- **dse.ldif 구성 파일을 백업합니다.**

  ```
  $ cp instance-path/config/dse.ldif archive-dir
  ```

  다음 작업을 수행하면 디렉토리 서버에서 dse.ldif 구성 파일을 *instance-path*/config 디렉토리에 자동으로 백업합니다.

  - 디렉토리 서버를 시작하면 dse.ldif 파일의 백업이 dse.ldif.startOK라는 이름의 파일로 만들어집니다.
  - cn=config 분기가 수정되면 서버는 먼저 이 파일을 config 디렉토리의 dse.ldif.bak 파일에 백업한 다음 dse.ldif 파일에 수정 사항을 씁니다.

파일 시스템 백업

이 절차에서는 **고정 모드** 기능을 사용합니다. 고정 모드를 사용하면 디스크의 데이터베이스 업데이트를 중지할 수 있으므로 파일 시스템 스냅샷을 안전하게 가져올 수 있습니다. 이 모드는 강력한 백업 보장을 위한 추가 조치로 사용할 수 있습니다.

파일 시스템 백업이 진행 중일 때에는 서버에서 사용자 데이터를 디스크에 쓰지 않아야 합니다. 해당 기간 동안 업데이트가 발생하지 않는 것이 확실한 경우에만 백업을 수행합니다. 업데이트가 발생할지 확실하지 않은 경우에는 서버를 고정 모드로 전환한 다음 백업합니다.

고정 모드에 있는 서버는 액세스 로그 및 오류 로그에 대한 쓰기를 계속합니다. 단일 서버 토폴로지에서 고정 모드가 설정된 상태에서 작업이 수신되면 LDAP 오류가 반환됩니다. 오프라인 데이터베이스에 대한 표준 오류가 오류 메시지로 기록됩니다. 복제된 토폴로지에서는 참조가 반환됩니다. 고정 모드가 제대로 작동하려면 데이터베이스에서 다른 작업을 실행하지 않아야 합니다.

고정 모드에 있는 서버의 데이터베이스는 읽기 전용 모드에 있는 서버의 데이터베이스보다 안정적입니다. 고정 모드와는 달리, 읽기 전용 모드는 만들 작업과 수정할 구성 항목을 허용합니다. 고정 모드를 설정하면 모든 구성된 데이터베이스가 오프라인으로 전환됩니다. 진행 중인 모든 내부 작업은 데이터베이스가 오프라인으로 전환된다는 알림을 받게 됩니다. 진행 중인 LDAP 작업이 완료되고 데이터베이스 환경이 플러시됩니다. 사용자 데이터 검색을 포함하여 이후에 수신되는 모든 작업은 고정 모드를 해제할 때까지 거부됩니다. 그러나 고정 모드가 설정되어 있는 동안에도 구성 매개 변수는 검색할 수 있습니다.

▼ 파일 시스템을 백업하는 방법

이 절차의 일부로, DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오. 해당 절차의 다른 부분은 명령줄에서만 수행할 수 있습니다.

1. **(옵션) 서버를 고정 모드로 전환합니다.**

   ```
   $ dsconf set-server-prop -h host -p port read-write-mode:frozen
   ```

2. **사용 중인 파일 시스템 유형에 적합한 도구를 사용하여 파일 시스템을 백업합니다.**

3. **서버가 고정 모드인 경우 서버를 다시 읽기-쓰기 모드로 지정합니다.**

   ```
   $ dsconf set-server-prop -h host -p port read-write-mode:read-write
   ```

   서버가 다른 서버로부터 복제 업데이트를 받을 경우 고정 모드를 해제하면 복제 업데이트가 바로 시작됩니다.

LDIF에 백업

LDIF에 백업을 사용하면 디렉토리 데이터를 서식이 지정된 LDIF 파일로 백업할 수 있습니다.

LDIF로 내보내기

LDIF를 사용하여 접미어의 내용을 내보내서 디렉토리 데이터를 백업할 수 있습니다. 데이터 내보내기는 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.

▼ 접미어를 LDIF로 내보내는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **다음 명령 중 하나를 사용하여 접미어를 LDIF 파일로 내보냅니다.**

- 서버가 로컬이고 중지된 경우 다음을 입력합니다.

```
$ dsadm export instance-path suffix-DN LDIF-file
```

- 서버가 원격이고 실행 중인 경우 다음을 입력합니다.

```
$ dsconf export -h host -p port suffix-DN LDIF-file
```

아래 예에서는 `dsconf export` 명령을 사용하여 두 접미어를 단일 LDIF 파일로 내보냅니다.

```
$ dsconf export -h host1 -p 1389 ou=people,dc=example,dc=com \
 ou=contractors,dc=example,dc=com /local/ds/ldif/export123.ldif
```

`dsadm export` 및 `dsconf export` 명령을 `--no-repl` 옵션과 함께 사용하여 복제 정보를 내보내지 않도록 지정할 수도 있습니다. 기본적으로 복제된 접미어는 LDIF 파일에 복제 정보와 함께 내보내집니다. 결과로 작성된 LDIF 파일에는 복제 메커니즘에서 사용하는 속성 하위 유형이 포함되어 있습니다. 그런 다음 245 페이지 “복제본 초기화”에 설명된 것처럼 사용자 서버에서 이 LDIF 파일을 가져와서 사용자 복제본을 초기화할 수 있습니다.

이러한 명령에 대한 자세한 내용은 `dsadm`(1M) 및 `dsconf`(1M) 설명서 페이지를 참조하십시오.

이진 복원

다음 절차에서는 디렉토리에서 접미어를 복원하는 방법에 대해 설명합니다. 서버는 198 페이지 “디렉토리 데이터만 백업”에 설명된 것처럼 백업되어 있어야 합니다. 복제 계약에 포함된 접미어를 복원하기 전에 206 페이지 “복제된 접미어 복원”을 읽어 보십시오.

주의 – 복원 작업 중에는 서버를 중지하지 마십시오. 서버를 복원하면 기존 데이터베이스 파일을 덮어쓰기 때문에 백업 이후에 수행한 데이터 수정 사항이 손실됩니다.

▼ 서버를 복원하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **다음 명령 중 하나를 사용하여 서버를 복원합니다.**
  - 서버가 로컬이고 중지된 경우 다음을 입력합니다.

```
$ dsadm restore instance-path archive-dir
```

예를 들어 백업 디렉토리에서 백업을 복원하려면 다음을 입력합니다.

```
$ dsadm restore /local/ds/ local/ds/bak/2006_07_01_11_34_00
```

- 서버가 원격이고 실행 중인 경우 다음을 입력합니다.

```
$ dsconf restore -h host -p port archive-dir
```

예를 들어 백업 디렉토리에서 백업을 복원하려면 다음을 입력합니다.

```
$ dsconf restore -h host1 -p 1389 /local/ds/bak/2006_07_01_11_34_00
```

이러한 명령에 대한 자세한 내용은 dsadm(1M) 및 dsconf(1M) 설명서 페이지를 참조하십시오.

dse.ldif 구성 파일 복원

디렉토리 서버는 아래 디렉토리에 dse.ldif 파일의 백업 복사본 두 개를 만듭니다.

instance-path/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 포함된 파일을 디렉토리에 복사합니다.

▼ dse.ldif 구성 파일을 복원하는 방법

이 절차의 일부로, DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오. 해당 절차의 다른 부분은 명령줄에서만 수행할 수 있습니다.

1 서버를 중지합니다.

```
$ dsadm stop instance-path
```

2 구성 파일이 포함된 디렉토리로 변경합니다.

```
$ cd instance-path/config
```

3 아래와 같이 유효한 백업 구성 파일로 dse.ldif 파일을 덮어씁니다.

```
$ cp dse.ldif.startOK dse.ldif
```

4 다음 명령을 실행하여 서버를 시작합니다.

```
$ dsadm start instance-path
```

LDIF 파일에서 데이터 가져오기

다음과 같은 방법으로 데이터를 디렉토리 서버 접미어로 가져올 수 있습니다.

- LDIF 파일에서 접미어를 초기화합니다. 이 작업에서는 접미어의 현재 데이터를 삭제한 다음 LDIF 파일의 내용으로 대체합니다.
- LDIF 파일을 사용하여 대량 `ldapadd`, `ldapmodify` 또는 `ldapdelete` 작업을 수행합니다. 그러면 디렉토리 접미어의 항목을 대량으로 추가, 수정 및 삭제할 수 있습니다.

아래 표에서는 접미어 초기화와 대량으로 항목 추가, 수정 및 삭제 작업의 차이점을 보여줍니다.

표 8–1 접미어 초기화와 대량 데이터 가져오기 비교

비교 영역	접미어 초기화	대량으로 항목 추가, 수정 및 삭제
내용 덮어쓰기	덮어쓰기 내용	내용 덮어쓰지 않음
LDAP 작업	추가만	추가, 수정, 삭제
성능	빠름	느림
서버 장애에 대한 응답	자동(장애 후에 모든 변경 사항 손실됨)	최상의 노력(장애가 발생하기 전의 모든 변경 사항 그대로 유지)
LDIF 파일 위치	로컬 콘솔 또는 로컬 서버	클라이언트 시스템
구성 정보(`cn=config`) 가져오기	구성 정보 가져오기	구성 정보 가져오지 않음

명령	서버가 로컬이고 중지된 경우:
	`dsadm import`
	서버가 원격이고 실행 중인 경우:
	`dsconf import`
		`ldapmodify -B`

접미어 초기화

접미어를 초기화하면 추가할 항목만 포함된 LDIF 파일의 내용이 접미어의 기존 데이터를 덮어씁니다.

접미어를 초기화하려면 디렉토리 관리자 또는 어드민 관리자로 인증되어야 합니다.

서버가 실행 중인 경우에는 디렉토리 관리자와 어드민 관리자만 루트 항목을 포함하는 LDIF 파일을 가져올 수 있습니다. 보안상 이러한 사용자에게만 접미어의 루트 항목(예: `dc=example,dc=com`.)에 액세스할 수 있습니다.

복제 계약에 포함된 접미어를 복원하기 전에 206 페이지 “복제된 접미어 복원”을 읽어 보십시오.

▼ 접미어를 초기화하는 방법

주 – UTF-8 문자 집합 인코딩을 사용하는 LDIF 파일만 가져올 수 있습니다.

접미어를 초기화할 경우에는 LDIF 파일에 해당 접미어의 모든 디렉토리 트리 노드와 루트 항목이 포함되어 있어야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **다음 명령 중 하나를 사용하여 LDIF 파일에서 접미어를 초기화합니다. 즉, 데이터베이스의 내용을 LDIF 파일로 가져옵니다.**

주의 – 이 명령은 접미어에 있는 데이터를 덮어씁니다.

  - 서버가 로컬이고 중지된 경우 다음을 입력합니다.

    ```
    $ dsadm import instance-path LDIF-file suffix-DN
    ```

    아래 예에서는 `dsadm import` 명령을 사용하여 LDIF 파일 두 개를 하나의 접미어로 가져옵니다.

    ```
    $ dsadm import /local/ds /local/file/example/demo1.ldif \
     /local/file/example/demo2.ldif dc=example,dc=com
    ```

  - 서버가 원격이고 실행 중인 경우 다음을 입력합니다.

    ```
    $ dsconf import -h host -p port LDIF-file suffix-DN
    ```

    다음 예에서는 `dsconf import`를 사용하여 LDIF 파일을 가져옵니다. 디렉토리 관리자와 같이 루트 권한을 가진 사용자로 인증되면 루트 권한이 없어도 명령을 실행할 수 있습니다.

    ```
    $ dsconf import -h host1 -p 1389 /local/file/example/demo1.ldif \
     ou=People,dc=example,dc=com
    ```

주 – `dsconf import` 또는 `dsconf reindex` 명령 중 하나 또는 모두를 여러 접미어에서 병렬로 실행할 경우 트랜잭션 로그가 계속 증가하여 성능이 저하될 수 있습니다.

이러한 명령에 대한 자세한 내용은 `dsadm`(1M) 및 `dsconf`(1M) 설명서 페이지를 참조하십시오.

대량으로 항목 추가, 수정 및 삭제

ldapmodify 작업을 수행할 때 항목을 대량으로 추가, 수정 또는 삭제할 수 있습니다. 기존 항목을 수정하거나 삭제할 업데이트 명령문이 포함된 LDIF 파일에서 항목이 지정됩니다. 이 작업에서 이미 존재하는 항목은 지우지 않습니다.

디렉토리 서버에서 관리되는 접미어를 대상으로 항목을 변경할 수 있습니다. 항목을 추가하는 다른 모든 작업과 마찬가지로 서버는 가져오는 새 항목을 모두 색인화합니다.

ldapmodify 명령은 LDAP를 통해 LDIF 파일을 가져온 다음 해당 파일에 포함되는 모든 작업을 수행합니다. 이 명령을 사용하여 모든 디렉토리 접미어에서 동시에 데이터를 수정할 수 있습니다.

복제 계약에 포함된 접미어를 복원하기 전에 206 페이지 "복제된 접미어 복원"을 참조하십시오.

▼ 대량으로 항목을 추가, 수정 및 삭제하는 방법

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.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **LDIF 파일에서 대량으로 추가, 수정 또는 삭제합니다.**

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -B baseDN -f LDIF-file
```

아래 예에서는 ldapmodify 명령을 사용하여 가져오기를 수행합니다. cn=Directory Manager 또는 cn=admin,cn=Administrators,cn=config와 같이 루트 권한이 있는 사용자로 인증되면 루트 권한이 없어도 이 명령을 실행할 수 있습니다. 마지막 매개변수는 가져올 LDIF 파일 이름을 지정합니다.

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - \
 -B dc=example,dc=com -f /local/ds/ldif/demo.ldif
```

복제된 접미어 복원

공급자 서버와 사용자 서버 간에 복제된 접미어를 복원하려면 몇 가지 주의해야 할 사항이 있습니다. 가능하면 백업을 사용하여 접미어를 복원하는 대신 복제 메커니즘을 통해 접미어를 업데이트합니다.

공급자나 허브 인스턴스를 복원할 때 서버 구성은 백업을 만들었을 당시의 구성과 같아야 합니다. 이렇게 하려면 `dse.ldif` 파일을 먼저 복원한 다음 디렉토리 서버 데이터를 복원합니다. 202 페이지 “`dse.ldif` 구성 파일 복원”을 참조하십시오.

이 절에서는 복제본의 복원 방법과 시기, 그리고 작업 후에 다른 복제본과 동기화하는 방법에 대해 설명합니다. 복제본을 초기화하려면 245 페이지 “복제본 초기화”를 참조하십시오.

복제된 접미어가 대용량이고 많은 항목을 추가하여 복제 업데이트가 올바르게 추가되도록 하려면 253 페이지 “대용량 복제된 접미어에 많은 항목을 증분하여 추가”를 참조하십시오.

이 절은 다음과 같은 내용으로 구성되어 있습니다.

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, 사용자 데이터가 백업보다 최신 버전일 수 있으므로 사용자 중 하나에 있는 데이터를 사용하는 방법을 고려합니다. 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 사용자를 다시 초기화해야 한다는 메시지가 기록됩니다.

다중 마스터 시나리오에서 공급자 복원

다중 마스터 복제에서는 복제된 데이터의 신뢰할 수 있는 복사본이 각각의 마스터에 저장되어 있습니다. 이전 백업은 현재의 복제본 내용으로 업데이트되면서 만료되었을 수 있으므로 복원할 수 없습니다. 가능하면 복제 메커니즘에서 다른 마스터의 내용을 사용하여 마스터를 최신 상태로 유지할 수 있도록 합니다.

이렇게 할 수 없는 경우에만 다음 방법 중 하나로 다중 마스터 복제본을 복원합니다.

- 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. 246 페이지 “LDIF에서 복제본 초기화”를 참조하십시오.
- 수백만 개의 항목이 있는 복제본의 경우, 이진 복사 기능을 사용하면 다른 마스터 중 하나의 최신 백업을 보다 신속하게 복원할 수 있습니다. 249 페이지 “이진 복사를 사용하여 복제된 접미어 초기화”를 참조하십시오.
- 마스터의 백업이 다른 마스터 중 **어느** 하나에 있는 변경 로그 내용의 최대 수명보다 오래되지 않은 경우 이 백업을 사용하여 마스터를 복원할 수 있습니다. 변경 로그 수명에 대한 자세한 내용은 239 페이지 “마스터 복제본에 대한 변경 로그 설정을 수정하는 방법”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. 209 페이지 “다중 마스터 시나리오에서 마스터 복원”에 설명된 것처럼 복제본은 이 동작을 통해 다른 마스터와 동기화할 수 있으며, 그 이후에 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 복제 메커니즘에서 자동으로 허브 복제본을 최신 상태로 유지할 수 없는 경우에만 적용됩니다. 예를 들어 데이터베이스 파일이 손상되었거나 복제가 장시간 중단되는 경우입니다. 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. 203 페이지 "접미어 초기화"를 참조하십시오.
- 수백만 개의 항목이 있는 복제본의 경우, 이진 복사 기능을 사용하면 다른 복제된 허브 접미어에서 받은 최신 백업을 보다 신속하게 복원할 수 있습니다. 249 페이지 "이진 복사를 사용하여 복제된 접미어 초기화"를 참조하십시오. 복사할 다른 허브 복제본이 없을 경우, 가능하면 이전 항목에 설명된 것처럼 허브를 다시 초기화하거나 다음 항목에 설명된 것처럼 허브를 복원합니다.
- 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. 허브를 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 사용자를 **반드시** 다시 초기화해야 합니다.

전용 소비자 복원

이 절의 내용은 복제 메커니즘에서 자동으로 전용 소비자 복제본을 최신 상태로 유지할 수 없는 경우에만 적용됩니다. 예를 들어 데이터베이스 파일이 손상되었거나 복제가 장시간 중단되는 경우입니다. 이 경우 다음 방법 중 하나를 사용하여 사용자를 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 소비자로 전송되어 바로 복제할 수 있는 상태가 됩니다. 246 페이지 "LDIF에서 복제본 초기화"를 참조하십시오.
- 수백만 개의 항목이 있는 복제본의 경우, 이진 복사 기능을 사용하면 다른 복제된 사용자 접미어에서 받은 최신 백업을 보다 신속하게 복원할 수 있습니다. 249 페이지 "이진 복사를 사용하여 복제된 접미어 초기화"를 참조하십시오. 복사할 다른 사용자가 없을 경우, 가능하면 이전 항목에 설명된 것처럼 복제본을 다시 초기화하거나 다음 항목에 설명된 것처럼 복제본을 복원합니다.
- 사용자의 백업이 해당 공급자 중 **어느** 하나(허브 또는 마스터 복제본)에 있는 변경 로그 내용의 최대 수명보다 오래되지 않은 경우 이 백업을 사용하여 이 사용자를 복원할 수 있습니다. 사용자를 복원하면 해당 공급자는 자신의 변경 로그를 사용하여 백업이 저장된 이후에 처리된 모든 수정 사항을 사용자에 업데이트합니다.

다중 마스터 시나리오에서 마스터 복원

다중 마스터 복제의 경우 특정 마스터를 복원하는 동안 다른 마스터에서 변경 작업을 처리할 수 있습니다. 따라서 복원이 완료되면 새 마스터는 복원 데이터에 없는 새 업데이트도 받아야 합니다. 마스터 복원에 상당한 시간이 걸리면 보류 중인 업데이트 수도 증가합니다.

보류 중인 업데이트의 수렴을 허용하기 위해 새로 복원된 마스터는 복원 후의 클라이언트 작업 시 자동으로 읽기 전용 모드로 설정됩니다. 이 설정은 명령줄에서 LDIF 파일을 사용하여 데이터를 가져오거나 백업을 통해 이진 복사를 수행하여 마스터를 복원하는 경우에만 적용됩니다.

따라서 다중 마스터 구성의 마스터는 복원 후에 복제 업데이트를 처리하고 읽기 작업을 허용하지만 클라이언트의 모든 쓰기 요청에 대해서는 참조를 반환합니다.

업데이트를 허용하기 전에 새 마스터가 다른 마스터와 완전히 동기화되었는지 확인하려면 초기화된 마스터에서 업데이트를 수동으로 활성화합니다.

주 – 새 기능으로 인해 마스터 복제본에서 참조를 보내는 경우 쓰기 작업을 수행하려는 클라이언트는 구성된 홉 수 제한에 도달할 수 있습니다. 이 경우 클라이언트가 사용 가능한 마스터에 도달할 수 있도록 이 클라이언트의 홉 수 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 최적화합니다.

▼ 명령줄을 통해 업데이트 허용

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

다중 마스터 복제본 초기화 프로세스를 자동화하는 스크립트에서 사용 가능한 명령은 다음과 같습니다.

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 `insync`(1) 설명서 페이지를 참조하십시오.

2 업데이트를 허용합니다.

```
$ dsconf set-suffix-prop -h host -p port suffix-DN repl-accept-client-update-enabled:on
```

이 명령은 서버를 읽기-쓰기 모드로 자동으로 설정합니다.

재해 복구

재해 복구를 위해 디렉토리 서버를 백업하거나 복원하려면 다음 절차를 수행합니다.

▼ 재해 복구를 위한 백업을 만드는 방법

이 절차의 일부로, DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오. 해당 절차의 다른 부분은 명령줄에서만 수행할 수 있습니다.

1. **dsadn backup 또는 dsconf backup 명령을 사용하여 데이터베이스 파일 백업을 만듭니다.**
   197 페이지 "이진 백업"의 절차를 수행하고 백업 파일을 안전한 장소에 저장합니다.
2. ***instance-path*/config 구성 디렉토리를 안전한 장소에 복사합니다.**
3. ***instance-path*/config/schema 스키마 디렉토리를 안전한 장소에 복사합니다.**
4. ***instance-path*/alias 별칭 디렉토리를 안전한 장소에 복사합니다.**

▼ 재해 복구를 위해 복원하는 방법

이 절차의 일부로, DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오. 해당 절차의 다른 부분은 명령줄에서만 수행할 수 있습니다.

1. **호스트에 이전에 설치한 것과 동일한 버전의 디렉토리 서버를 설치합니다.**
2. **dsadm create 명령을 사용하여 서버 인스턴스를 만듭니다.**
   백업 당시에 사용했던 것과 동일한 인스턴스를 사용합니다. 60 페이지 "접미어 만들기"를 참조하십시오.
3. ***instance-path*/config 구성 디렉토리를 복원합니다.**
4. ***instance-path*/config/schema 스키마 디렉토리를 복원합니다.**
5. ***instance-path*/alias 별칭 디렉토리를 복원합니다.**
6. **복원된 서버의 구성이 올바른지 확인합니다.**
   예를 들어 디렉토리 구조와 플러그인 구성이 백업된 서버에서와 동일해야 합니다.
7. **dsconf restore 명령을 사용하여 데이터베이스 파일을 복원합니다.**
   201 페이지 "이진 복원"의 절차를 수행합니다.

◆ ◆ ◆ 9 장

9 디렉토리 서버 그룹, 역할 및 CoS

디렉토리의 계층적 데이터 구조에 제한을 받지 않고 자유롭게 사용자 항목을 관리하기 위해 그룹을 만들어 공통 속성 값을 공유해야 하는 경우가 있습니다. 디렉토리 서버는 그룹, 역할 및 서비스 클래스(CoS)를 통해 이러한 고급 항목 관리 기능을 제공합니다.

이 장은 다음 내용으로 구성되어 있습니다.

그룹, 역할 및 서비스 클래스 정보

그룹, 역할 및 CoS는 다음과 같이 정의됩니다.

- 그룹은 구성원 목록이나 구성원 필터로 다른 항목의 이름을 지정하는 항목입니다. 구성원 목록으로 구성되는 그룹에 대해 디렉토리 서버는 각 사용자 항목의 `isMemberOf` 속성 값을 생성합니다. 따라서 사용자 항목의 `isMemberOf` 속성은 해당 항목이 속하는 모든 그룹을 표시합니다.
- 역할도 특정 역할의 각 구성원에 `nsrole` 속성을 생성하는 메커니즘을 통해 그룹과 동일한 기능을 제공하지만, 훨씬 더 기능이 다양합니다.
- CoS는 계산된 속성을 생성함으로써 각 항목에 속성을 저장할 필요 없이 여러 항목이 공통 속성 값을 공유할 수 있게 합니다.

  `isMemberOf` 속성을 사용하여 정적 그룹의 모든 구성원이 계산된 공통 속성 값으로부터 자동으로 상속되도록 만들 수 없습니다.

디렉토리 서버는 역할 값과 그룹 및 CoS에서 계산된 속성을 기반으로 검색을 수행하는 기능을 제공합니다. 작업에 사용되는 필터 문자열은 `nsRole` 속성이나 CoS 정의에 따라 생성되는 모든 속성을 포함할 수 있습니다. 또한 필터 문자열을 사용하여 이 속성 값에

대한 비교 작업을 수행할 수 있습니다. 그러나 계산된 CoS 속성은 색인화할 수 없습니다. 따라서 CoS에서 생성된 속성을 포함하는 모든 검색에서는 시간과 메모리 자원을 많이 소모할 수 있습니다.

역할, 그룹 및 서비스 클래스에서 제공하는 기능을 최대한 활용하려면 디렉토리 배포의 계획 단계에서 그룹화 전략을 결정합니다. 이러한 기능과 해당 기능이 토폴로지를 간소화하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 "Grouping Directory Data and Managing Attributes"를 참조하십시오.

역할 및 그룹 작업 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 8 장, "Directory Server Groups and Roles"를 참조하십시오. CoS에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 9 장, "Directory Server Class of Service"를 참조하십시오.

그룹 관리

그룹을 사용하면 관리 작업의 편의를 위해 항목을 연결할 수 있습니다. 예를 들어 그룹을 사용하여 액세스 제어 지침(ACI)을 쉽게 정의할 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.

그룹에는 정적 그룹과 동적 그룹의 두 유형이 있습니다.

- **정적 그룹.**정적 그룹을 정의하는 항목은 `groupOfNames` 또는 `groupOfUniqueNames` 객체 클래스에서 상속됩니다. 그룹 구성원은 DN에 따라 `member` 또는 `uniqueMember` 속성의 여러 값으로 나열됩니다.

  또는 정적 그룹의 `isMemberOf` 속성을 사용할 수 있습니다. `isMemberOf` 속성은 검색을 시작할 때 계산되어 사용자 항목에 추가됩니다. 그런 다음 검색이 완료되면 다시 제거됩니다. 이 기능을 사용하면 그룹을 쉽게 관리하고 빠르게 읽기 액세스할 수 있습니다.
- **동적 그룹.**동적 그룹을 정의하는 항목은 `groupOfURLs` 객체 클래스로부터 상속됩니다. 그룹 구성원은 여러 값을 갖는 `memberURL` 속성에 지정된 하나 이상의 필터로 정의됩니다. 동적 그룹의 구성원은 필터가 평가될 때마다 해당 필터 중 하나에 일치하는 항목입니다.

▼ 새 정적 그룹을 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1. **ldapmodify 명령을 사용하여 새 정적 그룹을 만듭니다.**

   예를 들어 System Administrators라는 새 정적 그룹을 만들고 일부 구성원을 추가하려면 다음 명령을 사용할 수 있습니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=System Administrators, ou=Groups, dc=example,dc=com
changetype: add
cn: System Administrators
objectclass: top
objectclass: groupOfNames
ou: Groups
member: uid=kvaughan, ou=People, dc=example,dc=com
member: uid=rdaugherty, ou=People, dc=example,dc=com
member: uid=hmiller, ou=People, dc=example,dc=com
```

2. **새 그룹이 만들어졌고 구성원이 추가되었는지 확인합니다.**

   예를 들어 Kirsten Vaughan이 새 System Administrators 그룹에 있는지 확인하려면 다음과 같이 입력합니다.

```
$ ldapsearch -b "dc=example,dc=com" uid=kvaughan isMemberOf
uid=kvaughan,ou=People,dc=example,dc=com
isMemberOf: cn=System Administrators, ou=Groups, dc=example,dc=com
isMemberOf: cn=HR Managers,ou=groups,dc=example,dc=com
```

▼ 새 동적 그룹을 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **ldapmodify 명령을 사용하여 새 동적 그룹을 만듭니다.**

  예를 들어 "3rd Floor"라는 새 동적 그룹을 만들고 방 번호가 3으로 시작하는 모든 구성원을 포함시키려면 다음 명령을 사용할 수 있습니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
dn: cn=3rd Floor, ou=Groups, dc=example,dc=com
changetype: add
cn: 3rd Floor
objectclass: top
objectclass: groupOfUrls
ou: Groups
memberURL: ldap:///dc=example,dc=com??sub?(roomnumber=3*)
```

역할 관리

역할은 응용 프로그램을 보다 쉽고 효율적으로 사용하기 위해 설계한 대체 그룹화 메커니즘입니다. 그룹과 마찬가지로 역할이 정의되어 있고 관리되는 동안, 각 구성원 항목에 생성된 역할 속성은 자동으로 항목의 역할을 나타냅니다. 예를 들어 응용 프로그램은 그룹을 선택하여 구성원 목록을 탐색할 필요 없이 항목의 역할을 읽을 수 있습니다.

기본적으로 역할의 범위는 해당 역할이 정의된 하위 트리로 제한됩니다. 그러나 중첩된 역할의 범위를 확장할 수 있습니다. 해당 범위에서 다른 하위 트리에 있는 역할을 중첩시키고 디렉토리의 어느 곳에서나 구성원을 가질 수 있습니다. 자세한 내용은 217 페이지 "역할 범위를 확장하는 방법" 및 216 페이지 "중첩된 역할 정의 예"를 참조하십시오.

이 절에서는 역할을 안전하게 사용하는 방법과 명령줄에서 역할을 관리하는 방법에 대해 설명합니다.

역할을 안전하게 사용

역할을 안전하게 사용하려면 액세스 제어 지침(ACI)을 설정하여 해당 속성을 보호해야 합니다. 예를 들어 사용자 A가 관리된 역할(MR)을 소유합니다. 관리된 역할은 정적 그룹에 해당하며 `nsRoleDN` 속성을 항목에 추가하여 각 구성원 항목에 역할을 명시적으로 할당합니다. 명령줄에서 계정 비활성화를 사용하여 MR 역할을 잠갔습니다. 즉, `nsAccountLock` 속성이 사용자 A에 대해 "true"로 계산되기 때문에 해당 사용자는 서버에 바인드할 수 없습니다. 그러나 사용자가 이미 바인드되어 있고 지금 MR 역할을 통해 잠긴다는 알림을 받았다고 가정합니다. 사용자가 `nsRoleDN` 속성에 쓰기 액세스하지 못하도록 금지하는 ACI가 없는 경우 사용자는 `nsRoleDN` 속성을 자신의 항목에서 제거하여 직접 잠금 해제할 수 있습니다.

사용자가 `nsRoleDN` 속성을 제거하지 못하도록 금지하려면 ACI를 적용해야 합니다. 역할을 필터링한 상태에서 사용자가 속성을 수정하여 필터링된 역할을 사용하지 못하도록 필터 부분을 보호해야 합니다. 사용자는 필터링된 역할에 사용되는 속성을 추가, 삭제 또는 수정할 수 없습니다. 이와 동일한 방법으로, 필터 속성 값이 계산되는 경우 필터 속성 값을 수정할 수 있는 모든 속성을 보호해야 합니다. 중첩된 역할은 필터링된 역할과 관리된 역할을 포함할 수 있으므로 중첩된 역할에 포함되는 각 역할에 대해서는 이전에 설명한 내용을 고려해야 합니다.

보안을 위한 ACI 설정에 대한 자세한 내용은 6 장을 참조하십시오.

명령줄에서 역할 관리

역할은 디렉토리 어드민 관리자가 명령줄 유틸리티를 통해 액세스할 수 있는 항목에 정의됩니다. 역할을 만들고 나면 다음과 같이 구성원을 역할에 지정합니다.

- 관리된 역할의 구성원에는 해당 항목의 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 예에서는 역할 유형별 추가 객체 클래스 및 관련 속성을 보여줍니다.

관리된 역할 정의 예

모든 마케팅 직원에 대한 역할을 만들려면 다음 ldapmodify 명령을 사용합니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
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.

Bob이라는 마케팅 직원의 항목을 다음과 같이 업데이트하여 역할을 할당합니다.

```
$ ldapmodify -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
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
```

nsRoleDN 속성은 항목이 관리된 역할의 구성원임을 나타냅니다. 관리된 역할은 역할 정의의 DN으로 식별됩니다. 사용자가 자신의 nsRoleDN 속성을 수정하도록 허용하고 nsManagedDisabledRole을 추가하거나 제거하지 못하도록 금지하려면 다음 ACI를 추가합니다.

```
aci: (targetattr="nsRoleDN")(targattrfilters="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 다음 `ldapmodify` 명령을 사용합니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
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 `ou=sales` 하위 항목이 있는 조직에서 모든 직원을 찾는 필터를 지정합니다. 예를 들면 다음과 같습니다.

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"
dn: cn=Carla Fuentes,ou=sales,ou=People,dc=example,dc=comcn: Carla Fuentes
isManager: TRUE...
nsRole: cn=ManagerFilter,ou=sales,ou=People,
dc=example,dc=com
```

주 – 필터링된 역할의 필터 문자열은 CoS 메커니즘에 의해 생성된 계산된 속성을 제외한 모든 속성에 기반을 둘 수 있습니다.

필터링된 역할 구성원이 사용자 항목인 경우 해당 사용자가 자신을 역할에 추가하거나 역할에서 제거하는 기능을 제한할 수도 있습니다. ACI를 사용하여 필터링된 속성을 보호합니다.

중첩된 역할 정의 예

중첩된 역할 내에서 중첩되는 역할은 `nsRoleDN` 속성을 사용하여 지정됩니다. 이전 예에서 만든 역할의 마케팅 직원과 영업 책임자 구성원을 모두 포함하는 역할을 만들려면 다음 명령을 사용합니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
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
nsRoleScopeDN: 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 있는 하위 트리와 nsRoleScopeDN 속성 값 아래의 하위 트리로 구성된 기본 범위가 포함됩니다. 이 경우 ManagerFilter는 ou=sales,ou=People,dc=example,dc=com 하위 트리에 있습니다. 이 하위 트리를 범위에 추가해야 합니다.

역할 범위 확장

디렉토리 서버는 역할 정의 항목의 하위 트리를 벗어나 역할 범위를 확장하도록 허용하는 속성을 제공합니다. 이 단일 값 속성 nsRoleScopeDN에는 기존 역할에 추가할 범위의 DN이 포함됩니다. nsRoleScopeDN 속성은 중첩된 역할에만 추가할 수 있습니다. 216 페이지 “중첩된 역할 정의 예”를 참조하십시오.

▼ 역할 범위를 확장하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

nsRoleScopeDN 속성을 사용하면 다른 하위 트리의 항목을 포함하도록 하위 트리의 역할 범위를 확장할 수 있습니다. 예를 들어 example.com 디렉토리 트리에 o=eng,dc=example,dc=com(엔지니어링 하위 트리) 및 o=sales,dc=example,dc=com(영업 하위 트리)의 두 개의 기본 하위 트리가 있다고 가정합니다.엔지니어링 하위 트리의 사용자는 영업 하위 트리의 역할(SalesAppManagedRole)을 따르는 영업 응용 프로그램에 액세스해야 합니다. 역할 범위를 확장하려면 다음을 수행합니다.

1 엔지니어링 하위 트리에 있는 사용자에 대한 역할을 만듭니다.

예를 들어 EngineerManagedRole 역할을 만듭니다. 이 예에서는 관리된 역할을 사용하지만 필터링된 역할이나 중첩된 역할을 사용할 수도 있습니다.

2 **예를 들어 영업 하위 트리에서 중첩된 역할 `SalesAppPlusEngNestedRole`을 사용하여 새로 만든 `EngineerManagedRole`과 초기 `SalesAppManagedRole`을 수용합니다.**

3 **추가할 엔지니어링 하위 트리 범위의 DN(이 경우 `o=eng,dc=example,dc=com`)을 사용하여 `nsRoleScopeDN` 속성을 `SalesAppPlusEngNestedRole`에 추가합니다.**

엔지니어링 사용자가 `SalesAppPlusEngNestedRole` 역할과 영업 응용 프로그램에 차례로 액세스할 수 있도록 필요한 권한을 부여해야 합니다. 또한, 역할의 전체 범위를 복제해야 합니다.

주 – 확장된 범위를 중첩된 역할로 제한하면 한 도메인에서 이전에 역할을 관리한 관리자는 다른 도메인에 이미 있는 역할에 대한 사용 권한만 가집니다. 관리자는 다른 도메인에서 임의의 역할을 만들 수 없습니다.

서비스 클래스

서비스 클래스(CoS) 메커니즘은 클라이언트 응용 프로그램에 대해 항목이 검색되면서 계산된 속성을 생성하여 항목 관리를 간소화하고 필요한 저장 공간을 줄여줍니다. CoS 메커니즘을 사용하면 항목 간에 속성을 공유할 수 있습니다. 또한 그룹 및 역할과 마찬가지로 CoS는 도우미 항목을 사용합니다.

배포 시 CoS를 사용하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 “Managing Attributes With Class of Service”를 참조하십시오.

디렉토리 서버에서 CoS가 구현되는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 9 장, “Directory Server Class of Service”를 참조하십시오.

주 – 모든 검색 작업으로 CoS에서 생성된 속성의 존재 여부를 테스트하거나 속성 값을 비교할 수 있습니다. 계산된 속성의 이름은 필터링된 역할에 사용된 내부 필터를 제외하고 클라이언트 검색 작업의 모든 필터 문자열에 사용할 수 있습니다.

안전하게 CoS 사용

다음 절에서는 각 CoS 항목의 데이터 읽기 및 쓰기 보호에 대한 일반 원칙에 대해 설명합니다. 개별 액세스 제어 지침(ACI)을 정의하는 자세한 절차는 6 장를 참조하십시오.

CoS 정의 항목 보호

CoS 정의 항목에는 생성된 속성 값이 포함되어 있지 않지만 해당 값을 찾을 수 있는 정보를 제공합니다. CoS 정의 항목 읽기는 값이 포함된 템플리트 항목을 찾는 방법을 보여줍니다. 이 항목에 대한 쓰기는 계산된 속성이 생성되는 방법을 수정합니다.

따라서 CoS 정의 항목에 대한 읽기 ACI와 쓰기 ACI를 모두 정의해야 합니다.

CoS 템플리트 항목 보호

CoS 템플리트 항목에는 생성된 CoS 속성 값이 포함되어 있습니다. 따라서 최소한 읽기와 업데이트 모두에 대해 ACI로 템플리트의 CoS 속성을 보호해야 합니다.

- **포인터** *CoS*의 경우 단일 템플리트 항목의 이름을 바꾸면 안 됩니다. 대부분의 경우 전체 템플리트 항목을 보호하는 것이 가장 간단합니다.
- **클래식** *CoS*의 경우 모든 템플리트 항목은 공통 부모가 정의 항목에 지정되어 있습니다. 이 부모 항목에 템플리트만 저장된 경우 부모 항목에 대한 액세스 제어를 통해 템플리트를 보호합니다. 그러나 부모 아래의 다른 항목에 액세스해야 하는 경우에는 템플리트 항목을 개별적으로 보호해야 합니다.
- **간접** *CoS*의 경우 액세스해야 하는 사용자 항목을 포함하여 디렉토리의 모든 항목이 템플리트가 될 수 있습니다. 필요에 따라 디렉토리를 통해 CoS 속성에 대한 액세스를 제어하거나 템플리트로 사용되는 각 항목에서 CoS 속성이 보안되는지 확인합니다.

CoS의 대상 항목 보호

계산된 CoS 속성이 생성되는 CoS 정의 범위 내의 모든 항목은 값 계산에도 사용됩니다.

CoS 속성이 대상 항목에 이미 있는 경우 CoS 메커니즘에서는 기본적으로 이 값을 무시하지 않습니다. 이 동작을 원하지 않는 경우 CoS를 정의하여 대상 항목을 무시하거나, 잠재적인 모든 대상 항목에서 CoS 속성을 보호합니다.

간접 CoS와 클래식 CoS는 모두 대상 항목의 지정자 속성을 사용합니다. 이 속성은 사용할 템플리트 항목의 DN 또는 RDN을 지정합니다. ACI를 사용하여 CoS 범위 전체에서 이 속성을 전역적으로 보호하거나 각 대상 항목에서 필요에 따라 개별적으로 보호해야 합니다.

기타 종속성 보호

생성된 다른 CoS 속성 및 역할을 기준으로 계산된 CoS 속성을 정의할 수 있습니다. 계산된 CoS 속성을 보호하려면 이러한 종속성을 파악하여 보호해야 합니다.

예를 들어 대상 항목의 CoS 지정자 속성이 `nsRole`일 수 있습니다. 따라서 ACI를 사용하여 역할 정의도 함께 보호해야 합니다.

일반적으로 계산된 속성 값 계산에 포함되는 모든 속성이나 항목은 읽기 및 쓰기 액세스 제어를 위한 ACI가 있어야 합니다. 따라서 복잡한 종속성을 체계적으로 계획하거나 간소화하여 이후의 액세스 제어 구현 시의 복잡성을 줄여야 합니다. 다른 계산된 속성에 대한 종속성을 최소로 유지하면 디렉토리 성능이 향상되고 유지 관리가 쉬워집니다.

명령줄에서 CoS 관리

모든 구성 정보와 템플리트 데이터는 디렉토리 항목으로 저장되기 때문에 LDAP 명령줄 도구를 사용하여 CoS 정의를 구성 및 관리할 수 있습니다. 이 절에서는 명령줄에서 CoS 정의 항목과 CoS 템플리트 항목을 만드는 방법에 대해 설명합니다.

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 9–1 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:` *attributeName override merge*

간접 CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosIndirectDefinition`
	`cosIndirectSpecifier:` *attributeName*
	`cosAttribute:` *attributeName override merge*

클래식 CoS	`objectclass: top`
	`objectclass: LDAPsubentry`
	`objectclass: cosSuperDefinition`
	`objectclass: cosClassicDefinition`
	`cosTemplateDN:` *DN*
	`cosSpecifier:` *attributeName*
	`cosAttribute:` *attributeName override merge*

모든 경우에 cosAttribute는 여러 값을 가집니다. 각 값은 CoS 메커니즘으로 생성되는 속성을 정의합니다.

CoS 정의 항목에서 다음 속성을 사용할 수 있습니다. 각 속성에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Man Page Reference**의 개별 속성을 참조하십시오.

표 9–2 CoS 정의 항목 속성

속성	CoS 정의 항목에서의 용도
cosAttribute *attributeName override merge*	값을 생성할 계산된 속성 이름을 정의합니다. 이 속성은 여러 값을 가지며, 각각의 값은 템플리트에서 값이 생성될 속성 이름을 나타냅니다. *override* 및 *merge* 한정자는 표 아래에 설명된 특별한 경우에 CoS 속성 값이 계산되는 방법을 지정합니다. *attributeName*에는 하위 유형이 포함될 수 없습니다. 하위 유형이 있는 속성 이름은 무시되지만 cosAttribute의 다른 값은 정상적으로 처리됩니다.
cosIndirectSpecifier *attributeName*	간접 CoS에서 값이 사용되어 템플리트 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 *attributeName*은 많은 템플리트를 지정하기 위해 여러 값을 가질 수 있습니다.
cosSpecifier *attributeName*	클래식 CoS에서 값이 사용되어 템플리트 항목을 식별하는 대상 항목의 속성 이름을 정의합니다. 이름이 지정된 속성을 지정자라고 하며 템플리트 항목의 RDN에 있는 문자열이 포함되어야 합니다. 이 속성은 한 개의 값을 갖지만 *attributeName*은 많은 템플리트를 지정하기 위해 여러 값을 가질 수 있습니다.
cosTemplateDN *DN*	포인터 CoS 정의에 템플리트 항목의 전체 DN을 제공하거나 클래식 CoS 정의에 템플리트 항목의 기본 DN을 제공합니다. 한 개의 값을 갖습니다.

주 – isMemberOf 속성을 CosSpecifier로 사용하여 정적 그룹의 모든 구성원이 계산된 공통 속성 값으로부터 자동으로 상속되도록 만들 수 없습니다.

cosAttribute 속성은 CoS 속성 이름 뒤에 두 개의 한정자(override 한정자 및 merge 한정자)를 허용합니다.

override 한정자는 CoS에 의해 동적으로 생성되는 속성이 항목에 이미 있는 경우의 동작을 설명합니다. *override* 한정자는 다음 중 하나입니다.

- default(한정자 없음) - 항목에 저장된 실제 속성 값이 계산된 속성과 같은 유형이면 서버에서 실제 속성 값을 무시하지 않음을 나타냅니다.
- override - 값이 항목과 함께 저장되더라도 서버는 항상 CoS에 의해 생성된 값을 반환함을 나타냅니다.

- operational - 속성이 검색에서 명시적으로 요청되는 경우에만 반환됨을 나타냅니다. 작동 가능 속성은 스키마 검사를 통과하지 않아도 반환될 수 있습니다. operational 한정자는 override 한정자와 동일하게 작동합니다.

  스키마에서 작동 가능으로 정의된 속성만 작동 가능으로 설정할 수 있습니다. 예를 들어 CoS에서 description 속성 값을 생성한 경우 description 속성은 스키마에서 작동 가능으로 표시되지 않았으므로 operational 한정자를 사용할 수 없습니다.

merge 한정자가 없거나 merge-schemes입니다. 이 한정자는 계산된 CoS 속성이 여러 템플리트나 여러 CoS 정의에서 여러 값을 가질 수 있도록 허용합니다. 자세한 내용은 222 페이지 "여러 값을 갖는 CoS 속성"을 참조하십시오.

실제 속성 값 무시

아래 명령을 실행하여 override 한정자가 있는 포인터 CoS 정의 항목을 만들 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 해당 항목이 postalCode 속성 값을 생성하는 템플리트 항목 cn=exampleUS, cn=data과 관련되어 있음을 나타냅니다. 또한 override 한정자는 해당 속성이 대상 항목에 있는 경우 postalCode 속성 값보다 우선함을 나타냅니다.

주 – CoS 속성이 operational 또는 override 한정자로 정의되는 경우 CoS 범위 내의 항목에 있는 해당 항목의 "실제" 값에 대해 쓰기 작업을 수행할 수 없습니다.

여러 값을 갖는 CoS 속성

merge-schemes 한정자를 지정하면 생성된 CoS 속성은 다음 두 가지 방법으로 여러 값을 가질 수 있습니다.

- 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 cosAttribute에 동일한 속성 이름이 포함된 여러 개의 CoS 정의 항목이 있을 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, 템플리트 항목의 cosPriority 속성을 사용하여 생성된 속성에 대해 모든 템플리트에 지정되는 한 개의 값을 결정합니다. 이 시나리오에 대해서는 다음 절에서 설명합니다.

merge-schemes 한정자는 대상에서 정의된 "실제" 값과 템플리트에서 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 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 CoS 정의가 있거나 여러 값을 가진 한정자가 있지만 merge-schemes 한정자가 없는 경우 디렉토리 서버는 우선 순위 속성을 사용하여 계산된 속성의 단일 값을 정의하는 단일 템플리트를 선택합니다.

cosPriority 속성은 모든 템플리트 중에서 특정 템플리트의 전역 우선 순위를 나타냅니다. 우선 순위 0(제로)이 가장 높은 우선 순위입니다. cosPriority 속성이 없는 템플리트는 가장 낮은 우선 순위로 간주됩니다. 두 개 이상의 템플리트가 속성 값을 제공하지만 우선 순위가 같은 경우(또는 없는 경우)에는 임의로 값이 선택됩니다.

merge-schemes 한정자를 사용하는 경우 템플리트 우선 순위는 고려되지 않습니다. 병합하는 경우에는 정의된 우선 순위에 관계 없이 해당되는 모든 템플리트가 값을 정의합니다. 다음 절에 설명된 것처럼 cosPriority 속성은 CoS 템플리트 항목에 정의됩니다.

주 – cosPriority 속성에 음수 값은 지정할 수 없습니다. 또한 간접 CoS에서 생성된 속성은 우선 순위를 지원하지 않습니다. 간접 CoS 정의의 템플리트 항목에는 cosPriority를 사용하지 마십시오.

명령줄에서 CoS 템플리트 항목 작성

포인터 CoS 또는 클래식 CoS를 사용할 경우 템플리트 항목에 LDAPsubentry 및 cosTemplate 객체 클래스가 포함되어 있습니다. 이 항목은 특별히 CoS 정의에 대해 작성되어야 합니다. CoS 템플리트 항목을 LDAPsubentry 객체 클래스의 한 인스턴스로 만들면 구성 항목에 영향을 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 계산된 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

다음 명령은 cosPointerDefinition 객체 클래스를 가진 포인터 CoS 정의 항목을 만듭니다. 이 정의 항목은 이전 절의 예에 설명된 CoS 템플리트 항목을 사용하여 ou=People,dc=example,dc=com 트리의 모든 항목에 대한 공통 우편 번호를 공유합니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
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 host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -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 속성의 속성 이름을 지정하여 각 대상에 고유한 템플리트를 찾습니다. 간접 CoS의 템플리트 항목은 다른 사용자 항목을 포함하여

디렉토리에 있는 모든 항목이 될 수 있습니다. 이 예에서 간접 CoS는 대상 항목의 manager 속성을 사용하여 CoS 템플리트 항목을 식별합니다. 템플리트 항목은 관리자의 사용자 항목입니다. 관리자의 사용자 항목은 생성할 속성 값을 포함합니다. 이 경우 값은 departmentNumber의 값입니다.

다음 명령은 cosIndirectDefinition 객체 클래스를 포함하는 간접 CoS 정의 항목을 만듭니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
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 host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
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 항목에 대해 계산됩니다. 그러나 departmentNumber 속성은 대상 항목의 일부로 반환됩니다. 예를 들어 Babs Jensen의 관리자가 Carla Fuentes로 정의된 경우 해당 부서 번호는 다음과 같습니다.

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -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 CoS로 우편 주소를 생성하는 방법을 보여줍니다. 생성된 값은 CoS 정의의 cosTemplateDN 및 cosSpecifier 속성 값 조합으로 배치되는 템플리트 항목에 지정됩니다. 다음 명령은 cosClassicDefinition 객체 클래스를 사용하여 정의 항목을 만듭니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
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 실행하여 각 건물의 우편 주소를 제공하는 템플리트 항목을 만듭니다.

```
dn: cn=B07,ou=People,dc=example,dc=com
objectclass: top
objectclass: LDAPsubentry
objectclass: extensibleobject
objectclass: cosTemplate
postalAddres: 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 템플리트 항목을 검색합니다. 이 예에서 Babs Jensen이 건물 B07에 지정되어 있으면 우편 주소는 다음과 같이 생성됩니다.

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -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

항목의 역할을 기반으로 항목에 대한 속성 값을 생성하는 클래식 CoS 체계를 만들 수 있습니다. 예를 들어 역할 기반의 속성을 사용하여 서버에서 항목별 제한을 조회하도록 설정할 수 있습니다.

역할 기반의 속성을 만들려면 nsRole 속성을 클래식 CoS의 CoS 정의 항목에 있는 cosSpecifier로 사용합니다. nsRole 속성은 여러 값을 가질 수 있으므로 두 개 이상의 템플리트 항목이 있는 CoS 체계를 정의할 수 있습니다. 사용할 템플리트 항목을 명확히 지정하기 위해 CoS 템플리트 항목에 cosPriority 속성을 추가할 수 있습니다.

예를 들어 관리자 역할의 구성원이 표준 메일함 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 만들어집니다.

```
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 템플리트 이름은 cosTemplateDn과 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 템플리트 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 계산된 속성이 지정됩니다.

```
$ ldapsearch -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -\
 -b "ou=People,dc=example,dc=com" -s sub "(cn=*Fuentes)"
dn: cn=Carla Fuentes,ou=People,dc=example,dc=comcn: 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

디렉토리 서버에서는 CoS 플러그인의 특정 특성을 모니터할 수 있습니다. CoS 모니터링 속성은 `cn=monitor,cn=Class of Service,cn=plugins,cn=config` 항목 아래에 저장됩니다. 이 항목 아래의 각 속성과 해당 속성이 제공하는 정보에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Man Page Reference**를 참조하십시오.

CoS 로깅 설정

디렉토리 서버는 해당하는 여러 정의 항목을 임의로 구별할 경우 경고 메시지를 기록합니다. 경고 메시지의 형식은 다음과 같습니다.

```
Definition /defDN1/ and definition /defDN2/ compete to provide attribute
 '/type/' at priority /level/
```

서버에서 해당하는 여러 정의 항목을 임의로 구별하려는 경우 정보 메시지를 기록하도록 디렉토리 서버를 구성할 수도 있습니다. 이렇게 하려면 플러그인의 메시지를 포함할 오류 로그를 설정합니다.

주 – 추가 로그 수준을 설정하면 로깅 부하가 높아질 수 있으므로 생산 서버에서는 로깅을 설정하지 않을 수 있습니다.

정보 메시지의 내용은 다음과 같습니다.

```
Definition /defDN1/ and definition /defDN2/ potentially compete
to provide attribute '/type/' at priority /level/
```

그런 다음, 정의 항목에서 CoS 우선 순위를 적절하게 설정하여 해당 CoS의 모호성을 해결할지를 선택할 수 있습니다.

참조 무결성 유지

참조 무결성은 항목 간의 관계가 유지되는지를 확인하는 플러그인 메커니즘입니다. 그룹 구성원의 속성과 같은 일부 속성 유형에는 다른 항목의 DN이 포함되어 있습니다. 참조 무결성을 사용하면 항목을 제거할 때 해당 DN이 포함된 모든 속성도 함께 제거됩니다.

예를 들어 사용자 항목을 디렉토리에서 제거하고 참조 무결성을 사용 가능하게 하면 사용자가 구성원인 모든 그룹에서 해당 사용자가 제거됩니다. 참조 무결성을 사용하지 않으면 관리자가 수동으로 이 사용자를 그룹에서 제거해야 합니다. 이 기능은 사용자 및 그룹 관리를 위해 디렉토리를 사용하는 디렉토리 서버와 다른 Sun Java System 제품을 통합하는 경우에 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 이동할 때마다 작업이 참조 무결성 로그 파일에 기록됩니다.

instance-path`/logs/referint`

업데이트 간격이라는 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.

참조 무결성 플러그인의 기본 구성이 활성화되면 삭제, 이름 바꾸기 또는 이동 작업을 수행한 후에 즉시 `member`, `uniquemember`, `owner`, `seeAlso` 및 `nsroledn` 속성에 대한 무결성 업데이트를 수행합니다. 하지만 사용자 요구에 맞게 참조 무결성 플러그인의 동작을 구성할 수 있습니다. 다음 동작을 구성할 수 있습니다.

- 참조 무결성 업데이트를 다른 파일에 기록합니다.
- 업데이트 간격을 수정합니다.

  참조 무결성 업데이트로 인한 시스템 영향을 줄이려면 업데이트 간격을 늘리는 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 플러그인을 구성하는 방법

주 – 참조 무결성 플러그인에 사용되는 모든 데이터베이스의 모든 속성을 색인화해야 합니다. 모든 데이터베이스 구성에 색인을 만들어야 합니다. 레트로 변경 로그가 활성화되면 `cn=changelog` 접미어를 색인화해야 합니다. 자세한 내용은 12 장를 참조하십시오.

복제 환경에서 참조 무결성 플러그인을 사용하는 경우 다음과 같은 몇 가지 제한 사항이 있습니다. 제한 목록은 254 페이지 "복제 및 참조 무결성"을 참조하십시오.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **모든 복제본이 구성되고 모든 복제 계약이 정의되어 있는지 확인합니다.**
2. **참조 무결성을 유지할 속성 집합과 마스터 서버에서 사용할 업데이트 간격을 결정합니다.**
3. **모든 마스터 서버에서 동일한 속성 집합과 업데이트 간격을 사용하여 참조 무결성 플러그인을 활성화합니다.**
   - 참조 무결성을 위한 속성을 정의하려면 다음 명령을 사용합니다.

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-attr:attribute-name \
      ref-integrity-attr:attribute-name
     ```
   - 참조 무결성 속성을 기존 속성 목록에 추가하려면 이 명령을 사용합니다.

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-attr+:attribute-name
     ```
   - 참조 무결성 업데이트 간격을 정의하려면 다음 명령을 사용합니다.

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-check-delay:duration
     ```
   - 참조 무결성을 활성화하려면 다음 명령을 사용합니다.

     ```
     $ dsconf set-server-prop -h host -p port ref-integrity-enabled:on
     ```
4. **모든 사용자 서버에서 참조 무결성 플러그인을 비활성화합니다.**

◆ ◆ ◆ 10장

10 디렉토리 서버 복제

복제는 디렉토리 내용을 디렉토리 서버에서 하나 이상의 다른 디렉토리 서버로 자동으로 복사하는 메커니즘입니다. 모든 쓰기 작업은 다른 디렉토리 서버에 자동으로 미러링됩니다. 복제 개념, 복제 시나리오 및 사용자 디렉토리 배포에 대한 복제 계획 방법 등에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**를 참조하십시오.

복제 토폴로지에서 일반적으로 한 서버의 접미어는 해당 서버의 다른 접미어에 복제되거나 다른 접미어로부터 복제됩니다. 따라서 복제본, 복제된 접미어, 복제된 서버 등의 용어를 같은 의미로 사용할 수 있습니다.

이 장에서는 명령줄에서 다양한 복제 시나리오를 설정하는 작업에 대해 설명하며, 다음 내용으로 구성되어 있습니다.

- 269 페이지 “복제 상태 가져오기”
- 271 페이지 “일반적인 복제 충돌 해결”

복제 배포 계획

마스터 수에 제한 없이 복제 배포를 구성할 수 있습니다. 허브나 소비자를 배포에 포함시킬 필요는 없습니다. 허브 및 소비자에 대한 복제 구성 절차는 이 장에 포함되어 있지만 선택 사항입니다.

복제 구성을 시작하기 전에 조직에서 복제가 배포되는 방법을 정확하게 이해해야 합니다. **Sun Java System Directory Server Enterprise Edition 6.2 Reference**에 설명된 복제 개념을 이해해야 합니다. 또한 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**에 제공된 설계 지침에 따라 이후의 복제 구성을 신중하게 계획해야 합니다.

복제 구성 및 관리에 권장되는 인터페이스

복제를 구성 및 관리하는 가장 쉬운 방법은 디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)를 사용하는 것입니다. DSCC를 사용하여 복제를 자동으로 구성할 수 있습니다. 복제 토폴로지를 설정하는 데 필요한 자동화 수준을 선택할 수 있습니다. 예를 들어 복제 구성 중에 접미어를 초기화할지 여부를 선택할 수 있습니다. DSCC는 오류가 발생하지 않도록 검사 기능을 제공합니다. 또한 DSCC는 복제 토폴로지에 대한 그래픽 보기를 제공합니다.

DSCC 온라인 도움말에서는 DSCC를 사용하여 복제를 설정하는 절차에 대해 설명합니다.

주 – DSCC를 사용하여 복제를 구성할 수 없는 경우에는 이 장에 설명된 명령줄 절차만 사용하십시오.

구성 복제 단계 요약

233 페이지 “구성 복제 단계 요약”에서는 단일 접미어를 복제한다고 가정합니다. 두 개 이상의 접미어를 복제하는 경우 접미어를 각 서버에 병렬로 구성할 수 있습니다. 즉, 각 단계를 반복하여 여러 접미어에 대한 복제를 구성할 수 있습니다.

이 장의 나머지 부분에서는 복제를 구성하는 방법에 대한 자세한 지침을 제공합니다.

▼ 구성 복제 단계 요약

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

복제 토폴로지를 구성하려면 이 절차에 요약된 일반 단계를 수행합니다.

1 **전용 소비자 복제본이 있는 모든 서버에서 다음을 수행하십시오.**

   a. **사용자 복제 접미어에 대해 빈 접미어를 만듭니다.**
   235 페이지 "소비자 복제본에 대한 접미어를 만드는 방법"을 참조하십시오.

   b. **사용자 복제 접미어를 활성화합니다.**
   235 페이지 "소비자 복제본을 활성화하는 방법"을 참조하십시오.

   c. **(옵션) 고급 사용자 설정을 구성합니다.**
   235 페이지 "고급 사용자 구성을 수행하는 방법"을 참조하십시오.

2 **가능한 경우 허브 복제 접미어가 있는 모든 서버에서 다음을 수행하십시오.**

   a. **허브 복제 접미어에 대해 빈 접미어를 만듭니다.**
   237 페이지 "허브 복제본에 대한 접미어를 만드는 방법"을 참조하십시오.

   b. **허브 복제 접미어를 활성화합니다.**
   237 페이지 "허브 복제본을 활성화하는 방법"을 참조하십시오.

   c. **(옵션) 고급 허브 설정을 구성합니다.**
   237 페이지 "허브 복제본에 대한 변경 로그 설정을 수정하는 방법"을 참조하십시오.

3 **마스터 복제 접미어가 있는 모든 서버에서 다음을 수행하십시오.**

   a. **마스터 복제 접미어에 대해 접미어를 만듭니다.**
   238 페이지 "마스터 복제본에 대한 접미어를 만드는 방법"을 참조하십시오.

   b. **마스터 복제 접미어를 활성화합니다.**
   238 페이지 "마스터 복제본을 활성화하는 방법"을 참조하십시오.

   c. **(옵션) 고급 마스터 설정을 구성합니다.**
   239 페이지 "마스터 복제본에 대한 변경 로그 설정을 수정하는 방법"을 참조하십시오.

주 – 복제 계약을 만든 후 소비자 복제본을 즉시 초기화할 수 있도록 복제 계약을 만들기 전에 모든 복제본을 활성화해야 합니다. 사용자 초기화는 항상 복제 설정의 마지막 단계에 수행해야 합니다.

4 **복제 관리자 구성이 완료되었는지 확인합니다.**

- 기본 관리자를 사용하려면 모든 서버에서 기본 복제 관리자 비밀번호를 설정합니다. 241 페이지 "기본 복제 관리자 비밀번호를 변경하는 방법"을 참조하십시오.
- 기본값이 아닌 복제 관리자를 사용하려면 모든 서버에서 대체 복제 관리자 항목을 정의합니다. 239 페이지 "기본값이 아닌 복제 관리자 사용"을 참조하십시오.

5 **모든 마스터 복제본에 대한 복제 계약을 다음과 같이 만듭니다.**

a. **다중 마스터 토폴로지의 마스터 간**

b. **마스터 및 전용 사용자 간**

c. **마스터 및 허브 복제본 간**

241 페이지 "복제 계약 만들기 및 변경"을 참조하십시오.

6 **(옵션) 단편 복제를 사용하려면 지금 구성합니다.**
243 페이지 "단편 복제"를 참조하십시오.

7 **(옵션) 복제 우선 순위를 사용하려면 지금 구성합니다.**
244 페이지 "복제 우선 순위"를 참조하십시오.

8 **허브 복제본과 해당 사용자 간에 복제 계약을 구성합니다.**
241 페이지 "복제 계약 만들기 및 변경"을 참조하십시오.

9 **다중 마스터 복제의 경우 데이터의 원래 복사본이 포함된 동일한 마스터 복제본에서 모든 마스터를 초기화합니다.**
245 페이지 "복제본 초기화"를 참조하십시오.

10 **허브와 소비자 복제본을 초기화합니다.**
245 페이지 "복제본 초기화"를 참조하십시오.

전용 소비자에 대한 복제 활성화

전용 소비자는 복제된 접미어의 읽기 전용 복사본으로,전용 소비자는 복제 관리자로 바인드하는 서버로부터 업데이트를 받아 항목을 변경합니다. 사용자 서버 구성은 복제된 접미어를 저장할 빈 접미어 준비와 해당 접미어에 대한 복제 활성화로 구성됩니다. 선택 사항인 고급 구성에는 참조 설정, 지연 제거 변경 및 등록 정보 수정이 포함될 수 있습니다.

다음 절에서는 서버에서 전용 사용자 복제 접미어를 구성하는 방법에 대해 설명합니다. 전용 사용자 복제 접미어가 포함될 각 서버에서 모든 절차를 반복합니다.

▼ 소비자 복제본에 대한 접미어를 만드는 방법

- **빈 접미어가 없으면 마스터 복제본과 동일한 DN을 가진 소비자에 대해 빈 접미어를 만듭니다.**
  자세한 내용은 60 페이지 "접미어 만들기"를 참조하십시오.

주의 – 내용이 포함된 접미어가 있으면 마스터를 사용하여 복제된 접미어를 초기화할 때 이 내용이 손실됩니다.

▼ 소비자 복제본을 활성화하는 방법

빈 접미어를 만든 후 사용자 복제 접미어를 활성화해야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **사용자 복제 접미어를 활성화합니다.**

```
$ dsconf enable-repl -h host -p port consumer suffix-DN
```

예를 들면 다음과 같습니다.

