Sun Java™ System

Web Proxy Server 4 .0.1
관리 설명서

2005Q4

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

부품 번호 : 819-3161

목차

설명서 정보

이 설명서는 Sun Java™ System Web Proxy Server 4(구 Sun™ ONE Web Proxy Server 및 iPlanet™ Web Proxy Server) 를 구성하고 관리하는 방법에 대해 설명합니다 (이하 Sun Java System Web Proxy Server 또는 Proxy Server 로 표기).

이 서문의 내용 :

설명서 사용 대상

이 설명서는 프로덕션 환경의 정보 기술 관리자를 대상으로 합니다 . 설명서는 사용자가 다음 영역에 익숙하다고 가정합니다 .

- 기본 시스템 관리 작업 수행
- 소프트웨어 설치
- 웹 브라우저 사용
- 단말기 창에서의 명령 실행

본 설명서의 구성

이 설명서는 여러 부로 나뉘어 있으며 각 부에서는 특정 영역 및 작업에 대해 설명합니다 . 설명서의 각 부 및 해당 내용은 아래 표의 목록을 참조하십시오 .

표 1 설명서 구성

부	설명
1 부 서버 기본	Proxy Server 및 Proxy Server 관리에 대한 개요를 제공합니다 . • 제 1 장 , “Sun Java System Web Proxy Server 소개 ” • 제 2 장 , “Sun Java System Web Proxy Server 관리 ”
2 부 Administration Server 사용	Administration Server 기본 설정 구성 , 사용자 및 그룹 관리 , Proxy Server 보안 , 서버 간 구성 공유를 위한 클러스터 사용에 대한 세부 사항을 설명합니다 . • 제 3 장 , “ 관리 기본 설정 지정 ” • 제 4 장 , “ 사용자 및 그룹 관리 ” • 제 5 장 , “ 인증서 및 키 사용 ” • 제 6 장 , “ 서버 클러스터 관리 ”
3 부 Proxy Server 구성 및 모니터링	서버 기본 설정 구성 , 액세스 제어 설정 및 서버 동작 모니터링에 대한 세부 사항을 설명합니다 . • 제 7 장 , “ 서버 기본 설정 구성 ” • 제 8 장 , “ 서버 액세스 제어 ” • 제 9 장 , “ 로그 파일 사용 ” • 제 10 장 , “ 서버 모니터 ”
4 부 Proxy Server 관리	Proxy Server 에서 요청을 처리하는 방법과 관련된 개념 및 작업에 대한 세부 사항을 설명합니다 . • 제 11 장 , “URL 프록시 및 라우팅 ” • 제 12 장 , “ 캐시 ” • 제 13 장 , “ 프록시를 통한 컨텐트 필터링 ” • 제 14 장 , “ 역방향 프록시 사용 ” • 제 15 장 , “SOCKS 사용 ” • 제 16 장 , “ 템플릿 및 리소스 관리 ” • 제 17 장 , “ 클라이언트 자동 구성 파일 사용 ”

표 1 설명서 구성

부	설명
5 부 부록	ACL(Access Control List) 파일 구문 및 서버 성능 조정에 대해 설명합니다 . • 부록 A, "ACL 파일 구문 " • 부록 B, " 서버 성능 조정 "

문서 규약

이 설명서에 사용된 문서 규약은 아래 표의 목록을 참조하십시오 .

표 2 문서 규약

요소	사용
파일 및 디렉토리 경로	UNIX 형식 (디렉토리 이름 사이를 슬래시 (/) 로 구분) 이 사용됩니다 .
설치 루트 디렉토리	*server_root* 로 표기합니다 . 기본 설치 디렉토리는 /proxyserver4 입니다 .
기울임꼴 텍스트	강조 , 용어 , 경로 내의 환경 변수 , 자리 표시자
monospace 텍스트	예제 코드 , 파일 이름 , 경로 이름 , 명령 이름 , 프로그래밍 언어 키워드 , 등록 정보
AaBbCc123	책 제목 , 장 , 절

관련 문서

Sun Java System Web Proxy Server 4 문서는 HTML 과 PDF 형식으로 다음 사이트에서 받을 수 있습니다 .

http://docs.sun.com/app/docs/prod/s1.webproxys#hic 혹은
http://docs.sun.com/app/docs/prod/s1.webproxys?l=ko#hic

각 설명서에 설명된 작업과 개념은 다음 표와 같습니다 .

표 3 Proxy Server 문서

내용	참조 문서
Proxy Server 릴리스 • 소프트웨어 및 설명서에 대한 최신 정보 • 새로운 기능 • 지원 플랫폼 및 환경 • 시스템 요구 사항 • 알려진 문제 및 해결 방법	릴리스 노트
설치 및 마이그레이션 작업 수행 : • Sun Java System Web Proxy Server 설치 • 버전 3.6 에서 버전 4 로 마이그레이션	Installation and Migration Guide
관리 작업 수행 : • 관리 및 명령줄 인터페이스 사용 • 서버 기본 설정 구성 • 사용자 및 그룹 관리 • 서버 작동 모니터링 및 로깅 • 인증서 및 공용 키 암호화를 사용하여 서버 보안 • 서버 액세스 제어 • URL 프록시 및 라우팅 • 캐시 • 내용 필터링 • 역방향 프록시 사용 • SOCKS 사용	관리자 설명서 (및 제품에 포함된 온라인 도움말)
사용자 정의 NSAPI(Netscape Server Application Programmer's Interface) 플러그인 만들기	NSAPI Developer’s Guide
구성 파일 편집	Configuration File Reference

Sun 기술 지원

제품에 포함된 문서에서 관련 내용을 찾을 수 없는 기술적인 질문은 다음 사이트를 참조하십시오 .

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

피드백

Sun 은 설명서의 향상에 최선을 다하고 있으며 귀사의 의견 및 제안을 환영합니다 . 의견을 보내려면 http://docs.sun.com 으로 이동하여 의견 보내기 링크를 누릅니다 . 온라인 양식에 문서 제목 및 부 번호를 입력하여 주십시오 . 사용자 의견을 제출할 때 해당 양식에 영문 설명서 제목과 부품 번호를 입력해야 할 수도 있습니다 . 본 설명서의 영문 부품 번호와 제목은 819-3650, Sun Java System Web Proxy Server 4.0.1 2005Q4 Administration Guide 입니다 .

타사 웹 사이트 참조

Sun 은 이 문서에 언급된 타사 웹 사이트의 가용성에 대해 책임지지 않습니다 . Sun 은 이러한 사이트 또는 리소스에 있거나 또는 이를 통하여 접할 수 있는 내용 , 광고 , 제품 또는 기타 자료를 보증하지 않으며 책임 또는 법적 의무를 지지 않습니다 . Sun 은 이러한 사이트 또는 리소스에 있거나 또는 이를 통하여 접할 수 있는 내용 , 상품 또는 서비스의 사용이나 의존 및 이와 관련한 것으로 인한 실질 또는 추정 손상이나 손실에 대한 책임 또는 법적 의무를 지지 않습니다 .

제 I 부

서버의 기본

제 1 장 , "Sun Java System Web Proxy Server 소개 "

제 2 장 , "Sun Java System Web Proxy Server 관리 "

1 장

Sun Java System Web Proxy Server 소개

이 장에서는 이번 릴리스의 새로운 기능에 대한 간단한 설명 및 Proxy Server 를 관리 , 구성하는 데 사용하는 웹 기반 사용자 인터페이스에 대한 개요를 포함한 Sun Java™ System Web Proxy Server 의 일반적인 개요를 제공합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

Sun Java System Web Proxy Server 정보

Sun Java System Web Proxy Server 는 고성능 인터넷 및 인트라넷 환경을 위한 HTTP 캐시 및 가속 기반입니다 . Proxy Server 는 웹 컨텐트를 캐시 및 필터링하고 , 네트워크 성능을 높이며 전체 네트워크 인프라와 교차 플랫폼 지원 , 중앙 집중식 관리 기능을 밀접하게 통합해 주는 강력한 시스템입니다 . 네트워크 트래픽 관리 프로그램 역할을 하며 정보를 효율적으로 분산 및 관리함으로써 네트워크 트래픽과 사용자 대기 시간을 줄였습니다 . Proxy Server 는 또한 컨텐트 분산을 위한 보안 게이트웨이를 제공하고 인터넷 트래픽에 대한 제어 지점 역할을 함으로써 사용자가 네트워크 리소스에 안전하고 생산적으로 액세스할 수 있도록 합니다 .

이 릴리스의 새로운 기능

Sun Java System Web Proxy Server 4 의 개선 사항은 다음과 같습니다 .

- 최신의 HTTP 코어
- Linux 및 Solaris™ x86 플랫폼에 대한 지원
- 모든 플랫폼에서 최신 SSL(Secure Sockets Layer) 지원
- 모든 플랫폼에서 다중 스레드 아키텍처
- 향상된 관리 , 그래픽 사용자 인터페이스 및 관리의 용이함
- 새로운 NSAPI(Netscape Server Application Programmer's Interface) 필터
- 강화된 LDAP(Lightweight Directory Access Protocol) 성능
- 향상된 확장성 및 성능
- 향상된 컨텐트 필터링
- `server.xml` 구성 파일의 구현

새로운 기능 및 개선 사항에 대한 자세한 내용은 다음 Proxy Server 릴리스 노트를 참조하십시오 .
http://docs.sun.com/app/docs/prod/s1.webproxys#hic

시작

Sun Java System Web Proxy Server 는 브라우저로 액세스되는 웹 기반 사용자 인터페이스인 Administration Server 및 Server Manager 를 사용하여 관리 및 구성됩니다 . Administration Server 는 시스템에 설치된 모든 Proxy Server 인스턴스에 대한 구성 관리에 사용되며 , Server Manager 는 개별 서버 인스턴스 설정 구성에 사용됩니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- Administration Server 개요
- Server Manager 개요
- 구성 파일
- 정규식

참고 서버를 구성하는 데 필요한 CGI 프로그램을 실행하려면 브라우저에서 쿠키를 사용할 수 있도록 설정해야 합니다 .

Administration Server 개요

Administration Server 는 시스템에 설치된 모든 Sun Java System Web Proxy Server 에 대한 구성을 관리하는 데 사용되는 웹 기반 사용자 인터페이스입니다 .

Administration Server 를 시작한 다음 ("Administration Server 시작 " (31 페이지) 참조) 브라우저를 실행하고 URL 을 입력하여 Administration Server 에 액세스할 수 있습니다 . URL 은 설치 시 지정한 호스트 이름과 포트 번호에 따라 달라집니다 . 예 : `http://myserver.mycorp.com:1234`

Administration Server 로의 접근 권한은 한 명 이상의 관리자에게 주어질 수 있습니다 . 분산 관리에 대한 더 자세한 내용은 " 복수 관리자 허용 " (40 페이지) 을 참조하십시오 .

Administration Server 에 액세스하려면 다음을 수행합니다 .

1. 브라우저를 실행하고 설치 시에 Administration Server 에 대해 지정한 호스트 이름과 포트 번호를 나타내는 URL 을 입력합니다 . 예 : `http://myserver.mycorp.com:1234`
2. 메시지가 표시되면 설치 시 지정한 아이디와 비밀 번호를 입력합니다 . Administration Server 사용자 인터페이스가 표시됩니다 .

Administration Server 설정은 특정 작업에 해당하는 탭으로 구성됩니다 . 아래 표에는 Administration Server 탭 및 각 탭의 용도에 대한 간단한 설명이 나와 있습니다 .

표 1-1 Administration Server 탭

탭	용도
Servers	Proxy Server 관리 , 추가 , 제거 , 마이그레이션
Preferences	Administration Server 종료 , 청취 소켓 편집 , 수퍼유저 액세스 구성 , 분산 관리 구성 (복수 관리자 허용), 액세스 로그와 오류 로그 사용자 정의 및 확인
Global Settings	디렉토리 서비스 구성 , 액세스 제어 지정 , SNMP 마스터 에이전트 설정 구성
Users and Groups	사용자 , 그룹 , 조직 단위 추가 및 관리
Security	새 신뢰 데이터베이스 생성 , VeriSign 및 기타 인증서 요청 및 설치 , 키 쌍 파일 비밀 번호 변경 , 설치된 인증서 확인 및 관리 , CRL(Certificate Revocation Lists) 과 CKL(Compromised Key Lists) 추가 및 대체 , CRL 및 CKL 관리 , 3.x 인증서 마이그레이션
Cluster	클러스터의 원격 서버 제어 , 원격 서버 추가 및 제거 , 서버 정보 수정

다음 버튼은 탭 또는 페이지에 관계없이 표시됩니다 .

- **Version** - Sun Java System Web Proxy Server 버전 정보 표시
- **Refresh** - 현재 페이지 새로 고침
- **Help** - 현재 페이지에 대한 온라인 도움말 표시

Administration Server 사용에 대한 자세한 내용은 제 2 장 , 31 페이지의 "Sun Java System Web Proxy Server 관리 " 를 참조하십시오 . 또한 Administration Server 탭 및 페이지에 대한 온라인 도움말을 참조하십시오 .

Server Manager 개요

Server Manager 는 Sun Java System Web Proxy Server 의 개별 인스턴스를 시작 , 정지 및 구성하는 데 사용되는 웹 기반 사용자 인터페이스입니다 .

Server Manager 에 액세스하려면 다음을 수행합니다 .

1. "Administration Server 개요 " (27 페이지) 에서 설명한 대로 Administration Server 에 액세스합니다 . Administration Server 에 Servers 탭이 표시됩니다 .
2. Manage Servers 페이지에서 관리하려는 서버 인스턴스에 대한 링크를 누릅니다 . Server Manager 사용자 인터페이스가 표시됩니다 .

Server Manager 설정은 특정 작업에 해당하는 탭으로 구성됩니다 . 아래 표에는 Server Manager 탭 및 각 탭의 용도에 대한 간단한 설명이 나와 있습니다 .

표 1-2 Server Manager 탭

탭	용도
Preferences	서버 시작 및 정지 , 서버 설정 확인 , 구성 정보 복원 , 시스템 기본 설정 구성 , Proxy Server 성능 조정 , 청취 소켓 추가 및 편집 , MIME 유형 관리 , 액세스 제어 관리 , ACL 및 DNS 캐시 구성 , DNS 로컬 하위 도메인 구성 , HTTP 연결 유지 설정 구성 , 암호 크기 설정
Routing	프록시 사용 여부 설정 , 라우팅 기본 설정 지정 , 클라이언트 인증서 전달 , Java IP 주소 확인 사용 , 자동 구성 파일 생성 및 편집 , 연결 모드 설정 , 기본 FTP 전송 모드 변경 , SOCKS 이름 서버 IP 주소 설정 , HTTP 요청 로드 밸런싱 구성
SOCKS	SOCKS 서버의 시작 및 정지 , SOCKS 인증 , 연결 , 라우팅 항목의 생성 및 관리
URL	URL 매핑 및 리디렉션 확인 , 생성 , 관리

표 1-2 Server Manager 탭

탭	용도
Caching	캐시 특성 설정 , 캐시 파티션 추가 및 수정 , 기존 파티션 간 구역 이동 , 캐시 용량 설정 , 가비지 수집 모드 설정 , 캐시 조정 , 가비지 수집 일정 예약 , 가비지 수집 설정 조정 , 특정 리소스에 대한 캐시 구성 , 로컬 호스트 캐시 사용 , 파일 캐시 설정 변경 , 캐시 일괄 업데이트 설정 , 캐시된 URL 기록에 대한 정보 확인 , ICP 이웃에서 프록시 구성 , 프록시 배열 구성원 목록 생성 및 업데이트 , 프록시 배열 구성원 구성 , PAT 파일의 정보 확인
Filters	필터 파일 생성 , 컨텐트 URL 재작성 설정 , user-agent 제한 및 요청 차단 설정 , 송신 헤더 제거 , MIME 필터 및 HTML 태그 필터 설정 , 필요 시 내용 압축
Server Status	로그 파일 확인 , 로그 보관 , 로그 기본 설정 지정 , 보고서 생성 , 현재 동작 모니터 , SNMP 하위 에이전트 구성 및 제어
Security	새 신뢰 데이터베이스 생성 , VeriSign 및 기타 인증서 요청 및 설치 , 키 쌍 파일 비밀 번호 변경 , 설치된 인증서 확인 및 관리 , CRL(Certificate Revocation Lists) 과 CKL(Compromised Key Lists) 추가 및 대체 , CRL 과 CKL 관리 , 3.x 인증서 마이그레이션
Templates	템플릿 생성 , 제거 , 적용 , 확인 및 리소스 제거

다음 버튼은 탭 또는 페이지에 관계없이 표시됩니다 .

- **Version** - Sun Java System Web Proxy Server 버전 정보 표시
- **Refresh** - 현재 페이지 새로 고침
- **Help** - 현재 페이지에 대한 온라인 도움말 표시

Refresh 버튼 아래에 Restart Required 링크가 표시되는 경우도 있습니다 . 이는 변경 사항이 발생해 서버를 다시 시작해야 함을 나타냅니다 . 변경 사항을 적용하려면 이 링크를 누르고 원하는 작업을 지정합니다 .

Server Manager 사용에 대한 자세한 내용은 이 설명서에서 관련 작업을 참조하십시오 . 또한 Server Manager 탭 및 페이지에 대한 온라인 도움말을 참조하십시오 .

구성 파일

Sun Java System Web Proxy Server 의 구성 및 작동은 구성 파일 설정에 따라 달라집니다 . 관리 인터페이스에서 구성된 설정은 구성 파일에 반영됩니다 . 파일은 수동으로도 편집할 수 있습니다 .

구성 파일은 *instance_dir*/config 디렉토리에 있습니다 . 여기에서 instance_dir 은 서버 인스턴스입니다 . config 디렉토리에는 여러 구성 요소를 제어하는 다양한 구성 파일이 포함되어 있습니다 . 구성 파일의 정확한 수와 이름은 사용하도록 설정되거나 로드된 구성 요소에 따라 다릅니다 . 서버 작업에 필수적인 4 가지 구성 파일은 항상 이 디렉토리에 포함됩니다 . 아래 표에 4 가지 필수 구성 파일 및 이 파일의 내용이 나와 있습니다 .

표 1-3 필수 구성 파일

파일	포함된 내용
server.xml	서버 구성의 대부분 (이 Proxy Server 릴리스의 새로운 기능)
magnus.conf	전역 서버 초기화 정보
obj.conf	클라이언트의 요청에 대한 처리 지시문
mime.types	요청된 리소스의 컨텐트 유형을 결정하기 위한 정보

이러한 구성 파일 및 기타 구성 파일에 대한 자세한 내용은 Proxy Server Configuration File Reference 를 참조하십시오 .

정규식

정규식은 리소스를 확인하고 Proxy Server 를 구성하여 여러 URL 의 요청을 다르게 처리하도록 하는 데 사용됩니다 . 정규식은 Administration Server 및 Server Manager 사용자 인터페이스를 사용하여 다양한 작업을 수행하면서 지정할 수 있습니다 . 정규식 사용에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 를 참조하십시오 .

2 장

Sun Java System Web Proxy Server 관리

이 장에서는 Administration Server 를 사용한 Sun Java System Web Proxy Server 관리의 기본을 소개합니다 . Administration Server 는 서버 관리 , 추가 , 제거 및 마이그레이션에 사용되는 웹 기반 사용자 인터페이스입니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

- Administration Server 시작
- Administration Server 정지
- 복수 Proxy Server 실행
- 서버 인스턴스 제거
- Proxy Server 3.6 에서 마이그레이션

Administration Server 기본 설정 구성에 대한 자세한 내용은 제 3 장 , 37 페이지의 " 관리 기본 설정 지정 " 을 참조하십시오 . 서버 클러스터를 사용하여 여러 Proxy Server 를 관리하는 방법에 대한 자세한 내용은 제 6 장 , 119 페이지의 " 서버 클러스터 관리 " 를 참조하십시오 .

Administration Server 시작

이 절에서는 다양한 플랫폼에서 Administration Server 를 시작하는 방법에 대해 설명합니다 . Administration Server 를 정지하는 방법에 대한 자세한 내용은 "Administration Server 정지 " (32 페이지) 를 참조하십시오 .

UNIX 또는 Linux 에서 Administration Server 를 시작하려면 다음을 수행합니다 .

- 명령줄에서 *server_root*/proxy-admserv 로 이동한 다음 ./start 를 입력하여 Administration Server 를 시작합니다 (Administration Server 를 재시작하려면 ./restart 입력).

Windows 에서 Administration Server 를 시작하려면 다음을 수행합니다 .

- 시작 > 프로그램 > Sun Microsystems > Sun Java System Web Proxy Server *버전* > Start Admin 을 사용합니다 .

  또는

  제어판 > 관리 도구 > 서비스 > Sun Java System Web Proxy Server 4.0 > 에서 시작을 누릅니다 .

  또는

- 명령 프롬프트에서 *server_root*\proxy-admserv 로 이동한 다음 startsvr.bat 를 입력하여 Administration Server 를 시작합니다 (Administration Server 를 재시작하려면 ./restart 입력).

Administration Server 가 시작되면 브라우저를 실행하고 설치 시 Administration Server 에 대해 지정한 호스트 이름과 포트 번호를 나타내는 URL 을 입력하여 액세스할 수 있습니다 (예 : http://myserver.mycorp.com:1234). 설치 시 지정한 아이디와 비밀 번호를 입력하라는 프롬프트가 표시됩니다 .

Administration Server 로의 접근 권한은 한 명 이상의 관리자에게 주어질 수 있습니다 . 분산 관리에 대한 더 자세한 내용은 " 복수 관리자 허용 " (40 페이지) 을 참조하십시오 .

Administration Server 정지

이 절에서는 다양한 플랫폼에서 Administration Server 를 정지하는 방법에 대해 설명합니다 . Administration Server 를 시작하는 방법에 대한 자세한 내용은 "Administration Server 시작 " (31 페이지) 을 참조하십시오 .

UNIX 또는 Linux 에서 Administration Server 를 정지하려면 다음을 수행합니다 .

- Administration Server 에 액세스하여 Preferences 탭을 누르고 Shutdown Server 링크를 누른 다음 OK 를 누릅니다 .

  또는

- 명령줄에서 *server_root*/proxy-admserv/ 로 이동하여 ./stop 을 입력합니다 .

Windows 에서 Administration Server 를 정지하려면 다음을 수행합니다 .

- 제어판 > 관리 도구 > 서비스 창에서 Sun Java System Proxy Server 4.0 Administration Server 서비스를 사용합니다 .

  또는

- 명령 프롬프트에서 *server_root*\proxy-admserv 로 이동하여 stopsvr.bat 를 입력합니다 .

복수 Proxy Server 실행

시스템에서 복수 Proxy Server 를 실행하려면 복수 서버 인스턴스를 설치 및 구성해야 합니다 . 다음 절차에서는 서버 인스턴스를 추가하는 방법에 대해 설명합니다 .

복수 서버 인스턴스를 설치하려면 다음을 수행합니다 .

1. Administration Server 로 액세스합니다 .
2. Servers 탭에서 Add Server 를 누릅니다 .
3. 필요한 정보를 입력하고 OK 를 누릅니다 . 특정 필드에 대한 자세한 내용은 온라인 도움말을 참조하십시오 .
4. 새 서버 인스턴스가 추가된 다음 표시되는 Success 페이지에서 원하는 경우 Configure Your New Server 링크를 누릅니다 . 서버 인스턴스 구성에 사용되는 Server Manager 인터페이스가 표시됩니다 .

서버 인스턴스 제거

Administration Server 를 사용하여 Proxy Server 인스턴스를 제거할 수 있습니다 . 이 작업을 실행 취소할 수 없으므로 다음 절차를 수행하기 전에 제거하려는 서버 인스턴스를 확인하십시오 .

서버 인스턴스를 제거하려면 다음을 수행합니다 .

1. Administration Server 로 액세스합니다 .
2. Servers 탭에서 Remove Server 를 누릅니다 .
3. 드롭다운 목록에서 제거하려는 서버 인스턴스를 선택합니다 .
4. 제거하려면 Confirming Server Removal 확인란을 선택하고 OK 를 누릅니다 .