```
$ dsconf enable-repl -h host1 -p 1389 consumer dc=example,dc=com
```

▼ 고급 사용자 구성을 수행하는 방법

고급 기능에 대한 사용자 복제 접미어를 구성하려면 지금 구성합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **SSL을 참조로 사용하려면 보안 참조를 설정합니다.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN referral-url:ldaps://servername:port
```

예를 들면 다음과 같습니다.

```
$ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
 referral-url:ldaps://server2:2389
```

복제 메커니즘은 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 바인드하려면 보안 *port* 번호를 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 소비자는 마스터 복제본이 아닌 LDAP URL에 대한 참조만 보내도록 할 수 있습니다. 예를 들어 클라이언트가 기본 포트가 아니라 마스터 서버의 보안 포트를 항상 참조한다고 가정합니다. 이러한 보안 포트에 대해 LDAP URL 목록을 만들고 해당 참조 사용을 위한 등록 정보를 설정합니다. 특정 마스터 또는 디렉토리 서버 프록시에서 모든 업데이트를 처리하도록 지정하려면 배타적 참조를 사용할 수도 있습니다.

2 **소비자에 대한 복제 지연 제거를 변경하려면 다음 명령을 사용합니다.**

```
$ dsconf set-suffix-prop -h host -p port suffix-DN repl-purge-delay:time
```

예를 들어 지연 제거를 2일로 설정하려면 다음을 입력합니다.

```
$ dsconf set-suffix-prop -h host1 -p 1389 edc=example,dc=com repl-purge-delay:2d
```

사용자 서버는 복제된 접미어 내용 업데이트에 대한 내부 정보를 저장하며, 지연 제거 매개 변수는 이 정보의 보관 기간을 지정합니다. 지연 제거는 소비자와 해당 마스터 간의 복제가 중단되고 정상적으로 복구될 수 있는 기간을 부분적으로 결정합니다. 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.

허브에서 복제 활성화

허브 복제본은 사용자 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 복제된 접미어를 저장할 빈 접미어 준비와 해당 접미어에 대한 복제 활성화로 구성됩니다. 선택 사항인 고급 구성에는 다른 복제 관리자 선택, 참조 설정, 지연 제거 설정, 변경 로그 매개 변수 수정 등이 있습니다.

다음 절에서는 단일 허브 서버를 구성하는 방법에 대해 설명합니다. 허브 복제 접미어가 포함될 각 서버에서 모든 절차를 반복합니다.

▼ 허브 복제본에 대한 접미어를 만드는 방법

- **빈 접미어가 없으면 마스터 복제본과 동일한 DN을 가진 허브 서버에 대해 빈 접미어를 만듭니다.**

  자세한 내용은 60 페이지 "접미어 만들기"를 참조하십시오.

  내용이 포함된 접미어가 있으면 마스터를 사용하여 복제된 접미어를 초기화할 때 이 내용이 손실됩니다.

▼ 허브 복제본을 활성화하는 방법

허브 복제본이 있는 경우 지금 활성화합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **허브 복제 접미어를 활성화합니다.**

  ```
  $ dsconf enable-repl -h host -p port hub suffix-DN
  ```

  예를 들면 다음과 같습니다.

  ```
  $ dsconf enable-repl -h host1 -p 1389 hub dc=example,dc=com
  ```

▼ 허브 복제본에 대한 변경 로그 설정을 수정하는 방법

고급 허브 구성의 경우 변경 로그와 관련된 매개 변수만 수정할 수 있습니다.
공급자로서의 허브 서버에는 변경 로그가 필요합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **허브에 대한 변경 로그 설정을 수정하려면 다음 명령 중 하나를 사용합니다.**

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-age:value
  ```

  ```
  $ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-entry-count:value
  ```

마스터 복제본에서 복제 활성화

마스터 복제본에는 데이터의 마스터 복사본이 포함되어 있으며 업데이트를 다른 모든 복제본으로 전파하기 전에 모든 수정 사항을 중앙 집중식으로 관리합니다. 마스터는 모든 변경 사항을 기록하고, 사용자 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, 마스터 복제본 활성화, 필요한 경우 고급 복제를 위한 구성 등의 단계로 이루어져 있습니다.

다음 절에서는 단일 마스터 서버를 구성하는 방법에 대해 설명합니다. 마스터 복제 접미어가 포함될 각 서버에서 모든 절차를 반복합니다.

▼ 마스터 복제본에 대한 접미어를 만드는 방법

- **마스터 서버에서 복제할 항목이 포함될 접미어를 선택하거나 새로 만듭니다.**

  자세한 내용은 60 페이지 "접미어 만들기"을 참조하십시오.

  다중 마스터 구성 및 활성화를 올바로 수행하려면 데이터가 있는 마스터 중 하나만 로드합니다. 그러면 다른 복제된 접미어의 데이터를 덮어쓰게 됩니다.

▼ 마스터 복제본을 활성화하는 방법

마스터에서 복제를 활성화할 경우 복제 아이디를 할당해야 합니다. 복제 아이디는 업데이트 문의 소유자를 구분하고 다중 마스터 복제에서 발생할 수 있는 충돌을 해결하는 데 사용됩니다. 따라서 복제 아이디는 이 접미어의 모든 마스터 복제본에 대해 고유해야 합니다. 복제 아이디는 설정하고 나면 변경할 수 없습니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **마스터 복제 접미어를 활성화합니다.**

  ```
  $ dsconf enable-repl -h host -p port -d ReplicaID master suffix-DN
  ```

  여기서 *ReplicaID*는 1부터 65534 사이의 정수입니다.

  예를 들어 복제 아이디가 1인 마스터 복제 접미어를 만들려면 다음 명령을 사용합니다.

  ```
  $ dsconf enable-repl -h host1 -p 1389 -d 1 master dc=example,dc=com
  ```

▼ 마스터 복제본에 대한 변경 로그 설정을 수정하는 방법

고급 마스터 구성의 경우 변경 로그 설정을 수정할 수 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **마스터에 대한 변경 로그 설정을 수정하려면 다음 명령 중 하나를 사용합니다.**

```
$ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-age:value

$ dsconf set-server-prop -h host -p port suffix-DN repl-cl-max-entry-count:value
```

복제 관리자 구성

이 절에서는 기본값이 아닌 복제 관리자를 구성하고 기본 복제 관리자 비밀번호를 설정하는 방법에 대해 설명합니다.

기본값이 아닌 복제 관리자 사용

복제 관리자는 공급자가 복제 업데이트를 보낼 때 사용자 서버에 바인드하는 데 사용할 사용자입니다. 업데이트를 받는 접미어가 포함된 모든 서버에 복제 관리자 항목이 적어도 한 개는 있어야 합니다.

디렉토리 서버에는 특히 단순 복제 시나리오에서 모든 서버에 사용할 수 있는 기본 복제 관리자 항목이 있습니다. `cn=replication manager,cn=replication,cn=config`. 복제 메커니즘은 자동으로 이 사용자를 사용하여 소비자 복제본을 구성함으로써 복제본의 배포를 간소화합니다.

더 복잡한 복제 시나리오에서는 여러 복제 관리자가 복제된 접미어마다 다른 비밀번호를 사용하도록 할 수 있습니다. 기존의 기본 복제 관리자를 하나 이상의 새 복제 관리자로 대체할 수 있습니다.

주의 – 복제 관리자의 DN과 비밀번호를 사용하여 서버에 대한 작업을 수행하거나 바인드하지 마십시오. 복제 관리자는 복제 메커니즘에서만 사용할 수 있습니다. 다른 용도로 사용하려면 복제본을 다시 초기화해야 할 수 있습니다.

디렉토리 관리자를 복제 관리자로 사용하지 마십시오. `cn=admin,cn=Administrators,cn=config` 항목이 다른 관리 작업에 사용되기 때문에 이 사용자나 관리자 그룹의 다른 사용자를 복제 관리자로 사용해서는 안 됩니다.

각 소비자에 대한 복제 관리자를 선택한 후 선택하거나 만든 복제 관리자 DN을 기억하십시오. 나중에 해당 공급자에서 소비자와의 복제 계약을 작성할 때 이 DN과 비밀번호를 사용해야 합니다.

▼ 기본값이 아닌 복제 관리자를 설정하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 모든 사용자(대상) 복제 접미어에서 새 복제 관리자와 비밀번호를 만듭니다.

```
$ ldapmodify -a -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn:"cn=new-replication-manager,cn=replication,cn=config"
objectclass: top
objectclass: person
userpassword:password
sn:new-replication-manager
```

예를 들면 다음과 같습니다.

```
$ ldapmodify -a -h host1 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn:"cn=ReplicationManager3,cn=replication,cn=config"
objectclass: top
objectclass: person
userpassword:secret
sn:ReplicationManager3
```

2 모든 사용자(대상) 복제 접미어에서 복제 관리자 바인드 DN을 설정합니다.

```
$ dsconf set-suffix-prop -h host -p port suffix-DN \
 repl-manager-bind-dn:"cn=new-replication-manager,cn=replication,cn=config"
```

예를 들면 다음과 같습니다.

```
$ dsconf set-suffix-prop -h host1 -p 1389 dc=example,dc=com \
 repl-manager-bind-dn:"cn=ReplicationManager3,cn=replication,cn=config"
```

3 모든 공급자(소스) 복제 접미어에서 만든 모든 복제 계약에 대해 복제 관리자 바인드 DN을 설정합니다.

a. 새 복제 관리자 비밀번호 설정을 위한 임시 파일을 만듭니다.

이 파일을 읽고 나중에 사용하기 위해 비밀번호를 저장합니다.

```
$ echo password > password-file
```

b. 업데이트를 수행할 때 복제 메커니즘에서 사용할 복제 관리자 바인드 DN 및 비밀번호를 설정합니다.

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN host:port \
 auth-bind-dn:"cn=new-replication-manager,cn=replication,cn=config" \
 auth-pwd-file:password-file
```

예를 들면 다음과 같습니다.

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 auth-bind-dn:"cn=ReplicationManager3,cn=replication,cn=config" \
 auth-pwd-file:pwd.txt
```

c. 임시 비밀번호 파일을 제거합니다.

```
$ rm password-file
```

▼ 기본 복제 관리자 비밀번호를 변경하는 방법

1 복제 관리자 비밀번호 설정을 위한 임시 파일을 만듭니다.

이 파일을 읽고 나중에 사용하기 위해 비밀번호를 저장합니다.

```
$ echo password > password-file
```

2 복제 토폴로지의 모든 사용자(대상) 서버에서 복제 관리자 바인드 비밀번호를 설정합니다.

```
$ dsconf set-server-prop -h host -p port def-repl-manager-pwd-file:password-file
```

예를 들면 다음과 같습니다.

```
$ dsconf set-server-prop -h host1 -p 1389 def-repl-manager-pwd-file:pwd.txt
```

3 임시 비밀번호 파일을 제거합니다.

```
$ rm password-file
```

복제 계약 만들기 및 변경

복제 계약은 지정된 소비자에게 업데이트를 보내는 방법을 구성하고 제어하는 공급자 매개 변수 집합입니다. 복제 계약은 소비자에게 업데이트를 보내는 공급자 복제 접미어에서 만들어야 합니다. 업데이트할 모든 소비자에 대해 공급자에서 복제 계약을 만들어야 합니다.

▼ 복제 계약을 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

DSCC를 사용하여 새 복제 계약을 만들려면 기존 복제 계약에서 복제 계약 구성 설정의 일부 또는 모두를 복사할 수 있습니다.

1 마스터 서버에서 복제할 각 소비자에 대해 복제 계약을 만듭니다.

```
$ dsconf create-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port [consumer-host:consumer-port]
```

예를 들면 다음과 같습니다.

```
$ dsconf create-repl-agmt -h host1 -p 1389 dc=example,dc=com host2:1389
```

명령줄을 사용하여 기존 복제 계약을 나열하려면 `dsconf list-repl-agmts` 명령을 사용합니다.

주 – 복제 중에 마스터에서 포트 번호를 변경할 경우 서버를 다시 초기화할 필요가 없습니다. 그러나 이전 주소(*host*:*old-port*)를 대상으로 하는 이전 복제 계약은 더 이상 사용할 수 없습니다. 포트 번호를 변경하기 전처럼 복제를 계속하려면 새 주소(*host*:*new-port*)를 사용하여 새 계약을 만들어야 합니다.

---

2 복제 계약이 올바로 만들어졌는지 확인합니다.

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN consumer-host:consumer-port
```

3 인증 상태가 비정상인 경우 `dsconf accord-repl-agmt` 명령을 실행합니다.

주 – 기본 복제 관리자를 사용하는 경우에만 `dsconf accord-repl-agmt` 명령을 사용합니다. 새 복제 관리자를 만든 경우에는 이 명령을 사용하지 마십시오. 그렇지 않으면 일부 필수 설정을 덮어쓰게 됩니다.

`dsconf accord-repl-agmt` 명령은 공급자 서버와 대상 서버가 동일한 복제 인증 설정을 공유하는지 확인합니다.

```
$ dsconf accord-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

예를 들면 다음과 같습니다.

```
$ dsconf accord-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

▼ 복제 계약의 대상을 변경하는 방법

이 절차를 수행하면 기존 복제 계약에 따라 대상이 지정된 원격 복제를 변경할 수 있습니다. 접미어 DN과 기존 계약의 구성은 그대로 유지됩니다.

- **복제 계약에 있는 원격 복제본의 호스트 이름과 포트 번호를 변경합니다.**

  ```
  $ dsconf change-repl-dest -h host -p port suffix-DN host:port new-host:new-port
  ```

  `-A` *protocol* 옵션을 사용하여 이 명령을 실행한 경우에는 복제에 사용된 인증 프로토콜을 변경할 수 있습니다.

단편 복제

기본적으로 복제 작업은 복제된 접미어의 전체 항목을 소비자 복제본에 복사합니다. 단편 복제 기능을 사용하면 사용할 접미어와 포함 또는 제외시킬 속성을 선택할 수 있습니다. 단편 복제는 복제 계약에 구성되므로 마스터의 각 사용자 복제 접미어에 대한 속성 집합을 정의할 수 있으며, 배포할 데이터를 제어하고 복제 대역폭과 사용자 자원을 보다 효율적으로 사용할 수 있습니다.

예를 들어 복제 대역폭을 줄이려면 `photo`, `jpegPhoto`, `audio`와 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 사용자 서버로 복제하도록 선택할 수도 있습니다.

단편 복제 시 고려 사항

주 – Directory Server 5.2 이전 버전에서는 단편 복제를 사용할 수 없습니다. 단편 복제 계약을 구성할 경우 마스터 복제본과 소비자 복제본 모두 Directory Server 5.2 이상을 사용해야 합니다.

속성의 단편 집합을 활성화하거나 수정하려면 소비자 복제본을 다시 초기화해야 하므로 배포 전에 단편 복제 요구를 결정하여 처음 복제된 접미어를 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 메커니즘의 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 수행하도록 허용하는 경우에는 사용자 서버에서 스키마 검사를 비활성화해야 합니다. 복제 메커니즘에서 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.

▼ 단편 복제를 구성하는 방법

단편 복제를 구성하려면 접미어를 지정하고 해당 접미어에 속성을 포함시킬지 여부를 결정한 다음 포함 또는 제외시킬 속성을 선택해야 합니다. 접미어에서 속성을 제외하도록 선택하면 모든 다른 속성이 자동으로 포함됩니다. 마찬가지로 접미어에서 속성을 포함하도록 선택하면 모든 다른 속성이 자동으로 제외됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **소스 서버에 있는 복제 계약에서 단편 복제를 구성합니다.**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port property:value
```

여기서 *property*는 `repl-fractional-exclude-attr` 또는 `repl-fractional-include-attr`입니다.

예를 들어 `dc=example,dc=com` 접미어에 대한 복제에서 JPEG 및 TIFF 사진을 제외하도록 단편 복제를 구성하려면 다음 명령을 사용합니다.

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389
 repl-fractional-exclude-attr:jpegPhoto repl-fractional-exclude-attr:tiffPhoto
```

제외할 기존 속성 목록에 속성을 추가하려면 다음 명령을 사용합니다.

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port repl-fractional-exclude-attr+:attribute
```

복제 우선 순위

복제 우선 순위 지정은 선택 사항입니다. 사용자 비밀번호 업데이트와 같은 특정 변경을 높은 우선 순위로 복제하도록 지정하는 복제 규칙을 만들 수 있습니다. 복제 규칙에 지정된 변경은 높은 우선 순위로 복제되고, 모든 다른 변경 사항은 일반 우선 순위로 복제됩니다.

주 – 복제 우선 순위 규칙은 마스터 서버에서만 만들어야 합니다. 허브 및 소비자에 대해서는 구성할 필요가 없습니다.

▼ 복제 우선 순위를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **마스터에서 새 복제 우선 순위 규칙을 만들려면 다음 명령을 사용합니다.**

```
$ dsconf create-repl-priority -h host -p port suffix-DN priority-name property:value
```

다음 등록 정보 중 하나 이상을 사용하여 복제 우선 순위를 설정할 수 있습니다.

- 작업 유형, `op-type`
- 바인드 DN, `bind-dn`
- 기본 DN, `base-dn`
- 속성 유형, `attr`

*priority-name*은 사용자 정의됩니다.

예를 들어 사용자 비밀번호 변경을 높은 우선 순위로 복제하도록 지정하는 복제 규칙을 만들려면 다음 명령을 사용합니다.

```
$ dsconf create-repl-priority -h host2 -p 1389 dc=example,dc=com pw-rule \
 attr:userPassword
```

현재 복제 규칙을 표시하려면 `dsconf list-repl-priorities -v` 명령을 사용합니다. 이 명령에서 `-v` 옵션을 사용하면 우선 순위 복제 규칙과 관련된 추가 정보가 표시됩니다.

```
$ dsconf list-repl-priorities -h host2 -p 1389 -v
```

자세한 내용은 `dsconf`(1M) 설명서 페이지를 참조하십시오.

복제본 초기화

복제 계약을 만들고 두 복제본을 모두 구성한 후 복제를 시작하려면 사용자 복제 접미어를 초기화해야 합니다. 초기화 중에 공급자 복제 접미어의 데이터가 사용자 복제 접미어로 복사됩니다.

또한, 특정 오류 조건이나 구성 변경 시에는 복제본을 다시 초기화해야 합니다. 예를 들어 어떤 이유로든 백업을 사용하여 단일 마스터 복제 접미어 데이터를 복원한 경우에는 백업에서 업데이트하는 모든 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. 이 절에 설명된 모든 초기화 방법은 자동으로 소비자 복제본의 색인을 재구성하므로 소비자가 클라이언트 읽기 요청에 대해 적절하게 응답할 수 있습니다.

다중 마스터 복제 시 토폴로지의 다른 마스터에서 업데이트한 소비자는 다시 초기화하지 않아도 됩니다.

▼ 원격(공급자) 서버에서 복제된 접미어를 초기화하는 방법

기존 복제 계약을 사용하여 원격 서버에서 접미어를 초기화할 수 있습니다. 이 방법은 다른 방법보다 더 간단하므로 가능하면 이 방법을 사용하십시오. 데이터 용량이 커서 가져오는 데 많은 시간이 소요되는 경우에만 다른 방법을 사용하십시오.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

DSCC를 사용하여 복제된 접미어를 온라인으로 초기화하면 소비자를 쉽게 초기화 또는 다시 초기화할 수 있습니다. 그러나 많은 항목을 초기화하는 경우 이 프로세스를 사용하면 시간이 많이 소요될 수 있습니다. 이 경우 명령줄에서 소비자를 오프라인으로 초기화하는 것이 더 효율적일 수 있습니다.

1 **복제본을 초기화합니다.**

```
$ dsconf init-repl-dest -h host -p port suffix-DN destination-host:destination-port [destination-host:destination-port]
```

여기서 *destination-host*:*destination-port*는 원격 서버에서 초기화할 대상 서버의 호스트와 포트입니다.

2 **(옵션) 각 계약에 대해 접미어가 초기화된 상태로 표시되는지 확인합니다.**

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN destination-host:destination-port
```

LDIF에서 복제본 초기화

▼ LDIF에서 복제된 접미어를 초기화하는 방법

이 절차에서는 LDIF 파일에서 복제된 접미어를 초기화하는 데 사용하는 일반 단계를 요약하여 설명합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

DSCC를 사용하여 복제된 접미어를 온라인으로 초기화하면 소비자를 쉽게 초기화 또는 다시 초기화할 수 있습니다. 그러나 많은 항목을 초기화하는 경우 이 프로세스를 사용하면 시간이 많이 소요될 수 있습니다. 이 경우 명령줄에서 소비자를 오프라인으로 초기화하는 것이 더 효율적일 수 있습니다.

1 **복제 계약을 설정했는지 확인합니다.**

이 작업은 복제본을 초기화하기 **전에** 수행해야 합니다.

2 **마스터 복제 접미어의 원본 접미어 데이터 복사본을 LDIF 파일로 내보냅니다.**

248 페이지 "복제된 접미어를 LDIF로 내보내는 방법"을 참조하십시오.

다중 마스터 복제 환경에서는 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 메타 데이터가 포함되지 않은 임의의 LDIF 파일을 사용하여 모든 복제본을 초기화할 수는 없습니다.

3 **단편 복제를 초기화할 경우 복제된 속성만 유지하도록 파일을 필터링한 다음 해당 파일을 모든 사용자 서버로 전송합니다.**

248 페이지 "단편 복제를 위한 LDIF 파일 필터링 "을 참조하십시오.

4 **복제본을 초기화합니다.**

다음 중 하나를 수행합니다.

- 오프라인(중지) 서버에서 빠르게 초기화하려면 `dsadm import` 명령을 사용합니다.

  ```
  $ dsadm import instance-path LDIF_file suffix-DN
  ```

- LDIF 파일에서 복제본을 온라인으로 초기화하려면 `dsconf import` 명령을 사용합니다.

  ```
  $ dsconf import -h host -p port LDIF_file suffix-DN
  ```

  `dsconf import` 명령을 사용하면 `dsadm import`를 사용할 때보다 더 느리지만 가져오기 작업을 수행하는 동안 서버를 중지할 필요가 없습니다.

접미어 초기화에 대한 자세한 내용과 예는 203 페이지 "접미어 초기화"를 참조하십시오. 명령 사용에 대한 자세한 내용은 `dsadm`(1M) 및 `dsconf`(1M)을 참조하십시오.

5 **(옵션) 각 계약에 대해 접미어가 초기화된 상태로 표시되는지 확인합니다.**

```
$ dsconf show-repl-agmt-status -h host -p port suffix-DN destination-host:destination-port
```

▼ 복제된 접미어를 LDIF로 내보내는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **다음 명령 중 하나를 사용하여 LDIF 파일의 복제된 접미어 내용을 내보냅니다.**
  - 오프라인 내보내기의 경우 다음을 입력합니다.

    ```
    $ dsadm export instance-path suffix-DN LDIF_file
    ```
  - 온라인 내보내기의 경우 다음을 입력합니다.

    ```
    $ dsconf export -h host -p port suffix-DN LDIF_file
    ```

다음 예에서는 전체 `dc=example,dc=com` 복제 접미어와 복제 정보를 `example_replica_export.ldif` 파일로 내보냅니다.

```
$ dsconf export -h host2 -p 1389 dc=example,dc=com  \
 /local/ds/ldif/example_export_replica.ldif
```

자세한 내용은 200 페이지 “LDIF에 백업”, `dsadm`(1M) 및 `dsconf`(1M) 설명서 페이지를 참조하십시오.

단편 복제를 위한 LDIF 파일 필터링

DSCC를 사용하면 단편 복제가 구성된 복제본의 초기화를 투명하게 처리할 수 있습니다. 선택한 속성만 초기화 중에 소비자로 보내집니다.

단편 복제를 구성한 경우 내보낸 LDIF 파일을 사용자 서버에 복사하기 전에 사용되지 않는 속성을 필터링해야 합니다. 디렉토리 서버는 이러한 용도의 `fildif` 도구를 제공합니다. 이 도구는 특정 LDIF 파일을 필터링하여 복제 계약에 정의된 속성 집합에서 허용하는 속성만 유지합니다.

이 도구는 서버의 구성을 확인하여 속성 집합 정의를 결정합니다. 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.

```
$ fildif -i /local/ds1/ldif/example_master.ldif \
 -o /local/ds1/ldif/filtered.ldif -b "cn=host2.example.com:1389, \
 cn=replica,cn=\\"dc=example,dc=com\\",cn=mapping tree,cn=config" -p /local/ds1
```

`fildif` 명령의 위치는 34 페이지 “명령 위치”를 참조하십시오.

`-i` 및 `-o` 옵션은 각각 입력 파일과 출력 파일을 나타냅니다. `-b` 옵션은 단편 복제가 정의된 복제 계약의 DN입니다. 다음 명령을 사용하여 이 DN을 찾을 수 있습니다.

```
$ ldapsearch -h host -p port -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=config" "(&(objectclass=nsds5replicationagreement) (nsDS5ReplicaPort=replica-port) \
 (nsDS5ReplicaHost=replica-host))" dn
```

예를 들면 다음과 같습니다.

```
$ ldapsearch -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w - \
 -b "cn=config" "(&(objectclass=nsds5replicationagreement) \
 (nsDS5ReplicaPort=2090)(nsDS5ReplicaHost=host2))" dn
Enter bind password:
version: 1
dn: cn=host2:1389,cn=replica,cn=dc\=example\,dc\=com,cn=mapping tree,cn=config
```

`fildif` 도구에 대한 전체 명령줄 구문은 `fildif`(1) 설명서 페이지를 참조하십시오.

그런 다음 `fildif` 도구에서 생성된 `filtered.ldif` 파일을 사용하여 이 복제 계약의 소비자를 초기화할 수 있습니다. 파일을 사용자 서버로 전송한 다음 203 페이지 “LDIF 파일에서 데이터 가져오기”에 설명된 것처럼 가져옵니다.

이진 복사를 사용하여 복제된 접미어 초기화

이진 복사는 한 서버의 이진 백업 파일을 사용하여 다른 서버에 동일한 디렉토리 내용을 복원함으로써 전체 서버를 복제합니다. 이진 복사를 사용하면 마스터 또는 허브 서버의 이진 복사본에서 서버를 초기화 또는 다시 초기화하거나, 다른 사용자 서버의 이진 복사본에서 소비자를 초기화 또는 다시 초기화할 수 있습니다.

주 – 이 고급 절차는 디렉토리 서버의 데이터베이스 파일과 상호 작용하며 숙련된 관리자만 사용해야 합니다.

이 기능의 특정 제한 사항으로 인해 이진 복사는 수백만 개의 항목이 포함된 복제본과 같이 대규모 데이터베이스 파일이 있는 복제본에만 실용적이며 시간 효율적입니다.

복제에서 이진 복사 사용을 위한 제한 사항

이진 복사 기능은 한 시스템의 데이터베이스 파일을 다른 시스템으로 이동하기 때문에 메커니즘에 다음과 같은 엄격한 제한이 적용됩니다.

- 두 시스템은 서비스 팩이나 패치를 비롯한 동일한 운영 체제를 실행해야 합니다.
- 두 시스템은 동일한 프로세서 아키텍처를 공유해야 합니다. 예를 들어 두 UltraSPARC® T1 프로세서 간에는 이진 복사를 수행할 수 있지만 UltraSPARC T1과 AMD Opteron 프로세서 간에는 이진 복사를 수행할 수 없습니다.
- 두 시스템은 모두 빅 엔디언 또는 리틀 엔디언 중 하나여야 합니다.
- 두 시스템이 메모리를 동일한 방식으로 매핑해야 합니다. 예를 들어 32비트 시스템의 서버 인스턴스와 64비트 시스템의 다른 서버 인스턴스 간이 아니라 두 64비트 시스템의 서버 인스턴스 간에 이진 복사를 수행할 수 있습니다.

- 두 시스템에 설치된 디렉토리 서버는 이진 형식(32비트 또는 64비트), 서비스 팩 및 패치 수준까지 동일해야 합니다.
- 두 서버의 디렉토리 트리와 구성된 접미어는 동일해야 합니다. **모든** 접미어에 대한 데이터베이스 파일을 **반드시** 함께 복사해야 합니다. 개별 접미어를 복사할 수는 없습니다.
- VLV(가상 목록 보기) 색인을 비롯한 두 서버의 각 접미어 색인은 동일하게 구성되어야 하며,접미어의 데이터베이스 이름도 같아야 합니다.
- 각 서버에는 복제본과 동일한 접미어가 구성되어 있어야 합니다.
- 단편 복제를 구성하는 경우 모든 서버에서 동일하게 구성해야 합니다.
- 속성 암호화는 두 서버에서 모두 사용할 수 없습니다.
- 속성 값 고유성 플러그인을 사용하는 경우 두 서버에서 동일하게 구성해야 하며, 아래 절차에 설명된 것처럼 새 복사본에서 다시 구성해야 합니다.

  이러한 절차에서는 이진 복사를 수행하는 대체 방법에 대해 설명합니다. 서버를 중지할 필요가 없는 이진 복사 및 최소 디스크 공간을 사용하는 이진 복사

서버 초기화를 위한 이진 복사본 만들기

이 절에서는 서버 초기화를 위한 이진 복사본을 만드는 방법과 최소 디스크 공간을 사용하는 이진 복사본을 만드는 방법에 대해 설명합니다.

▼ 서버 초기화를 위한 이진 복사본을 만드는 방법

이 절차에서는 복제된 서버 초기화를 위한 이진 복사를 수행합니다. 이 작업에서는 정상적인 백업 기능을 사용하여 서버 데이터베이스 파일을 복사합니다. 정상적인 백업을 수행하면 서버를 중지할 필요 없이 모든 데이터베이스 파일을 일관된 상태로 유지할 수 있습니다.

이 절차에는 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.

이 절차의 일부로, DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오. 해당 절차의 다른 부분은 명령줄에서만 수행할 수 있습니다.

1 **새 복제 접미어의 대상 시스템에 디렉토리 서버를 설치하고, 필요한 경우 새 서버 인스턴스를 만든 다음 249 페이지 "복제에서 이진 복사 사용을 위한 제한 사항"에 따라 서버를 구성합니다.**

2. **복제 토폴로지에 이 복제된 접미어를 포함하는 모든 복제 계약을 만듭니다.**
   이 복제본에 공급자의 계약을 포함시킵니다. 이 복제본이 전용 소비자가 아닌 경우 복제본의 계약을 해당 소비자에 포함시킵니다. 241 페이지 "복제 계약 만들기 및 변경"을 참조하십시오.

3. **초기화하려는 유형과 같은 유형(마스터, 허브 또는 사용자)의, 완전히 구성 및 초기화된 복제본을 선택하고 197 페이지 "이진 백업"의 절차에 따라 정상적인 백업을 수행합니다.**

4. **`ftp` 명령을 사용하여 백업 디렉토리의 파일을 대상 시스템의 디렉토리로 복사 또는 전송합니다.**

5. **다중 복제 시나리오에서 새 마스터를 초기화한 경우 209 페이지 "다중 마스터 시나리오에서 마스터 복원"의 절차를 수행합니다.**

▼ 이진 복사를 사용하여 최소 디스크 공간에서 서버를 초기화하는 방법

이 절차에서는 데이터베이스 파일의 백업 복사본을 만들지 않으므로 더 적은 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 사용자 서버를 다시 초기화하거나 새 마스터 서버를 초기화하는 경우에만 사용해야 합니다. 기존 마스터 복제본을 다시 초기화하려면 온라인 초기화를 사용하거나, LDIF 파일을 가져오거나, 250 페이지 "서버 초기화를 위한 이진 복사본 만들기"의 절차를 수행합니다.

이 절차의 일부로, DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오. 해당 절차의 다른 부분은 명령줄에서만 수행할 수 있습니다.

1. **새 복제 접미어의 대상 시스템에 디렉토리 서버를 설치하고, 필요한 경우 새 서버 인스턴스를 만든 다음 249 페이지 "복제에서 이진 복사 사용을 위한 제한 사항"에 따라 서버를 구성합니다.**

2. **복제 토폴로지에 이 복제본과 연결된 모든 복제 계약을 작성합니다.**
   이 복제본에 공급자의 계약을 포함시킵니다. 이 복제본이 전용 소비자가 아닌 경우 복제본의 계약을 해당 소비자에 포함시킵니다. 241 페이지 "복제 계약 만들기 및 변경"을 참조하십시오.

3. **59 페이지 "디렉토리 서버 인스턴스 시작, 중지 및 다시 시작"에 설명된 것처럼 초기화 또는 다시 초기화할 대상 서버를 중지합니다.**

4. **초기화하려는 유형과 같은 유형(마스터, 허브 또는 사용자)의, 완전히 구성 및 초기화된 복제본을 선택하고 서버도 중지합니다.**

   다중 마스터 구성의 마스터 복제본을 복제하는 경우 이 복제본을 중지하기 전에 다른 마스터의 최신 변경 사항이 모두 적용되어 있는지 확인합니다.

5. **트랜잭션 로그, 변경 로그 및 지역 파일(`__db.`*xxx* `files`)을 포함한 모든 데이터베이스 파일을 대상 서버에서 제거합니다.**

   파일 위치를 변경하지 않은 경우 데이터베이스 파일과 트랜잭션 로그는 *instance-path*`/db` 디렉토리에 있습니다.

6. **`ftp` 등의 명령을 사용하여 트랜잭션 로그 및 변경 로그를 포함한 모든 데이터베이스 파일을 소스 복제본 시스템에서 대상 시스템으로 복사하거나 전송합니다.**

   파일 위치를 변경하지 않은 경우 데이터베이스 파일과 트랜잭션 로그는 *instance-path*`/db` 디렉토리에 있습니다.

   마스터 또는 허브 복제본을 초기화할 경우 기본적으로 *instance-path*`/changelog`에 있는 변경 로그의 모든 파일도 함께 복사해야 합니다.

7. **소스 및 대상 서버를 모두 다시 시작합니다.**

계단식 복제 시 복제본 초기화

계단식 복제의 경우 복제본을 다음 절차에 표시된 순서로 항상 초기화합니다.

▼ 계단식 복제 시 복제본을 초기화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1. **다중 마스터 복제가 있는 경우 하나의 마스터에 복제할 전체 데이터 집합이 있는지 확인한 다음 이 마스터를 사용하여 각 마스터에서 복제본을 초기화합니다.**

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.**

복제된 접미어 색인화

색인은 다른 서버 인스턴스로 자동으로 복제되지 않습니다. 복제된 접미어를 보유하는 모든 서버 인스턴스에 대한 속성을 색인화하려면 다음 작업 중 하나를 수행합니다.

- 복제된 접미어를 보유하는 모든 서버 인스턴스를 DSCC에서 서버 그룹으로 관리합니다. 그룹의 한 서버에 색인을 추가한 다음 서버 구성 복사 작업을 사용하여 색인 설정을 그룹의 다른 서버에 복사합니다.

  DSCC에 대한 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스"를 참조하십시오.
- 12 장에 설명된 것처럼 `dsconf` 명령을 사용하여 각 서버 인스턴스에서 색인을 관리합니다.
- 249 페이지 "이진 복사를 사용하여 복제된 접미어 초기화"에 설명된 것처럼 이진 복사를 사용하여 접미어를 초기화합니다.

대용량 복제된 접미어에 많은 항목을 증분하여 추가

항목이 많은 디렉토리에 대용량의 항목을 추가하려면 `ldapmodify -a` 명령을 사용하지 마십시오. 그러면 많은 시간이 소요됩니다. 대신 복제된 토폴로지에서 항목을 추가하는 옵션과 함께 `dsconf import` 명령을 사용하여 새 항목을 증분하여 추가하십시오. 항목을 가져오면 복제 메타 데이터와 추가 항목이 포함된 LDIF 파일이 생성됩니다. 그런 다음 생성된 이 LDIF 파일을 다른 복제본으로 가져옵니다. 생성된 LDIF 파일은 데이터를 추가하는 복제본을 통해 복제 동기화가 일관되게 수행되는지 확인합니다.

▼ 대용량 복제된 접미어에 많은 항목을 추가하는 방법

시작하기 전에 이 절차에서는 대용량 LDIF 파일을 생성합니다. 첫 번째 `dsconf import` 명령을 실행하기 전에 생성되는 LDIF 파일을 위한 충분한 디스크 공간이 있는지 확인합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

주의 – 이 절차를 사용하여 여러 번에 걸쳐 많은 항목이 있는 서버를 초기화할 수 있습니다. 그러나 가져오기 중 하나가 실패할 경우 전체 데이터베이스가 손실될 수 있습니다. 따라서 가져오기를 수행하기 전에 항상 데이터를 백업합니다.

1 마스터 복제본에서 항목을 가져옵니다.

```
$ dsconf import -h host -p port -K generated-LDIF-file suffix-DN
```

`-K` 옵션을 사용하면 기존 데이터가 제거되지 않습니다. 또한 복제 프로세스에 필요한 새로운 항목과 정보를 포함하는 *generated-LDIF-file* 파일을 생성합니다.

2 **모든 다른 복제본에서 이전 단계에서 생성된 파일을 가져옵니다.**

```
$ dsconf import -h host -p port \
-K -f incremental-output=no generated-LDIF-file suffix-DN
```

`-f incremental-output=no` 옵션은 추가 LDIF 파일이 생성되지 않도록 지정합니다. 이 절차에는 생성된 LDIF 파일이 하나만 필요합니다.

복제 및 참조 무결성

복제에서 참조 무결성 플러그인을 사용하는 경우 모든 마스터 서버에서 해당 플러그인을 활성화해야 합니다. 허브나 사용자 서버에서는 활성화할 필요가 없습니다.

복제 환경에서 참조 무결성 플러그인을 사용하려면 다음과 같은 제한 사항이 있습니다.

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

참조 무결성 플러그인을 구성하는 방법에 대한 자세한 내용은 230 페이지 “참조 무결성 플러그인을 구성하는 방법”을 참조하십시오.

SSL을 통한 복제

모든 복제 작업이 SSL 연결을 통해 수행되도록 복제에 포함된 디렉토리 서버를 구성할 수 있습니다.

▼ SSL에 대한 복제 작업을 구성하는 방법

이 절차에서는 두 개의 마스터가 있는 복제 토폴로지에서 복제를 설정하는 명령 예를 보여줍니다.

주 – 이 예에서는 자체 서명된 인증서를 사용하는 간단한 복제 구성을 보여줍니다. 작업 환경에서 SSL을 통해 복제를 설정할 경우 인증 기관의 신뢰할 수 있는 인증서를 사용하면 보안이 향상됩니다.

공급자 서버 인증서가 SSL 핸드셰이크 중에 클라이언트 역할을 할 수 없는 SSL 서버 전용 인증서이면 SSL을 통한 복제가 제대로 수행되지 않습니다.

복제가 SSL에 의해 보안되는 동안에도 간단한 바인드 및 비밀번호를 사용하여 복제 관리자 인증이 계속 수행됩니다. 클라이언트 기반 인증을 사용하여 복제를 완벽하게 보안할 수 있지만, 이렇게 하려면 보다 복잡한 설정이 필요합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 새 서버를 만든 다음 시작합니다.

```
$ dsadm create -p 1389 -P 1636 /local/ds1
$ dsadm create -p 2389 -P 2636 /local/ds2

$ dsadm start /local/ds1
$ dsadm start /local/ds2
```

2 모든 서버에서 빈 접미어를 만듭니다.

```
$ dsconf create-suffix -e -i -p 1389 dc=example,dc=com
$ dsconf create-suffix -e -i -p 2389 dc=example,dc=com
```

3 모든 서버에서 다중 마스터 비밀번호 파일을 설정합니다.

```
$ dsconf set-server-prop -e -i -h example1.server -p 1389 \
 def-repl-manager-pwd-file:/local/ds1/replmanrpwd1.txt
$ dsconf set-server-prop -e -i -h example2.server -p 2389 \
 def-repl-manager-pwd-file:/local/ds1/replmanrpwd2.txt
```

4 모든 서버에서 복제를 활성화합니다.

```
$ dsconf enable-repl -h example1.server -p 1389 -e -i -d 1 master dc=example,dc=com
$ dsconf enable-repl -h example2.server -p 2389 -e -i -d 2 master dc=example,dc=com
```

5 모든 서버에서 기존 기본 인증서를 봅니다.

```
$ dsadm show-cert -F der -o certfile1 /local/ds1 defaultCert
$ dsadm show-cert -F der -o certfile2 /local/ds2 defaultCert
```

6 모든 서버에서 모든 다른 서버의 신뢰할 수 있는 CA 인증서를 추가합니다.

```
$ dsadm add-cert --ca /local/ds1 "ds2 Repl Manager Cert" certfile2
$ dsadm add-cert --ca /local/ds2 "ds1 Repl Manager Cert" certfile1
```

7 모든 마스터와 허브(소스) 서버에서 모든 사용자(대상) 서버를 사용하여 복제 계약을 만듭니다.

보안 LDAP 포트가 복제 계약에 사용됩니다.

```
$ dsconf create-repl-agmt -h example1.server -p 1389 -e -i \
 --auth-protocol "ssl-simple" dc=example,dc=com example2.server:2636
$ dsconf create-repl-agmt -h example2.server -p 2389 -e -i \
 --auth-protocol "ssl-simple" dc=example,dc=com example1.server:1636
```

8 모든 복제 계약에 대해 복제 계약에 있는 사용자(대상) 서버의 복제 관리자 파일이 되도록 인증 비밀번호 파일을 구성합니다.

```
$ dsconf set-repl-agmt-prop -h example1.server -p 1389 -e -i \
 dc=example,dc=com example2.server:2636 auth-pwd-file:/local/ds1/replmanrpwd2.txt
```

```
$ dsconf set-repl-agmt-prop -h example2.server -p 2389 -e -i \
 dc=example,dc=com example1.server:1636 auth-pwd-file:/local/ds1/replmanrpwd1.txt
```

접미어 초기화가 끝나면 공급자는 SSL을 통해 모든 복제 업데이트 메시지를 소비자에게 보내며, 해당 옵션을 선택한 경우 인증서를 사용합니다. DSCC에서 SSL용으로 구성된 계약을 사용하면 사용자 초기화 시에도 보안 연결이 사용됩니다.

9 모든 서버에서 서버를 다시 시작하여 구성 변경을 적용합니다.

```
$ dsadm restart /local/ds1
$ dsadm restart /local/ds2
```

10 마스터 서버 중 하나에서 접미어를 초기화합니다.

```
$ dsconf import -h example1.server -p 1389 -e -i /tmp/Example.ldif dc=example,dc=com
```

11 아직 초기화되지 않은 모든 서버에서 복제 계약을 사용하여 서버를 초기화합니다.

```
$ dsconf init-repl-dest -e -i -h example1.server -p 1389 \
 dc=example,dc=com example1.server:2636
```

WAN을 통한 복제

디렉토리 서버에서는 WAN을 통해 연결된 시스템 간의 다중 마스터 복제를 포함하여 모든 형태의 복제를 수행할 수 있습니다. 공급자 서버에서는 이 복제를 통해 높은 대기 시간과 낮은 대역폭을 가진 네트워크의 대역폭을 최대한 활용하여 소비자를 초기화하고 업데이트할 수 있습니다.

주 – WAN을 통해 복제되는 복제 토폴로지를 배포하거나 문제를 해결할 경우 네트워크 속도, 대기 시간 및 패킷 손실을 확인해야 합니다. 이러한 영역의 네트워크 문제는 복제 지연을 초래할 수 있습니다.

또한 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, 오류율, 평균 대기 시간, 평균 대역폭 등 다양한 요인에 의해 결정되며, 이에 제한되지 않습니다.

사용자 환경의 복제에 대한 질문 사항이 있는 경우 Sun 서비스 공급자에게 문의하십시오.

복제 메커니즘의 내부 매개 변수는 기본적으로 WAN에 최적화되어 있습니다. 하지만 위에 명시된 요인들로 인해 복제 속도가 느려질 경우 경험적으로 창 크기 및 그룹 크기 매개 변수를 조정할 수 있습니다. 네트워크 사용량이 많은 시간을 피해 복제를 예약함으로써 전체적인 네트워크 사용을 향상시킬 수도 있습니다. 마지막으로 디렉토리 서버는 대역폭 사용 최적화를 위해 복제 데이터 압축을 지원합니다.

네트워크 매개 변수 구성

창 및 그룹 네트워크 매개 변수는 네트워크에서 보다 효율적으로 항목을 전송하기 위해 복제 메커니즘에서 항목을 그룹화하는 방법을 결정합니다. 이 두 매개 변수는 공급자 및 소비자가 복제 업데이트 메시지와 승인을 교환하는 방법에도 영향을 줍니다. 매개 변수는 모든 복제 계약에서 구성할 수 있으므로 각 소비자의 특정 네트워크 조건에 따라 복제 성능을 조정할 수 있습니다.

수정 결과를 모니터하고 매개 변수를 적절하게 조정합니다. 자세한 내용은 269 페이지 "복제 상태 가져오기"를 참조하십시오. 창 크기 및 그룹 크기 매개 변수를 수정하는 경우 복제를 중단할 필요가 없습니다.

창 크기 구성

창 크기(기본값 10)는 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 LAN(Local Area Network)으로 연결된 복제본에서도 창 크기를 높게 설정하는 것이 좋습니다.

▼ 창 크기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **창 크기를 수정합니다.**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN consumer-host:consumer-port transport-window-size:value
```

예를 들면 다음과 같습니다.

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-window-size:20
```

그룹 크기 구성

그룹 크기(기본값 1)는 하나의 업데이트 메시지로 처리할 수 있는 최대 데이터 수정 항목 수를 나타냅니다. 네트워크 연결이 복제 과정에 방해가 되는 경우, 그룹 크기를 기본값보다 높은 값(예: 10)으로 증가시키고 복제 성능을 다시 확인합니다.

그룹 크기를 늘릴 경우 다음을 확인해야 합니다.

- 창 크기가 그룹 크기보다 훨씬 높게 설정되어 있는지 확인합니다.
- 창 크기를 그룹 크기로 나눈 값이 소비자의 `cn=config`에 설정된 `nsslapd-maxThreadsPerConn` 값보다 훨씬 큰지 확인합니다(일반적으로 두 배).

  그룹 크기가 1보다 높게 설정되어 있으면 공급자는 그룹이 채워질 때까지 기다리지 않고 소비자에게 업데이트를 보냅니다.

▼ 그룹 크기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **그룹 크기를 수정합니다.**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 consumer-host:consumer-port transport-group-size:value
```

예를 들면 다음과 같습니다.

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-group-size:10
```

복제 작업 예약

복제본 간에 즉시 동기화할 필요가 없으면 네트워크 사용량이 적은 기간 동안 복제를 예약할 수 있습니다. 네트워크 사용량이 적을수록 데이터 복제가 더 빠르게 완료됩니다.

복제를 특정 시간, 매일 또는 매주 시작하고 끝내도록 예약할 수 있습니다. 복제 계약을 통해 이 작업을 모든 소비자에 대해 개별적으로 수행할 수 있습니다. 새 일정이 즉시 적용되므로 해당 소비자에 대한 다음 데이터 복제는 일정에 처음 부합될 때까지 지연됩니다.

▼ 복제 작업을 예약하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **복제 일정을 수정합니다.**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 host:port repl-schedule:value
```

예를 들어 매일 밤 2시와 4시 사이에 복제가 발생하도록 설정하려면 다음을 입력합니다.

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 repl-schedule:"0200-0400 0123456"
```

여기서 0123456은 요일을 나타냅니다. 즉, 0은 일요일을, 1은 월요일을 나타냅니다.

복제 압축 구성

복제의 대역폭 사용량을 줄이려면 소비자를 업데이트할 때 데이터를 압축하여 보내도록 복제를 구성할 수 있습니다. 복제 메커니즘은 Zlib 압축 라이브러리를 사용합니다.. 압축을 사용하려면 공급자와 소비자 모두 Solaris 또는 Linux 플랫폼에서 실행해야 합니다.

명시적인 테스트를 통해 사용자의 WAN 환경의 예상 복제 사용에 대해 가장 우수한 결과를 제공하는 압축 수준을 선택해야 합니다. 압축 및 압축 해제 계산으로 인해 복제가 느려지므로 네트워크 대역폭이 충분한 LAN에서는 이 매개 변수를 설정하지 마십시오.

▼ 복제 압축을 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **마스터 서버에서 복제 계약 항목에 대한 복제 압축을 구성합니다.**

```
$ dsconf set-repl-agmt-prop -h host -p port suffix-DN \
 consumer-host:consumer-port transport-compression:level
```

여기서 *level*은 high, medium, low 또는 none입니다.

예를 들어 복제 업데이트를 host1:1389에 있는 소비자에게 보낼 때 가장 빠른 압축을 사용하려면 다음을 입력합니다.

```
$ dsconf set-repl-agmt-prop -h host2 -p 1389 dc=example,dc=com host1:1389 \
 transport-compression:high
```

압축 수준 설정에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

복제 토폴로지 수정

이 절에서는 기존 복제 토폴로지 관리의 해당 특징에 대해 설명합니다.

복제 관리자 변경

복제 계약을 편집하여 사용자 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 메커니즘은 복제 복구 설정의 제한 내에서 오류 수정에 필요한 모든 업데이트를 자동으로 보냅니다. 절차는 239 페이지 "기본값이 아닌 복제 관리자 사용"을 참조하십시오.

복제 계약 관리

복제 계약을 비활성화, 활성화 또는 삭제할 수 있습니다.

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 메커니즘을 다시 시작하는 방법에 대한 자세한 내용은 260 페이지 "복제 계약 활성화"를 참조하십시오.

▼ 복제 계약을 비활성화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **복제 계약을 비활성화합니다.**

  ```
  $ dsconf disable-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
  ```

  예를 들면 다음과 같습니다.

  ```
  $ dsconf disable-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
  ```

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 이 공급자 변경 로그 및 사용자 지연 제거의 최대 크기와 사용 기간입니다(235 페이지 "고급 사용자 구성을 수행하는 방법" 참조).

중단 기간이 짧고 복제를 복구할 수 있으면 마스터는 계약이 다시 활성화될 때 자동으로 소비자를 업데이트합니다.

▼ 복제 계약을 활성화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **복제 계약을 활성화합니다.**

```
$ dsconf -h host -p port enable-repl-agmt suffix-DN consumer-host:consumer-port
```

예를 들면 다음과 같습니다.

```
$ dsconf -h host2 -p 1389 enable-repl-agmt dc=example,dc=com host1:1389
```

복제 계약 삭제

복제 계약을 삭제하면 해당 소비자에 대한 복제가 중지되고 계약에 대한 모든 구성 정보가 제거됩니다. 나중에 복제를 계속하려면 260 페이지 "복제 계약 비활성화"에 설명된 것처럼 계약을 비활성화합니다.

▼ 복제 계약을 삭제하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **복제 계약을 삭제합니다.**

```
$ dsconf delete-repl-agmt -h host -p port suffix-DN consumer-host:consumer-port
```

예를 들면 다음과 같습니다.

```
$ dsconf delete-repl-agmt -h host2 -p 1389 dc=example,dc=com host1:1389
```

복제본 수준 올리기 또는 내리기

복제본 수준을 올리거나 내리면 복제 토폴로지에서 해당 역할이 변경됩니다. 전용 사용자의 수준을 올리면 허브가 되고 허브의 수준을 올리면 마스터가 됩니다. 이와 마찬가지로 마스터의 수준을 내리면 허브가 되고 허브의 수준을 내리면 전용 사용자가 됩니다. 하지만 마스터 수준을 직접 사용자로 내리거나 사용자 수준을 직접 마스터로 올릴 수는 없습니다.

다중 마스터 복제 메커니즘에서 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 로드에 포함되어 있으면 신속한 로컬 변경을 허용하는 마스터가 되어 다른 사이트의 다른 마스터로 변경 사항을 복제할 수 있습니다.

접미어 수준을 올리거나 내릴 경우 다음에 유의하십시오.

- 소비자의 수준을 올리면 허브가 되고 허브의 수준을 올리면 마스터가 됩니다. 서버의 수준을 소비자에서 마스터로 직접 올릴 수 없습니다. 수준을 소비자에서 허브로 올린 다음 허브에서 마스터로 올려야 합니다. 마찬가지로 수준을 마스터에서 소비자로 내릴 경우 마스터에서 허브로 내린 다음 허브에서 소비자로 내려야 합니다.
- 마스터 수준을 허브로 내리면 복제본은 읽기 전용이 되어 나머지 마스터로 참조를 보내도록 구성됩니다. 새 허브는 허브나 전용 사용자 등 해당 사용자를 모두 유지합니다.
- 단일 마스터 수준을 허브로 내리면 마스터 복제본이 없는 토폴로지가 됩니다. 디렉토리 서버에서는 새 마스터가 정의될 것이라는 가정 하에 이 작업을 허용합니다. 하지만 새 마스터를 다중 마스터로 추가하여 다른 마스터의 수준을 내리기 전에 초기화하는 것이 바람직합니다.
- 허브의 수준을 소비자로 내리기 전에 허브에서 모든 복제 계약을 비활성화하거나 삭제해야 합니다. 그렇지 않으면 수준 내리기 작업이 실패하고 오류가 발생합니다. `LDAP_OPERATIONS_ERROR "Unable to demote a hub to a read-only replica if some agreements are enabled"`.

  다른 허브나 마스터가 허브 소비자를 업데이트하지 않은 경우 해당 소비자가 더 이상 업데이트되지 않습니다. 나머지 허브나 마스터에서 새 계약을 작성하여 소비자를 업데이트해야 합니다.
- 사용자 수준을 허브로 올리면 해당 변경 로그가 활성화되며 사용자와의 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 수준을 올리거나 내리는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **다음 명령 중 하나를 사용하여 복제본의 수준을 올리거나 내립니다.**

```
$ dsconf promote-repl -h host -p port role suffix-DN

$ dsconf demote-repl -h host -p port role suffix-DN
```

여기서 *role*은 `master`, `hub` 또는 `consumer`입니다.

복제된 접미어 비활성화

복제된 접미어를 비활성화하면 복제 토폴로지에서 이 복제 접미어가 제거되어 마스터, 허브 또는 사용자 역할에 따라 업데이트되거나 업데이트를 보내지 않습니다. 공급자 서버에서 접미어를 비활성화하면 모든 복제 계약이 삭제되며 다시 복제본을 활성화할 경우 새로 복제 계약을 만들어야 합니다.

▼ 복제된 접미어를 비활성화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **복제된 접미어를 비활성화합니다.**

  ```
  $ dsconf disable-repl -h host -p port suffix-DN
  ```

  예를 들면 다음과 같습니다.

  ```
  $ dsconf disable-repl -h host2 -p 1389 dc=example,dc=com
  ```

복제된 접미어를 동기화된 상태로 유지

정기 유지관리 작업을 위해 복제에 포함된 디렉토리 서버를 중지했다가 다시 온라인으로 전환할 경우 복제를 통해 즉시 업데이트되는지 확인해야 합니다. 다중 마스터 환경의 마스터인 경우 다중 마스터 집합의 다른 마스터가 디렉토리 정보를 업데이트해야 합니다. 유지 관리를 위해 오프라인 상태로 설정했던 허브 서버나 전용 사용자 서버가 다시 온라인 상태가 되면 마스터 서버가 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 있고, **또한** 소비자가 초기화된 경우에만 사용할 수 있습니다.

복제 재시도 알고리즘

소스 복제본이 대상에 복제되지 않은 경우 증분된 시간 간격으로 복제를 주기적으로 다시 시도합니다. 재시도 간격은 오류 유형에 따라 다릅니다.

항상 소스 복제본과 대상 복제본의 동기화를 유지하도록 복제 계약을 구성한 경우에도 5분 이상 오프라인 상태였던 복제본을 즉시 업데이트하기는 어렵습니다.

▼ 복제를 강제로 업데이트하는 방법

복제가 중지된 경우 대상 접미어에 대한 복제 업데이트를 강제로 수행할 수 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **소스 서버에서 대상 서버에 대한 복제 업데이트를 다시 시작합니다.**

  ```
  $ dsconf update-repl-dest-now -h host -p port suffix-DN destination-host:destination-port
  ```

예를 들면 다음과 같습니다.

```
$ dsconf update-repl-dest-now -h host2 -p 1389 dc=example,dc=com host1:1389
```

새 시스템으로 마스터 복제본 이동

경우에 따라 마스터 복제본을 다른 시스템으로 이동해야 할 수도 있습니다. 동일한 호스트 이름과 포트 번호를 사용할 필요가 없는 경우 `dsconf change-repl-dest`를 사용하여 원격 복제본의 호스트 이름과 포트 번호를 변경합니다. 자세한 내용은 243 페이지 "복제 계약의 대상을 변경하는 방법"을 참조하십시오.

동일한 호스트 이름과 포트 번호를 유지해야 할 경우 기존 토폴로지에서 마스터를 제거한 다음 해당 마스터를 토폴로지에 다시 추가해야 합니다.

DSCC는 영향 받는 모든 복제 계약을 다루기 때문에 DSCC를 사용하여 해당 작업을 쉽게 수행할 수 있습니다. 그러나 DSCC를 사용하면 마스터가 원래 토폴로지에 있었던 것과 동일한 복제 아이디를 지정할 수 없습니다. 동일한 복제 아이디를 사용하려면 명령줄을 사용하여 다음과 같이 해당 작업을 수행해야 합니다.

▼ 기존 복제 토폴로지에서 마스터를 제거하는 방법

시작하기 전에 마스터의 모든 변경 사항이 이미 복제되었는지 확인합니다.

1. **가능한 경우 변경 사항이 손실되지 않도록 이진 복사를 사용하여 마스터를 백업합니다.**
2. **마스터 복제본의 수준을 허브 복제본의 수준으로 내립니다.**
   261 페이지 "복제본 수준 올리기 또는 내리기"를 참조하십시오.
3. **허브에서 다른 서버로 복제를 시작할 때까지 기다립니다.**
   허브에서 토폴로지의 다른 서버에 복제 세션을 여는 경우 허브는 RUV에 남아 있지만 더 이상 참조에서 사용되지 않습니다.
4. **허브를 중지합니다.**
   59 페이지 "디렉토리 서버 인스턴스 시작, 중지 및 다시 시작"을 참조하십시오.
5. **토폴로지에서 허브를 제거합니다.**
   262 페이지 "복제된 접미어 비활성화"를 참조하십시오.

▼ 기존 복제 토폴로지에 마스터를 추가하는 방법

1. **동일한 복제 아이디를 사용하여 마스터 복제본을 추가합니다.**
   238 페이지 "마스터 복제본에서 복제 활성화"를 참조하십시오.

2. **토폴로지의 해당 마스터에서 다른 복제본까지 복제 계약을 다시 만듭니다.**
3. **새 마스터를 초기화합니다.**
   a. **마스터를 백업할 수 있는 경우 이 백업에서 마스터를 초기화합니다.**
   b. **마스터를 백업할 수 없는 경우(시스템 중단으로 인해) 토폴로지에 있는 다른 마스터에서 마스터를 초기화합니다.**

Directory Server 6.2 이전 버전의 복제

이 절에서는 Directory Server 6.2 이전 버전에서 복제를 구성하는 방법에 대해 설명합니다.

Directory Server 6.2 및 Directory Server 5.1 또는 5.2 간의 복제

Directory Server 5.1, 5.2 및 6.2은 복제 구성이 호환되지만 다음과 같은 예외가 있습니다.

- Directory Server 6.2 이전 버전에서 복제 우선 순위가 지원되지 않습니다. 6.2 마스터 복제본에서 복제 우선 순위를 구성한 경우 Directory Server 6.2을 실행하는 소비자에게는 복제 우선 순위가 전송되지만, 디렉토리 서버 이전 버전을 실행하는 소비자에게는 복제 우선 순위가 전송되지 않습니다.
- Directory Server 5.1 또는 5.2 마스터를 포함하는 복제 토폴로지에서는 마스터 수가 제한되어 지원됩니다. Directory Server 6.2은 복제 토폴로지에서 무제한의 마스터를 지원하지만, 복제 토폴로지에 Directory Server 5.2 마스터 서버가 포함되어 있는 경우에는 이 수가 4로 제한됩니다. Directory Server 5.1은 다중 마스터 복제를 지원하지 않습니다.

레트로 변경 로그 사용

레트로 변경 로그는 LDAP 클라이언트에서 디렉토리 서버 데이터에 대한 변경 기록을 확인하는 데 사용됩니다. 레트로 변경 로그는 디렉토리 서버 변경 로그와는 다른 별도의 데이터베이스에서 `cn=changelog` 접미어 아래에 저장됩니다.

레트로 변경 로그는 독립형 서버나 복제 토폴로지의 각 서버에서 활성화할 수 있습니다. 레트로 변경 로그를 서버에서 활성화하면 기본적으로 해당 서버의 모든 접미어 업데이트가 기록됩니다. 지정된 접미어에 대한 업데이트만 기록하도록 레트로 변경 로그를 구성할 수 있습니다.

복제된 토폴로지에서의 레트로 변경 로그 사용에 대한 자세한 내용과 레트로 변경 로그 사용 제한 사항은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Replication and the Retro Change Log Plug-In"을 참조하십시오.

레트로 변경 로그의 항목 속성에 대한 자세한 내용은 `changeLogEntry`(5dsoc) 설명서 페이지를 참조하십시오.

레트로 변경 로그 수정에 대한 자세한 내용은 `dsconf`(1M) 설명서 페이지를 참조하십시오.

이 절에서는 레트로 변경 로그를 사용할 수 있는 다양한 방법에 대해 설명합니다.

▼ 레트로 변경 로그를 활성화하는 방법

레트로 변경 로그를 사용하려면 먼저 활성화해야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 레트로 변경 로그 구성 항목을 수정합니다.

```
$ dsconf set-server-prop -h host -p port retro-cl-enabled:on
```

2 서버를 다시 시작합니다.

자세한 내용은 59 페이지 "디렉토리 서버 인스턴스 시작, 중지 및 다시 시작"을 참조하십시오.

▼ 지정된 접미어에 대한 업데이트를 기록하도록 레트로 변경 로그를 구성하는 방법

레트로 변경 로그를 서버에서 활성화하면 기본적으로 해당 서버의 모든 접미어 업데이트가 기록됩니다. 이 절차에서는 지정된 접미어에 대한 업데이트만 기록하도록 레트로 변경 로그를 구성하는 방법에 대해 설명합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 레트로 변경 로그 구성 항목을 수정합니다.

```
$ dsconf set-server-prop -h host -p port retro-cl-suffix-dn:suffix-DN
```

예를 들어 cn=Contractors,dc=example,dc=com 접미어와 ou=People,dc=example,dc=com 접미어에 대해서만 변경을 기록하려면 다음 명령을 사용합니다.

```
$ dsconf set-server-prop -h host2 -p 1389 \
 retro-cl-suffix-dn:"cn=Contractors,dc=example,dc=com" \
 retro-cl-suffix-dn:"ou=People,dc=example,dc=com"
```

지정된 접미어의 기존 목록에 접미어를 추가하려면 이 명령을 사용합니다.

```
$ dsconf set-server-prop -h host -p port retro-cl-suffix-dn+:suffix-DN
```

2 서버를 다시 시작합니다.

자세한 내용은 59 페이지 "디렉토리 서버 인스턴스 시작, 중지 및 다시 시작"을 참조하십시오.

▼ 삭제된 항목의 속성을 기록하도록 레트로 변경 로그를 구성하는 방법

이 절차에서는 삭제된 항목의 지정된 속성을 기록하도록 레트로 변경 로그를 구성하는 방법에 대해 설명합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 기록해야 할 속성을 지정합니다.

```
$ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr: \
 attribute1 attribute2
```

예를 들어 삭제된 항목의 UID 속성을 기록하도록 레트로 변경 로그를 설정하려면 다음 명령을 사용합니다.

```
$ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr:uid
```

지정된 속성의 기존 목록에 속성을 추가하려면 이 명령을 사용합니다.

```
$ dsconf set-server-prop -h host -p port retro-cl-deleted-entry-attr+:attribute
```

2 서버를 다시 시작합니다.

자세한 내용은 59 페이지 "디렉토리 서버 인스턴스 시작, 중지 및 다시 시작"을 참조하십시오.

▼ 레트로 변경 로그를 지우는 방법

레트로 변경 로그의 항목은 지정된 기간이 경과한 후 자동으로 제거할 수 있습니다. 항목을 자동으로 삭제할 기간을 구성하려면 레트로 변경 로그가 활성화되어 있는지 확인한 다음 `cn=Retro Changelog Plugin, cn=plugins, cn=config` 항목의 `nsslapd-changelogmaxage` 구성 속성을 설정합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 레트로 변경 로그가 활성화되어 있는지 확인합니다.

```
$ dsconf get-server-prop -h host -p port retro-cl-enabled
```

2 레트로 변경 로그가 활성화되지 않은 경우 활성화합니다.

```
$ dsconf set-server-prop -h host -p port retro-cl-enabled:on
```

3 기록된 변경 사항에 대한 최대 사용 기간을 설정합니다.

```
$ dsconf set-server-prop -h host -p port retro-cl-max-age:duration
```

여기서 *duration*은 `undefined`(사용 기간 제한 없음) 또는 다음 중 하나입니다.

- `s` - 초
- `m` - 분
- `h` - 시
- `d` - 일
- `w` - 주

예를 들어 레트로 변경 로그 최대 사용 기간을 2일로 설정하려면 다음을 입력합니다.

```
$ dsconf set-server-prop -h host 2 -p 1389 retro-cl-max-age:2d
```

레트로 변경 로그는 변경 로그에 대한 다음 작업을 수행할 때 지워집니다.

액세스 제어 및 레트로 변경 로그

레트로 변경 로그는 검색 작업을 지원하며, 다음과 같은 형식의 필터를 사용한 검색에 최적화되어 있습니다.

```
(&(changeNumber>=X)(changeNumber<=Y))
```

일반적으로 레트로 변경 로그 항목에서는 추가 또는 수정 작업을 수행하지 않습니다. 항목을 삭제하여 로그 크기를 줄일 수 있습니다. 기본 액세스 제어 정책을 수정하는 경우에만 레트로 변경 로그에 대한 수정 작업을 수행해야 합니다.

레트로 변경 로그가 만들어지고 기본적으로 다음 액세스 제어 규칙이 적용됩니다.

- 레트로 변경 로그 상위 항목인 `cn=changelog`에 대해 인증된 모든 사용자(`userdn=all`인 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.

  레트로 변경 로그 항목에는 비밀번호와 같은 중요한 정보의 수정 사항이 포함될 수 있으므로 익명 사용자에게 읽기 액세스 권한을 부여하지 마십시오. 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 수정합니다. 6 장을 참조하십시오.

복제 상태 가져오기

DSCC 또는 명령줄 도구를 사용하여 복제 상태를 볼 수 있습니다.

DSCC에서 복제 상태 가져오기

접미어 탭을 사용하여 복제 계약, 복제 지연 등을 포함하여 복제를 그래픽으로 볼 수 있습니다. 자세한 내용은 DSCC 온라인 도움말을 참조하십시오.

또한 DSCC를 사용하여 아래 그림과 같이 복제 토폴로지를 볼 수 있습니다.

그림 10–1 복제 토폴로지 예

복제 상태 명령줄 사용 가져오기

DSCC를 사용할 수 없는 경우 명령줄 도구를 사용하여 복제 배포에 대한 정보를 가져옵니다.

설명서 페이지는 이러한 도구의 사용 예와 전체 명령줄 구문을 제공합니다.

- `repldisc` - 복제 배포 시 알려진 모든 서버를 "찾아서" 테이블을 구성합니다. `repldisc`(1) 설명서 페이지를 참조하십시오.
- `insync` - 공급자와 하나 이상의 소비자 복제본 간의 동기화 상태를 나타냅니다. `insync`(1) 설명서 페이지를 참조하십시오.
- `entrycmp` - 두 개 이상의 복제본에 있는 동일 항목을 비교합니다. `entrycmp`(1) 설명서 페이지를 참조하십시오.

이러한 명령이 있는 디렉토리를 찾으려면 34 페이지 “명령 위치”를 참조하십시오.

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. 대부분의 해결은 자동으로 수행됩니다. 예를 들어 각 서버에서 변경과 연관된 타임스탬프는 우선 순위를 가진 최신 변경에 따라 해결됩니다. 일부 변경 충돌은 해결을 위해 수동으로 개입해야 하는 경우도 있습니다.

이 절은 다음 내용으로 구성되어 있습니다.

DSCC를 사용하여 복제 충돌 해결

복제 충돌을 해결하는 가장 쉬운 방법은 DSCC를 사용하는 것입니다. 자세한 내용은 DSCC 온라인 도움말을 참조하십시오.

명령줄을 사용하여 복제 충돌 해결

명령줄을 사용하여 복제 충돌을 해결할 수 있습니다. 복제 프로세스에서 자동으로 해결할 수 없는 변경 충돌이 있는 항목에는 작동 가능 속성인 `nsds5ReplConflict`가 충돌 표식으로 포함되어 있습니다.

충돌하는 항목을 찾으려면 이 속성을 포함하는 항목을 주기적으로 검색합니다. 예를 들어 다음 `ldapsearch` 명령을 사용하여 충돌하는 항목을 찾을 수 있습니다.

```
$ ldapsearch -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config \
 -w - -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 만들면 복제 후에 다음 두 항목이 두 서버에 모두 포함됩니다.

- uid=bjensen,ou=People,dc=example,dc=com
- nsuniqueid=66446001-1dd211b2-66225011-2ee211db+uid=bjensen,dc=example,dc=com

두 번째 항목에 유용한 DN을 지정해야 합니다. 충돌하는 항목을 삭제하고 충돌하지 않는 이름을 사용하여 항목을 다시 추가할 수도 있지만 항목의 이름을 바꾸면 해당 내용이 변경되지 않습니다. 이름 변경 절차는 이름 지정 속성이 한 개 값을 갖는지 또는 여러 값을 갖는지에 따라 달라집니다. 다음 절차를 참조하십시오.

▼ 여러 값을 갖는 이름 지정 속성이 있는 충돌 항목의 이름을 바꾸는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 이전 RDN 값을 유지하면서 항목의 이름을 바꿉니다. 예를 들면 다음과 같습니다.

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: nsuniqueid=66446001-1dd211b2-66225011-2ee211db+uid=bjensen,dc=example,dc=com
changetype: modrdn
newrdn: uid=bj66446001
deleteoldrdn: 0
^D
```

삭제할 수 없는 작동 가능 속성인 nsuniqueid도 포함되어 있으므로 이 단계에서는 이전 RDN 값을 삭제할 수 없습니다.

2 이름 지정 속성의 이전 RDN 값과 충돌 표식 속성을 제거합니다. 예를 들면 다음과 같습니다.

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: uid=bj66446001,dc=example,dc=com
changetype: modify
delete: uid
uid: bjensen
-
delete: nsds5ReplConflict
^D
```

▼ 단일 값을 갖는 이름 지정 속성이 있는 충돌 항목의 이름을 바꾸는 방법

중복 항목의 이름 지정 속성이 한 개의 값을 가지면(예: dc) 항목의 이름을 단순히 동일한 속성의 다른 값으로 바꿀 수 없으므로 임시 이름을 지정해야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **다른 이름 지정 속성을 사용하여 항목의 이름을 바꾸고 이전 RDN을 유지합니다. 예를 들면 다음과 같습니다.**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: nsuniqueid=66446001-1dd211b2-66225011-2ee211db+dc=HR,dc=example,dc=com
changetype: modrdn
newrdn: o=TempHREntry
deleteoldrdn: 0
^D
```

삭제할 수 없는 작동 가능 속성인 `nsuniqueid`도 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 host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: o=TempHREntry,dc=example,dc=com
changetype: modify
replace: dc
dc: NewHR
delete: nsds5ReplConflict
^D
```

3 **항목의 이름을 다시 해당 이름 지정 속성으로 바꿉니다. 예를 들면 다음과 같습니다.**

```
$ ldapmodify -h host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: dc=NewHR,dc=example,dc=com
changetype: modrdn
newrdn: dc=HR
deleteoldrdn: 1
^D
```

`deleteoldrdn` 속성 값을 `1`로 설정하여 임시 속성 값 쌍인 `o=TempName`을 삭제합니다. 이 속성을 유지하려면 `deleteoldrdn` 속성 값을 `0`으로 설정합니다.

고아 항목 충돌 해결

삭제 작업을 복제할 때 사용자 서버에서 삭제할 항목에 자식 항목이 있음을 발견하면 충돌 해결 절차는 **연결** 항목을 만들어 디렉토리에 고아 항목이 발생하지 않도록 방지합니다.

이와 마찬가지로 추가 작업을 복제할 때 사용자 서버에서 부모 항목을 찾을 수 없으면 충돌 해결 절차는 새 항목이 고아 항목이 되지 않도록 부모가 될 연결 항목을 만듭니다.

연결 항목은 glue 및 extensibleObject 객체 클래스를 포함하는 임시 항목으로, 다음과 같은 여러 가지 방법으로 만들 수 있습니다.

- 충돌 해결 절차에서 일치하는 고유 식별자를 발견한 경우에는 해당 항목에 대한 연결 항목이 만들어집니다. 또한 glue 객체 클래스와 nsds5ReplConflict 속성을 포함합니다.

  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 및 extensibleObject 객체 클래스가 있는 최소 항목을 만듭니다.

  이 경우 항목을 수정하여 의미 있는 항목으로 설정하거나 항목 및 해당 자식 항목을 모두 삭제해야 합니다.

잠재적 상호 운용성 문제 해결

메일 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 host2 -p 1389 -D cn=admin,cn=Administrators,cn=config -w -
Enter bind 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.

◆ ◆ ◆ 11 장

11 디렉토리 서버 스키마

디렉토리 서버는 수백 개의 객체 클래스 및 속성이 포함된 표준 스키마와 함께 제공됩니다. 표준 객체 클래스 및 속성만으로도 대부분의 요구 사항을 충족시킬 수 있지만, 새로운 객체 클래스 및 속성을 만들어 스키마를 확장해야 하는 경우도 있습니다. 표준 스키마에 대한 개요와 배포에 적합한 스키마 설계에 대한 지침은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**를 참조하십시오.

이 장은 스키마 관리 방법에 대해 설명하며 다음 내용으로 구성되어 있습니다.

스키마 검사 관리

스키마 검사가 활성화되어 있으면 디렉토리 서버에서 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 디렉토리 서버는 수정되는 속성만이 아닌 전체 항목에 대해 스키마 검사를 수행합니다. 따라서 항목의 객체 클래스 또는 속성이 스키마에 맞지 않으면 작업이 실패할 수 있습니다.

그러나 스키마 검사는 구문과 관련하여 속성 값의 유효성을 확인하지는 않습니다.

스키마 검사는 기본적으로 활성화되며 일반적으로 스키마 검사를 활성화한 상태에서 디렉토리 서버를 실행합니다. 대부분의 클라이언트 응용 프로그램은 스키마 검사를 활성화하면 모든 항목이 스키마에 맞을 것이라고 가정합니다. 그러나 스키마 검사를 활성화해도 디렉토리 서버에서 디렉토리의 기존 내용은 확인되지 않습니다. 모든 디렉토리 내용이 스키마에 맞도록 하려면 항목을 추가하거나 모든 항목을 다시 초기화하기 전에 스키마 검사를 활성화해야 합니다.

스키마에 맞는 LDAP 파일의 가져오기 작업 속도를 향상시키기 위해 예외적으로 스키마 검사를 비활성화할 수도 있습니다. 그러나 스키마 검사가 비활성화되면 스키마에 맞지 않는 항목을 가져올 위험이 있으며 가져온 항목이 스키마에 맞지 않으면 감지되지 않습니다.

복제된 환경에서 스키마 검사를 사용하는 방법에 대한 자세한 내용은 294 페이지 “디렉토리 스키마 복제”를 참조하십시오.

▼ 스키마 호환 문제를 해결하는 방법

항목이 스키마에 맞지 않으면 해당 항목을 검색하지 못할 수 있으며 이 항목에 대한 수정 작업이 실패할 수 있습니다. 이 문제를 해결하려면 이 절차의 단계를 수행합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

시작하기 전에 스키마 호환 문제를 발생시키지 않으려면 배포 전에 스키마를 계획하여 스키마 변경을 최소화합니다. 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**를 참조하십시오.

1. **항목이 스키마에 맞지 않는 이유를 확인하려면 항목을 검색한 다음 현재 정의된 스키마와 수동으로 비교합니다.**

   자세한 내용은 286 페이지 “속성 유형을 보는 방법” 및 289 페이지 “객체 클래스를 보는 방법”을 참조하십시오.

2. **스키마와 맞도록 항목을 수정하거나 항목과 맞도록 스키마를 수정합니다.**

사용자 정의 스키마 정보

표준 스키마가 디렉토리 요구 사항에 대해 너무 제한되어 있는 경우 이를 확장할 수 있습니다. 스키마를 사용자 정의하는 경우 다음 지침을 수행합니다.

- 가능한 경우 기존 스키마 요소를 다시 사용합니다.
- 각 객체 클래스에 정의하는 필수 속성 수를 최소화합니다.
- 객체 클래스 또는 속성을 동일한 용도로 두 개 이상 정의하지 않습니다.
- 스키마를 최대한 단순하게 유지합니다.

스키마를 사용자 정의할 때 표준 스키마에서 속성 또는 객체 클래스의 기존 정의를 수정, 삭제 또는 대체하지 마십시오. 이렇게 하면 다른 디렉토리 및 LDAP 클라이언트 응용 프로그램과의 호환성 문제가 발생할 수 있습니다.

디렉토리 서버의 내부 작업 속성을 수정하지 마십시오. 대신 외부 응용 프로그램에 사용자 고유의 작업 변수를 만들 수 있습니다.

`objectClass: extensibleObject`를 사용하는 대신 항상 객체 클래스를 정의합니다. 디렉토리 서버는 객체 클래스 `extensibleObject`가 있는 항목에 대해 스키마 검사를 수행하지 않으므로 해당 항목에 대한 속성을 제한하거나 검사하지 않습니다. `givenName` 속성 유형에 대해 `giveName`과 같은 응용 프로그램의 오타는 디렉토리 서버에서 감지되지 않습니다. 디렉토리 서버에서는 다른 한 편으로 `extensibleObject` 항목의 정의되지 않은 모든 속성에 여러 값을 가지고 있고, 대소문자를 구분하지 않는 문자열 구문이 있어야 한다고 가정해야 합니다. 또한, 일부 응용 프로그램은 특정 객체 클래스가 있는 항목을 사용합니다. 일반적으로 응용 프로그램에 객체 클래스 확장이 필요한 경우에는 스키마 관리를 계속 유지하고 대신, 응용 프로그램에 필요한 속성이 포함된 보조 객체 클래스를 만듭니다.

이 절은 기본 디렉토리 스키마에 대한 내용과 사용자 정의 속성 및 객체 클래스 만들기에 대한 내용으로 구성되어 있습니다.

기본 디렉토리 서버 스키마

디렉토리 서버와 함께 제공된 스키마는 *instance-path*`/config/schema/` 디렉토리에 저장된 파일 집합에서 설명합니다.

이 디렉토리에는 디렉토리 서버에 대한 일반 스키마 및 관련 제품이 모두 포함되어 있습니다. LDAP v3 표준 사용자 및 조직 스키마는 `00core.ldif` 파일에 있습니다. 이전 버전의 디렉토리에서 사용된 구성 스키마는 `50ns-directory.ldif` 파일에 있습니다.

주 – 서버가 실행 중일 때에는 이 디렉토리에서 파일을 수정하지 마십시오.

객체 식별자

각 LDAP 객체 클래스 또는 속성에는 고유 이름 및 객체 식별자(OID)를 할당해야 합니다. 스키마를 정의하는 경우 해당 조직에 고유한 OID가 필요합니다. OID는 한 개만 있어도 모든 스키마 요구 사항을 충족합니다. 이후에 사용자의 속성 및 객체 클래스에 따라 해당 OID에 새 분기를 추가합니다.

사용자 스키마에서 OID를 얻고 할당하려면 다음 작업을 수행합니다.

- IANA(Internet Assigned Numbers Authority) 또는 정부 기관에서 사용자 조직에 필요한 OID를 얻습니다.

  일부 국가의 경우 OID를 이미 할당받은 기관도 있습니다. 사용자 조직에 아직 OID가 없는 경우 IANA에서 OID를 얻을 수 있습니다.
- OID 할당을 추적할 수 있도록 OID 레지스트리를 만듭니다.

  OID 레지스트리는 사용자가 유지관리하는 목록으로서 디렉토리 스키마에 사용되는 OID 및 OID 설명을 제공합니다. OID 레지스트리는 OID가 두 가지 이상의 용도로 사용되지 않도록 합니다.
- OID 트리에 스키마 요소를 수용하기 위한 분기를 만듭니다.

  속성의 경우 *OID*.1을, 객체 클래스의 경우 *OID*.2를 사용하여 OID 분기 또는 디렉토리 스키마 아래에 둘 이상의 분기를 만듭니다. 사용자 고유의 일치 규칙 또는 컨트롤을 정의하려면 필요에 따라 *OID*.3과 같은 새 분기를 추가할 수 있습니다.

속성 및 객체 클래스 이름 지정

새 속성 및 객체 클래스에 이름을 만드는 경우 스키마에서 사용하기 편하도록 의미있는 이름을 만듭니다.

사용자 정의 요소에 고유 접두어를 포함시켜 사용자 정의 스키마 요소와 기존 스키마 요소 간의 이름 충돌을 방지합니다. 예를 들어 Example.com Corporation의 경우 각 사용자 정의 스키마 요소 앞에 접두어 `Example`을 추가할 수 있습니다. 또한, 해당 디렉토리에서 Example.com 직원을 식별할 수 있도록 `ExamplePerson`이라는 특수 객체 클래스를 추가할 수도 있습니다.

LDAP에서 속성 유형 이름 및 객체 클래스 이름은 **대소문자를 구분하지 않습니다**. 응용 프로그램에서 이 이름은 대소문자를 구분하지 않는 문자열로 처리해야 합니다.

새 객체 클래스를 정의하는 경우

기존 객체 클래스가 디렉토리 항목에 저장해야 할 모든 정보를 지원하지 않는 경우 새 객체 클래스를 추가합니다.

새 객체 클래스를 만드는 방법에는 두 가지가 있습니다.

- 속성을 추가할 각 객체 클래스 구조를 위한 객체 클래스를 포함하여 새 객체 클래스를 여러 개 만듭니다.
- 디렉토리에 대해 만든 모든 속성을 지원하는 단일 객체 클래스를 만듭니다. 이러한 종류의 객체 클래스를 만들려면 해당 클래스를 AUXILIARY 객체 클래스로 정의합니다.

  사용자 사이트에서 ExampleDepartmentNumber 및 ExampleEmergencyPhoneNumber 속성을 만든다고 가정합니다. 이 속성의 일부 하위 집합을 허용하는 객체 클래스를 여러 개 만들 수 있습니다. ExamplePerson이라는 객체 클래스를 만들고 이 클래스가 ExampleDepartmentNumber 및 ExampleEmergencyPhoneNumber 속성을 허용하도록 설정할 수 있습니다. ExamplePerson의 부모는 inetOrgPerson이 됩니다. 그런 다음 ExampleOrganization이라는 객체 클래스를 만들고 이 클래스도 ExampleDepartmentNumber 및 ExampleEmergencyPhoneNumber 속성을 허용하도록 설정할 수 있습니다. ExampleOrganization의 부모는 organization 객체 클래스가 됩니다.

  새 객체 클래스는 다음과 같이 LDAP v3 스키마 형식으로 표시됩니다.

```
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.3 NAME 'ExamplePerson'
 DESC 'Example Person Object Class' SUP inetorgPerson STRUCTURAL MAY
 (ExampleDepartmentNumber $ ExampleEmergencyPhoneNumber) )
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.4 NAME
 'ExampleOrganization' DESC 'Example Organization Object Class' SUP
 organization STRUCTURAL MAY (ExampleDepartmentNumber
 $ ExampleEmergencyPhoneNumber) )
```

  또는 이러한 속성을 모두 허용하는 단일 객체 클래스를 만들 수도 있습니다. 그리고 나면 이 객체 클래스는 속성을 사용할 모든 항목에 사용할 수 있습니다. 단일 객체 클래스는 다음과 같이 표시됩니다.

```
objectclasses: (1.3.6.1.4.1.42.2.27.999.1.2.5 NAME 'ExampleEntry'
 DESC  'Example Auxiliary Object Class' SUP top  AUXILIARY MAY
 (ExampleDepartmentNumber $ ExampleEmergencyPhoneNumber) )
```

  새 ExampleEntry 객체 클래스는 AUXILIARY로 표시되며, 이는 구조적 객체 클래스와는 상관없이 모든 항목에 사용할 수 있다는 것을 의미합니다.

새 객체 클래스를 구현하는 방법을 결정할 때에는 다음 사항을 고려해야 합니다.

- STRUCTURAL 객체 클래스가 여러 개 있으면 스키마 요소를 더 많이 만들고 유지관리해야 할 수 있습니다.

  일반적으로 요소 수를 줄이면 유지관리가 줄어듭니다. 스키마에 세 개 또는 네 개 이상의 객체 클래스를 추가하려는 경우에는 단일 객체 클래스를 사용하는 것이 더 쉽습니다.
- 여러 개의 STRUCTURAL 객체 클래스는 보다 세심하고 견고한 데이터 설계가 필요합니다.

  견고한 데이터 설계를 위해 사용자는 모든 데이터 조각이 있는 객체 클래스 구조를 고려하게 됩니다. 이러한 제한은 유용할 수 있지만 번거로울 수도 있습니다.
- 단일 AUXILIARY 객체 클래스는 둘 이상의 객체 클래스 구조 유형에 데이터를 넣으려 할 경우 데이터 설계를 단순화합니다.

  예를 들어 개인 및 그룹 항목 모두에 preferredOS가 필요하다고 가정합니다. 이 속성을 허용하도록 한 개의 객체 클래스만 만들 수 있습니다.
- 실제 그룹화를 구성하는 그룹 요소 및 실제 객체와 관련된 객체 클래스를 설계합니다.
- 새 객체 클래스의 속성이 요청되지 않도록 합니다.

  속성이 요청되면 스키마의 유연성이 감소될 수 있습니다. 새 객체 클래스를 만들 때 속성을 요청하지 않고 허용하도록 합니다.

  새 객체 클래스를 정의한 후에는 객체 클래스가 어떤 속성을 허용하고 요청하는지와 어떤 객체 클래스에서 상속받는지를 결정해야 합니다.

새 속성을 정의하는 경우

기존 속성이 디렉토리 항목에 저장해야 할 모든 정보를 지원하지 않는 경우 새 속성을 추가합니다. 가능한 경우 표준 속성을 사용합니다. 기본 디렉토리 스키마에 이미 있는 속성을 검색하고 이 속성을 새 객체 클래스와 연관하여 사용합니다.

예를 들어 person, organizationalPerson 또는 inetOrgPerson 객체 클래스의 지원보다는 개인 항목에 대한 자세한 정보를 저장하려고 할 수 있습니다. 디렉토리에 생년월일을 저장하려면 표준 디렉토리 서버 스키마 내에 속성이 없어야 하며 dateOfBirth라는 새 속성을 만들 수 있습니다. 이 속성을 허용하는 새 보조 클래스를 정의하여 이 속성이 사람을 나타내는 항목에 사용될 수 있도록 허용합니다.

사용자 정의 스키마 파일을 만드는 경우

사용자 정의 스키마 파일을 만드는 경우, 특히 복제를 사용하는 경우에는 다음 사항에 유의하십시오.

- 새 스키마 요소를 추가할 때 모든 속성을 객체 클래스에 사용하려면 먼저 해당 속성을 정의해야 합니다. 동일한 스키마 파일에서 속성과 객체 클래스를 정의할 수 있습니다.
- 사용자가 만들 각 사용자 정의 속성 또는 객체 클래스는 한 스키마 파일에서만 정의해야 합니다. 이렇게 하면 서버에서 가장 최근에 만들어진 스키마를 로드할 때 이전 정의를 무시하지 않습니다. 디렉토리 서버는 먼저 숫자순으로 스키마 파일을 로드하고 그 다음 알파벳순으로 파일을 로드합니다.
- 새 스키마 정의를 수동으로 정의할 때에는 일반적으로 이러한 정의를 `99user.ldif` 파일에 추가하는 것이 좋습니다.

  LDAP를 사용하여 스키마 요소를 업데이트하는 경우 새 요소가 `99user.ldif` 파일에 자동으로 쓰여집니다. 따라서, 사용자 정의 스키마 파일에서 변경한 다른 스키마 정의 내용을 덮어쓸 수 있습니다. `99user.ldif` 파일만 사용하면 스키마 요소의 중복 가능성과 스키마 변경 내용을 덮어쓸 수 있는 위험성이 방지됩니다.
- 디렉토리 서버가 먼저 숫자순으로 스키마 파일을 로드하고 그 다음 알파벳순으로 파일을 로드하므로 사용자 정의 스키마 파일의 이름을 다음과 같이 지정해야 합니다.

  `[00-99]` *filename*`.ldif`

  이 숫자는 이미 정의된 디렉토리 표준 스키마보다 큽니다.

  사용자 정의 스키마 파일의 이름이 표준 스키마 파일보다 작은 숫자로 지정되면 스키마를 로드할 때 서버에서 오류가 발생할 수 있으며 또한, 모든 표준 속성과 객체 클래스는 사용자 정의 스키마 요소가 로드된 후에 로드됩니다.
- 디렉토리 서버는 내부 스키마 관리에 우선 순위가 가장 높은 파일을 사용하기 때문에 사용자 정의 스키마 파일의 이름은 `99user.ldif`보다 숫자순으로나 알파벳순으로 우선 순위가 더 높아서는 안 됩니다.

  예를 들어 스키마 파일을 만들고 이름을 `99zzz.ldif`로 지정한 경우 다음에 스키마를 업데이트하면 `X-ORIGIN` 값이 `'user defined'`인 모든 속성이 `99zzz.ldif`에 쓰여집니다. 결과적으로 두 개의 LDIF 파일에 중복된 정보가 포함되며, `99zzz.ldif` 파일의 일부 정보가 지워질 수도 있습니다.

- 일반적으로 추가하는 사용자 정의 스키마 요소는 다음 두 가지 항목으로 식별합니다.
  - 사용자 정의 스키마 파일에서 X-ORIGIN 필드의 'user defined'
  - 다른 관리자가 사용자 정의 스키마 요소를 쉽게 이해할 수 있도록 잘 설명된 X-ORIGIN 필드의 'Example.com Corporation defined'와 같은 레이블(예: X-ORIGIN ('user defined' 'Example.com Corporation defined'))

    스키마 요소를 수동으로 추가하는 경우 X-ORIGIN 필드에 'user defined'를 사용하지 않으면 스키마 요소가 DSCC에서 읽기 전용으로 표시됩니다.

    LDAP 또는 DSCC를 사용하여 사용자 정의 스키마 정의를 추가하면 'user defined' 값이 서버에서 자동으로 추가됩니다. 그러나, X-ORIGIN 필드에 의미가 잘 설명된 값을 추가하지 않으면 나중에 스키마와 관련된 내용을 이해하기 어려울 수 있습니다.

이러한 변경 사항은 자동으로 복제되지 않기 때문에 사용자 정의 스키마 파일을 모든 서버로 수동으로 전달합니다.

디렉토리 스키마를 변경할 때 서버는 스키마가 변경되었을 때의 타임스탬프를 유지합니다. 각 복제 세션 시작 시에 서버는 자신의 타임스탬프와 사용자의 타임스탬프를 비교하여, 필요한 경우 스키마 변경 사항을 푸시합니다. 서버는 사용자 정의 스키마 파일에 대해 99user.ldif 파일과 연관된 타임스탬프를 한 개만 유지관리합니다. 즉, 99user.ldif 이외의 사용자 정의 스키마 파일에 대한 변경 사항 또는 추가 사항은 복제되지 않습니다. 따라서 모든 스키마 정보를 전체 토폴로지에 제공하려면 사용자 정의 스키마 파일을 다른 모든 서버에 전달해야 합니다.

LDAP를 통한 속성 유형 관리

이 절에서는 LDAP를 통해 속성 유형을 만들고 보고 삭제하는 방법에 대해 설명합니다.

속성 유형 만들기

cn=schema 항목에는 디렉토리 스키마에 각 속성 유형의 정의가 포함된, 여러 값을 갖는 속성 attributeTypes가 있습니다. ldapmodify(1) 명령을 사용하면 이러한 정의에 다른 내용을 추가할 수 있습니다.

새 속성 유형 정의 및 사용자 정의 속성 유형에 대한 변경 사항은 99user.ldif 파일에 저장됩니다.

새 속성 유형을 정의하려면 각 속성 유형 정의에 반드시 OID를 입력해야 합니다. 또한 새 속성 유형에 반드시 다음 요소를 사용해야 합니다.

- **속성 OID.** 속성의 객체 식별자에 해당합니다. OID는 스키마 객체를 고유하게 식별하는 문자열이며 대체로 점으로 구분된 10진수로 구성됩니다.

  LDAP v3를 엄격히 준수하려면 유효한 숫자 OID를 입력해야 합니다. OID에 대한 자세한 내용을 보거나 사용자 회사에 대한 접두어를 요청하려면 IANA(Internet Assigned Number Authority)로 전자 메일(`iana@iana.org`)을 보내거나 IANA 웹 사이트(`http://www.iana.org`)를 참조하십시오.
- **속성 이름.** 속성의 고유 이름에 해당하며 속성 유형이라고도 합니다. 속성 이름은 문자로 시작해야 하며 ASCII 문자, 숫자 및 하이픈만 사용할 수 있습니다.

  속성 이름에는 대문자를 사용할 수 있지만, LDAP 클라이언트가 속성을 구별할 목적이라면 대소문자를 사용해서는 안 됩니다. 속성 이름은 RFC 4512(`http://www.ietf.org/rfc/rfc4512.txt`)의 섹션 2.5에 따라 대소문자를 구분하지 않고 처리해야 합니다.

  별칭으로도 참조되는 대체 속성 이름을 속성 유형에 포함시킬 수도 있습니다.
- **속성 설명.** 속성의 용도를 설명하는 짧은 설명 텍스트입니다.
- **구문.** OID에서 참조되며 속성에 포함할 데이터를 설명합니다.

  OID에 따른 속성 구문은 RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`)에 나열되어 있습니다.
- **허용되는 값 수.** 기본적으로 속성은 여러 값을 갖지만 속성이 단일 값을 갖도록 제한할 수 있습니다.

▼ 속성 유형을 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1. **RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`)에 지정된 구문에 따라 속성 유형 정의를 준비합니다.**

2. **`ldapmodify`(1) 명령을 사용하여 속성 유형 정의를 추가합니다.**

   디렉토리 서버에서는 사용자가 입력한 정의에 `X-ORIGIN 'user defined'`를 추가합니다.

예 11–1 속성 유형 만들기

다음 예에서는 `ldapmodify` 명령을 사용하여 디렉토리 문자열 구문으로 새 속성 유형을 추가합니다.

```
$ cat blogURL.ldif
dn: cn=schema
changetype: modify
```

```
add: attributeTypes
attributeTypes: ( 1.2.3.4.5.6.7
 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15
 SINGLE-VALUE )

$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f blogURL.ldif
Enter bind password:
modifying entry cn=schema

$
```

작업 환경에서 1.2.3.4.5.6.7이 아닌 유효한 고유 OID를 입력합니다.

속성 유형 보기

cn=schema 항목에는 디렉토리 스키마에서 각 속성 유형의 정의가 포함된, 여러 값을 갖는 속성 attributeTypes가 있습니다. ldapsearch(1) 명령을 사용하면 이러한 정의를 읽을 수 있습니다.

▼ 속성 유형을 보는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **ldapsearch 명령을 사용하여 현재 디렉토리 스키마에 있는 모든 속성 유형 정의를 봅니다.**

예 11–2 속성 유형 보기

다음 명령을 실행하면 모든 속성 유형 정의가 표시됩니다.

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" attributeTypes
```

-T 옵션을 사용하면 ldapsearch 명령 실행 시 LDIF 줄이 겹치지 않으므로 grep 또는 sed와 같은 명령을 사용하여 출력 작업을 쉽게 수행할 수 있습니다. 그런 다음 grep 명령을 통해 이 명령의 출력을 파이프하면 디렉토리 스키마에 대해 사용자 정의 확장만 볼 수 있습니다. 예를 들면 다음과 같습니다.

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" attributeTypes | grep "user defined"
 attributeTypes: ( 1.2.3.4.5.6.7 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 X-ORIGIN 'user defined' )
```

속성 유형 삭제

cn=schema 항목에는 디렉토리 스키마에서 각 속성 유형의 정의가 포함된, 여러 값을 갖는 속성 attributeTypes가 있습니다. ldapmodify(1) 명령을 사용하면 X-ORIGIN 'user defined'가 포함된 정의를 삭제할 수 있습니다.

스키마는 cn=schema에 있는 LDAP 뷰에서 정의되기 때문에 ldapsearch 및 ldapmodify 유틸리티를 사용하여 온라인으로 스키마를 보고 수정할 수 있습니다. 그러나 X-ORIGIN 필드에 대해서는 'user defined' 값이 있는 스키마 요소만 삭제할 수 있습니다. 서버에서 다른 정의는 삭제되지 않습니다.

사용자 정의 속성에 대한 변경 사항은 99user.ldif 파일에 저장됩니다.

▼ 속성 유형을 삭제하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **삭제할 속성 유형 정의를 봅니다.**
   자세한 내용은 286 페이지 "속성 유형을 보는 방법"을 참조하십시오.

2. **ldapmodify(1) 명령을 사용하여 스키마에 표시된 대로 속성 유형 정의를 삭제합니다.**

예 11–3 속성 유형 삭제

다음 명령을 실행하면 예 11–1에서 만든 속성 유형이 삭제됩니다.

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=schema
changetype: delete
delete: attributeTypes
attributeTypes: ( 1.2.3.4.5.6.7 NAME ( 'blog' 'blogURL' )
 DESC 'URL to a personal weblog'
 SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE
 X-ORIGIN 'user defined' )
^D
```

이 스키마 정의를 확장으로 분류하려면 디렉토리 서버에서 추가된 X-ORIGIN 'user defined'를 포함시켜야 합니다.

LDAP를 통한 객체 클래스 관리

이 절에서는 LDAP를 통해 객체 클래스를 만들고 보고 삭제하는 방법에 대해 설명합니다.

객체 클래스 만들기

cn=schema 항목에는 디렉토리 스키마에서 각 객체 클래스의 정의가 포함된, 여러 값을 갖는 속성 objectClasses가 있습니다. ldapmodify(1) 명령을 사용하면 이러한 정의에 다른 내용을 추가할 수 있습니다.

새 객체 클래스 정의 및 사용자 정의 객체 클래스에 대한 변경 사항은 99user.ldif 파일에 저장됩니다.

다른 객체 클래스에서 상속 받는 객체 클래스를 여러 개 만드는 경우에는 먼저 부모 객체 클래스를 만들어야 합니다. 새 객체 클래스에 사용자 정의 속성이 사용되면 이러한 속성도 먼저 정의해야 합니다.

각 객체 클래스 정의에 반드시 OID를 입력해야 합니다. 또한 새 객체 클래스에 반드시 다음 요소를 사용해야 합니다.

- **객체 클래스 OID.** 객체 클래스의 객체 식별자에 해당합니다. OID는 스키마 객체를 고유하게 식별하는 문자열이며 대체로 점으로 구분된 10진수로 구성됩니다.

  LDAP v3를 엄격히 준수하려면 유효한 숫자 OID를 입력해야 합니다. OID에 대한 자세한 내용을 보거나 사용자 회사에 대한 접두어를 요청하려면 IANA(Internet Assigned Number Authority)로 전자 메일(iana@iana.org)을 보내거나 IANA 웹 사이트(http://www.iana.org)를 참조하십시오.
- **객체 클래스 이름.** 객체 클래스의 고유 이름에 해당합니다.
- **부모 객체 클래스.** 이 객체 클래스가 속성을 상속 받을 기존 객체 클래스입니다.

  이 객체 클래스가 다른 특정 객체 클래스에서 상속 받지 않게 하려면 top을 사용합니다.

  일반적으로 사용자 항목에 새 속성을 추가하려는 경우에는 inetOrgPerson 객체 클래스가 부모입니다. 기업 항목에 새 속성을 추가하려는 경우에는 대체로 organization 또는 organizationalUnit이 부모입니다. 그룹 항목에 새 속성을 추가하려는 경우에는 대체로 groupOfNames 또는 groupOfUniqueNames가 부모입니다.
- **필수 속성.** 이 객체 클래스에 **반드시** 사용해야 하는 속성을 나열하고 정의합니다.
- **허용된 속성.** 이 객체 클래스에 사용할 수 있는 추가 속성을 나열하고 정의합니다.

▼ 객체 클래스를 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **RFC 4517 (http://www.ietf.org/rfc/rfc4517.txt)에 지정된 구문에 따라 객체 클래스 정의를 준비합니다.**

2 **ldapmodify(1) 명령을 사용하여 객체 클래스 정의를 추가합니다.**
디렉토리 서버에서는 사용자가 입력한 정의에 X-ORIGIN 'user defined'를 추가합니다.

예 11–4 객체 클래스 만들기

다음 예에서는 ldapmodify 명령을 사용하여 새 객체 클래스를 추가합니다.

```
$ cat blogger.ldif
dn: cn=schema
changetype: modify
add: objectClasses
objectClasses: ( 1.2.3.4.5.6.8
 NAME 'blogger'
 DESC 'Someone who has a blog'
 SUP inetOrgPerson
 STRUCTURAL
 MAY blog )

$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w - -f blogger.ldif
Enter bind password:
modifying entry cn=schema

$
```

작업 환경에서 1.2.3.4.5.6.8이 아닌 유효한 고유 OID를 입력합니다.

객체 클래스 보기

cn=schema 항목에는 디렉토리 스키마에서 각 객체 클래스의 정의가 포함된, 여러 값을 갖는 속성 objectClasses가 있습니다. ldapsearch(1) 명령을 사용하면 이러한 정의를 읽을 수 있습니다.

▼ 객체 클래스를 보는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **ldapsearch 명령을 사용하여 현재 디렉토리 스키마에 있는 모든 객체 클래스 정의를 봅니다.**

예 11–5 객체 클래스 보기

다음 명령을 실행하면 모든 객체 클래스의 정의가 표시됩니다.

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" objectClasses
```

`-T` 옵션을 사용하면 `ldapsearch` 명령 실행 시 LDIF 줄이 겹치지 않으므로 `grep` 또는 `sed`와 같은 명령을 사용하여 출력 작업을 쉽게 수행할 수 있습니다. 그런 다음 `grep` 명령을 통해 이 명령의 출력을 파이프하면 디렉토리 스키마에 대해 사용자 정의 확장만 볼 수 있습니다. 예를 들면 다음과 같습니다.

```
$ ldapsearch -T -b cn=schema "(objectclass=*)" objectClasses | grep "user defined"
objectClasses: ( 1.2.3.4.5.6.8 NAME 'blogger'
 DESC 'Someone who has a blog' STRUCTURAL MAY blog
 X-ORIGIN 'user defined' )
$
```

객체 클래스 삭제

`cn=schema` 항목에는 디렉토리 스키마에서 각 객체 클래스의 정의가 포함된, 여러 값을 갖는 속성 `objectClasses`가 있습니다. `ldapmodify`(1) 명령을 사용하면 `X-ORIGIN 'user defined'`가 포함된 정의를 삭제할 수 있습니다.

스키마는 `cn=schema`에 있는 LDAP 뷰에서 정의되기 때문에 `ldapsearch` 및 `ldapmodify` 유틸리티를 사용하여 온라인으로 스키마를 보고 수정할 수 있습니다. 그러나 `X-ORIGIN` 필드에 대해서는 '`user defined`' 값이 있는 스키마 요소만 삭제할 수 있습니다. 서버에서 다른 정의는 삭제되지 않습니다.

사용자 정의 요소에 대한 변경 사항은 `99user.ldif` 파일에 저장됩니다.

▼ 객체 클래스를 삭제하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **삭제할 객체 클래스의 정의를 봅니다.**
   자세한 내용은 289 페이지 "객체 클래스를 보는 방법"을 참조하십시오.

2. **`ldapmodify`(1) 명령을 사용하여 스키마에 표시된 대로 객체 클래스 정의를 삭제합니다.**

예 11–6 객체 클래스 삭제

다음 명령을 실행하면 예 11–4에서 만든 객체 클래스가 삭제됩니다.

```
$ ldapmodify -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=schema
changetype: delete
delete: objectClasses
objectClasses: ( 1.2.3.4.5.6.8 NAME 'blogger' DESC 'Someone who has a blog'
 STRUCTURAL MAY blog X-ORIGIN 'user defined' )
^D
```

이 스키마 정의를 확장으로 분류하려면 디렉토리 서버에서 추가된 `X-ORIGIN 'user defined'`를 포함시켜야 합니다.

디렉토리 서버 스키마 확장

스키마에 새 속성을 추가하는 경우 새 속성이 포함될 새 객체 클래스를 만들어야 합니다. 대부분의 필수 속성이 포함되어 있는 기존의 객체 클래스에 속성을 추가하는 것이 더 편리할 수도 있지만 LDAP 클라이언트와의 상호 운용성이 저하되는 단점이 있습니다.

디렉토리 서버와 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.

디렉토리 서버 스키마는 `cn=schema` 항목 속성에 저장됩니다. 구성 항목과 마찬가지로 이 항목은 서버 시작 중에 파일에서 읽은 스키마의 LDAP 뷰입니다.

디렉토리 서버 스키마를 확장하는 데 사용하는 방법은 스키마 확장이 저장되는 파일 이름을 제어할지 여부에 따라 달라집니다. 또한 복제를 통해 변경 사항을 사용자에게 푸시할지 여부에 따라서도 달라집니다. 특정 상황에 맞게 수행할 절차를 결정하려면 다음 표를 참조하십시오.

표 11–1 스키마 확장 방법

작업	지침
복제를 사용하지 않고 사용자 정의 스키마 파일을 추가하여 스키마를 확장합니다.	293 페이지 "사용자 정의 스키마 파일을 사용하여 스키마를 확장하는 방법"
LDAP를 통해 스키마를 확장합니다.	293 페이지 "LDAP를 통해 스키마를 확장하는 방법"
복제를 사용하고 모든 서버에 사용자 정의 스키마 파일의 파일 이름을 유지합니다.	293 페이지 "사용자 정의 스키마 파일을 사용하여 스키마를 확장하는 방법"

표 11–1 스키마 확장 방법 *(계속)*

작업	지침
복제를 사용하고 마스터 복제본에 사용자 정의 스키마 파일을 추가하여 스키마를 확장합니다. 그런 다음 복제 메커니즘을 통해 스키마 확장을 사용자 서버에 복사합니다.	294 페이지 “스키마 파일 및 복제를 사용하여 스키마를 확장하는 방법”

객체 클래스, 속성 및 디렉토리 스키마에 대한 자세한 내용과 스키마 확장에 대한 지침은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 “Designing a Directory Schema”를 참조하십시오. 표준 속성 및 객체 클래스에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Man Page Reference**를 참조하십시오.

이 절에서는 디렉토리 스키마를 확장하는 다양한 방법에 대해 설명합니다.

사용자 정의 스키마 파일을 사용한 스키마 확장

스키마 파일은 *instance-path*/config/schema/에 있는 LDIF 파일입니다. *instance-path*는 디렉토리 서버 인스턴스가 있는 파일 시스템 디렉토리에 해당합니다. 예를 들어 인스턴스는 /local/ds/에 있을 수 있습니다. 이 파일은 디렉토리 서버에서 사용하는 표준 스키마와 디렉토리 서버를 사용하는 모든 서버를 정의합니다. 이 파일과 표준 스키마는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference** 및 **Sun Java System Directory Server Enterprise Edition 6.2 Man Page Reference**에서 설명합니다.

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.

LDIF 파일에 스키마를 직접 정의하는 경우 X-ORIGIN 필드에 'user defined' 값을 사용하지 마십시오. 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 DSCC에서 추가된 모든 스키마 정의가 포함됩니다. 새 스키마 정의가 추가되면 99user.ldif 파일을 덮어쓰기 때문에 이 파일을 수정하려는 경우에는 즉시 서버를 다시 시작하여 변경 사항을 저장해야 합니다.

다른 스키마 파일에 정의된 표준 스키마를 수정할 수는 없지만 대신 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 다음 모든 서버를 다시 시작하여 새 스키마 파일을 로드해야 합니다.

▼ 사용자 정의 스키마 파일을 사용하여 스키마를 확장하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1. **`98mySchema.ldif`와 같은 사용자 고유의 스키마 정의 파일을 만듭니다.**
   스키마 파일의 정의 구문은 RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`)에서 설명합니다.

2. **(옵션) 이 서버가 다른 서버에 업데이트를 전송하는 마스터 복제본인 경우 스키마 정의 파일을 복제 토폴로지의 각 서버 인스턴스에 복사합니다.**
   스키마가 포함된 LDIF 파일에서 직접 변경한 사항은 복제 메커니즘에서 감지할 수 없으므로 마스터를 다시 시작해도 변경 사항이 사용자에 복제되지는 않습니다.

3. **스키마 정의 파일이 복사된 각 디렉토리 서버 인스턴스를 다시 시작합니다.**
   서버가 다시 시작되어 스키마 정의가 다시 로드되면 변경 사항이 적용됩니다.

LDAP를 통한 스키마 확장

스키마는 `cn=schema`에 있는 LDAP 뷰에서 정의되기 때문에 `ldapsearch` 및 `ldapmodify` 유틸리티를 사용하여 온라인으로 스키마를 보고 수정할 수 있습니다. 그러나 `X-ORIGIN` 필드에 대해서는 ’`user defined`’ 값이 있는 스키마 요소만 수정할 수 있습니다. 다른 정의에 대한 수정은 서버에서 모두 거부합니다.

사용자 정의 요소에 대한 변경 사항과 새로운 요소 정의는 `99user.ldif` 파일에 저장됩니다.

▼ LDAP를 통해 스키마를 확장하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

시작하기 전에 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.

1. **`ldapmodify`(1) 명령을 사용하여 개별 `attributeTypes` 속성 값을 추가하거나 삭제합니다.**
   자세한 내용은 285 페이지 “속성 유형을 만드는 방법” 또는 287 페이지 “속성 유형을 삭제하는 방법”을 참조하십시오.

2. **`ldapmodify`(1) 명령을 사용하여 개별 `objectClasses` 속성 값을 추가하거나 삭제합니다.**
   자세한 내용은 288 페이지 “객체 클래스를 만드는 방법” 또는 290 페이지 “객체 클래스를 삭제하는 방법”을 참조하십시오.

참조 값 중 하나를 수정하려면 특정 값을 삭제한 다음 이 값을 새 값으로 추가해야 합니다. 이 프로세스는 속성이 여러 값을 갖기 때문에 필요합니다. 자세한 내용은 94 페이지 "여러 값을 갖는 속성의 값 하나만 수정"을 참조하십시오.

스키마 파일 및 복제를 사용한 스키마 확장

사용자 정의 스키마 파일에 대한 자세한 내용은 292 페이지 "사용자 정의 스키마 파일을 사용한 스키마 확장"을 참조하십시오. 다음 절차에서는 복제 메커니즘을 사용하여 스키마 확장을 토폴로지의 모든 서버에 전달하는 방법에 대해 설명합니다.

▼ 스키마 파일 및 복제를 사용하여 스키마를 확장하는 방법

이 절차의 일부로, DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오. 해당 절차의 다른 부분은 명령줄에서만 수행할 수 있습니다.

1 다음 방법 중 하나를 사용하여 스키마 확장을 준비합니다.

- **`98mySchema.ldif`와 같은 사용자 고유의 스키마 정의 파일을 만듭니다.**
- **스키마 확장을 `99user.ldif`에 추가합니다.**

스키마 파일의 정의 구문은 RFC 4517 (`http://www.ietf.org/rfc/rfc4517.txt`)에서 설명합니다.

2 스키마 정의 파일이 있는 마스터 서버에서 `schema_push` 명령을 실행합니다.

이 스크립트는 실제로 스키마를 복제본으로 푸시하지는 않습니다. 대신 스크립트는 스키마 파일이 로드되는 즉시 복제되도록 특수 속성을 스키마 파일에 씁니다. 자세한 내용은 `schema_push`(1M) 설명서 페이지를 참조하십시오.

3 스키마 정의 파일이 있는 마스터 서버를 다시 시작합니다.

스키마가 포함된 LDIF 파일에서 직접 변경한 사항은 복제 메커니즘에서 감지할 수 없으므로 그러나 `schema_push`를 실행한 후 서버를 다시 시작하면 서버에서 모든 스키마 파일이 로드된 후 복제 메커니즘이 새 스키마를 사용자에 복제합니다.

디렉토리 스키마 복제

두 서버 간에 하나 이상의 접미어 복제를 구성하면 스키마 정의도 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.

그러나 스키마 복제는 LDAP를 통해 스키마를 수정할 때에도 즉시 수행되지 않습니다. 스키마 복제는 디렉토리 데이터에 대한 업데이트 시 또는 스키마가 수정된 후 첫 번째 복제 세션 시작 시에 실행됩니다.

모든 복제본에서 스키마를 실행하려면 반드시 모든 마스터에서 스키마 검사를 활성화해야 합니다. LDAP 작업이 수행되는 마스터에서 스키마를 검사하기 때문에 사용자를 업데이트할 때는 스키마를 검사할 필요가 없습니다. 성능을 향상시키기 위해 복제 메커니즘은 소비자 복제본에 대한 스키마 검사를 무시합니다.

주 – 허브 및 전용 사용자에서는 스키마 검사를 비활성화하지 마십시오. 스키마 검사는 사용자 성능에 영향을 주지 않으므로 복제본 내용이 스키마에 맞는지 나타내려면 스키마 검사를 계속 활성화 상태로 유지합니다.

마스터 서버는 사용자 초기화 중에, 그리고 DSCC 또는 명령줄 도구를 통해 스키마를 수정할 때에 자동으로 스키마를 해당 사용자에 복제합니다. 기본적으로 전체 스키마가 복제되며, 사용자에 없는 추가 스키마 요소는 사용자에 새로 만들어져 `99user.ldif` 파일에 저장됩니다.

예를 들어 마스터 서버를 시작할 때 해당 서버에 `98mySchema.ldif` 파일의 스키마 정의가 포함되고 이후 다른 서버(마스터, 허브 또는 전용 사용자)에 대한 복제 계약을 정의한다고 가정합니다. 나중에 이 마스터에서 복제본을 초기화하면 복제된 스키마에는 `98mySchema.ldif`의 정의가 포함되지만 이 정의는 복제본 서버의 `99user.ldif`에 저장됩니다.

스키마가 사용자 초기화 중에 복제된 경우 마스터의 `cn=schema`에 있는 스키마를 수정해도 전체 스키마가 사용자에 복제되므로 명령줄 유틸리티나 DSCC를 통한 마스터 스키마의 모든 수정 사항이 사용자에 복제됩니다. 이러한 수정 사항은 마스터의 `99user.ldif`에 저장되며, 또한 이전 설명과 동일한 메커니즘으로 사용자의 `99user.ldif`에도 저장됩니다.

복제된 환경에서 스키마의 일관성을 유지하려면 다음 지침을 수행합니다.

- 사용자 서버의 스키마는 수정하지 마십시오.

  사용자 서버의 스키마를 수정하면 복제 오류가 발생할 수 있습니다. 사용자의 스키마가 변경되면 공급자로부터의 업데이트가 사용자의 스키마와 맞지 않을 수 있기 때문입니다.
- 다중 마스터 복제 환경에서는 단일 마스터 서버의 스키마를 수정합니다.

  두 개의 마스터 서버에서 스키마를 수정하면 최신으로 업데이트된 마스터에서 해당 버전의 스키마를 사용자에 전달합니다. 그러면 사용자의 스키마가 다른 마스터의 스키마와 일치하지 않을 수 있습니다.

단편 복제를 구성할 때에는 다음 사항도 고려해야 합니다.

- 단편 복제 구성에서는 공급자가 스키마를 푸시하므로 단편 소비자 복제본의 스키마는 마스터 복제본 스키마의 복사본입니다. 따라서, 스키마가 현재 적용 중인 단편 복제 구성에 맞지 않을 수 있습니다.
- 일반적으로 디렉토리 서버는 스키마 위반을 방지하기 위해 스키마에 정의된 대로 모든 필수 속성을 각 항목에 복제합니다. 필수 속성을 필터링하도록 단편 복제를 구성하는 경우 스키마 검사를 비활성화해야 합니다.
- 단편 복제 시 스키마 검사를 활성화하면 복제본을 오프라인으로 초기화하지 못할 수 있습니다. 필수 속성이 필터링되면 디렉토리 서버를 통해 LDIF에서 데이터를 로드할 수 없습니다.
- 단편 소비자 복제본에 대한 스키마 검사를 비활성화하면 단편 소비자 복제본이 있는 전체 서버 인스턴스에서 스키마 검사가 실행되지 않습니다. 따라서, 단편 사용자와 동일한 서버 인스턴스에는 공급자 복제본을 구성하지 마십시오.

스키마 복제 제한

기본적으로 복제 메커니즘은 스키마를 복제할 때 항상 전체 스키마를 사용자에게 보냅니다. 다음 두 가지 상황과 같이 전체 스키마를 사용자에게 보내지 않아야 하는 경우도 있습니다.

- DSCC 또는 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.
- 디렉토리 서버의 마스터를 Directory Server 5.1의 사용자에 복제하면 두 버전의 구성 속성 스키마가 다르기 때문에 충돌이 발생합니다. 이 경우 **반드시** 사용자 정의 스키마 요소만 복제해야 합니다.

주 – 디렉토리 서버는 `11rfc2307.ldif` 스키마 파일을 사용합니다. 이 스키마 파일은 RFC 2307 (`http://www.ietf.org/rfc/rfc2307.txt`)을 준수합니다.

Directory Server 5.2 이전 버전에서는 `10rfc2307.ldif` 스키마 파일을 사용합니다.

▼ 스키마 복제를 제한하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

● **사용자 정의 스키마만 복제되도록 스키마 복제를 제한합니다.**

```
$ dsconf set-server-prop -h host -p port repl-user-schema-enabled:on
```

기본값 `off`를 설정하면 필요한 경우 전체 스키마가 복제됩니다.

◆ ◆ ◆ 12 장

12 디렉토리 서버 색인화

책 색인과 마찬가지로 디렉토리 서버 색인은 검색 문자열을 디렉토리 내용에 대한 참조와 연결하여 검색 속도를 향상시킵니다.

색인 유형 및 색인 조정에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 6 장, "Directory Server Indexing"을 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

색인 관리

이 절에서는 특정 속성에 대한 색인을 관리하는 방법에 대해 설명합니다. 또한 색인 만들기, 수정 및 삭제 관련 정보가 들어 있습니다. 가상 목록 보기(VLV) 작업에 해당하는 절차는 303 페이지 "찾아보기 색인 관리"를 참조하십시오.

▼ 색인을 나열하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **기존 색인과 해당 등록 정보를 나열하려면 다음 명령을 사용합니다.**

```
$ dsconf list-indexes -h host -p port -v suffix-DN
```

▼ 색인을 만드는 방법

주 – 새 시스템 색인은 작성할 수 없습니다. 디렉토리 서버에서 내부적으로 정의한 기존의 시스템 색인만 유지관리됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **새 색인 구성을 만듭니다.**

색인화할 속성을 지정하여 새 색인 정보를 구성하려면 `dsconf create-index` 명령줄 유틸리티를 사용합니다.

예를 들어 `preferredLanguage` 속성에 대한 색인 항목을 만들려면 다음 명령을 사용합니다.

```
$ dsconf create-index -h host -p port dc=example,dc=com preferredLanguage
```

주 – `dsconf create-index` 명령은 색인 구성을 설정하지만, 검색에 필요한 색인 파일을 실제로 만들지는 않습니다. 색인 파일을 생성하면 성능에 영향을 줄 수 있습니다. 색인화 절차를 보다 자세히 제어하려면 새 색인 구성을 만든 후 색인 파일을 수동으로 생성합니다.

색인을 만들 경우 속성의 기본 이름을 항상 사용합니다. 속성의 별칭을 사용하지 마십시오. 기본 속성 이름은 스키마에서 해당 속성에 지정된 이름입니다(예: `userid` 속성의 경우 `uid`).

2 **(옵션) `dsconf set-index-prop` 명령을 사용하여 색인 등록 정보를 설정합니다.**

`dsconf create-index` 명령은 기본 등록 정보를 사용하여 색인을 만듭니다. 이러한 등록 정보를 수정하려면 `dsconf set-index-prop` 명령을 사용합니다. 색인 등록 정보 수정에 대한 자세한 내용은 298 페이지 "색인을 수정하는 방법"을 참조하십시오.

3 **색인 파일을 생성합니다.**

299 페이지 "색인을 생성하는 방법"을 참조하십시오.

4 **색인화할 모든 서버에 대해 이전 단계를 반복합니다.**

▼ 색인을 수정하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **색인 등록 정보를 수정합니다.**

   ```
   $ dsconf set-index-prop -h host -p port suffix-DN attr-name property:value
   ```

   예를 들어 `preferredLanguage` 색인에 대한 근사 색인 `approx-enabled`를 활성화하려면 다음 명령을 사용합니다.

```
$ dsconf set-index-prop -h host -p port dc=example,dc=com preferredLanguage approx-enabled:on
```

   각 색인에 대해 다음 등록 정보를 수정할 수 있습니다.

   - `eq-enabled` 동일
   - `pres-enabled` 존재
   - `sub-enabled` 하위 문자열

   수정할 등록 정보 중 하나가 선택적 `nsMatchingRule` 속성입니다. 이 속성에는 서버에 알려진 일치 규칙에 대한 OID가 포함되어 있습니다. 또한 국가별 색인에 대한 언어 조합 순서의 OID를 비롯하여 `CaseExactMatch`와 같은 기타 일치 규칙을 사용할 수 있습니다. 관련된 조사 순서의 OID와 지원되는 로켈 목록은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

   색인 구성 속성에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

2. **새 색인을 다시 생성합니다.**

   299 페이지 "색인을 생성하는 방법"을 참조하십시오.

3. **수정된 속성 색인을 포함하는 모든 서버에 대해 이전 단계를 반복합니다.**

▼ 색인을 생성하는 방법

이 절차에서는 새 색인이나 수정된 색인을 검색할 수 있도록 색인 파일을 생성합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **다음 중 한 가지 방법으로 색인 파일을 생성합니다.**
  - 온라인으로 새 색인 파일을 생성합니다.

    ```
    $ dsconf reindex -h host -p port [-t attr] suffix-DN
    ```

    여기서 `-t`는 모든 속성이 아니라 지정된 속성만 다시 색인화하도록 지정합니다.

    예를 들어 `preferredLanguage` 색인을 다시 생성하려면 다음을 입력합니다.

    ```
    $ dsconf reindex -h host -p port -t preferredLanguage dc=example,dc=com
    ```

dsconf reindex 명령을 실행하는 동안 접미어 내용이 서버에서 사용 가능한 상태로 유지됩니다. 그러나, 검색은 명령이 완료될 때까지 색인화되지 않습니다. 다시 색인화하려면 상당한 자원이 필요하므로 서버의 다른 작업 성능이 저하될 수 있습니다.

- 오프라인으로 새 색인 파일을 생성합니다.

  ```
  $ dsadm reindex -t attr instance-path suffix-DN
  ```

  예를 들어 preferredLanguage 색인을 다시 생성하려면 다음을 입력합니다.