Proxy Server 3.6 에서 마이그레이션

Sun™ One Web Proxy Server 3.6(또는 iPlanet™ Web Proxy Server) 을 Sun Java System Web Proxy Server 4 로 마이그레이션할 수 있습니다 . 3.6 서버는 보존되며 새로운 버전 4 서버가 동일한 설정으로 생성됩니다 . 버전 3.6 에서 4.0 으로의 마이그레이션에 대한 자세한 내용은 Proxy Server Installation and Migration Guide 를 참조하십시오 . 또한 Proxy Server 사용자 인터페이스의 마이그레이션 관련 페이지에 대한 자세한 내용은 온라인 도움말을 참조하십시오 . 인증서 마이그레이션에 대한 자세한 내용은 이 설명서의 " 인증서 마이그레이션 " (84 페이지) 을 참조하십시오 .

제 2 부

Administration Server 사용

3 장

관리 기본 설정 지정

이 장에서는 Administration Server 를 사용하여 관리 기본 설정을 구성하는 방법에 대해 설명합니다 . 서버를 구성하는 데 필요한 CGI 프로그램을 실행하려면 브라우저가 쿠키를 사용하도록 설정해야 합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

청취 소켓 만들기 및 관리

서버가 요청을 처리하려면 먼저 청취 소켓에서 해당 요청을 허용한 다음 이를 올바른 서버로 보내야 합니다 . Proxy Server 를 설치하는 경우 청취 소켓 한 개 (ls1) 가 자동으로 만들어집니다 . 이 청취 소켓은 IP 주소 0.0.0.0 과 설치 도중 Administration Server 포트 번호로 지정된 포트 번호를 사용합니다 .

Administration Server 의 Edit Listen Sockets 페이지를 사용하여 청취 소켓을 추가 , 편집 및 삭제할 수 있습니다 . 서버에 액세스하기 위한 청취 소켓이 한 개 이상 있어야 합니다 . 목록에 있는 유일한 청취 소켓일 경우에는 삭제할 수 없습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

청취 소켓 추가

청취 소켓을 추가하려면 다음을 수행합니다 .

1. Administration Server 에 액세스하고 Preferences 탭을 누릅니다 .
2. Edit Listen Sockets 링크를 누릅니다 .
3. New 버튼을 누릅니다 .
4. 설정을 지정하고 OK 를 누릅니다 . 특정 필드에 대한 자세한 내용은 온라인 도움말을 참조하십시오 .

청취 소켓 편집

청취 소켓을 편집하려면 다음을 수행합니다 .

1. Administration Server 에 액세스하고 Preferences 탭을 누릅니다 .
2. Edit Listen Sockets 링크를 누릅니다 .
3. 편집하려는 청취 소켓에 대한 링크를 누르고 원하는 사항을 변경한 다음 OK 를 누릅니다 .

청취 소켓 삭제

청취 소켓을 삭제하려면 다음을 수행합니다 .

1. Administration Server 에 액세스하고 Preferences 탭을 누릅니다 .
2. Edit Listen Sockets 링크를 누릅니다 .
3. 삭제하려는 청취 소켓 옆에 있는 확인란을 선택하고 OK 를 누릅니다 . 삭제를 확인하는 메시지가 표시됩니다 . 서버에 액세스하기 위한 청취 소켓이 한 개 이상 있어야 합니다 . 목록에 있는 유일한 청취 소켓일 경우에는 삭제할 수 없습니다 .

수퍼유저 설정 변경

Administration Server 에 대한 수퍼유저 액세스를 구성할 수 있습니다 . 이 설정은 오직 수퍼유저 계정에만 적용됩니다 . Administration Server 가 분산 관리를 사용하는 경우 허용된 관리자에 대해 추가 액세스 제어를 구성해야 합니다 .

주의 Sun Java™ System Directory Server 를 사용하여 사용자 및 그룹을 관리하는 경우 수퍼유저 아이디나 비밀 번호를 변경하기 *전에* 해당 디렉토리에서 수퍼유저 항목을 업데이트해야 합니다 . 디렉토리를 먼저 업데이트하지 않으면 Administration Server 의 Users and Groups 인터페이스에 액세스할 수 없습니다 . 이 문제를 해결하려면 디렉토리에 대한 액세스 권한이 있는 관리자 계정으로 Administration Server 에 액세스하거나 Directory Server 의 콘솔 또는 구성 파일을 사용하여 디렉토리를 업데이트합니다 .

Administration Server 에 대한 수퍼유저 설정을 변경하려면 다음을 수행합니다 .

1. Administration Server 에 액세스하고 Preferences 탭을 누릅니다 .
2. Control Superuser Access 링크를 누릅니다 .
3. 원하는 사항을 변경한 다음 OK 를 누릅니다 . 특정 필드에 대한 자세한 내용은 온라인 도움말을 참조하십시오 .

수퍼유저의 아이디와 비밀 번호는 *server_root*/proxy-admserv/config 에 있는 admpw 파일에 저장됩니다 . 파일의 형식은 *username*:*password* 입니다 . 이 파일에서 아이디는 확인할 수 있지만 비밀 번호는 암호화되어 읽을 수 없습니다 . 비밀 번호를 잊은 경우에는 admpw 파일을 편집하여 암호화된 비밀 번호를 삭제합니다 . 그 후 다음을 수행합니다 .

1. 비밀 번호 없이 아이디로만 Administration Server 에 액세스합니다 .
2. Preferences 탭을 누릅니다 .
3. Control Superuser Access 링크를 누릅니다 .
4. 새 비밀 번호를 입력하고 OK 를 누릅니다 .

주의 admpw 파일을 편집할 수 있으므로 서버 컴퓨터를 안전한 위치에 두고 이 서버의 파일 시스템에 대한 액세스를 제한하는 것이 중요합니다 .

UNIX 및 Linux 시스템의 경우 파일의 소유권을 변경하여 루트나 Administration Server 데몬을 실행하는 시스템 사용자만 쓸 수 있도록 합니다 . Windows 시스템의 경우 파일의 소유권을 Administration Server 가 사용하는 사용자 계정으로 제한합니다 .

복수 관리자 허용

여러 관리자가 분산 관리를 통하여 서버의 특정한 부분을 변경할 수 있습니다 . 분산 관리를 사용하려면 먼저 디렉토리 서버를 설치해야 합니다 . 기본 디렉토리 서비스는 LDAP 기반이어야 합니다 .

분산 관리에는 수퍼유저와 관리자 , 두 가지 수준의 사용자가 있습니다 .

- 수퍼유저는 *server_root*/proxy-admserv/config/admpw 의 목록에 있는 사용자입니다 . 이는 설치 시 지정한 아이디와 비밀 번호입니다 . 이 사용자는 Users and Groups 양식을 제외한 Administration Server 의 모든 양식에 액세스할 수 있습니다 . Users 및 Groups 양식에 대한 액세스는 수퍼유저가 LDAP 서버에 유효한 계정이 있어야 합니다 .
- 관리자는 Administration Server 를 포함하여 특정 서버의 Server Manager 양식으로 직접 이동할 수 있습니다 . 표시되는 양식은 관리자용으로 구성된 액세스 제어 규칙 (보통 수퍼유저가 설정) 에 따라 다릅니다 . 관리자는 제한된 관리 작업을 수행하며 사용자 추가나 액세스 제어 변경 등 다른 사용자에게 영향을 미치는 사항을 변경할 수 있습니다 .

액세스 제어에 대한 자세한 내용은 제 8 장 , 147 페이지의 " 서버 액세스 제어 " 를 참조하십시오 .

분산 관리를 사용하려면 다음을 수행합니다 .

1. 디렉토리 서버가 설치되어 있는지 확인합니다 .
2. Administration Server 로 액세스합니다 .
3. 디렉토리 서버를 설치한 다음 관리자 그룹을 만들지 않은 경우 만들어야 합니다 . 그룹을 만들려면 다음을 수행합니다 .
   a. Users and Groups 탭을 누릅니다 .
   b. Create Group 링크를 누릅니다 .

c. LDAP 디렉토리에 관리자 그룹을 만들고 Administration Server 또는 해당 서버 루트에 설치된 서버의 구성 권한을 부여하려는 사용자의 이름을 추가합니다 . 특정 필드에 대한 자세한 내용은 온라인 도움말을 참조하십시오 .

   관리자 그룹의 모든 사용자는 Administration Server 전체에 액세스할 수 있으나 액세스 제어를 사용하여 구성할 수 있는 서버 및 양식을 제한할 수 있습니다 .

   액세스 제어 목록을 만들면 분산 관리자 그룹이 목록에 추가됩니다 . 관리자 그룹의 이름을 변경하는 경우 직접 액세스 제어 목록을 편집하여 제어가 참조하는 그룹을 변경해야 합니다 .

4. Preferences 탭을 누릅니다 .
5. Configure Distributed Administration 링크를 누릅니다 .
6. Yes 를 선택하고 관리자 그룹을 지정한 다음 OK 를 누릅니다 .

로그 파일 옵션 지정

Administration Server 로그 파일은 발생한 오류의 유형 및 서버 액세스에 대한 정보를 포함하여 Administration Server 에 대한 데이터를 기록합니다 . 이 로그 파일을 통하여 서버 동작을 모니터하고 문제를 해결할 수 있습니다 . Log Preferences 페이지의 많은 옵션을 사용하여 Administration Server 로그에 기록되는 데이터의 유형과 형식을 지정합니다 . 서버에 대한 고정된 양의 정보를 제공하는 Common Logfile Format 을 선택하거나 또는 사용자의 요구 사항에 맞추어 로그 파일 형식을 사용자 정의할 수 있습니다 .

Administration Server Log Preferences 페이지에 액세스하려면 Preferences 탭을 누른 다음 Set Access Log Preferences 또는 Set Error Log Preferences 링크를 누릅니다 . 로그 파일 및 로그 파일 옵션 설정에 대한 자세한 내용은 제 9 장 , 179 페이지의 " 로그 파일 사용 " 및 온라인 도움말을 참조하십시오 .

로그 파일 확인

Administration Server 로그 파일은 *server_root*/proxy-admserv/logs 에 있습니다 . Proxy Server 관리 콘솔 또는 텍스트 편집기를 통하여 오류 및 액세스 로그를 모두 확인할 수 있습니다 .

액세스 로그 파일

액세스 로그 파일에는 서버로 오고 가는 요청에 대한 정보가 기록됩니다 .

액세스 로그 파일을 확인하려면 다음을 수행합니다 .

1. Administration Server 에 액세스하고 Preferences 탭을 누릅니다 .
2. View Access Log 링크를 누릅니다 .

특정 필드에 대한 자세한 내용은 온라인 도움말 및 제 9 장 , 179 페이지의 " 로그 파일 사용 " 을 참조하십시오 .

오류 로그 파일

오류 로그에는 로그 파일이 생성된 후 서버에 발생한 모든 오류가 나열됩니다 . 또한 서버가 시작된 시간과 서버에 로그인을 시도했으나 실패한 사용자 등 서버에 대한 정보 메시지가 포함되어 있습니다 .

오류 로그 파일을 확인하려면 다음을 수행합니다 .

1. Administration Server 에 액세스하고 Preferences 탭을 누릅니다 .
2. View Error Log 링크를 누릅니다 .

특정 필드에 대한 자세한 내용은 온라인 도움말 및 제 9 장 , 179 페이지의 " 로그 파일 사용 " 을 참조하십시오 .

디렉토리 서비스 사용

LDAP 를 사용하여 단일 디렉토리 서버에서 아이디 및 비밀 번호와 같은 정보를 저장하고 관리할 수 있습니다 . 또한 사용자가 쉽게 액세스할 수 있는 여러 네트워크 위치에서 디렉토리 정보를 검색할 수 있도록 서버를 구성할 수 있습니다 . 디렉토리 서비스 사용에 대한 자세한 내용은 제 4 장 , 45 페이지의 " 사용자 및 그룹 관리 " 를 참조하십시오 .

서버 액세스 제한

Proxy Server 는 수신 요청을 평가할 때 ACE(Access-control Entries) 라고 하는 규칙 계층에 따라 액세스를 결정한 다음 일치되는 항목을 사용하여 요청의 허가 여부를 결정합니다 . 각 ACE 는 서버가 계층의 다음 ACE 로 계속할 것인지의 여부를 지정합니다 . ACE 의 컬렉션을 ACL(Access-control List) 이라고 합니다 .

Administration Server 및 파일 , 디렉토리 , 파일 유형과 같은 서버 인스턴스 내의 특정 리소스에 대해 액세스 제어를 구성할 수 있습니다 . Administration Server 에 대한 액세스 제어는 Administration Server 의 Global Settings 탭에서 구성합니다 . 서버 인스턴스 내의 리소스에 대한 액세스 제어는 Server Manager 의 Preferences 탭에서 구성합니다 . 액세스 제어 설정에 대한 자세한 내용은 제 8 장 , 147 페이지의 " 서버 액세스 제어 " 를 참조하십시오 .

참고 서버 액세스를 제한하려면 먼저 분산 관리를 사용하도록 설정해야 합니다 . 자세한 내용은 " 복수 관리자 허용 " (40 페이지) 을 참조하십시오

SNMP 마스터 에이전트 설정

SNMP(Simple Network Management Protocol) 는 네트워크 활동을 위한 데이터 교환에 사용하는 프로토콜입니다 . 이 정보는 하위 에이전트 및 마스터 에이전트의 사용을 통하여 네트워크 관리 스테이션과 서버 사이에 전송됩니다 .

SNMP 마스터 에이전트 설정은 Administration Server 의 Global Settings 탭을 사용하여 구성합니다 . 마스터 에이전트는 Administration Server 에 설치됩니다 . SNMP 및 에이전트 설정에 대한 자세한 내용은 제 10 장 , 205 페이지의 " 서버 모니터 " 를 참조하십시오 . 또한 Administration Server 의 Global Settings 탭에 있는 마스터 에이전트 페이지와 Server Manager 의 Server Status 탭에 있는 하위 에이전트 페이지에 대해서는 온라인 도움말을 참조하십시오 .

4 장

사용자 및 그룹 관리

이 장에서는 Proxy Server 에 액세스할 수 있는 사용자 및 그룹을 추가 , 삭제 , 수정 및 관리하는 방법에 대해 설명합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

사용자 및 그룹에 대한 정보 액세스

Administration Server 에서 사용자 계정 , 그룹 목록 , 액세스 권한 , 조직 단위 및 기타 사용자 및 그룹 특정 정보에 대한 응용 프로그램 데이터에 액세스할 수 있습니다 .

사용자 및 그룹 정보는 보통 파일 형식이나 Sun Java™ System Directory Server 와 같이 LDAP(Lightweight Directory Access Protocol) 를 지원하는 디렉토리 서버 안에 저장됩니다 . LDAP 는 개방형 디렉토리 액세스 프로토콜로 TCP/IP(Transmission Control Protocol/Internet Protocol) 에서 실행되며 전세계적 규모의 수 백만 항목을 수용하도록 확장될 수 있습니다 .

디렉토리 서비스 정보

디렉토리 서비스는 한 소스에서 모든 사용자 정보를 관리할 수 있도록 합니다. Proxy Server 를 통해 LDAP, 키 파일 및 다이제스트 파일 등 , 세 가지 유형의 디렉토리 서비스를 구성할 수 있습니다 .

다른 디렉토리 서비스가 구성되지 않은 경우 디렉토리 서비스를 새로 만들면 , 이는 유형에 상관 없이 `default` 값으로 설정됩니다 . 디렉토리 서비스를 만들면 *server_root*`/userdb/dbswitch.conf` 파일의 디렉토리 서비스 세부 사항이 업데이트됩니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

LDAP 디렉토리 서비스

LDAP 디렉토리 서비스에서 사용자 및 그룹 정보는 LDAP 기반 디렉토리 서비스에 저장됩니다 .

LDAP 서비스가 기본 서비스인 경우 `dbswitch.conf` 파일이 아래의 예와 같이 업데이트됩니다 .

```
directory default
ldap://test22.india.sun.com:589/dc%3Dindia%2Cdc%3Dsun%2Cdc%3Dcom
default:binddn cn=Directory Manager
default:encoded bindpw YWRtaW5hZG1pbg==
```

LDAP 서비스가 기본 서비스가 아닌 경우 `dbswitch.conf` 파일이 아래의 예와 같이 업데이트됩니다 .

```
directory ldap
ldap://test22.india.sun.com:589/dc%3Dindia%2Cdc%3Dsun%2Cdc%3Dcom
ldap:binddn cn=Directory Manager
ldap:encoded bindpw YWRtaW5hZG1pbg==
```

키 파일 디렉토리 서비스

키 파일은 해시 형식의 사용자 비밀 번호와 사용자가 속한 그룹 목록이 포함된 텍스트 파일입니다 . 키 파일 형식은 HTTP 기본 인증을 사용할 때만 사용할 수 있습니다 . 이 인증 방법에 대한 자세한 내용은 " 사용자 및 그룹 지정 " (164 페이지) 을 참조하십시오 .

키 파일 기반 데이터베이스를 만들면 `dbswitch.conf` 파일이 다음 예와 같이 업데이트됩니다 .

```
directory keyfile 파일
keyfile:syntax keyfile
keyfile:keyfile D:\test22\keyfile\keyfiledb
```

다이제스트 파일 디렉토리 서비스

다이제스트 파일은 암호화된 사용자 이름 및 비밀 번호에 기반하여 사용자 및 그룹 정보를 저장합니다 .

다이제스트 파일 형식은 HTTP 다이제스트 인증 지원을 뜻하지만 기본 인증도 지원하므로 두 인증 방법에 모두 사용할 수 있습니다 . 이러한 두 가지 방법에 대한 자세한 내용은 " 사용자 및 그룹 지정 " (164 페이지) 을 참조하십시오 .

다이제스트 기반 데이터베이스를 만들면 `dbswitch.conf` 파일이 다음 예와 같이 업데이트됩니다 .

```
directory digest file
digest:syntax digest
digest:digestfile D:\test22\digest\digestdb
```

참고 분산 관리를 구성하려면 LDAP 기반 디렉토리 서비스가 기본 디렉토리 서비스여야 합니다 .

디렉토리 서비스 구성

Administration Server 의 Global Settings 탭에서 디렉토리 서비스를 만들고 구성합니다 . 그런 다음 Administration Server 의 Users and Groups 탭에서 사용자 , 그룹 및 조직 단위를 만들고 관리합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 디렉토리 서비스 만들기
- 디렉토리 서비스 편집

디렉토리 서비스 만들기

디렉토리 서비스를 만들려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Global Settings 탭을 누릅니다 .
2. Configure Directory Service 링크를 누릅니다 .
3. Create New Service of Type 드롭다운 목록에서 만들려는 디렉토리 서비스 유형을 선택하고 New 를 누릅니다 . 해당 디렉토리 서비스에 대한 구성 페이지가 표시됩니다 .
4. 구성 정보를 입력하고 Save Changes 를 누릅니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

참고 다른 디렉토리 서비스가 구성되지 않은 경우 디렉토리 서비스를 새로 만들면 유형에 상관 없이 `default` 값으로 설정됩니다 .

디렉토리 서비스 편집

디렉토리 서비스를 편집하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Global Settings 탭을 누릅니다 .
2. Configure Directory Service 링크를 누릅니다 .
3. 편집할 디렉토리 서비스의 링크를 누르고 필요한 사항을 변경한 후 Save Changes 를 누릅니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

DN(Distinguished Name) 에 대한 이해

사용자 , 그룹 및 조직 단위를 만들거나 수정하려면 Administration Server 의 Users and Groups 탭을 사용합니다 . 사용자는 회사 직원과 같이 LDAP 데이터베이스에 있는 개인입니다 . 그룹은 공통 속성을 공유하는 둘 이상의 사용자입니다 . 조직 단위는 회사 내의 하위 부서로 `organizationalUnit` 개체 클래스를 사용합니다 . 사용자 , 그룹 및 조직 단위는 이 장의 뒷부분에서 자세히 설명합니다 .

기업의 각 사용자와 그룹은 DN(Distinguished Name) 속성으로 구분됩니다 . DN 속성은 연결된 사용자 , 그룹 또는 개체에 대한 구분 정보가 있는 문자열입니다 . 사용자 또는 그룹 디렉토리 항목이 변경될 때마다 DN 을 사용합니다 . 예를 들어 디렉토리 항목을 변경하고 , 액세스 제어를 구성하고 , 메일이나 게시 같은 응용 프로그램용 사용자 계정을 구성할 때마다 DN 정보를 입력해야 합니다 . Proxy Server 의 Users and Groups 인터페이스를 통해 DN 을 만들거나 수정합니다 .

다음은 전형적인 Sun Microsystems 의 직원용 DN 의 예입니다 .

```
uid=doe,e=doe@sun.com,cn=John Doe,o=Sun Microsystems Inc.,c=US
```

이 예에서 사용하는 약어의 의미는 다음과 같습니다 .

- `uid` 는 사용자 아이디 (user ID) 입니다 .
- `e` 는 전자 메일 (e-mail) 주소입니다 .
- `cn` 은 사용자의 공통 이름 (common name) 입니다 .
- `o` 는 조직 (organization) 입니다 .
- `c` 는 국가 (country) 입니다 .

DN 은 다양한 이름 - 값 쌍을 포함할 수 있으며 LDAP 를 지원하는 디렉토리의 인증서 주체 및 항목을 식별하는 데 사용됩니다 .

LDIF 사용

현재 디렉토리가 없으나 기존 디렉토리에 새 하위 트리를 추가하려는 경우 디렉토리 서버의 의 LDIF(Lightweight Directory Interchange Format) 가져오기 기능을 사용할 수 있습니다 . 이 기능은 LDIF 가 포함된 파일을 받아서 LDIF 항목에서 디렉토리를 구축하거나 새 하위 트리를 만듭니다 . 또한 디렉토리 서버의 LDIF 내보내기 기능을 사용하여 현재 디렉토리를 LDIF 로 내보낼 수 있습니다 . 이 기능은 디렉토리에 대한 LDIF 형식 파일을 만듭니다 . 가능한 경우 `ldapmodify` 명령줄 유틸리티와 적절한 LDIF 업데이트 문을 사용하여 항목을 추가하거나 편집할 수 있습니다 .

LDIF 를 사용하여 데이터베이스에 항목을 추가하려면 우선 LDIF 파일에 항목을 정의한 후 디렉토리 서버에서 LDIF 파일을 가져옵니다 .

사용자 만들기

사용자 항목을 만들고 수정하려면 Administration Server 의 Users and Groups 탭을 사용합니다 . 사용자 항목에는 데이터베이스의 개인 또는 개체에 대한 정보가 포함됩니다 .

참고 리소스에 대한 권한 없는 액세스를 차단하여 서버 보안을 지키도록 합니다 . Proxy Server 는 ACL 기반 인증과 인증 모델을 사용합니다 . ACL 기반 보안에 대한 자세한 내용은 제 8 장 , 147 페이지의 " 서버 액세스 제어 " 를 참조하십시오 . 추가 보안 정보는 제 5 장 , 75 페이지의 " 인증서 및 키 사용 " 을 참조하십시오 .

이 절에서는 다음 항목에 대해 설명합니다 .

- LDAP 기반 인증 데이터베이스에서 사용자 만들기
- 키 파일 인증 데이터베이스에 사용자 만들기
- 다이제스트 파일 인증 데이터베이스에 사용자 만들기

LDAP 기반 인증 데이터베이스에서 사용자 만들기

LDAP 기반 디렉토리 서비스에 사용자 항목을 추가하는 경우 기본 LDAP 기반 디렉토리 서버의 서비스가 사용자를 인증하고 권한을 부여하는 데 사용됩니다 . 이 절에서는 LDAP 기반 인증 데이터베이스를 사용할 때 고려할 지침을 제시하고 Proxy Server Administration Server 를 통해 사용자를 추가하는 방법에 대해 설명합니다 .

LDAP 기반 사용자 항목 만들기 지침

Proxy Server 관리 콘솔을 사용하여 LDAP 기반 디렉토리 서비스에 새 사용자 항목을 만드는 경우 다음의 지침을 고려하십시오 .