  ```
  $ dsadm reindex -t preferredLanguage /local/ds dc=example,dc=com
  ```

- 접미어를 다시 초기화하여 모든 색인을 오프라인으로 빠르게 다시 생성합니다.

  접미어를 다시 초기화하면 모든 색인 파일이 자동으로 다시 생성됩니다. 디렉토리의 크기에 따라 접미어를 다시 초기화하는 것이 두 개 이상의 속성을 다시 색인화하는 것보다 더 빠릅니다. 그러나 초기화 중에는 접미어를 사용할 수 없습니다. 자세한 내용은 302 페이지 "다시 초기화하여 접미어 다시 색인화"를 참조하십시오.

주 – dsconf import 또는 dsconf reindex 명령 중 하나 또는 모두를 여러 접미어에서 병렬로 실행할 경우 트랜잭션 로그가 계속 증가하여 성능이 저하될 수 있습니다.

▼ 색인을 삭제하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **속성에 대해 구성된 모든 색인을 제거합니다.**

  ```
  $ dsconf delete-index -h host -p port suffix-DN attr-name
  ```

  예를 들어 다음 명령은 preferredLanguage 속성에 대한 모든 색인을 삭제합니다.

  ```
  $ dsconf delete-index -h host -p port dc=example,dc=com preferredLanguage
  ```

  기본 색인을 삭제하면 디렉토리 서버 기능에 영향을 줄 수 있으므로 주의하십시오.

색인 목록 임계값 변경

시스템 색인 목록 크기가 색인 목록 임계값을 초과하면 검색이 느려질 수 있습니다. 색인 목록 임계값은 각 색인 키 값의 최대 수입니다. 색인 목록 임계값 크기를 초과했는지 확인하려면 액세스 로그를 확인하십시오. 액세스 로그 RESULT 메시지의 끝에 있는 notes=U 플래그는 색인화되지 않은 검색이 수행되었음을 나타냅니다. 동일한 연결 및 작업에 대한 이전 SRCH 메시지는 사용된 검색 필터를 지정합니다. 아래 두 줄의 예에서는 10000개 항목을 반환하는 색인화되지 않은 cn=Smith 검색을 추적합니다. 타임스탬프는 메시지에서 제거되었습니다.

```
conn=2 op=1 SRCH base="o=example.com" scope=0 filter="(cn=Smith)"
conn=2 op=1 RESULT err=0 tag=101 nentries=10000 notes=U
```

시스템에서 색인 목록 임계값이 자주 초과되는 경우 성능 향상을 위해 임계값을 높이십시오. 다음 절차에서는 `dsconf set-server-prop` 명령을 사용하여 `all-ids-threshold` 등록 정보를 수정합니다. 색인 및 `all-ids-threshold` 등록 정보 조정에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Tuning Indexes for Performance"를 참조하십시오.

▼ 색인 목록 임계값을 변경하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 색인 목록 임계값을 조정합니다.

다음 수준으로 색인 목록 임계값을 조정할 수 있습니다.

- 인스턴스 수준에서:

  ```
  dsconf set-server-prop -h host -p port all-ids-threshold:value
  ```

- 접미어 수준에서:

  ```
  dsconf set-suffix-prop -h host -p port suffix-DN all-ids-threshold:value
  ```

- 항목 수준에서:

  ```
  dsconf set-index-prop -h host -p port suffix-DN all-ids-threshold:value
  ```

- 색인 수준에서, 검색 유형별:

  ```
  dsconf set-index-prop -h host -p port suffix-DN all-ids-threshold search-type:value
  ```

  여기서 *search-type***은 다음 중 하나입니다**.

  - `eq-enabled` 동일
  - `pres-enabled` 존재
  - `sub-enabled` 하위 문자열

  근사 색인에 대해서는 `all-ids-threshold` 등록 정보를 구성할 수 없습니다.

DSCC를 사용하여 색인 수준에서 검색 유형별로 임계값을 설정할 수 있습니다. 자세한 내용은 디렉토리 서버 온라인 도움말을 참조하십시오.

2 접미어 색인을 다시 생성합니다.

299 페이지 "색인을 생성하는 방법"을 참조하십시오.

3 데이터베이스 캐시 크기를 이전 `all IDs` 임계값에 대해 조정했고 서버에 적절한 물리적 메모리가 있는 경우 데이터베이스 캐시 크기를 늘려 보십시오.

데이터베이스 캐시 크기를 `all IDs` 임계값 증분의 25%만큼 늘립니다.

즉, `all IDs` 임계값을 4000에서 6000으로 늘리면 색인 목록 크기 증가를 수용하기 위해 데이터베이스 캐시 크기를 약 12 ½%만큼 늘릴 수 있습니다.

데이터베이스 캐시 크기는 `dbcachesize` 속성을 사용하여 설정합니다. 생산 서버에 변경 사항을 적용하기 전에 실제 테스트를 통해 최적 크기를 확인합니다.

접미어 다시 색인화

색인 파일이 손상된 경우 접미어를 다시 색인화하여 해당 데이터베이스 디렉토리에서 색인 파일을 다시 만들어야 합니다. 디렉토리 서버가 실행 중인 동안 접미어를 다시 색인화하거나 다시 초기화하여 접미어를 다시 색인화할 수 있습니다.

디렉토리 서버가 실행 중인 동안 접미어 다시 색인화

접미어를 다시 색인화하면 서버는 접미어에 포함된 모든 항목을 확인하여 색인 파일을 다시 작성합니다. 다시 색인화하는 동안 접미어의 내용은 읽기 전용입니다. 서버는 다시 색인화하는 모든 속성에 대한 전체 접미어를 검사해야 하므로 수백 만 개의 항목으로 구성된 접미어의 경우 이 프로세스를 수행하는 데 몇 시간이 걸릴 수도 있습니다. 시간은 구성하는 색인에 따라 다릅니다. 또한 접미어를 다시 색인화하는 동안에는 색인을 사용할 수 없으며 서버 성능에 영향을 줍니다.

▼ 접미어의 모든 색인을 다시 색인화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **접미어의 모든 색인을 다시 색인화합니다.**

  ```
  $ dsconf reindex -h host -p port suffix-DN
  ```

  예를 들어 `dc=example,dc=com` 접미어의 모든 색인을 다시 초기화하려면 다음 명령을 사용합니다.

  ```
  $ dsconf reindex -h host -p port dc=example,dc=com
  ```

다시 초기화하여 접미어 다시 색인화

접미어를 다시 초기화하면 새 내용을 가져와서 접미어 내용을 교체하고 새 색인 파일이 만들어집니다. 항목을 로드하면 모든 속성이 병렬로 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.

▼ 다시 초기화하여 접미어를 다시 색인화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1. **63 페이지 "참조 설정 및 접미어 읽기 전용 만들기"에 설명된 것처럼 접미어를 읽기 전용으로 설정합니다.**

2. **200 페이지 "LDIF에 백업"에 설명된 것처럼 전체 접미어를 LDIF 파일로 내보냅니다.**

3. **203 페이지 "LDIF 파일에서 데이터 가져오기"에 설명된 것처럼 동일한 LDIF 파일을 가져와서 접미어를 다시 초기화합니다.**

   초기화하는 동안에는 접미어를 사용할 수 없습니다. 초기화가 완료되면 구성된 모든 색인을 사용할 수 있습니다.

4. **63 페이지 "참조 설정 및 접미어 읽기 전용 만들기"에 설명된 것처럼 접미어를 쓰기 가능으로 다시 지정합니다.**

찾아보기 색인 관리

찾아보기 색인은 서버측 결과 정렬을 요청하는 검색 작업에만 사용되는 특수 색인입니다. 디렉토리 서버에서 찾아보기 색인을 사용하는 방법은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**를 참조하십시오.

클라이언트 검색에 대한 찾아보기 색인

클라이언트 검색 결과를 정렬하기 위한 사용자 정의 찾아보기 색인을 수동으로 정의해야 합니다. 찾아보기 색인 또는 가상 목록 보기(VLV) 색인을 만들려면 다음 절차를 사용합니다. 이 절에서는 찾아보기 색인 항목을 추가하거나 수정하는 절차와 찾아보기 색인을 다시 생성하는 절차에 대해서도 설명합니다.

▼ 찾아보기 색인을 만드는 방법

이 절차의 일부로, DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오. 해당 절차의 다른 부분은 명령줄에서만 수행할 수 있습니다.

1. **`ldapmodify` 명령을 사용하여 새 찾아보기 색인 항목을 추가하거나 기존 찾아보기 색인 항목을 편집합니다.**

   자세한 내용은 304 페이지 "찾아보기 색인 항목을 추가하거나 수정하는 방법"을 참조하십시오.

2. **`dsconf reindex` 명령을 실행하여 서버에서 유지관리할 새 찾아보기 색인 집합을 생성합니다.**

   자세한 내용은 305 페이지 "찾아보기 색인을 다시 생성하는 방법"을 참조하십시오.

▼ 찾아보기 색인 항목을 추가하거나 수정하는 방법

찾아보기 색인은 지정된 기본 항목 및 해당 하위 트리별로 작성됩니다. 또한, 찾아보기 색인 구성은 이 항목이 포함된 접미어의 데이터베이스 구성에 정의됩니다.

1 디렉토리 서버의 각 찾아보기 색인에 대한 `vlvBase`, `vlvScope` 및 `vlvFilter` 속성을 구성합니다.

이러한 속성은 검색 기준, 검색 범위 및 검색 필터를 구성합니다. 이러한 속성은 `vlvSearch` 객체 클래스를 사용합니다.

2 각 찾아보기 색인에 대한 `vlvSort` 속성을 구성합니다.

이 속성은 색인을 정렬하는 속성의 이름을 지정합니다. 이 항목은 첫 번째 항목의 자식이며 `vlvIndex` 객체 클래스를 사용하여 정렬할 속성과 순서를 지정합니다.

아래 예에서는 `ldapmodify` 명령을 사용하여 찾아보기 색인 구성 항목을 만듭니다.

```
$ ldapmodify -a -h host -p port -D cn=admin,cn=Administrators,cn=config -w -
Enter bind password:
dn: cn=people_browsing_index, cn=database-name,
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=people_browsing_index,
 cn=database-name,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
cn: Sort rev employeenumber
vlvSort: -employeenumber
^D
```

`vlvScope`는 다음 중 하나입니다.

- `0` - 기본 항목만 검색하는 경우
- `1` - 기본 항목의 직계 자식만 검색하는 경우
- `2` - 기본 항목을 루트로 하는 전체 하위 트리를 검색하는 경우

`vlvFilter`는 클라이언트 검색 작업에 사용되는 것과 동일한 LDAP 필터입니다. 찾아보기 색인 항목은 모두 같은 위치에 있으므로 항목을 잘 설명하는 `cn` 값을 사용하여 찾아보기 색인 이름을 지정하는 것이 좋습니다.

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=admin,cn=Administrators,cn=config -w -
dn: cn=Sort sn givenname uid, cn=people_browsing_index,
 cn=database-name,cn=ldbm database,cn=plugins,cn=config
objectClass: top
objectClass: vlvIndex
cn: Sort sn givenname uid
vlvSort: sn givenname uid
^D
```

3. **찾아보기 색인 구성을 수정하려면 해당 vlvSearch 항목 또는 vlvIndex 항목을 편집합니다.**

4. **찾아보기 색인이 더 이상 서버에서 유지관리되지 않도록 찾아보기 색인을 제거하려면 개별 vlvIndex 항목을 제거합니다.**

   vlvIndex 항목이 하나만 있는 경우에는 vlvSearch 항목과 vlvIndex 항목을 모두 제거합니다.

▼ 찾아보기 색인을 다시 생성하는 방법

- **찾아보기 색인 항목을 만든 후 지정된 속성에 대한 새 찾아보기 색인을 생성합니다.**

  ```
  $ dsadm reindex -l -t attr-index instance-path suffix-DN
  ```

  이 명령은 디렉토리 내용을 검사하여 찾아보기 색인에 대한 데이터베이스 파일을 만듭니다.

  아래 예에서는 이전 절에서 정의한 찾아보기 색인을 생성합니다.

  ```
  $ dsadm reindex -l -b database-name -t Browsing /local/ds \
   ou=People,dc=example,dc=com
  ```

  dsadm reindex 명령에 대한 자세한 내용은 dsadm(1M) 설명서 페이지를 참조하십시오.

◆ ◆ ◆ 13 장

13 디렉토리 서버 속성 값 고유성

UID 고유성 플러그인을 사용하면 주어진 속성 값이 디렉토리 또는 하위 트리의 모든 항목에서 고유하도록 만들 수 있습니다. 이 플러그인을 사용하면 지정된 속성에 대한 기존 값이 포함된 항목을 추가하려는 작업이나 속성을 디렉토리에 있는 값으로 추가 또는 수정하려는 작업이 모두 중지됩니다.

UID 고유성 플러그인은 기본적으로 비활성화되어 있습니다. 이 플러그인을 활성화하면 기본적으로 `uid` 속성의 고유성이 유지됩니다. 플러그인의 새 인스턴스를 만들어 다른 속성에 대한 고유 값을 적용할 수도 있습니다. UID 고유성 플러그인은 단일 서버에서 속성 값 고유성을 유지합니다.

이 장은 다음 내용으로 구성되어 있습니다.

속성 값 고유성 개요

UID 고유성 플러그인은 사전 작업 플러그인으로, 서버에서 디렉토리를 업데이트하기 전에 LDAP 추가, 수정 및 DN 수정 작업을 검사하여 두 항목이 동일한 속성 값을 갖게 되는지 여부를 확인합니다. 동일한 속성 값을 갖게 되면 서버는 작업을 종료하고 오류 19 `LDAP_CONSTRAINT_VIOLATION`을 클라이언트로 반환합니다.

디렉토리에 있는 하나 이상의 하위 트리 또는 특정 객체 클래스 항목에 고유성이 적용되도록 플러그인을 구성할 수 있습니다. 이 구성은 속성 값에 대해 고유성을 적용할 항목 집합을 지정합니다.

다른 속성에 대한 고유성을 적용하려면 UID 고유성 플러그인의 인스턴스를 여러 개 정의합니다. 값이 고유해야 하는 각 속성에 한 개의 플러그인 인스턴스를 정의합니다. 동일한 속성에 여러 개의 플러그인 인스턴스를 정의하여 각 항목 집합에 "별도의" 고유성을 적용할 수도 있습니다. 지정된 속성 값은 각 하위 트리 집합에서 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 복제에 영향을 주기 때문에 기본적으로 비활성화됩니다. 복제 사용 시 UID 고유성 플러그인을 활성화할 수도 있지만 310 페이지 "복제 시 고유성 플러그인 사용"에 설명된 동작에 주의해야 합니다.

uid 및 다른 속성에 대한 고유성 적용

이 절에서는 uid 속성에 대한 기본 고유성 플러그인을 활성화하고 구성하는 방법 및 다른 속성에 대한 고유성을 적용하는 방법에 대해 설명합니다.

▼ uid 속성에 대한 고유성을 적용하는 방법

다음 절차에서는 dsconf 명령을 사용하여 UID 고유성 플러그인을 활성화하고 구성하는 방법에 대해 설명합니다. 플러그인 구성 항목의 DN은 cn=uid uniqueness,cn=plugins,cn=config입니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

DSCC를 사용하는 경우 다른 속성에 대한 고유성을 적용하기 위해 기본 UID 고유성 플러그인을 수정해서는 안 됩니다. UID 고유성 플러그인을 사용하지 않으려면 플러그인을 비활성화된 상태로 두고 309 페이지 "다른 속성에 대한 고유성을 적용하는 방법"에 설명된 것처럼 다른 속성에 대한 새 플러그인 인스턴스를 만듭니다.

1 **플러그인을 활성화합니다.**

```
$ dsconf enable-plugin -h host -p port "uid uniqueness"
```

2 **고유성을 적용할 하위 트리의 지정 방법에 따라 플러그인 인수를 수정합니다.**

- 단일 하위 트리의 기본 DN을 지정하려면 다음을 입력합니다.

```
$ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:uid argument:subtreeBaseDN
```

예를 들면 다음과 같습니다.

```
$ dsconf set-plugin-prop -h host1 -p 1389 "uid uniqueness" argument:uid \
 argument:dc=People,dc=example,dc=com
```

- 하위 트리를 두 개 이상 지정하려면 다음 명령을 실행하여 하위 트리의 전체 기본 DN이 각 인수 값으로 지정된 인수를 추가합니다.

```
$ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:uid \
 argument:subtreeBaseDN argument:subtreeBaseDN
```

- 기본 항목의 객체 클래스에 따라 하위 트리를 지정하려면 인수를 아래의 값으로 설정합니다. uid 속성에 대한 고유성은 *baseObjectClass*가 포함된 항목 아래의 하위 트리에서 적용됩니다. 이 객체 클래스가 있는 항목을 대상으로 하는 작업에만 고유성을 적용하도록 선택 사항으로 세 번째 인수에 *entryObjectClass*를 지정할 수도 있습니다.

  ```
  $ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument:attribute=uid \
   argument:markerObjectClass=baseObjectClass argument:entryObjectClass=baseObjectClass
  ```

- 기존 인수 목록에 인수를 추가하려면 다음 명령을 사용합니다.

  ```
  $ dsconf set-plugin-prop -h host -p port "uid uniqueness" argument+:argument-value
  ```

3 변경 사항을 적용하려면 서버를 다시 시작합니다.

▼ 다른 속성에 대한 고유성을 적용하는 방법

UID 고유성 플러그인을 사용하여 모든 속성에 대한 고유성을 적용할 수 있습니다. 디렉토리에서 cn=plugins,cn=config 아래에 새 항목을 작성하여 플러그인의 새 인스턴스를 만들어야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 새 플러그인을 만듭니다.

```
$ dsconf create-plugin -h host -p port -H lib-path -F init-func \
 -Y type plugin-name
```

*plugin-name*은 속성 이름이 포함된, 자신을 잘 나타내는 짧은 이름이어야 합니다. 예를 들어 메일 아이디 속성에 대한 고유성 플러그인을 만들려면 다음 명령을 사용합니다.

```
$ dsconf create-plugin -h host1 -p 1389 -H /opt/SUNWdsee/ds6/lib/uid-plugin.so \
 -F NSUniqueAttr_Init -Y preoperation "mail uniqueness"
```

2 플러그인 등록 정보를 설정합니다.

```
$ dsconf set-plugin-prop -h host -p port plugin-name property:value
```

예를 들어 메일 고유성 플러그인에 대한 등록 정보를 설정하려면 다음 명령을 사용합니다.

```
$ dsconf set-plugin-prop -h host1 -p 1389 "mail uniqueness" \
 desc:"Enforce unique attribute values..." version:6.0 \
 vendor:"Sun Microsystems, Inc." depends-on-type:database
```

3 플러그인을 활성화합니다.

```
$ dsconf enable-plugin -h host -p port plugin-name
```

4 **플러그인 인수를 지정합니다.**

이 인수는 고유성을 적용할 하위 트리를 지정하는 방법에 따라 달라집니다.

- 기본 DN을 따라 하위 트리를 하나 이상 정의하려면 첫 번째 인수는 고유한 값이 있어야 하는 속성의 이름이어야 합니다. 두 번째 인수는 하위 트리 기본 항목의 전체 DN입니다.

  ```
  $ dsconf set-plugin-prop -h host -p port plugin-name argument:attribute-name \
   argument:subtreeBaseDN argument:subtreeBaseDN...
  ```

- 기존 인수 목록에 인수를 추가하려면 다음 명령을 사용합니다.

  ```
  $ dsconf set-plugin-prop -h host -p port plugin-name argument+:argument-value
  ```

- 기본 항목의 객체 클래스에 따라 하위 트리를 정의하려면 `attribute=`*attribute-name*을 첫 번째 인수에 포함시켜 고유한 값이 있어야 하는 속성의 이름을 지정해야 합니다. 두 번째 인수는 고유성을 적용할 하위 트리의 기본 항목을 지정하는 *baseObjectClass*여야 합니다. 이 객체 클래스가 있는 항목을 대상으로 하는 작업에만 플러그인에 고유성을 적용하도록 세 번째 인수에 *entryObjectClass*를 지정할 수도 있습니다.

```
$ dsconf set-plugin-prop -h host -p port plugin-name argument:attribute=attribute-name \
 argument:markerObjectClass=baseObjectClass argument:requiredObjectClass=entryObjectClass
```

모든 플러그인 인수에서 = 기호의 앞뒤에는 공백이 없어야 합니다.

5 **변경 사항을 적용하려면 서버를 다시 시작합니다.**

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 시에는 속성 고유성이 자동으로 적용되지 않습니다.

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 사용자 서버에서 플러그인을 사용할 필요는 없습니다.

단일 마스터의 사용자에서 UID 고유성 플러그인을 활성화해도 복제나 정상적인 서버 작업을 방해하지는 않지만 성능이 약간 저하될 수 있습니다.

다중 마스터 복제 시나리오

UID 고유성 플러그인은 다중 마스터 복제 시나리오에 적합하지 않습니다. 다중 마스터 복제 시에는 느슨하게 일관적인 복제 모델을 사용하기 때문에 두 서버에서 모두 플러그인을 사용해도 같은 속성 값이 두 서버에 동시에 추가되는 것을 감지하지 못합니다.

그러나, 이름 지정 속성에 대한 고유성 검사를 수행하는 경우와 모든 마스터에서 동일한 하위 트리 내의 동일한 속성에 대해 고유성 플러그인을 사용하는 경우 UID 고유성 플러그인을 사용할 수 있습니다.

이러한 조건을 만족하면 복제 시 고유성 충돌이 이름 지정 충돌로 보고됩니다. 이름 지정 충돌은 수동으로 해결해야 합니다. 자세한 내용은 271 페이지 “일반적인 복제 충돌 해결”을 참조하십시오.

◆ ◆ ◆ 14장

14 디렉토리 서버 로깅

이 장에서는 디렉토리 서버 로그를 관리하는 방법에 대해 설명합니다.

로깅 전략을 정의하는 데 도움이 되는 정보는 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 "Designing a Logging Strategy"에 있는 로깅 정책 정보를 참조하십시오.

로그 파일 및 해당 내용에 대한 자세한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 7 장, "Directory Server Logging"을 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

로그 분석 도구

Directory Server Resource Kit에서는 디렉토리 서버 액세스 로그를 분석하는 데 사용할 수 있는 `logconv`라는 로그 분석 도구를 제공합니다. 이 로그 분석 도구는 사용 통계를 추출하고 중요 이벤트의 발생 횟수를 계산합니다. 이 도구에 대한 자세한 내용은 `logconv`(1) 설명서 페이지를 참조하십시오.

디렉토리 서버 로그 보기

서버의 기본 *instance-path*/logs 파일에서 로그를 직접 볼 수 있습니다. 기본 경로를 수정한 경우 다음과 같이 dsconf 명령을 사용하여 로그 파일 위치를 찾을 수 있습니다.

```
$ dsconf get-log-prop -h host -p port log-type path
```

디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)를 통해 로그 파일을 볼 수도 있습니다. DSCC를 사용하여 로그 항목을 보고 정렬할 수 있습니다.

아래 그림은 DSCC의 디렉토리 서버 액세스 로그 샘플을 보여줍니다.

그림 14–1 DSCC 액세스 로그

▼ 디렉토리 서버 로그 미행 방법

dsadm 명령을 사용하여 디렉토리 서버 로그의 지정된 줄 수를 표시하거나, 지정된 수명보다 짧은 로그 항목을 표시할 수 있습니다. 이 예에서는 오류 로그를 미행합니다. 액세스 로그를 미행하려면 show-error-log 대신 show-access-log를 사용합니다.

1 특정 수명보다 짧은 오류 로그 항목을 표시합니다.

```
$ dsadm show-error-log -A duration instance-path
```

기간에 대해 단위를 지정해야 합니다. 예를 들어 24시간보다 짧은 오류 로그 항목을 표시하려면 다음을 입력합니다.

```
$ dsadm show-error-log -A 24h /local/ds
```

2 오류 로그에서 지정된 줄 수를 표시합니다(끝에서 시작).

```
$ dsadm show-error-log -L last-lines instance-path
```

줄 수는 정수로 표시됩니다. 예를 들어 마지막 100개 행을 표시하려면 다음을 입력합니다.

```
$ dsadm show-error-log -L 100 /local/ds
```

값을 지정하지 않은 경우 표시되는 기본 줄 수는 20입니다.

디렉토리 서버에 대한 로그 구성

로그 파일을 다양하게 수정할 수 있습니다. 예를 들면 다음과 같습니다.

- 감사 로그 활성화

  액세스 로그 및 오류 로그와는 달리, 감사 로그는 기본적으로 사용되지 않습니다. 자세한 내용은 316 페이지 "감사 로그를 활성화하는 방법"을 참조하십시오.
- 일반 설정
  - 로깅 활성화 또는 비활성화
  - 로그 버퍼링 활성화 또는 비활성화
  - 로그 파일 위치
  - 세부 정보 로깅
  - 로그 수준
- 로그 회전 설정
  - 정기적으로 새 로그 만들기
  - 새 로그 파일이 만들어지기 전의 최대 로그 파일 크기
- 로그 삭제 설정
  - 삭제 이전의 최대 파일 사용 기간
  - 삭제 이전의 최대 파일 크기

- 삭제 이전의 최대 사용 가능한 디스크 공간

다음 절차에서는 로그 구성을 수정하는 방법과 감사 로그를 활성화하는 방법에 대해 설명합니다.

▼ 로그 구성을 수정하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 수정할 로그의 설정을 봅니다.

```
$ dsconf get-log-prop -h host -p port log-type
```

예를 들어 기존 오류 로그 설정을 나열하려면 다음을 입력합니다.

```
$ dsconf get-log-prop -h host1 -p 1389 error
Enter "cn=Directory Manager" password:
buffering-enabled         :  off
enabled                   :  on
level                     :  default
max-age                   :  1M
max-disk-space-size       :  100M
max-file-count            :  2
max-size                  :  100M
min-free-disk-space-size  :  5M
path                      :  /tmp/ds1/logs/errors
perm                      :  600
rotation-interval         :  1w
rotation-min-file-size    :  unlimited
rotation-time             :  undefined
verbose-enabled           :  off
```

2 새 값을 설정합니다.

등록 정보에 대해 원하는 값을 설정합니다.

```
$ dsconf set-log-prop -h host -p port log-type property:value
```

예를 들어 오류 로그에 대한 회전 간격을 2일로 설정하려면 다음 명령을 사용합니다.

```
$ dsconf set-log-prop -h host1 -p 1389 error rotation-interval:2d
```

▼ 감사 로그를 활성화하는 방법

액세스로그 및 오류 로그와 달리 감사 로그는 기본적으로 사용되지 않습니다. 감사 로그를 보려면 먼저 로그를 활성화해야 합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **감사 로그를 활성화합니다.**

```
$ dsconf set-log-prop -h host -p port audit enabled:on
```

수동으로 디렉토리 서버 로그 회전

로그가 계속해서 커지는 경우 언제든지 로그를 수동으로 회전할 수 있습니다. 회전은 기존 로그 파일을 백업하고 새 로그 파일을 만듭니다.

▼ 로그 파일을 수동으로 회전하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **로그 파일을 회전합니다.**

```
$ dsconf rotate-log-now -h host -p port log-type
```

예를 들어 액세스 로그를 회전하는 방법

```
$ dsconf rotate-log-now -h host1 -p 1389 access
```

◆ ◆ ◆ 15 장

15 디렉토리 서버 모니터링

디렉토리 서버는 다양한 방법을 사용하여 모니터할 수 있습니다. 이러한 방법은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 3 장, "Directory Server Monitoring"에서 설명합니다.

이 장에서는 디렉토리 서버 모니터링을 설정하고 관리하는 방법에 대해 설명합니다.

이 장은 다음 내용으로 구성되어 있습니다.

디렉토리 서버에 대한 SNMP 설정

이 절에서는 SNMP를 통해 서버를 모니터할 수 있도록 설정하는 방법에 대해 설명합니다.

Directory Server의 SNMP 구현에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Directory Server and SNMP"를 참조하십시오.

▼ SNMP를 설정하는 방법

이 절차의 일부로, DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오. 해당 절차의 다른 부분은 명령줄에서만 수행할 수 있습니다.

1 **Java ES 관리 프레임워크 플러그인을 활성화합니다.**

320 페이지 "Java ES MF 모니터링 활성화" 절차를 사용합니다. 이 절차를 사용하면 Java ES MF의 일부인 일반 에이전트 컨테이너를 활성화할 수도 있습니다.

2 **MIB에 정의되고 에이전트를 통해 제공된 SNMP 관리 대상 객체에 액세스합니다.**

이 단계에 필요한 작업은 전적으로 사용자의 SNMP 관리 시스템에 따라 결정됩니다. 관련 지침을 보려면 SNMP 관리 시스템 설명서를 참조하십시오.

MIB를 표시할 때 이 MIB에 대한 RFC 텍스트 파일을 사용할 수 있습니다. 이 파일은 `http://www.ietf.org/rfc/rfc2605.txt` 및 `http://www.ietf.org/rfc/rfc2788.txt`에서 볼 수 있습니다.

Java ES MF 모니터링 활성화

모니터링에 Java ES MF(Sun Java ES Management Framework)를 사용하려면 Java ES MF 플러그인을 활성화해야 합니다.

Java ES MF 관리에 대한 자세한 내용은 **Sun Java Enterprise System 5 Monitoring Guide**를 참조하십시오.

▼ Java ES MF 모니터링을 활성화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 **Java ES 모니터링 프레임워크를 초기화하고 등록합니다.**

```
$ dsccsetup mfwk-reg
```

이 명령의 위치는 34 페이지 “명령 위치”를 참조하십시오.

2 **Java ES 관리 프레임워크 플러그인을 활성화합니다.**

```
$ dsconf enable-plugin -h host -p port "Monitoring Plugin"
Enter "cn=Directory Manager" password:
Directory Server must be restarted for changes to take effect.
```

3 **디렉토리 서버 인스턴스를 다시 시작합니다.**

```
$ dsadm restart instance-path
```

4 **Java ES 관리 프레임워크 플러그인이 활성화되어 있는지 확인합니다.**

```
$ dsconf get-plugin-prop -h host -p port -v "Monitoring Plugin"
Enter "cn=Directory Manager" password:
Reading property values of the plugin "Monitoring Plugin"...
argument          :
depends-on-named  :
depends-on-type   :  database
desc              :  Monitoring plugin
enabled           :  on
```

```
feature           :  Monitoring
init-func         :  mf_init
lib-path          :  /opt/SUNWdsee/ds6/lib/mf-plugin.so
type              :  object
vendor            :  Sun Microsystems, Inc.
version           :  6.0
```

Java ES MF 모니터링 문제 해결

Java ES MF 모니터링이 작동하지 않으면 **Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide**의 2 장, "Installing Directory Server Enterprise Edition 6.2"에 설명된 것과 같이 Common Agent Container를 정확히 설치했는지 확인합니다.

문제가 지속되면 **Sun Java Enterprise System 5 Monitoring Guide**를 참조하십시오.

cn=monitor를 사용한 서버 모니터링

서버 상태, 복제 상태, 자원 사용 및 기타 모니터링 정보는 DSCC를 통해 사용할 수 있습니다.

또는 다음 항목에 대한 검색 작업을 수행하여 LDAP 클라이언트에서 디렉토리 서버의 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`

  *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=admin,cn=Administrators,cn=config -w - \
   -s base -b "cn=monitor" "(objectclass=*)"
  ```

  이 항목에 사용할 수 있는 모든 모니터링 속성에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Directory ServerMonitoring Attributes"를 참조하십시오.

  모니터할 수 있는 대부분의 매개 변수는 디렉토리 서버 성능을 나타내며 구성 및 조정에 의해 영향을 받습니다. 구성 가능한 각 속성에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Man Page Reference**에 있는 속성 설명서 페이지를 참조하십시오.

2

디렉토리 프록시 서버 관리

◆ ◆ ◆ 16 장

16 디렉토리 프록시 서버 도구

Sun Java™ System Directory Proxy Server 는 디렉토리 프록시 서버 인스턴스를 등록하고 관리할 수 있는 브라우저 인터페이스 및 명령줄 도구를 제공합니다. 브라우저 인터페이스는 디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)라고 합니다. 이 장에서는 DSCC 또는 명령줄을 사용하여 디렉토리 프록시 서버를 관리하는 데 필요한 기본 작업에 대해 설명합니다.

특정 작업을 수행하는 데 DSCC를 사용할지, 아니면 명령줄을 사용할지를 결정하려면 42 페이지 "DSCC 및 명령줄 사용 시기 결정"을 참조하십시오.

관리 프레임워크에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 "Directory Server Enterprise Edition Administration Model"을 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

디렉토리 프록시 서버의 DSCC 사용

이 절에서는 디렉토리 프록시 서버의 DSCC에 액세스하는 방법에 대해 설명합니다.

▼ 디렉토리 프록시 서버의 DSCC에 액세스하는 방법

1 **디렉토리 서버에서와 동일한 방법으로 DSCC에 액세스합니다.**
44 페이지 "DSCC에 액세스하는 방법"을 참조하십시오.

2 **디렉토리 프록시 서버를 보고 관리하려면 프록시 서버 탭을 누릅니다.**
다음 그림은 디렉토리 프록시 서버의 초기 창을 보여줍니다.

그림 16–1 디렉토리 프록시 서버의 초기 DSCC 창

3 **해당 서버를 보거나 관리하려면 디렉토리 프록시 서버 인스턴스를 누릅니다.**

주 – DSCC 사용에 대한 자세한 내용은 온라인 도움말을 참조하십시오.

디렉토리 프록시 서버의 명령줄 도구

디렉토리 프록시 서버 작업에 사용하는 명령줄 도구는 `dpadm` 및 `dpconf`라고 합니다. 이 명령을 사용하는 방법에 대한 자세한 내용은 `dpadm`(1M) 및 `dpconf`(1M) 설명서 페이지를 참조하십시오.

이 절에서는 `dpadm` 및 `dpconf` 명령 위치에 대해 설명합니다. 또한 환경 변수, 명령 간 비교와 명령 사용에 관한 도움말 위치에 대한 정보도 제공합니다.

디렉토리 프록시 서버 명령 위치

디렉토리 프록시 서버 명령줄 도구는 기본적으로 다음 디렉토리에 있습니다.

install-path/dps6/bin

설치 경로는 운영 체제에 따라 달라집니다. 모든 운영 체제에 대한 설치 경로는 33 페이지 "기본 경로 및 명령 위치"에 나와 있습니다.

dpconf의 환경 변수 설정

dpconf 명령에는 환경 변수를 사용하여 미리 설정할 수 있는 몇 가지 옵션이 필요합니다. 이 명령을 사용할 때 옵션을 지정하지 않거나 환경 변수를 설정하지 않으면 기본 설정이 사용됩니다. 다음 옵션에 대한 환경 변수를 구성할 수 있습니다.

-D *userDN*	사용자 바인드 DN. 환경 변수: LDAP_ADMIN_USER. 기본값: cn=Proxy Manager
-w *password-file*	사용자 바인드 DN에 대한 비밀번호 파일. 환경 변수: LDAP_ADMIN_PWF. 기본값: 비밀번호를 요청하는 메시지 표시
-h *host*	호스트 이름 또는 IP 주소. 환경 변수: DIR_PROXY_HOST . 기본값: localhost
-p *LDAP-port*	LDAP 포트 번호. 환경 변수: DIR_PROXY_PORT. 기본값: 서버 인스턴스가 root로 실행 중인 경우 389이고, 서버 인스턴스가 일반 사용자로 실행 중인 경우 1389입니다.
-e, --unsecured	dpconf는 기본적으로 명확한 연결을 열도록 지정합니다. 환경 변수: DIR_PROXY_UNSECURED. 이 변수가 설정되지 않은 경우 dpconf는 기본적으로 보안 연결을 엽니다.

자세한 내용은 dpconf(1M) 설명서 페이지를 참조하십시오.

dpadm 및 dpconf 비교

다음 표는 dpadm 및 dpconf 명령에 대한 비교 내용을 보여줍니다.

표 16–1 dpadm과 dpconf 명령 비교

	dpadm 명령	dpconf 명령
용도	디렉토리 프록시 서버의 로컬 인스턴스에서 프로세스 또는 파일 관리	디렉토리 프록시 서버의 로컬 또는 원격 인스턴스 구성

표 16–1 dpadm과 dpconf 명령 비교 *(계속)*

	dpadm 명령	dpconf 명령
사용자	운영 체제 사용자	LDAP 사용자
로컬 또는 원격	이 명령은 인스턴스에 대해 로컬로 **실행해야 합니다**. 즉, 서버가 실행 중인 호스트에서 명령을 실행해야 합니다.	이 명령은 인스턴스에 대해 로컬에서뿐만 아니라 네트워크상의 어느 위치에서도 **실행할 수 있습니다**.
명령 사용 예	디렉토리 프록시 서버 인스턴스를 만듭니다. 디렉토리 프록시 서버 인스턴스를 시작하고 중지합니다. 인증서 데이터베이스를 관리합니다.	디렉토리 프록시 서버 인스턴스의 구성을 수정합니다. 데이터 보기를 만듭니다. 데이터 소스 풀에서 로드 균형 조정을 구성합니다.
서버 상태	서버가 실행 중이거나 중지된 상태일 수 있습니다.	서버가 **실행 중이어야 합니다**.
명령에서 서버 인스턴스를 식별하는 방법	인스턴스 경로를 지정합니다. 인스턴스 경로는 상대 경로이거나 절대 경로일 수 있습니다.	호스트 이름 또는 IP 주소 및 포트 번호를 지정합니다. 이 명령은 LDAP 포트(`-p`) 또는 LDAPS 보안 포트(`-P`)를 사용합니다. 명령줄에 포트 번호를 지정하지 않으면 환경 변수 PROXY_PORT가 사용됩니다. 환경 변수가 설정되어 있지 않으면 기본 포트가 사용됩니다.

dpconf로 값이 여러 개인 등록 정보 설정

특정 디렉토리 프록시 서버 등록 정보의 값은 여러 개일 수 있습니다. 다음 값을 지정하려면 아래 구문을 사용하십시오.

```
$ dpconf set-container-prop -h host -p port \
 property:value [property:value]
```

예를 들어 이름이 `my-view`인 LDAP 데이터 보기에 쓰기 가능한 속성을 여러 개 설정하려면 다음 명령을 입력합니다.

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 \
 writable-attr:uid writable-attr:cn writable-attr:userPassword
```

값이 이미 여러 개인 등록 정보에 값을 추가하려면 다음 명령을 입력합니다.

```
$ dpconf set-container-prop -h host -p port \
 property+:value
```

값이 이미 여러 개인 등록 정보에서 값을 제거하려면 다음 명령을 입력합니다.

```
$ dpconf set-container-prop -h host -p port\
 property-:value
```

예를 들어 이전에 설명한 시나리오에서 sn을 쓰기 가능한 속성 목록에 추가하려면 다음 명령을 입력합니다.

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 \
writable-attr+:sn
```

쓰기 가능한 속성 목록에서 cn을 제거하려면 다음 명령을 입력합니다.

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 \
writable-attr-:cn
```

dpadm 및 dpconf 사용에 대한 도움말 보기

dpadm 및 dpconf 명령을 사용하는 방법에 대한 자세한 내용은 dpadm(1M) 및 dpconf(1M) 설명서 페이지를 참조하십시오.

- 하위 명령 목록을 보려면 다음 명령을 입력합니다.

  ```
  $ dpadm --help
  ```

  ```
  $ dpconf --help
  ```

- 하위 명령을 사용하는 방법에 대한 자세한 내용을 보려면 다음 명령을 입력합니다.

  ```
  $ dpadm subcommand --help
  ```

  ```
  $ dpconf subcommand --help
  ```

- dpconf 명령에 사용되는 구성 등록 정보에 대한 자세한 내용을 보려면 다음을 입력합니다.

  ```
  $ dpconf help-properties
  ```

- 하위 명령의 구성 등록 정보에 대한 자세한 내용을 보려면 다음 명령을 사용합니다.

  ```
  $ dpconf help-properties subcommand-entity
  ```

  예를 들어 액세스 로그 등록 정보에 대한 자세한 내용을 찾으려면 다음을 입력합니다.

  ```
  $ dpconf help-properties access-log
  ```

- 하위 명령에 사용되는 등록 정보에 대한 자세한 내용을 보려면 다음 명령을 사용합니다.

  ```
  $ dpconf help-properties subcommand-entity property
  ```

  예를 들어 `set-access-log-prop` 하위 명령의 `log-search-filters` 등록 정보에 대한 자세한 내용을 찾으려면 다음을 입력합니다.

  ```
  $ dpconf help-properties access-log log-search-filters
  ```

- 데이터 보기 또는 연결 처리기와 같은 엔티티 그룹의 키 등록 정보를 나열하려면 `list` 하위 명령에 세부 정보 표시 옵션 `-v`를 사용합니다.

  예를 들어 모든 연결 처리기에 대한 키 등록 정보 및 상대 우선 순위를 표시하려면 다음 명령을 사용합니다.

  ```
  $ dpconf -h host -p port list-connection-handlers -v
  Name                        is-enabled  priority  description
  --------------------------  ----------  --------  --------------------------
  anonymous                   false       99        unauthenticated connections
  default connection handler  true        100       default connection handler
  dscc administrators            true          1         Administrators connection
                                                        handler
  ```

  개별 등록 정보에 대한 자세한 내용은 각 등록 정보에 해당하는 설명서 페이지를 참조하십시오.

◆ ◆ ◆ 17장

17 디렉토리 프록시 서버 인스턴스

이 장에서는 디렉토리 프록시 서버 인스턴스를 관리하는 방법에 대해 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

디렉토리 프록시 서버 인스턴스 만들기 및 삭제

디렉토리 프록시 서버의 인스턴스를 만들 경우 인스턴스에 필요한 파일과 디렉토리가 지정된 경로에 만들어집니다.

▼ 디렉토리 프록시 서버 인스턴스를 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

DSCC를 사용하여 새 서버 인스턴스를 만들 경우 기존 서버에서 서버 구성 설정 중 일부 또는 모두를 복사하도록 선택할 수 있습니다.

1 **디렉토리 프록시 서버 인스턴스를 만듭니다.**

```
$ dpadm create -p port instance-path
```

예를 들어 `/local/dps` 디렉토리에서 새 인스턴스를 만들려면 다음 명령을 사용합니다.

```
$ dpadm create -p 2389 /local/dps
```

인스턴스의 다른 매개 변수를 지정하려면 `dpadm`(1M) 설명서 페이지를 참조하십시오.

2 **필요한 경우 비밀번호를 입력합니다.**

3 **인스턴스의 상태를 확인하여 인스턴스가 만들어졌는지 확인합니다.**

```
$ dpadm info instance-path
```

4 **(옵션) Sun Java™ Enterprise System 설치 프로그램이나 기본 패키지 설치를 사용하여 디렉토리 프록시 서버를 설치했으며 OS에서 서비스 관리 솔루션을 제공할 경우 다음 표와 같이 서버를 서비스로 관리하도록 활성화할 수 있습니다.**

운영 체제	명령
Solaris 10	`dpadm enable-service --type SMF` *instance-path*
Solaris 9	`dpadm autostart` *instance-path*
Linux, HP-UX	`dpadm autostart` *instance-path*
Windows	`dpadm enable-service --type WIN_SERVICE` *instance-path*

5 **(옵션) 다음 방법 중 하나를 사용하여 서버 인스턴스를 등록합니다.**

- URL `https://localhost:6789`를 통해 DSCC에 액세스하고 브라우저 인터페이스에 로그인합니다.
- `dsccreg add-server` 명령을 사용합니다.

  자세한 내용은 `dsccreg`(1M) 설명서 페이지를 참조하십시오.

▼ 디렉토리 프록시 서버 인스턴스를 삭제하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **(옵션) 디렉토리 프록시 서버 인스턴스를 중지합니다.**

```
$ dpadm stop instance-path
```

인스턴스를 중지하지 않으면 삭제 명령은 인스턴스를 자동으로 중지합니다. 그러나 서비스 관리 솔루션에서 인스턴스를 활성화한 경우 수동으로 중지해야 합니다.

2 **(옵션) 이전에 DSCC를 사용하여 서버를 관리했다면 명령줄을 사용하여 서버를 등록 취소합니다.**

```
$ dsccreg remove-server /local/dps
Enter DSCC administrator's password:
/local/dps is an instance of DPS
Enter password of "cn=Proxy Manager" for /local/dps:
Unregistering /local/dps from DSCC on localhost.
Connecting to /local/dps
Disabling DSCC access to /local/dps
```

자세한 내용은 `dsccreg`(1M) 설명서 페이지를 참조하십시오.

3 **(옵션) 이전에 서비스 관리 솔루션에서 서버 인스턴스를 활성화한 경우 서버를 서비스로 관리하지 않도록 비활성화합니다.**

운영 체제	명령
Solaris 10	`dpadm disable-service --type SMF` *instance-path*
Solaris 9	`dpadm autostart --off` *instance-path*
Linux, HP-UX	`dpadm autostart --off` *instance-path*
Windows	`dpadm disable-service --type WIN_SERVICE` *instance-path*

4 **인스턴스를 삭제합니다.**

```
$ dpadm delete instance-path
```

디렉토리 프록시 서버 인스턴스의 상태 확인

이 절차에서는 디렉토리 프록시 서버의 인스턴스 상태를 확인하는 방법에 대해 설명합니다.

▼ 디렉토리 프록시 서버 인스턴스의 상태를 확인하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **디렉토리 프록시 서버 인스턴스의 상태를 확인합니다.**

```
$ dpadm info instance-path
```

디렉토리 프록시 서버 인스턴스 시작, 중지 및 다시 시작

이 절에서는 명령줄에서 디렉토리 프록시 서버를 시작, 중지 및 다시 시작하는 방법에 대해 설명합니다.

▼ 디렉토리 프록시 서버를 시작 및 중지하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **디렉토리 프록시 서버를 시작하거나 중지하려면 다음 중 하나를 수행합니다.**

- 디렉토리 프록시 서버를 시작하려면 다음을 입력합니다.

  ```
  $ dpadm start instance-path
  ```

  예를 들어 `/local/dps`에서 인스턴스를 시작하려면 다음 명령을 사용합니다.

  ```
  $ dpadm start /local/dps
  ```

- 디렉토리 프록시 서버를 중지하려면 다음을 입력합니다.

  ```
  $ dpadm stop instance-path
  ```

  예를 들면 다음과 같습니다.

  ```
  $ dpadm stop /local/dps
  ```

▼ 디렉토리 프록시 서버 인스턴스를 다시 시작해야 하는지 여부를 확인하는 방법

구성 변경을 적용하려면 서버를 다시 시작해야 하는 경우가 있습니다. 이 절차를 사용하여 구성을 변경한 후에 디렉토리 프록시 서버 인스턴스를 다시 시작해야 하는지 여부를 확인합니다.

- **서버를 다시 시작해야 하는지 여부를 확인합니다.**

  ```
  $ dpconf get-server-prop -h host -p port is-restart-required
  ```

  - 이 명령에서 `true`를 반환할 경우 디렉토리 프록시 서버 인스턴스를 다시 시작해야 합니다.
  - 이 명령에서 `false`를 반환할 경우 디렉토리 프록시 서버 인스턴스를 다시 시작할 필요가 없습니다.

▼ 디렉토리 프록시 서버를 다시 시작하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **디렉토리 프록시 서버를 다시 시작합니다.**

  ```
  $ dpadm restart instance-path
  ```

  예를 들어 `/local/dps`에서 인스턴스를 다시 시작하려면 다음 명령을 사용합니다.

  ```
  $ dpadm restart /local/dps
  ```

◆ ◆ ◆ 18 장

18 디렉토리 프록시 서버 구성

이 장에서는 디렉토리 프록시 서버 인스턴스를 구성하는 방법에 대해 설명합니다. 이 장의 절차에서는 `dpadm` 및 `dpconf` 명령을 사용합니다. 이러한 명령에 대한 자세한 내용은 `dpadm`(1M) 및 `dpconf`(1M) 설명서 페이지를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

구성 예

이 절에서는 로드 균형 조정과 데이터 배포라는 두 개의 디렉토리 프록시 서버 구성 예에 대해 설명합니다. 가상 디렉토리에 대한 자세한 내용은 411 페이지 "가상 구성 예"을 참조하십시오.

로드 균형 조정을 수행하기 위해 디렉토리 프록시 서버 구성

로드 균형 조정의 간단한 사례는 검색 및 비교 작업을 한 디렉토리 집합에 보내고, 기타 작업을 다른 집합에 보내는 것으로 구성되어 있습니다. Directory Proxy Server는 모든 클라이언트 작업을 수신합니다. 서버에서는 어떤 집합이 읽기를 가져오고, 어떤 집합이 기타 작업을 가져오는지를 결정해야 합니다.

이 로드 균형 조정 시나리오를 처리하도록 Directory Proxy Server를 구성하는 핵심 단계는 다음과 같습니다.

1. 디렉토리를 디렉토리 프록시 서버에 대한 데이터 소스로 추가합니다.
2. 데이터 소스를 데이터 소스 풀에 추가합니다.
3. 검색 및 비교를 허용하도록 일부 데이터 소스를 구성하고 추가, 바인드, 삭제, 수정 및 DN 작업 수정을 허용하도록 기타 데이터 소스를 구성합니다.
4. 데이터 소스 풀을 데이터 보기에 추가합니다.

다음 예는 포트 9389에서 수신하는 Directory Proxy Server와 관련되어 있습니다. 프록시는 검색 및 비교 작업을 처리하는 한 디렉토리 서버 인스턴스 `ds1:1389`와 기타 작업을 처리하는 다른 디렉토리 서버 인스턴스 `ds2:2389`에 설명된 대로 로드 균형 조정을 수행하기 위해 여기서 구성됩니다.

첫 번째 단계에서는 데이터 소스를 만들고 활성화합니다. 이 단계에서는 프록시 서버를 다시 시작해야 합니다.

```
$ dpconf create-ldap-data-source -p 9389 ds1 localhost:1389
$ dpconf create-ldap-data-source -p 9389 ds2 localhost:2389
$ dpconf set-ldap-data-source-prop -p 9389 ds1 is-enabled:true
$ dpconf set-ldap-data-source-prop -p 9389 ds2 is-enabled:true
$ dpadm restart /local/dps
```

두 번째 단계에서는 데이터 소스를 데이터 소스 풀에 추가합니다.

```
$ dpconf create-ldap-data-source-pool -p 9389 "Directory Pool"
$ dpconf attach-ldap-data-source -p 9389 "Directory Pool" ds1 ds2
```

세 번째 단계에서는 검색 및 비교 작업을 허용하도록 `ds1`을 구성하고, 기타 작업을 허용하도록 `ds2`를 구성합니다.

```
$ dpconf set-attached-ldap-data-source-prop -p 9389 "Directory Pool" ds1 \
add-weight:disabled bind-weight:disabled compare-weight:1 delete-weight:disabled \
modify-dn-weight:disabled modify-weight:disabled search-weight:1
$ dpconf set-attached-ldap-data-source-prop -p 9389 "Directory Pool" ds2 \
add-weight:1 bind-weight:1 compare-weight:disabled delete-weight:1 \
modify-dn-weight:1 modify-weight:1 search-weight:disabled
```

네 번째 단계에서는 클라이언트 응용 프로그램 요청이 풀에 전달되도록 데이터 소스 풀을 데이터 보기에 추가합니다.

```
$ dpconf create-ldap-data-view -p 9389 "Balanced View" "Directory Pool" \
dc=example,dc=com
```

접미어 데이터의 배포를 위해 디렉토리 프록시 서버 구성

데이터 배포의 간단한 사례는 A에서 시작하여 M까지의 UID가 한 디렉토리 집합에 있는 항목을 저장하고, N에서 시작하여 Z까지의 UID가 다른 디렉토리 집합에 있는 항목을 저장하는 것으로 구성되어 있습니다. 디렉토리 프록시 서버는 모든 클라이언트 작업을 수신합니다. 서버에서는 어떤 디렉토리 집합이 A에서 M까지 처리하고, 어떤 디렉토리 집합이 N에서 Z까지 처리하는지를 결정해야 합니다.

이 데이터 배포 시나리오를 처리하도록 Directory Proxy Server를 구성하는 핵심 단계는 다음과 같습니다.

1. 디렉토리를 디렉토리 프록시 서버에 대한 데이터 소스로 추가합니다.
2. 서로 다른 데이터 배포를 처리하도록 데이터 소스를 데이터 소스 풀에 추가합니다.
3. 클라이언트 요청을 적합한 데이터 풀에 배포하도록 설계된 데이터 보기를 만듭니다.
4. LDIF가 적합한 데이터 소스로 로드되도록 분할합니다.
5. 분할 LDIF를 적합한 데이터 소스로 가져옵니다.
6. 적합한 데이터 풀에 첨부된 데이터 소스의 작업 기반 가중치를 조정합니다.

다음 예는 포트 9389에서 수신하는 Directory Proxy Server와 관련되어 있습니다. 이 예를 간단하게 하기 위해 프록시는 세 개의 디렉토리 서버 인스턴스에서만 설명된 대로 배포하도록 여기서 구성됩니다. 가용성 및 읽기 확장성을 위해 복제된 디렉토리 토폴로지를 사용하여 LDAP 데이터를 저장합니다. 한 Directory Server 인스턴스 `dsA-M:1389`에서는 A에서 시작하여 M까지의 UID가 있는 사용자 항목을 처리합니다. 다른 디렉토리 서버 인스턴스 `dsN-Z:2389`에서는 N에서 시작하여 Z까지의 UID가 있는 사용자 항목을 처리합니다. 마지막 디렉토리 인스턴스에서는 접미어가 `dsBase:3389`인 기본 항목을 처리합니다.

첫 번째 단계에서는 데이터 소스를 만들고 활성화합니다. 기본 데이터 소스는 UID가 없는 접미어의 루트 근처에 항목을 저장합니다. 일반 배포 시 이러한 항목 수는 배포된 항목 수보다 훨씬 적습니다.

```
$ dpconf create-ldap-data-source -p 9389 dsA-M localhost:1389
$ dpconf set-ldap-data-source-prop -p 9389 dsA-M is-enabled:true

$ dpconf create-ldap-data-source -p 9389 dsN-Z localhost:2389
$ dpconf set-ldap-data-source-prop -p 9389 dsN-Z is-enabled:true

$ dpconf create-ldap-data-source -p 9389 dsBase localhost:3389
$ dpconf set-ldap-data-source-prop -p 9389 dsBase is-enabled:true
```

두 번째 단계에서는 데이터 소스를 데이터 소스 풀에 추가합니다.

```
$ dpconf create-ldap-data-source-pool -p 9389 "Base Pool"
$ dpconf attach-ldap-data-source -p 9389 "Base Pool" dsBase
```

```
$ dpconf create-ldap-data-source-pool -p 9389 "A-M Pool"
$ dpconf attach-ldap-data-source -p 9389 "A-M Pool" dsA-M

$ dpconf create-ldap-data-source-pool -p 9389 "N-Z Pool"
$ dpconf attach-ldap-data-source -p 9389 "N-Z Pool" dsN-Z
```

세 번째 단계에서는 클라이언트 요청을 적합한 데이터 풀에 배포하도록 설계된 데이터 보기를 만듭니다. 기본 풀에서 dc=example,dc=com을 처리하는 한편, UID 값에 따라 배포된 데이터를 저장하는 풀에서 ou=people,dc=example,dc=com을 처리하는 방식을 살펴봅니다. 이 단계에서는 서버를 다시 시작해야 합니다.

```
$ dpconf create-ldap-data-view -p 9389 "Base View" "Base Pool" \
dc=example,dc=com

$ dpconf create-ldap-data-view -p 9389 "A-M View" "A-M Pool" \
ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop -p 9389 "A-M View" \
distribution-algorithm:lexicographic lexicographic-attrs:uid \
lexicographic-lower-bound:a lexicographic-upper-bound:m
The proxy server will need to be restarted in order for the changes to take effect

$ dpconf create-ldap-data-view -p 9389 "N-Z View" "N-Z Pool" \
ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop -p 9389 "N-Z View" \
distribution-algorithm:lexicographic lexicographic-attrs:uid \
lexicographic-lower-bound:n lexicographic-upper-bound:z
The proxy server will need to be restarted in order for the changes to take effect
$ dpadm restart /local/dps
```

네 번째 단계에서는 LDIF가 적합한 데이터 소스로 로드되도록 분할합니다. 이 예에서는 초기 분할을 수행하는 dsadm split-ldif 명령과 모든 데이터 소스에서 최상위 항목을 유지하기 위한 일부 파일 편집 기능을 모두 사용합니다. 이렇게 하면 액세스 제어 지침을 지정하는 최상위 항목을 유지할 뿐만 아니라 각 데이터 소스에 대해 단일 가져오기 명령을 사용할 수 있습니다.

```
$ dpadm split-ldif /local/dps /local/ds6/ldif/Example.ldif /tmp/
[14/May/2007:21:14:13 +0200] - STARTUP    - INFO  - Java Version: 1.5.0_09
 (Java Home: /local/jre)
[14/May/2007:21:14:13 +0200] - STARTUP    - INFO  - Java Heap Space: Total Memory
 (-Xms) = 3MB,
 Max Memory (-Xmx) = 63MB
[14/May/2007:21:14:13 +0200] - STARTUP    - INFO  - Operating System: SunOS/sparc 5.10
[14/May/2007:21:14:15 +0200] - INTERNAL   - ERROR - Entry starting at line 0 does not
 start with a DN
[14/May/2007:21:14:15 +0200] - INTERNAL   - ERROR - Unable to parse line "# Kirsten is
 a Directory Administrator and therefore should not" of entry "uid=kvaughan, ou=People,
 dc=example,dc=com" starting at line 112 as an attribute/value pair -- no colon found.
[14/May/2007:21:14:15 +0200] - INTERNAL   - ERROR - Unable to parse line "# Robert is
```

```
 a Directory Administrator and therefore should not" of entry "uid=rdaugherty,
 ou=People, dc=example,dc=com" starting at line 298 as an attribute/value pair --
 no colon found.
[14/May/2007:21:14:16 +0200] - INTERNAL   - ERROR - Unable to parse line "# Harry is
 a Directory Administrator and therefore should not" of entry "uid=hmiller, ou=People,
 dc=example,dc=com" starting at line 556 as an attribute/value pair -- no colon found.
[14/May/2007:21:14:16 +0200] - INTERNAL   - INFO  - SplitLDIF processing complete.
  Processed 156 entries.
$ ls /tmp/*ldif
/tmp/a-m view.ldif    /tmp/base view.ldif   /tmp/n-z view.ldif
```

또한 이 단계에서는 가져오기 전에 LDIF에 추가된 최상위 항목이 필요합니다.

```
$ cp /local/ds6/ldif/Example.ldif /tmp/top.ldif
$ vi /tmp/top.ldif
$ cat /tmp/top.ldif
dn: dc=example,dc=com
objectclass: top
objectclass: domain
dc: example
aci: (target ="ldap:///dc=example,dc=com")(targetattr !=
 "userPassword")(version 3.0;acl "Anonymous read-search access";
 allow (read, search, compare)(userdn = "ldap:///anyone");)
aci: (target="ldap:///dc=example,dc=com") (targetattr =
  "*")(version 3.0; acl "allow all Admin group"; allow(all) groupdn =
  "ldap:///cn=Directory Administrators,ou=Groups,dc=example,dc=com";)

$ cat /tmp/top.ldif /tmp/base\ view.ldif > /tmp/top\ and\ base\ view.ldif
$ cat /tmp/top.ldif /tmp/a-m\ view.ldif > /tmp/top\ and\ a-m\ view.ldif
$ cat /tmp/top.ldif /tmp/n-z\ view.ldif > /tmp/top\ and\ n-z\ view.ldif
```

다섯 번째 단계에서는 분할 LDIF를 적합한 데이터 소스로 가져옵니다. 여기서 기본 항목을 처리하는 디렉토리는 포트 3389에 있습니다. A에서 M까지 처리하는 디렉토리는 포트 1389에서 수신합니다. N에서 Z까지 처리하는 디렉토리는 포트 2389에서 수신합니다.

```
$ dsconf import -p 1389 /tmp/top\ and\ a-m\ view.ldif dc=example,dc=com
...
Task completed (slapd exit code: 0).

$ dsconf import -p 2389 /tmp/top\ and\ n-z\ view.ldif dc=example,dc=com
...
Task completed (slapd exit code: 0).
$ dsconf import -p 3389 /tmp/top\ and\ base\ view.ldif dc=example,dc=com
...
Task completed (slapd exit code: 0).
```

여섯 번째 단계에서는 적합한 데이터 풀에 첨부된 데이터 소스의 작업 기반 가중치를 조정합니다. 클라이언트 응용 프로그램이 검색 이외의 작업을 수행하면 해당 작업에 대한 가중치도 설정해야 합니다.

```
$ dpconf set-attached-ldap-data-source-prop -p 9389 "Base Pool" dsBase search-weight:1
$ dpconf set-attached-ldap-data-source-prop -p 9389 "A-M Pool" dsA-M search-weight:1
$ dpconf set-attached-ldap-data-source-prop -p 9389 "N-Z Pool" dsN-Z search-weight:1
```

작업 기반 가중치가 설정되면 클라이언트 응용 프로그램은 데이터가 물리적으로 배포되지 않았던 것처럼 디렉토리 프록시 서버를 통해 검색할 수 있습니다.

다음 검색에서는 UID가 R로 시작하는 사용자를 찾습니다.

```
$ ldapsearch -p 9389 -b dc=example,dc=com uid=rfisher
version: 1
dn: uid=rfisher, ou=People, dc=example,dc=com
cn: Randy Fisher
sn: Fisher
givenName: Randy
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
ou: Human Resources
ou: People
l: Cupertino
uid: rfisher
mail: rfisher@example.com
telephoneNumber: +1 408 555 1506
facsimileTelephoneNumber: +1 408 555 1992
roomNumber: 1579
```

다음 검색에서는 기본 항목 중 하나를 찾습니다.

```
$ ldapsearch -p 9389 -b ou=groups,dc=example,dc=com cn=hr\ managers
version: 1
dn: cn=HR Managers,ou=groups,dc=example,dc=com
objectClass: top
objectClass: groupOfUniqueNames
cn: HR Managers
ou: groups
uniqueMember: uid=kvaughan, ou=People, dc=example,dc=com
uniqueMember: uid=cschmith, ou=People, dc=example,dc=com
description: People who can manage HR entries
```

디렉토리 프록시 서버 구성 수정

이 절에서는 디렉토리 프록시 서버의 구성을 수정하는 방법에 대해 설명합니다.

▼ 디렉토리 프록시 서버 구성을 수정하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 디렉토리 프록시 서버의 현재 구성을 찾습니다.

```
$ dpconf get-server-prop -h host -p port
```

또는 하나 이상의 구성 등록 정보에 대해 현재 설정을 봅니다.

```
$ dpconf get-server-prop -h host -p port property-name ...
```

예를 들어 다음 명령을 실행하여 인증되지 않은 작업이 허용되는지 확인합니다.

```
$ dpconf get-server-prop -h host -p port allow-unauthenticated-operations
allow-unauthenticated-operations  :  true
```

2 하나 이상의 구성 매개 변수를 변경합니다.

```
$ dpconf set-server-prop -h host -p port property:value ...
```

예를 들어 다음 명령을 실행하여 인증되지 않은 작업을 허용하지 않습니다.