- 이름과 성을 입력할 경우 사용자의 전체 이름과 사용자 아이디가 자동 입력됩니다 . 사용자 아이디는 사용자 이름의 첫 자와 사용자 성을 조합하여 만듭니다 . 예를 들어 사용자의 이름이 Billie Holiday 인 경우 사용자 아이디는 자동으로 `bholiday` 가 됩니다 . 원하는 경우 이 사용자 아이디는 원하는 아이디로 바꿀 수 있습니다 .
- 사용자 아이디는 반드시 고유해야 합니다. Administration Server는 검색 기반(기본 DN) 에서 시작하여 전체 디렉토리에서 해당 사용자 아이디가 사용되는지 검색하여 해당 사용자 아이디가 고유한지 확인합니다 . 그러나 `ldapmodify` 명령줄 유틸리티 (사용 가능한 경우) 를 통해 사용자를 만들 경우 해당 사용자 아이디가 고유하다고는 보장할 수 없습니다 . 디렉토리의 사용자 아이디가 중복되는 경우 관련 사용자는 디렉토리에 대해 인증되지 않습니다 .
- 기본 DN 은 디렉토리 검색이 기본으로 수행될 위치의 고유 이름과 디렉토리 트리에서 모든 Proxy Server Administration Server 항목이 위치할 고유 이름을 지정합니다 . DN 은 디렉토리 서버에 있는 항목 이름을 문자열로 나타낸 것입니다 .
- 새 사용자 항목을 만들 때는 최소한 다음의 사용자 정보를 지정해야 합니다 .
  - 성
  - 전체 이름
  - 사용자 아이디
- 디렉토리에 대해 조직 단위가 정의된 경우 Administration Server 의 Create User 페이지에 있는 Add New User To 목록을 사용하여 신규 사용자를 추가할 위치를 지정할 수 있습니다 . 기본 위치는 디렉토리의 기본 DN(또는 루트 지점) 입니다 .

LDAP 기반 사용자 항목 만들기

사용자 항목을 만들려면 "LDAP 기반 사용자 항목 만들기 지침 " (51 페이지) 에 있는 지침을 읽고 다음 절차를 수행합니다 .

LDAP 기반 인증 데이터베이스에서 사용자를 만들려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Create User 링크를 누릅니다 .
3. 드롭다운 목록에서 LDAP 디렉토리 서비스를 선택하고 Select 를 누릅니다 .

4. 표시되는 페이지에 해당 정보를 입력합니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 . 또한 " 디렉토리 서버 사용자 항목 " (52 페이지) 도 참조할 수 있습니다 .
5. 사용자 항목을 만들려면 Create 를 누르고 사용자 항목을 만든 다음 해당 항목에 대한 편집 페이지로 이동하려면 Create and Edit 를 누릅니다 .

디렉토리 서버 사용자 항목

디렉토리 서버 사용자 항목 참고 사항 :

- 사용자 항목은 `inetOrgPerson`, `organizationalPerson` 및 `person` 개체 클래스를 사용합니다 .
- 기본적으로 사용자용 고유 이름의 형식은 다음과 같습니다 .

  `cn=` *전체 이름* `,ou=` *조직* `,...,o=` *기본 조직* `,c=` *국가*

  예를 들어 Billie Holiday 에 대한 사용자 항목이 조직 단위 Marketing 안에 만들어졌으며 디렉토리의 기본 DN 이 `o=Ace Industry`, `c=US` 인 경우 이 사용자의 DN 은 다음과 같습니다 .

  ```
  cn=Billie Holiday,ou=Marketing,o=Ace Industry,c=US
  ```

  그러나 이 형식은 사용자 아이디 (uid) 기반의 고유한 이름으로 변경할 수 있습니다 .
- 사용자 양식 필드의 값은 LDAP 속성으로 저장됩니다 .

  다음 표는 Proxy Server 인터페이스에 새 사용자를 만들 때 표시하는 필드와 해당하는 LDAP 속성을 설명합니다 .

표 4-1 LDAP 속성 - 새 항목 만들기

사용자 필드	LDAP 속성
Given Name	`givenName`
Surname	`sn`
Full Name	`cn`
User ID	`uid`
Password	`userPassword`
E-mail Address	`mail`

다음 표는 사용자 항목을 편집할 때 함께 표시하는 필드와 해당 LDAP 속성을 설명합니다 .

표 4-2 LDAP 속성 - 사용자 항목 편집

사용자 필드	LDAP 속성
Title	`title`
Phone Number	`telephoneNumber`

키 파일 인증 데이터베이스에 사용자 만들기

키 파일은 해시 형식의 사용자 비밀 번호와 사용자가 속한 그룹 목록이 포함된 텍스트 파일입니다 .

키 파일 인증 데이터베이스에서 사용자를 만들려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Create User 링크를 누릅니다 .
3. 드롭다운 목록에서 키 파일 기반 디렉토리 서비스를 선택하고 Select 를 누릅니다 .
4. 표시되는 페이지에 해당 정보를 입력하고 Create User 를 누릅니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

다이제스트 파일 인증 데이터베이스에 사용자 만들기

다이제스트 파일 인증 데이터베이스는 암호화된 양식으로 사용자 및 그룹 정보를 저장합니다 .

다이제스트 파일 인증 데이터베이스에서 사용자를 만들려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Create User 링크를 누릅니다 .
3. 드롭다운 목록에서 다이제스트 파일 기반 디렉토리 서비스를 선택하고 Select 를 누릅니다 .
4. 표시되는 페이지에 해당 정보를 입력하고 Create User 를 누릅니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

참고 Proxy Server ACL 사용자 인터페이스를 사용하여 다이제스트 인증을 사용하는 ACL 을 만드는 경우 동일한 영역 문자열을 지정해야 합니다 . 자세한 내용은 " 액세스 제어 설정 " (159 페이지) 을 참조하십시오 .

사용자 관리

사용자 속성은 Administration Server 의 Users and Groups 탭에 있는 Manage Users 페이지에서 편집합니다 . 이 페이지에서 사용자 항목을 검색 , 변경 , 삭제하거나 이름을 변경할 수 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

사용자 정보 찾기

사용자 항목을 편집하기 전에 , 다음 절차에 따라 항목을 찾아 표시해야 합니다 .

사용자 정보를 찾으려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Users 링크를 누릅니다 .
3. 드롭다운 목록에서 디렉토리 서비스를 선택하고 Select 를 누릅니다 . 키 파일 또는 다이제스트 파일 디렉토리 서비스에 대한 사용자 목록이 표시됩니다 . LDAP 기반 디렉토리 서비스에 대한 검색 필드가 표시됩니다 .

4. 사용자 정보를 찾습니다 .

   키 파일 또는 다이제스트 파일 디렉토리 서비스의 경우 해당 사용자에 대한 링크를 눌러 편집 페이지를 표시하고 변경합니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

   LDAP 기반 디렉토리 서비스의 경우 다음을 수행합니다 .

   a. Find User 필드에 편집할 항목에 대한 기술적인 값을 입력합니다 . 다음과 같은 항목을 입력할 수 있습니다 .

      - 이름 . 전체 또는 이름의 일부를 입력합니다 . 검색 문자열과 일치하는 모든 항목이 검색됩니다 . 해당 항목을 찾을 수 없는 경우 검색 문자열을 포함하는 모든 항목이 검색됩니다 . 해당 항목을 찾을 수 없는 경우 검색 문자열과 발음이 비슷한 모든 항목이 검색됩니다 .
      - 사용자 아이디 . 사용자 아이디의 일부만 입력하면 문자열을 포함하는 모든 항목이 검색됩니다 .
      - 전화번호 . 번호를 부분적으로 입력하면 검색 번호로 끝나는 전화번호를 포함하는 모든 항목이 검색됩니다 .
      - 전자 메일 주소 . @ 기호를 포함하는 모든 검색 문자열은 전자 메일 주소인 것으로 가정합니다 . 정확히 일치하는 검색 결과가 없을 경우 검색 문자열로 시작하는 모든 전자 메일 주소를 찾습니다 .
      - 디렉토리에 있는 모든 항목을 보려면 별표 (*) 를 사용합니다 . 필드에 아무런 값을 입력하지 않아도 동일한 결과를 얻을 수 있습니다 .
      - 모든 LDAP 검색 필터 . 등호 (=) 가 포함된 문자열은 검색 필터로 간주됩니다 .

      다른 방법으로 Find All Users Whose 부분의 드롭다운 메뉴를 사용하여 검색 범위를 좁힐 수 있습니다 . 자세한 내용은 " 사용자 정의 검색 쿼리 구축 " (56 페이지) 을 참조하십시오 .

   b. Look Within 필드에서 검색하려는 항목의 상위 조직 단위를 선택합니다 . 기본값은 디렉토리의 루트 지점 (또는 최상단 항목) 입니다 .

   c. 결과를 화면에 표시할지 , 프린터에서 출력할지를 드롭다운 목록에서 선택합니다 .

   d. 이 프로세스의 아무 단계에서나 Find 버튼을 누릅니다 . 검색 기준과 일치하는 모든 사용자가 표시됩니다 .

   e. 표시할 항목에 대한 링크를 누릅니다 .

사용자 정의 검색 쿼리 구축

LDAP 서비스의 경우 Find All Users Whose 부분을 사용하면 사용자 정의 검색 필터를 만들 수 있습니다 . 이 필드를 사용하면 Find User 검색으로 반환되는 검색 결과를 좁힐 수 있습니다 .

왼쪽의 드롭다운 목록에서 검색 기준으로 사용할 속성을 지정할 수 있습니다 . 다음 표는 사용 가능한 검색 속성 옵션을 설명합니다 .

표 4-3 검색 속성 옵션

옵션	일치 검색
Full name	각 항목의 전체 이름
Last name	각 항목의 성
User ID	각 항목의 사용자 아이디
Phone number	각 항목의 전화번호
E-mail address	각 항목의 전자 메일 주소

가운데 드롭다운 목록에서 검색의 유형을 지정합니다 . 다음 표는 사용 가능한 검색 유형 옵션을 설명합니다 .

표 4-4 검색 유형 옵션

옵션	설명
Contains	하위 문자열 검색이 수행되도록 합니다 . 지정한 검색 문자열을 포함하는 속성 값을 가진 항목이 반환됩니다 . 예를 들어 사용자 이름에 "Dylan" 이란 단어가 포함된 것을 알고 있는 경우 이 옵션을 검색 문자열 "Dylan" 과 함께 사용하여 해당 사용자 항목을 찾을 수 있습니다 .
Is	정확하게 일치하는 항목을 검색합니다 (동등 검색 지정). 사용자 속성의 값을 정확히 아는 경우 이 옵션을 사용합니다 . 예를 들어 사용자 이름의 정확한 철자를 아는 경우입니다 .
Isn't	검색 문자열과 정확히 일치하지 않는 속성값을 가진 모든 항목을 검색합니다 . 디렉토리에서 이름이 "John Smith" 가 아닌 모든 사용자를 찾을 때 이 옵션을 사용합니다 . 그러나 이 옵션을 사용하면 지나치게 많은 항목이 검색될 수 있으므로 주의해야 합니다 .
Sounds like	근사치 또는 발음에 의한 검색이 수행됩니다 . 속성의 값은 알지만 철자법이 확실하지 않은 경우 이 옵션을 사용합니다 . 예를 들어 사용자의 이름이 "Sarret", "Sarette" 또는 "Sarett" 인지 확실하지 않은 경우입니다 .

표 4-4 검색 유형 옵션

옵션	설명
Starts with	하위 문자열 검색이 수행되도록 합니다 . 지정된 검색 문자열로 시작하는 속성 값을 가진 모든 항목을 검색합니다 . 예를 들어 사용자 이름이 "Miles" 로 시작되지만 나머지 이름은 알지 못하는 경우 이 옵션을 사용합니다 .
Ends with	하위 문자열 검색이 수행되도록 합니다 . 지정된 검색 문자열로 끝나는 속성 값을 가진 모든 항목을 검색합니다 . 예를 들어 사용자 이름이 "Dimaggio" 로 끝나지만 나머지 이름은 알지 못하는 경우 이 옵션을 사용합니다 .

오른쪽 텍스트 필드에 검색 문자열을 입력합니다 . Look within 필드에 지정된 디렉토리의 모든 사용자 항목을 표시하려면 별표 (*) 를 입력하거나 이 필드를 공란으로 남겨둡니다 .

사용자 정보 편집

사용자 항목을 편집하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Users 링크를 누릅니다 .
3. " 사용자 정보 찾기 " (54 페이지) 에서 설명한 것과 같이 사용자 항목을 표시합니다 .
4. 원하는 내용을 변경합니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

참고 사용자 편집 페이지에서 표시되지 않는 속성 값을 변경하고자 할 수 있습니다 . 이 때는 Directory Server `ldapmodify` 명령줄 유틸리티 (사용 가능한 경우) 를 사용하십시오 .

사용자 아이디 변경에 대한 자세한 내용은 " 사용자 이름 변경 " (58 페이지) 을 참조하십시오 .

사용자 비밀 번호 관리

다음 절차는 사용자 비밀 번호를 변경하거나 만드는 방법을 설명합니다 .

사용자 비밀 번호를 만들거나 변경하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Users 링크를 누릅니다 .
3. " 사용자 정보 찾기 " (54 페이지) 에서 설명한 것과 같이 사용자 항목을 표시합니다 .
4. 원하는 내용을 변경합니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

LDAP 데이터베이스의 경우 사용자 비밀 번호 정보를 편집하는 데 사용하는 페이지 (Manage Users 페이지에서 액세스) 에서 Disable Password 버튼을 눌러 사용자 비밀 번호를 사용하지 않도록 설정할 수 있습니다 . 이렇게 하면 사용자의 디렉토리 항목을 삭제할 필요 없이 해당 사용자의 서버 로그인을 방지할 수 있습니다 . 사용자가 다시 액세스할 수 있도록 하려면 새 비밀 번호를 입력합니다 .

사용자 이름 변경

LDAP 데이터베이스의 경우 이름 변경 기능은 사용자 아이디만 변경하며 다른 모든 필드는 그대로 유지됩니다 . 이름 변경 기능을 사용하여 항목을 하나의 조직 단위에서 다른 단위로 옮길 수는 없습니다 .

사용자 항목 이름을 변경하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Users 링크를 누릅니다 .
3. " 사용자 정보 찾기 " (54 페이지) 에서 설명한 것과 같이 사용자 항목을 표시합니다 .
4. 사용자 편집 페이지에서 Rename User 버튼을 누르고 , 표시되는 페이지에 사용자 아이디를 입력한 다음 Save Changes 를 누릅니다 .

참고

`keepOldValueWhenRenaming` 매개 변수를 기본값인 `false` 로 설정하면 , Administration Server 가 항목 이름 변경 시 기존 값을 보관하지 않도록 지정할 수 있습니다 . 이 매개 변수는 다음 파일에 있습니다 .

server_root`/proxy-admserv/config/dsgw-orgperson.conf`

사용자 제거

사용자 항목을 제거하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Users 링크를 누릅니다 .
3. " 사용자 정보 찾기 " (54 페이지) 에서 설명한 것과 같이 사용자 항목을 표시합니다 .
4. Delete User(LDAP) 또는 Remove User(키 파일 및 다이제스트 파일) 를 누릅니다 .

그룹 만들기

그룹은 LDAP 데이터베이스에 있는 일련의 개체를 기술하는 개체입니다 . Sun Java System 서버 그룹은 공통 속성을 공유하는 사용자로 구성됩니다 . 예를 들어 일련의 개체는 회사의 Marketing 부서에서 근무하는 다수의 직원일 수 있습니다 . 이들 직원들은 Marketing 이라는 이름의 그룹에 속할 수 있습니다 .

LDAP 서비스의 경우 정적 및 동적 등 두 가지 방법으로 그룹 구성원을 정의할 수 있습니다 . 정적 그룹은 구성원 개체를 명시적으로 열거합니다 . 정적 그룹은 공통 이름 (CN) 이며 `uniqueMembers` 및 / 또는 `memberURLs` 및 / 또는 `memberCertDescriptions` 를 포함합니다 . 정적 그룹의 경우 구성원은 `cn=`*groupname* 속성을 제외한 공통 속성을 공유하지 않습니다 .

동적 그룹을 사용하면 LDAP URL 을 사용하여 그룹 구성원에만 적용되는 일련의 규칙을 정의할 수 있습니다 . 동적 그룹의 경우 구성원은 공통 속성 또는 `memberURL` 필터에 정의된 일련의 속성을 공유합니다 . 예를 들어 이미 `ou=Sales,o=Airius.com` 아래 LDAP 데이터베이스에 있는 Sales 의 모든 직원을 포함하는 그룹이 필요하면 구성원 URL 이 다음과 같은 동적 그룹을 정의합니다 .

```
ldap:///ou=Sales,o=sun??sub?(uid=*)
```

이에 따라 그룹은 `ou=Sales,o=sun` 위치 아래 트리에 `uid` 속성을 가진 모든 개체를 포함하게 됩니다 . 즉 모든 Sales 구성원이 포함됩니다 .

정적 및 동적 그룹의 경우 `memberCertDescription` 을 사용하면 구성원이 인증서에 있는 공통 속성을 공유할 수 있습니다 . 참고로 이는 ACL 에서 SSL 메소드를 사용하는 경우에만 적용됩니다 .

새 그룹을 만든 후에는 사용자 (구성원) 을 해당 그룹에 추가할 수 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

정적 그룹 정보

LDAP 서비스의 경우 Administration Server 를 사용하면 사용자 수에 상관없이 DN 에서 동일한 그룹 속성을 지정하여 정적 그룹을 만들 수 있습니다 . 정적 그룹은 사용자를 추가하거나 제거하지 않는 한 변경되지 않습니다 .

정적 그룹 만들기 지침

Administration Server 인터페이스를 사용하여 새 정적 그룹을 만드는 경우 다음의 지침을 고려하십시오 .

- 정적 그룹에는 다른 정적 또는 동적 그룹이 포함될 수 있습니다 .
- 디렉토리에 대해 조직 단위가 정의된 경우 Administration Server 인터페이스의 Create Group 페이지에 있는 Add New User To 목록을 사용하여 신규 사용자를 추가할 위치를 지정합니다 . 기본 위치는 디렉토리의 루트 지점 (최상단 항목) 입니다 .
- 그룹 편집에 대한 자세한 내용은 " 그룹 항목 편집 " (66 페이지) 을 참조하십시오 .

정적 그룹 만들기

정적 그룹을 만들려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Create Group 링크를 누릅니다 .
3. Type of Group 드롭다운 목록에서 New Group 을 선택하고 Go 를 누릅니다 .
4. Create Group 페이지에 정보를 입력합니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .
5. 그룹을 만들려면 Create 를 누르고 그룹을 만든 다음 해당 그룹에 대한 편집 페이지로 이동하려면 Create and Edit 를 누릅니다 .

동적 그룹 정보

LDAP 서비스의 경우 Proxy Server 에서는 그룹 사용자가 자동으로 임의의 속성에 기반하도록 하거나 일치하는 DN 이 있는 특정 그룹에 ACL 을 적용하려는 경우 동적 그룹을 만들 수 있습니다 . 예를 들어 `department=marketing` 속성이 있는 DN 이 자동으로 포함되도록 그룹을 만들 수 있습니다 . `department=marketing` 에검색 필터를 적용하면 `department=marketing` 속성이 있는 모든 DN 을 포함하는 그룹이 검색됩니다 . 그 후 , 이 필터에 기반하여 검색 결과에서 동적 그룹을 정의할 수 있습니다 . 따라서 결과의 동적 그룹에 대한 ACL 을 정의할 수 있습니다 .

동적 그룹 구현 방법

Proxy Server 는 LDAP 서버 스키마에 동적 그룹을 `objectclass=groupOfURLs` 로 구현합니다 . `groupOfURLs` 클래스에는 여러 `memberURL` 속성이 있을 수 있으며 , 이 각각은 디렉토리의 개체 세트를 나열하는 LDAP URL 입니다 . 그룹의 구성원은 이 세트의 조합이 됩니다 . 예를 들어 다음 그룹은 오직 하나의 구성원 URL 만 포함합니다 .

```
ldap:///o=mcom.com??sub?(department=marketing)
```

이 예는 부서가 `marketing` 인 `o=mcom.com` 아래의 모든 개체로 구성되는 세트입니다 . LDAP URL 은 검색 기반 DN, 범위 및 필터 등을 포함하지만 호스트 이름과 포트는 포함하지 않습니다 . 따라서 동일한 LDAP 서버에 있는 개체만 참조할 수 있습니다 . 범위는 모두 지원됩니다 . LDAP URL 에 대한 자세한 내용은 " 동적 그룹 만들기 지침 " (62 페이지) 을 참조하십시오 .

DN 은 자동으로 포함되므로 직접 개인을 그룹에 추가할 필요가 없습니다 . ACL 검증을 위하여 그룹 조회가 필요할 때마다 Proxy Server 가 LDAP 서버 검색을 수행하므로 그룹은 동적으로 변경됩니다 . ACL 파일에서 사용된 사용자 및 그룹 이름은 LDAP 데이터베이스에 있는 개체의 `cn` 속성에 대응됩니다 .

주 Proxy Server 는 `cn` 속성을 ACL 용 그룹 이름으로 사용합니다 .

ACL 에서 LDAP 데이터베이스로의 매핑은 `dbswitch.conf` 파일 (실제 LDAP 데이터베이스 URL 로 ACL 데이터베이스 이름과 연결) 및 ACL 파일 (ACL 용으로 사용할 데이터베이스 정의) 모두에 정의됩니다 . 예를 들어 `staff` 라는 이름의 그룹의 구성원에게 기본 액세스 권한을 부여하려는 경우 ACL 코드는 개체 클래스가 `groupOf`*anything* 이며 CN 이 `staff` 로 설정된 개체를 조회합니다 . 개체는 구성원 ND 을 직접 나열 (정적 그룹용 `groupOfUniqueNames` 와 동일) 하거나 또는 LDAP URL 을 지정 (예 : `groupOfURLs`) 하여 그룹의 구성원을 정의합니다 .

주 그룹은 정적 및 동적이 될 수 있습니다 . 그룹 객체는 `objectclass=groupOfUniqueMembers` 및 `objectclass=groupOfURLs` 를 모두 가질 수 있습니다 . 따라서 `uniqueMember` 및 `memberURL` 속성은 모두 유효합니다 . 그룹의 구성원은 동적 및 정적 구성원의 조합입니다 .

서버 성능에 미치는 동적 그룹의 영향

동적 그룹을 사용하면 서버 성능에 영향을 미칩니다 . 그룹 구성원을 시험하며 DN 이 정적 그룹의 구성원이 아닌 경우 Proxy Server 는 데이터베이스의 기본 DN 에 있는 모든 동적 그룹을 확인합니다 . Proxy Server 는 기본 DN 과 사용자의 DN 을 확인하여 각 `memberURL` 이 일치하는지 결정한 후 사용자 DN 을 기본 DN 으로 사용하고 `memberURL` 필터를 통해 기본 검색을 수행합니다 . 이 절차에는 많은 개별 검색이 연관될 수 있습니다 .

동적 그룹 만들기 지침

Administration Server 인터페이스를 사용하여 새 동적 그룹을 만드는 경우 다음의 지침을 고려하십시오 .

- 동적 그룹에는 다른 그룹이 포함될 수 없습니다 .
- 다음 형식으로 그룹의 LDAP URL 을 입력합니다 (호스트 및 포트 정보는 무시되므로 생략).

  `ldap:///`*base_dn*`?`*attributes*`?`*scope*`?(`*filter*`)`

  LDAP URL 의 필수 매개 변수 목록은 다음 표와 같습니다 .

표 4-5 LDAP URL 필수 매개 변수

매개 변수 이름	설명
`base_dn`	검색 기반용 DN 또는 LDAP 디렉토리에서 검색이 수행되는 지점 . 간혹 이 매개 변수는 `o=mcom.com` 등의 디렉토리의 접미사 또는 루트로 설정됩니다 .
`attributes`	검색이 반환할 수 있는 속성 목록 . 여러 속성을 지정하려면 속성을 쉼표 (,) 로 구분합니다 (예 : `cn,mail,telephoneNumber`). 속성을 지정하지 않으면 모든 속성을 반환합니다 . 동적 그룹 구성원 확인의 경우 이 매개 변수는 무시됩니다 .

표 4-5 LDAP URL 필수 매개 변수

매개 변수 이름	설명
scope	이 매개 변수는 필수입니다 . 검색의 범위로 다음 중 한 가지 값을 가집니다 . • base는 해당 URL에 지정된 고유 이름 (base_dn) 에 대한 정보를 검색합니다 . • one 은 해당 URL 에 지정된 고유 이름 (base_dn) 보다 한 수준 아래의 항목에 대한 정보를 검색합니다 . 기본 항목은 이 범위에 포함되지 않습니다 . • sub 는 해당 URL 에 지정된 고유 이름 (base_dn) 보다 한 수준 아래의 항목에 대한 정보를 검색합니다 . 기본 항목은 이 범위에 포함되지 않습니다 .
(filter)	이 매개 변수는 필수입니다 . 검색의 지정된 범위 안에 있는 항목에 적용되는 검색 필터 . Administration Server 인터페이스를 사용하는 경우 반드시 이 속성을 지정해야 합니다 . 괄호는 필수입니다 .

attributes, scope 및 (filter) 매개 변수는 URL 에서의 위치에 따라 구분됩니다 . 속성을 지정하지 않으려는 경우에도 해당 필드에 물음표를 넣어 구분해야 합니다 .

동적 그룹 만들기 지침 계속

- 디렉토리에 대해 조직 단위가 정의된 경우 Administration Server 인터페이스의 Create Group 페이지에 있는 Add New User To 목록을 사용하여 신규 사용자를 추가할 위치를 지정합니다 . 기본 위치는 디렉토리의 루트 지점 (최상단 항목) 입니다 .
- 그룹 편집에 대한 자세한 내용은 " 그룹 항목 편집 " (66 페이지) 을 참조하십시오 .

동적 그룹 만들기

동적 그룹을 만들려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Create Group 링크를 누릅니다 .
3. Type of Group 드롭다운 목록에서 Dynamic Group 을 선택하고 Go 를 누릅니다 .
4. Create Group 페이지에 정보를 입력합니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

5. 그룹을 만들려면 Create 를 누르고 그룹을 만든 다음 해당 그룹에 대한 편집 페이지로 이동하려면 Create and Edit 를 누릅니다 .

그룹 관리

LDAP 서비스의 경우 Administration Server 에서 그룹을 편집하고 Administration Server 의 Users 및 Groups 탭에 있는 Manage Groups 에서 그룹 구성원을 관리할 수 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 그룹 항목 찾기
- 그룹 항목 편집
- 그룹 구성원 추가
- 그룹 구성원 목록에 그룹 추가
- 그룹 구성원 목록에서 항목 제거
- 소유자 관리
- 추가 참조 관리
- 그룹 이름 변경
- 그룹 제거

그룹 항목 찾기

그룹 항목을 편집하기 전에 다음 절차에 따라 항목을 찾아 표시해야 합니다 .

그룹 항목을 찾으려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Groups 링크를 누릅니다 .
3. Find Group 필드에 찾으려는 그룹 이름을 입력합니다 . 다음과 같은 항목을 입력할 수 있습니다 .
    - 이름 . 전체 또는 이름의 일부를 지정합니다 . 검색 문자열과 일치하는 모든 항목이 검색됩니다 . 해당 항목을 찾을 수 없는 경우 검색 문자열을 포함하는 모든 항목이 검색됩니다 . 해당 항목을 찾을 수 없는 경우 검색 문자열과 발음이 비슷한 모든 항목이 검색됩니다 .

- 현재 디렉토리에 있는 그룹을 모두 보려면 별표 (*) 를 사용합니다 . 필드에 아무런 값을 입력하지 않아도 동일한 결과를 얻을 수 있습니다 .
- 모든 LDAP 검색 필터 . 등호 (=) 가 있는 모든 문자열은 검색 필터로 간주됩니다 .

또는 Find All Groups Whose 부분을 사용하여 사용자 정의 검색 필터를 구축하고 검색 결과 범위를 좁힐 수 있습니다 . 자세한 내용은 "Find All Groups Whose 항목 " (65 페이지) 을 참조하십시오 .

4. Look within 필드에서 검색하려는 항목의 상위 조직 단위를 선택합니다 . 기본값은 디렉토리의 루트 지점 (또는 최상단 항목) 입니다 .
5. 결과를 화면에 표시할지 , 프린터에서 출력할지를 드롭다운 목록에서 선택합니다 .
6. 이 프로세스의 아무 단계에서나 Find 버튼을 누릅니다 . 검색 기준과 일치하는 모든 그룹이 표시됩니다 .
7. 표시할 항목에 대한 링크를 누릅니다 .

Find All Groups Whose 항목

LDAP 서비스의 경우 Find All Groups Whose 항목을 사용하면 사용자 정의 검색 필터를 만들 수 있습니다 . 이 항목의 필드를 사용하면 Find Group 에서 반환되는 검색 결과를 더욱 좁힐 수 있습니다 .

왼쪽의 드롭다운 목록에서 검색 기준으로 사용할 속성을 지정합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .

- **Name.** 각 항목의 전체 이름이 일치하도록 검색합니다 .
- **Description.** 각 그룹 항목의 설명이 일치되는지 검색합니다 .

가운데 드롭다운 목록에서 검색의 유형을 지정합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .

- **Contains.** 하위 문자열 검색이 수행되도록 합니다 . 지정한 검색 문자열을 포함하는 속성 값을 가진 항목이 반환됩니다 . 예를 들어 그룹 이름에 "Administrator" 라는 단어가 포함된 것을 알고 있는 경우 이 옵션을 검색 문자열 "Administrator" 와 함께 사용하여 해당 그룹 항목을 찾을 수 있습니다 .
- **Is.** 정확히 일치하는 항목을 검색합니다 . 그룹 속성의 값을 정확히 아는 경우 이 옵션을 사용합니다 . 예를 들어 그룹 이름의 정확한 철자를 아는 경우입니다 .

- **isn't**. 검색 문자열과 정확히 일치하지 않는 속성값을 가진 모든 항목을 검색합니다 . 디렉토리에서 이름에 "administrator" 가 포함되지 않는 모든 그룹을 찾으려면 이 옵션을 사용합니다 . 그러나 이 옵션을 사용하면 지나치게 많은 수의 항목이 검색될 수 있으므로 주의해야 합니다 .
- **Sounds like.** 근사치 또는 발음에 의한 검색이 수행됩니다 . 속성의 값은 알지만 철자법이 확실하지 않은 경우 이 옵션을 사용합니다 . 예를 들어 그룹의 이름이 "Sarret's list", "Sarette's list" 또는 "Sarett's list" 인지 확실하지 않은 경우입니다 .
- **Starts with.** 하위 문자열 검색이 수행되도록 합니다 . 지정된 검색 문자열로 시작하는 속성 값을 가진 모든 항목을 검색합니다 . 예를 들어 그룹 이름이 "Product" 로 시작되지만 나머지 이름은 알지 못하는 경우 이 옵션을 사용합니다 .
- **Ends with.** 하위 문자열 검색이 수행되도록 합니다 . 지정된 검색 문자열로 끝나는 속성 값을 가진 모든 항목을 검색합니다 . 예를 들어 그룹 이름이 "Product" 로 끝나지만 나머지 이름은 알지 못하는 경우 이 옵션을 사용합니다 .

오른쪽 텍스트 필드에 검색 문자열을 입력합니다 . 검색 위치 디렉토리에 포함된 모든 그룹 항목을 표시하려면 별표 (*) 를 입력하거나 이 필드를 공란으로 남겨둡니다 .

그룹 항목 편집

그룹 항목을 편집하려면 다음과 같이 합니다 .
이 절차는 LDAP 서비스에만 적용됩니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Groups 링크를 누릅니다 .
3. " 그룹 항목 찾기 " (64 페이지) 에서 설명한 대로 편집할 조직 단위를 찾습니다 .
4. 원하는 내용을 변경합니다 . 특정 필드와 버튼에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

참고 그룹 편집 페이지에서 표시되지 않는 속성 값을 변경하고자 할 수 있습니다 . 이 때는 디렉토리 서버 `ldapmodify` 명령줄 유틸리티 (사용 가능한 경우) 를 사용하십시오 .

그룹 구성원 추가

그룹에 구성원을 추가하려면 다음과 같이 합니다 .
이 절차는 LDAP 서비스에만 적용됩니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Groups 링크를 누릅니다 .
3. " 그룹 항목 찾기 " (64 페이지) 에 설명한 것과 같이 관리하려는 그룹을 찾아 표시한 후 Group Members 아래의 Edit 버튼을 누릅니다 . 표시되는 페이지에 기존 그룹 구성원이 나열됩니다 . Search 필드도 표시됩니다 .
   - 구성원 목록에 사용자를 추가하려면 Find 드롭다운 목록에서 Users 를 선택해야 합니다 .
   - 그룹 항목에 그룹을 추가하려면 Groups 를 선택해야 합니다 .
4. Matching 텍스트 필드에 검색 문자열을 입력합니다 . 다음 옵션을 입력합니다 .
   - 이름 . 전체 또는 이름의 일부를 지정합니다 . 검색 문자열과 일치되는 이름의 항목이 검색됩니다 . 해당 항목을 찾을 수 없는 경우 검색 문자열을 포함하는 모든 항목이 검색됩니다 . 해당 항목을 찾을 수 없는 경우 검색 문자열과 발음이 비슷한 모든 항목이 검색됩니다 .
   - 사용자 아이디 . 사용자 아이디의 일부만 입력하면 문자열을 포함하는 모든 항목이 검색됩니다 .
   - 전화번호 . 번호를 부분적으로 입력하면 검색 번호로 끝나는 전화번호를 포함하는 모든 항목이 검색됩니다 .
   - 전자 메일 주소 . @ 기호를 포함하는 모든 검색 문자열은 전자 메일 주소인 것으로 가정합니다 . 정확히 일치하는 검색 결과가 없을 경우 검색 문자열로 시작하는 모든 전자 메일 주소를 찾습니다 .
   - 현재 디렉토리에 있는 모든 항목이나 그룹을 보려면 이 텍스트 필드에 별표 (*) 를 입력하거나 빈 칸으로 남겨놓습니다 .
   - 모든 LDAP 검색 필터 . 등호 (=) 가 있는 모든 문자열은 검색 필터로 간주됩니다 .
5. LDAP 데이터베이스에서 모든 일치하는 항목을 찾아 해당 그룹에 추가하려면 Add 를 누릅니다 . 검색 결과에 그룹에 추가하지 않으려는 항목이 포함된 경우 Remove From List 열의 해당 확인란을 선택합니다 . (그룹에서 제거할 항목과 일치하도록 검색 필터를 지정하고 Remove 를 누를 수도 있습니다 . 자세한 내용은 " 그룹 구성원 목록에서 항목 제거 " (68 페이지) 를 참조하십시오 .)

6. 그룹 구성원 목록이 완료되었으면 Save Changes 를 누릅니다 . 항목이 그룹 구성원 목록에 추가됩니다 .

그룹 구성원 목록에 그룹 추가

LDAP 서비스의 경우 그룹의 구성원 목록에 그룹 (개별 구성원 대신) 을 추가할 수 있습니다 . 이렇게 하면 포함된 그룹에 속한 사용자는 모두 대상 그룹의 구성원이 됩니다 . 예를 들어 Neil Armstrong 이 Engineering Managers 그룹의 구성원이며 Engineering Managers 그룹을 Engineering Personnel 그룹의 구성원으로 추가하면 Neil Armstrong 또한 Engineering Personnel 그룹의 구성원이 됩니다 .

그룹을 다른 그룹의 구성원 목록에 추가하려면 그룹이 사용자 항목인 것처럼 추가합니다 . 자세한 내용은 " 그룹 구성원 추가 " (67 페이지) 를 참조하십시오 .

그룹 구성원 목록에서 항목 제거

이 절차는 LDAP 서비스에만 적용됩니다 .

그룹 구성원 목록에서 항목을 제거하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Groups 링크를 누릅니다 .
3. " 그룹 항목 찾기 " (64 페이지) 에 설명한 것과 같이 관리하려는 그룹을 찾은 후 Group Members 옆의 Edit 버튼을 누릅니다 .
4. 목록에서 제거하려는 각 구성원에 대해 Remove From List 열 해당 확인란을 선택합니다 . 또한 제거하려는 항목과 일치하는 검색 필터를 만든 후 Remove 를 누르면 됩니다 . 검색 필터를 만드는 방법에 대한 자세한 내용은 " 그룹 구성원 추가 " (67 페이지) 를 참조하십시오 .
5. Save Changes 를 누릅니다 . 그룹 구성원 목록에서 해당 항목이 삭제됩니다 .

소유자 관리

LDAP 서비스의 경우 그룹 소유자 목록은 그룹 구성원 목록과 같은 방식으로 관리됩니다 .

자세한 정보를 제공하는 본 설명서의 항목 목록은 다음 표와 같습니다 .

표 4-6 소유자 관리

수행 작업	참조 항목
그룹에 소유자 추가	" 그룹 구성원 추가 " (67 페이지)
소유자 목록에 그룹 추가	" 그룹 구성원 목록에 그룹 추가 " (68 페이지)
소유자 목록에서 항목 제거	" 그룹 구성원 목록에서 항목 제거 " (68 페이지)

추가 참조 관리

추가 참조는 현재 그룹과 관련될 수 있는 다른 디렉토리 항목을 참조합니다 . 여기에서 현재 그룹과 관련된 사용자 및 기타 그룹의 항목을 쉽게 찾을 수 있습니다 . 그룹 구성원 목록을 관리하는 것과 마찬가지로 추가 참조를 관리할 수 있습니다 .

자세한 정보를 제공하는 본 설명서의 항목 목록은 다음 표와 같습니다 .

표 4-7 추가 참조 관리

수행 작업	참조 항목
추가 참조에 사용자 추가	" 그룹 구성원 추가 " (67 페이지)
추가 참조에 그룹 추가	" 그룹 구성원 목록에 그룹 추가 " (68 페이지)
추가 참조에서 항목 제거	" 그룹 구성원 목록에서 항목 제거 " (68 페이지)

그룹 이름 변경

이 절차는 LDAP 서비스에만 적용됩니다 . 그룹 항목의 이름을 변경하면 그룹의 이름만 변경됩니다 . Rename Group 기능을 사용하여 항목을 하나의 조직 단위에서 다른 단위로 옮길 수는 없습니다 . 예를 들어 회사에 다음과 같은 조직이 있는 것으로 가정합니다 .

- Marketing 및 Product Management 를 위한 조직 단위
- Marketing 조직 단위 아래의 Online Sales 그룹

이 예에서 그룹의 이름을 Online Sales 에서 Internet Investments 로 변경할 수 있으나 Marketing 조직 단위 아래에 있는 Online Sales 가 Product Management 조직 단위 아래의 Online Sales 로 되도록 항목의 이름을 변경할 수는 없습니다 .

그룹 이름을 변경하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. " 그룹 항목 찾기 " (64 페이지) 의 설명에서처럼 , Manage Group 링크를 누르고 관리할 그룹을 찾습니다 .
3. Rename Group 버튼을 누르고 표시되는 페이지에 새 그룹 이름을 지정한 다음 Save Changes 를 누릅니다 .

그룹 제거

이 절차는 LDAP 서비스에만 적용됩니다 .

그룹을 제거하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Groups 링크를 누릅니다 .
3. " 그룹 항목 찾기 " (64 페이지) 에서 설명한 것처럼 삭제할 그룹을 찾은 다음 Delete Group 을 누릅니다 .

참고 그룹의 개별 구성원은 제거되지 않습니다 . 그룹 항목만 제거됩니다 .

조직 단위 만들기

LDAP 서비스의 경우 조직 단위에는 그룹 구성원이 포함될 수 있으며 보통 사업 단위 , 부서 또는 기타 명확히 구분되는 개체를 나타냅니다 . DN 은 하나 이상의 조직 단위에 존재할 수 있습니다 .

조직 단위를 만들려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Create Organizational Unit 링크를 누릅니다 .
3. 정보를 입력하고 Create 를 누릅니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

조직 단위 참고 사항

- 새 조직 단위는 `organizationalUnit` 개체 클래스를 사용하여 만들어집니다 .

- 새 조직 단위용 고유 이름의 형식은 다음과 같습니다 .

  ou= *새 조직* ,ou= *상위 조직* , . . . ,o= *기본 조직* ,c= *국가*

예를 들어 Accounting 이라는 이름의 새 조직을 West Coast 라는 이름의 조직 단위 내에 만들며 o=Ace Industry,c=USUS 인 경우 , 새 조직 단위의 DN 은 다음과 같습니다 .

```
ou=Accounting,ou=West Coast,o=Ace Industry,c=US
```

조직 단위 관리

LDAP 서비스의 경우 조직 단위는 Manage Organizational Units 페이지의 Administration Server Users and Groups 탭에서 편집 및 관리할 수 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 조직 단위 찾기
- 조직 단위 속성 편집
- 조직 단위 이름 변경
- 조직 단위 제거

조직 단위 찾기

이 절차는 LDAP 서비스에만 적용됩니다 .

조직 단위를 찾으려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Organizational Units 링크를 누릅니다 .
3. Find Organizational Unit 필드에 찾으려는 단위의 이름을 입력합니다 . 다음과 같은 항목을 입력할 수 있습니다 .
   - 이름 . 전체 또는 이름의 일부를 지정합니다 . 검색 문자열과 일치하는 모든 항목이 검색됩니다 . 해당 항목을 찾을 수 없는 경우 검색 문자열을 포함하는 모든 항목이 검색됩니다 . 해당 항목을 찾을 수 없는 경우 검색 문자열과 발음이 비슷한 모든 항목이 검색됩니다 .
   - 디렉토리에 있는 그룹을 모두 보려면 별표 (*) 를 사용합니다 . 필드에 아무런 값을 입력하지 않아도 동일한 결과를 얻을 수 있습니다 .

- 모든 LDAP 검색 필터 . 등호 (=) 가 있는 모든 문자열은 검색 필터로 간주됩니다 .

다른 방법으로 Find All Units Whose 항목의 드롭다운 메뉴를 사용하여 검색 범위를 좁힐 수 있습니다 . 자세한 내용은 "Find All Groups Whose 항목 " (72 페이지) 을 참조하십시오 .

4. Look within 필드에서 검색하려는 항목의 상위 조직 단위를 선택합니다 . 기본값은 디렉토리의 루트 지점 (최상단 항목) 입니다 .
5. 결과를 화면에 표시할지 , 프린터에서 출력할지를 드롭다운 목록에서 선택합니다 .
6. 이 프로세스의 아무 단계에서나 Find 버튼을 누릅니다 . 검색 기준과 일치하는 모든 조직 단위가 표시됩니다 .
7. 표시할 항목에 대한 링크를 누릅니다 .

Find All Groups Whose 항목

LDAP 서비스의 경우 Find All Units Whose 항목을 사용하면 사용자 정의 검색 필터를 만들 수 있습니다 . 이 필드를 사용하면 Find Organizational Unit 에서 반환되는 검색 결과를 더욱 좁힐 수 있습니다 .

왼쪽의 드롭다운 목록에서 검색 기준으로 사용할 속성을 지정합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .

- **Unit name.** 각 항목의 전체 이름이 일치하도록 검색합니다 .
- **Description.** 각 조직 단위 항목의 설명이 일치되는지 검색합니다 .

가운데 드롭다운 목록에서 검색의 유형을 지정합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .

- **Contains.** 하위 문자열 검색이 수행되도록 합니다 . 지정한 검색 문자열을 포함하는 속성 값을 가진 항목이 반환됩니다 . 예를 들어 , 조직 단위의 이름에 "Administrator" 라는 단어가 포함된 것을 알고 있는 경우 이 옵션을 검색 문자열 "Administrator" 와 함께 사용하여 해당 조직 단위 항목을 찾을 수 있습니다 .
- **Is.** 정확히 일치하는 항목을 검색합니다 . 조직 단위 속성의 값을 정확히 아는 경우 이 옵션을 사용합니다 . 예를 들어 조직 단위의 정확한 철자를 아는 경우입니다 .
- **isn't** . 검색 문자열과 정확히 일치하지 않는 속성값을 가진 모든 항목을 검색합니다 . 예를 들어 디렉토리에서 이름에 "administrator" 가 포함되지 않는 모든 조직 단위를 찾으려면 이 옵션을 사용합니다 . 그러나 이 옵션을 사용하면 지나치게 많은 수의 항목이 검색될 수 있으므로 주의해야 합니다 .

- **Sounds like.** 근사치 또는 발음에 의한 검색이 수행됩니다 . 속성의 값은 알지만 철자법이 확실하지 않은 경우 이 옵션을 사용합니다 . 예를 들어 조직 단위의 이름이 "Sarret's list", "Sarette's list" 또는 "Sarett's list" 인지 확실하지 않은 경우입니다 .
- **Starts with.** 하위 문자열 검색이 수행되도록 합니다 . 지정된 검색 문자열로 시작하는 속성 값을 가진 모든 항목을 검색합니다 . 예를 들어 조직 단위 이름이 "Product"로 시작되지만 나머지 이름은 알지 못하는 경우 이 옵션을 사용합니다.
- **Ends with.** 하위 문자열 검색이 수행되도록 합니다 . 지정된 검색 문자열로 끝나는 속성 값을 가진 모든 항목을 검색합니다 . 예를 들어 조직 단위 이름이 "development" 로 끝나지만 나머지 이름은 알지 못하는 경우 이 옵션을 사용합니다 .

오른쪽 텍스트 입력란에 검색 문자열을 입력합니다 . 검색 위치 디렉토리에 포함된 모든 조직 단위 항목을 표시하려면 별표 (*) 를 입력하거나 이 필드를 빈 칸으로 남겨 둡니다 .

조직 단위 속성 편집

이 절차는 LDAP 서비스에만 적용됩니다 .

조직 단위 항목을 편집하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Organizational Units 링크를 누릅니다 .
3. " 조직 단위 찾기 " (71 페이지) 에서 설명한 대로 편집하려는 조직 단위를 찾습니다 .
4. 원하는 내용을 변경합니다 . 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

참고 그룹 단위 편집 페이지에서 표시되지 않는 속성 값을 변경하고자 할 수 있습니다 . 이 경우 사용할 수 있으면 디렉토리 서버 `ldapmodify` 명령줄 유틸리티를 사용하십시오 .

조직 단위 이름 변경

이 절차는 LDAP 서비스에만 적용됩니다 . 조직 단위 항목의 이름을 변경하면 조직 단위의 이름만 변경됩니다 . 이름 변경 기능을 사용하여 항목을 하나의 조직 단위에서 다른 단위로 옮길 수는 없습니다 .

조직 단위 항목의 이름을 변경하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Organizational Units 링크를 누릅니다 .
3. " 조직 단위 찾기 " (71 페이지) 에서 설명한 대로 편집하려는 조직 단위를 찾습니다 .
4. Rename Group 버튼을 누르고 표시되는 페이지에 새 조직 단위 이름을 입력한 다음 Save Changes 를 누릅니다 .

조직 단위 제거

이 절차는 LDAP 서비스에만 적용됩니다 .

조직 단위 항목을 삭제하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Users and Groups 탭을 누릅니다 .
2. Manage Organizational Units 링크를 누릅니다 .
3. " 조직 단위 찾기 " (71 페이지) 에서 설명한 대로 삭제하려는 조직 단위를 찾습니다 .
4. Delete 버튼을 누르고 나타나는 확인란에서 OK 를 선택합니다 .

5 장

인증서 및 키 사용

이 장에서는 인증서 및 키 인증을 사용하여 secure Sun Java System Web Proxy Server 의 보안을 수행하는 방법에 대해 설명합니다 . Proxy Server 는 모든 Sun Java System 서버의 보안 아키텍처를 통합하며 , 최대의 상호 운용성과 일관성을 위해 업계 표준 및 공용 프로토콜을 기반으로 구축되었습니다 .

이 장에서는 사용자가 암호화 및 복호화 , 공용 및 개인 키 , 디지털 인증서 , 암호화 프로토콜 등과 같은 공용 키 암호화에 대한 기본 개념을 알고 있다고 가정합니다 . 자세한 내용은 SSL 개요를 참조하십시오 . 이 문서는 http://docs.sun.com/source/816-6156-10/index.htm 에서 제공합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

인증서 기반 인증

인증은 신분을 확인하는 과정입니다 . 네트워크 상호작용이라는 맥락에서 인증은 한 쪽이 다른 쪽의 신분을 명확히 확인하는 것입니다 . 인증서는 인증을 지원하는 방법 중 한 가지입니다 .

인증서는 개인 , 회사 또는 기타 단체의 이름을 명시하는 디지털 데이터로 구성되며 인증서에 포함된 공용 키가 해당 단체의 소유인지 검사합니다 .