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:false
```

잘못된 변경을 수행할 경우 해당 변경 사항은 적용되지 않습니다. 예를 들어 `allow-unauthenticated-operations` 매개 변수를 `false` 대신에 `f`로 설정할 경우 다음 오류가 발생합니다.

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:f
The value "f" is not a valid value for the property "allow-unauthenticated-operations".
Allowed property values: BOOLEAN
The "set-server-prop" operation failed.
```

3 필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

디렉토리 프록시 서버 인스턴스 구성 정보 표시

디렉토리 프록시 서버 인스턴스 구성을 표시하려면 `dpconf info`를 입력합니다.

```
$ dpconf info
Instance Path           :  instance path
Host Name               :  host
Secure listen address   :  IP address
Port                    :  port
Secure port             :  secure port
SSL server certificate  :  defaultServerCert

Directory Proxy Server needs to be restarted.
```

dpconf info에는 해당 등록 정보가 기본값이 아닌 값으로 설정된 경우에만 Secure listen address 및 Non-secure listen address가 표시됩니다. 위 출력에는 이 등록 정보가 기본값 이외의 값으로 설정되지 않았으므로 Non-secure listen address가 표시되지 않습니다.

또한 dpconf info는 다시 시작해야 할 경우 인스턴스를 다시 시작하도록 사용자에게 알립니다.

dpadm info를 사용하여 디렉토리 프록시 서버 인스턴스 구성 정보를 표시할 수도 있습니다.

디렉토리 프록시 서버 인스턴스 백업 및 복원

dpadm을 사용하여 디렉토리 프록시 서버를 백업할 경우 구성 파일과 서버 인증서가 백업됩니다. 디렉토리 프록시 서버 가상 ACI를 구현한 경우 ACI도 백업됩니다.

디렉토리 프록시 서버는 서버가 성공적으로 시작될 때마다 conf.ldif 파일을 자동으로 백업합니다.

▼ 디렉토리 프록시 서버 인스턴스를 백업하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 디렉토리 프록시 서버 인스턴스를 중지합니다.

```
$ dpadm stop instance-path
```

2 디렉토리 프록시 서버 인스턴스를 백업합니다.

```
$ dpadm backup instance-path archive-dir
```

archive-dir 디렉토리는 backup 명령으로 만들어지며 이 명령을 실행하기 전에 존재하지 않아야 합니다. 이 디렉토리에는 각 구성 파일과 인증서의 백업이 포함됩니다.

▼ 디렉토리 프록시 서버 인스턴스를 복원하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

복원 작업을 시작하기 전에 디렉토리 프록시 서버 인스턴스를 만들어야 합니다.

1 디렉토리 프록시 서버 인스턴스를 중지합니다.

```
$ dpadm stop instance-path
```

2 디렉토리 프록시 서버 인스턴스를 복원합니다.

```
$ dpadm restore instance-path archive-dir
```

- 인스턴스 경로가 있을 경우 복원 작업이 자동으로 수행됩니다. *archive-dir* 디렉토리의 구성 파일과 인증서는 *instance-path* 디렉토리의 구성 파일과 인증서를 대체합니다.
- 인스턴스 경로가 없을 경우 복원 작업이 실패합니다.

프록시 관리자 구성

프록시 관리자는 권한을 가진 관리자, UNIX® 시스템의 루트 사용자와 유사합니다. 프록시 관리자 항목은 디렉토리 프록시 서버 인스턴스가 만들어질 때 정의됩니다. 프록시 관리자의 기본 DN은 `cn=Proxy Manager`입니다.

다음 절차에 표시된 것처럼 프록시 관리자 DN 및 비밀번호를 보고 변경할 수 있습니다.

▼ 프록시 관리자를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 프록시 관리자의 구성을 찾습니다.

```
$ dpconf get-server-prop -h host -p port configuration-manager-bind-dn configuration-manager-bind-pwd
configuration-manager-bind-dn   :  cn=proxy manager
configuration-manager-bind-pwd  :  {3DES}U77v39WX8MDpcWVrueetB0lfJlBc6/5n
```

프록시 관리자의 기본값은 `cn=proxy manager`입니다. 구성 관리자 비밀번호에 해시된 값이 반환됩니다.

2 프록시 관리자의 DN을 변경합니다.

```
$ dpconf set-server-prop -h host -p port configuration-manager-bind-dn:bindDN
```

3 **프록시 관리자의 비밀번호를 포함하는 파일을 만들고 해당 파일을 가리키는 등록 정보를 설정합니다.**

```
$ dpconf set-server-prop -h host -p port configuration-manager-bind-pwd-file:filename
```

서버를 다시 시작해야 하는 구성 변경 사항

디렉토리 프록시 서버 및 해당 엔티티에 대한 구성 변경 사항은 대부분 온라인으로 수행할 수 있습니다. 특정 변경 사항의 경우 변경 사항을 적용하려면 서버를 다시 시작해야 합니다. 다음 목록의 등록 정보에 대한 구성 변경 사항을 수행한 경우 서버를 다시 시작해야 합니다.

```
aci-data-view
bind-dn
client-cred-mode
custom-distribution-algorithm
db-name
db-pwd
db-url
db-user
distribution-algorithm
ldap-address
ldap-port
ldaps-port
listen-address
listen-port
load-balancing-algorithm
num-bind-init
num-read-init
num-write-init
number-of-search-threads
number-of-threads
number-of-worker-threads
ssl-policy
use-external-schema
```

등록 정보의 rws 및 rwd 키워드는 등록 정보를 변경한 경우 서버를 다시 시작해야 하는지 여부를 나타냅니다.

- 등록 정보에 rws(read, write, static) 키워드가 있는 경우 등록 정보를 변경할 때 서버를 다시 시작해야 합니다.
- 등록 정보에 rwd(read, write, dynamic) 키워드가 있는 경우 서버를 다시 시작하지 않아도 등록 정보의 수정 사항이 동적으로 구현됩니다.

등록 정보를 변경한 경우 서버를 다시 시작해야 하는지 확인하려면 다음 명령을 실행합니다.

```
$ dpconf help-properties | grep property-name
```

예를 들어 LDAP 데이터 소스의 바인드 DN을 변경한 경우 서버를 다시 시작해야 하는지 확인하려면 다음 명령을 실행합니다.

```
$ dpconf help-properties | grep bind-dn
connection-handler       bind-dn-filters        rwd  STRING | any
This property specifies a set of regular expressions. The bind DN
of a client must match at least one regular expression in order for
the connection to be accepted by the connection handler. (Default: any)
ldap-data-source      bind-dn               rws  DN | ""
This property specifies the DN to use when binding to the LDAP data
source. (Default: undefined)
```

구성을 변경한 후에 서버를 다시 시작해야 하는지 확인하려면 다음 명령을 실행합니다.

```
$ dpconf get-server-prop -h host -p port is-restart-required
```

디렉토리 프록시 서버를 사용하여 디렉토리 서버에 대한 구성 항목 액세스

디렉토리 프록시 서버에 대한 구성 항목은 cn=config에 있습니다. 디렉토리 프록시 서버를 사용하여 구성 항목에 액세스하는 경우 기본적으로 디렉토리 프록시 서버의 구성 항목에 액세스합니다.

디렉토리 서버의 구성 항목에 액세스하려면 디렉토리 프록시 서버가 아닌 디렉토리 서버에 직접 연결하는 것이 좋습니다. 디렉토리 서버를 구성하는 방법에 대한 자세한 내용은 3 장을 참조하십시오.

주의 – 디렉토리 프록시 서버를 다시 구성하여 디렉토리 서버의 구성 항목에 액세스할 경우 디렉토리 프록시 서버의 관리 프레임워크가 손상될 수 있습니다.

디렉토리 프록시 서버를 사용하여 디렉토리 서버의 구성 항목에 액세스하려면 디렉토리 프록시 서버의 관리 프레임워크가 손상되지 않도록 특별 단계를 수행합니다. 이 절에서는 디렉토리 프록시 서버를 사용하여 디렉토리 서버의 구성 항목에 액세스하는 방법에 대해 설명합니다.

▼ 디렉토리 프록시 서버를 사용하여 디렉토리 서버의 구성 항목에 액세스하는 방법

1 **357 페이지 "LDAP 데이터 소스 만들기 및 구성"에 설명된 것처럼 하나 이상의 데이터 소스를 만듭니다.**

2 **360 페이지 "LDAP 데이터 소스 풀 만들기 및 구성"에 설명된 것처럼 LDAP 데이터 소스 풀을 만듭니다.**

3 **361 페이지 "데이터 소스 풀에 LDAP 데이터 소스 첨부"에 설명된 것처럼 하나 이상의 데이터 소스를 데이터 소스 풀에 첨부합니다.**

- **특정 데이터 소스의 구성 항목을 표시하려면 하나의 LDAP 데이터 소스만 LDAP 데이터 소스 풀에 첨부합니다.**

  ```
  $ dpconf attach-ldap-data-source -h host -p port pool-name data-source-name
  ```

  이 단계를 수행하고 나면 클라이언트는 디렉토리 프록시 서버에 연결된 데이터 소스의 구성 항목에 액세스할 수 있습니다.

- **모든 데이터 소스의 구성 항목을 표시하려면 둘 이상의 LDAP 데이터 소스를 LDAP 데이터 소스 풀에 첨부합니다.**

  ```
  $ dpconf attach-ldap-data-source -h host -p port pool-name data-source-name \
   data-source-name ...
  ```

  이 단계를 수행하고 나면 클라이언트는 디렉토리 프록시 서버에 연결된 데이터 소스 중 하나의 구성 항목에 액세스할 수 있습니다. 그러나 클라이언트는 구성 항목이 속하는 데이터 소스를 알 수 없습니다.

4 **`cn=config`를 표시하려면 LDAP 데이터 보기를 만듭니다.**

```
$ dpconf create-ldap-data-view -h host -p port view-name pool-name cn=config
```

◆ ◆ ◆ 19 장

19 디렉토리 프록시 서버 인증서

이 장에서는 디렉토리 프록시 서버에서 인증서를 구성하는 방법에 대해 설명합니다. **디렉토리 서버**에서 인증서를 구성하는 방법에 대한 자세한 내용은 108 페이지 "인증서 관리"를 참조하십시오.

이 장의 절차에서는 `dpadm` 및 `dpconf` 명령을 사용합니다. 이러한 명령에 대한 자세한 내용은 `dpadm`(1M) 및 `dpconf`(1M) 설명서 페이지를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

자체 서명된 기본 인증서

디렉토리 프록시 서버 인스턴스를 만들 때 인스턴스에 자체 서명된 기본 인증서가 포함됩니다. 자체 서명된 인증서는 공개 키가 디렉토리 프록시 서버에서 자체 서명된 공개 키와 개인 키 쌍입니다.

▼ 자체 서명된 기본 인증서 보기

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **자체 서명된 기본 인증서를 봅니다.**

  ```
  $ dpadm show-cert instance-path defaultservercert
  ```

디렉토리 프록시 서버에 대한 인증서 만들기, 요청 및 설치

디렉토리 프록시 서버에서 SSL(Secure Sockets Layer)을 실행하려면 자체 서명된 인증서 또는 PKI(Public Key Infrastructure) 솔루션을 사용해야 합니다.

PKI 솔루션에는 외부 인증 기관(CA)이 필요합니다. 즉, PKI 솔루션에는 공개 키와 개인 키가 모두 포함된 CA 서명된 서버 인증서가 필요합니다. 이 인증서는 하나의 디렉토리 프록시 서버 인스턴스에만 적용됩니다. 또한 공개 키가 포함된 **신뢰할 수 있는** *CA* **인증서**도 필요합니다. 신뢰할 수 있는 CA 인증서를 사용하면 CA의 모든 서버 인증서가 신뢰됩니다. 이 인증서를 CA 루트 키 또는 루트 인증서라고도 합니다.

기본값 이외의 자체 서명된 인증서를 만들고 CA 서명된 인증서를 요청 및 설치하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 디렉토리 프록시 서버에 기본값 이외의 자체 서명된 인증서를 만드는 방법

디렉토리 프록시 서버 인스턴스를 만드는 경우 자체 서명된 기본 인증서가 자동으로 제공됩니다. 기본값 이외의 설정으로 자체 서명된 인증서를 만들려면 다음 절차를 사용합니다.

이 절차를 수행하면 공개 키가 디렉토리 프록시 서버에서 서명된 서버 인증서에 대해 공개 키와 개인 키 쌍이 만들어집니다. 자체 서명된 인증서는 3개월 동안 유효합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

- **디렉토리 프록시 서버에 대해 기본값 이외의 자체 서명된 인증서를 만들려면 다음을 입력합니다.**

  ```
  $ dpadm add-selfsign-cert instance-path cert-alias
  ```

  여기서 *cert-alias*는 자체 서명된 인증서의 이름입니다.

  예를 들어 다음과 같이 `my-self-signed-cert`라는 인증서를 만들 수 있습니다.

  ```
  $ dpadm add-selfsign-cert /local/dps my-self-signed-cert
  ```

  모든 명령 옵션에 대한 설명을 보려면 `dpadm`(1M) 설명서 페이지를 참조하거나 명령줄에 `dpadm add-selfsign-cert --help`를 입력하십시오.

▼ 디렉토리 프록시 서버에 대해 CA 서명된 인증서를 요청하는 방법

자체 서명된 인증서는 테스트용으로 유용합니다. 그러나, 작업 환경에서는 신뢰할 수 있는 인증 기관(CA) 인증서를 사용하는 것이 보다 안전합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 CA 서명된 서버 인증서를 요청합니다.

```
$ dpadm request-cert instance-path cert-alias
```

여기서 *cert-alias*는 요청하는 인증서의 이름입니다. 인증 기관에서는 서버를 식별하는 데 모든 명령 옵션이 필요할 수 있습니다. 모든 명령 옵션에 대한 설명은 `dpadm`(1M) 설명서 페이지를 참조하십시오.

CA 인증서를 얻기 위한 프로세스는 사용하는 CA에 따라 다릅니다. 일부 상용 CA에서는 인증서를 다운로드할 수 있는 웹 사이트를 제공합니다. 다른 CA에서는 인증서를 전자 메일로 보냅니다.

예를 들어 다음과 같이 `my-CA-signed-cert`라는 인증서를 요청할 수 있습니다.

```
$ dpadm request-cert -S cn=my-request,o=test /local/dps my-CA-signed-cert
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBYDCBygIBADAhMQ0wCwYDVQQDEwRnZXJpMRAwDgYDVQQDEwdteWNlcnQ0MIGfMA0GCSqGSIb3
DQEBAQUAA4GNADCBiQKBgQC3v9ubG468wnjBDAMbRrEkmFDTQzT+LO30D/ALLXOiElVsHrtRyWhJ
PG9cURI9uwqs15crxCpJvho1kt3SB9+yMB8Ql+CKnCQDHlNAfnn30MjFHShv/sAuEygFsN+Ekci5
W1jySYE2rzE0qKVxWLSILFo1UFRVRsUnORTX/Nas7QIDAQABoAAwDQYJKoZIhvcNAQEEBQADgYEA
fcQMnZNLpPobiX1xy1ROefPOhksVz8didY8Q2fjjaHG5lajMsqORO zubsuQ9Xh4ohT8kIA6xcBNZ
g8FRNIRAHCtDXKOdOm3CpJ8da+YGI/ttSawIeNAKU1DApF9zMb7c2lS4yEfWmreoQdXIC9YeKtF6
zwbn2EmIpjHzETtS5Nk=
-----END NEW CERTIFICATE REQUEST-----
```

`dpadm request-cert` 명령을 사용하여 인증서를 요청할 때 이 인증서 요청은 PEM(Privacy Enhanced Mail) 형식의 PKCS #10 인증서 요청입니다. PEM은 RFC 1421부터 1424까지 지정된 형식입니다. 자세한 내용은 `http://www.ietf.org/rfc/rfc1421.txt`를 참조하십시오. PEM 형식은 base64로 인코딩된 인증서 요청을 ASCII 형식으로 나타냅니다.

CA 서명된 인증서를 요청할 때 자체 서명된 임시 인증서가 만들어집니다. CA에서 CA 서명된 인증서를 받아 설치하면 자체 서명된 임시 인증서가 새 인증서로 대체됩니다.

2 이 절차에 따라 인증서 요청을 CA에 보냅니다.

요청을 전송한 후에는 CA에서 이 요청에 대한 응답으로 인증서를 보내줄 때까지 기다려야 합니다. 요청에 대한 응답 시간은 경우에 따라 달라집니다. 예를 들어 회사 내부의 CA인 경우 응답 시간이 단축될 수 있습니다. 그러나, 회사 외부의 CA인 경우에는 요청에 대한 응답을 받을 때까지 몇 주가 걸릴 수도 있습니다.

3 **CA에서 받은 인증서를 저장합니다.**

인증서를 텍스트 파일로 저장하고 안전한 위치에 백업합니다.

▼ 디렉토리 프록시 서버에 CA 서명된 서버 인증서를 설치하는 방법

CA 서명된 서버 인증서를 신뢰하려면 인증서를 디렉토리 프록시 서버 인스턴스에 설치해야 합니다. 이 절차를 수행하면 CA 인증서의 공개 키가 디렉토리 프록시 서버의 인증서 데이터베이스에 설치됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **이 CA의 신뢰할 수 있는 CA 인증서가 이미 설치되었는지 확인합니다.**

이 작업을 수행하려면 352 페이지 "CA 인증서를 나열하는 방법"에 설명된 것처럼 설치된 CA 인증서를 모두 나열합니다.

2 **신뢰할 수 있는 CA 인증서가 설치되어 있지 않으면 디렉토리 프록시 서버 인스턴스의 인증서 데이터베이스에 이 인증서를 추가합니다.**

```
$ dpadm add-cert instance-path cert-alias cert-file
```

여기서 *cert-alias*는 신뢰할 수 있는 CA 인증서의 이름이고 *cert-file*은 신뢰할 수 있는 CA 인증서가 포함된 파일의 이름입니다.

3 **CA 서명된 서버 인증서를 인증서 데이터베이스에 설치합니다.**

```
$ dpadm add-cert instance-path cert-alias cert-file
```

여기서 *cert-alias*는 CA 서명된 서버 인증서의 이름이고 *cert-file*은 CA 서명된 서버 인증서가 포함된 파일의 이름입니다. 이 *cert-alias*는 인증서 요청에서 사용된 *cert-alias*와 동일해야 합니다.

예를 들어 이름이 `CA-cert`인 CA 서명된 서버 인증서를 다음과 같이 `/local/dps`의 인증서 데이터베이스에 추가할 수 있습니다.

```
$ dpadm add-cert /local/dps CA-cert /local/safeplace/ca-cert-file.ascii
```

디렉토리 프록시 서버의 만료된 CA 서명 인증서 갱신

이 절에서는 만료된 CA 서명 서버 인증서를 갱신하는 방법에 대해 설명합니다.

▼ 디렉토리 프록시 서버의 만료된 CA 서명 서버 인증서를 갱신하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **해당 CA에서 업데이트된 인증서를 얻습니다.**

2 **디렉토리 프록시 서버 인스턴스에 인증서를 설치합니다.**

```
$ dpadm renew-cert instance-path cert-alias cert-file
```

여기서 *cert-alias*는 새 인증서의 이름이고 *cert-file*은 인증서가 포함된 파일의 이름입니다. 모든 명령 옵션에 대한 설명은 dpadm(1M) 설명서 페이지를 참조하십시오.

인증서 나열

서버 인증서 및 CA 인증서를 나열하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 서버 인증서를 나열하는 방법

이 절차를 수행하면 디렉토리 프록시 서버 인스턴스에 설치된 모든 인증서가 나열됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **디렉토리 프록시 서버 인스턴스의 인증서 데이터베이스에 있는 서버 인증서를 나열합니다.**

```
$ dpadm list-certs instance-path
```

기본적으로 디렉토리 프록시 서버 인스턴스에는 이름이 defaultservercert인 서버 인증서가 포함되어 있습니다. Same as issuer 텍스트는 자체 서명된 서버 인증서가 기본 인증서임을 나타냅니다.

예를 들면 다음과 같습니다.

```
$ dpadm list-certs /local/dps
Alias             Valid from       Expires on       Self-signed? Issued by          Issued to
----------------- ---------------- ---------------- ------------ ------------------ --------------
defaultservercert 2006/06/01 04:15 2008/05/31 04:15 y            CN=myserver:myport Same as issuer
1 certificate found.
```

▼ CA 인증서를 나열하는 방법

이 절차를 수행하면 디렉토리 프록시 서버 인스턴스에 설치된 CA 인증서가 나열됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **디렉토리 프록시 서버 인스턴스의 인증서 데이터베이스에 있는 CA 인증서를 나열합니다.**

  ```
  $ dpadm list-certs -C instance-path
  ```

  예를 들면 다음과 같습니다.

  ```
  $ dpadm list-certs -C /local/dps
  Alias   Valid from       Expires on        Built-in  Issued by    Issued to
  ------  ----------       ----------------  --------- ---------    ---------
  CAcert1 1999/06/21 06:00 2020/06/21 06:00  y         CN=company1, O=company2
  ...
  ```

백엔드 LDAP 서버의 인증서를 디렉토리 프록시 서버의 인증서 데이터베이스에 추가

이 절에서는 백엔드 LDAP 서버의 인증서를 디렉토리 프록시 서버의 인증서 데이터베이스에 추가하는 방법에 대해 설명합니다.

▼ 백엔드 디렉토리 서버의 인증서를 디렉토리 프록시 서버의 인증서 데이터베이스에 추가하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **다음 명령 구문을 사용하여 백엔드 디렉토리 서버의 인증서를 PEM 형식으로 표시합니다.**

   ```
   dsadm show-cert -F ascii instance-path [cert-alias]
   ```

   *cert-alias*를 지정하지 않으면 기본 서버 인증서가 표시됩니다. 모든 명령 옵션에 대한 설명은 dsadm(1M) 설명서 페이지를 참조하십시오.

예를 들어 자체 서명된 기본 서버 인증서는 다음과 같이 표시됩니다.

```
$ dsadm show-cert -F ascii /local/ds defaultCert
-----BEGIN CERTIFICATE-----
MIICJjCCAY+gAwIBAgIFAIKL36kwDQYJKoZIhvcNAQEEBQAwVzEZMBcGA1UEChMQ
U3VuIE1pY3Jvc3lzdGVtczEZMBcGA1UEAxMQRGlyZWN0b3J5IFNlcnZlcjENMAsG
A1UEAxMEMjAxMTEQMA4GA1UEAxMHY29uZHlsZTAeFw0wNjA1MjIxMTQxNTVaFw0w
NjA4MjIxMTQxNTVaMFcxGTAXBgNVBAoTEFN1biBNaWNyb3N5c3RlbXMxGTAXBgNV
BAMTEERpcmVjdG9yeSBTZXJ2ZXIxDTALBgNVBAMTBDIwMTExEDAOBgNVBAMTB2Nv
bmR5bGUwgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAK9U3ry3sJmEzwQY8CGd
7S2MTZuBedo03Vea1lfDtD08WIsdDMzhHplTdeHAkWWNc8g2PDcEFXeWp9UXFMuD
Pcia7t8HtFkm73VmlriWhMd8nn3l2vkxhsPK2LHFEeOIUDR9LBBiMiEeLkjdoEhE
VLMSoYKqKI+Aa5grINdmtFzBAgMBAAEwDQYJKoZIhvcNAQEEBQADgYEAF4eDbSd7
qy2l10dIogT+rnXZ362gLTlQFCblhbGpmmptbegUdL1ITGv/62q1isPV2rW7CkjM
Cqb0fo3k5UkKKvW+JbMowpQeAPnlgpX612HuDr1tldnKV4eyU7gpG31t/cpACALQ
7OPi1A7oVb2Z8OJKfEJHkp3txBSsiI2gTkk=
-----END CERTIFICATE-----
```

2 인증서를 저장합니다.

인증서를 텍스트 파일로 저장하고 안전한 위치에 백업합니다.

3 백엔드 LDAP 서버의 인증서를 디렉토리 프록시 서버 인스턴스의 인증서 데이터베이스에 추가합니다.

```
$ dpadm add-cert instance-path cert-alias cert-file
```

여기서 *cert-alias*는 인증서의 이름이고 *cert-file*은 인증서가 포함된 파일의 이름입니다.

예를 들어 다음과 같이 `defaultCert` 인증서를 추가할 수 있습니다.

```
$ dpadm add-cert /local/dps defaultCert /local/safeplace/defaultCert.ascii
```

인증서를 백엔드 LDAP 서버로 내보내기

백엔드 LDAP 서버에 디렉토리 프록시 서버의 인증서가 필요할 수 있습니다. 이 절에서는 디렉토리 프록시 서버를 구성하여 인증서를 백엔드 LDAP 서버로 내보내는 방법에 대해 설명합니다.

▼ 디렉토리 프록시 서버를 구성하여 클라이언트 인증서를 백엔드 LDAP 서버로 내보내는 방법

1 백엔드 LDAP 서버로 보낼 인증서를 지정합니다.

```
$ dpconf set-server-prop -h host -p port ssl-client-cert-alias:cert-alias
```

여기서 *cert-alias*는 인증서의 이름입니다. 모든 명령 옵션에 대한 설명은 dpconf(1M) 설명서 페이지를 참조하십시오.

2 **인증서 내용을 파일에 복사합니다.**

```
$ dpadm show-cert -F ascii -o filename instance-path cert-alias
```

3 **112 페이지 "CA 서명된 서버 인증서 및 신뢰할 수 있는 CA 인증서를 추가하는 방법"에 설명된 것처럼 인증서를 백엔드 LDAP 서버의 인증서 데이터베이스에 추가합니다.**

다음 순서 클라이언트 인증을 위해 백엔드 LDAP 서버를 구성합니다. 디렉토리 서버에서 이 작업을 수행하는 방법에 대한 자세한 내용은 119 페이지 "자격 증명 수준 및 인증 방법 구성"을 참조하십시오.

참조 클라이언트와 디렉토리 프록시 서버 간에 인증서 기반 인증을 구성하는 방법에 대한 자세한 내용은 455 페이지 "인증서 기반 인증을 구성하는 방법"을 참조하십시오.

디렉토리 프록시 서버의 인증서 데이터베이스 백업 및 복원

서버 인증서는 dpadm을 사용하여 디렉토리 프록시 서버를 백업할 때 함께 백업됩니다. 백업된 인증서는 *archive-path*/alias 디렉토리에 저장됩니다.

디렉토리 프록시 서버를 백업 및 복원하는 방법에 대한 자세한 내용은 342 페이지 "디렉토리 프록시 서버 인스턴스 백업 및 복원"을 참조하십시오.

인증서 데이터베이스에 액세스할 때 비밀번호 요청을 프롬프트

기본적으로 인증서 데이터베이스의 비밀번호는 내부적으로 관리됩니다. 따라서, 인증서 비밀번호를 입력하거나 비밀번호 파일을 지정할 필요가 없습니다. 인증서 데이터베이스를 저장된 비밀번호를 통해 내부적으로 관리할 경우에는 해당 비밀번호가 보안 환경에 저장됩니다.

인증서에 대한 보안과 제어를 강화하려면 명령줄에서 비밀번호 요청을 프롬프트하도록 디렉토리 프록시 서버를 구성합니다. 그러면 autostart, backup, disable-service, enable-service, info, restore 및 stop을 제외한 dpadm의 모든 하위 명령에서 비밀번호를 입력하라는 메시지가 표시됩니다.

비밀번호 요청을 프롬프트하거나 프롬프트하지 않도록 디렉토리 프록시 서버를 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 인증서 데이터베이스에 액세스할 때 비밀번호 요청을 프롬프트하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 서버를 중지합니다.

```
$ dpadm stop instance-path
Directory Proxy Server instance 'instance-path' stopped
```

2 비밀번호 프롬프트 플래그를 on으로 설정한 다음 인증서 데이터베이스 비밀번호를 입력하고 확인합니다.

```
$ dpadm set-flags instance-path cert-pwd-prompt=on
Choose the certificate database password:
Confirm the certificate database password:
```

3 서버를 시작한 다음 인증서 데이터베이스 비밀번호를 입력합니다.

```
$ dpadm start instance-path
Enter the certificate database password:
```

▼ 인증서 데이터베이스에 액세스할 때 비밀번호 요청 프롬프트를 비활성화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 서버를 중지합니다.

```
$ dpadm stop instance-path
Directory Proxy Server instance 'instance-path' stopped
```

2 비밀번호 프롬프트 플래그를 off로 설정한 다음 기존 비밀번호를 입력합니다.

```
$ dpadm set-flags instance-path cert-pwd-prompt=off
Enter the old password:
```

3 서버를 시작합니다.

```
$ dpadm start instance-path
```

◆ ◆ ◆ 20 장

20 LDAP 데이터 소스 및 데이터 소스 풀

이 장에서는 dpconf 명령을 사용하여 LDAP 데이터 소스 및 데이터 소스 풀을 만들고 구성하는 방법에 대해 설명합니다. 이러한 항목에 대한 참조 정보는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "LDAP Data Sources"를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

LDAP 데이터 소스 만들기 및 구성

LDAP 데이터 소스를 만들고 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ LDAP 데이터 소스를 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 데이터 소스를 만듭니다.

```
$ dpconf create-ldap-data-source -h host -p port source-name host:port
```

이 명령에서 *source-name*은 새 데이터 소스에 할당하는 이름입니다. *host* 및 *port*는 LDAP 서버가 실행 중인 호스트와 포트를 나타냅니다. 데이터 소스는 기본적으로 SSL을 사용하지 않습니다.

호스트가 IP V6 주소로 지정된 경우에는 데이터 소스를 만들 때 IP V6 참조를 사용해야 합니다. 예를 들어 디렉토리 프록시 서버가 포트 2389에서 IP V6 주소 fe80::209:3dff:fe00:8c93을 가진 호스트에 바인드될 경우 다음 명령을 사용하여 데이터 소스를 만듭니다.

```
$ dpconf create-ldap-data-source -h host1 -p 1389 ipv6-host \
 [fe80::209:3dff:fe00:8c93]:2389
```

콘솔을 사용하여 데이터 소스를 만들 경우 대괄호 없이 실제 IP V6 주소를 지정해야 합니다.

LDAP 데이터 소스의 등록 정보를 수정하는 방법에 대한 자세한 내용은 358 페이지 “LDAP 데이터 소스를 구성하는 방법”을 참조하십시오.

2 (옵션) 데이터 소스의 목록을 봅니다.

```
$ dpconf list-ldap-data-sources -h host -p port
```

▼ LDAP 데이터 소스를 구성하는 방법

이 절차에서는 디렉토리 프록시 서버와 LDAP 데이터 소스 간 인증을 구성합니다. 또한 디렉토리 프록시 서버에서 LDAP 데이터 소스를 모니터링하는 방법도 구성합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 다음 명령 구문을 사용하여 데이터 소스의 등록 정보를 봅니다.

```
dpconf get-ldap-data-source-prop -h host -p port [-M unit] [-Z unit] source-name [property...]
```

이 명령에서 -M 및 -Z는 데이터를 표시할 단위를 나타냅니다. M 옵션은 시간 단위를 지정합니다. -M 값은 월, 주, 일, 시, 분, 초 또는 밀리초를 나타내는 M, w, d, h, m, s 또는 ms가 될 수 있습니다. -Z 옵션은 데이터 크기 단위를 지정합니다. -Z 값은 테라바이트, 기가바이트, 메가바이트, 킬로바이트 또는 바이트를 나타내는 T, G, M, k 또는 b가 될 수 있습니다.

등록 정보를 지정하지 않으면 모든 등록 정보가 표시됩니다. LDAP 데이터 소스의 기본 등록 정보는 다음과 같습니다.

```
bind-dn                       :  -
bind-pwd                      :  -
client-cred-mode              :  use-client-identity
connect-timeout               :  10s
description                   :  -
is-enabled                    :  false
is-read-only                  :  true
ldap-address                  :  host
ldap-port                     :  port
```

```
ldaps-port                    :  ldaps
monitoring-bind-timeout       :  5s
monitoring-entry-dn           :  ""
monitoring-entry-timeout      :  5s
monitoring-inactivity-timeout :  2m
monitoring-interval           :  30s
monitoring-mode               :  proactive
monitoring-search-filter      :  (|(objectClass=*)(objectClass=ldapSubEntry))
num-bind-incr                 :  10
num-bind-init                 :  10
num-bind-limit                :  1024
num-read-incr                 :  10
num-read-init                 :  10
num-read-limit                :  1024
num-write-incr                :  10
num-write-init                :  10
num-write-limit               :  1024
proxied-auth-check-timeout    :  1.8s
proxied-auth-use-v1           :  false
ssl-policy                    :  never
use-tcp-no-delay              :  true
```

2 데이터 소스를 활성화합니다.

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name is-enabled:true
```

3 기본 설정을 변경하려면 단계 1에 나열된 모든 등록 정보를 구성합니다.

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name property:value
```

예를 들어 데이터 소스의 항목을 수정하려면 쓰기 작업을 허용하도록 데이터 소스를 구성합니다.

```
$ dpconf set-ldap-data-source-prop -h host -p port source-name is-read-only:false
```

하위 명령에 사용되는 등록 정보에 대한 정보를 확인하려면 다음 명령을 실행합니다.

```
$ dpconf help-properties ldap-data-source property
```

데이터 소스에 대한 키 등록 정보를 나열하려면 `list` 하위 명령과 함께 세부 정보 표시 옵션 `-v`를 사용합니다.

```
$ dpconf list-ldap-data-sources -v
Name         is-enabled  ldap-address  ldap-port  ldaps-port  description
-----------  ----------  ------------  ---------  ----------  -----------
datasource0  true        myHost        myPort     ldaps       -
datasource1  true        myHost        myPort     ldaps       -
```

4 **필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.**

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오. 서버를 다시 시작해야 하는 구성 변경 사항 목록은 344 페이지 "서버를 다시 시작해야 하는 구성 변경 사항"을 참조하십시오.

LDAP 데이터 소스 풀 만들기 및 구성

데이터 소스 풀을 만들고 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ LDAP 데이터 소스 풀을 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **데이터 소스 풀을 하나 이상 만듭니다.**

```
$ dpconf create-ldap-data-source-pool -h host -p port pool-name
```

첫 번째 *pool-name* 다음에 데이터 소스 풀을 추가로 지정할 수 있습니다. 데이터 소스 풀의 등록 정보를 수정하는 방법에 대한 자세한 내용은 360 페이지 "LDAP 데이터 소스 풀을 구성하는 방법"을 참조하십시오.

2 **(옵션) 데이터 소스 풀의 목록을 봅니다.**

```
$ dpconf list-ldap-data-source-pools -h host -p port
```

▼ LDAP 데이터 소스 풀을 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **다음 명령 구문을 사용하여 데이터 소스 풀의 등록 정보를 봅니다.**

```
dpconf get-ldap-data-source-pool-prop -h host -p port [-M unit] [-Z unit] \
 pool-name [property...]
```

이 명령에서 `-M` 및 `-Z`는 데이터를 표시할 단위를 나타냅니다. `M` 옵션은 시간 단위를 지정합니다. `-M` 값은 월, 주, 일, 시, 분, 초 또는 밀리초를 나타내는 `M`, `w`, `d`, `h`, `m`, `s` 또는 `ms`가 될 수 있습니다. `-Z` 옵션은 데이터 크기 단위를 지정합니다. `-Z` 값은 테라바이트, 기가바이트, 메가바이트, 킬로바이트 또는 바이트를 나타내는 `T`, `G`, `M`, `k` 또는 `b`가 될 수 있습니다.

등록 정보를 지정하지 않으면 모든 등록 정보가 표시됩니다. LDAP 데이터 소스 풀의 기본 등록 정보는 다음과 같습니다.

```
client-affinity-policy    :  write-affinity-after-write
client-affinity-timeout   :  20s
description               :  -
enable-client-affinity    :  false
load-balancing-algorithm  :  proportional
```

2 단계 1에 나열된 등록 정보를 구성합니다.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 property:value
```

로드 균형 조정 및 클라이언트 선호도를 위해 데이터 소스 풀의 등록 정보를 구성하는 방법에 대한 자세한 내용은 21 장를 참조하십시오.

데이터 소스 풀에 LDAP 데이터 소스 첨부

데이터 소스 풀에 첨부되는 데이터 소스를 **첨부된 데이터 소스**라고 합니다. 첨부된 데이터 소스의 등록 정보는 데이터 소스 풀의 로드 균형 조정 구성을 결정합니다. 첨부된 데이터 소스의 가중치를 구성할 경우 데이터 소스 풀에 있는 모든 첨부된 데이터 소스의 가중치를 고려합니다. 필요에 따라 가중치가 함께 작동하는지 확인합니다. 로드 균형 조정을 위해 가중치를 구성하는 방법에 대한 자세한 내용은 364 페이지 “로드 균형 조정에 대한 가중치를 구성하는 방법”을 참조하십시오.

▼ 데이터 소스 풀에 LDAP 데이터 소스를 첨부하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 데이터 소스 풀에 데이터 소스를 하나 이상 첨부합니다.

```
$ dpconf attach-ldap-data-source -h host -p port pool-name \
 source-name [source-name ...]
```

2 (옵션) 지정된 데이터 소스 풀에 대해 첨부된 데이터 소스 목록을 봅니다.

```
$ dpconf list-attached-ldap-data-sources -h host -p port -E pool-name
```

이 명령에서 `-E`는 선택 사항이며 한 줄에 하나의 등록 정보 값을 표시하도록 표시 출력을 수정합니다.

3 (옵션) 지정된 데이터 소스 풀에 대해 첨부된 데이터 소스의 키 등록 정보를 봅니다.

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
```

이 명령에서 -v는 세부 정보 표시 출력을 지정합니다. 예를 들어 데이터 소스 풀의 등록 정보 예를 봅니다.

```
$ dpconf list-attached-ldap-data-sources -h host1 -p 1389 -v My-pool
Name         add-weight  bind-weight  compare-weight
-----------  ----------  -----------  --------------
datasource0  disabled    disabled     disabled
datasource1  disabled    disabled     disabled

delete-weight  modify-dn-weight  modify-weight  search-weight
-------------  ----------------  -------------  -------------
disabled       disabled          disabled       disabled
disabled       disabled          disabled       disabled
```

4 (옵션) 다음 명령 구문을 사용하여 첨부된 데이터 소스의 등록 정보를 봅니다.

```
$ dpconf get-attached-ldap-data-source-prop -h host -p port [-M unit] [-Z unit] \
 pool-name source-name [property...]
```

이 명령에서 -M 및 -Z는 데이터를 표시할 단위를 나타냅니다. M 옵션은 시간 단위를 지정합니다. -M 값은 월, 주, 일, 시, 분, 초 또는 밀리초를 나타내는 M, w, d, h, m, s 또는 ms가 될 수 있습니다. -Z 옵션은 데이터 크기 단위를 지정합니다. -Z 값은 테라바이트, 기가바이트, 메가바이트, 킬로바이트 또는 바이트를 나타내는 T, G, M, k 또는 b가 될 수 있습니다.

등록 정보를 지정하지 않으면 모든 등록 정보가 표시됩니다.

첨부된 데이터 소스의 등록 정보는 로드 균형 조정에서 각 작업 유형에 대한 가중치를 정의합니다. 첨부된 데이터 소스의 기본 가중치는 다음과 같습니다.

```
add-weight        :  disabled
bind-weight       :  disabled
compare-weight    :  disabled
delete-weight     :  disabled
modify-dn-weight  :  disabled
modify-weight     :  disabled
search-weight     :  disabled
```

로드 균형 조정을 위해 첨부된 데이터 소스의 가중치를 구성하는 방법에 대한 자세한 내용은 364 페이지 “로드 균형 조정에 대한 가중치를 구성하는 방법”을 참조하십시오.

◆ ◆ ◆ 21 장

21 디렉토리 프록시 서버 로드 균형 조정 및 클라이언트 선호도

로드 균형 조정 및 클라이언트 선호도에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 16 장, "Directory Proxy Server Load Balancing and Client Affinity"를 참조하십시오. 이 장은 다음 내용으로 구성되어 있습니다.

로드 균형 조정 구성

로드 균형 조정에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Load Balancing"을 참조하십시오. 이 절에서는 로드 균형 조정을 구성하는 방법에 대해 설명하며 구성 예를 제공합니다.

▼ 로드 균형 조정 알고리즘을 선택하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 LDAP 데이터 소스 풀의 등록 정보를 확인하여 현재 로드 균형 조정 알고리즘을 얻습니다.

```
$ dpconf get-ldap-data-source-pool-prop -h host -p port pool-name
```

LDAP 데이터 소스 풀의 기본 등록 정보는 다음과 같습니다.

```
client-affinity-policy    :  write-affinity-after-write
client-affinity-timeout   :  20s
description               :  -
enable-client-affinity    :  false
load-balancing-algorithm  :  proportional
```

로드 균형 조정 알고리즘의 기본값은 `proportional`입니다.

2 알고리즘을 사용하려면 LDAP 데이터 소스 풀을 구성합니다.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
  load-balancing-algorithm:selected-algorithm
```

여기서 *selected-algorithm*은 다음 값 중 하나입니다.

- `failover`
- `operational-affinity`
- `proportional`
- `saturation`

알고리즘에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Introduction to Load Balancing"을 참조하십시오.

3 디렉토리 프록시 서버 인스턴스를 다시 시작합니다.

```
$ dpadm restart instance-path
```

▼ 로드 균형 조정에 대한 가중치를 구성하는 방법

첨부된 데이터 소스의 가중치는 데이터 소스 풀에 첨부된 다른 데이터 소스의 가중치와 관련하여 구성해야 합니다. 첨부된 모든 데이터 소스의 가중치를 고려합니다. 작업 유형에 대해 데이터 소스에 `disabled` 가중치가 부여되면 이 데이터 소스에 해당 유형의 요청이 전송되지 않습니다. 데이터 소스에 `0`(영)의 가중치가 부여되면 다른 모든 데이터 소스를 사용할 수 없는 경우가 아닌 한, 이 데이터 소스에 요청이 배포되지 않습니다. 따라서, `0`의 가중치로 구성된 데이터 소스는 다른 모든 데이터 소스를 사용할 수 없는 경우에 사용됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 데이터 소스 풀에 첨부된 데이터 소스 목록을 봅니다.

```
$ dpconf list-attached-ldap-data-sources -h host -p port pool-name
```

2 첨부된 데이터 소스 중 하나에 대한 등록 정보를 봅니다.

```
$ dpconf get-attached-ldap-data-source-prop pool-name \
 attached-data-source-name
```

첨부된 데이터 소스의 등록 정보는 각 작업 유형에 대한 가중치를 정의합니다. 첨부된 데이터 소스의 기본 가중치는 다음과 같습니다.

```
add-weight        :  disabled
bind-weight       :  disabled
compare-weight    :  disabled
delete-weight     :  disabled
modify-dn-weight  :  disabled
```

```
modify-weight     :  disabled
search-weight     :  disabled
```

3 첨부된 데이터 소스 중 하나에 대한 가중치를 구성합니다.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name \
 attached-data-source-name add-weight:value \
 bind-weight:value compare-weight:value delete-weight:value \
 modify-dn-weight:value modify-weight:value search-weight:value
```

4 다른 첨부된 데이터 소스에 대해 단계 2 및 단계 3을 반복합니다.

5 첨부된 데이터 소스의 키 매개 변수를 비교합니다.

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
```

예를 들어 데이터 소스 풀에 다음 가중치가 부여된 데이터 소스가 포함될 수 있습니다.

```
$ dpconf list-attached-ldap-data-sources -h host1 -p 1389 -v myPool
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
---- ---------- ----------- -------------- ------------- ---------------- ------------- -------------
DS-1 disabled   3               disabled       disabled      disabled         disabled      disabled
DS-2 2          2           2              2             2                2             2
DS-3 1          1           1              1             1                1             1
```

로드 균형 조정 구성 예

이 절에는 각 로드 균형 조정 알고리즘을 구성하는 절차 예가 포함되어 있습니다.

▼ 로드 균형 조정에 대한 비례 알고리즘을 구성하는 방법

비례 알고리즘에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Proportional Algorithm for Load Balancing"을 참조하십시오.

위의 예에서 데이터 소스 *ds–1*은 다른 두 데이터 소스의 가중치에 비해 2배로 구성되어 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

시작하기 전에 데이터 소스 풀에 데이터 소스가 세 개 이상 첨부되어 있는지 확인합니다. 데이터 소스 및 데이터 소스 풀을 만드는 방법에 대한 자세한 내용은 20 장을 참조하십시오.

1 데이터 소스 풀이 로드 균형 조정에 대한 비례 알고리즘을 사용하도록 구성합니다.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
  load-balancing-algorithm:proportional
```

2 첫 번째 데이터 소스의 등록 정보를 구성합니다.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

3 두 번째 데이터 소스의 등록 정보를 구성합니다.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4 세 번째 데이터 소스의 등록 정보를 구성합니다.

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 첨부된 데이터 소스의 키 매개 변수를 비교합니다.

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
---  ---------- ----------- -------------- ------------- ---------------- ------------- -------------
ds-1 2          2           2              2             2                2             2
ds-2 1          1           1              1             1                1             1
ds-3 1          1           1              1             1                1             1
```

6 디렉토리 프록시 서버 인스턴스를 다시 시작합니다.

```
$ dpadm restart instance-path
```

▼ 로드 균형 조정에 대한 포화 알고리즘을 구성하는 방법

포화 알고리즘에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Saturation Algorithm for Load Balancing"을 참조하십시오.

위의 예에서 데이터 소스 `ds-1`은 대부분의 바인드 작업을 수행하지만 다른 유형의 작업은 수행하지 않습니다. 세 데이터 소스는 다음과 같은 가중치로 구성되어 있습니다.

- `ds-1`은 바인드 작업에 대해 가중치 3으로 구성되어 있으며 다른 모든 유형의 작업에 대해서는 비활성화되어 있습니다.
- `ds-2`는 모든 작업에 대해 가중치 2로 구성되어 있습니다.
- `ds-3`은 모든 작업에 대해 가중치 1로 구성되어 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

시작하기 전에 데이터 소스 풀에 데이터 소스가 세 개 이상 첨부되어 있는지 확인합니다. 데이터 소스 및 데이터 소스 풀을 만드는 방법에 대한 자세한 내용은 20 장을 참조하십시오.

1. **데이터 소스 풀이 로드 균형 조정에 대한 포화 알고리즘을 사용하도록 구성합니다.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:saturation
```

2. **첫 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:disabled bind-weight:3 compare-weight:disabled delete-weight:disabled \
 modify-dn-weight:disabled modify-weight:disabled search-weight:disabled
```

3. **두 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

4. **세 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5. **첨부된 데이터 소스의 키 매개 변수를 비교합니다.**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
---  ---------- ----------- -------------- ------------- ---------------- ------------- -------------
ds-1 disabled   3               disabled       disabled      disabled         disabled      disabled
ds-2 2          2           2              2             2                2             2
ds-3 1          1           1              1             1                1             1
```

6. **디렉토리 프록시 서버 인스턴스를 다시 시작합니다.**

```
$ dpadm restart instance-path
```

▼ 전역 계정 잠금에 대한 작업 선호도 알고리즘을 구성하는 방법

이 알고리즘에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Operational Affinity Algorithm for Global Account Lockout"을 참조하십시오.

이 예에서는 세 개의 데이터 소스를 사용합니다. 데이터 소스 `ds-1`이 모든 바인드 요청을 수신하도록 구성됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

시작하기 전에 데이터 소스 풀에 데이터 소스가 세 개 이상 첨부되어 있는지 확인합니다. 데이터 소스 및 데이터 소스 풀을 만드는 방법에 대한 자세한 내용은 20 장을 참조하십시오.

1. **데이터 소스 풀이 작업 선호도 알고리즘을 사용하도록 구성합니다.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:operational-affinity
```

2. **첫 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:1 bind-weight:100 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

3. **두 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4. **세 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5. **첨부된 데이터 소스의 키 매개 변수를 비교합니다.**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
--   ---------- ----------- -------------- ------------- ---------------- ------------- -------------
ds-1 1          1           1              1             1                1             1
ds-2 1          100         1              1             1                1             1
ds-3 1          1           1              1             1                1             1
```

6. **디렉토리 프록시 서버 인스턴스를 다시 시작합니다.**

```
$ dpadm restart instance-path
```

▼ 캐시 최적화에 대한 작업 선호도 알고리즘을 구성하는 방법

이 알고리즘에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Operational Affinity Algorithm for Cache Optimization"을 참조하십시오.

이 예에서는 세 개의 데이터 소스를 사용합니다. 모든 검색 및 비교 작업이 데이터 소스 `ds-1`에서 처리됩니다. `ds-1`이 요청에 응답하면 대상 항목이 캐시에 저장됩니다. `ds-1`이 동일한 요청에 대해 반복적으로 응답하면 데이터 소스가 캐시된 데이터를 사용할 수 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

시작하기 전에 데이터 소스 풀에 데이터 소스가 세 개 이상 첨부되어 있는지 확인합니다. 데이터 소스 및 데이터 소스 풀을 만드는 방법에 대한 자세한 내용은 20 장을 참조하십시오.

1. **데이터 소스 풀이 작업 선호도 알고리즘을 사용하도록 구성합니다.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:operational-affinity
```

2. **첫 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:1 bind-weight:1 compare-weight:100 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:100
```

3. **두 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

4. **세 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5. **첨부된 데이터 소스의 키 매개 변수를 비교합니다.**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
---  ---------- ----------- -------------- ------------- ---------------- ------------- -------------
ds-1 1          1           100            1             1                1             100
ds-2 1          1           1              1             1                1             1
ds-3 1          1           1              1             1                1             1
```

6. **디렉토리 프록시 서버 인스턴스를 다시 시작합니다.**

```
$ dpadm restart instance-path
```

▼ 로드 균형 조정에 대한 페일오버 알고리즘을 구성하는 방법

페일오버 알고리즘에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Failover Algorithm for Load Balancing"을 참조하십시오.

이 예에서는 세 개의 데이터 소스를 사용합니다. 데이터 소스 `ds-1`이 모든 요청을 수신합니다. `ds-1`이 실패하면 `ds-1`이 복구될 때까지 `ds-2`가 모든 요청을 수신합니다. `ds-1`이 복구되기 전에 `ds-2`가 실패하면 `ds-3`이 모든 요청을 수신합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

시작하기 전에 데이터 소스 풀에 데이터 소스가 세 개 이상 첨부되어 있는지 확인합니다. 데이터 소스 및 데이터 소스 풀을 만드는 방법에 대한 자세한 내용은 20 장을 참조하십시오.

1 **데이터 소스 풀이 로드 균형 조정에 대한 페일오버 알고리즘을 사용하도록 구성합니다.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 load-balancing-algorithm:failover
```

2 **첫 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-1 \
 add-weight:3 bind-weight:3 compare-weight:3 delete-weight:3 modify-dn-weight:3 \
 modify-weight:3 search-weight:3
```

3 **두 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-2 \
 add-weight:2 bind-weight:2 compare-weight:2 delete-weight:2 modify-dn-weight:2 \
 modify-weight:2 search-weight:2
```

4 **세 번째 데이터 소스의 등록 정보를 구성합니다.**

```
$ dpconf set-attached-ldap-data-source-prop -h host -p port pool-name ds-3 \
 add-weight:1 bind-weight:1 compare-weight:1 delete-weight:1 modify-dn-weight:1 \
 modify-weight:1 search-weight:1
```

5 **첨부된 데이터 소스의 키 매개 변수를 비교합니다.**

```
$ dpconf list-attached-ldap-data-sources -h host -p port -v pool-name
Name add-weight bind-weight compare-weight delete-weight modify-dn-weight modify-weight search-weight
---  ---------- ----------- -------------- ------------- ---------------- ------------- -------------
ds-1 3          3           3              3             3                3             3
ds-2 2          2           2              2             2                2             2
ds-3 1          1           1              1             1                1             1
```

6 **디렉토리 프록시 서버 인스턴스를 다시 시작합니다.**

```
$ dpadm restart instance-path
```

클라이언트 선호도 구성

클라이언트 선호도는 로드 균형 조정 배포 시 전달 지연의 위험을 줄여줍니다. 클라이언트 선호도에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Client Affinity"를 참조하십시오. 이 절에서는 클라이언트 연결과 데이터 소스 간에 선호도를 구성하는 방법에 대해 설명하며 구성 예를 제공합니다.

▼ 클라이언트 선호도를 구성하는 방법

이 절차에서는 클라이언트 연결과 데이터 소스 간에 선호도를 구성하는 방법에 대해 설명합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 데이터 소스 풀의 등록 정보를 표시하여 현재 로드 균형 조정 알고리즘을 봅니다.

```
$ dpconf get-ldap-data-source-pool-prop -h host -p port pool-name
```

데이터 소스 풀의 기본 등록 정보는 다음과 같습니다.

```
client-affinity-policy    :  write-affinity-after-write
client-affinity-timeout   :  20s
description               :  -
enable-client-affinity    :  false
load-balancing-algorithm  :  proportional
```

`client-affinity-policy`, `client-affinity-timeout` 및 `enable-client-affinity` 매개변수는 클라이언트 선호도를 구성합니다. 등록 정보에 대한 설명과 유효한 값 목록을 보려면 다음을 입력하십시오.

```
dpconf help-properties ldap-data-source-pool client-affinity-policy \
 client-affinity-timeout enable-client-affinity
```

등록 정보에 대한 자세한 내용은 다음 설명서 페이지를 참조하십시오.
`client-affinity-policy`(5dpconf), `client-affinity-timeout`(5dpconf) 및 `enable-client-affinity`(5dpconf)

2 클라이언트 선호도를 활성화합니다.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 enable-client-affinity:true
```

3 클라이언트 선호도에 대한 정책을 선택합니다.

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:selected-policy
```

여기서 *selected-policy*는 다음 값 중 하나입니다.

`write-affinity-after-write`
  첫 번째 쓰기 요청 이후 쓰기 요청에 대한 선호도

`read-write-affinity-after-write`
  첫 번째 쓰기 요청 이후 모든 요청에 대한 선호도

`read-write-affinity-after-any`
  첫 번째 읽기 요청 또는 쓰기 요청 이후 모든 요청에 대한 선호도

`read-affinity-after-write`
  쓰기 요청 이후 첫 번째 읽기 요청에 대한 선호도

4 **클라이언트 선호도 기간을 구성합니다.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-timeout:time-out[unit]
```

시간 초과의 기본 *unit*은 밀리초입니다.

클라이언트 선호도 구성 예

이 절에는 클라이언트 선호도와 관련된 구성 예와 복제 지연, 쓰기 작업 확인 및 연결 기반 라우팅에 대한 예가 포함되어 있습니다.

▼ 데이터 소스 풀에 마스터 및 사용자가 포함된 경우 복제 지연에 대한 클라이언트 선호도를 구성하는 방법

이 절차를 수행하면 첫 번째 쓰기 작업 이후 최대 3초간 발생하는 모든 읽기 및 쓰기 작업에 대한 클라이언트 선호도가 구성됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **데이터 소스 풀의 선호도 매개 변수를 구성합니다.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-write-affinity-after-write client-affinity-timeout:3000 \
 enable-client-affinity:true
```

▼ 클라이언트 선호도가 읽기 작업으로 각 쓰기 작업을 확인하도록 구성하는 방법

이 절차를 수행하면 각 쓰기 작업 이후 첫 번째 읽기 작업에 대한 클라이언트 선호도가 구성됩니다. 이 예는 지정된 바인드 DN에서 읽기 작업을 수행하여 각 쓰기 작업을 검증하는 응용 프로그램에 적용할 수 있는 내용입니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **데이터 소스 풀의 선호도 매개 변수를 구성합니다.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-affinity-after-write enable-client-affinity:true
```

▼ 연결 기반 라우팅에 대한 클라이언트 선호도를 구성하는 방법

Directory Proxy Server 6.0 이전 버전에서는 클라이언트와 LDAP 서버 간에 한 개의 연결이 열렸습니다. 연결이 닫힐 때까지 클라이언트의 모든 요청에 대해 동일한 연결이 사용되었습니다. 이러한 라우팅 유형을 **연결 기반 라우팅***(connection-based routing)*이라고 합니다. 이 절차에서는 연결 기반 라우팅에 대한 클라이언트 선호도를 구성하는 방법에 대해 설명합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

시작하기 전에 모든 데이터 소스가 데이터 소스 풀에 첨부되었고 `clientCredentialsForwarding`이 `useBind`로 설정되었는지 확인합니다.

- **데이터 소스 풀의 선호도 매개 변수를 구성합니다.**

```
$ dpconf set-ldap-data-source-pool-prop -h host -p port pool-name \
 client-affinity-policy:read-write-affinity-after-any enable-client-affinity:true
```

◆ ◆ ◆ 22 장

22 디렉토리 프록시 서버 Distribution

디렉토리 프록시 서버 배포 개요와 사용 사례 예에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 17 장, “Directory Proxy Server Distribution”을 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

LDAP 데이터 보기 만들기 및 구성

LDAP 데이터 보기를 만들고 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ LDAP 데이터 보기를 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 LDAP 데이터 보기를 만듭니다.

```
$ dpconf create-ldap-data-view -h host -p port view-name pool-name suffix-DN
```

LDAP 데이터 보기의 등록 정보를 수정하는 방법에 대한 자세한 내용은 376 페이지 “LDAP 데이터 보기를 구성하는 방법”을 참조하십시오.

2 LDAP 데이터 보기의 목록을 봅니다.

```
$ dpconf list-ldap-data-views -h host -p port
```

▼ LDAP 데이터 보기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 LDAP 데이터 보기의 등록 정보를 봅니다.

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name
```

등록 정보를 구성하지 않고 데이터 보기를 만든 경우 데이터 보기의 구성은 다음과 같습니다.

```
alternate-search-base-dn               :  ""
attr-name-mappings                        :  none
base-dn                                   :  suffix-DN
contains-shared-entries                   :  false
custom-distribution-algorithm-class       :  none
description                               :  -
distribution-algorithm                    :  none
dn-join-rule                              :  none
dn-mapping-attrs                          :  none
dn-mapping-source-base-dn                 :  none
excluded-subtrees                         :  -
filter-join-rule                          :  none
is-enabled                                :  true
is-read-only                              :  false
is-routable                               :  true
ldap-data-source-pool                     :  pool-name
lexicographic-attrs                       :  all
lexicographic-lower-bound                 :  none
lexicographic-upper-bound                 :  none
non-viewable-attr                         :  none
non-writable-attr                         :  none
numeric-attrs                             :  all
numeric-default-data-view                 :  false
numeric-lower-bound                       :  none
numeric-upper-bound                       :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression    :  all
pattern-matching-one-level-search-filter  :  all
pattern-matching-subtree-search-filter    :  all
process-bind                              :  -
replication-role                          :  master
viewable-attr                             :  all except non-viewable-attr
writable-attr                             :  all except non-writable-attr
```

주 – 프록시 관리자를 제외한 모든 사용자는 백엔드 서버의 `cn=config` 및 `cn=monitor` 접미어를 볼 수 있습니다. 기본적으로 백엔드 서버의 데이터는 프록시 관리자가 사용할 수 없습니다. 프록시 관리자가 사용할 수 있는 `cn=config` 및 `cn=monitor` 하위 트리는 프록시 자체에 해당하는 항목입니다.

디렉토리 프록시 서버 인스턴스를 만들 경우 프록시 관리자를 위한 연결 처리기가 빈 데이터 보기 정책을 사용하여 만들어집니다. 프록시 관리자에게 백엔드 데이터에 대한 액세스 권한이 필요한 경우 프록시 관리자 연결 처리기의 데이터 보기 정책에 데이터 보기를 추가해야 합니다. 이러한 데이터 보기에서 `cn=config` 및 `cn=monitor` 하위 트리는 기본적으로 제외됩니다.

2 단계 1에 나열된 등록 정보 중 하나 이상을 변경합니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 property:value [property:value ... ]
```

예를 들어 데이터 소스의 `dc=example,dc=com` 하위 트리에 액세스하려면 데이터 보기에서 `base-dn`을 지정합니다.

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView base-dn:dc=example,dc=com
```

값이 여러 개인 등록 정보에 값을 추가하려면 이 명령을 사용합니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name property+:value
```

값이 여러 개인 등록 정보에서 값을 제거하려면 이 명령을 사용합니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name property-:value
```

3 필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

▼ 사용자 정의 배포 알고리즘을 구성하는 방법

사용자 정의 배포 알고리즘은 모든 유형의 데이터 보기(`ldap-data-view`, `jdbc-data-view`, `ldif-data-view` 및 `join-data-view`)에 구성할 수 있습니다. 다음 절차에서 알고리즘은 `ldap-data-view`에 대해서만 설정됩니다.

1 배포 알고리즘 클래스가 들어 있는 JAR(Java Archive) 파일의 경로를 포함하도록 `extension-jar-file-url` 등록 정보를 설정합니다.

```
$ dpconf set-server-prop -h host -p port extension-jar-file-url:jar file path
```

*jar file path*는 `file:/expt/dps/custom_plugin/myjar.jar`과 같은 유효한 JAR 파일 경로로 대체될 수 있습니다.

2 **custom-distribution-algorithm을 구성하기 전에 distribution-algorithm을 none으로 설정합니다.**

```
$ dpconf set-ldap-data-view-prop view name distribution-algorithm:none
```

3 **custom-distribution-algorithm 등록 정보를 사용자 정의 배포 알고리즘 클래스로 설정합니다.**

```
$ dpconf set-ldap-data-view-prop view name custom-distribution-algorithm:PackageName.AlgoClassName
```

속성 및 DN 이름 바꾸기

디렉토리의 각 항목은 DN 및 속성과 해당 값의 집합으로 식별됩니다. 클라이언트측에 정의된 DN 및 속성은 서버측에 정의된 DN 및 속성에 매핑되지 않는 경우가 많습니다. 데이터 보기를 정의하여 DN 및 속성의 이름을 바꿀 수 있습니다. 클라이언트에서 요청할 경우 서버측 이름과 일치하도록 DN 및 속성의 이름이 바뀝니다. 클라이언트에 결과가 반환되면 클라이언트측 이름과 일치하도록 DN 및 속성이 다시 변경됩니다.

속성 및 DN 이름 바꾸기에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Attribute Renaming and DN Renaming"을 참조하십시오. 속성 및 DN 이름을 바꾸는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 속성 이름 바꾸기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **속성 매핑을 구성할 데이터 보기에서 하나 이상의 attr-name-mappings 등록 정보를 설정합니다.**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 attr-name-mappings:client-side-attribute-name#server-side-attribute-name
  [attr-name-mappings:client-side-attribute-name#server-side-attribute-name ...]
```

예를 들어 클라이언트측의 surname 이름을 서버측의 sn으로 바꿉니다.

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView \
 attr-name-mappings:surname#sn
```

기존 매핑 목록에 속성 매핑을 추가하려면 이 명령을 사용합니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 attr-name-mappings+:client-side-attribute-name#server-side-attribute-name
```

기존 매핑 목록에서 속성 매핑을 제거하려면 이 명령을 사용합니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 attr-name-mappings-:client-side-attribute-name#server-side-attribute-name
```

▼ DN 이름 바꾸기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 DN의 이름을 바꾸려는 데이터 보기의 base-dn 등록 정보 및 DN 매핑 등록 정보를 봅니다.