클라이언트와 서버는 모두 인증서를 가질 수 있습니다 . 서버 인증이란 클라이언트가 서버의 신분을 확인하는 것을 말합니다 . 즉 , 특정 네트워크 주소에서 서버에 대한 책임이 있는 단체의 신분을 확인하는 것입니다 . 클라이언트 인증이란 서버가 클라이언트의 신분을 확인하는 것으로 , 즉 , 클라이언트 소프트웨어를 사용하는 사람의 신분을 확인하는 것입니다 . 개인이 여러 개의 신분증을 가질 수 있는 것처럼 클라이언트에는 여러 개의 인증서가 있을 수 있습니다 .

인증서는 인증 기관 (또는 CA) 이 발행하고 디지털로 서명됩니다 . CA 는 인증서를 판매하는 회사이거나 , 회사의 인트라넷 또는 익스트라넷용으로 인증서를 발행하는 부서일 수 있습니다 . 다른 사람의 신분을 확인하는 데 충분히 신뢰할 수 있는 CA 를 선택합니다 .

인증서에 의하여 확인되는 공용 키와 단체의 이름에 더하여 인증서에는 또한 만기일 , 인증서를 발행한 CA 의 이름 및 발행 CA 의 디지털 서명이 포함됩니다 .

인증서의 내용과 형식에 대한 자세한 내용은 SSL 개요를 참조하십시오 .

지원되는 인증서 확장자에 대한 자세한 내용은 All About Certificate Extensions 를 참조하십시오 . 이 문서는 http://www.mozilla.org/projects/security/pki/nss/tech-notes/tn3.html 에서 제공합니다 .

참고	서버 인증서는 반드시 암호화 기능을 사용하기 전에 설치되어야 합니다 .

신뢰 데이터베이스 만들기

서버 인증서를 요청하기 전에 반드시 신뢰할 수 있는 데이터베이스를 만들어야 합니다 . Proxy Server 에서는 Administration Server 와 각 서버 인스턴스에 자체의 신뢰 데이터베이스를 부여할 수 있습니다. 신뢰 데이터베이스는 로컬 컴퓨터에만 만들 수 있습니다 .

신뢰 데이터베이스를 만들 때 키 쌍 파일용으로 사용할 비밀 번호를 지정합니다 . 또한 암호화된 통신을 사용하여 서버를 시작할 때에도 이 비밀 번호가 필요합니다 . 암호를 변경하는 경우 고려해야 할 지침은 " 강력한 비밀 번호 선택 " (114 페이지) 을 참조하십시오 .

신뢰 데이터베이스에서 공용 및 개인 키를 만들고 저장할 수 있습니다 . 이는 키 쌍 파일이라고 합니다 . 키 쌍 파일은 SSL 암호화에 사용됩니다 . 키 쌍 파일은 서버 인증서를 요청하고 설치할 때 사용합니다 . 인증서가 설치되면 신뢰 데이터베이스에 저장됩니다 .

키 쌍 파일은 다음 디렉토리에 암호화되어 저장됩니다 .

server_root/alias/proxy-*serverid*-key3.db

Administration Server 에는 오직 하나의 신뢰 데이터베이스만 있습니다 . 각 서버 인스턴스에는 자체의 신뢰 데이터베이스를 부여할 수 있습니다 .

신뢰 데이터베이스를 만들려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Create Database 링크를 누릅니다 .
3. 신뢰 데이터베이스용 비밀 번호를 입력합니다 .
4. 비밀 번호를 다시 입력하고 OK 를 누릅니다 .

password.conf 사용

기본적으로 시작하기 전에 관리자에게 키 데이터베이스 비밀 번호를 입력하라는 프롬프트가 Proxy Server 에 표시됩니다 . 무인 작업으로 Proxy Server 를 재시작하려면 비밀 번호를 password.conf 파일에 저장해 놓아야 합니다 . 시스템이 적절히 보호되어 이 파일과 키 데이터베이스가 조작되지 않을 경우에만 이 기능을 사용하십시오 .

보통 UNIX SSL 을 사용하는 서버의 경우 시작하기 전에 비밀 번호가 필요하므로 /etc/rc.local 또는 /etc/inittab 파일을 사용할 수 없습니다 . 비밀 번호를 일반 텍스트 파일에 저장하여 SSL 을 사용하는 서버를 자동으로 시작할 수는 있으나 , 이는 권장하지 않습니다 . 서버의 password.conf 파일은 루트 또는 서버를 설치한 사용자의 소유여야 하며 , 오직 소유자만 이 파일을 읽고 쓸 수 있어야 합니다 .

UNIX 의 경우 SSL 을 사용하는 서버의 비밀 번호를 password.conf 파일에 남겨두면 보안상의 위험이 커집니다 . 파일에 액세스할 수 있는 사용자는 모두 SSL 을 사용하는 서버의 비밀 번호를 알 수 있습니다 . SSL 을 사용하는 서버의 비밀 번호를 password.conf 파일에 보관하기 전에 보안의 위험에 대해 고려해야 합니다 .

Windows 에서 NTSF 파일 시스템을 사용하는 경우 , password.conf 파일을 사용하지 않는 경우라도 이 파일이 들어 있는 디렉토리에 대한 액세스를 제한하여 보호해야 합니다 . 디렉토리에는 Administration Server 사용자 및 Proxy Server 사용자용 읽기 및 쓰기 권한이 있어야 합니다 . 디렉토리를 보호하면 다른 사람이 잘못된 password.conf 파일을 만들 수 없도록 방지합니다 . FAT 파일 시스템의 경우 액세스를 제한하는 경우에도 디렉토리를 보호할 수 없습니다 .

SSL 을 사용하는 서버 자동 시작

SSL 을 사용하는 서버를 자동 시작하려면 다음과 같이 합니다 .

1. SSL 을 사용하도록 설정되었는지 확인합니다 .
2. Proxy Server 인스턴스의 config 하위 디렉토리에 password.conf 파일을 새로 만듭니다 .
   - Proxy Server 와 함께 제공되는 내부 PKCS#11 소프트웨어 암호화 모듈을 사용하는 경우에는 다음 정보를 입력합니다 .
     internal:*your_ password*
   - 다른 PKCS #11 모듈 (하드웨어 암호화 또는 하드웨어 가속기용) 을 사용하는 경우에는 해당 PKCS #11 모듈의 이름 다음에 비밀 번호를 지정합니다 . 예 :
     nFast:*your_ password*

password.conf 파일을 만든 후라도 Proxy Server 를 시작할 때에는 항상 비밀 번호를 입력하라는 프롬프트가 표시됩니다 .

VeriSign 인증서 요청 및 설치

VeriSign 은 Proxy Server 권장 인증 기관 (CA) 입니다 . 이 회사의 기술은 인증서 요청 프로세스를 간소화합니다 . VeriSign 에는 인증서를 사용자의 서버로 직접 회신할 수 있다는 장점이 있습니다 .

서버용 인증서 신뢰 데이터베이스를 만든 후 인증서를 요청하고 이를 인증기관 (CA) 에 제출할 수 있습니다 . 회사에 내부 CA 가 있는 경우에는 해당 부서로 인증서를 요청합니다 . 상용 CA 로부터 인증서를 구매할 계획인 경우에는 CA 를 선택하고 필요한 정보의 형식이 있는지 문의합니다 .

Administration Server 에는 오직 하나의 서버 인증서만 부여할 수 있습니다 . 각 서버 인스턴스에는 자체의 서버 인증서를 부여할 수 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

VeriSign 인증서 요청

VeriSign 인증서를 요청하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Request VeriSign Certificate 링크를 누릅니다 .
3. 표시되는 페이지에 나타난 단계를 확인하고 OK 를 누릅니다 . VeriSign 등록 마법사가 표시되어 필요한 단계를 안내합니다 .

VeriSign 인증서 설치

VeriSign 인증서를 설치하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Install VeriSign Certificate 링크를 누릅니다 .
3. 외부 암호화 모듈을 사용하지 않는 한 Cryptographic Module 드롭다운 목록에서 내부 모듈을 선택합니다 .

4. 키 쌍 파일 비밀 번호 또는 PIN 을 입력합니다 .
5. 드롭다운 목록에서 Transaction ID to Retrieve 를 선택하고 OK 를 누릅니다 .

기타 서버 인증서 요청 및 설치

VeriSign 외에도 다른 인증 기관에 인증서를 요청하여 설치할 수 있습니다 . 회사나 조직에서 자체의 내부 인증서를 제공할 수도 있습니다 . 여기에서는 다른 종류의 서버 인증서를 요청하고 설치하는 방법에 대해 설명합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

필수 CA 정보

요청 프로세스를 시작하기 전에 CA 가 요구하는 정보가 무엇인지 알아야 합니다 . 요청되는 정보의 형식은 CA 에 따라 다르지만 보통 다음과 같은 정보를 입력하도록 요청합니다 . 참고로 이 정보의 대부분은 인증서 갱신의 경우에는 필요하지 않습니다 .

- **Requestor name.** 인증서가 발행된 이름입니다 .
- **Telephone number.** 요청자의 전화번호입니다 .
- **Common name.**DNS 조회에 사용되는 유효한 호스트 이름입니다 (예 : `www.example.com`).
- **Email address.** 업체와 CA 사이의 통신용으로 사용할 사업용 전자 메일 주소입니다 .
- **Organization.** 회사 , 교육기관 , 단체 등의 공식적 , 법적 이름입니다 . 대부분의 CA 는 정보에 대해 법적 서류 (사업자 등록 등) 로 확인할 것을 요구합니다 .
- **Organizational unit.** 회사 내 조직 단위의 설명입니다 .
- **Locality.** 조직의 시 , 도 또는 국가에 대한 설명입니다 .
- **State or Province.** 회사가 위치한 시 또는 도입니다 .
- **Country.** 국가 이름의 두 자리 약자입니다 (ISO 형식). 미국의 국가 코드는 US 이며 , 한국은 KR 입니다 .

이 모든 정보는 DN(Distinguished Name) 이라고 하는 일련의 속성 값 쌍으로 조합되어 인증서의 개체를 고유하게 구분합니다 .

상용 CA 에서 인증서를 구매하는 경우에는 반드시 CA 에 연락하여 인증서를 발행하기 위하여 필요한 추가 정보가 있는지 확인해야 합니다 . 대부분의 CA 는 신분에 대한 증명을 요구합니다 . 예를 들어 , 회사 이름에 대한 확인 , 회사가 서버를 관리하도록 지정한 사용자 등을 확인하며 사용자가 제공하는 정보를 사용할 법적 권한이 있는지 확인할 것입니다 .

일부 상용 CA 는 완벽한 신분을 제공하는 조직이나 개인에게 더 자세하고 정확한 인증서를 제공합니다 . 예를 들어 사용자가 `www.example.com` 컴퓨터에 대한 관리의 권한이 사용자에게 있는지 확인하지 않았으나 , 3 년간 경영해온 회사로 유의할 고객 소송이 없었다는 사실을 표시하는 인증서를 구매할 수 있습니다 .

기타 서버 인증서 요청

기타 서버 인증서를 요청하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Request Certificate 링크를 누릅니다 .
3. 신규 인증서인지 또는 인증서 갱신인지 지정합니다 . 인증서는 대부분 6 개월이나 1 년 등 , 일정 시간이 경과하면 무효화됩니다 . CA 에 따라 자동으로 갱신을 송신하는 경우도 있습니다 .
4. 인증서 요청을 제출할 방법을 지정합니다 .
   - 전자 메일로 요청을 제출하려면 CA Email Address 를 선택하고 이 요청에 해당하는 전자 메일 주소를 입력합니다 .
   - CA 의 웹 사이트를 통해 요청을 제출하려면 CA URL 을 선택하고 이 요청에 해당하는 URL 을 입력합니다 .
5. Cryptographic Module 드롭다운 목록에서 , 인증서를 요청할 때 키 쌍 파일에 사용할 암호화 모듈을 선택합니다 .
6. 키 쌍 파일용 비밀 번호를 입력합니다 . 내부 모듈이 아닌 다른 암호화 모듈을 선택하지 않은 한 , 이 비밀 번호는 신뢰 데이터를 만들 때 지정한 비밀 번호입니다 . 서버는 비밀 번호를 사용하여 개인 키를 구하고 CA 로 전송되는 메시지를 암호화합니다 . 그런 후 , 서버는 공용 키와 암호화된 메시지를 모두 CA 로 전송합니다 . CA 는 공용 키를 사용하여 메시지를 해독합니다 .

7. 이름이나 전화번호 같은 신분 정보를 입력합니다 . 이 정보의 형식은 CA 에 따라 다릅니다 . 참고로 이 정보의 대부분은 인증서 갱신의 경우에는 필요하지 않습니다 .
8. 입력한 사항이 정확한지 다시 한 번 확인하고 OK 를 누릅니다 . 정보가 정확할수록 인증서가 더욱 빨리 승인될 수 있습니다 . 요청이 서버 인증을 위한 것인 경우에는 요청을 제출하기 전에 양식 정보를 확인하라는 프롬프트가 표시됩니다 .

서버가 정보를 포함하는 인증서 요청을 생성합니다 . 요청에는 개인 키를 사용하여 만든 전자 서명이 포함됩니다 . CA 는 전자 서명을 사용하여 요청이 서버 컴퓨터에서 CA 로 라우팅되는 동안 조작되지 않았는지 검사합니다 . 드물지만 요청이 조작된 경우에는 보통 CA 가 전화를 통하여 사용자에게 문의합니다 .

요청을 전자 메일로 보내는 경우 서버가 요청이 포함된 전자 메일 메시지를 작성한 후 CA 로 전송합니다 . 이후 보통 인증서는 전자 메일로 전달됩니다 . 인증 서버의 URL 을 지정하는 경우에는 서버가 URL 을 사용하여 요청을 인증 서버에 제출합니다 . CA 에 따라 전자 메일 또는 다른 수단을 통해 응답을 받을 수 있습니다 .

CA 가 인증서 발행에 동의하는 경우 해당 사실을 통지합니다 . 대부분의 경우 CA 는 전자 메일로 인증서를 전송합니다 . 회사에서 인증 서버를 사용하는 경우 인증서 서버의 형식을 사용하여 인증서를 검색할 수 있습니다 .

참고 상용 CA 로 인증서를 요청하는 모든 사람에게 인증서가 발행되는 것은 아닙니다 . 많은 CA 가 인증서를 발행하기 전에 신분 증명을 요구합니다 . 또한 승인에는 하루에서 몇 주까지 걸릴 수 있습니다 . 사용자는 CA 에 필요한 정보를 모두 신속히 제공할 책임이 있습니다 .

인증서를 받으면 설치합니다 . 그 동안에는 SSL 없이 Proxy Server 를 계속 사용할 수 있습니다 .

기타 서버 인증서 설치

CA 에서 인증서를 수신하면 인증서는 공용 키로 암호화되므로 오직 귀사만 이를 해독할 수 있습니다 . 오직 정확한 신뢰 데이터베이스용 비밀 번호를 입력해야만 인증서를 해독하고 설치할 수 있습니다 .

인증서에는 세 가지 유형이 있습니다 .

- 클라이언트에게 제시할 자체 서버의 인증서
- 인증서 체인에서 사용할 CA 의 자체 인증서

- 신뢰된 CA 의 인증서

인증서 체인이란 연속적인 인증 기관이 서명한 일련의 계층적 인증서를 말합니다 . CA 인증서는 인증 기관 (CA) 을 확인하고 해당 기관이 발행한 인증서에 서명하는 데 사용됩니다 . 이 CA 인증서는 다시 상위 CA 의 CA 인증서에 의하여 서명되는 과정을 되풀이하여 루트 CA 의 서명까지 이어집니다 .

참고 CA 가 자동으로 자신들의 인증서를 보내지 않은 경우에는 요청해야 합니다 . 많은 CA 가 귀사의 인증서가 있는 전자 메일에 자체의 인증서를 포함하며 , 서버는 이 두 인증서를 동시에 설치합니다 .

CA 에서 인증서를 수신하면 인증서는 공용 키로 암호화되므로 오직 귀사만 이를 해독할 수 있습니다 . 인증서를 설치할 때 Proxy Server 는 사용자가 지정한 키 쌍 파일 비밀 번호를 사용하여 이를 해독합니다 . 전자 메일을 서버에 액세스할 수 있는 다른 위치에 저장하거나 전자 메일의 텍스트를 복사한 후 Install Certificate 형식의 해당 위치에 붙여 넣을 수 있도록 합니다 .

다른 서버 인증서를 설치하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Install Certificate 링크를 누릅니다 .
3. Certificate For 옆에서 설치할 인증서 유형을 선택합니다 .
   - This Server
   - Server Certificate Chain
   - Certification Authority

   특정 설정에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .
4. 드롭다운 목록에서 암호화 모듈을 선택합니다 .
5. 키 쌍 파일 비밀 번호를 입력합니다 .
6. 3 단계에서 Server Certificate Chain 이나 Certification Authority 를 선택한 경우에만 인증서 이름을 입력합니다 .
7. 다음 중 하나를 수행하여 인증 정보를 입력합니다 .
   - Message Is In This 을 선택한 후 CA 인증서를 포함하는 파일의 전체 경로 이름을 입력합니다 .

- Message Text(with headers) 를 선택한 다음 CA 인증서의 내용을 복사하여 붙여 넣습니다 . 시작 및 종료 하이픈을 포함하여 Begin Certificate 및 End Certificate 헤더를 포함하도록 합니다 .

8. OK 를 누릅니다 .
9. 다음 중 한 가지를 선택합니다 .
   - Add Certificate. 신규 인증서를 설치하는 경우
   - Replace Certificate. 인증서 갱신을 설치하는 경우

인증서는 서버의 인증서 데이터베이스에 저장됩니다 . 예 :

server_root/alias/proxy-*serverid*-cert8.db

인증서 마이그레이션

Sun ONE Web Proxy Server 3.6(iPlanet Web Proxy Server 라고도 함) 에서 Sun Java System Web Proxy Server 4 로 마이그레이션할 경우 , 신뢰 및 인증서 데이터베이스를 포함한 사용자 파일이 자동 업데이트됩니다 .

Proxy Server 4 Administration Server 가 기존 3.x 데이터베이스 파일에 대한 읽기 권한이 있어야 합니다 . 파일 이름은 *alias*-cert.db 및 *alias*-key.db 이며 , *3.x_server_root*/alias 디렉토리에 있습니다 .

키 쌍 파일 및 인증서는 서버에서 보안 기능을 사용하는 경우에만 마이그레이션됩니다 . 또한 Administration Server 및 Server Manager 의 Security 탭에 있는 Migrate 3.x Certificates 옵션을 사용하여 자체적으로 키와 인증서를 마이그레이션할 수도 있습니다 . 특정 설정에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

이전 버전의 경우 인증서와 키 쌍 파일은 여러 서버 인스턴스가 사용할 수 있는 별칭에 의하여 참조되었습니다 . Administration Server 가 모든 별칭과 해당 구성 인증서를 관리했습니다 . Sun Java System Web Proxy Server 4 의 경우 Administration Server 와 각 서버 인스턴스에는 자체의 인증서 및 키 쌍 파일이 있으며 , 이는 별칭이 아닌 신뢰 인증서라고 합니다 .

신뢰 데이터베이스 및 해당 구성 인증서는 Administration Server 자체에서 , 그리고 서버 인스턴스에 대한 Server Manager 에서 관리합니다 . 이제 인증서와 키 쌍 데이터베이스 파일은 이를 사용하는 서버 인스턴스의 이름을 따라 이름이 지정됩니다 . 이전 버전에서 여러 서버 인스턴스가 동일한 별칭을 공유한 경우 , 마이그레이션된 인증서와 키 쌍 파일의 이름은 새로운 서버 인스턴스용으로 변경됩니다 .

서버 인스턴스와 연결된 신뢰 데이터베이스 전체가 마이그레이션됩니다 . 이전 데이터베이스의 모든 CA 가 Proxy Server 4 데이터베이스로 마이그레이션됩니다 . CA 가 중복되는 경우 유효 기간 동안 이전 CA 를 사용합니다 . 중복되는 CA 를 삭제하면 안 됩니다 .

Proxy Server 3.x 인증서는 지원되는 NSS(Network Security Services) 형식으로 마이그레이션됩니다 . 인증서의 이름은 액세스된 Proxy Server 페이지에 따라 결정됩니다 . 즉 , Administration Server 의 Security 탭에서 액세스되었는지 , Server Manager 의 Security 탭에서 액세스되었는지에 따라 달라집니다 .

인증서를 마이그레이션하려면 다음과 같이 합니다 .

1. 로컬 컴퓨터에서 Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Migrate 3.x Certificates 링크를 누릅니다 .
3. 3.6.X 서버가 설치된 루트 디렉토리를 지정합니다 .
4. 이 컴퓨터의 별칭을 지정합니다 .
5. 관리자의 비밀 번호를 다시 입력하고 OK 를 누릅니다 .

내장 루트 인증서 모듈 사용

동적으로 로드할 수 있는 루트 인증서 모듈이 Proxy Server 에 포함되어 있으며 , 여기에는 VeriSign 을 비롯하여 많은 CA 용 루트 인증서가 있습니다 . 루트 인증서 모듈을 사용하면 이전보다 훨씬 쉽게 루트 인증서를 신규 버전으로 업그레이드할 수 있습니다 . 이전에는 오래된 루트 인증서를 한 번에 하나씩 삭제하고 새 인증서를 한 번에 하나씩 설치해야 했습니다 . 이제 잘 알려진 CA 인증서를 설치하는 경우 , 간단히 Proxy Server 의 신규 버전이나 Service Pack 이 발표될 때 루프 인증서 모듈 파일을 신규 버전으로 업데이트하면 됩니다 .

루트 인증서는 PKCS #11 암호화 모듈로 구현되므로 여기에 포함된 루트 인증서는 삭제할 수 없으며 , 해당 인증서를 관리하는 경우 삭제 옵션은 사용할 수 없게 됩니다 . 서버 인스턴스에서 루트 인증서를 제거하려면 서버의 별칭 파일에서 다음을 삭제하여 루트 인증서 모듈을 사용하지 않도록 설정해야 합니다 .

- `libnssckbi.so`(대부분의 UNIX 플랫폼에 적용)
- `nssckbi.dll`(Windows)

이후 루트 인증서 모듈을 복구하려면 *server_root*/bin/proxy/lib(UNIX) 또는 *server_root*\bin\proxy\bin(Windows) 에서 확장자를 별칭 하위 디렉토리로 복사합니다 .

루트 인증서의 신뢰 정보를 수정할 수 있습니다 . 신뢰 정보는 루트 인증서 모듈 그 자체가 아니라 편집하는 서버 인스턴스용 인증서 데이터베이스에 기록되어 있습니다 .

인증서 관리

서버에 설치된 다양한 인증서의 신뢰 설정을 확인 , 삭제 또는 편집할 수 있습니다 . 여기에는 귀사 자체의 인증서와 CA 의 인증서가 포함됩니다 .

인증서를 관리하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Manage Certificates 링크를 누릅니다 .
   - 내부 암호화 모듈을 사용하는 기본 구성용 인증서를 관리하는 경우에는 설치된 모든 인증서의 목록이 해당 유형 및 유효 기간과 함께 표시됩니다 . 모든 인증서는 *server_root*/alias 디렉토리에 저장됩니다 .
   - 하드웨어 가속기 등의 외부 암호화 모듈을 사용하는 경우에는 우선 해당 모듈용 비밀 번호를 입력하고 OK 를 눌러야 합니다 . 인증서 목록이 해당 모듈에 있는 인증서를 포함하여 업데이트됩니다 .
3. 관리할 인증서의 이름을 누릅니다 . 해당 인증서 유형에 대한 관리 옵션을 표시하는 페이지가 나타납니다 . 오직 CA 인증서의 경우에만 클라이언트 신뢰를 설정 또는 해제할 수 있습니다 . 외부 암호화 모듈에 따라 인증서를 삭제할 수 없는 경우도 있습니다 .
4. 필요한 작업을 지정합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .
   - Delete Certificate 또는 Quit(내부 인증서용)
   - Set client trust, Unset server trust 또는 Quit(CA 인증서용)

인증서 정보에는 소유자와 발행자가 표시됩니다 . 신뢰 설정을 이용하여 클라이언트 신뢰를 설정하거나 서버 신뢰를 해제할 수 있습니다 . LDAP 서버 인증서의 경우 서버가 반드시 신뢰되어야 합니다 .

CRL 및 KRL 설치 및 관리

인증서 철회 목록 (CRL) 과 변조된 키 목록 (CKL) 은 클라이언트나 서버 사용자가 더 이상 신뢰하면 안 되는 인증서를 표시합니다 . 예를 들어 인증서의 유효 기간이 끝나기 전에 사용자가 사무실을 이전하거나 퇴사하는 등 인증서의 데이터가 변경되면 인증서는 취소되며 CRL 에 해당 데이터가 표시됩니다 . 키가 조작 또는 변형되는 경우 해당 키와 데이터가 CKL 에 표시됩니다 . CRL 과 CKL 은 모두 CA 에 의하여 만들어지고 주기적으로 업데이트됩니다 . 이 목록은 해당하는 특정 CA 에 문의하십시오 .

이 절에서는 다음 항목에 대해 설명합니다 .

CRL 또는 CKL 설치

CRL 또는 CKL 을 설치하려면 다음과 같이 합니다 .

1. CA 로부터 CRL 이나 CKL 을 받아 로컬 디렉토리에 다운로드합니다 .
2. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
3. Install CRL/CKL 링크를 누릅니다 .
4. 다음 중 한 가지를 선택합니다 .
    - Certificate Revocation List
    - Compromised Key List
5. 해당 파일의 전체 경로 이름을 입력하고 OK 를 누릅니다 . CRL 또는 CKL 정보를 나타내는 Add Certificate Revocation List 나 Add Compromised Key List 페이지가 표시됩니다 . 데이터베이스에 이미 CRL 또는 CKL 이 있는 경우에는 Replace Certificate Revocation List 또는 Replace Compromised Key List 페이지가 표시됩니다 .
6. CRL 이나 CKL 을 추가 또는 교체합니다 .

CRL 및 CKL 관리

CRL 및 CKL 을 관리하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Manage CRL/CKL 링크를 누릅니다 . 설치된 서버의 CRL 과 CKL 및 유효 기간의 목록을 표시하는 Manage Certificate Revocation List/Compromised Key List 페이지가 나타납니다 .
3. Server CRLs 또는 Server CKLs 목록에서 인증서를 선택합니다 .
4. CRL 또는 CKL 을 삭제하려면 Delete CRL 또는 Delete CKL 을 , 관리 페이지로 돌아가려면 Quit 를 선택합니다 .

보안 기본 설정

인증서가 있으면 서버의 보안 작업을 시작할 수 있습니다 . Sun Java System Web Proxy Server 는 이 절에서 설명하는 여러 보안 요소를 제공합니다 .

암호화는 정보를 변환하여 의도된 수신자 외에 아무도 알아볼 수 없도록 하는 프로세스입니다 . 복호화는 암호화된 정보를 변환하여 다시 알아볼 수 있도록 하는 프로세스입니다 . Proxy Server 는 SSL(Secure Sockets Layer) 과 TLS(Transport Layer Security) 암호화 프로토콜을 지원합니다 .

암호**는** 암호화 알고리즘 (수학적 함수) 으로 암호화 또는 복호화에 사용됩니다 . SSL 및 TLS 프로토콜에는 다양한 암호 제품군이 포함됩니다 . 보안의 안전성과 강도는 암호마다 다릅니다. 일반적으로 암호가 사용하는 비트의 수가 많을수록 데이터를 해독하는 것이 어렵습니다 .

양방향 암호화 프로세스에서 양쪽에는 반드시 동일한 암호가 있어야 합니다. 다양한 암호를 사용할 수 있으므로 서버를 가장 공통적으로 사용되는 암호용으로 사용 설정해야 합니다 .

보안 연결에서 클라이언트와 서버는 양쪽이 통신에 사용할 수 있는 가장 강력한 암호화를 사용하도록 동의합니다 . 암호는 SSL 2.0, SSL 3.0 및 TLS 프로토콜 중 선택할 수 있습니다 .

참고 SSL 2.0 을 사용하면 보안과 성능이 향상됩니다 . SSL 3.0 을 사용할 수 있는 클라이언트에서는 SSL 2.0 을 사용하지 마십시오 . SSL 2.0 암호에서는 클라이언트 인증서가 제대로 작동한다고 보장할 수 없습니다.

암호화 프로세스 그 자체로는 서버의 비밀 정보를 보안하는 데 충분하지 않습니다 . 실제의 암호화 결과를 얻거나 이전에 암호화된 정보를 해독하려면 암호화 암호와 함께 키가 사용되어야 합니다 . 이를 위해 암호화 프로세스는 공용 키와 개인 키 등 , 두 가지 키를 사용합니다 . 공용 키로 암호화된 정보는 오직 연결된 개인 키로만 해독할 수 있습니다 . 공용 키는 인증서의 일부로 게시됩니다 . 연결된 개인 키만 보호할 수 있습니다 .

다양한 암호 제품군에 대한 설명과 키 및 인증서에 대한 자세한 내용은 SSL 개요를 참조하십시오 .

서버에서 사용할 수 있는 암호를 지정하려면 Proxy Server 사용자 인터페이스의 목록에서 해당 암호를 선택합니다 . 최적 암호화보다 적게 암호를 사용하도록 설정하고자 할 수도 있으나 , 특정 암호를 사용하지 말아야 하는 정당한 이유가 없는 한 모든 암호를 선택합니다 .

주의 Enable No Encryption, Only MD5 Authentication 을 선택하지 마십시오 . 클라이언트 측에 사용 가능한 다른 암호가 없는 경우 서버는 설정을 기본값으로 되돌리며 암호화가 수행되지 않습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- SSL 및 TLS 프로토콜
- SSL 을 사용하여 LDAP 와 통신
- Proxy Server 를 통한 SSL 터널링
- SSL 터널링 구성
- 청취 소켓용 보안 사용 설정
- 전역적 보안 구성

SSL 및 TLS 프로토콜

Proxy Server 는 암호화된 통신용으로 SSL 및 TLS 프로토콜을 지원합니다 . SSL 과 TLS 는 응용 프로그램 종속적인 더 높은 수준의 프로토콜로 , SSL 및 TLS 와 투명하게 레이어로 적용될 수 있습니다 .

SSL 과 TLS 프로토콜은 서버와 클라이언트가 서로를 인증하고 , 인증서를 전송하며 세션 키를 설정하는 등의 작업에 사용되는 다양한 암호를 지원합니다. 클라이언트와 서버는 지원하는 프로토콜 , 암호화 정도에 대한 회사 정책 , 암호화된 소프트웨어의 수출에 대한 정부 규제 등 , 다양한 요인에 따라 지원하는 암호 제품군이 달라집니다 . 여러 기능 중 SSL 과 TLS 핸드셰이크 프로토콜에 따라 서버와 클라이언트가 통신에 사용할 암호 제품군을 선택하는 방식이 결정됩니다 .

SSL 을 사용하여 LDAP 와 통신

Administration Server 가 SSL 을 사용하여 LDAP 와 통신하도록 해야 합니다 .

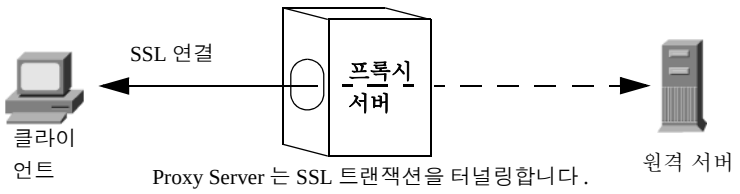
Administration Server 에서 SSL 을 사용하도록 설정하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Global Settings 탭을 누릅니다 .
2. Configure Directory Service 링크를 누릅니다 .
3. 나타나는 표에서 해당 디렉토리 서비스에 대한 링크를 누릅니다 . Configure Directory Service 페이지가 표시됩니다 . LDAP 기반 디렉토리 서비스를 아직 만들지 않았으면 Create New Service of Type 드롭다운 목록에서 LDAP Server 를 선택하고 New 를 눌러 디렉토리 서비스를 구성합니다 . LDAP 기반 디렉토리 서비스에 대해 표시되는 특정 필드에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .
4. 연결에 SSL 을 사용하도록 Yes 를 선택한 다음 Save Changes 를 누릅니다 .

Proxy Server 를 통한 SSL 터널링

Proxy Server(프록시) 를 전방향으로 실행 중이고 클라이언트가 프록시를 통해 보안 서버에 대한 SSL 연결을 요청하면 , 프록시는 보안 서버와 연결한 다음 보안 트랜잭션을 간섭하지 않고 단순히 데이터를 양방향으로 복사합니다 . 이 프로세스를 SSL 터널링이라고 하며 다음 그림에서 설명합니다 .

그림 5-1 SSL 연결을 통해 Proxy Server 는 전송한 데이터를 볼 수 없습니다 .



HTTPS URL 에 터널링된 SSL 을 사용하려면 클라이언트는 SSL 과 HTTPS 를 모두 지원해야 합니다 . HTTPS 는 일반 HTTP 와 SSL 로 구현됩니다 . HTTPS 가 없는 클라이언트도 Proxy Server 의 HTTPS 프록시 기능을 통해 HTTPS 문서에 액세스할 수 있습니다 .

SSL 터널링은 응용 프로그램 수준 (HTTPS) 에 영향을 미치지 않는 더 낮은 수준의 작동입니다 . SSL 터널링은 프록시가 없는 SSL 과 똑같이 안전합니다 . 프록시가 있어도 보안이 손상되거나 SSL 기능이 줄어들지는 않습니다 .

SSL 이 있으면 데이터 스트림이 암호화되므로 프록시가 실제 트랜잭션에 액세스할 수 없습니다 . 이에 따라 액세스 로그가 원격 서버로부터 수신한 상태 코드나 헤더 길이를 나열할 수 없게 됩니다 . 또한 이를 통해 프록시나 기타 제 3 자가 트랜잭션을 엿보는 것을 방지할 수 있습니다 .

프록시가 데이터를 볼 수 없으므로 클라이언트와 원격 서버 간에 작동하는 프로토콜이 SSL 인지 확인할 수 없습니다 . 즉 프록시도 다른 프로토콜이 통과되는 것을 방지할 수 없습니다 . IANA(Internet Assigned Numbers Authority) 에서 할당한 HTTPS 용 포트 443 이나 SNEWS 용 포트 563 과 같이 잘 알려진 SSL 포트로 SSL 연결을 한정해야 합니다 . 다른 포트에서 보안 서버를 실행하는 사이트가 있을 경우 명시적으로 예외를 지정하여 해당 호스트의 다른 포트에 대한 연결을 허용할 수 있습니다 . `connect://.*` 리소스를 사용하여 지정할 수 있습니다 .

SSL 터널링 기능은 사실상 일반적인 SOCKS 와 유사한 프로토콜 독립적인 기능이므로 다른 서비스에 사용할 수도 있습니다 . Proxy Server 는 HTTPS 나 SNEWS 프로토콜뿐 아니라 SSL 을 지원하는 모든 응용 프로그램에 대한 SSL 터널링을 처리할 수 있습니다 .

SSL 터널링 구성

다음 절차는 Proxy Server 가 SSL 을 터널링하도록 구성하는 방법을 설명합니다 .

SSL 터널링을 구성하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 Routing 탭을 누릅니다 .
2. Enable/Disable Proxying 링크를 누릅니다 .
3. 드롭다운 목록에서 `connect://.*.443` 리소스를 선택합니다 . `connect://` 메소드는 내부 프록시의 표기 방식이며 프록시 외부에 존재하지 않습니다 . `connect` 에 대한 자세한 내용은 "SSL 터널링 기술 세부 사항 " (92 페이지) 을 참조하십시오 . 다른 포트로의 연결을 허용하기 위해 템플릿에 유사한 URL 패턴을 사용할 수 있습니다 . 템플릿에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 를 참조하십시오 .

4. Enable Proxying Of This Resource 를 선택하고 OK 를 누릅니다 .

주의 프록시가 잘못 구성되면 telnet 호핑에 SSL 프록시를 악용할 수 있습니다 . 누군가 해당 프록시를 사용하여 telnet 연결이 실제 연결하는 호스트가 아닌 프록시 호스트에서 기원한 것처럼 나타나게 할 수 있습니다 . 이 때문에 꼭 필요하지 않은 포트를 허용해서는 안되며 프록시에서 클라이언트 호스트를 제한하는 액세스 제어를 사용해야 합니다 .

SSL 터널링 기술 세부 사항

내부적으로 SSL 터널링은 대상 호스트 이름과 포트 번호가 있는 `CONNECT` 메소드와 이에 따르는 빈 줄을 매개 변수로 사용합니다 .

```
CONNECT energy.example.com:443 HTTP/1.0
```

Proxy Server 의 성공적인 응답은 다음과 같으며 이 뒤에 빈줄이 따릅니다 .

```
HTTP/1.0 200 Connection established
Proxy-agent: Sun-Java-System-Web-Proxy-Server/4.0
```

그러면 클라이언트와 원격 서버 간에 연결이 설정되고 한 쪽이 연결을 종료할 때까지 데이터를 양방향으로 전송할 수 있습니다 .

내부적으로 URL 패턴에 따른 일반적인 구성 메커니즘을 활용하려면 다음과 같이 호스트 이름 및 포트 번호 (`energy.example.com:443`) 를 URL 에 자동 매핑해야 합니다 .

```
connect://energy.example.com:443
```

`connect://` 는 Proxy Server 가 구성을 용이하게 하고 다른 URL 패턴과 일치하도록 하기 위해 사용하는 내부 노테이션에 불과합니다 . Proxy Server 외부에는 `connect` URL 이 없으므로 Proxy Server 가 네트워크에서 이런 URL 을 수신하면 이를 잘못되었다고 판단하고 이 요청에 대한 서비스를 거절합니다 .

청취 소켓용 보안 사용 설정

다음과 같이 서버의 청취 소켓을 보안할 수 있습니다 .

- 보안 기능 사용
- 청취 소켓용 서버 인증서 선택
- 암호 선택

보안 기능 사용

청취 소켓용으로 다른 보안 설정을 구성하기 전에 반드시 보안을 사용하도록 설정해야 합니다 . 보안은 새 청취 소켓을 만들거나 기존 청취 소켓을 편집할 때 사용 설정할 수 있습니다 .

청취 소켓을 만들 때 보안을 실행하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Add Listen Socket 링크를 누릅니다 .
3. 필요한 정보를 입력합니다 . 보안을 사용하려면 Security 드롭다운 목록에서 Enabled 를 선택하고 OK 를 누릅니다 . 서버 인증서가 설치되어 있지 않으면 Disabled 만 선택할 수 있습니다 . 특정 설정에 관한 자세한 내용은 온라인 도움말을 참조하십시오 .

참고 청취 소켓을 만든 후 Edit Listen Sockets 링크를 사용하여 보안 설정을 구성합니다 .

청취 소켓을 편집할 때 보안을 실행하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Edit Listen Sockets 링크를 누릅니다 .
3. 편집하려는 청취 소켓에 해당하는 링크를 누릅니다 .
4. 보안을 사용하려면 Security 드롭다운 목록에서 Enabled 를 선택하고 OK 를 누릅니다 . 서버 인증서가 설치되어 있지 않으면 Disabled 만 선택할 수 있습니다 .

청취 소켓용 서버 인증서 선택

Administration Server 나 Server Manager 에서 청취 소켓을 구성하여 요청 및 설치한 서버 인증서를 사용할 수 있습니다 .

참고 최소한 하나의 인증서를 설치해야 합니다 .

청취 소켓용 서버 인증서를 선택하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Edit Listen Sockets 링크를 누릅니다 .
3. 편집하려는 사용자 정의 리소스에 해당하는 링크를 누릅니다 .

4. 보안을 사용하려면 Security 드롭다운 목록에서 Enabled 를 선택하고 OK 를 누릅니다 . 서버 인증서가 설치되어 있지 않으면 Disabled 만 선택할 수 있습니다 .
5. Enabled 를 선택하고 OK 를 누른 후 , 청취 소켓에 대한 Server Certificate Name 드롭다운 목록에서 서버 인증서를 선택한 후 OK 를 누릅니다 .

암호 선택

Proxy Servers 의 보안을 유지하려면 SSL 을 사용하도록 설정해야 합니다 . SSL 2.0, SSL 3.0 및 TLS 암호화 프로토콜을 선택할 수 있으며 다양한 암호 제품군을 선택할 수 있습니다 . SSL 및 TLS 는 Administration Server 에 대한 청취 소켓에서 사용하도록 설정할 수 있습니다 . Server Manager 에 대한 청취 소켓에서 SSL 및 TLS 를 사용하도록 설정하면 특정 서버 인스턴스에 대한 보안 기본 설정이 구성됩니다 . 최소한 하나의 인증서를 설치해야 합니다 .

참고 청취 소켓에서 SSL 을 실행하도록 설정하는 것은 역방향 프록시 시나리오에만 적용됩니다 . 즉 Proxy Server 가 역방향 프록시를 수행하도록 구성되었을 때만 가능합니다 .

기본 설정의 경우 가장 많이 사용되는 암호를 허용합니다 . 특정 암호를 사용하면 안 되는 충분한 이유가 있지 않는 한 , 모두 선택해야 합니다 . 특정 암호에 대한 자세한 내용은 SSL 개요를 참조하십시오 .

TLS Rollback 에 대한 기본 및 권장 설정은 Enabled 입니다 . 이렇게 구성하면 서버가 중간개입자 (man-in-the-middle) 버전 롤백 공격 시도를 감지할 수 있습니다 . TLS 표준을 잘못 구현한 일부 클라이언트와의 상호 운용성을 위하여 이 설정을 Disabled 로 해야 하는 경우도 있습니다 .

TLS 롤백을 사용하지 않으면 연결이 버전 롤백 공격에 취약한 상태로 남는다는 점을 유의하십시오 . 버전 롤백 공격은 제 3 자가 클라이언트와 서버로 하여금 SSL 2.0 등의 덜 안전한 이전 프로토콜을 사용하도록 하는 기법입니다 . SSL 2.0 에는 알려진 결함이 있으므로 버전 롤백 공격을 감지하지 못하는 경우 제 3 자가 암호화된 연결을 가로채어 해독할 수 있습니다 .

SSL 및 TLS 를 사용하도록 설정하려면 다음과 같이 하십시오 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .

2. Edit Listen Sockets 링크를 누른 다음 편집할 청취 소켓에 대한 링크를 누릅니다 . 보안 청취 소켓에서는 사용 가능한 암호 설정이 표시됩니다 .

참고 청취 소켓에서 보안을 사용하지 않는 경우 SSL 및 TLS 정보를 나열할 수 없습니다 . 암호를 사용하려면 선택한 청취 소켓에서 보안이 사용되도록 설정해야 합니다 . 자세한 내용은 " 청취 소켓용 보안 사용 설정 " (92 페이지) 을 참조하십시오 .

3. 필요한 암호화 설정에 해당하는 확인란을 선택하고 OK 를 누릅니다 .

참고 Netscape Navigator 6.0 의 경우 TLS 및 SSL 3.0 을 모두 선택합니다 . TLS Rollback 에서는 TLS 를 선택하고 SSL 3.0 과 SSL 2.0 을 사용하지 않도록 설정합니다 .

서버에서 SSL 을 사용 설정하면 URL 은 `http` 가 아닌 `https` 를 사용합니다 . SSL 을 사용하는 서버의 문서를 가리키는 URL 의 형식은 다음과 같습니다 .

`https://`*servername*`.`*domain*`.`*dom*`:`*port*

예 : `https://admin.example.com:443`

기본 보안 HTTP 포트 (443) 를 사용하는 경우 URL 에 포트 번호를 입력하지 않아도 됩니다 .

전역적 보안 구성

SSL 을 사용하는 서버를 설치하면 `magnus.conf` 파일 (서버의 기본 구성 파일) 에 전역 보안 매개 변수용 지시문 항목이 만들어집니다 .

SSL 구성 파일 지시문에 대한 값을 설정하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스합니다 .
2. 구성하려는 청취 소켓에서 보안을 사용하는지 확인하십시오 . 자세한 내용은 " 청취 소켓용 보안 사용 설정 " (92 페이지) 을 참조하십시오 .
3. 직접 `magnus.conf` 파일을 편집하여 다음 설정에 대한 값을 입력합니다 .
   - `SSLSessionTimeout`
   - `SSLCacheEntries`
   - `SSL3SessionTimeout`

SSL 구성 파일 지시문의 설명은 다음과 같습니다 . magnus.conf 에 대한 자세한 내용은 Proxy Server Configuration File Reference 를 참조하십시오 .

SSLSessionTimeout

SSLSessionTimeout 은 SSL 2.0 세션 캐시를 제어합니다 .

구문

SSLSessionTimeout *seconds*

seconds 는 캐시된 SSL 세션의 유효 시간을 초 단위로 나타냅니다 . 기본값은 100 입니다 . SSLSessionTimeout 지시문이 지정되면 초 단위 값은 자동으로 5 에서 100 초 사이로 제한됩니다 .

SSLCacheEntries

캐시할 수 있는 SSL 세션의 수를 지정합니다 .

SSL3SessionTimeout

SSL3SessionTimeout 지시문은 SSL 3.0 및 TLS 세션 캐시를 제어합니다 .

구문

SSL3SessionTimeout *seconds*

seconds 는 캐시된 SSL 3.0 세션의 유효 시간을 초 단위로 나타냅니다 . 기본값은 86400(24 시간) 입니다 . SSL3SessionTimeout 지시문이 지정되면 초 단위 값은 자동으로 5 에서 86400 초 사이로 제한됩니다 .

외부 암호화 모듈 사용

Proxy Server 는 다음과 같이 스마트 카드나 토큰 링 등의 외부 암호화 모듈을 사용하는 방법을 지원합니다 .

- PKCS #11
- FIPS 140

FIPS 140 암호화 표준을 사용하기 전에 PKCS #11 모듈을 추가해야 합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- PKCS #11 모듈 설치

- FIPS 140 표준

PKCS #11 모듈 설치

Proxy Server 는 PKCS(Public Key Cryptography Standard) #11 을 지원합니다 . 이는 SSL 과 , PKCS #11 모듈 사이의 통신용으로 사용되는 인터페이스를 정의합니다 . PKCS #11 모듈은 SSL 하드웨어 가속기에 대한 표준 기반 연결용으로 사용됩니다 . 외부 하드웨어 가속기용으로 가져온 인증서와 키는 `secmod.db` 에 저장되며 , 이 파일은 PKCS#11 이 설치될 때 생성됩니다 . 이 파일은 *server_root*`/alias` 디렉토리에 있습니다 .

modutil 을 사용하여 PKCS #11 모듈 설치 s

`modutil` 도구를 사용하여 PKCS#11 모듈을 `.jar` 파일 또는 개체 파일 형식으로 설치할 수 있습니다 .

modutil 을 사용하여 PKCS #11 모듈을 설치하려면 다음과 같이 합니다 .

1. Administration Server 를 포함한 모든 서버를 중지합니다 .
2. 데이터베이스가 있는 *server_root*`/alias` 디렉토리로 이동합니다 .
3. *server_root*`/bin/proxy/admin/bin` 을 PATH 에 추가합니다 .
4. *server_root*`/bin/proxy/admin/bin` 에서 `modutil` 을 찾습니다 .
5. 환경을 설정합니다 . 예 :
   - UNIX: `setenv`

     `LD_LIBRARY_PATH` *server_root*`/bin/proxy/lib:${LD_LIBRARY_PATH}`
   - Windows 의 경우 이를 `PATH` 에 추가합니다 .

     `LD_LIBRARY_PATH` *server_root*`/bin/proxy/bin`

     다음에서 컴퓨터에 대한 PATH 를 찾을 수 있습니다 .
     server_root`/proxy-admserv/start`.
6. 다음 명령을 입력합니다 . `modutil` 옵션이 나열됩니다 .
7. 필요한 조치를 수행합니다 .

   예를 들어 UNIX 에서 PKCS #11 모듈을 추가하려면 다음을 입력합니다 .

   `modutil -add` (*name of PKCS#11 file*) `-libfile` (*your libfile for PKCS #11*) `-nocertdb -dbdir` (*사용하는 db 디렉토리*

pk12util 사용

pk12util 을 사용하면 내부 데이터베이스에서 인증서와 키를 내보내고 이를 내부 또는 외부 PKCS #11 모듈로 가져올 수 있습니다 . 언제라도 인증서와 키를 내부 데이터베이스로 내보낼 수 있으나 , 외부 토큰의 경우 대부분 인증서와 키를 내보낼 수 없습니다 . 기본적으로 pk12util 은 cert8.db 와 key3.db 라는 이름의 인증서 및 키 데이터베이스를 사용합니다 .

pk12util 을 사용하여 내보내기

내부 데이터베이스에서 인증서와 키를 내보내려면 다음과 같이 합니다 .