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
```

이러한 등록 정보의 의미는 다음과 같습니다.

- `base-dn`은 데이터 보기의 기본 DN와 동등한 클라이언트측에 있는 하위 트리의 DN입니다.
- `dn-mapping-source-base-dn`은 서버측에 있는 하위 트리의 DN입니다.
- `dn-mapping-attrs`는 항목의 DN을 포함하는 속성 목록을 정의합니다.

예를 들어 클라이언트측에서 `dc=example,dc=com` 데이터베이스에 대한 데이터 보기 값은 DN 이름 바꾸기가 정의되지 않을 때 다음과 같습니다.

```
$ dpconf get-ldap-data-view-prop myDataView base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
 base-dn                   :  dc=example,dc=com
 dn-mapping-attrs          :  none
 dn-mapping-source-base-dn  :  none
```

2 클라이언트측의 DN을 서버측의 DN에 매핑합니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
  dn-mapping-source-base-dn:server-side-dn
```

예를 들어 클라이언트측의 `dc=example,dc=com` 데이터베이스를 서버측의 `dc=example,dc=org`에 매핑합니다.

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView \
 dn-mapping-source-base-dn:dc=example,dc=org
```

3 이러한 속성에 DN이 포함된 경우 단계 2의 영향을 받는 DIT 부분에서 속성의 이름을 바꿉니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
  dn-mapping-attrs:attribute-name [dn-mapping-attrs:attribute-name ...]
```

예를 들어 group 속성에 DN이 포함된 경우 단계 2의 이름 바꾸기 작업에 영향을 받는 이름 공간에서 속성의 이름을 다음과 같이 바꿉니다.

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 myDataView dn-mapping-attrs:group
```

기존 매핑 목록에 DN 매핑을 추가하려면 이 명령을 사용합니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name dn-mapping-attrs+:attribute-name
```

기존 매핑 목록에서 DN 매핑을 제거하려면 이 명령을 사용합니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name dn-mapping-attrs-:attribute-name
```

4 **DN의 이름을 바꾼 데이터 보기의 base-dn 등록 정보 및 DN 매핑 등록 정보를 봅니다.**

```
$ dpconf get-ldap-data-view-prop -h host -p port view-name base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
```

예를 들어 클라이언트측의 dc=example,dc=com 데이터베이스에 대한 데이터 보기 값은 DN의 이름을 바꾼 후에 다음과 같습니다.

```
$ dpconf get-ldap-data-view-prop -h host1 -p 1389 myDataView base-dn \
 dn-mapping-source-base-dn dn-mapping-attrs
 base-dn                   :  dc=example,dc=com
 dn-mapping-attrs          :  group
 dn-mapping-source-base-dn  :  dc=example,dc=org
```

excluded-subtrees 및 alternate-search-base-dn 구성

하위 데이터 보기가 만들어지면 디렉토리 프록시 서버는 상위 데이터 보기에서 하위 데이터 보기를 자동으로 제외합니다. 하위 데이터 보기가 요청의 대상인 경우 요청은 상위 데이터 보기가 아닌, 하위 데이터 보기로 보내집니다.

하위 데이터 보기에서 대체 검색 기준을 지정하면 상위 데이터 보기를 대상으로 하는 검색 작업이 하위 데이터 보기에서도 수행됩니다.

기본적으로 디렉토리 프록시 서버는 excluded-subtrees 및 alternate-search-base-dn 등록 정보를 자동으로 구성합니다. 다음 절차에서는 이러한 등록 정보를 수동으로 구성하는 방법에 대해 설명합니다.

▼ excluded-subtrees 및 alternate-search-base-dn 등록 정보를 수동으로 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 **요청을 수동으로 전달하도록 디렉토리 프록시 서버를 구성합니다.**

```
$ dpconf set-server-prop -h host -p port data-view-automatic-routing-mode:manual
```

`data-view-automatic-routing-mode`가 `manual`인 경우 디렉토리 프록시 서버는 `excluded-subtrees` 및 `alternate-search-base-dn` 등록 정보를 생성하지 않습니다. 이러한 등록 정보의 값을 수동으로 설정해야 합니다. 여기에서 설정한 값은 디렉토리 프록시 서버에서 확인되지 않습니다. 이러한 값을 잘못 설정하면 관리 경로가 손상될 수 있습니다.

또는 요청을 부분적으로 전달하도록 디렉토리 프록시 서버를 수동으로 구성합니다.

```
$ dpconf set-server-prop -h host -p port data-view-automatic-routing-mode:limited
```

`data-view-automatic-routing-mode`가 `limited`인 경우 디렉토리 프록시 서버는 `excluded-subtrees` 및 `alternate-search-base-dn` 등록 정보를 생성하지 않습니다. 그러나 디렉토리 프록시 서버는 여기에서 설정된 값이 관리 경로와 충돌하지 않는지 확인합니다.

2 **보기 제외 기준을 구성합니다.**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name excluded-subtrees:suffix-DN
```

보기 제외 기준은 해당 항목이 데이터 보기에서 표시되지 않는 DIT의 분기를 결정합니다.

3 **대체 검색 기준을 구성합니다.**

```
$ dpconf set-ldap-data-view-prop -h host -p port view-name \
 alternate-search-base-dn:search-base-DN
```

대체 검색 기준은 이 데이터 보기에 속하는 항목이 배치될 수 있는 DIT의 다른 분기를 결정합니다. 기본 DN은 기본적으로 모든 데이터 보기에서 대체 검색 기준으로 정의됩니다.

사용 사례 예를 위한 데이터 보기 만들기 및 구성

이 절에서는 데이터 보기에 대한 다음 정보와 데이터 보기를 만들고 구성하는 방법에 대해 설명합니다.

- 390 페이지 “계층 및 배포 알고리즘이 있는 데이터 보기”

이 절의 예에서는 연결 처리기를 통해 디렉토리 프록시 서버가 모든 클라이언트 연결을 처리할 수 있다고 가정합니다.

기본 데이터 보기

등록 정보를 구성하지 않고 데이터 보기를 만든 경우 데이터 보기의 구성은 다음과 같습니다.

```
alternate-search-base-dn                    :  ""
alternate-search-base-dn                    :  base-DN
attr-name-mappings                          :  none
base-dn                                     :  suffix-DN
contains-shared-entries                     :  -
description                                 :  -
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  -
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
ldap-data-source-pool                       :  pool-name
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
process-bind                                :  -
replication-role                            :  master
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

요청의 대상 DN에 상관없이 모든 요청을 전달하는 데이터 보기

이 절에서는 요청의 대상 DN에 상관없이 모든 요청을 데이터 소스 풀에 전달하는 데이터 보기의 구성을 보여줍니다. 이 데이터 보기를 **루트 데이터 보기**라고 합니다. 루트 데이터 보기는 디렉토리 프록시 서버 인스턴스를 만들 때 기본적으로 만들어집니다. 루트 데이터 보기에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Data Views to Route All Requests, Irrespective of the Target DN of the Request"를 참조하십시오.

루트 데이터 보기의 구성은 다음과 같습니다.

```
alternate-search-base-dn                    :  -
attr-name-mappings                          :  none
base-dn                                     :  ""
contains-shared-entries                     :  -
description                                 :  Automatically-generated data view
                                               able to route client operations
                                               independently of the operation base dn
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  ""
excluded-subtrees                           :  cn=config
excluded-subtrees                           :  cn=monitor
excluded-subtrees                           :  cn=proxy manager
excluded-subtrees                           :  cn=virtual access controls
excluded-subtrees                           :  dc=example,dc=com
filter-join-rule                            :  -
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
ldap-data-source-pool                       :  defaultDataSourcePool
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
```

```
process-bind                          :  -
replication-role                      :  master
viewable-attr                         :  all except non-viewable-attr
writable-attr                         :  all except non-writable-attr
```

여러 데이터와 동등한 데이터 소스에 하위 트리 목록이 저장된 경우 요청을 전달하는 데이터 보기

이 절에서는 하위 트리 목록을 대상으로 하는 요청을 데이터와 동등한 데이터 소스 집합에 전달하는 데이터 보기를 구성하는 방법에 대해 설명합니다. 이 배포 유형에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Data Views to Route Requests When a List of Subtrees Are Stored on Multiple, Data-Equivalent Data Sources"를 참조하십시오.

이 절의 예에는 동일한 하위 트리 집합이 포함된 여러 데이터 소스가 있습니다. 데이터 소스는 데이터와 동등하며 로드 균형 조정을 위해 하나의 데이터 소스 풀에 풀링됩니다. 클라이언트 요청의 하위 트리를 표시하기 위해 각 하위 트리에 대한 데이터 보기가 구성됩니다. 다음 그림은 배포 예를 보여줍니다.

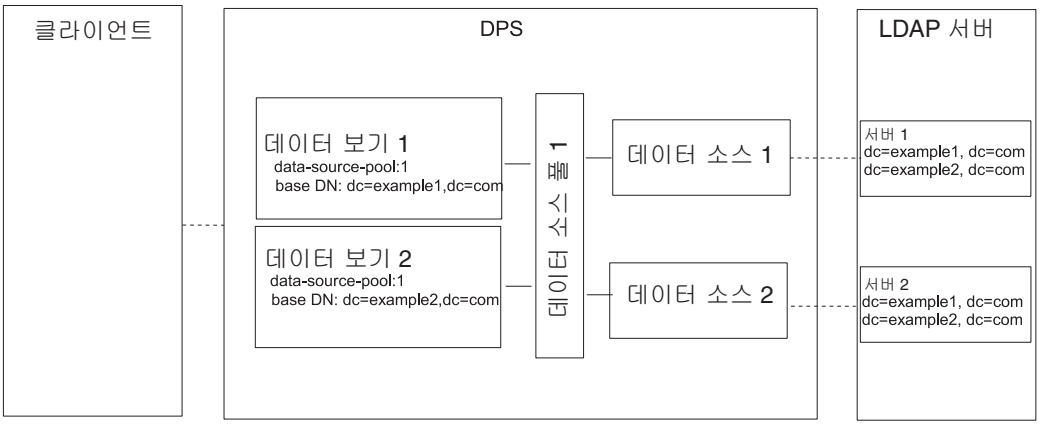


그림 22–1 여러 데이터와 동등한 데이터 소스에 하위 트리 목록이 저장된 경우 요청을 전달하는 배포 예

▼ 여러 데이터와 동등한 데이터 소스에 하위 트리 목록이 저장된 경우 요청을 전달하는 데이터 보기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **357 페이지 "LDAP 데이터 소스 만들기 및 구성"에 설명된 것처럼 각 LDAP 서버에 대한 데이터 소스를 만듭니다.**

2 **360 페이지 "LDAP 데이터 소스 풀 만들기 및 구성"에 설명된 것처럼 데이터 소스 풀을 만듭니다.**

3 **361 페이지 "데이터 소스 풀에 LDAP 데이터 소스 첨부"에 설명된 것처럼 데이터 소스를 데이터 소스 풀에 첨부합니다.**

4 **(옵션) 로드 균형 조정을 구성합니다.**

자세한 내용은 363 페이지 "로드 균형 조정 구성"을 참조하십시오.

5 **데이터 소스 풀을 참조하는 dc=example1,dc=com에서 기본 DN이 있는 데이터 보기를 만듭니다.**

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-1 \
  base-dn:dc=example1,dc=com ldap-data-source-pool:data-source-pool-1
```

6 **데이터 소스 풀을 참조하는 dc=example2,dc=com에서 기본 DN이 있는 다른 데이터 보기를 만듭니다.**

```
$ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-2 \
  base-dn:dc=example2,dc=com ldap-data-source-pool:data-source-pool-1
```

데이터 보기의 다른 등록 정보는 382 페이지 "기본 데이터 보기"의 기본 데이터 보기와 동일합니다.

7 **필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.**

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

다른 하위 트리가 다른 데이터 소스에 저장된 경우 단일 액세스 지점을 제공하는 데이터 보기

이 절에서는 여러 데이터 소스에 저장된 다른 하위 트리에 단일 액세스 지점을 제공하는 데이터 보기를 구성하는 방법에 대해 설명합니다. 이 배포 유형에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Data Views to Provide a Single Point of Access When Different Subtrees Are Stored on Different Data Sources"를 참조하십시오.

이 절의 예에는 각 하위 트리에 대한 데이터 보기가 포함됩니다. 데이터와 동등한 데이터 소스의 각 집합에 대해 데이터 소스 풀이 구성됩니다. 다음 그림은 배포 예를 보여줍니다.

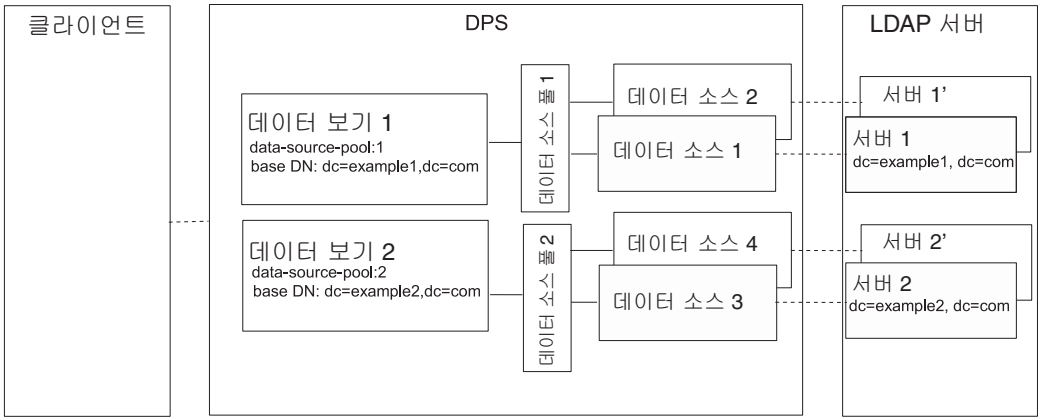


그림 22–2 다른 하위 트리가 다른 데이터 소스에 저장된 경우 단일 액세스 지점을 제공하는 배포 예

▼ 다른 하위 트리가 다른 데이터 소스에 저장된 경우 단일 액세스 지점을 제공하는 데이터 보기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **357 페이지 "LDAP 데이터 소스 만들기 및 구성"에 설명된 것처럼 각 LDAP 서버에 대한 데이터 소스를 만듭니다.**

2. **360 페이지 "LDAP 데이터 소스 풀 만들기 및 구성"에 설명된 것처럼 두 개의 데이터 소스 풀을 만듭니다.**

3. **361 페이지 "데이터 소스 풀에 LDAP 데이터 소스 첨부"에 설명된 것처럼 `dc=example1,dc=com`을 포함하는 데이터 소스를 `data-source-pool-1`에 첨부하고 `dc=example2,dc=com`을 포함하는 데이터 소스를 `data-source-pool-2`에 첨부합니다.**

4. **(옵션) 로드 균형 조정을 구성합니다.**

   자세한 내용은 363 페이지 "로드 균형 조정 구성"을 참조하십시오.

5. **`data-source-pool-1`을 참조하는 `dc=example1,dc=com`에서 기본 DN이 있는 데이터 보기를 만듭니다.**

   ```
   $ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-1 \
     base-dn:dc=example1,dc=com ldap-data-source-pool:data-source-pool-1
   ```

6. **`data-source-pool-2`를 참조하는 `dc=example2,dc=com`에서 기본 DN이 있는 다른 데이터 보기를 만듭니다.**

   ```
   $ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-2 \
     base-dn:dc=example2,dc=com ldap-data-source-pool:data-source-pool-2
   ```

데이터 보기의 다른 등록 정보는 382 페이지 “기본 데이터 보기”의 기본 데이터 보기와 동일합니다.

7 **필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.**

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 “디렉토리 프록시 서버를 다시 시작하는 방법”을 참조하십시오.

하위 트리의 다른 부분이 다른 데이터 소스에 저장된 경우 단일 액세스 지점을 제공하는 데이터 보기

이 절에서는 하위 트리의 다른 부분에 대한 단일 액세스 지점을 제공하는 데이터 보기를 구성하는 방법에 대해 설명합니다. 이 예에는 동일한 기본 DN을 가진 두 개의 데이터 보기가 포함됩니다. 항목을 다른 데이터 보기로 구분하기 위해 숫자 배포 알고리즘이 사용됩니다. 데이터와 동등한 데이터 소스의 각 집합에 대해 데이터 소스 풀이 구성됩니다. 다음 그림은 배포 예를 보여줍니다.

이 배포 유형에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 “Data Views to Route Requests When Different Parts of a Subtree Are Stored in Different Data Sources”를 참조하십시오.

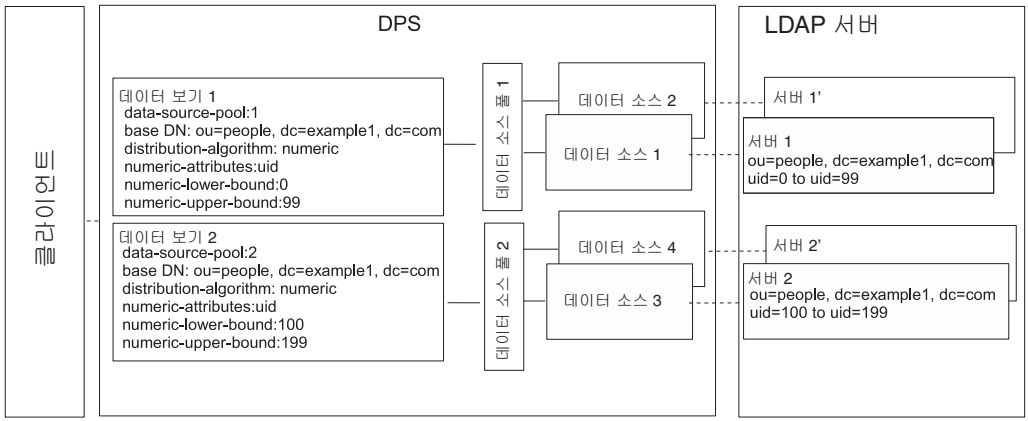


그림 22–3 하위 트리의 다른 부분이 다른 데이터 소스에 저장된 경우 단일 액세스 지점을 제공하는 배포 예

▼ 하위 트리의 다른 부분이 다른 데이터 소스에 저장된 경우 단일 액세스 지점을 제공하는 데이터 보기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1. **357 페이지 "LDAP 데이터 소스 만들기 및 구성"에 설명된 것처럼 각 LDAP 서버에 대한 데이터 소스를 만듭니다.**

2. **360 페이지 "LDAP 데이터 소스 풀 만들기 및 구성"에 설명된 것처럼 두 개의 데이터 소스 풀을 만듭니다.**

3. **361 페이지 "데이터 소스 풀에 LDAP 데이터 소스 첨부"에 설명된 것처럼 하위 트리의 한 부분을 포함하는 데이터 소스를 `data-source-pool-1`에 첨부하고 하위 트리의 다른 부분을 포함하는 데이터 소스를 `data-source-pool-2`에 첨부합니다.**

4. **(옵션) 로드 균형 조정을 구성합니다.**

   자세한 내용은 363 페이지 "로드 균형 조정 구성"을 참조하십시오.

5. **배포 알고리즘이 있는 데이터 보기를 만들어 `0` 및 `99` 사이의 `uid`가 있는 `ou=people,dc=example,dc=com`에서 항목을 선택하고 데이터 보기를 구성하여 요청을 `data-source-pool-1`에 보냅니다.**

   ```
   $ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-1 \
    ldap-data-source-pool:data-source-pool-1 base-dn:ou=people,dc=example,dc=com \
    distribution-algorithm    :numeric numeric-attrs:uid numeric-lower-bound    :0 \
    numeric-upper-bound    :99
   ```

6. **배포 알고리즘이 있는 다른 데이터 보기를 만들어 `100` 및 `199` 사이의 `uid`가 있는 `ou=people,dc=example,dc=com`에서 항목을 선택하고 데이터 보기를 구성하여 요청을 `data-source-pool-2`에 보냅니다.**

   ```
   $ dpconf set-ldap-data-view-prop -h host1 -p 1389 dataview-2 \
    ldap-data-source-pool:data-source-pool-2 base-dn:ou=people,dc=example,dc=com \
    distribution-algorithm:numeric numeric-attrs:uid numeric-lower-bound:100
    numeric-upper-bound    :199
   ```

   데이터 보기의 다른 등록 정보는 382 페이지 "기본 데이터 보기"의 기본 데이터 보기와 동일합니다.

7. **필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.**

   디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

상위 및 하위 트리가 다른 데이터 소스에 저장된 경우 단일 액세스 지점을 제공하는 데이터 보기

이 절에서는 하위 트리의 상위 분기가 하위 분기의 다른 데이터 소스에 저장된 경우 단일 액세스 지점에 대한 데이터 보기를 구성하는 방법에 대해 설명합니다. 이 배포 유형에

대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Data Views to Route Requests When Superior and Subordinate Subtrees Are Stored in Different Data Sources"를 참조하십시오.

이 절의 예에는 세 개의 데이터 보기가 포함됩니다. 데이터 보기 1의 기본 DN은 데이터 보기 2의 기본 DN 및 데이터 보기 3의 기본 DN에 대해 상위입니다. 즉, 데이터 소스 2 및 데이터 소스 3에는 데이터 소스 1의 하위 트리에 종속하는 하위 트리가 포함됩니다. 다음 그림은 배포 예를 보여줍니다.

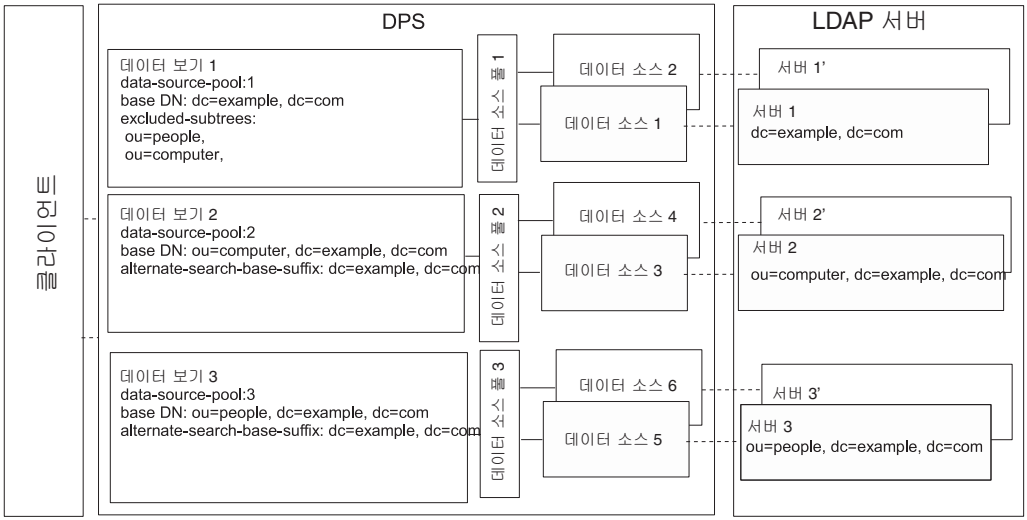


그림 22–4 상위 및 하위 트리가 다른 데이터 소스에 저장된 경우 요청을 전달하는 배포 예

디렉토리 프록시 서버는 하위 분기가 별개 데이터 보기의 기본 DN으로 구성된 경우 하위 트리의 하위 분기를 데이터 보기에서 자동으로 제외합니다.

▼ 상위 및 하위 트리가 다른 데이터 소스에 저장된 경우 단일 액세스 지점을 제공하는 데이터 보기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **357 페이지 "LDAP 데이터 소스 만들기 및 구성"에 설명된 것처럼 각 LDAP 서버에 대한 데이터 소스를 만듭니다.**
2. **360 페이지 "LDAP 데이터 소스 풀 만들기 및 구성"에 설명된 것처럼 세 개의 데이터 소스 풀을 만듭니다.**
3. **361 페이지 "데이터 소스 풀에 LDAP 데이터 소스 첨부"의 지침에 따라 데이터 소스를 데이터 소스 풀에 첨부합니다.**

- dc=example,dc=com을 포함하는 데이터 소스를 data-source-pool-1에 첨부합니다.
- ou=computer,dc=example,dc=com을 포함하는 데이터 소스를 data-source-pool-2에 첨부합니다.
- ou=people,dc=example,dc=com을 포함하는 데이터 소스를 data-source-pool-3에 첨부합니다.

4 (옵션) 로드 균형 조정을 구성합니다.

자세한 내용은 363 페이지 "로드 균형 조정 구성"을 참조하십시오.

5 dc=example,dc=com 및 데이터 소스 풀 data-source-pool-1에서 기본 DN이 있는 데이터 보기를 만듭니다.

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
 data-source-pool-1 dc=example,dc=com
```

6 ou=computer,dc=example,dc=com 및 데이터 소스 풀 data-source-pool-2에서 기본 DN이 있는 데이터 보기를 만듭니다.

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
 data-source-pool-2 ou=computer,dc=example,dc=com
```

7 ou=people,dc=example,dc=com 및 데이터 소스 풀 data-source-pool-3에서 기본 DN이 있는 데이터 보기를 만듭니다.

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-3 \
 data-source-pool-3 ou=people,dc=example,dc=com
```

8 excluded-subtrees 매개 변수를 확인하여 ou=computer,dc=example, dc=com 및 ou=people,dc=example, dc=com 하위 트리가 dataview-1에서 제외되었는지 확인합니다.

```
$ dpconf get-ldap-data-view-prop -h host1 -p 1389 dataview-1 excluded-subtrees
```

제외된 하위 트리 목록이 반환됩니다.

9 필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

계층 및 배포 알고리즘이 있는 데이터 보기

이 절에서는 계층을 배포 알고리즘과 결합하도록 데이터 보기를 구성하는 방법에 대해 설명합니다. 이 배포 유형에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Data Views With Hierarchy and a Distribution Algorithm"을 참조하십시오.

이 절의 예에는 네 개의 데이터 보기가 포함됩니다. 데이터 보기 1의 기본 DN은 다른 데이터 보기의 기본 DN에 대해 상위입니다. 데이터 보기 3 및 데이터 보기 4에는 동일한 DN이 있지만 숫자 배포 알고리즘이 항목을 다른 데이터 보기로 구분합니다.

디렉토리 프록시 서버는 하위 분기가 별개 데이터 보기의 기본 DN으로 구성된 경우 하위 트리의 하위 분기를 데이터 보기에서 자동으로 제외합니다. 숫자 배포 알고리즘은 동일한 하위 트리 항목을 다른 데이터 보기로 구분합니다. 데이터와 동등한 데이터 소스의 각 집합에 대해 데이터 소스 풀이 구성됩니다.

다음 그림은 배포 예를 보여줍니다.

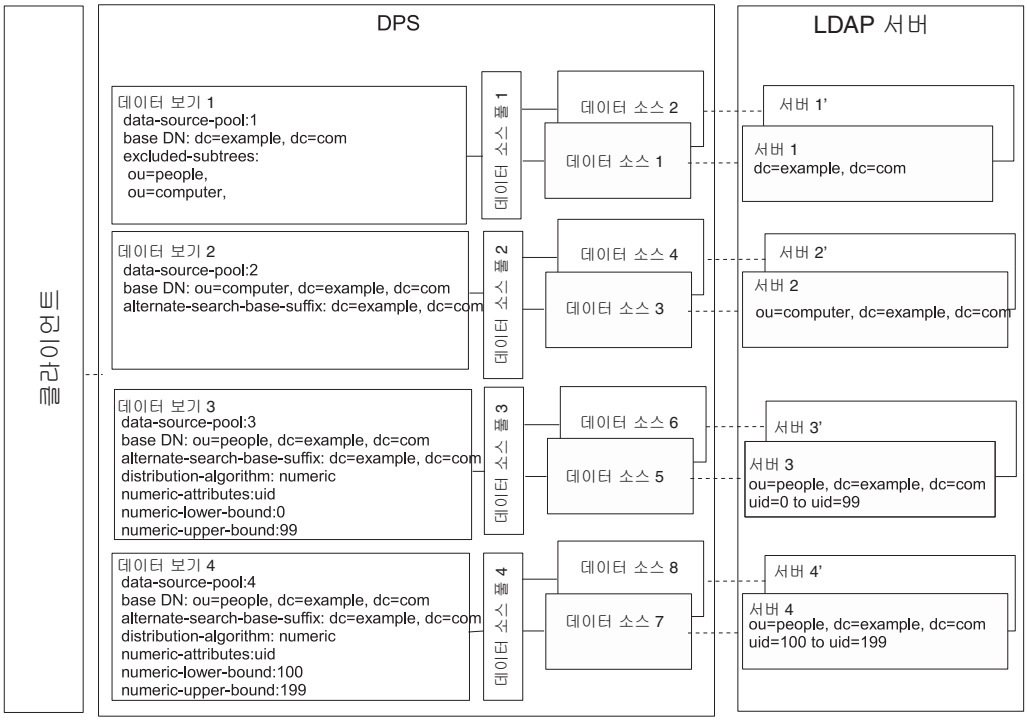


그림 22–5 계층 및 배포 알고리즘이 있는 데이터 보기 예

▼ 계층 및 배포 알고리즘이 있는 데이터 보기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 **357 페이지 "LDAP 데이터 소스 만들기 및 구성"에 설명된 것처럼 각 LDAP 서버에 대한 데이터 소스를 만듭니다.**

2 **360 페이지 "LDAP 데이터 소스 풀 만들기 및 구성"에 설명된 것처럼 네 개의 데이터 소스 풀을 만듭니다.**

3 **361 페이지 "데이터 소스 풀에 LDAP 데이터 소스 첨부"의 지침에 따라 데이터 소스를 데이터 소스 풀에 첨부합니다.**

- dc=example,dc=com을 포함하는 데이터 소스를 data-source-pool-1에 첨부합니다.
- ou=computer,dc=example,dc=com을 포함하는 데이터 소스를 data-source-pool-2에 첨부합니다.
- 0 및 99 사이의 uid가 있는 ou=people,dc=example,dc=com에서 항목을 포함하는 데이터 소스를 data-source-pool-3에 첨부합니다.
- 100 및 199 사이의 uid가 있는 ou=people,dc=example,dc=com에서 항목을 포함하는 데이터 소스를 data-source-pool-4에 첨부합니다.

4 **(옵션) 로드 균형 조정을 구성합니다.**

자세한 내용은 363 페이지 "로드 균형 조정 구성"을 참조하십시오.

5 **data-source-pool-1을 참조하는 dc=example,dc=com에서 기본 DN이 있는 데이터 보기를 만듭니다.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-1 \
 data-source-pool-1 dc=example,dc=com
```

6 **data-source-pool-2를 참조하는 ou=computer,dc=example,dc=com에서 기본 DN이 있는 데이터 보기를 만듭니다.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-2 \
 data-source-pool-2 ou=computer,dc=example,dc=com
```

7 **data-source-pool-3을 참조하는 ou=people,dc=example,dc=com에서 기본 DN이 있는 데이터 보기를 만듭니다. 0 및 99 사이의 uid가 있는 항목을 선택하도록 데이터 보기에서 배포 알고리즘을 구성합니다.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-3 \
 data-source-pool-3 ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop dataview-3 distribution-algorithm:numeric \
 numeric-attrs:uid numeric-lower-bound:0 numeric-upper-bound:99
```

8 **data-source-pool-4를 참조하는 ou=people,dc=example,dc=com에서 기본 DN이 있는 데이터 보기를 만들고 100 및 199 사이의 uid가 있는 항목을 선택하도록 데이터 보기에서 배포 알고리즘을 구성합니다.**

```
$ dpconf create-ldap-data-view -h host1 -p 1389 dataview-4 \
 data-source-pool-4 ou=people,dc=example,dc=com
$ dpconf set-ldap-data-view-prop dataview-4 distribution-algorithm:numeric \
 numeric-attrs:uid numeric-lower-bound:100 numeric-upper-bound:199
```

9 **excluded-subtrees 매개 변수를 확인하여 ou=computer,dc=example, dc=com 및 ou=people,dc=example, dc=com 하위 트리가 dataview-1에서 제외되었는지 확인합니다.**

```
$ dpconf get-ldap-data-view-prop -h host1 -p 1389 dataview-1 excluded-subtrees
```

제외된 하위 트리 목록이 반환됩니다.

10 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

◆ ◆ ◆ 23장

23 디렉토리 프록시 서버 가상화

이 장에서는 가상 데이터 보기를 만드는 방법에 대해 설명합니다. **가상 데이터 보기**는 소스 데이터를 변환하고 해당 데이터의 다른 보기를 클라이언트 응용 프로그램에 제공합니다. 가상 데이터 보기에는 변환된 LDAP 데이터 보기, LDIF 데이터 보기, 결합 데이터 보기 및 JDBC™ 데이터 보기가 포함됩니다. 가상 데이터 보기의 기능 개요와 사용 사례 예에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 18 장, "Directory Proxy Server Virtualization"을 참조하십시오.

디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)를 사용하여 이 장의 절차를 수행할 수 없습니다. 명령줄을 사용해야 합니다.

이 장은 다음 내용으로 구성되어 있습니다.

LDIF 데이터 보기 만들기 및 구성

LDIF 데이터 보기는 LDIF 파일이 LDAP 데이터 소스처럼 표시되는 간단한 가상 데이터 보기입니다. LDAP 데이터 보기와 달리 LDIF 데이터 보기를 설정할 때는 데이터 소스 또는 데이터 소스 풀을 만들지 않습니다. 대신, 데이터 보기를 만들 때 LDIF 파일을 지정합니다. 기본적으로 LDIF 데이터 보기에는 쓸 수 없습니다. 자세한 내용은 408 페이지 "가상 데이터 보기에서 액세스 제어 정의"를 참조하십시오.

LDIF 데이터 보기를 만들고 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ LDIF 데이터 보기를 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 LDIF 데이터 보기를 만듭니다.

```
$ dpconf create-ldif-data-view -h host -p port view-name path-to-ldif-file suffix-dn
```

2 (옵션) LDIF 데이터 보기 목록을 봅니다.

```
$ dpconf list-ldif-data-views -h host -p port
```

`virtual access controls` 데이터 보기는 유일한 기본 LDIF 데이터 보기입니다. 이 데이터 보기는 서버에서 생성되며 가상 액세스 제어 지침(ACI)에 요청을 전달하도록 합니다.

▼ LDIF 데이터 보기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 LDIF 데이터 보기의 등록 정보를 봅니다.

```
$ dpconf get-ldif-data-view-prop -h host -p port view-name
```

LDIF 데이터 보기에는 다음과 같은 기본 등록 정보가 있습니다.

```
alternate-search-base-dn                    :  ""
alternate-search-base-dn                    :  dc=com
attr-name-mappings                          :  none
base-dn                                     :  suffixDN
bind-pwd-attr                               :  userPassword
contains-shared-entries                     :  -
db-pwd-encryption                           :  clear-text
description                                 :  -
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  -
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
ldif-data-source                            :  /path/to/filename.ldif
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
```

```
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
process-bind                                :  -
replication-role                            :  master
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

2 **단계 1에 나열된 하나 이상의 등록 정보를 변경합니다.**

```
$ dpconf set-ldif-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

예를 들어 데이터 보기에 대한 소스 LDIF 파일을 변경하려면 `ldif-data-source` 등록 정보를 설정합니다.

```
$ dpconf set-ldif-data-view-prop -h host1 -p 1389 -D cn="Proxy Manager" \
myLDIFDataView ldif-data-source:/local/files/example.ldif
```

가상 데이터 변환 구성

가상 데이터 변환은 기존 데이터 보기에서 정의되고, 물리적 데이터 보기에서 가상 데이터 보기를 만듭니다. 작동하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Virtual Data Transformations"를 참조하십시오.

모든 유형의 데이터 보기, 즉 LDAP 데이터 보기, LDIF 데이터 보기, 결합 데이터 보기 또는 JDBC 데이터 보기에 가상 데이터 변환을 추가할 수 있습니다.

▼ 가상 변환을 추가하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 **데이터 보기에 변환을 추가합니다.**

```
$ dpconf add-virtual-transformation -h host -p port view-name \
 transformation-model transformation-action attribute-name [parameters...]
```

*parameters*는 *transformation-model* 및 *transformation-action*에 따라 필수일 수 있습니다. 변환 모델, 변환 작업 및 변환 매개 변수에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Virtual Data Transformations"를 참조하십시오.

2 (옵션) 데이터 보기에 정의된 가상 변환 목록을 봅니다.

```
$ dpconf list-virtual-transformations -h host -p port view-name
```

결합 데이터 보기 만들기 및 구성

결합 데이터 보기는 여러 데이터 보기의 집계입니다. 결합 데이터 보기가 작동하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Join Data Views"를 참조하십시오.

결합 데이터 보기를 만들고 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 결합 데이터 보기를 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 결합 보기를 형성하도록 집계할 기본 및 보조 데이터 보기를 식별합니다.

결합 보기를 만들려면 먼저 기본 및 보조 데이터 보기가 있어야 합니다. 기본 및 보조 보기는 LDAP 데이터 보기, LDIF 데이터 보기, JDBC 데이터 보기 또는 다른 결합 데이터 보기를 비롯한 모든 유형의 데이터 보기가 될 수 있습니다. 결합 보기의 소스로 작동하려면 보조 보기에서 특정 등록 정보를 구성해야 합니다. 자세한 내용은 400 페이지 "결합 보기의 보조 보기를 구성하는 방법"을 참조하십시오.

2 결합 데이터 보기를 만듭니다.

```
$ dpconf create-join-data-view -h host -p port view-name primary-view secondary-view \
 suffix-dn
```

3 (옵션) 데이터 보기가 성공적으로 만들어졌는지 확인하려면 결합 보기 목록을 봅니다.

```
$ dpconf list-join-data-views -h host -p port
```

▼ 결합 데이터 보기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 결합 데이터 보기의 등록 정보를 봅니다.

```
$ dpconf get-join-data-view-prop -h host -p port view-name
```

결합 데이터 보기의 기본 등록 정보는 다음과 같습니다.

```
alternate-search-base-dn                    :  ""
alternate-search-base-dn                    :  dc=com
attr-name-mappings                          :  none
base-dn                                     :  suffixDN
contains-shared-entries                     :  -
description                                 :  -
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  -
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
join-rule-control-enabled                   :  false
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
primary-view                                :  primary-view
process-bind                                :  -
replication-role                            :  master
secondary-view                              :  secondary-view
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

2 **단계 1에 나열된 하나 이상의 등록 정보를 변경합니다.**

```
$ dpconf set-join-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

예를 들어 데이터 소스의 기본 데이터 보기를 `myLDAPDataView`로 변경하려면 다음 명령을 사용합니다.

```
$ dpconf set-join-data-view-prop -h host1 -p 1389 -D cn="Proxy Manager" \
 myJoinDataView primary-view:myLDAPDataView
```

3 **결합 데이터 보기가 구성된 경우 기본 데이터 보기와 보조 데이터 보기에서 `viewable-attr` 및 `writable-attr` 등록 정보를 설정합니다.**

이 등록 정보를 설정하면 기본 데이터 보기와 보조 데이터 보기에서 검색 필터를 적절하게 분할할 수 있습니다. 그렇지 않으면 검색 필터에 보조 데이터 보기의 속성이 포함되어 있을 때 검색 결과가 불일치할 수 있습니다.

4 **필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.**

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

▼ 여러 결합 데이터 보기에서 데이터 보기를 참조할 수 있도록 결합 데이터 보기를 구성하는 방법

결합 데이터 보기에서 결합 규칙 구성 정보를 설정하면 여러 결합 데이터 보기에서 데이터 보기를 참조할 수 있습니다. 이렇게 하려면 다음을 수행합니다.

1 **결합 데이터 보기에서 `join-rule-control-enabled`를 `true`로 설정합니다.**

```
$ dpconf set-join-data-view-prop view-name join-rule-control-enabled:true
```

`join-rule-control-enabled`를 `true`로 설정한 후에는 결합 데이터 보기에 저장된 결합 규칙 구성 정보가 서버에서 사용됩니다. 결합 규칙 구성 정보가 있는 결합 데이터 보기가 보조 데이터 보기에 저장되어 있는 경우 이 정보는 서버에서 사용되지 않습니다. 서버에서 이 정보를 사용하려면 결합 데이터 보기 수준에서 구성 정보를 수동으로 추가해야 합니다.

2 **보조 보기가 기본 보기와 관련된 방식을 결정하는 결합 규칙을 정의합니다.**

결합 규칙은 다음 중 하나일 수 있습니다.

- DN 결합 규칙

```
$ dpconf set-join-data-view-prop view-name \
dn-join-rule:uid=\${primary-view-name.uid},ou=People,dc=example
```

- 필터 결합 규칙

```
$ dpconf set-join-data-view-prop view-name \
filter-join-rule:uid=\${primary-view-name.uid}
```

▼ 결합 보기의 보조 보기를 구성하는 방법

결합 보기의 소스로 작동하려면 보조 데이터 보기에서 특정 등록 정보를 구성해야 합니다. 보조 보기는 모든 유형의 데이터 보기가 될 수 있으므로 사용하는 명령은 데이터 보기 유형에 따라 달라집니다. 다음 명령 예에서는 보조 보기를 LDAP 데이터 보기로

가정합니다. 여기에 설명된 등록 정보에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Additional Secondary Data View Properties"를 참조하십시오.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 보조 보기가 기본 보기와 관련된 방식을 결정하는 결합 규칙을 정의합니다.

결합 규칙은 다음 중 하나일 수 있습니다.

- DN 결합 규칙

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name \
dn-join-rule:uid=\${primary-view-name.uid},ou=People,dc=example
```

- 필터 결합 규칙

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name \
filter-join-rule:uid=\${primary-view-name.uid}
```

`dn-join-rule` 및 `filter-join-rule` 등록 정보의 구성은 결합 데이터 보기의 `join-rule-control-enabled` 등록 정보가 `false`로 설정된 경우에만 서버에서 사용됩니다. 그렇지 않으면 결합 데이터 보기에 `join-rule-control-enabled` 등록 정보가 `true`로 설정된 경우 보조 보기에 설정된 정보는 무시됩니다.

2 결합 데이터 보기에 필터 결합 규칙이 설정되어 있는 경우 결합 데이터 보기에 항목을 추가하려면 보조 데이터 보기에 가상 변환 규칙을 설정해야 합니다.

```
dpconf add-virtual-transformation secondary-view-name \
write add-attr-value dn uid=\${uid}
```

주 – 이 규칙을 설정하지 않으면 결합 데이터 보기에 항목을 추가할 수 없습니다.

---

3 (옵션) 보조 보기에서 바인드가 허용되는지 여부를 지정합니다.

기본적으로 바인드는 모든 데이터 보기에서 허용됩니다. 보조 데이터 보기에 대한 바인드를 금지하려면 다음 명령을 실행합니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name process-bind:false
```

이 등록 정보에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Handling of Binds"를 참조하십시오.

4 (옵션) 보조 보기에 공유 항목이 포함되는지 여부를 지정합니다.

```
$ dpconf set-ldap-data-view-prop -h host -p port secondary-view-name \
contains-shared-entries:true
```

이 등록 정보에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Handling of Shared Entries"를 참조하십시오.

JDBC 데이터 보기 만들기 및 구성

JDBC 데이터 보기를 사용하면 LDAP 클라이언트 응용 프로그램이 관계형 데이터베이스에 액세스할 수 있습니다. JDBC 데이터 보기가 작동하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "JDBC Data Views"를 참조하십시오.

JDBC 데이터 보기를 만들고 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ JDBC 데이터 보기를 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 관계형 데이터베이스에 대한 JDBC 데이터 소스를 만듭니다.

```
$ dpconf create-jdbc-data-source -h host -p port -b db-name -B db-url -J driver-url \
[-J driver-url]... -S driver-class source-name
```

현재는 각 JDBC 데이터 보기에 대해 한 개의 JDBC 데이터 소스만 지원됩니다. 즉, 여러 JDBC 데이터 소스에서 로드 균형 조정을 수행할 수 없습니다. 여러 JDBC 데이터 소스에 액세스하려면 각 데이터 소스에 대한 데이터 보기를 만들어 결합 데이터 보기와 결합할 수 있습니다.

JDBC 데이터 소스를 만드는 경우 다음 등록 정보를 설정해야 합니다.

`db-name` 관계형 데이터베이스 이름(예: `payrolldb`).

`db-url` `jdbc:` *vendor*`:`*driver*`://`*dbhost*`:` *dbport* 형식의 데이터베이스 URL.

`db-url`에는 데이터베이스 이름이 없으므로 완전한 JDBC 데이터베이스 URL이 **아닙니다**. (데이터베이스 이름은 `db-name` 등록 정보로 지정됩니다.)

MySQL, DB2 및 Derby 데이터베이스의 경우에는 `db-url`이 `/`로 끝나야 하고, Oracle 데이터베이스의 경우에는 `:`으로 끝나야 합니다.

`driver-class` JDBC 드라이버 클래스(예: `org.hsqldb.jdbcDriver`).

`driver-url` JDBC 드라이버 경로(예: `file:///` *path/to/hsqldb*`/lib/hsqldb.jar`).

`driver-url` 등록 정보는 여러 값을 가집니다. 따라서 `driver-url`은 JDBC 드라이버에 여러 개의 JAR 파일을 지원하여 여러 플랫폼의 JDBC 소스에 연결할 수 있습니다.

2 JDBC 데이터 소스 풀을 만듭니다.

```
$ dpconf create-jdbc-data-source-pool -h host -p port pool-name
```

3 **JDBC 데이터 소스를 JDBC 데이터 소스 풀에 첨부합니다.**

```
$ dpconf attach-jdbc-data-source -h host -p port pool-name source-name
```

4 **JDBC 데이터 보기를 만듭니다.**

```
$ dpconf create-jdbc-data-view -h host -p port view-name pool-name suffix-DN
```

5 **(옵션) 데이터 보기가 성공적으로 만들어졌는지 확인하려면 JDBC 데이터 보기 목록을 봅니다.**

```
$ dpconf list-jdbc-data-views -h host -p port
```

▼ JDBC 데이터 보기를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 **JDBC 데이터 보기의 등록 정보를 봅니다.**

```
$ dpconf get-jdbc-data-view-prop -h host -p port view-name
```

JDBC 데이터 보기의 기본 등록 정보는 다음과 같습니다.

```
alternate-search-base-dn                    :  -
attr-name-mappings                          :  none
base-dn                                     :  o=sql1
contains-shared-entries                     :  -
description                                 :  -
distribution-algorithm                      :  -
dn-join-rule                                :  -
dn-mapping-attrs                            :  none
dn-mapping-source-base-dn                   :  none
excluded-subtrees                           :  -
filter-join-rule                            :  -
is-enabled                                  :  true
is-read-only                                :  false
is-routable                                 :  true
jdbc-data-source-pool                       :  pool-name
lexicographic-attrs                         :  all
lexicographic-lower-bound                   :  none
lexicographic-upper-bound                   :  none
non-viewable-attr                           :  -
non-writable-attr                           :  -
numeric-attrs                               :  all
numeric-default-data-view                   :  false
numeric-lower-bound                         :  none
numeric-upper-bound                         :  none
```

```
pattern-matching-base-object-search-filter  :  all
pattern-matching-dn-regular-expression      :  all
pattern-matching-one-level-search-filter    :  all
pattern-matching-subtree-search-filter      :  all
process-bind                                :  -
replication-role                            :  master
viewable-attr                               :  all except non-viewable-attr
writable-attr                               :  all except non-writable-attr
```

2 단계 1에 나열된 하나 이상의 등록 정보를 변경합니다.

```
$ dpconf set-jdbc-data-view-prop -h host -p port view-name property:value \
 [property:value ... ]
```

▼ JDBC 테이블, 속성 및 객체 클래스 구성

JDBC 데이터 보기를 구성할 경우에는 다음 객체도 구성해야 합니다.

- **JDBC 객체 클래스.** 하나 이상의 JDBC 테이블을 LDAP 객체 클래스에 매핑합니다.
- **JDBC 테이블.** 각 관계형 데이터베이스 테이블에 대해 정의됩니다.
- **JDBC 속성.** JDBC 테이블의 지정된 열에서 LDAP 속성을 정의합니다.

1 관계형 데이터베이스에서 각 테이블에 대한 JDBC 테이블을 만듭니다.

```
% dpconf create-jdbc-table jdbc-table-name db-table
```

*db-table*의 이름은 대소문자를 구분합니다. 관계형 데이터베이스에 사용된 것과 동일한 대소문자를 사용해야 하며, 그렇지 않으면 해당 테이블에 대한 작업이 실패할 수 있습니다.

2 각 관계형 데이터베이스 테이블에서 각 열에 대한 JDBC 속성을 만듭니다.

```
% dpconf add-jdbc-attr table-name attr-name sql-column
```

JDBC 속성을 만들면 테이블 열이 LDAP 속성에 매핑됩니다.

3 (옵션) 관계형 데이터베이스의 열이 대소문자를 구분할 경우 JDBC 속성의 LDAP 구문을 변경합니다.

```
% dpconf set-jdbc-attr-prop table-name attr-name ldap-syntax:ces
```

`ldap-syntax`의 값은 기본적으로 `cis`입니다. 이는 `jdbc-attr`이 대소문자를 구분하지 않는다는 것을 의미합니다. 관계형 데이터베이스가 대소문자를 구분할 경우 값을 `ces`로 변경합니다.

Oracle 및 DB2와 같은 특정 관계형 데이터베이스는 기본적으로 대소문자를 구분합니다. LDAP는 기본적으로 대소문자를 구분하지 않습니다. 디렉토리 프록시 서버에서 관계형 데이터베이스 테이블의 열이 대소문자를 구분한다는 것을 감지하는 경우 필터에서 해당 속성이 있는 `ldapsearch` 쿼리가 `UPPER` 함수를 사용하여 SQL 쿼리로 변환됩니다.

예를 들어 `ldapsearch -b "dc=mysuffix" "(attr=abc)"` 쿼리는 다음 SQL 쿼리로 변환됩니다.

```
SELECT * FROM mytable WHERE (UPPER(attr)='ABC')
```

기본적으로 이 유형의 쿼리는 색인화되지 않습니다. 따라서 이 특성을 가진 쿼리는 성능에 큰 영향을 줄 수 있습니다.

다음 두 가지 방법으로 성능에 미치는 영향을 줄일 수 있습니다.

- jdbc-attr의 `ldap-syntax` 등록 정보를 `ces`로 설정
- LDAP 필터에서 사용할 수 있는 각 `jdbc-attr`에 대해 `UPPER` 함수를 사용하여 색인 만들기

주 – 관계형 데이터베이스가 대소문자를 구분하지 않는 경우 `ldap-syntax`를 기본값 즉, `cis`로 사용합니다. `ldap-syntax:ces`는 대소문자를 구분하지 않는 데이터베이스에서 지원되지 않습니다.

---

4 LDAP 관계형 데이터베이스 테이블에 대한 JDBC 객체 클래스를 만듭니다.

```
% dpconf create-jdbc-object-class view-name objectclass primary-table \
  [secondary-table... ] DN-pattern
```

기본적으로 JDBC 객체 클래스를 만들면 해당 테이블과 연관되는 LDAP 객체 클래스가 지정됩니다. 또한 JDBC 객체 클래스는 기본 테이블과 보조 테이블을 지정합니다(있을 경우).

JDBC 객체 클래스를 만들 때 DN 패턴을 지정합니다. DN 패턴은 항목의 DN이 구성되는 방법을 보여줍니다.

JDBC 객체 클래스의 DN 패턴에서 정의된 모든 하위 트리 구성 요소에는 해당 구성 요소에 대해 정의된 JDBC 객체 클래스가 있어야 합니다. 예를 들어 JDBC 객체 클래스에 DN 패턴 `uid,ou`가 있는 경우 JDBC 객체 클래스 정의에는 DN 패턴 `ou`가 포함되어야 합니다. 디렉토리 프록시 서버에서는 올바로 구성된 DIT를 구성해야 합니다. 그렇지 않으면 `ou=xxx,`*base-DN*과 같은 값이 있는 하위 트리는 검색 결과로 반환되지 않습니다.

5 보조 테이블이 있을 경우 기본 테이블과 보조 테이블 간에 결합 규칙을 정의합니다.

```
% dpconf set-jdbc-table-prop secondary-table-name filter-join-rule:join-rule
```

결합 규칙은 보조 테이블에 정의되며 해당 테이블의 데이터가 기본 테이블의 테이터에 연결되는 방법을 결정합니다. 객체 클래스의 기본 테이블과 보조 테이블 간에 관계를 정의하는 방법은 중요합니다. 자세한 내용은 406 페이지 “JDBC 테이블 간의 관계 정의”를 참조하십시오.

6 JDBC 객체 클래스에 대한 수퍼 클래스를 지정합니다.

```
% dpconf set-jdbc-object-class-prop view-name objectclass super-class:value
```

수퍼 클래스는 JDBC 객체 클래스가 상속하는 LDAP 객체 클래스를 나타냅니다.

JDBC 테이블 간의 관계 정의

가장 단순한 경우 JDBC 객체 클래스는 단일(기본) 테이블만 포함합니다. 이 테이블에는 보조 테이블이 없으므로 테이블 간에 관계를 정의할 필요가 없습니다.

객체 클래스에 둘 이상의 테이블이 포함된 경우에는 해당 테이블 간의 관계를 분명하게 정의해야 합니다. 테이블 간의 관계는 항상 보조 테이블에서 정의됩니다. 보조 테이블의 다음 등록 정보를 사용하여 이러한 관계를 정의할 수 있습니다.

- `is-single-row-table`은 테이블에서 LDAP 항목과 일치하는 행이 한 개만 있도록 지정합니다.
- `contains-shared-entries`는 보조 테이블의 행이 기본 테이블의 둘 이상의 행에 사용되도록 지정합니다.
- `filter-join-rule`은 기본 테이블의 항목을 기준으로 보조 테이블에서 항목을 검색하는 방법을 나타냅니다.

다음 예는 처음 두 개의 등록 정보 값을 기준으로 필터 결합 규칙을 정의하는 방법을 보여줍니다. 다음 예에서는 객체 클래스에 기본 테이블과 보조 테이블이 한 개씩 있다고 가정합니다.

예 23–1 `is-single-row-table:true` 및 `contains-shared-entries:true`

이 값은 해당 등록 정보의 기본값입니다. 이 경우 기본 테이블과 보조 테이블 간의 관계는 *n->1*입니다. 즉, 기본 테이블의 *n*개 행이 보조 테이블의 공유된 행 하나를 참조합니다.

관계형 데이터베이스에서 외래 키(FK)가 기본 테이블에 정의되고 보조 테이블의 열을 가리킵니다.

예를 들어 조직에서 여러 직원이 동일한 관리자를 공유할 수 있다고 가정합니다. 이 경우 다음 구조를 가진 두 개의 관계형 데이터베이스 테이블이 정의됩니다.

```
primary table : EMPLOYEE [ID, NAME, FK_MANAGER_ID]
secondary table : MANAGER  [ID, NAME]
```

다음 객체 클래스와 속성이 정의됩니다.

```
object-class : employee
attr : name (from primary EMPLOYEE.NAME)
attr : manager (from secondary MANAGER.NAME)
```

다음 필터 결합 규칙이 보조 테이블에 정의됩니다.

```
"${ID}=${EMPLOYEE.FK_MANAGER_ID}"
```

예 23–1 is-single-row-table:true 및 contains-shared-entries:true *(계속)*

이 구성을 사용하면 LDAP 작업에 대해 다음 동작이 발생합니다.

- **직원 항목 추가.**직원 항목의 관리자가 테이블에 없는 경우 새 행이 만들어집니다. 관리자가 있는 경우 기존 행이 사용됩니다.
- **항목에서 "manager" 속성 값 바꾸기.**MANAGER.NAME 행의 값이 변경됩니다.
- **직원 항목 삭제.**관리자 항목이 공유되므로 보조 테이블의 행이 삭제되지 않습니다.
- **항목에서 "manager" 속성 삭제.**보조 테이블의 행이 삭제되고 외래 키(EMPLOYEE.FK_MANAGER_ID)가 NULL로 설정됩니다.

예 23–2 is-single-row-table:true 및 contains-shared-entries:false

이 경우 기본 테이블과 보조 테이블 간의 관계는 *1->1* 또는 *1<-1*입니다. 즉, 기본 테이블의 한 행을 보조 테이블의 한 행에서 참조합니다.

관계형 데이터베이스에서 외래 키(FK)는 기본 테이블 또는 보조 테이블에 정의할 수 있습니다.

예를 들어 조직에서 직원의 UID가 하나의 테이블에 저장되고 직원의 성이 다른 테이블에 저장된다고 가정합니다. 이 경우 다음 구조를 가진 두 개의 관계형 데이터베이스 테이블이 정의됩니다.

```
primary table : UID [ID, VALUE, FK_SN_ID]
secondary table : SN [ID, VALUE]
```

다음 객체 클래스와 속성이 정의됩니다.

```
object-class : employee
attr : uid (from primary UID.VALUE)
attr : sn (from secondary ID.VALUE)
```

다음 필터 결합 규칙이 보조 테이블에 정의됩니다.

```
"${ID}=${UID.FK_SN_ID}"
```

이 구성은 외래 키 FK_UID_ID가 보조 테이블에 저장되고 UID.ID를 가리키는 정반대 구성이 될 수 있습니다.

예 23–3 is-single-row-table:false 및 contains-shared-entries:false

이 경우 기본 테이블과 보조 테이블 간의 관계는 *1->n*입니다. 즉, 기본 테이블의 한 행을 보조 테이블의 *n*개 행에서 참조합니다. 이 예는 값이 여러 개인 속성의 경우를 보여줍니다. 값이 여러 개인 속성은 보조 테이블에서 행 집합으로 표시되며 속성 값당 한 행을 가집니다.

관계형 데이터베이스에서 외래 키는 보조 테이블에 정의되고 기본 테이블의 열을 가리킵니다.

예를 들어 조직에서 직원이 여러 전화 번호를 가질 수 있다고 가정합니다. 이 경우 다음 구조를 가진 두 개의 관계형 데이터베이스 테이블이 정의됩니다.

```
primary table : EMPLOYEE [ID, NAME]
secondary table : PHONE [ID, VALUE, USER_ID]
```

다음 객체 클래스와 속성이 정의됩니다.

```
object-class : employee
attr : cn (from primary EMPLOYEE.NAME)
attr : telephoneNumber (from secondary PHONE.VALUE)
```

다음 필터 결합 규칙이 보조 테이블에 정의됩니다.

```
"${USER_ID}=${EMPLOYEE.ID}"
```

예 23–4 is-single-row-table:false 및 contains-shared-entries:true

이 경우는 현재 디렉토리 프록시 서버에서 지원되지 않습니다.

가상 데이터 보기에서 액세스 제어 정의

가상 데이터 보기에서 ACI는 LDAP 디렉토리 또는 LDIF 파일에 저장할 수 있습니다. 가상 ACI가 작동하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Access Control On Virtual Data Views"를 참조하십시오.

디렉토리 프록시 서버 인스턴스를 만드는 경우 가상 액세스 제어에 대해 다음 기본 구성이 정의됩니다.

- ACI가 기본적으로 저장되는 LDIF 파일(*instance-path*/config/access_controls.ldif)
- 이름이 **가상 액세스 제어**인 LDIF 데이터 보기

  이 데이터 보기를 사용하여 디렉토리 프록시 서버에서 LDIF 파일에 저장된 ACI에 액세스할 수 있습니다.

▼ 새 ACI 저장소를 정의하는 방법

앞에 설명된 기본 ACI 구성을 사용하지 않으려면 다른 저장소를 정의할 수 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 가상 ACI가 저장되는 저장소에 대한 데이터 보기를 만듭니다.

- ACI가 LDAP 디렉토리에 저장되는 경우 375 페이지 "LDAP 데이터 보기 만들기 및 구성"에 설명된 것처럼 LDAP 데이터 소스 및 LDAP 데이터 보기를 만듭니다.
- ACI가 LDIF 파일에 저장되는 경우 395 페이지 "LDIF 데이터 보기 만들기 및 구성"에 설명된 것처럼 LDIF 데이터 보기를 만듭니다.

2 이전 단계에서 만든 데이터 보기의 이름을 ACI 데이터 보기로 지정합니다.

```
$ dpconf set-virtual-aci-prop -h host -p port aci-data-view:data-view-name
```

3 ACI 저장소가 LDAP 디렉토리인 경우 ACI 데이터 보기를 액세스하는 데 필요한 자격 증명을 정의합니다.

```
$ dpconf set-virtual-aci-prop -h host -p port aci-manager-bind-dn:bind-dn
$ dpconf set-virtual-aci-prop -h host -p port aci-manager-bind-pwd-file:filename
```

▼ 가상 액세스 제어를 구성하는 방법

사용하는 ACI 저장소에 상관없이 가상 액세스 제어를 구성해야 합니다.

주 – 프록시 관리자만 ACI 데이터 보기를 통해 ACI 풀을 만들고 ACI를 직접 관리할 수 있습니다. ACI 저장소가 LDAP 디렉토리인 경우 `aciSource` 객체 클래스 및 `dpsaci` 속성을 포함하도록 해당 디렉토리의 스키마를 수정해야 합니다. 스키마를 사용자 정의하는 방법에 대한 자세한 내용은 291 페이지 "디렉토리 서버 스키마 확장"을 참조하십시오.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 ACI 저장소에서 ACI 풀을 만들고 전역 ACI를 설정합니다.

전역 ACI에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Global ACIs"를 참조하십시오. 전역 ACI를 설정하려면 ACI 데이터 보기의 보기 기준 아래에 `aciSource` 항목을 추가합니다. 예를 들면 다음과 같습니다.

```
% ldapmodify -p port -D "cn=proxy manager" -w -
dn: cn=data-source-name,cn=virtual access controls
changetype: add
```

```
objectclass: aciSource
dpsaci: (targetattr="*") (target = "ldap:///ou=people,o=virtual") (version 3.0; \
 acl "perm1"; allow(all) groupdn="ldap:///cn=virtualGroup1,o=groups,o=virtual";)
cn: data-source-name
```

2 이 ACI 풀을 사용하도록 하나 이상의 연결 처리기를 구성합니다.

```
% dpconf set-connection-handler-prop -h host -p port connection-handler \
aci-source:data-source-name
```

3 필요한 ACI를 데이터에 추가합니다.

이렇게 하려면 ACI를 포함하는 가상 항목을 만듭니다. 예를 들면 다음과 같습니다.