1. 데이터베이스가 있는 *server_root*/alias 디렉토리로 이동합니다 .
2. *server_root*/bin/proxy/admin/bin 을 PATH 에 추가합니다 .
3. *server_root*/bin/proxy/admin/bin 에서 pk12util 을 찾습니다 .
4. 환경을 설정합니다 . 예 :
   - UNIX: setenv

     LD_LIBRARY_PATH/*server_root*/bin/proxy/lib:${LD_LIBRARY_PATH}
   - Windows 의 경우 이를 PATH 에 추가합니다 .

     LD_LIBRARY_PATH *server_root*/bin/proxy/bin

     다음에서 컴퓨터에 대한 PATH 를 찾을 수 있습니다 .
     server_root/proxy-admserv/start
5. 다음 명령을 입력합니다 . pk12util 옵션이 나열됩니다 .
6. 필요한 조치를 수행합니다 .

   예를 들어 UNIX 의 경우 다음을 입력합니다 .

   ```
   pk12util -o certpk12 -n Server-Cert [-d /server/alias] [-P https-test-host]
   ```
7. 데이터베이스 암호를 입력합니다 .
8. pkcs12 비밀 번호를 입력합니다 .

pk12util 을 사용하여 가져오기

내부 또는 외부 PKCS #11 모듈로 인증서와 키를 가져오려면 다음과 같이 합니다 .

1. 데이터베이스가 있는 *server_root*/alias 디렉토리로 이동합니다 .
2. *server_root*/bin/proxy/admin/bin 을 PATH 에 추가합니다 .
3. *server_root*/bin/proxy/admin/bin 에서 pk12util 을 찾습니다 .

4. 환경을 설정합니다 . 예 :
   - UNIX: `setenv`

     `LD_LIBRARY_PATH/`*server_root*`/bin/proxy/lib:${LD_LIBRARY_PATH}`
   - Windows 의 경우 이를 `PATH` 에 추가합니다 .

     `LD_LIBRARY_PATH` *server_root*`/bin/proxy/bin`

     다음에서 컴퓨터에 대한 PATH 를 찾을 수 있습니다 .
     server_root`/proxy-admserv/start`.
5. 다음 명령을 입력합니다 . `pk12util` 옵션이 나열됩니다 .
6. 필요한 조치를 수행합니다 .

   예를 들어 UNIX 의 경우 다음을 입력합니다 .

   ```
   pk12util -i pk12_sunspot [-d certdir][-h "nCipher"][-P
   https-jones.redplanet.com-jones-]
   ```

   `-P` 는 반드시 `-h` 뒤에 있고 마지막 인수여야 합니다 .

   대문자와 인용 부호 사이의 공백을 포함하여 토큰 이름을 정확히 입력합니다 .
7. 데이터베이스 암호를 입력합니다 .
8. `pkcs12` 비밀 번호를 입력합니다 .

외부 인증서를 사용하여 서버 시작

서버용 인증서를 외부 PKCS #11 모듈 (하드웨어 가속기 등) 에 설치한 경우 , `server.xml` 파일을 편집하거나 아래에 설명한 것과 같이 인증서 이름을 지정하지 않으면 서버가 인증서를 사용하여 시작할 수 없습니다 .

서버는 항상 이름이 Server-Cert 인 인증서를 사용하여 시작하려고 합니다 . 그러나 외부 PKCS #11 모듈의 인증서는 모듈 토큰 이름 중 하나를 해당 식별자에 포함합니다 . 예를 들어 외부 스마트카드 판독기에 설치된 서버 인증서가 `smartcard0` 인 경우 , 이름은 `smartcard0:Server-Cert` 가 됩니다 .

외부 모듈에 설치된 인증서를 사용하여 서버를 시작하려면 서버가 실행되는 청취 소켓용 인증서 이름을 지정해야 합니다 .

청취 소켓용 인증서 이름 선택

청취 소켓용 인증서 이름을 선택하려면 다음과 같이 합니다 .
청취 소켓에서 보안이 사용되지 않는 경우에는 인증서 정보가 표시되지 않습니다 . 청취 소켓용 인증서 이름을 선택하려면 반드시 우선 청취 소켓에서 보안을 사용하도록 설정해야 합니다 . 자세한 내용은 " 청취 소켓용 보안 사용 설정 " (92 페이지) 을 참조하십시오 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Edit Listen Sockets 링크를 누릅니다 .
3. 인증서에 연결할 청취 소켓에 해당하는 링크를 누릅니다 .
4. 해당 청취 소켓용 Server Certificate Name 드롭다운 목록에서 서버 인증서를 선택하고 OK 를 누릅니다 . 이 목록에는 설치된 모든 내부 및 외부 인증서가 표시됩니다 .

또한 `server.xml` 파일을 직접 편집하여 서버가 해당 인증서를 사용하여 시작하도록 할 수 있습니다 . `SSLPARAMS` 의 `servercertnickname` 을 다음으로 변경합니다 .

```
$TOKENNAME:Server-Cert
```

`$TOKENNAME` 용으로 사용할 값을 찾으려면 서버의 Security 탭으로 이동하여 Manage Certificate 링크를 선택합니다 . Server-Cert 가 저장된 외부 모듈로 로그인하면 해당 인증서가 `$TOKENNAME:$NICKNAME` 형식의 목록에 표시됩니다 .

참고 신뢰 데이터베이스를 만들지 않은 경우에는 외부 PKCS #11 모듈에서 인증서를 요청하거나 설치하면 자동으로 만들어집니다 . 만들어진 기본 데이터베이스에는 비밀 번호가 없으며 액세스할 수 없습니다 . 외부 모듈은 작동하지만 서버 인증서를 요청하거나 설치할 수는 없습니다 . 기본 데이터베이스가 비밀 번호 없이 만들어진 경우 Create Database 페이지의 Security 탭에서 비밀 번호를 설정합니다 .

FIPS 140 표준

PKCS #11 API 를 사용하면 암호화 작업을 수행하는 소프트웨어 또는 하드웨어 모듈과 통신할 수 있습니다 . PKCS #11 을 Proxy Server 에 설치한 후에는 서버가 FIPS-140(Federal Information Processing Standards) 과 호환되도록 구성할 수 있습니다 . 이 라이브러리는 오직 SSL 3.0 에만 포함되어 있습니다 .

FIPS-140 을 사용하도록 설정하려면 다음과 같이 합니다 .

1. FIPS 140 의 설명을 따라 플러그인을 설치합니다 .
2. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
3. Edit Listen Sockets 링크를 누릅니다 . 보안 청취 소켓에 대해 Edit Listen Sockets 페이지에 사용 가능한 보안 설정이 표시됩니다 .

참고 FIPS 140 을 사용하려면 선택한 청취 소켓에서 보안이 사용되도록 설정해야 합니다 . 자세한 내용은 " 청취 소켓용 보안 사용 설정 " (92 페이지) 을 참조하십시오 .

4. SSL Version 3 드롭다운 목록에서 Enabled 를 아직 선택하지 않았으면 이를 선택합니다 .
5. 적합한 FIPS-140 암호 제품군을 선택하고 OK 를 누릅니다 .
   - 168 비트 암호화와 SHA 인증이 포함된 Triple DES(FIPS) 사용
   - 56 비트 암호화와 SHA 인증이 포함된 DES(FIPS) 사용

클라이언트 보안 요구 사항 설정

서버 보안 단계를 모두 수행 한 후 클라이언트에 대한 추가 보안 요구 사항을 설정할 수 있습니다 .

SSL 연결에 클라이언트 인증이 꼭 필요한 것은 아니나 암호화된 정보가 확실히 정확한 대상에게 전달되도록 하는 데 일조할 수 있습니다 . 역방향 프록시에서 클라이언트 인증을 사용하여 컨텐트 서버가 권한 없는 프록시나 클라이언트와 확실히 정보를 공유하지 않도록 할 수 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

클라이언트 인증 필수화

Administration Server 용 청취 소켓을 사용하도록 설정하고 각 서버 인스턴스가 클라이언트 인증을 요청하도록 할 수 있습니다 . 클라이언트 인증을 사용하면 서버가 쿼리에 대한 응답을 보내기 전에 클라이언트에 인증서를 요구합니다 .

Proxy Server 는 클라이언트 인증서에 있는 CA 와 클라이언트 인증서 서명용으로 신뢰된 CA 를 비교하여 클라이언트 인증서를 인증합니다 . 클라이언트 인증서 서명용으로 신뢰된 CA 의 목록은 Manage Certificates 페이지의 Security 탭에서 확인할 수 있습니다 .

Proxy Server 가 신뢰된 CA 의 인증서를 보유하지 않은 클라이언트를 거부하도록 구성할 수 있습니다 . CA 를 승인 또는 거부하려면 반드시 해당 CA 용 클라이언트 신뢰를 설정해야 합니다 . 자세한 내용은 " 인증서 관리 " (86 페이지) 를 참조하십시오 .

Proxy Server 는 인증서의 유효 기간이 만료된 경우 오류를 기록하고 인증서를 거부하며 클라이언트에게 메시지를 반송합니다 . 또한 Manage Certificates 페이지에서 만기된 인증서를 확인할 수 있습니다 .

서버가 인증서 클라이언트에서 정보를 수집하여 이를 LDAP 디렉토리에 있는 사용자 항목과 비교하도록 구성할 수 있습니다 . 이렇게 하면 클라이언트의 인증서가 유효하며 LDAP 디렉토리에 항목이 보관되도록 합니다 . 또한 클라이언트 인증서가 LDAP 디렉토리의 항목 중 하나와 일치되도록 합니다 . 이에 대한 방법은 " 클라이언트 인증서와 LDAP 매핑 " (105 페이지) 을 참조하십시오 .

클라이언트 인증서를 액세스 제어와 조합할 수 있으므로 신뢰된 CA 의 요구 사항 이외에 인증서에 연결된 사용자는 반드시 액세스 제어 규칙 (ACL) 과 일치되어야 합니다 . 자세한 내용은 " 액세스 제어 파일 사용 " (156 페이지) 을 참조하십시오 .

클라이언트 인증을 요구하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Edit Listen Sockets 링크를 누릅니다 .
3. 클라이언트 인증을 요구할 청취 소켓에 대한 링크를 누릅니다 .
4. Client Authentication 드롭다운 목록을 사용하여 청취 소켓에 대한 클라이언트 인증을 요구하고 OK 를 누릅니다 .

역방향 프록시에서의 클라이언트 인증

역방향 프록시에서는 다음 시나리오 중 하나에 따라 클라이언트 인증을 구성할 수 있습니다 .

- **Proxy-Authenticates-Client** 이 시나리오에서는 사용 가능한 인증서가 있는 모든 클라이언트에 대한 액세스를 허용하거나 , 사용 가능한 인증서가 있고 Proxy Server 의 ACL 에서 인식된 사용자인 클라이언트에 대해서만 액세스를 허용할 수 있습니다 .
- **Content-Server-Authenticates-Proxy** 이 시나리오에서는 컨텐트 서버가 실제로 Proxy Server 와 연결 중이며 다른 서버와는 연결하고 있지 않음을 확인할 수 있습니다 .
- **Proxy-Authenticates-Client and Content-Server-Authenticates-Proxy** 이 시나리오는 역방향 프록시에서 최고의 보안과 인증 수준을 제공합니다 .

이러한 시나리오 구성 방법에 대한 자세한 내용은 " 역방향 프록시에서의 클라이언트 인증 설정 " (103 페이지) 을 참조하십시오 .

역방향 프록시에서의 클라이언트 인증 설정

보안 역방향 프록시에서 클라이언트 인증을 사용하면 연결의 보안을 더욱 강화할 수 있습니다 . 다음 지침은 선택한 시나리오에 따라 클라이언트 인증을 구성하는 방법을 설명합니다 .

참고	각 시나리오에서는 보안 클라이언트에서 프록시 연결과 보안 프록시에서 컨텐트 서버 연결을 모두 사용한다고 가정합니다 .

Proxy-Authenticates-Client

Proxy-Authenticates-Client 시나리오를 구성하려면 다음과 같이 합니다 .

1. 제 14 장 , 313 페이지의 " 역방향 프록시 사용 " 의 " 역방향 프록시 설정 " 에서 제공하는 클라이언트에서 프록시 연결 지침 및 보안 프록시에서 컨텐트 서버 시나리오에 따릅니다 .
2. 서버 인스턴스에 대한 Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
3. Edit Listen Sockets 링크를 누른 다음 나타나는 표에서 원하는 청취 소켓에 대한 링크를 누릅니다 . Add Listen Socket 링크를 사용하여 청취 소켓을 구성 및 추가합니다 .

4. 클라이언트 인증 요구 사항을 지정합니다 .

   유효한 인증서가 있는 모든 사용자에게 액세스를 허용하려면 다음과 같이 합니다 .

   - Security 부분의 Client Authentication 설정을 사용하여 이 청취 소켓에 클라이언트 인증을 요구하도록 설정합니다 . 서버 인증서가 설치되어 있지 않으면 이 설정이 나타나지 않습니다 .

   유효한 인증서가 있고 ACL 에서 승인된 사용자에 대해서만 액세스를 허용하려면 다음과 같이 합니다 .

   a. Security 부분에서 Client Authentication 설정을 off 로 유지합니다 . 서버 인증서가 설치되어 있지 않으면 이 설정이 나타나지 않습니다 .

   b. 이 서버 인스턴스에 대한 Server Manager 탭에서 Administer Access Control 링크를 누릅니다 .

   c. ACL 을 선택한 후 Edit 버튼을 누릅니다 . Access Control Rules For 페이지가 표시됩니다 (요청할 경우 인증 필요).

   d. 액세스 제어를 실행합니다 (Access control Is On 확인란을 선택하지 않았으면 선택).

   e. Proxy Server 가 역방향 프록시로 인증하도록 설정합니다 . 자세한 내용은 " 역방향 프록시 설정 " (319 페이지) 을 참조하십시오 .

   f. 원하는 액세스 제어 규칙에 대한 Rights 링크를 누르고 , 아래쪽 창에서 액세스 권한을 지정한 다음 Update 를 눌러 이 항목을 업데이트합니다 .

   g. Users/Groups 링크를 누릅니다 . 아래쪽 창에서 사용자와 그룹을 지정하고 , 인증 방법으로 SSL 을 선택한 후 Update 를 눌러 입력을 업데이트합니다 .

   h. 위쪽 창에서 Submit 을 눌러 입력을 저장합니다 .

   액세스 제어에 대한 자세한 내용은 제 8 장 , 147 페이지의 " 서버 액세스 제어 " 를 참조하십시오 .

Content Server-Authenticates-Proxy

Content Server-Authenticates-Proxy 시나리오를 구성하려면 다음과 같이 합니다 .

1. " 역방향 프록시 설정 " (319 페이지) 의 보안 클라이언트에서 프록시 및 보안 프록시에서 컨텐트 서버 시나리오 구성을 위한 지침에 따릅니다 .
2. 컨텐트 서버에서 클라이언트 인증을 실행합니다 .

참고 비보안 클라이언트의 Proxy Server 연결 및 컨텐트 서버에 대한 보안 연결을 수행하고 컨텐트 서버가 Proxy Server 를 인증하도록 이 시나리오를 수정할 수 있습니다 . 이를 위해 다음 절차의 설명에 따라 암호화를 중지하고 프록시가 인증서만 초기화하도록 지시합니다 .

Proxy-Authenticates-Client and Content Server-Authenticates-Proxy

Proxy-Authenticates-Client and Content Server-Authenticates-Proxy 시나리오를 구성하려면 다음과 같이 합니다 .

1. "Proxy-Authenticates-Client" (103 페이지) 의 Proxy-Authenticates-Client 시나리오 구성 지침을 따르십시오 .
2. 컨텐트 서버에서 클라이언트 인증을 실행합니다 .

클라이언트 인증서와 LDAP 매핑

이 부분에서는 Proxy Server 가 클라이언트 인증서를 LDAP 디렉토리의 항목과 매핑하는 데 사용하는 프로세스에 대해 설명합니다 .

서버가 클라이언트의 요청을 수신하면 이를 처리하기 전에 클라이언트의 인증서를 요구합니다 . 클라이언트에 따라 서버에 요청과 함께 클라이언트를 전송하는 경우도 있습니다 .

참고 클라이언트 인증서와 LDAP 를 매핑하기 전에 필수 ACL 을 구성해야 합니다 . 자세한 내용은 제 8 장 , 147 페이지의 " 서버 액세스 제어 " 를 참조하십시오 .

서버는 CA 를 Administration Server 에 있는 신뢰 CA 의 목록과 대조합니다 . 일치 항목과 없으면 Proxy Server 는 연결을 종료합니다 . 일치 항목이 있으면 서버가 요청 처리를 계속합니다 .

신뢰 CA 에서 발행한 인증서임을 확인한 후 , 서버는 다음과 같이 인증서를 LDAP 항목과 매핑합니다 .

- 클라이언트 인증서에 있는 발행자와 대상 DN 을 LDAP 디렉토리의 분기점과 매핑합니다 .
- LDAP 디렉토리에 클라이언트 인증서의 대상 (최종 사용자) 에 대한 정보와 일치하는 항목이 있는지 검색합니다 .

- (선택) 클라이언트 인증서를 DN 에 해당하는 LDAP 항목 중 하나와 확인합니다 .

서버는 certmap.conf 라는 인증서 매핑 파일을 사용하여 LDAP 검색 수행 방법을 결정합니다 . 서버는 매핑 파일에 따라 클라이언트 인증서에서 가져올 값 (최종 사용자의 이름 , 전자 메일 주소 등) 을 결정합니다 . 서버는 이들 값을 사용하여 LDAP 디렉토리에서 사용자 항목을 검색하지만 , 우선 서버가 LDAP 디렉토리에서 검색을 시작할 위치를 결정해야 합니다 . 서버는 또한 인증서 매핑 파일에서 시작 위치를 알 수 있습니다 .

서버가 검색을 시작할 위치와 검색할 항목을 결정하면 (위의 1 지점) LDAP 디렉토리에서 검색을 수행합니다 (2 지점). 일치 항목이 없거나 일치 항목이 여러 개인 경우 매핑이 인증서 확인으로 설정되지 *않고* 검색은 실패합니다 .

예상되는 검색 결과의 목록은 다음 표와 같습니다 . 예상되는 행동을 ACL 에 지정할 수 있습니다 . 예를 들어 인증서 일치가 실패하면 Proxy Server 가 해당 사용자만을 허용하도록 지정할 수 있습니다 . ACL 기본 설정 방법에 대한 자세한 내용은 " 액세스 제어 파일 사용 " (156 페이지) 을 참조하십시오 .

표 5-1 LDAP 검색 결과

LDAP 검색 결과	인증서 검증 ON	인증서 검증 OFF
검색된 항목 없음	인증 실패	인증 실패
정확히 한 개 항목 일치	인증 실패	인증 성공
여러 항목 일치	인증 실패	인증 실패

서버가 LDAP 디렉토리에서 일치 항목과 인증서를 찾으면 해당 정보를 사용하여 트랜잭션을 처리할 수 있습니다 . 예를 들어 어떤 서버에서는 인증서 LDAP 매핑을 사용하여 서버에 대한 액세스를 결정합니다 .

certmap.conf 파일 사용

인증서 매핑에 따라 서버가 LDAP 디렉토리에서 사용자 항목을 찾는 방법이 결정됩니다 . certmap.conf 를 사용하여 이름으로 명시된 인증서를 LDAP 항목과 일치시키는 방법을 구성할 수 있습니다 . 이 파일을 편집하고 항목을 추가하여 LDAP 디렉토리의 조직을 검색하고 사용자에게 부여할 인증서 목록을 표시할 수 있습니다 . 사용자는 사용자 ID, 전자 메일 또는 subjectDN 에서 사용되는 다른 값을 기준으로 인증될 수 있습니다 . 특히 , 매핑 파일에는 다음의 정보가 정의됩니다 .

- 서버가 검색을 시작하는 LDAP 트리 내 위치

- LDAP 디렉토리에서 항목을 검색할 때 서버가 검색 범주로 사용할 인증서 속성
- 서버가 추가의 검증 과정을 수행할 것인지의 여부

인증서 매핑 파일의 위치는 다음과 같습니다 .

server_root/userdb/certmap.conf

파일에는 하나 이상의 이름 매핑이 있으며 , 각각의 매핑은 서로 다른 CA 에 적용됩니다 . 매핑의 구문은 다음과 같습니다 .

certmap *name issuerDN*
name:*property* [*value*]

첫 번째 줄은 항목의 이름과 CA 인증서에 있는 고유 이름을 구성하는 속성을 지정합니다 . *name* 은 임의적이며 사용자가 원하는 값으로 정의할 수 있습니다 . 그러나 *issuerDN* 은 반드시 클라이언트 인증서를 발행한 CA 의 발행자 DN 과 정확히 일치해야 합니다 . 예를 들어 아래의 발행자 DN 줄의 차이는 단지 속성을 구분하는 공백이지만 서버는 이 두 항목을 서로 다른 것으로 처리합니다 .

```
certmap sun1 ou=Sun Certificate Authority,o=Sun,c=US
certmap sun2 ou=Sun Certificate Authority, o=Sun, c=US
```

팁 Sun Java System Directory Server 를 사용하여 발행자 DN 을 대조하는 데 문제가 발생할 경우 , Directory Server 오류 로그에 유용한 정보가 있는지 확인하십시오 .

이름 매핑의 두 번째 및 이후 줄은 등록 정보를 값과 매핑합니다 . certmap.conf 파일에는 인증서 API 를 사용하여 사용자 정의할 수 있는 6 가지 등록 정보가 있습니다 .

- DNComps 는 쉼표로 분리된 속성 목록으로 , LDAP 디렉토리에서 사용자 정보 (즉 , 클라이언트 인증서의 소유자) 와 일치하는 항목 검색을 시작할 위치를 결정하는 데 사용합니다 . 서버는 클라이언트 인증서에서 이들 속성 값을 수집하고 값을 사용하여 LDAP DN 을 구성합니다 . 이후 LDAP 디렉토리에서 서버가 검색을 시작할 위치를 결정합니다 . 예를 들어 , DNComps 를 DN 의 o 와 c 속성을 사용하도록 설정한 경우 , 서버는 LDAP 디렉토리의 o=*org*, c= *country* 항목으로부터 검색을 시작합니다 . 여기서 *org* 및 *country* 는 인증서의 DN 값으로 교체할 수 있습니다 .

  다음 상황에 유의하십시오 .

  - 매핑에 DNComps 항목이 없는 경우에는 서버는 CmapLdapAttr 설정을 사용하거나 클라이언트 인증서에 있는 전체 대상 DN(즉 , 최종 사용자의 정보) 을 사용합니다 .
  - DNComps 항목은 있으나 값이 없는 경우 , 서버는 전체 LDAP 트리에서 필터와 일치하는 항목을 검색합니다 .

- FilterComps 는 쉼표로 분리된 속성 목록으로 클라이언트 인증서에 있는 사용자의 DN 에서 정보를 수집하여 필터를 만드는 데 사용합니다 . 서버는 이 속성의 값을 사용하여 LDAP 디렉토리에서 항목을 대조하는 데 사용할 검색 기준을 구성합니다 . LDAP 에서 인증서에서 수집한 사용자의 정보와 일치하는 항목이 하나 이상 검색되는 경우 검색은 성공적이며 서버는 선택적으로 검증을 수행합니다 .

  예를 들어 FilterComps 가 전자 메일과 사용자 아이디 속성을 사용하도록 설정된 경우 (FilterComps=e, uid), 서버는 디렉토리에서 전자 메일과 사용자 ID 값이 클라이언트 인증서에서 수집한 사용자 정보와 일치하는 항목을 검색합니다 . 전자 메일 주소와 사용자 ID 는 일반적으로 디렉토리 내에서 고유하므로 좋은 필터가 될 수 있습니다 . LDAP 데이터베이스에서 오직 하나의 항목만 검색하려면 필터가 구체적이어야 합니다 .

  필터용 속성 이름은 LDAP 디렉토리가 아닌 인증서의 속성 이름이어야 합니다 . 예를 들어 일부 인증서에는 사용자의 전자 메일 주소에 대한 속성으로 e 가 있는 반면 LDAP 에서 이 속성의 이름은 mail 입니다 .