```
% ldapmodify -p port -D "cn=virtual application,ou=application users,dc=com" -w -
dn: ou=people,o=virtual
changetype: modify
add: dpsaci
dpsaci: (targetattr="*")(version 3.0; acl "perm1"; allow(all) userdn ="ldap:///self";)
dpsaci: (targetattr="*")(version 3.0; acl "perm1"; allow(search, read, compare) \
 userdn ="ldap:///anyone";)
```

주 – 적절한 액세스 권한을 가진 모든 사용자가 데이터 보기를 통해 가상 ACI를 추가하고 검색할 수 있습니다.

가상 데이터 보기에서 스키마 검사 정의

일반적으로 LDAP 데이터 보기의 경우 스키마 검사는 백엔드 디렉토리의 스키마를 사용하여 백엔드 디렉토리에서 수행됩니다. 디렉토리 프록시 서버에서 스키마 검사를 수행하려면 다음 절차를 사용합니다.

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

요청, 특히 DN을 정규화하려면 다음과 같이 서버의 `use-external-schema` 등록 정보를 설정합니다.

▼ 스키마 검사를 정의하는 방법

1 서버 인스턴스가 외부 스키마를 사용해야 함을 나타냅니다.

```
$ dpconf set-server-prop -h host -p port use-external-schema:true
```

2 **연결 처리기에서 스키마 검사를 활성화합니다.**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler \
 schema-check-enabled:true
```

3 **cn=schema를 표시하는 데이터 보기를 만듭니다.**

외부 스키마가 LDAP 디렉토리에 정의된 경우 375 페이지 "LDAP 데이터 보기 만들기 및 구성"에 설명된 것처럼 cn=schema의 보기 기준을 사용하여 LDAP 데이터 보기를 만듭니다.

외부 스키마가 LDIF 파일에 정의된 경우 395 페이지 "LDIF 데이터 보기 만들기 및 구성"에 설명된 것처럼 cn=schema의 보기 기준을 사용하여 LDIF 데이터 보기를 만듭니다.

4 **연결 처리기에서 표시되는 데이터 보기 목록에 이 데이터 보기를 추가합니다.**

기본적으로 모든 데이터 보기는 연결 처리기에서 표시됩니다. 연결 처리기에서 표시되는 데이터 보기의 사용자 정의 목록을 정의한 경우 이 데이터 보기를 해당 목록에 추가합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler \
 data-view-routing-custom-list+:data-view-name
```

가상 구성 예

다음 절에서는 두 개의 구성 예에 대해 설명합니다. 이러한 구성은 가상 디렉토리의 기본 기능과 이러한 기능을 구성하는 방법을 보여줍니다.

LDAP 디렉토리 및 MySQL 데이터베이스 결합

이 절의 절차에서는 LDAP 디렉토리 및 MySQL 데이터베이스를 결합하는 가상 구성 예에 대해 설명합니다. LDAP 디렉토리는 대부분의 사용자 정보를 포함하는 기본 데이터 소스입니다. mySQL 데이터베이스에는 사용자에 대한 추가 정보가 포함됩니다. 결과로 표시된 구성은 다음 그림과 같습니다.

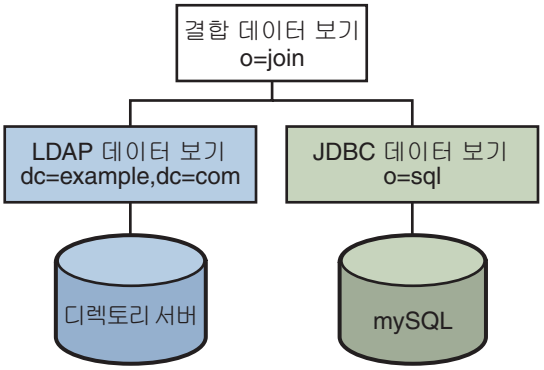


그림 23–1 가상 구성 예

install-path /ds6/ldif/Example.ldif에 제공된 샘플 데이터를 사용하여 이 예를 그대로 사용하거나 샘플 데이터를 고유한 데이터로 바꿀 수 있습니다.

이 구성은 다음 세 개의 절로 구분할 수 있습니다.

- LDAP 데이터 보기 구성 및 테스트
- JDBC 데이터 보기 구성 및 테스트
- 결합 데이터 보기 구성 및 테스트

단순하게 만들기 위해 이 절의 모든 명령에서는 디렉토리 프록시 서버가 /local/dps의 로컬 호스트에서 실행 중이라고 가정합니다. 또한, 다음 환경 변수가 설정되었다고 가정합니다.

DIR_PROXY_PORT	1389
LDAP_ADMIN_PWF	pwd.txt, 관리자 비밀번호를 포함하는 파일
DIRSERV_PORT	4389
LDAP_ADMIN_USER	cn=Directory Manager

LDAP 데이터 보기 구성 및 테스트

▼ LDAP 데이터 보기를 구성하는 방법

시작하기 전에 이 절의 작업에서는 다음 정보를 가정합니다.

- 디렉토리 서버 인스턴스가 포트 4389의 host1에서 실행 중입니다.
- 디렉토리 서버의 데이터가 dc=example,dc=com 접미어 아래에 저장됩니다. 이 예를 그대로 사용하려면 디렉토리 서버 인스턴스를 만들고 dc=example,dc=com 접미어를 만든 다음 *install-path*/ds6/ldif/Example.ldif의 샘플 데이터를 가져옵니다.

1 **디렉토리 서버 인스턴스에 대해 myds1이라는 LDAP 데이터 소스를 만듭니다.**

```
% dpconf create-ldap-data-source myds1 host1:4389
```

2 **데이터 소스를 활성화하고 데이터 소스에 대한 쓰기 작업을 허용합니다.**

```
% dpconf set-ldap-data-source-prop myds1 is-enabled:true is-read-only:false
```

3 **myds1-pool이라는 LDAP 데이터 소스 풀을 만듭니다.**

```
% dpconf create-ldap-data-source-pool myds1-pool
```

4 **LDAP 데이터 소스를 LDAP 데이터 소스 풀에 첨부합니다.**

```
% dpconf attach-ldap-data-source myds1-pool myds1
```

5 **데이터 소스가 해당 데이터 소스 풀에서 바인드, 추가, 검색 및 수정 작업을 100% 받아야 함을 지정합니다.**

```
% dpconf set-attached-ldap-data-source-prop myds1-pool myds1 add-weight:100 \
 bind-weight:100 modify-weight:100 search-weight:100
```

6 **dc=example,dc=com의 기본 DN을 사용하여 데이터 소스 풀에 대한 myds1–view라는 LDAP 데이터 보기를 만듭니다.**

```
% dpconf create-ldap-data-view myds1-view myds1-pool dc=example,dc=com
```

▼ LDAP 데이터 보기를 테스트하는 방법

1 **dc=example,dc=com의 사용자로 LDAP 데이터 소스의 모든 항목을 검색하여 데이터 보기에서 읽을 수 있는지 확인합니다.**

```
% ldapsearch -p 1389 -D "uid=kvaughan,ou=people,dc=example,dc=com" -w bribery \
 -b dc=example,dc=com "objectclass=*"
```

주 – dc=example,dc=com의 사용자에 대한 자격 증명을 사용해야 합니다. cn=Directory Manager를 사용하려면 해당 DN을 처리하도록 데이터 보기를 정의해야 합니다.

---

2 **dc=example,dc=com의 사용자로 userPassword 속성을 수정하여 데이터 보기에 쓸 수 있는지 확인합니다.**

```
% ldapmodify -p 1389 -D "uid=kvaughan,ou=people,dc=example,dc=com" -w bribery
dn: uid=kvaughan,ou=people,dc=example,dc=com
changetype: modify
replace: userPassword
userPassword: myNewPassword
```

주 – 디렉토리 서버의 기본 ACI를 사용하여 자신의 비밀번호를 수정할 수 있습니다.

JDBC 데이터 보기 구성 및 테스트

다음 작업에서는 mySQL 데이터베이스가 설치되어 실행 중이고 데이터로 채워져 있으며, 다음과 같은 특징이 있다고 가정합니다.

- 데이터베이스 이름: `sample_sql`
- 데이터베이스 URL: `host2.example.com:3306/`
- JDBC 드라이버 URL: `file:/net/host2.example/local/mysql/lib/jdbc.jar`
- 드라이버 클래스: `com.mysql.jdbc.Driver`
- 데이터베이스 사용자: `root`
- 데이터베이스 비밀번호 파일: `mysqlpwd.txt`

다음 표에서는 데이터베이스의 테이블과 해당 복합 필드에 대해 설명합니다. JDBC 데이터 보기를 설정하려면 이 정보가 필요합니다.

mySQL 테이블	필드
EMPLOYEE	ID, SURNAME,PASSWORD, TITLE, COUNTRY_ID
COUNTRY	ID, NAME
PHONE	USER_ID, NUMBER

▼ JDBC 데이터 보기를 구성하는 방법

1 SQL 데이터베이스에 mysql1이라는 JDBC 데이터 소스를 만듭니다.

```
% dpconf create-jdbc-data-source -b sample_sql -B jdbc:mysql://host2.example.com:3306 \
 -J file:/net/host2.example/local/mysql/lib/jdbc.jar -S com.mysql.jdbc.Driver mysql1
```

2 SQL 데이터베이스에 대한 사용자 이름과 비밀번호 파일을 지정합니다.

```
% dpconf set-jdbc-data-source-prop mysql1 db-pwd-file:sqlpwd.txt db-user:root
```

3 프록시 서버를 다시 시작합니다.

```
% dpadm restart /local/dps
```

4 데이터 소스를 활성화하고 데이터 소스에 대한 쓰기 작업을 허용합니다.

```
% dpconf set-jdbc-data-source-prop mysql1 is-enabled:true is-read-only:false
```

5 `mysql1_pool`이라는 JDBC 데이터 소스 풀을 만듭니다.

```
% dpconf create-jdbc-data-source-pool mysql1-pool
```

6 JDBC 데이터 소스를 데이터 소스 풀에 첨부합니다.

```
% dpconf attach-jdbc-data-source mysql1-pool mysql1
```

7 o=sql의 기본 DN을 사용하여 데이터 소스 풀에 대한 myjdbc1–view라는 JDBC 데이터 보기를 만듭니다.

```
% dpconf create-jdbc-data-view mysql1-view mysql1-pool o=sql
```

8 MySQL 데이터베이스의 각 테이블에 대한 JDBC 테이블을 만듭니다.

```
% dpconf create-jdbc-table employee1 EMPLOYEE
% dpconf create-jdbc-table country1 COUNTRY
% dpconf create-jdbc-table phone1 PHONE
```

SQL 데이터베이스의 테이블 이름은 대소문자를 구분합니다. SQL 데이터베이스에 사용된 것과 동일한 대소문자를 사용해야 합니다.

9 각 테이블의 열마다 JDBC 속성을 만듭니다.

JDBC 속성을 만들면 MySQL 열이 LDAP 속성에 매핑됩니다.

```
% dpconf add-jdbc-attr employee1 uid ID
% dpconf add-jdbc-attr employee1 sn SURNAME
% dpconf add-jdbc-attr employee1 userPassword PASSWORD
% dpconf add-jdbc-attr employee1 room ROOM
% dpconf add-jdbc-attr phone1 tel NUMBER
% dpconf add-jdbc-attr country1 country NAME
```

phone1 user_id 및 country1 id 열은 MySQL 데이터베이스의 컨텍스트에서만 사용되므로 이러한 열에 대한 JDBC 속성을 만들 필요가 없습니다. 이러한 열에는 해당 LDAP 속성이 없습니다.

10 LDAP person 객체 클래스에 대한 JDBC 객체 클래스를 만듭니다.

이 단계에서 employee1 테이블은 기본 테이블로 식별되고 country1 및 phone1 테이블은 보조 테이블로 식별됩니다. 또한 JDBC 객체 클래스를 만들려면 DN이 필요합니다. 이 예에서 DN은 데이터 보기의 기본 DN 및 uid 속성에서 구성됩니다.

```
% dpconf create-jdbc-object-class mysql1-view person employee1 country1 phone1 uid
```

11 기본 테이블과 보조 테이블 간의 결합 규칙을 정의합니다.

결합 규칙은 보조 테이블에 정의되며 해당 테이블의 데이터가 기본 테이블의 데이터에 연결되는 방법을 결정합니다.

```
% dpconf set-jdbc-table-prop country1 filter-join-rule:'ID=${EMPLOYEE.COUNTRY_ID}'
% dpconf set-jdbc-table-prop phone1 filter-join-rule:'USER_ID=${EMPLOYEE.ID}'
```

12 JDBC 객체 클래스에 대한 수퍼 클래스를 지정합니다.

수퍼 클래스는 JDBC 객체 클래스가 속성을 상속하는 LDAP 객체 클래스를 나타냅니다.

```
% dpconf set-jdbc-object-class-prop mysql1-view person super-class:top
```

▼ 필요한 ACI를 만드는 방법

JDBC 데이터 보기를 테스트하기 전에 ACI를 구성하여 데이터 보기에 대한 쓰기 액세스 권한을 활성화해야 합니다. 기본적으로 비 LDAP 데이터 보기에 대한 쓰기 액세스 권한은 거부됩니다. 이 예에서는 한 개의 전역 ACI만 추가하면 사용자가 자신의 비밀번호를 수정할 수 있습니다.

1 프록시 관리자로 ACI 풀을 JDBC 데이터 소스에 추가하고 사용자가 자신의 항목을 수정할 수 있도록 전역 ACI를 추가합니다.

```
% ldapmodify -p 1389 -D "cn=proxy manager" -w password
dn: cn=mysql1,cn=virtual access controls
changetype: add
objectclass: acisource
dpsaci: (targetattr="*") (target = "ldap:///o=sql") \
 (version 3.0; acl "enable all access for all users "; allow(all) \
userdn="ldap:///uid=kvaughan,o=sql";)
cn: mysql1
```

2 연결 처리기를 만들어 o=sql 도메인에 대한 연결을 처리합니다.

```
% dpconf create-connection-handler mysql1-handler
```

3 연결 처리기를 활성화하고 o=sql 도메인에 있는 사용자의 모든 바인드를 처리하도록 구성합니다.

```
% dpconf set-connection-handler-prop mysql1-handler is-enabled:true \
 bind-dn-filters:"uid=.*,o=sql"
```

4 이전에 추가한 ACI 풀을 사용하도록 연결 처리기를 구성합니다.

```
% dpconf set-connection-handler-prop mysql1-handler aci-source:mysql1
```

▼ JDBC 데이터 보기를 테스트하는 방법

1 o=sql의 사용자로 JDBC 데이터 소스를 검색하여 데이터 보기에서 읽기 가능한지 확인합니다.

```
% ldapsearch -p 1389 -D "uid=kvaughan,o=sql" -w mypwd -b o=sql "objectclass=*"
```

주 – o=sql 또는 익명 바인드의 사용자에 대한 자격 증명을 사용해야 합니다.

---

2 o=sql의 사용자로 userPassword 속성을 수정하여 데이터 보기에 쓰기 가능한지 확인합니다.

```
% ldapmodify -p 1389 -D "uid=kvaughan,o=sql" -w mypwd
dn: uid=kvaughan,o=sql
changetype: modify
replace: userPassword
userPassword: myNewpwd
```

결합 데이터 보기 만들기 및 테스트

▼ 결합 데이터 보기를 만드는 방법

1 **myjoin1–view라는 결합 데이터 보기를 만듭니다.**

LDAP 데이터 보기를 기본 데이터 보기로 지정하고 JDBC 데이터 보기를 보조 데이터 보기로 지정합니다.

```
% dpconf create-join-data-view myjoin1-view myds1-view mysql1-view o=join
```

2 **보조 데이터 보기에서 결합 규칙을 정의합니다.**

다음 결합 규칙은 보조 데이터 보기에 있는 항목의 `uid` 속성이 기본 데이터 보기에 있는 항목의 `uid` 속성과 일치해야 함을 지정합니다.

```
% dpconf set-jdbc-data-view-prop mysql1-view filter-join-rule:uid='${myds1-view.uid}'
```

3 **결합 데이터 보기에 필터 결합 규칙이 설정되어 있는 경우 결합 데이터 보기에 항목을 추가하려면 보조 데이터 보기에 가상 변환 규칙을 설정해야 합니다.**

```
dpconf add-virtual-transformation secondary-view-name \
write add-attr-value dn uid=\${uid}
```

주 – 이 규칙을 설정하지 않으면 결합 데이터 보기에 항목을 추가할 수 없습니다.

---

4 **결합 데이터 보기를 통해 기본 데이터 보기에서 읽고 쓸 수 있는 속성 집합을 정의합니다.**

```
% dpconf set-ldap-data-view-prop myds1-view viewable-attr:dn viewable-attr:cn \
 viewable-attr:sn viewable-attr:givenName viewable-attr:objectClass viewable-attr:ou \
 viewable-attr:l viewable-attr:uid viewable-attr:mail viewable-attr:telephoneNumber \
 viewable-attr:facsimileTelephoneNumber viewable-attr:roomNumber viewable-attr:userPassword
% dpconf set-ldap-data-view-prop myds1-view writable-attr:dn writable-attr:cn \
 writable-attr:sn writable-attr:givenName writable-attr:objectClass writable-attr:ou \
 writable-attr:l writable-attr:uid writable-attr:mail writable-attr:telephoneNumber \
 writable-attr:facsimileTelephoneNumber writable-attr:roomNumber writable-attr:userPassword
```

이러한 정의는 **결합** 보기의 컨텍스트에서만 적용됩니다. 기본적으로 LDAP 데이터 보기에 직접 액세스할 경우 모든 속성을 읽고 쓸 수 있습니다.

5 **결합 데이터 보기를 통해 보조 데이터 보기에서 읽고 쓸 수 있는 속성 집합을 정의합니다.**

```
% dpconf set-jdbc-data-view-prop mysql1-view viewable-attr:dn viewable-attr:objectclass \
 viewable-attr:sn viewable-attr:room viewable-attr:userpassword viewable-attr:jobtitle \
 viewable-attr:country viewable-attr:tel
% dpconf set-jdbc-data-view-prop mysql1-view writable-attr:dn writable-attr:objectclass \
 writable-attr:sn writable-attr:room writable-attr:userpassword writable-attr:jobtitle \
 writable-attr:country writable-attr:tel
```

이러한 정의는 **결합** 보기의 컨텍스트에서만 적용됩니다. 기본적으로 JDBC 데이터 보기에 직접 액세스할 경우 모든 속성을 읽고 쓸 수 있습니다.

▼ 필요한 ACI를 만드는 방법

1 프록시 관리자로 결합 데이터 보기에 대한 익명 액세스를 허용하는 전역 ACI를 추가합니다.

```
% ldapmodify -p 1389 -D "cn=proxy manager" -w password
dn: cn=myjoin1,cn=virtual access controls
changetype: add
objectclass: acisource
dpsaci: (targetattr="*") (target = "ldap:///o=join") \
 (version 3.0; acl "anonymous_access"; allow(all) userdn="ldap:///anyone";)
cn: myjoin1
```

2 연결 처리기를 만들어 o=join 도메인에 대한 연결을 처리합니다.

```
% dpconf create-connection-handler myjoin1-handler
```

3 연결 처리기를 활성화하고 o=join에 있는 사용자의 모든 바인드를 처리하도록 구성합니다.

```
% dpconf set-connection-handler-prop myjoin1-handler is-enabled:true \
 bind-dn-filters:"uid=.*,ou=people,o=join"
```

4 이전에 추가한 ACI 풀을 사용하도록 연결 처리기를 구성합니다.

```
% dpconf set-connection-handler-prop myjoin1-handler aci-source:myjoin1
```

▼ 결합 데이터 보기를 테스트하는 방법

1 익명 사용자로 결합 데이터 보기를 검색합니다.

이 단계에서는 Kirsten Vaughan의 항목을 검색하여 두 결합 보기의 데이터가 검색되는지 여부를 확인합니다.

```
% ldapsearch -p 1389 -b o=join "uid=kvaughan"
```

반환된 항목에 LDAP 데이터 보기 및 JDBC 데이터 보기 속성이 모두 포함되어 있는지 확인합니다.

2 o=join의 사용자로 userPassword 속성을 수정하여 결합 데이터 보기에 쓰기 가능한지 확인합니다.

```
% ldapmodify -p 1389 -D "uid=kvaughan,ou=people,o=join" -w myNewPassword
dn: uid=kvaughan,ou=people,o=join
changetype: modify
replace: userPassword
userPassword: myPassword
```

별도의 여러 데이터 소스 결합

이 구성에서는 가상 디렉토리의 일부 기능이 특정 디렉토리 서비스 요구 사항을 충족하는 조직인 Example.com에 대해 설명합니다.

데이터 저장소 시나리오

Example.com은 별도의 여러 데이터 소스에 조직 데이터를 저장합니다. 레거시 이유로 인해 사용자 데이터는 LDAP 디렉토리, 플랫 LDIF 파일 및 SQL 데이터베이스에 분산됩니다. HR 부서는 `o=example.com`의 기본 DN을 사용하여 LDAP 디렉토리에 사용자 데이터를 저장합니다. 급여 부서는 데이터를 SQL 데이터베이스에 저장합니다. 관리 부서는 `dc=example,dc=com`의 기본 DN을 사용하여 부서 및 건물 번호와 같은 관리 데이터를 LDIF 파일에 저장합니다.

또한 Example.com은 Company22라는 회사를 인수했습니다. Company 22 역시 `dc=company22,dc=com`의 기본 DN을 사용하여 사용자 데이터를 LDAP 디렉토리에 저장합니다.

다음 다이어그램은 Example.com의 사용자 데이터가 저장되는 방법에 대한 상위 수준 보기를 제공합니다.

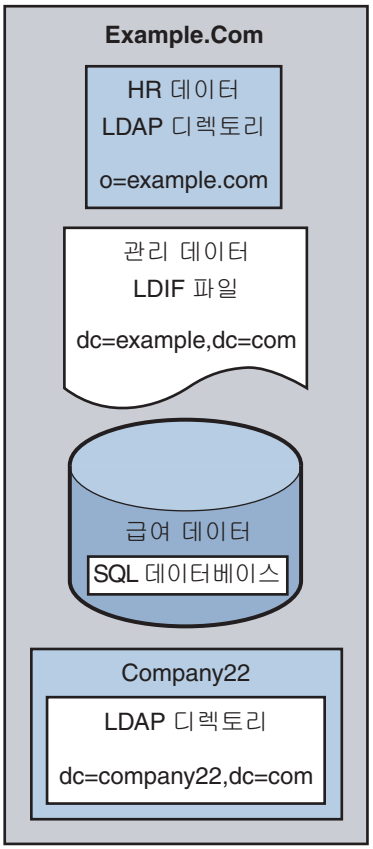


그림 23–2 별도의 소스에 있는 데이터 저장소

클라이언트 응용 프로그램 요구 사항

Example.com에는 별도의 데이터 소스에 저장된 데이터에 대한 액세스 권한을 필요로 하는 여러 LDAP 클라이언트 응용 프로그램이 있습니다. 클라이언트 응용 프로그램의 요구 사항은 각각 다릅니다. 다른 데이터 보기가 필요합니다. 클라이언트가 데이터를 집계하도록 요구하는 경우도 있습니다. 또한 일부 클라이언트 응용 프로그램에는 Example.com의 새 직원을 이전 직원과 함께 관리할 수 있도록 Company22의 사용자 데이터에 대한 액세스 권한이 필요합니다.

다음 다이어그램은 Example.com의 클라이언트 응용 프로그램 요구 사항에 대한 상위 수준 보기를 제공합니다.

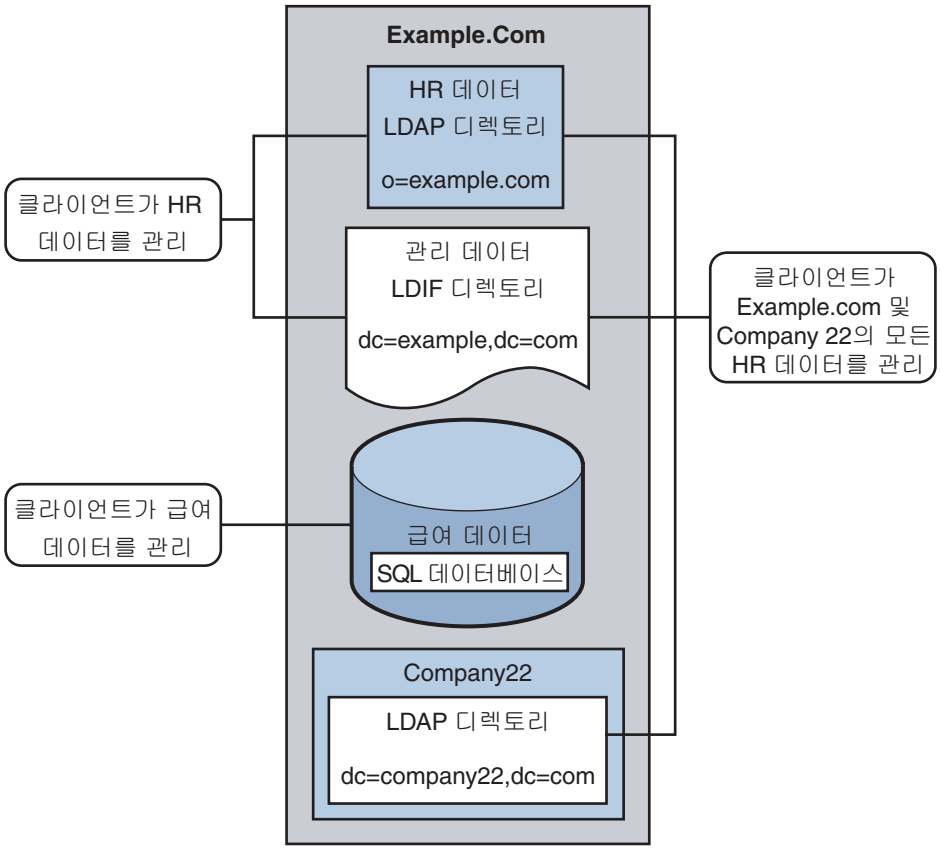


그림 23–3 클라이언트 응용 프로그램 요구 사항

다음 절에서는 이 샘플 시나리오에 설명된 클라이언트 응용 프로그램의 요구 사항을 충족할 수 있는 충분한 구성 디렉토리 프록시 서버 데이터 보기를 안내합니다. 데이터 보기가 작동하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 17 장, “Directory Proxy Server Distribution”과 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 18 장, “Directory Proxy Server Virtualization”을 참조하십시오.

샘플 시나리오의 구성은 다음 절로 구분됩니다.

- 422 페이지 “HR LDAP 디렉토리 및 관리 LDIF 파일의 데이터 집계”
- 424 페이지 “DN의 이름을 바꾸어 Company 22의 데이터를 Example.Com의 DIT에 추가”
- 426 페이지 “Company 22의 데이터를 HR 데이터에 추가”
- 427 페이지 “SQL 데이터베이스의 급여 데이터에 액세스할 수 있도록 LDAP 클라이언트 활성화”
- 430 페이지 “가상 액세스 제어 추가”

HR LDAP 디렉토리 및 관리 LDIF 파일의 데이터 집계

HR 부서는 직원 이름, 작업 시작 데이터 및 작업 수준과 같은 정보를 저장합니다. 관리 부서는 건물 코드 및 사무실 번호와 같은 추가 데이터를 저장합니다. HR 데이터를 처리하는 클라이언트 응용 프로그램에는 두 소스의 결합된 데이터에 대한 액세스 권한이 필요합니다. 두 데이터 소스에는 각 항목에 존재하는 공통 속성인 `employeeNumber`가 있습니다.

다음 다이어그램은 클라이언트 응용 프로그램의 요구 사항을 보여줍니다.

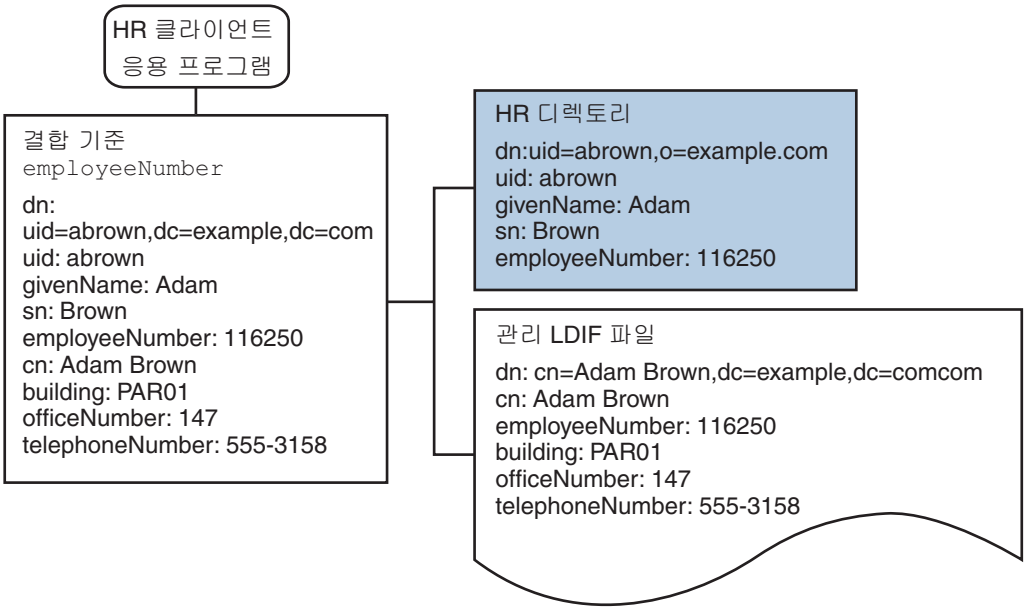


그림 23–4 LDAP 디렉토리 및 LDIF 파일의 데이터 집계

이 응용 프로그램의 요구 사항을 충족하기 위해 급여 디렉토리 및 관리 LDIF 파일에 대한 데이터 보기가 만들어집니다. 그런 다음 이러한 두 데이터 보기가 결합되어 집계된 데이터에 대한 액세스를 제공합니다. 이 공통 속성을 사용하여 디렉토리 프록시 서버는 각 사용자에 대한 데이터를 집계할 수 있습니다.

단순하게 만들기 위해 이 절에 사용된 명령은 다음 정보를 가정합니다.

- 디렉토리 프록시 서버 인스턴스는 기본 LDAP 포트(389)를 사용하여 로컬 호스트에서 실행됩니다.
- 디렉토리 프록시 서버 인스턴스는 `/local/myDPS`에 있습니다.
- 프록시 관리자 비밀번호를 포함하는 파일에 대한 경로가 `LDAP_ADMIN_PWF` 변수로 설정되었습니다. 디렉토리 프록시 서버 환경 변수를 설정하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide**의 “Environment Variables”를 참조하십시오.

- 급여 LDAP 디렉토리는 포트 2389의 payrollHost라는 호스트에서 실행됩니다.
- 관리 데이터를 저장하는 데 사용되는 LDIF 파일의 이름이 example.ldif로 지정됩니다.

각 명령의 완전한 구문을 얻으려면 명령을 옵션 없이 실행합니다. 예를 들면 다음과 같습니다.

```
$ dpconf create-ldap-data-view
Operands are missing
Usage: dpcfg create-ldap-data-view VIEW_NAME POOL_NAME SUFFIX_DN
```

▼ 급여 디렉토리에 대한 LDAP 데이터 보기 만들기 및 활성화

1 급여 디렉토리에 대한 LDAP 데이터 소스를 만듭니다.

```
$ dpconf create-ldap-data-source payroll-directory payrollHost:2389
```

2 급여 디렉토리에 대한 LDAP 데이터 소스 풀을 만듭니다.

```
$ dpconf create-ldap-data-source-pool payroll-pool
```

3 급여 데이터 소스를 데이터 소스 풀에 첨부합니다.

```
$ dpconf attach-ldap-data-source payroll-pool payroll-directory
```

4 첨부된 데이터 소스의 가중치를 구성합니다.

```
$ dpconf set-attached-ldap-data-source-prop -h payrollHost -p 2389 \
payroll-pool payroll-directory add-weight:2 \
bind-weight:2 compare-weight:2 delete-weight:2 \
modify-dn-weight:2 modify-weight:2 search-weight:2
```

5 급여 디렉토리에 대한 LDAP 데이터 보기를 만듭니다.

```
$ dpconf create-ldap-data-view payroll-view payroll-pool o=example.com
```

6 클라이언트 요청을 이 데이터 보기로 전달할 수 있도록 LDAP 데이터 보기를 활성화합니다.

```
$ dpconf set-ldap-data-view-prop payroll-view is-enabled:true
```

7 디렉토리 프록시 서버를 다시 시작하여 변경 사항을 적용합니다.

```
$ dpadm restart /local/myDPS
```

▼ 관리 데이터에 대한 LDIF 데이터 보기 만들기 및 활성화

1 관리 데이터에 대한 LDIF 데이터 보기를 만듭니다.

```
$ dpconf create-ldif-data-view admin-view example.ldif dc=example,dc=com
```

2 **관리 데이터에 대한 LDIF 데이터 보기를 활성화합니다.**

```
$ dpconf set-ldif-data-view-prop admin-view is-enabled:true
```

3 **관리 보기가 급여 보기에서 둘 이상의 항목에 사용되는 항목을 포함하도록 지정합니다.**

```
$ dpconf set-ldif-data-view-prop admin-view contains-shared-entries:true
```

이 등록 정보를 TRUE로 설정할 경우 급여 데이터 보기에서 항목을 삭제하면 관리 데이터 보기의 공유 항목이 삭제되지 않습니다. 급여 데이터 보기에 항목을 추가하면 항목이 보조 데이터 보기에만 추가됩니다(아직 없는 경우).

4 **디렉토리 프록시 서버를 다시 시작하여 변경 사항을 적용합니다.**

```
$ dpadm restart /local/myDPS
```

▼ 급여 데이터 보기 및 관리 데이터 보기 결합

1 **관리 데이터 보기에서 데이터가 집계되는 방법을 지정하는 필터 결합 규칙을 만듭니다.**

다음 결합 규칙은 사용자 항목의 `employeeNumber` 속성을 기준으로 데이터를 결합해야 함을 지정합니다.

```
$ dpconf set-ldif-data-view-prop admin-view \
filter-join-rule:'employeeNumber=\${payroll-view.employeeNumber}'
```

2 **두 데이터 보기를 집계하는 결합 데이터 보기를 만듭니다.**

결합 데이터 보기의 경우 조직에서는 접미어 DN `dc=example,dc=com`을 사용합니다.

```
$ dpconf create-join-data-view example-join-view payroll-view admin-view \
dc=example,dc=com
```

DN의 이름을 바꾸어 Company 22의 데이터를 Example.Com의 DIT에 추가

Company 22의 사용자 데이터는 DN `dc=company22,dc=com`에 저장됩니다. Example.com에서는 대부분 이 사용자 데이터를 별도로 유지하려고 하지만 한 클라이언트 응용 프로그램은 나머지 Example.com 직원과 함께 Company 22 직원을 관리해야 합니다. 이 클라이언트 응용 프로그램의 경우 Company 22 사용자 데이터의 모양은 Example.com 데이터와 **같아야** 합니다.

다음 다이어그램은 클라이언트 응용 프로그램의 요구 사항을 보여줍니다.

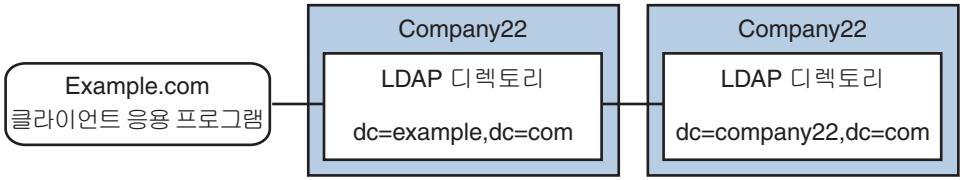


그림 23–5 DN 이름 바꾸기

이 응용 프로그램의 요구 사항을 충족하기 위해 `dc=example,dc=com`의 가상 DN을 사용하여 데이터 보기가 Company 22의 디렉토리에 대해 만들어집니다.

단순하게 만들기 위해 이 절에 사용된 명령은 다음 정보를 가정합니다.

- 디렉토리 프록시 서버 인스턴스는 기본 LDAP 포트(389)를 사용하여 로컬 호스트에서 실행됩니다.
- 디렉토리 프록시 서버 인스턴스는 `/local/myDPS`에 있습니다.
- 프록시 관리자 비밀번호를 포함하는 파일에 대한 경로가 `LDAP_ADMIN_PWF` 변수로 설정되었습니다. 디렉토리 프록시 서버 환경 변수를 설정하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide**의 "Environment Variables"를 참조하십시오.
- Company 22 LDAP 디렉토리는 포트 2389의 이름이 `company22Host`인 호스트에서 실행됩니다.

▼ 가상 DN을 사용하여 Company 22의 디렉토리에 대한 데이터 보기 만들기

1 Company 22의 디렉토리에 대한 LDAP 데이터 소스를 만듭니다.

```
$ dpconf create-ldap-data-source company22-directory company22Host:2389
```

2 Company 22의 디렉토리에 대한 LDAP 데이터 소스 풀을 만듭니다.

```
$ dpconf create-ldap-data-source-pool company22-pool
```

3 Company 22의 데이터 소스를 데이터 소스 풀에 첨부합니다.

```
$ dpconf attach-ldap-data-source company22-pool company22-directory
```

4 첨부된 데이터 소스의 가중치를 구성합니다.

```
$ dpconf set-attached-ldap-data-source-prop -h company22Host -p 2389 \
company22-pool company22-directory add-weight:2 \
bind-weight:2 compare-weight:2 delete-weight:2 \
modify-dn-weight:2 modify-weight:2 search-weight:2
```

5 **dc=example,dc=com의 가상 DN을 사용하여 Company 22의 디렉토리에 대한 LDAP 데이터 보기를 만듭니다.**

```
$ dpconf create-ldap-data-view company22-view company22-pool dc=example,dc=com
```

6 **이 가상 DN을 Company 22의 디렉토리에 있는 실제 DN에 매핑하도록 디렉토리 프록시 서버에 지시합니다.**

```
$ dpconf set-ldap-data-view-prop company22-view \
dn-mapping-source-base-dn:dc=company22,dc=com
```

7 **클라이언트 요청을 이 데이터 보기에 전달할 수 있도록 Company 22의 디렉토리에 대한 LDAP 데이터 보기를 활성화합니다.**

```
$ dpconf set-ldap-data-view-prop company22-view is-enabled:true
```

8 **디렉토리 프록시 서버를 다시 시작하여 변경 사항을 적용합니다.**

```
$ dpadm restart /local/myDPS
```

Company 22의 데이터를 HR 데이터에 추가

HR 부서에서는 Example.com과 새로 인수한 Company 22에 대한 HR 데이터의 집계된 보기가 필요합니다. 다음 다이어그램은 전역 HR 응용 프로그램의 요구 사항을 보여줍니다.

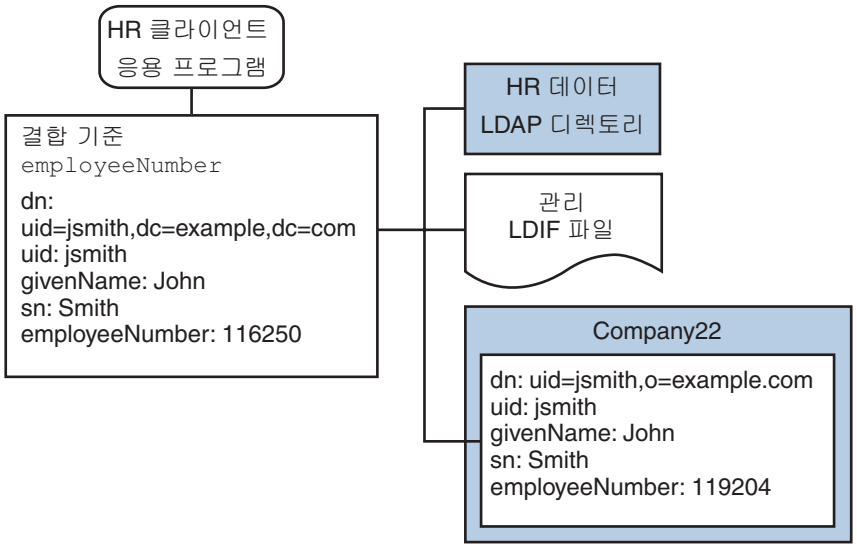


그림 23–6 결합 데이터 보기 및 LDAP 데이터 보기의 데이터 집계

▼ 결합 데이터 보기 및 Company 22 데이터 보기 예 결합

1 Company 22 데이터 보기에서 데이터가 집계되는 방법을 지정하는 필터 결합 규칙을 만듭니다.

다음 결합 규칙은 사용자 항목의 `employeeNumber` 속성을 기준으로 데이터를 결합해야 함을 지정합니다.

```
$ dpconf set-ldif-data-view-prop company22-view \
filter-join-rule:'employeeNumber=\${example-join-view.employeeNumber}'
```

2 Company 22의 데이터 보기와 Example.com의 결합 데이터 보기를 집계하는 결합 데이터 보기를 만듭니다.

```
$ dpconf create-join-data-view global-join-view example-join-view \
company22-view dc=example,dc=com
```

SQL 데이터베이스의 급여 데이터에 액세스할 수 있도록 LDAP 클라이언트 활성화

Example.com의 급여 부서는 급여 데이터를 SQL 데이터베이스에 저장합니다. 이 데이터베이스에는 두 개의 테이블(`employee` 및 `salary`)이 있습니다. Example.com에는 해당 데이터에 대한 액세스 권한을 필요로 하는 LDAP 클라이언트 응용 프로그램이 있습니다. 이 클라이언트 응용 프로그램의 경우 SQL 데이터의 모양이 LDAP 데이터와 **같아야** 합니다.

다음 다이어그램은 클라이언트 응용 프로그램의 요구 사항을 보여줍니다.

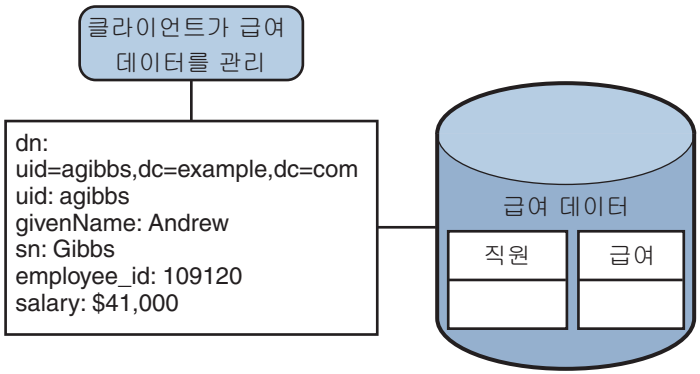


그림 23–7 SQL 데이터베이스에 대한 액세스를 제공하는 JDBC 데이터 보기

이 응용 프로그램의 요구 사항을 충족하기 위해 SQL 테이블의 열을 LDAP 속성에 매핑하는 JDBC 데이터 보기가 만들어집니다.

단순하게 만들기 위해 이 절에 사용된 명령은 다음 정보를 가정합니다.

- 디렉토리 프록시 서버 인스턴스는 기본 LDAP 포트(389)를 사용하여 로컬 호스트에서 실행됩니다.
- 디렉토리 프록시 서버 인스턴스는 `/local/myDPS`에 있습니다.
- 프록시 관리자 비밀번호를 포함하는 파일에 대한 경로가 `LDAP_ADMIN_PWF` 변수로 설정되었습니다. 디렉토리 프록시 서버 환경 변수를 설정하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Installation Guide**의 "Environment Variables"를 참조하십시오.
- SQL 데이터베이스가 가동되어 실행 중입니다.
- `JAVA_HOME` 변수가 올바른 Java 경로로 설정되었습니다.
- SQL 데이터베이스의 비밀번호는 `myPasswordFile` 파일에 저장된 `myPassword`입니다.

▼ Example.com의 급여 데이터베이스에 대한 JDBC 데이터 보기 만들기

1 급여 데이터베이스에 대한 JDBC 데이터 소스를 만듭니다.

```
$ dpconf create-jdbc-data-source -b payrollsqldb \
  -B jdbc:payrollsqldb:payrollsql://localhost/ \
  -J file://payrollsqldb.jar \
  -S org.payrollsqldb.jdbcDriver payroll-src
```

2 SQL 데이터베이스의 등록 정보를 사용하여 JDBC 데이터 소스를 구성합니다.

```
$ dpconf set-jdbc-data-source-prop payroll-src \
  db-user:proxy
  db-pwd-file:password-file-location/myPasswordFile
```

3 JDBC 데이터 소스를 활성화합니다.

```
$ dpconf set-jdbc-data-source-prop payroll-src is-enabled:true
```

4 급여 데이터베이스에 대한 JDBC 데이터 소스 풀을 만듭니다.

```
$ dpconf create-jdbc-data-source-pool payroll-pool
```

5 급여 데이터 소스를 데이터 소스 풀에 첨부합니다.

```
$ dpconf attach-jdbc-data-source payroll-pool payroll-src
```

6 `o=payroll`의 가상 DN을 사용하여 급여 데이터베이스에 대한 JDBC 데이터 보기를 만듭니다.

```
$ dpconf create-jdbc-data-view payroll-view payroll-pool o=payroll
```

7 SQL 데이터베이스의 각 테이블에 대한 JDBC 테이블을 만듭니다.

```
$ dpconf create-jdbc-table jdbc-employee employee
$ dpconf create-jdbc-table jdbc-salary salary
```

8 SQL 테이블의 각 열에 대한 JDBC 속성을 추가합니다.

```
$ dpconf add-jdbc-attr jdbc-employee eid employee_id
$ dpconf add-jdbc-attr jdbc-employee first firstname
$ dpconf add-jdbc-attr jdbc-employee last lastname
$ dpconf add-jdbc-attr jdbc-employee description description
$ dpconf add-jdbc-attr jdbc-employee spouse spousename
$ dpconf add-jdbc-attr jdbc-salary salary salary
$ dpconf add-jdbc-attr jdbc-salary social ssn
```

9 JDBC 데이터 보기를 통해 보기 가능한 속성과 쓰기 가능한 속성을 지정합니다.

```
$ dpconf set-jdbc-data-view-prop payroll-view \
 viewable-attr:eid \
 viewable-attr:first \
 viewable-attr:last \
 viewable-attr:desc \
 viewable-attr:spouse \
 viewable-attr:salary \
 viewable-attr:social
 $ dpconf set-jdbc-data-view-prop payroll-view \
 writable-attr:eid \
 writable-attr:first \
 writable-attr:last \
 writable-attr:description \
 writable-attr:spouse \
 writable-attr:salary \
writable-attr:social
```

10 LDAP 객체 클래스에 매핑되는 JDBC 객체 클래스를 만듭니다.

다음 명령은 LDAP `person` 객체 클래스에 매핑되는 객체 클래스를 만듭니다. 객체 클래스는 직원 테이블을 기본 테이블로 사용하고 급여 테이블을 보조 테이블로 사용해야 함을 지정합니다. `eid` 속성을 사용하여 DN을 구성해야 합니다.

```
$ dpcfg create-jdbc-object-class payroll-view \
 person jdbc-employee jdbc-salary eid
```

11 보조 테이블의 데이터가 기본 테이블의 데이터에 연결되는 방법을 지정하는 필터 결합 규칙을 보조 테이블에서 만듭니다.

다음 결합 규칙은 `employee_id` 속성을 기준으로 데이터를 결합해야 함을 지정합니다.

```
$ dpconf set-jdbc-table-prop jdbc-salary \
filter-join-rule:'employee_id=\${employee.employee_id}'
```

12 JDBC 객체 클래스에서 수퍼 클래스를 만듭니다.

```
$ set-jdbc-object-class-prop payroll-view person super-class:extensibleObject
```

가상 액세스 제어 추가

LDAP 디렉토리에 대한 액세스 제어는 디렉토리 자체에 ACI를 정의하여 처리됩니다. 가상 데이터 보기를 통해 데이터 소스에 액세스할 경우 이러한 데이터 보기를 통해 표시되는 데이터에만 적용하도록 ACI를 정의해야 합니다.

디렉토리 프록시 서버를 통한 모든 액세스는 **연결 처리기**에서 제어됩니다. 연결 처리기에 대한 자세한 내용은 25 장을 참조하십시오.

▼ 익명 액세스를 허용하는 ACI 추가

1 ACI를 추가합니다.

```
$ ldapadd -v -D "cn=proxy manager" -w password -p 389
dn: cn=ldifonly-acis,cn=virtual access controls
objectclass: top
objectclass: aciSource
cn: ldifonly-acis
dpsaci: (targetattr="*")(version 3.0; acl "anonymous_access"; allow(all) \
(userdn="ldap:///anyone");)
```

2 연결 처리기에서 가상 ACI를 가리킵니다.

```
$ dpconf set-connection-handler-prop anonymous aci-source:ldifonly-acis
```

3 연결 처리기를 활성화합니다.

```
$ dpconf set-connection-handler-prop anonymous is-enabled:true
```

◆ ◆ ◆ 24장

24 디렉토리 프록시 서버와 백엔드 LDAP 서버 간의 연결

이 장에서는 디렉토리 프록시 서버와 백엔드 LDAP 서버 간에 연결을 구성하는 방법에 대해 설명합니다. 이 장은 다음 내용으로 구성되어 있습니다.

디렉토리 프록시 서버와 백엔드 LDAP 서버 간의 연결 구성

LDAP 데이터 소스를 만든 경우 LDAP 데이터 소스에 열린 기본 연결 수는 6개입니다(읽기, 바인드 및 쓰기 작업에 각각 2개씩). 기본 연결을 확인하려면 다음 명령을 입력합니다.

```
dpconf get-ldap-data-source-prop src-name num-read-init num-write-init num-bind-init
num-bind-init   :  2
num-read-init   :  2
num-write-init  :  2
```

연결 수는 트래픽이 증가하면 자동으로 늘어납니다.

디렉토리 프록시 서버와 백엔드 LDAP 서버 간에 연결을 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 디렉토리 프록시 서버와 백엔드 LDAP 서버 간에 연결 수를 구성하는 방법

주 – 이 절차에서는 바인드 작업을 위한 연결 수를 구성합니다. 읽기 또는 쓰기 작업을 위한 연결 수를 구성하려면 동일한 절차를 수행하되, `bind`를 `read` 또는 `write`로 바꿉니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **바인드 작업을 위해 디렉토리 프록시 서버와 백엔드 LDAP 서버 간에 초기 연결 수를 구성합니다.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    num-bind-init:new-value
   ```

2. **바인드 작업을 위한 연결 증분을 구성합니다.**

   증분은 현재 연결 수보다 많은 연결이 요청될 때마다 추가되는 연결 수입니다.

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    num-bind-incr:new-value
   ```

3. **바인드 작업을 위한 최대 연결 수를 구성합니다.**

   이 최대 연결 수에 도달하면 더 이상 연결이 추가되지 않습니다.

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
    num-bind-limit:new-value
   ```

▼ 연결 시간 초과를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **디렉토리 프록시 서버에서 데이터 소스에 연결을 시도할 수 있는 최대 시간을 구성합니다.**

  ```
  $ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
   connect-timeout:new-value
  ```

  예를 들어 연결 시간 초과를 10밀리초로 구성합니다.

  ```
  $ dpconf set-ldap-data-source-prop -h host1 -p 1389 data-source-name connect-timeout:10
  ```

▼ 연결 풀 대기 시간 초과를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **연결 풀에 설정된 연결을 사용할 수 있을 때까지 디렉토리 프록시 서버가 대기할 수 있는 최대 시간을 구성합니다.**

```
$ dpconf set-server-prop -h host -p port data-source-name \
 connection-pool-wait-timeout:value
```

예를 들어 대기 시간 초과를 20초로 구성합니다.

```
$ dpconf set-ldap-data-source-prop -h host1 -p 1389 data-source-name \
 connection-pool-wait-timeout:20000
```

디렉토리 프록시 서버와 백엔드 LDAP 서버 간에 SSL 구성

다음 절차에서는 디렉토리 프록시 서버와 백엔드 LDAP 서버 간에 SSL을 구성하는 방법에 대해 설명합니다.

▼ 디렉토리 프록시 서버와 백엔드 LDAP 서버 간에 SSL을 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **디렉토리 프록시 서버와 백엔드 LDAP 서버 간에 보안 포트를 구성합니다.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 ldaps-port:port-number
```

2. **SSL이 디렉토리 프록시 서버와 백엔드 LDAP 서버 간의 연결에 사용되는 시점을 구성합니다.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name ssl-policy:value
```

   - *value*가 `always`인 경우 SSL이 항상 연결에 사용됩니다.
   - *value*가 `client`인 경우 클라이언트가 SSL을 사용하는 중이면 SSL이 사용됩니다.

   연결에서 SSL을 사용하고 있지 않으면 `startTLS` 명령을 사용하여 연결 수준을 SSL로 올릴 수 있습니다.

3. **434 페이지 "디렉토리 프록시 서버에 대한 SSL 암호 및 SSL 프로토콜 선택"에 설명된 것처럼 SSL에 대한 프로토콜 및 암호를 선택합니다.**

4 **백엔드 LDAP 서버의 SSL 서버 인증서를 확인하도록 디렉토리 프록시 서버를 구성합니다.**
자세한 내용은 352 페이지 "백엔드 디렉토리 서버의 인증서를 디렉토리 프록시 서버의 인증서 데이터베이스에 추가하는 방법"을 참조하십시오.

5 **백엔드 LDAP 서버가 디렉토리 프록시 서버에서 인증서를 요청할 경우 SSL 클라이언트 인증서를 보내도록 디렉토리 프록시 서버를 구성합니다.**
자세한 내용은 353 페이지 "인증서를 백엔드 LDAP 서버로 내보내기"를 참조하십시오.

6 **디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.**
디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

디렉토리 프록시 서버에 대한 SSL 암호 및 SSL 프로토콜 선택

디렉토리 프록시 서버에 사용될 수 있는 암호 및 프로토콜은 사용 중인 JVM™(Java™ Virtual Machine)에 따라 다릅니다. 기본적으로 디렉토리 프록시 서버는 JVM 시스템에 대해 활성화되는 기본 암호 및 프로토콜을 사용합니다.

▼ 암호 및 프로토콜 목록을 선택하는 방법

이 절차를 사용하여 지원되는 암호 및 프로토콜과 활성화된 암호 및 프로토콜을 검색합니다. 암호 또는 프로토콜이 지원되는 경우 이를 활성화하거나 비활성화할 수 있습니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **지원되는 암호 및 프로토콜 목록을 봅니다.**

```
$ dpconf get-server-prop -h host -p port supported-ssl-cipher-suites \
 supported-ssl-protocols
```

2 **활성화된 암호 및 프로토콜 목록을 봅니다.**

```
$ dpconf get-server-prop -h host -p port enabled-ssl-cipher-suites \
 enabled-ssl-protocols
```

3 **지원되는 암호 또는 프로토콜을 하나 이상 활성화합니다.**

a. **지원되는 암호를 하나 이상 활성화합니다.**

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-suites:supported-ssl-cipher-suite \
 [enabled-ssl-cipher-suites:supported-ssl-cipher-suite ...]
```

지원되는 기존 암호 목록에 암호를 추가하려면 이 명령을 사용합니다.

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-suites+:supported-ssl-cipher-suite
```

b. 지원되는 프로토콜을 하나 이상 활성화합니다.

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-protocols:supported-ssl-cipher-protocol \
 [enabled-ssl-cipher-protocols:supported-ssl-cipher-protocol ...]
```

지원되는 기존 프로토콜 목록에 프로토콜을 추가하려면 이 명령을 사용합니다.

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-protocols+:supported-ssl-cipher-protocol
```

4 (옵션) 지원되는 암호 또는 프로토콜을 비활성화합니다.

```
$ dpconf set-server-prop -h host -p port \
 enabled-ssl-cipher-protocols-:supported-ssl-cipher-protocol
```

백엔드 LDAP 서버에 요청 전달

이 절에는 디렉토리 프록시 서버에서 백엔드 LDAP 서버에 요청을 전달하는 데 사용할 수 있는 다양한 방법에 대한 정보가 들어 있습니다.

바인드 재생을 사용하여 요청 전달

디렉토리 프록시 서버의 클라이언트 자격 증명을 위한 바인드 재생에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Directory Proxy Server Configured for BIND Replay"를 참조하십시오. 다음 절차에서는 바인드 재생을 사용하여 디렉토리 프록시 서버에서 백엔드 LDAP 서버에 요청을 전달하는 방법에 대해 설명합니다.

▼ 바인드 재생을 사용하여 요청을 전달하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **클라이언트에서 제공된 자격 증명을 사용하여 백엔드 LDAP 서버에 대해 인증하도록 데이터 소스 클라이언트 자격 증명을 구성합니다.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-client-identity
```

프록시 인증을 사용하여 요청 전달

디렉토리 프록시 서버의 프록시 인증에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Directory Proxy Server Configured for Proxy Authorization"을 참조하십시오.

이 절은 프록시 인증 및 프록시 인증 제어를 사용하여 요청을 전달하는 절차로 구성되어 있습니다.

▼ 프록시 인증을 사용하여 요청을 전달하는 방법

1 버전 1 또는 버전 2의 프록시 인증 제어가 필요하도록 데이터 소스를 구성합니다.

예를 들어 버전 1의 프록시 인증 제어가 필요하도록 데이터 소스를 구성합니다.

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 proxied-auth-use-v1:true
```

또는 버전 2의 프록시 인증 제어가 필요하도록 데이터 소스를 구성합니다.

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 proxied-auth-use-v1:false
```

2 프록시 인증을 사용하여 백엔드 LDAP 서버에 대해 인증하도록 데이터 소스를 구성합니다.

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-proxy-auth
```

쓰기 작업에만 프록시 인증을 사용하여 백엔드 LDAP 서버에 대해 인증하도록 데이터 소스를 구성하려면 다음 명령을 실행합니다.

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-proxy-auth-for-write
```

쓰기 작업만 프록시 인증 제어를 사용하여 수행할 경우 읽기 요청에서는 클라이언트 아이디가 LDAP 서버에 전달되지 않습니다. 클라이언트 아이디 없이 요청을 전달하는 방법에 대한 자세한 내용은 437 페이지 "클라이언트 아이디 없이 요청 전달"을 참조하십시오.

3 디렉토리 프록시 서버의 바인드 자격 증명을 사용하여 데이터 소스를 구성합니다.

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 bind-dn:DPS-bind-dn bind-pwd-file:filename
```

4 시간 초과를 사용하여 데이터 소스를 구성합니다.

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 proxied-auth-check-timeout:value
```

디렉토리 프록시 서버는 `getEffectiveRights` 명령을 사용하여 프록시 인증에 대한 적절한 ACI가 클라이언트 DN에 있는지 확인합니다. 결과는 디렉토리 프록시 서버에 캐시되고 `proxied-auth-check-timeout` 만료 시에 갱신됩니다.

5 **필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.**

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

▼ 요청에 프록시 인증 제어가 포함된 경우 프록시 인증을 사용하여 요청을 전달하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **버전 1이나 2 또는 두 버전 모두 프록시 인증 제어를 허용하도록 디렉토리 프록시 서버를 구성합니다.**

```
$ dpconf set-server-prop -h host -p port allowed-ldap-controls:proxy-auth-v1 \
 allowed-ldap-controls:proxy-auth-v2
```

클라이언트 아이디 없이 요청 전달

다음 절차에서는 클라이언트 아이디를 전달하지 않고 디렉토리 프록시 서버에서 백엔드 LDAP 서버에 요청을 전달하는 방법에 대해 설명합니다.

▼ 클라이언트 아이디 없이 요청을 전달하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **디렉토리 프록시 서버의 자격 증명을 사용하여 백엔드 LDAP 서버에 대해 인증하도록 데이터 소스를 구성합니다.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 client-cred-mode:use-specific-identity
```

2 **디렉토리 프록시 서버의 바인드 자격 증명을 사용하여 데이터 소스를 구성합니다.**

```
$ dpconf set-ldap-data-source-prop -h host -p port data-source-name \
 bind-dn:bind-dn-of-DPS bind-pwd-file:filename
```

3 **필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.**

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

대체 사용자로 요청 전달

이 절은 요청을 대체 사용자로 전달하는 방법에 대한 정보로 구성되어 있습니다.

▼ 원격 사용자 매핑을 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 대체 사용자로 전달할 작업을 활성화합니다.

```
$ dpconf set-server-prop -h host -p port enable-user-mapping:true
```

2 원격 매핑을 위한 아이디를 포함하는 속성의 이름을 지정합니다.

```
$ dpconf set-server-prop -h host -p port \
 remote-user-mapping-bind-dn-attr:attribute-name
```

3 클라이언트 아이디를 원격으로 매핑하도록 디렉토리 프록시 서버를 활성화합니다.

```
$ dpconf set-server-prop -h host -p port enable-remote-user-mapping:true
```

4 기본 매핑을 구성합니다.

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-default-bind-dn:default-mapping-bind-dn \
 user-mapping-default-bind-pwd-file:filename
```

매핑된 아이디를 원격 LDAP 서버에서 찾을 수 없는 경우 클라이언트 아이디는 기본 아이디에 매핑됩니다.

5 원격 LDAP 서버의 클라이언트에 대한 항목에서 사용자 매핑을 구성합니다.

디렉토리 서버에서 사용자 매핑을 구성하는 방법에 대한 자세한 내용은 156 페이지 "프록시 인증"을 참조하십시오.

▼ 로컬 사용자 매핑을 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 대체 사용자로 전달할 작업을 활성화합니다.

```
$ dpconf set-server-prop -h host -p port enable-user-mapping:true
```

2 디렉토리 프록시 서버가 클라이언트 아이디를 원격으로 매핑하도록 구성되지 않았는지 확인합니다.

```
$ dpconf set-server-prop -h host -p port enable-remote-user-mapping:false
```

3 **기본 매핑을 구성합니다.**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-default-bind-dn:default-mapping-bind-dn \
 user-mapping-default-bind-pwd-file:filename
```

원격 LDAP 서버의 매핑이 실패할 경우 클라이언트 아이디는 이 DN에 매핑됩니다.

4 **인증되지 않은 사용자가 작업을 수행하도록 허용할 경우 인증되지 않은 클라이언트에 대한 매핑을 구성합니다.**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-anonymous-bind-dn:anonymous-mapping-bind-dn \
 user-mapping-anonymous-bind-pwd-file:filename
```

인증되지 않은 사용자가 작업을 수행하도록 허용하는 방법에 대한 자세한 내용은 455 페이지 "익명 액세스를 구성하는 방법"을 참조하십시오.

5 **클라이언트의 아이디를 구성합니다.**

```
$ dpconf set-user-mapping-prop -h host -p port \
 user-bind-dn:client-bind-dn user-bind-pwd-file:filename
```

6 **대체 사용자의 아이디를 구성합니다.**

```
$ dpconf set-user-mapping-prop -h host -p port \
 mapped-bind-dn:alt-user-bind-dn mapped-bind-pwd-file:filename
```

▼ 익명 클라이언트에 대한 사용자 매핑을 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **인증되지 않은 클라이언트에 대한 매핑을 구성합니다.**

```
$ dpconf set-server-prop -h host -p port \
 user-mapping-anonymous-bind-dn:anonymous-mapping-bind-dn \
 user-mapping-anonymous-bind-pwd-file:filename
```

익명 클라이언트에 대한 항목이 원격 LDAP 서버에 포함되어 있지 않으므로 익명 클라이언트에 대한 매핑이 디렉토리 프록시 서버에서 구성됩니다.

인증되지 않은 사용자가 작업을 수행하도록 허용하는 방법에 대한 자세한 내용은 455 페이지 "익명 액세스를 구성하는 방법"을 참조하십시오.

◆ ◆ ◆ 25 장

25 클라이언트와 디렉토리 프록시 서버 간의 연결

클라이언트와 디렉토리 프록시 서버 간의 연결, 연결 처리기의 개요와 연결 처리기에 사용되는 기준 및 정책에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 20 장, "Connections Between Clients and Directory Proxy Server"를 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

연결 처리기 만들기, 구성 및 삭제

연결 처리기 만들기, 구성 및 삭제 방법과 데이터 보기에 대한 선호도 구성 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 연결 처리기를 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **연결 처리기를 만듭니다.**

```
$ dpconf create-connection-handler -h host -p port connection-handler-name
```

2 **(옵션) 연결 처리기 목록을 봅니다.**

```
$ dpconf list-connection-handlers -h host -p port
```

▼ 연결 처리기를 구성하는 방법

시작하기 전에 연결 처리기의 등록 정보는 디렉토리 프록시 서버 인스턴스에 정의된 다른 연결 처리기의 등록 정보와 관련하여 정의해야 합니다. 모든 연결 처리기의 등록 정보를 고려하여 서로 다른 기준 집합을 지정함으로써 우선 순위가 올바르게 지정되도록 합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 연결 처리기의 키 등록 정보와 상대적 우선 순위를 확인하려면 세부 정보 표시 목록을 봅니다.

```
$ dpconf list-connection-handlers -h host -p port -v
Name                        is-enabled  priority  description
--------------------------  ----------  --------  --------------------------
anonymous                   false       99        unauthenticated connections
default connection handler  true        100       default connection handler
```

연결 처리기 `anonymous` 및 `default connection handler`는 디렉토리 프록시 서버 인스턴스를 만들 때 함께 만들어집니다.

2 연결 처리기의 모든 등록 정보를 봅니다.

```
$ dpconf get-connection-handler-prop -h host -p port connection-handler-name
```

새 연결 처리기의 기본 등록 정보는 다음과 같습니다.

```
aci-source                    :  -
allowed-auth-methods          :  anonymous
allowed-auth-methods          :  sasl
allowed-auth-methods          :  simple
allowed-ldap-ports            :  ldap
allowed-ldap-ports            :  ldaps
bind-dn-filters               :  any
data-view-routing-custom-list :  -
data-view-routing-policy      :  all-routable
description                   :  -
domain-name-filters           :  any
enable-data-view-affinity     :  false
ip-address-filters            :  any
is-enabled                    :  false
is-ssl-mandatory              :  false
priority                      :  99
request-filtering-policy      :  no-filtering
resource-limits-policy        :  no-limits
schema-check-enabled          :  false
user-filter                   :  any
```

3 연결 처리기의 우선 순위를 구성합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name priority:value
```

우선 순위는 1에서 100까지의 번호 중 하나일 수 있으며, 여기서 1이 우선 순위가 가장 높습니다. 디렉토리 프록시 서버 인스턴스에 대해 연결 처리기가 우선 순위대로 평가됩니다.

4 (옵션) 연결 처리기의 DN 필터링 등록 정보를 지정합니다.

이 등록 정보를 사용하면 바인드 DN의 일부 또는 전체를 기준으로 액세스를 제어할 수 있습니다. 등록 정보 값은 정규 표현식입니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 bind-dn-filters:regular-expression
```

바인드 DN 필터는 Java™ 정규 표현식의 형식을 사용합니다. Java 정규 표현식 만들기에 대한 자세한 내용은 `http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html`을 참조하십시오.

예를 들어 `ou=people,dc=example,dc=com`의 사용자의 모든 바인드를 이름이 `secure-handler`인 연결 처리기에 보내려면 다음과 같이 `bind-dn-filters` 등록 정보를 설정합니다.

```
$ dpconf set-connection-handler-prop -h host1 -p 1389 secure-handler \
 bind-dn-filters:"uid=.*,ou=people,dc=example,dc=com"
```

5 (옵션) 이 연결 처리기에 사용하는 요청 필터링 정책의 이름을 지정합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 request-filtering-policy:policy-name
```

여기서 *policy-name*은 기존 요청 필터링 정책의 이름입니다. 요청 필터링 정책을 만들고 구성하는 방법에 대한 자세한 내용은 445 페이지 "요청 필터링 정책 및 검색 데이터 숨기기 규칙 만들기 및 구성"을 참조하십시오.

6 (옵션) 이 연결 처리기에 사용하는 자원 제한 정책의 이름을 지정합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 resource-limits-policy:policy-name
```

여기서 *policy-name*은 기존 자원 제한 정책의 이름입니다. 자원 제한 정책을 만들고 구성하는 방법에 대한 자세한 내용은 448 페이지 "자원 제한 정책 만들기 및 구성"을 참조하십시오.

7 단계 2에 나열된 다른 등록 정보를 구성합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 property:value [property:value ...]
```

예를 들어 연결 처리기가 SSL 연결만을 허용하도록 구성합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 is-ssl-mandatory:true
```

등록 정보에 대한 설명과 유효한 값 목록을 보려면 다음 명령을 실행합니다.

```
$ dpconf help-properties connection-handler
```

8 연결 처리기를 활성화합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name is-enabled:true
```

9 필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.

디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 "디렉토리 프록시 서버를 다시 시작하는 방법"을 참조하십시오.

▼ 연결 처리기를 삭제하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 (옵션) 연결 처리기 목록을 봅니다.

```
$ dpconf list-connection-handlers -h host -p port
```

2 연결 처리기를 한 개 이상 삭제합니다.

```
$ dpconf delete-connection-handler -h host -p port connection-handler-name [connection-handler-name ... ]
```

▼ 데이터 보기에 대한 선호도를 구성하는 방법

연결이 연결 처리기에 할당될 때 해당 연결의 요청이 해당 연결 처리기에 구성된 데이터 보기 목록 또는 구성된 모든 데이터 보기에 표시됩니다. 해당 연결의 연속적인 요청은 첫 번째 요청에 사용된 데이터 보기에만 표시됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 데이터 보기에 대한 선호도를 활성화합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 enable-data-view-affinity:true
```

2 (옵션) 연결 처리기가 요청을 데이터 보기의 사용자 정의 목록에 전달하도록 구성합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name data-view-routing-policy:custom
```

3 **(옵션) 데이터 보기 목록을 구성합니다.**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 data-view-routing-custom-list:view-name [data-view-routing-custom-list:view-name ...]
```

데이터 보기의 기존 목록에 데이터 보기를 추가하려면 이 명령을 사용합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 data-view-routing-custom-list+:view-name
```

데이터 보기의 기존 목록에서 데이터 보기를 제거하려면 이 명령을 사용합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 data-view-routing-custom-list-:view-name
```

요청 필터링 정책 및 검색 데이터 숨기기 규칙 만들기 및 구성

요청 필터링 정책에 대한 개요는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Request Filtering Policies for Connection Handlers"를 참조하십시오. 검색 데이터 숨기기 규칙에 대한 개요는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Search Data Hiding Rules in the Request Filtering Policy"를 참조하십시오.

요청 필터링 정책 및 검색 데이터 숨기기 규칙을 만들고 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 요청 필터링 정책을 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **요청 필터링 정책을 만듭니다.**

```
$ dpconf create-request-filtering-policy policy-name
```

2 **요청 필터링 정책을 연결 처리기에 연결합니다.**

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 request-filtering-policy:policy-name
```

▼ 요청 필터링 정책을 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **요청 필터링 정책의 등록 정보를 봅니다.**

```
$ dpconf get-request-filtering-policy-prop -h host -p port policy-name
```

요청 필터링 정책의 기본 등록 정보는 다음과 같습니다.

```
allow-add-operations                :  true
allow-bind-operations               :  true
allow-compare-operations            :  true
allow-delete-operations             :  true
allow-extended-operations           :  true
allow-inequality-search-operations  :  true
allow-modify-operations             :  true
allow-rename-operations             :  true
allow-search-operations             :  true
allowed-comparable-attrs            :  all
allowed-search-scopes               :  base
allowed-search-scopes               :  one-level
allowed-search-scopes               :  subtree
allowed-subtrees                    :  ""
description                         :  -
prohibited-comparable-attrs         :  none
prohibited-subtrees                 :  none
```

2 **단계 1에 나열된 등록 정보를 하나 이상 설정하여 요청 필터링 정책을 구성합니다.**

```
$ dpconf set-request-filtering-policy-prop -h host -p port policy-name \
  property:value [property:value ...]
```

단계 1에 나열된 등록 정보를 설정하여 다음과 같은 요청 필터링 정책의 기능을 구성합니다.

- 클라이언트가 수행할 수 있는 작업 유형
- 클라이언트에게 표시되거나 표시되지 않는 하위 트리
- 검색 작업 범위
- 검색 필터 유형
- 검색 및 비교 작업에서 비교하거나 비교할 수 없는 속성 유형

▼ 검색 데이터 숨기기 규칙을 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 **요청 필터링 정책에 대한 검색 데이터 숨기기 규칙을 하나 이상 만듭니다.**

```
$ dpconf create-search-data-hiding-rule -h host -p port policy-name rule-name \
 [rule-name ...]
```

2 검색 데이터 숨기기 규칙의 등록 정보를 봅니다.

```
$ dpconf get-search-data-hiding-rule-prop policy-name rule-name
```

검색 데이터 숨기기 규칙의 기본 등록 정보는 다음과 같습니다.

```
attrs                              :  -
rule-action                        :  hide-entry
target-attr-value-assertions       :  -
target-dn-regular-expressions      :  -
target-dns                         :  -
```

3 단계 2에 나열된 등록 정보를 하나 이상 설정하여 검색 데이터 숨기기 규칙을 구성합니다.

```
$ dpconf set-search-data-hiding-rule-prop -h host -p port policy-name rule-name \
  property:value [property:value ...]
```

다음 규칙 작업 중 하나를 사용할 수 있습니다.

`hide-entry` 대상 항목이 반환되지 않습니다.

`hide-attributes` 대상 항목이 반환되지만 지정된 속성은 필터링됩니다.

`show-attributes` 대상 항목이 반환되지만 지정되지 않은 속성은 필터링됩니다.

이 규칙은 다음 항목에 적용할 수 있습니다.

`target-dns` 지정된 DN이 있는 항목

`target-dn-regular-expressions` 지정된 DN 패턴이 있는 항목

`target-attr-value-assertions` 지정된 속성 이름 및 속성 값 쌍이 있는 항목(*attrName#attrValue*)

다음 구성은 `inetorgperson` 유형의 항목을 숨기는 검색 데이터 숨기기 규칙을 정의합니다.

```
$ dpconf set-search-data-hiding-rule-prop -h host1 -p port my-policy my-rule \
  target-attr-value-assertions:objectclass#inetorgperson
```

요청 필터링 정책 및 검색 데이터 숨기기 규칙의 예

다음 예에는 요청 필터링 정책 및 검색 데이터 숨기기 규칙이 포함되어 있습니다. 요청 필터링 정책이 검색 데이터 숨기기 규칙과 결합되면 데이터에 대한 액세스가 다음과 같이 제한됩니다.

- 추가, 삭제, 확장, 수정 및 이름 바꾸기와 같은 작업 유형은 허용되지 않습니다.
- `ou=people,dc=sun,dc=com` 하위 트리만 액세스할 수 있습니다.
- 검색 작업으로 `inetorgperson` 유형 이외의 항목이 반환됩니다.

예 25–1 요청 필터링 정책 예

```
allow-add-operations                :  false
allow-bind-operations               :  true
allow-compare-operations            :  true
allow-delete-operations             :  false
allow-extended-operations           :  false
allow-inequality-search-operations  :  true
allow-modify-operations             :  false
allow-rename-operations             :  false
allow-search-operations             :  true
allowed-comparable-attrs            :  all
allowed-search-scopes               :  base
allowed-search-scopes               :  one-level
allowed-search-scopes               :  subtree
allowed-subtrees                    :  ou=people,dc=sun,dc=com
description                         :  myRequestFilteringPolicy
prohibited-comparable-attrs         :  none
prohibited-subtrees                 :  none
```

예 25–2 검색 데이터 숨기기 규칙 예

```
attrs                             :  -
rule-action                       :  hide-entry
target-attr-value-assertions      :  objectclass:inetorgperson
target-dn-regular-expressions     :  -
target-dns                        :  -
```

자원 제한 정책 만들기 및 구성

자원 제한 정책에 대한 개요는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 “Resource Limits Policies for Connection Handlers”를 참조하십시오. 자원 제한 정책을 만들고 구성하는 방법과 검색 제한을 사용자 정의하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 자원 제한 정책을 만드는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오.

1 자원 제한 정책을 만듭니다.

```
$ dpconf create-resource-limits-policy -h host -p port policy-name
```

자원 제한 정책의 등록 정보를 수정하는 방법에 대한 자세한 내용은 449 페이지 "자원 제한 정책을 구성하는 방법"을 참조하십시오.

2 자원 제한 정책을 연결 처리기에 연결합니다.

```
$ dpconf set-connection-handler-prop -h host -p port connection-handler-name \
 resource-limits-policy:policy-name
```

▼ 자원 제한 정책을 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 자원 제한 정책의 등록 정보를 봅니다.

```
$ dpconf get-resource-limits-policy-prop -h host -p port policy-name
```

자원 제한 정책의 기본 등록 정보는 다음과 같습니다.

```
description                                 :  -
max-client-connections                      :  unlimited
max-connections                             :  unlimited
max-simultaneous-operations-per-connection  :  unlimited
max-total-operations-per-connection         :  unlimited
minimum-search-filter-substring-length      :  unlimited
referral-bind-policy                        :  default
referral-hop-limit                          :  default
referral-policy                             :  default
search-size-limit                           :  unlimited
search-time-limit                           :  unlimited
```

2 단계 1에 나열된 등록 정보를 하나 이상 설정하여 자원 제한 정책을 구성합니다.

```
$ dpconf set-resource-limits-policy-prop -h host -p port policy-name \
  property:value [property:value ...]
```

▼ 검색 제한을 사용자 정의하는 방법

검색 기준 및 검색 범위에 따라 사용자 정의된 제한을 검색 작업에 정의할 수 있습니다. 검색 작업의 대상 DN 및 범위가 지정된 기준과 일치하면 최대 검색 결과 크기가 제한됩니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **사용자 정의 검색 제한을 하나 이상 만듭니다.**

```
$ dpconf create-custom-search-size-limit -h host -p port policy-name \
  custom-search-limit-name [custom-search-limit-name ...]
```

2 **사용자 정의 검색 제한의 기준을 설정합니다.**

```
$ dpconf set-custom-search-size-limit-prop -h host -p port policy-name \
  custom-search-limit-name one-level-search-base-dn:value subtree-search-base-dn:value
```

3 **검색이 단계 2의 기준 중 하나를 충족하면 반환되는 결과 수에 대한 제한을 설정합니다.**

```
$ dpconf set-custom-search-size-limit-prop -h host -p port policy-name \
 custom-search-limit-name search-size-limit:value
```

4 **사용자 정의 검색 제한의 등록 정보를 봅니다.**

```
$ dpconf get-custom-search-size-limit-prop -h host -p port policy-name \
  custom-search-limit-name
```

사용자 정의 검색 제한의 기본 등록 정보는 다음과 같습니다.

```
one-level-search-base-dn  :  -
search-size-limit         :  unlimited
subtree-search-base-dn    :  -
```

디렉토리 프록시 서버를 연결 기반 라우터로 구성

Directory Proxy Server 5.2는 연결 기반 라우터입니다. Directory Proxy Server 5.2에서 클라이언트 연결이 특정 디렉토리 서버로 전달됩니다. 이 클라이언트 연결의 모든 요청은 연결이 끊어지거나 클라이언트가 바인딩 해제되기 전까지 동일한 디렉토리 서버로 전송됩니다.

Directory Proxy Server 6.2는 작업 기반 라우터입니다. 그러나, 호환을 위해 이 버전의 디렉토리 프록시 서버를 다음 절차에 설명된 것처럼 연결 기반 라우터로 구성할 수 있습니다.

▼ 디렉토리 프록시 서버를 연결 기반 라우터로 구성하는 방법

1 **441 페이지"연결 처리기 만들기, 구성 및 삭제"에 설명된 것처럼 연결 처리기를 하나 이상 만들고 구성합니다.**

기본 연결 처리기를 사용할 수도 있습니다.

2 **모든 연결 처리기가 요청을 root data view에만 전달하도록 구성합니다.**

예를 들면 다음과 같습니다.

```
$ dpconf set-connection-handler-prop -h host1 -p 1389 myConnectionHandler \
 data-view-routing-policy:custom data-view-routing-custom-list:"root data view"
```

3 **357 페이지 "LDAP 데이터 소스 만들기 및 구성"에 설명된 것처럼 각 백엔드 LDAP 서버에 데이터 소스를 만들고 구성합니다.**

예를 들면 다음과 같습니다.

```
$ dpconf create-ldap-data-source -h host1 -p 1389 myDataSource host2:2389
```

4 **360 페이지 "LDAP 데이터 소스 풀 만들기 및 구성"에 설명된 것처럼 데이터 소스 풀을 만들고 구성합니다.**

예를 들면 다음과 같습니다.

```
$ dpconf create-ldap-data-source-pool -h host1 -p 1389 myDataSourcePool
```

5 **361 페이지 "데이터 소스 풀에 LDAP 데이터 소스 첨부"에 설명된 것처럼 모든 데이터 소스를 데이터 소스 풀에 연결합니다.**

예를 들면 다음과 같습니다.

```
$ dpconf attach-ldap-data-source -h host1 -p 1389 myDataSourcePool myDataSource
```

6 **435 페이지 "바인드 재생을 사용하여 요청 전달"에 설명된 것처럼 각 데이터 소스가 BIND 재생을 사용하여 클라이언트를 인증하도록 구성합니다.**

예를 들면 다음과 같습니다.

```
$ dpconf set-ldap-data-source-prop -h host1 -p 1389 myDataSource \
 client-cred-mode:use-client-identity
```

7 **370 페이지 "클라이언트 선호도 구성"에 설명된 것처럼 클라이언트 연결과 데이터 소스 풀 간에 선호도를 구성합니다.**

예를 들면 다음과 같습니다.

```
$ dpconf set-ldap-data-source-pool-prop -h host1 -p 1389 myDataSourcePool \
 enable-client-affinity:true client-affinity-policy:read-write-affinity-after-write
```

◆ ◆ ◆ 26 장

26 디렉토리 프록시 서버 클라이언트 인증

디렉토리 프록시 서버에서 클라이언트 인증에 대한 개요는 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 21 장, “Directory Proxy Server Client Authentication”을 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

클라이언트와 디렉토리 프록시 서버 간의 수신기 구성

디렉토리 프록시 서버에서는 클라이언트와의 통신을 위해 보안 수신기와 비보안 수신기가 제공됩니다. 디렉토리 프록시 서버의 수신기에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 “Directory Proxy Server Client Listeners”를 참조하십시오. 이 절에서는 수신기를 구성하는 방법에 대해 설명합니다.

▼ 클라이언트와 디렉토리 프록시 서버 간에 수신기를 구성하는 방법

주 – 이 절차에서는 클라이언트와 디렉토리 프록시 서버 간에 비보안 수신기를 구성합니다. 보안 수신기를 구성하려면 동일한 절차를 수행하되 `ldap`를 `ldaps`로 바꿉니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 “디렉토리 서비스 제어 센터 인터페이스” 및 DSCC 온라인 도움말을 참조하십시오. DSCC에서는 성능 탭에서 이 등록 정보를 구성할 수 있습니다.

1 **비보안 수신기의 등록 정보를 봅니다.**

```
$ dpconf get-ldap-listener-prop -h host -p port
```

비보안 수신기의 기본 등록 정보는 다음과 같습니다.

```
connection-idle-timeout          :  1h
connection-read-data-timeout     :  2s
connection-write-data-timeout    :  1h
is-enabled                       :  true
listen-address                   :  0.0.0.0
listen-port                      :  port-number
max-connection-queue-size        :  128
max-ldap-message-size            :  unlimited
number-of-threads                :  2
use-tcp-no-delay                 :  true
```

2 **요구 사항에 따라 단계 1에 나열된 등록 정보 중 하나 이상을 변경합니다.**

```
$ dpconf set-ldap-listener-prop -h host -p port property:new-value
```

예를 들어 host1에서 실행 중인 디렉토리 프록시 서버 인스턴스의 비보안 포트를 비활성화하려면 다음 명령을 실행합니다.

```
$ dpconf set-ldap-listener-prop -h host1 -p 1389 is-enabled:false
```

주의 – 권한이 없는 포트 번호를 사용하려면 디렉토리 프록시 서버를 루트로 실행해야 합니다.

비보안 포트 번호를 변경하려면 다음 명령을 실행합니다.

```
$ dpconf set-ldap-listener-prop -h host -p port listen-port:new-port-number
```

3 **필요한 경우 디렉토리 프록시 서버 인스턴스를 다시 시작하여 변경 사항을 적용합니다.**

특정 수신기 등록 정보를 변경한 후에는 서버를 다시 시작해야 합니다. 서버를 다시 시작해야 하는 경우 `dpconf`는 경고를 표시합니다. 디렉토리 프록시 서버를 다시 시작하는 방법에 대한 자세한 내용은 334 페이지 “디렉토리 프록시 서버를 다시 시작하는 방법”을 참조하십시오.

디렉토리 프록시 서버에 대해 클라이언트 인증

기본적으로 디렉토리 프록시 서버는 단순 바인드 인증에 맞게 구성됩니다. 단순한 바인드 인증은 추가로 구성할 필요가 없습니다.

클라이언트와 디렉토리 프록시 서버 간 인증에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 “Client Authentication Overview”를 참조하십시오. 인증을 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 인증서 기반 인증을 구성하는 방법

클라이언트의 인증서 기반 인증에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Configuring Certificates in Directory Proxy Server"를 참조하십시오. 이 절에서는 인증서 기반 인증을 구성하는 방법에 대해 설명합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

주 – 인증서 기반 인증은 SSL 연결을 통해서만 수행할 수 있습니다.

- **클라이언트가 SSL 연결을 설정할 때 클라이언트가 인증서를 제공하도록 디렉토리 프록시 서버를 구성합니다.**

```
$ dpconf set-server-prop -h host -p port allow-cert-based-auth:require
```

▼ 익명 액세스를 구성하는 방법

익명 액세스에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Anonymous Access"를 참조하십시오. 익명 클라이언트의 아이디를 다른 아이디에 매핑하는 방법에 대한 자세한 내용은 438 페이지 "대체 사용자로 요청 전달"을 참조하십시오.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **인증되지 않은 사용자가 작업을 수행하도록 허용합니다.**

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:true
```

▼ SASL 외부 바인드에 대해 디렉토리 프록시 서버를 구성하는 방법

SASL 외부 바인드에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Using SASL External Bind"를 참조하십시오.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **인증되지 않은 작업을 허용하지 않습니다.**

```
$ dpconf set-server-prop -h host -p port allow-unauthenticated-operations:false
```

2. **연결을 설정할 때 클라이언트가 인증서를 제공하도록 요구합니다.**

```
$ dpconf set-server-prop -h host -p port allow-cert-based-auth:require
```

클라이언트는 DN을 포함하는 인증서를 제공합니다.

3 SASL 외부 바인드로 클라이언트 인증을 활성화합니다.

```
$ dpconf set-server-prop -h host -p port allow-sasl-external-authentication:true
```

4 백엔드 LDAP 서버에서 클라이언트 인증서를 매핑하려면 디렉토리 프록시 서버에 사용되는 아이디를 구성합니다.

```
$ dpconf set-server-prop -h host -p port cert-search-bind-dn:bind-DN \
 cert-search-bind-pwd-file:filename
```

5 디렉토리 프록시 서버가 검색하는 하위 트리의 기본 DN을 구성합니다.

디렉토리 프록시 서버는 하위 트리를 검색하여 클라이언트 인증서에 매핑되는 사용자 항목을 찾습니다.

```
$ dpconf set-server-prop -h host -p port cert-search-base-dn:base-DN
```

6 클라이언트 인증서의 정보를 LDAP 서버의 인증서에 매핑합니다.

a. 인증서를 포함하는 LDAP 서버의 속성에 이름을 지정합니다.

```
$ dpconf set-server-prop cert-search-user-attribute:attribute
```

b. 클라이언트 인증서의 속성을 인증서가 포함된 LDAP 서버 항목의 DN에 매핑합니다.

```
$ dpconf set-server-prop -h host -p port \
 cert-search-attr-mappings:client-side-attribute-name:server-side-attribute-name
```

예를 들어 DN `cn=user1,o=sun,c=us`가 있는 클라이언트 인증서를 DN `uid=user1,o=sun`이 있는 LDAP 항목에 매핑하려면 다음 명령을 실행합니다.

```
$ dpconf set-server-prop -h host1 -p 1389 cert-search-attr-mappings:cn:uid \
 cert-search-attr-mappings:o:o
```

7 (옵션) SASL 외부 바인드 작업에 대한 요청을 모든 데이터 보기 또는 데이터 보기의 사용자 정의 목록에 전달합니다.

- 모든 데이터 보기에 요청을 전달하려면 다음 명령을 실행합니다.

```
$ dpconf set-server-prop -h host -p port cert-data-view-routing-policy:all-routable
```

- 데이터 보기 목록에 요청을 전달하려면 다음 명령을 실행합니다.

```
$ dpconf set-server-prop -h host -p port cert-data-view-routing-policy:custom \
 cert-data-view-routing-custom-list:view-name [view-name...]
```

◆ ◆ ◆ 27 장

27 디렉토리 프록시 서버 로깅

디렉토리 프록시 서버는 정보를 액세스 로그 및 오류 로그에 기록합니다. 디렉토리 서버와는 달리 디렉토리 프록시 서버에는 감사 로그가 없습니다. 디렉토리 프록시 서버 로그에 대한 설명은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 23 장, "Directory Proxy Server Logging"을 참조하십시오.

이 장은 다음 내용으로 구성되어 있습니다.

디렉토리 프록시 서버 로그 보기

디렉토리 프록시 서버 로그는 로그 파일을 직접 열거나 디렉토리 서비스 제어 센터(Directory Service Control Center, DSCC)를 사용하여 볼 수 있습니다.

기본적으로 로그는 다음 디렉토리에 저장됩니다.

instance-path`/logs`

다음 그림은 DSCC에서 캡처한 디렉토리 프록시 서버 오류 로그 화면을 보여줍니다.

그림 27–1 디렉토리 프록시 서버의 오류 로그 창

디렉토리 프록시 서버 로그 구성

디렉토리 프록시 서버 오류 로그 및 액세스 로그는 `dpconf` 명령 또는 DSCC를 사용하여 구성할 수 있습니다. DSCC를 사용하여 로그를 구성하는 방법에 대한 자세한 내용은 디렉토리 프록시 서버 온라인 도움말을 참조하십시오. 이 절에서는 `dpconf` 명령을 사용하여 디렉토리 프록시 서버 로그를 구성하는 방법에 대해 설명합니다.

다음 명령을 실행하여 허용되는 값 및 기본값과 함께 구성 옵션의 전체 목록을 검색할 수 있습니다.

```
$ dpconf help-properties error-log

$ dpconf help-properties access-log
```

▼ 디렉토리 프록시 서버 액세스 로그 및 오류 로그를 구성하는 방법

이 절차를 수행하면 디렉토리 프록시 서버 액세스 로그가 구성됩니다. 디렉토리 프록시 서버 오류 로그를 구성하려면 동일한 절차를 수행하되 `access`를 `error`로 바꿉니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 액세스 로그의 등록 정보를 봅니다.

```
$ dpconf get-access-log-prop -h host -p port
```

액세스 로그의 기본 등록 정보는 다음과 같습니다.

```
default-log-level                :  info
enable-log-rotation                 :  true
log-buffer-size                  :  9.8k
log-file-name                    :  logs/access
log-file-perm                    :  600
log-level-client-connections     :  -
log-level-client-disconnections  :  -
log-level-client-operations      :  -
log-level-connection-handlers    :  -
log-level-data-sources           :  -
log-level-data-sources-detailed  :  -
log-min-size                        :  100M
log-rotation-frequency           :  1h
log-rotation-policy              :  size
log-rotation-size                :  100M
log-rotation-start-day           :  1
log-rotation-start-time          :  0000
log-search-filters               :  false
max-age                          :  unlimited
max-log-files                    :  10
max-size                         :  unlimited
min-free-disk-space-size         :  1M
```

2 단계 1에 나열된 하나 이상의 등록 정보를 변경합니다.

```
$ dpconf set-access-log-prop -h host -p port property:value \
 [property:value ...]
```

예를 들어 모든 메시지 범주의 기본 로그 수준을 경고로 설정하려면 `default-log-level` 등록 정보 값을 `warning`으로 설정합니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 default-log-level:warning
```

각 메시지 범주의 로그 수준에 상관없이 모든 로그를 비활성화하려면 `default-log-level` 등록 정보 값을 `none`으로 설정합니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 default-log-level:none
```

특정 로그 수준을 기본 로그 수준으로 재설정하려면 해당 로그 수준 등록 정보를 `inherited`로 설정합니다. 예를 들어 클라이언트 연결에 대한 로그 수준을 재설정하려면 다음 명령을 실행합니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-level-client-connections:inherited
```

`set-access-log-prop` 하위 명령으로 설정할 수 있는 등록 정보에 대한 자세한 내용을 보려면 다음을 입력합니다.

```
$ dpconf help-properties access-log
```

디렉토리 프록시 서버 로그 회전 구성

기본적으로 로그 파일은 파일 크기가 100MB에 도달할 때 회전됩니다. 기본적으로 10개의 로그 파일이 유지되며, 이후에 회전 프로시저는 가장 오래된 로그 파일을 덮어쓰기 시작합니다. 이 절에서는 디렉토리 프록시 서버 로그 회전을 예약하는 방법, 로그를 수동으로 회전하는 방법 및 로그 회전을 비활성화하는 방법에 대해 설명합니다. 구성 예는 462 페이지 "로그 회전 구성 예"를 참조하십시오.

▼ 액세스 로그 및 오류 로그가 주기적으로 회전하도록 구성하는 방법

이 절차를 수행하면 디렉토리 프록시 서버 액세스 로그가 구성됩니다. 디렉토리 프록시 서버 오류 로그를 구성하려면 동일한 절차를 수행하되 `access`를 `error`로 바꿉니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 (옵션) 액세스 로그의 등록 정보를 봅니다.

```
$ dpconf get-access-log-prop -h host -p port
```

2 (옵션) 액세스 로그의 등록 정보에 대해 유효한 값을 봅니다.

```
$ dpconf help-properties access-log
```

3 로그가 특정 크기에 도달할 때 로그를 회전하려면 다음 등록 정보를 설정합니다.

```
$ dpconf set-access-log-prop -h host -p port \
 log-rotation-policy:size log-rotation-size:maximum file size
```

최대 파일 크기의 단위가 지정되어 있지 않으면 **바이트**가 기본 단위로 사용됩니다. 로그 파일이 정의된 크기에 도달할 때 로그가 회전됩니다. 파일 크기는 1MB 이상, 2GB 이하여야 합니다.

크기를 기준으로 로그를 회전하는 방법에 대한 예는 462 페이지 "로그 크기를 기준으로 로그 회전"을 참조하십시오.

4 **로그 크기에 상관없이 로그를 주기적으로 회전하려면 다음 등록 정보를 설정합니다.**

```
$ dpconf set-access-log-prop -h host -p port \
 log-rotation-frequency:interval in months, weeks, hours, or minutes \
 log-rotation-policy:periodic \
 log-rotation-start-day:day in week (1-7) or day in the month (1-31) \
 log-rotation-start-time:time of day (hhmm)
```

로그가 매월 31일에 회전하도록 구성되었으나 해당 월의 일수가 31일 미만이면 로그는 다음 달 1일에 회전됩니다.

로그를 주기적으로 회전하는 방법에 대한 예는 462 페이지 "시간을 기준으로 로그 회전"을 참조하십시오.

5 **파일 크기가 큰 경우에 로그를 주기적으로 회전하려면** `log-rotation-frequency` **및** `log-min-size` **등록 정보를 설정합니다.**

```
$ dpconf set-access-log-prop -h host -p port \
 log-rotation-frequency:interval in months, weeks, hours, or minutes \
 log-rotation-policy:periodic log-min-size:minimum file size
 log-rotation-start-day:day in week (1-7) or day in the month (1-31) \
 log-rotation-start-time:time of day (hhmm)
```

`log-min-size` 등록 정보는 로그의 최소 크기를 나타냅니다. 로그 파일이 지정된 크기보다 큰 경우에만 예약된 시간에 회전이 수행됩니다.

로그가 매월 31일에 회전하도록 구성되었으나 해당 월의 일수가 31일 미만이면 로그는 다음 달 1일에 회전됩니다.

파일 크기가 큰 경우에 로그를 주기적으로 회전하는 방법에 대한 예는 463 페이지 "시간 및 로그 크기를 기준으로 로그 회전"을 참조하십시오.

▼ 액세스 로그 및 오류 로그 파일을 수동으로 회전하는 방법

이 절차를 수행하면 디렉토리 프록시 서버 액세스 로그가 회전됩니다. 디렉토리 프록시 서버 오류 로그를 회전하려면 동일한 절차를 수행하되 `access`를 `error`로 바꿉니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **액세스 로그를 회전합니다.**

  ```
  $ dpconf rotate-log-now -h host -p port access
  ```

▼ 액세스 로그 및 오류 로그 회전을 비활성화하는 방법

이 절차를 수행하면 디렉토리 프록시 서버 액세스 로그 회전이 비활성화됩니다. 디렉토리 프록시 서버 오류 로그 회전을 비활성화하려면 동일한 절차를 수행하되 `access`를 `error`로 바꿉니다.

- **로그 파일 회전을 비활성화합니다.**

  ```
  $ dpconf set-access-log-prop -h host -p port enable-log-rotation:false
  ```

로그 회전 구성 예

로그 크기, 시간 또는 두 가지 모두를 기준으로 로그 회전을 구성하는 방법에 대한 예는 다음과 같습니다.

로그 크기를 기준으로 로그 회전

이 절에 나와 있는 예는 로그 크기만을 기준으로 로그 회전을 구성하는 방법을 보여줍니다. 다음과 같이 구성하면 로그가 마지막으로 회전된 시간에 상관없이 로그 크기가 10MB에 도달할 때 로그를 회전합니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-policy:size \
  log-rotation-size:10M
```

시간을 기준으로 로그 회전

이 절에 나와 있는 예는 로그 크기에 상관없이 마지막 회전 시간을 기준으로 로그 회전을 구성하는 방법을 보여줍니다.

- 다음과 같이 구성하면 로그 파일 크기에 상관없이 금일 3시에 로그를 회전한 다음 8시간마다 회전합니다.

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:8h \
   log-rotation-policy:periodic log-rotation-start-time:0300
  ```

- 다음과 같이 구성하면 로그 파일 크기에 상관없이 매일 3시, 13시 및 23시에 로그를 회전합니다. `log-rotation-start-time` 매개 변수가 `log-rotation-frequency` 매개 변수보다 우선적으로 적용되기 때문에 로그가 23시에 회전된 다음 4시간 후에 주기적으로 회전됩니다. 또한 23시에 회전되지 않고 10시간 후에 회전됩니다.

  ```
  $ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:10h \
   log-rotation-policy:periodic log-rotation-start-time:0300
  ```

- 다음과 같이 구성하면 로그 파일 크기에 상관없이 매주 월요일 정오에 로그를 회전합니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:1w \
 log-rotation-policy:periodic log-rotation-start-day:2 log-rotation-start-time:1200
```

- 다음과 같이 구성하면 로그 파일 크기에 상관없이 월요일 정오에 로그를 회전한 다음 3일마다 회전합니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:3d \
 log-rotation-policy:periodic log-rotation-start-day:2 log-rotation-start-time:1200
```

  로그가 월요일, 목요일, 일요일, 수요일 등의 순서로 회전됩니다. `log-rotation-start-day` 매개 변수는 첫 번째 주에만 적용됩니다. 따라서, 로그는 두 번째 주 월요일에는 회전되지 않습니다.

- 다음과 같이 구성하면 로그 크기에 상관없이 매월 22일 정오에 로그를 회전합니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:1m \
 log-rotation-policy:periodic log-rotation-start-day:22 \
 log-rotation-start-time:1200
```

`log-rotation-start-day`가 31로 설정되어 있고 해당 월의 일수가 30일이면 로그는 다음 달 1일에 회전됩니다. `log-rotation-start-day`가 31로 설정되어 있고 해당 월의 일수가 28일(2월)이면 로그는 다음 달 3일에 회전됩니다.

시간 및 로그 크기를 기준으로 로그 회전

다음 예는 파일 크기가 큰 경우에 지정된 간격으로 로그 회전을 구성하는 방법을 보여줍니다.

다음과 같이 구성하면 로그 파일 크기가 1MB를 초과하는 경우 매일 3시, 11시 및 19시에 로그를 회전합니다. 로그 파일 크기가 1MB를 초과하지 않으면 로그 파일은 회전되지 않습니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 log-rotation-frequency:8h \
 log-rotation-policy:periodic log-min-size:1M log-rotation-start-time:0300
```

디렉토리 프록시 서버 로그 삭제

디렉토리 프록시 서버에서 시간, 크기 또는 여유 디스크 공간(기본값)을 기준으로 로그 삭제를 구성할 수 있습니다. 이러한 삭제 정책에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Log File Deletion"을 참조하십시오.

다음 절차를 수행하면 액세스 로그에 대한 로그 삭제가 구성됩니다. 오류 로그에 대한 로그 삭제를 구성하려면 동일한 명령을 사용하되 `access`를 `error`로 바꿉니다.

▼ 시간을 기준으로 액세스 로그 및 오류 로그 삭제를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **로그 파일의 최대 사용 기간을 지정합니다.**

```
$ dpconf set-access-log-prop -h host -p port max-age:duration
```

여기서 *duration*은 일(d), 주(w) 또는 월(M) 단위로 구성됩니다. 예를 들어 5일 이상 지난 백업 로그 파일을 삭제하려면 다음 명령을 사용합니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 max-age:5d
```

▼ 파일 크기를 기준으로 액세스 로그 및 오류 로그 삭제를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **로그 파일의 최대 크기를 지정합니다.**

```
$ dpconf set-access-log-prop -h host -p port max-size:memory-size
```

예를 들어 1MB를 초과하는 백업 로그 파일을 삭제하려면 다음 명령을 사용합니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 max-size:1M
```

▼ 여유 디스크 공간을 기준으로 액세스 로그 및 오류 로그 삭제를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

- **사용 가능한 최소 디스크 공간을 지정합니다.**

```
$ dpconf set-access-log-prop -h host -p port min-free-disk-space-size:memory-size
```

예를 들어 사용 가능한 디스크 공간이 2MB 미만일 때 백업 로그 파일을 삭제하려면 다음 명령을 사용합니다.

```
$ dpconf set-access-log-prop -h host1 -p 1389 min-free-disk-space-size:2M
```

syslogd 데몬에 경고 로깅

이 절에서는 syslogd 데몬에 대한 경고 메시지 로깅을 구성하는 방법과 운영 체제에서 syslog 경고를 허용하도록 구성하는 방법에 대해 설명합니다.

▼ 디렉토리 프록시 서버에서 경고를 syslogd 데몬에 기록하도록 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 (옵션) 시스템 로그 경고의 현재 등록 정보 값을 봅니다.

```
$ dpconf get-server-prop -h host -p port syslog-alerts-enabled \
 syslog-alerts-facility syslog-alerts-host
```

시스템 로그 경고의 기본 등록 정보는 다음과 같습니다.

```
syslog-alerts-enabled   :  false
syslog-alerts-facility  :  USER
syslog-alerts-host      :  localhost
```

syslog-alerts-host 등록 정보는 메시지를 보낼 syslogd 데몬의 호스트 이름을 정의합니다. syslog-alerts-facility 등록 정보는 읽기 전용이며 메시지를 시스템 로그의 user 범주로 보냅니다.

2 경고 메시지를 syslogd 데몬에 기록하도록 활성화합니다.

```
$ dpconf set-server-prop -h host -p port syslog-alerts-enabled:true
```

3 (옵션) 경고 메시지를 다른 호스트의 syslogd 데몬으로 보냅니다.

```
$ dpconf set-server-prop -h host -p port syslog-alerts-host:hostname
```

운영 체제에서 syslog 경고를 허용하도록 구성

이 절에서는 Solaris™, Linux 및 HP-UX 운영 체제에서 syslog 경고를 허용하도록 구성하는 방법에 대해 설명합니다.

▼ Solaris OS에서 syslog 경고를 허용하도록 구성하는 방법

1 syslog 구성 파일에 적절한 기능을 추가합니다.

예를 들어 USER 기능을 사용하여 모든 경고를 저장하려면 /etc/syslog.conf에 다음 행을 추가합니다.

```
user.info       /var/adm/info
```

여기서 /var/adm/info는 메시지를 저장할 로컬 디렉토리 예입니다. 작업을 계속하기 전에 /var/adm/info가 있는지 확인합니다.

2 syslogd 데몬을 다시 시작합니다.

a. Solaris 8 및 9의 경우 다음을 입력하여 syslogd를 다시 시작합니다.

```
$ /etc/init.d/syslog stop | start
```

b. Solaris 10의 경우 다음을 입력하여 syslogd를 다시 시작합니다.

```
$ svcadm restart system/system-log
```

3 메시지가 syslog에 기록되는지 확인합니다.

```
$ logger -p user.info "Test message"
$ cat /var/adm/info
 Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
```

▼ Linux에서 syslog 경고를 허용하도록 구성하는 방법

1 syslog 구성 파일에 적절한 기능을 추가합니다.

예를 들어 USER 기능을 사용하여 모든 경고를 저장하려면 /etc/syslog.conf에 다음 행을 추가합니다.

```
user.info       /var/adm/info
```

여기서 /var/adm/info는 메시지를 저장할 로컬 디렉토리 예입니다. 작업을 계속하기 전에 /var/adm/info가 있는지 확인합니다.

2 syslogd 데몬이 -r 옵션과 함께 실행되도록 구성합니다.

이 옵션을 사용하면 syslogd가 네트워크 연결을 허용합니다. 기본적으로 -r 옵션은 설정되어 있지 않습니다.

-r 옵션을 설정하려면 /etc/sysconfig/syslog에 다음 행을 추가합니다.

```
SYSLOGD_OPTIONS="-m 0 -r"
```

/etc/sysconfig/syslog가 없으면 동일한 행을 /etc/init.d/syslog에 추가합니다.

3 syslogd 데몬을 다시 시작합니다.

```
$ /etc/init.d/syslog stop | start
```

4 메시지가 syslog에 기록되는지 확인합니다.

```
$ logger -p user.info "Test message"
$ cat /var/adm/info
 Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
```

▼ HP-UX에서 syslog 경고를 허용하도록 구성하는 방법

1 syslog 구성 파일에 적절한 기능을 추가합니다.

예를 들어 USER 기능을 사용하여 모든 경고를 저장하려면 /etc/syslog.conf에 다음 행을 추가합니다.

```
user.info       /var/adm/info
```

여기서 /var/adm/info는 메시지를 저장할 로컬 디렉토리 예입니다. 작업을 계속하기 전에 /var/adm/info가 있는지 확인합니다.

2 syslogd 데몬을 다시 시작합니다.

```
$ /sbin/init.d/syslogd stop | start
```

3 메시지가 syslog에 기록되는지 확인합니다.

```
$ logger -p user.info "Test message"
$ cat /var/adm/info
 Jun 19 17:18:38 host user: [ID 12345 user.info] Test message
```

디렉토리 프록시 서버 및 디렉토리 서버 액세스 로그를 통해 클라이언트 요청 추적

클라이언트 요청 경로를 추적하려면 디렉토리 프록시 서버 액세스 로그 및 디렉토리 서버 액세스 로그에 요청이 기록되는 방식을 알고 있어야 합니다. 이 절의 내용을 이해하려면 먼저 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Tracking Client Requests Through Directory Proxy Server and Directory Server Access Logs"를 읽으십시오.

▼ 디렉토리 서버에서 디렉토리 프록시 서버를 통해 클라이언트 응용 프로그램에 대한 작업을 추적하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 디렉토리 서버 액세스 로그에서 추적할 작업의 연결 번호를 찾습니다.

예를 들어 액세스 로그에서 다음 행은 연결 번호가 conn=12839인 op=2 작업을 나타냅니다.

```
[20/Jul/2006:18:01:49 -0500] conn=12839 op=2 msgId=4 - SRCH base="dc=example,dc=com"
 scope=2 filter="(objectClass=organizationalunit)" attrs=ALL
```

2 이 연결에 대한 디렉토리 프록시 서버 연결 정보를 얻습니다.

이 정보를 얻으려면 디렉토리 서버 액세스 로그를 검색하여 연결 번호가 일치하는 작업을 모두 찾습니다. 예를 들어 UNIX 시스템에서 다음과 같이 `grep` 명령을 실행하여 디렉토리 서버 액세스 로그에서 연결 `conn=12839`와 일치하는 행을 모두 찾습니다.

```
$ grep conn=12839 access
```

초기 LDAP 연결을 보여주는 행이 찾고 있는 정보이며 이는 다음과 유사합니다.

```
[19/Jul/2006:16:32:51 -0500] conn=12839 op=-1 msgId=-1 - fd=27 slot=27
 LDAP connection from 129.153.160.175:57153 to 129.153.160.175
```

위의 행은 **129.153.160.175:57153**에서 디렉토리 서버로의 LDAP 연결이 있음을 나타냅니다. 포트 번호(**57153**)는 디렉토리 프록시 서버 액세스 로그로 다시 연결하는 데 필요한 정보입니다. 포트 번호를 사용하면 디렉토리 프록시 서버 로그에서 일치하는 연결을 찾을 수 있으며 이 연결에서 클라이언트 정보를 찾아볼 수 있습니다.

연결이 처음 구성된 이후에 로그 파일이 회전된 적이 있다면 현재 액세스 로그 파일과 아카이브된 로그 파일을 모두 검색해야 합니다.

3 디렉토리 프록시 서버 액세스 로그에서 일치하는 연결을 찾습니다.

이 정보를 얻으려면 디렉토리 프록시 서버 액세스 로그를 검색하여 포트 번호가 일치하는 작업을 모두 찾습니다.

로그 파일에서 포트 번호가 동일한 항목을 여러 개 찾을 수 있습니다. 올바른 항목을 찾으려면 디렉토리 서버 로그 항목의 타임스탬프를 검색에 포함시킵니다.

예를 들어 UNIX 시스템의 경우 다음과 같이 `grep` 명령을 실행하여 디렉토리 서버 로그에서 찾은 타임스탬프 및 포트 번호와 일치하는 연결 항목을 찾습니다.

```
$ grep 19/Jul/2006:16:32 access | grep 57153
```

초 값은 서버 시간에 약간의 차이를 두기 위해 타임스탬프에서 제외됩니다.

디렉토리 프록시 서버 로그에서 일치하는 행은 다음과 유사합니다.

```
[19/Jul/2006:16:32:51 -0500] - SERVER_OP  - INFO  - Created BIND  LDAP connection
 s_conn=sunds-d1m1-9389:34 client=0.0.0.0:57153
 server=idm160.central.sun.com:9389 main
```

이 행은 디렉토리 프록시 서버가 `s_conn=sunds-d1m1-9389:34`에 대한 `BIND` 연결을 만들었음을 나타냅니다. 디렉토리 프록시 서버는 TCP 포트 `57153`에서 자신을 `client=0.0.0.0` 클라이언트로 자체 식별합니다.

로그의 이 행에서 얻을 수 있는 중요 정보는 서버 아이디 및 포트 번호(**s_conn=sunds-d1m1-9389:34**)입니다.

4 이전 단계에서 식별된 서버 아이디 및 포트 번호와 일치하는 작업을 모두 찾습니다.

이 정보를 얻으려면 디렉토리 프록시 서버 액세스 로그를 검색하여 서버 아이디 및 포트 번호가 일치하는 작업을 모두 찾습니다.

예를 들어 UNIX 시스템에서 다음과 같이 grep 명령을 실행하여 이전 단계에서 찾은 서버 아이디와 일치하는 작업을 찾습니다.

```
$ grep s_conn=sunds-d1m1-9389:34 access
```

이 경우, 이러한 작업은 며칠 동안 지속되었을 수 있기 때문에 타임스탬프 검색은 유용하지 않습니다. 그러나, 검색에서 반환된 작업이 올바른 것인지 확인해야 합니다. Create 연결 문이 여러 개인 경우 원래 검색 문과 일치하는 항목을 찾아야 합니다. 이 작업을 수행하려면 타임스탬프를 단계 1에서 찾은 타임스탬프와 일치시킵니다.

디렉토리 프록시 서버 액세스 로그에서 추출된 다음 내용은 s_conn=sunds-d1m1-9389:34에 대해 반환된 모든 작업을 보여줍니다.

```
[19/Jul/2006:16:32:51 -0500] - SERVER_OP  - INFO - Created BIND LDAP connection
 s_conn=sunds-d1m1-9389:34 client=0.0.0.0:57153 server=idm160.central.sun.com:9389 main
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=0 BIND dn="cn=directory manager"
 method="SIMPLE" s_msgid=3 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=0 BIND RESPONSE err=0 msg=""
 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=1 SEARCH base="dc=example,dc=com"
 scope=2 s_msgid=4 s_conn=sunds-d1m1-9389:34
[20/Jul/2006:18:01:49 -0500] - SERVER_OP  - INFO  - conn=31 op=1 SEARCH RESPONSE err=0 msg=""
 nentries=1 s_conn=sunds-d1m1-9389:34
```

이 정보를 통해 디렉토리 프록시 서버의 검색 작업에 해당하는 연결 아이디가 31(conn=31)임을 알 수 있습니다.

5 이전 단계에서 찾은 연결 아이디와 일치하는 클라이언트 연결 IP 주소를 찾습니다.

이 정보를 얻으려면 디렉토리 프록시 서버 액세스 로그를 검색하여 연결 아이디 및 타임스탬프가 일치하는 작업을 모두 찾습니다. 사용할 타임스탬프는 단계 1의 원래 검색 문에 있는 타임스탬프입니다.

예를 들어 UNIX 시스템에서 다음과 같이 grep 명령을 실행하여 클라이언트 연결 IP 주소를 찾습니다.

```
$ grep "20/Jul/2006:18:01" access | grep conn=31
```

찾으려는 행은 다음과 유사합니다.

```
[20/Jul/2006:18:01:49 -0500] - CONNECT - INFO  - conn=31 client=129.150.64.156:2031
server=0.0.0.0:11389 protocol=LDAP
```

6 이전 단계에서 찾은 IP 주소의 소유자를 확인합니다.

이 정보를 통해 디렉토리 서버에서 작업을 수행한 담당자를 정확하게 설정할 수 있습니다.

◆ ◆ ◆ 28장

28 디렉토리 프록시 서버 모니터링 및 경고

모니터링은 디렉토리 프록시 서버 및 해당 데이터 소스의 오류를 감지합니다.

디렉토리 프록시 서버의 모니터링 프레임워크와 `cn=monitor` 항목의 세부 레이아웃에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Monitoring Directory Proxy Server"를 참조하십시오. 이 장은 다음 내용으로 구성되어 있습니다.

디렉토리 프록시 서버에 대한 모니터링된 데이터 검색

디렉토리 프록시 서버에 대한 모니터링된 데이터를 검색하려면 `cn=monitor` 항목을 사용합니다. 이 항목은 로컬 메모리에 저장된 데이터베이스에서 디렉토리 프록시 서버에 의해 관리됩니다. `cn=monitor` 항목에 대한 LDAP 검색을 수행하여 `cn=monitor` 아래에서 속성을 검색할 수 있습니다. 이 항목을 검색하려면 프록시 관리자로 바인드해야 합니다.

JVM을 사용하여 모니터링된 데이터를 검색하는 방법에 대한 자세한 내용은 476 페이지 "JVM을 사용하여 디렉토리 프록시 서버에 대한 모니터링된 데이터 검색"을 참조하십시오.

데이터 소스에 대한 모니터링된 데이터 검색

디렉토리 프록시 서버가 데이터 소스의 상태를 모니터링하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Reference**의 "Monitoring Data Sources"를 참조하십시오. 이 절에서는 데이터 소스의 모니터링을 구성하는 방법에 대해 설명합니다.

▼ 오류를 수신하여 데이터 소스를 모니터링하는 방법

이 모니터링 유형에서 디렉토리 프록시 서버는 디렉토리 프록시 서버 및 데이터 소스 간의 트래픽에서 오류를 수신합니다. 디렉토리 프록시 서버는 오류가 감지될 경우 반응하지만 데이터 소스를 적극적으로 테스트하지 않기 때문에 이 모니터링 유형을 사후 행동 모니터링이라고 합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **데이터 소스에 대한 모니터링 모드를 `reactive`로 설정합니다.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:reactive
   ```

2. **474 페이지 "디렉토리 프록시 서버에 대한 관리 경고 구성"에 설명된 것처럼 오류가 감지되거나 데이터 소스가 오프라인 또는 온라인 상태가 될 경우 경고를 보내도록 구성합니다.**

▼ 전용 연결을 정기적으로 설정하여 데이터 소스를 모니터링하는 방법

디렉토리 프록시 서버는 지정된 간격 동안 데이터 소스에 요청이나 응답이 없을 경우 데이터 소스에 대한 전용 연결을 만듭니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **데이터 소스에 대한 모니터링 모드를 `proactive`로 설정합니다.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:proactive
   ```

2. **전용 연결을 설정하기 전에 디렉토리 프록시 서버에서 데이터 소스 작업이 없는지 감지하는 최대 시간을 설정합니다.**

   ```
   $ dpconf set-ldap-data-source-prop -h host -p port datasource \
    monitoring-inactivity-timeout:time
   ```

   기본적으로 비활성 시간 초과는 120초입니다.

3 **474 페이지 "디렉토리 프록시 서버에 대한 관리 경고 구성"에 설명된 것처럼 데이터 소스가 오프라인 또는 온라인 상태로 감지된 경우 경고를 보내도록 구성합니다.**

▼ 설정된 연결을 테스트하여 데이터 소스를 모니터링하는 방법

이 모니터링 유형에서 디렉토리 프록시 서버는 각 데이터 소스에 대한 연결에서 검색을 정기적으로 수행합니다. 이 방법으로 디렉토리 프록시 서버는 닫힌 연결을 감지하고 비활성으로 인해 연결이 끊어지지 않도록 방지합니다.

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 **데이터 소스에 대한 모니터링 모드를 `proactive`로 설정합니다.**

```
$ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-mode:proactive
```

2 **디렉토리 프록시 서버에서 수행되는 모니터링 검색 요청을 구성합니다.**

```
$ dpconf set-ldap-data-source-prop -h host -p port datasource \
  monitoring-bind-timeout:timeout monitoring-entry-dn:dn \
  monitoring-search-filter:filter monitoring-entry-timeout:timeout
```

검색 요청에서 다음 등록 정보가 사용됩니다.

`monitoring-bind-timeout`	데이터 소스에 대한 연결이 설정될 때까지 디렉토리 프록시 서버가 대기하는 시간. 기본적으로 이 등록 정보의 값은 5초입니다.
`monitoring-entry-dn`	검색 요청에서 대상 항목의 DN. 기본적으로 이 등록 정보는 루트 DSE 항목입니다("").
`monitoring-search-filter`	검색 필터.
`monitoring-entry-timeout`	디렉토리 프록시 서버가 검색 응답을 위해 대기하는 시간. 기본적으로 이 등록 정보의 값은 5초입니다.

3 **폴링 간격을 설정합니다.**

```
$ dpconf set-ldap-data-source-prop -h host -p port datasource monitoring-interval:interval
```

연결이 끊어진 경우 디렉토리 프록시 서버는 이 간격마다 연결을 폴링하여 복구를 감지합니다. 기본적으로 모니터링 간격은 30초입니다.

4 **474 페이지 "디렉토리 프록시 서버에 대한 관리 경고 구성"에 설명된 것처럼 데이터 소스가 오프라인 또는 온라인 상태로 감지된 경우 경고를 보내도록 구성합니다.**

디렉토리 프록시 서버에 대한 관리 경고 구성

관리 경고를 구성하는 방법에 대한 자세한 내용은 다음 절차를 참조하십시오.

▼ 관리 경고를 활성화하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1 활성화된 경고를 봅니다.

```
% dpconf get-server-prop -h host -p port enabled-admin-alerts
```

2 하나 이상의 관리 경고를 활성화합니다.

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts:alert1 \
  [enabled-admin-alerts:alert2 ...]
```

예를 들어 사용 가능한 모든 경고를 활성화하려면 다음 명령을 실행합니다.

```
% dpconf set-server-prop -h host -p port \
 enabled-admin-alerts:error-configuration-reload-failure-with-impact \
 enabled-admin-alerts:error-server-shutdown-abrupt \
 enabled-admin-alerts:info-configuration-reload \
 enabled-admin-alerts:info-data-source-available \
 enabled-admin-alerts:info-server-shutdown-clean \
 enabled-admin-alerts:info-server-startup \
 enabled-admin-alerts:warning-configuration-reload-failure-no-impact \
 enabled-admin-alerts:warning-data-source-unavailable \
 enabled-admin-alerts:warning-data-sources-inconsistent \
 enabled-admin-alerts:warning-listener-unavailable
```

모든 경고를 비활성화하려면 다음 명령을 실행합니다.

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts:none
```

활성화된 경고의 기존 목록에 경고를 추가하려면 이 명령을 실행합니다.

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts+:alert-name
```

활성화된 경고의 기존 목록에서 경고를 제거하려면 이 명령을 실행합니다.

```
% dpconf set-server-prop -h host -p port enabled-admin-alerts-:alert-name
```

기본적으로 경고가 활성화되지 않습니다.

참조 자세한 내용은 `enabled-admin-alerts`(5dpconf)를 참조하십시오.

▼ Syslog에 보내도록 관리 경고를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **474 페이지 "관리 경고를 활성화하는 방법"에 설명된 것처럼 `syslog` 데몬에 보낼 경고를 선택합니다.**

2. **`syslog` 데몬에 보내도록 경고를 활성화합니다.**

   ```
   $ dpconf set-server-prop -h host -p port syslog-alerts-enabled:true
   ```

   USER 기능으로 모든 경고가 `syslog`에 보내집니다.

3. **경고를 보낼 `syslog` 데몬의 호스트 이름을 설정합니다.**

   ```
   $ dpconf set-server-prop -h host -p port syslog_hostname:hostname
   ```

▼ 전자 메일로 보내도록 관리 경고를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **474 페이지 "관리 경고를 활성화하는 방법"에 설명된 것처럼 `syslog`에 보낼 경고를 선택합니다.**

2. **전자 메일의 주소와 특성을 구성합니다.**

   ```
   $ dpconf set-server-prop -h host -p port email-alerts-smtp-host:host-name \
     email-alerts-smtp-port:port-number \
     email-alerts-message-from-address:sender-email-address \
     email-alerts-message-to-address:receiver-email-address \
     [email-alerts-message-to-address:receiver-email-address ...] \
     email-alerts-message-subject:email-subject
   ```

3. **전자 메일로 보내도록 경고를 활성화합니다.**

   ```
   $ dpconf set-server-prop -h host -p port email-alerts-enabled:true
   ```

4. **(옵션) 전자 메일에 경고 코드를 포함하도록 플래그를 설정합니다.**

   ```
   $ dpconf set-server-prop -h host -p port \
    email-alerts-message-subject-includes-alert-code:true
   ```

▼ 스크립트를 실행하도록 관리 경고를 구성하는 방법

DSCC를 사용하여 이 작업을 수행할 수 있습니다. 자세한 내용은 43 페이지 "디렉토리 서비스 제어 센터 인터페이스" 및 DSCC 온라인 도움말을 참조하십시오.

1. **474 페이지 "관리 경고를 활성화하는 방법"에 설명된 것처럼 `syslog`에 보낼 경고를 선택합니다.**

2. **스크립트를 실행하도록 경고를 활성화합니다.**

   ```
   $ dpconf set-server-prop -h host -p port scriptable-alerts-enabled:true
   ```

3. **실행할 스크립트의 이름을 설정합니다.**

   ```
   $ dpconf set-server-prop -h host -p port scriptable-alerts-command:script-name
   ```

JVM을 사용하여 디렉토리 프록시 서버에 대한 모니터링된 데이터 검색

디렉토리 프록시 서버는 JVM(Java Virtual Machine) 내에서 실행되고 JVM 시스템의 메모리에 따라 달라집니다. 디렉토리 프록시 서버가 올바르게 실행되고 있는지 확인하려면 JVM 시스템의 메모리 사용을 모니터링해야 합니다.

JVM 시스템의 매개 변수를 조정하는 방법에 대한 자세한 내용은 **Sun Java System Directory Server Enterprise Edition 6.2 Deployment Planning Guide**의 "Hardware Sizing For Directory Proxy Server"를 참조하십시오.

기본적으로 JVM 시스템의 힙 크기는 250MB입니다. 디렉토리 프록시 서버에 물리적 메모리가 충분하지 않은 경우 힙 크기는 250MB보다 작을 수 있습니다.

디렉토리 프록시 서버가 실행 중인 경우 JVM 시스템의 힙 크기를 모니터링하여 메모리가 부족해지지 않도록 할 수 있습니다. 이렇게 하려면 JDK(Java Development Kit)와 함께 제공되는 표준 도구를 사용합니다. 이러한 도구는 `$JAVA_HOME/bin/jps` 및 `$JAVA_HOME/bin/jstat` 디렉토리에 있습니다.

▼ JVM의 힙 크기를 보는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

- **JVM의 힙 크기를 봅니다.**

  ```
  $ dpadm get-flags instance-path jvm-args
  jvm-args: -Xms250M  -Xmx250M
  ```

▼ 디렉토리 프록시 서버가 실행 중인 경우 JVM의 힙 크기를 모니터링하는 방법

DSCC를 사용하여 이 작업을 수행할 수 없습니다. 이 절차에 설명된 것처럼 명령줄을 사용하십시오.

1 디렉토리 프록시 서버 인스턴스의 PID를 봅니다.

```
$ jps
```

2 JVM 시스템에 사용된 메모리를 봅니다.

```
$ jstat -gcutil PID
```

- 0 열이 거의 100%에 도달하면 JVM 시스템에 메모리가 부족한 것입니다.
- FGC는 전체 가비지 컬렉션(GC) 이벤트 수입니다. 가비지 컬렉션은 광범위합니다.
- GCT(Garbage Collection Time)는 GC에 사용된 시간입니다.

색인

E

G

I

J

K

L

M

N

R

S

T

U

V

가

객

검

결