  다음 표는 x509v3 인증서 속성을 나열합니다 .

표 5-2 x509v3 인증서 속성

속성	설명
c	국가
o	조직
cn	공통 이름
l	위치
st	주
ou	조직 단위
uid	UNIX/Linux 사용자 ID
email	전자 메일 주소

- 서버는 verifycert 의 설정에 따라 클라이언트의 인증서를 LDAP 디렉토리에서 검색된 인증서와 비교할지 여부를 결정합니다 . on 및 off 등 , 두 가지 값을 갖습니다 . 이 등록 정보는 LDAP 디렉토리에 인증서가 있는 경우에만 사용하십시오 . 이 기능은 최종 사용자의 인증서가 유효하며 취소되지 않았는지 확인하는 데 유용합니다 .

- `CmapLdapAttr` 은 LDAP 디렉토리에 있는 속성 이름으로 사용자에게 속한 모든 인증서의 대상 DN 을 포함합니다 . 이 등록 정보의 기본값은 `certSubjectDN` 입니다 . 이 등록 정보는 표준 LDAP 속성이 아니므로 이 등록 정보를 사용하려면 반드시 LDAP 스키마를 확장해야 합니다 . 자세한 내용은 SSL 개요를 참조하십시오 .

  `certmap.conf` 에 이 등록 정보가 있으면 서버는 전체 LDAP 디렉토리에서 속성이 대상의 전체 DN(인증서에서 가져온 DN) 과 일치하는 항목을 검색합니다 . 일치하는 항목이 없으면 서버는 `DNComps` 및 `FilterComps` 매핑을 사용하여 검색을 다시 시도합니다 .

  인증서를 LDAP 항목과 일치시키는 이러한 방식의 접근은 `DNComps` 와 `FilterComps` 를 사용하여 항목을 일치시키는 것이 어려울 때 유용합니다 .

- `Library` 는 공유 라이브러리나 DLL 의 경로 이름을 값으로 갖는 속성입니다 . 인증서 API 를 사용하여 고유의 속성을 만든 경우에만 이 속성을 사용하십시오 .

- `InitFn` 은 값이 사용자 정의 라이브러리에 있는 `init` 함수의 이름인 등록 정보입니다 . 인증서 API 를 사용하여 고유의 속성을 만든 경우에만 이 속성을 사용하십시오 .

이 등록 정보에 대한 자세한 내용은 " 매핑 예제 " (109 페이지) 에서 설명하는 예제를 참조하십시오 .

사용자 정의 등록 정보 생성

클라이언트 인증서 API 를 사용하여 고유의 등록 정보를 만들 수 있습니다 . 사용자 정의 매핑이 있는 경우 다음과 같이 해당 매핑을 참조합니다 .

name`:library` *path_to_shared_library*
name`:InitFN` *name_of_init_function*

예 :

```
certmap default1 o=Sun Microsystems, c=US
default1:library /usr/sun/userdb/plugin.so
default1:InitFn plugin_init_fn
default1:DNComps  ou o c
default1:FilterComps l
default1:verifycert on
```

매핑 예제

`certmap.conf` 파일에는 최소한 한 개 이상의 항목이 있어야 합니다 . 다음 예제에서는 `certmap.conf` 를 사용하는 다양한 방법을 설명합니다 ..

예제 1

이 예제의 certmap.conf 파일에는 오직 한 개의 기본 매핑만 있습니다 .

```
certmap default default
default:DNComps ou, o, c
default:FilterComps e, uid
default:verifycert on
```

이 예제를 사용하면 서버는 ou=*orgunit*, o=*org*, c=*country* 항목을 포함하는 LDAP 분기점에서 검색을 시작하며 , 여기에서 이탤릭체로 표시된 텍스트는 클라이언트 인증서에 있는 대상 DN 의 값으로 대체됩니다 .

이후 서버는 인증서에 있는 전자 메일 주소와 사용자 ID 값을 사용하여 LDAP 디렉토리에 일치하는 항목이 있는지 검색합니다 . 항목이 검색되면 서버는 클라이언트가 전송한 인증서와 디렉토리에 있는 인증서를 비교하여 인증서를 검증합니다 .

예제 2

다음 예제 파일에는 두 가지 매핑이 있습니다 . 기본 매핑과 미국 우편 서비스에 대한 매핑입니다 .

```
certmap default default
default:DNComps
default:FilterComps e, uid

certmap usps ou=United States Postal Service, o=usps, c=US
usps:DNComps ou,o,c
usps:FilterComps e
usps:verifycert on
```

서버가 미국 우편 서비스가 아닌 다른 곳에서 인증서를 받은 경우 , LDAP 트리의 최상단에서 검색을 시작하고 클라이언트의 전자 메일과 사용자 ID 와 일치하는 항목을 검색하는 기본 매핑을 사용합니다 . 미국 우편 서비스의 인증서일 경우 , 서버는 조직 단위를 포함하는 LDAP 분기점에서 검색을 시작하고 일치하는 전자 메일 주소를 검색합니다 . 또한 미국 우편 서비스의 인증서인 경우 서버는 인증서를 검증합니다 . 다른 인증서는 검증하지 않습니다 .

주의 인증서의 발행자 DN(즉 , CA 정보) 은 반드시 매핑의 첫 번째 줄 목록에 있는 발행자 DN 과 동일해야 합니다 . 앞의 예제에서 o=United States Postal Service,c=US 인 발행자 DN 의 인증서는 o 와 c 속성 사이에 공백이 없으므로 일치되지 않습니다 .

예제 3

다음 예제에서는 CmapLdapAttr 등록 정보를 사용하여 LDAP 데이터베이스에서 certSubjectDN 이라고 하는 속성을 검색합니다 . 이 속성의 값은 클라이언트 인증서에서 가져온 대상 DN 전체와 정확히 일치합니다 .

```
certmap myco ou=My Company Inc, o=myco, c=US
myco:CmapLdapAttr certSubjectDN
myco:DNComps  o, c
myco:FilterComps mail, uid
myco:verifycert on
```

클라이언트 인증서 대상이 다음인 경우 ,

```
uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

서버는 우선 다음 정보를 포함한 항목을 검색합니다 .

```
certSubjectDN=uid=Walt Whitman, o=LeavesOfGrass Inc, c=US
```

일치하는 항목이 하나 이상인 경우에는 서버가 항목을 검증합니다 . 검색된 항목이 없는 경우 서버는 DNComps 와 FilterComps 를 사용하여 일치하는 항목을 검색합니다 . 이 예제에서 서버는 o=LeavesOfGrass Inc, c=US 아래의 모든 항목에서 uid=Walt Whitman 을 검색합니다 .

참고 이 예제에서는 LDAP 디렉토리에 certSubjectDN 속성이 있는 항목이 포함된 것으로 가정합니다 .

고급 보안 설정

Server Manager Preferences 탭의 Set Cipher Size 옵션에서는 액세스용 보안 키 크기를 168, 128 또는 56 비트 중에서 선택하거나 제한을 설정하지 않을 수 있습니다 . 제한에 맞지 않는 경우 서비스될 파일을 지정할 수 있습니다 . 파일을 지정하지 않으면 Proxy Server 는 Forbidden 상태를 반환합니다 .

선택한 액세스용 키 크기가 Security Preferences 의 현재 암호 설정과 맞지 않는 경우 Proxy Server 에 더 큰 보안 키로 암호를 사용해야 한다는 경고 대화 상자가 표시됩니다 .

키 크기 제한은 Service fn=key-toosmall 이 아닌 NSAPI PathCheck 지시문을 기준으로 구현됩니다 . 지시문은 다음과 같습니다 .

```
PathCheck fn="ssl-check" [secret-keysize=nbits] [bong-file=filename]
```

여기에서 *nbits* 는 보안 키에 필요한 최소 비트 수이며 *filename* 은 제한에 맞지 않을 경우 서비스할 파일의 이름 (URL 아님) 입니다 .

SSL 을 사용하지 않거나 `secret-keysize` 매개 변수를 지정하지 않은 경우 `PathCheck` 는 `REQ_NOACTION` 을 반환합니다 . 현재 세션의 보안 키 크기가 지정된 `secret-keysize` 보다 작은 경우 , `bong-file` 이 지정되지 않으면 이 함수는 `REQ_ABORTED` 를 `PROTOCOL_FORBIDDEN` 상태와 함께 반환합니다 . 그렇지 않은 경우 `REQ_PROCEED` 를 반환하며 path 변수가 `bong-file` *filename* 으로 설정됩니다 . 또한 키 크기 제한을 만족하지 않은 경우 현재 세션용 SSL 세션 캐시가 무효화되어 다음 번 클라이언트가 서버로 연결하면 전체 SSL 핸드셰이크가 발생합니다 .

참고 Set Cipher Size 양식은 `PathCheck fn=ssl-check` 를 추가할 때 개체의 모든 `Service fn=key-toosmall` 지시문을 제거합니다 .

더 강력한 암호를 설정하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Set Cipher Size 링크를 누릅니다 .
3. 드롭다운 목록에서 더 강한 암호를 적용할 리소스를 선택한 다음 Select 를 누릅니다 . 정규 쉘 표현식을 지정할 수도 있습니다 . 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 를 참조하십시오 .
4. 보안 키 크기 제한을 선택합니다 .
   - 168 bit or larger
   - 128 bit or larger
   - 56 bit or larger
   - No restrictions
5. 액세스를 거부할 메시지의 위치를 지정하고 OK 를 누릅니다 .

자세한 내용은 SSL 개요를 참조하십시오 .

기타 보안 관련 고려 사항

누군가 암호를 해독하려는 시도 외에도 다른 보안 위험이 있습니다 . 네트워크는 서버와 서버의 정보에 액세스하려는 다양한 방법을 사용하는 외부 및 내부 해커의 위험에 직면해 있습니다 . 서버 암호화 사용 외에도 추가적인 보안 조치가 필요합니다 . 예를 들어 서버 컴퓨터를 안전한 곳에 배치하거나 신뢰할 수 없는 개인이 서버에 프로그램을 업로드하지 못하도록 해야 합니다 . 다음 절에서는 서버를 한층 안전하게 보호하기 위한 주요 사항을 설명합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

실제 액세스 제한

이 간단한 보안 수단을 간과하는 경우가 많습니다 . 서버 컴퓨터를 폐쇄된 장소에 배치하여 권한이 있는 사람만 출입할 수 있도록 합니다 . 이렇게 하면 서버 컴퓨터 자체에 대한 해킹을 방지할 수 있습니다 . 또한 관리 (루트) 비밀 번호가 있다면 이 비밀 번호를 보호하십시오 .

관리 액세스 제한

원격 구성을 사용하는 경우 오직 몇몇의 사용자와 컴퓨터에만 관리를 허용하도록 액세스 제어를 설정해야 합니다 . Administrator Server 가 LDAP 사용자에게 LDAP 서버나 로컬 디렉토리 정보에 대한 액세스를 부여하도록 하는 경우 , SSL 을 사용하는 Administration Server 가 마스터 서버의 역할을 하고 사용자가 기타 Administration Server 에 액세스할 수 있도록 두 대의 Administration Server 를 유지 및 클러스터 관리 사용을 고려하십시오 . 클러스터에 대한 자세한 내용은 제 6 장 , 119 페이지의 " 서버 클러스터 관리 " 를 참조하십시오 .

또한 Administration Server 에서 이에 해당하는 암호화를 실행해야 합니다 . 관리용 SSL 연결을 사용하지 않는 경우에는 비보안 네트워크를 통하여 원격 서버 관리를 수행할 때 주의가 필요합니다 . 누구라도 관리 비밀 번호를 가로채어 서버를 재구성할 수 있습니다 .

강력한 비밀 번호 선택

서버에는 관리 비밀 번호 , 개인 키 비밀 번호 , 데이터베이스 비밀 번호 등 여러 가지 비밀 번호를 사용합니다 . 관리 비밀 번호는 누구라도 컴퓨터에 있는 모든 서버를 구성할 수 있으므로 가장 중요한 비밀 번호입니다 . 다음으로는 개인 키 비밀 번호가 중요합니다 . 개인 키와 개인 키 비밀 번호가 있으면 사용자의 것으로 보이는 가짜 서버를 만들거나 서버로 오고 가는 통신을 가로채어 변경할 수 있습니다 .

자신은 기억할 수 있지만 다른 사람은 추측할 수 없는 비밀 번호가 좋은 비밀 번호입니다 . 예를 들어 *MCi12!mo* 를 "My Child is 12 months old!" 라고 기억할 수 있습니다 . 좋지 않은 암호의 예로는 자녀의 이름이나 생일 등이 있습니다 .

알아내기 힘든 비밀 번호

강력한 비밀 번호를 만드는 데 도움이 되는 간단한 지침이 있습니다 . 다음의 규칙을 모두 적용할 필요는 없지만 더 많은 규칙을 적용한 비밀 번호일수록 알아내기가 더욱 어려워질 것입니다 . 강력한 암호를 만들기 위한 몇 가지 팁은 다음과 같습니다 .

- 암호의 길이는 6-14 자여야 합니다 .
- *, " 나 공백과 같은 잘못된 문자를 사용해서는 안 됩니다 .
- 사전의 단어를 사용하면 안 됩니다 (모든 언어).
- E 를 3 으로 , L 을 1 로 하는 등의 일반적인 문자 대체를 사용해서는 안 됩니다 .
- 다음 종류의 문자를 가능한 한 많이 포함시킵니다 .

- 대문자
- 소문자
- 숫자
- 기호

비밀 번호 또는 PIN 변경

주기적으로 신뢰 데이터베이스 / 키 쌍 파일 비밀 번호 또는 PIN 을 변경하는 것이 좋습니다 . Administration Server 에서 SSL 을 사용하는 경우 서버를 시작할 때 이 비밀 번호가 필요합니다 . 주기적으로 비밀 번호를 변경하면 서버 보호를 한 단계 높일 수 있습니다 .

이 비밀 번호는 로컬 컴퓨터에서만 변경해야 합니다 . 비밀 번호를 변경하는 경우 고려해야 할 지침 목록은 " 알아내기 힘든 비밀 번호 " (114 페이지) 를 참조하십시오 .

신뢰 데이터베이스 / 키 쌍 파일 비밀 번호를 변경하려면 다음과 같이 합니다 .

1. Administration Server 또는 Server Manager 에 액세스하고 Security 탭을 누릅니다 .
2. Change Key Pair File Password 링크를 누릅니다 .
3. Cryptographic Module 드롭다운 목록에서 비밀 번호를 변경할 보안 토큰을 선택합니다 . 기본적으로 이것은 내부 키 데이터베이스용 내부 비밀 번호입니다 . PKCS #11 모듈이 설치된 경우 모든 보안 토큰 목록이 나타납니다 .
4. 현재 비밀 번호를 입력합니다 .
5. 새 비밀 번호를 입력합니다 .
6. 다시 입력하고 OK 를 누릅니다 .

키 쌍 파일이 보호되는지 확인합니다 . Administration Server 는 키 쌍 파일을 *server_root*/`alias` 디렉토리에 저장합니다 .

또한 파일이 백업 테이프에 저장되어 있는지 또는 기타 다른 사람이 가로챌 가능성이 있는지 확인하는 것이 중요합니다 . 이 경우 백업을 서버와 마찬가지로 완벽하게 보호해야 합니다 .

서버에서 기타 응용 프로그램 제한

서버와 동일한 컴퓨터에서 실행되는 모든 응용 프로그램을 신중하게 고려해야 합니다. 서버에서 실행되는 다른 프로그램의 취약점을 악용하여 서버의 보안을 우회하는 것이 가능합니다 . 필요하지 않은 모든 프로그램과 서비스를 종료합니다 . 예를 들어 UNIX `sendmail` 데몬은 안전하게 구성하는 것이 어려우며 서버 컴퓨터에서 해로운 프로그램을 실행하도록 프로그래밍될 수 있습니다 .

UNIX 및 Linux

`inittab` 및 `rc` 스크립트에서 시작하는 프로세스를 신중하게 선택합니다 . 서버 컴퓨터에서 `telnet` 또는 `rlogin` 을 실행하지 마십시오 . 또한 서버 컴퓨터에 `rdist` 를 두지 마십시오 . 파일을 배포할 수는 있지만 서버 컴퓨터의 파일을 업데이트하는 데도 사용될 수 있습니다 .

Windows

다른 컴퓨터와 공유할 드라이브 및 디렉토리를 신중히 고려합니다 . 또한 계정이나 Guest 권한을 부여할 사용자를 고려합니다 . 서버에 배치할 프로그램이나 다른 사용자가 설치할 수 있도록 허용할 프로그램을 결정하는 데도 주의가 필요합니다 . 다른 사람의 프로그램에는 보안 취약점이 있을 수 있습니다 . 또한 특히 보안을 손상시키도록 설계된 악의적 프로그램을 업로드할 수도 있습니다 . 서버에서 프로그램을 허용하기 전에 신중하게 프로그램을 평가해야 합니다 .

클라이언트가 SSL 파일을 캐시하지 못하도록 방지

HTML 파일의 `<HEAD>` 부분에 다음 줄을 추가하여 클라이언트가 미리 암호화된 파일을 캐시할 수 없도록 방지할 수 있습니다 .

```
<meta http-equiv="pragma" content="no-cache">
```

포트 제한

컴퓨터에서 사용되지 않는 포트는 모두 비활성화합니다 . 라우터 또는 방화벽 구성을 사용하여 정말 필요한 최소 포트 이외의 포트로 들어오는 연결을 차단합니다. 이렇게 하면 실제로 이미 제한된 영역에 위치한 서버의 컴퓨터를 사용할 때에만 컴퓨터에서 쉘을 사용할 수 있게 됩니다 .

서버의 한계 파악

서버는 서버와 클라이언트 사이에 안전한 연결을 제공합니다 . 클라이언트가 일단 정보를 보유하면 정보의 보안을 제어할 수 없으며 서버 컴퓨터 자체와 해당 디렉토리 및 파일에 대한 액세스는 제어할 수 없습니다 .

이러한 제한을 알고 있으면 피해야 할 상황을 이해하는 데 도움이 됩니다 . 예를 들어 SSL 연결을 통하여 신용 카드 번호를 구할 수 있으나 이 번호가 서버 컴퓨터의 보안 파일에 저장되어 있는지 , SSL 연결이 종료된 후 이 번호는 어떻게 될지에 대해 고려할 수 있습니다 . 클라이언트가 SSL 을 통해 사용자에게 전송하는 모든 정보의 보안에 유의하도록 합니다 .

6 장

서버 클러스터 관리

이 장에서는 Sun Java System Web Proxy Server 클러스터링의 개념을 설명하고 클러스터를 사용하여 서버 사이에서 구성을 공유하는 방법에 대해 설명합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

서버 클러스터 정보

클러스터는 단일 Administration Server 로 관리할 수 있는 Sun Java System Web Proxy Server 그룹입니다 . 각 클러스터에는 반드시 마스터 Administration Server 로 지정된 서버가 하나 있어야 합니다 .

서버를 클러스터로 조직하면 다음을 수행할 수 있습니다 .

- 중앙에서 모든 Proxy Server 관리
- 서버 사이에서 하나 이상의 구성 파일 공유
- 하나의 마스터 Administration Server 에서 모든 서버를 시작 및 종료
- 특정 서버의 액세스 및 오류 로그 확인

클러스터 사용에 대한 지침

다음 목록은 Proxy Server 그룹을 클러스터로 구성하는 데 대한 몇 가지 지침입니다 .

- 특정 클러스터에 포함할 모든 서버는 클러스터를 만들기 전에 설치해야 합니다 .
- 클러스터에 포함된 모든 서버는 같은 유형 (UNIX 또는 Windows) 이어야 합니다 . 클러스터는 반드시 같은 종류여야 합니다 .
- 클러스터의 모든 서버는 Proxy Server 버전 4 여야 합니다 . Proxy Server 버전 4 서버만 클러스터에 추가할 수 있습니다 .
- 모든 Administration Server 는 같은 프로토콜 (HTTP 또는 HTTPS) 을 사용해야 합니다 . 클러스터에 있는 Administration Server 중 하나의 프로토콜을 변경한 경우 모든 Administration Server 의 프로토콜을 변경해야 합니다 . 자세한 내용은 " 서버 정보 수정 " (122 페이지) 을 참조하십시오 .
- 각 클러스터의 Administration Server 는 마스터 Administration Server 와 동일한 아이디와 비밀 번호를 가져야 합니다 . 분산 관리를 사용하여 각 Administration Server 에 있는 여러 관리자를 구성할 수 있습니다 .
- 하나의 클러스터에 대한 Administration Server 를 마스터 Administration Server 로 지정해야 합니다 . 서버 종류는 관계 없습니다 .
- 마스터 Administration Server 는 각 클러스터에 대한 Administration Server 에 액세스할 수 있어야 합니다 . 마스터 Administration Server 는 설치된 모든 Sun Java System Web Proxy Server 에 대한 정보를 검색합니다 .

클러스터 설정

다음은 Proxy Server 클러스터를 설정할 때 따르는 일반적인 단계입니다 .

Proxy Server 클러스터를 설정하려면 다음을 수행합니다 .

1. 클러스터에 포함할 Proxy Server 를 설치합니다 . 클러스터에 대한 Administration Server 의 아이디 및 비밀 번호는 마스터 Administration Server 가 인증용으로 사용하는 것과 동일해야 합니다 . 기본 아이디 및 비밀 번호를 사용하거나 분산 관리를 구성하면 동일한 아이디 및 비밀 번호를 부여할 수 있습니다 .
2. 마스터 Administration Server 가 포함될 Proxy Server 를 설치하고 아이디와 비밀 번호가 제 1 단계에서 설정한 것과 동일한지 확인합니다 .
3. 클러스터 목록에 서버를 추가합니다 . 자세한 내용은 " 클러스터에 서버 추가 " (121 페이지) 를 참조하십시오 .

4. Control Cluster 페이지에서 Server Manager 인터페이스에 액세스하거나 클러스터에 있는 한 서버의 구성 파일을 다른 서버로 복사하여 원격 서버를 관리합니다 .

클러스터에 서버 추가

Proxy Server 를 클러스터에 추가하면 Administration Server 와 포트 번호가 지정됩니다 . 이 Administration Server 에 하나 이상의 서버에 대한 정보가 있는 경우 해당 서버 모두가 클러스터에 추가됩니다 . 나중에 개별 서버를 제거할 수 있습니다 .

참고 원격 Administration Server 에 클러스터에 대한 정보가 있는 경우 원격 클러스터에 있는 서버는 추가되지 않습니다 . 마스터 Administration Server 는 원격 컴퓨터에 실제로 설치된 서버만 추가합니다 .

클러스터에 원격 서버를 추가하려면 다음을 수행합니다 .

1. 마스터 Administration Server 가 작동 중인지 확인합니다 .
2. 마스터 Administration Server 에 액세스하고 Cluster 탭을 누릅니다 .
3. Add Server 링크를 누릅니다 .
4. 원격 Administration Server 에서 사용할 프로토콜을 다음과 같이 선택합니다 .
   - 일반적인 Administration Server 인 경우 HTTP
   - 보안 Administration Server 인 경우 HTTPS
5. 원격 Administration Server 의 정규화된 호스트 이름을 `magnus.conf` 파일에 표시된 대로 입력합니다 (예 : `plaza.example.com`).
6. 원격 Administration Server 의 포트 번호를 입력합니다 .
7. 원격 Administration Server 의 관리자 아이디와 비밀 번호를 입력하고 OK 를 누릅니다 . 마스터 Administration Server 가 원격 서버와의 연결을 시도합니다 . 연결이 성공하면 서버를 클러스터에 추가할 것인지 묻는 메시지가 표시됩니다 .

참고 클러스터 제어를 사용하도록 설정한 경우 클러스터의 마스터는 클러스터 내 각 슬레이브에 대한 `proxy-`*serverid*`/config/cluster/`*server_name*`/proxy-`*serverid* 디렉토리에 여러 개의 파일을 만듭니다 . 이 파일의 구성은 변경할 수 없습니다 .

서버 정보 수정

Administration Server 의 Cluster 탭에 있는 Modify Server 옵션은 슬레이브 서버에서 슬레이브 관리 포트 번호가 변경된 경우 이를 업데이트할 때만 사용합니다 . 클러스터에 있는 원격 Administration Server 의 포트 번호를 변경하는 경우 , 클러스터에 저장된 Administration Server 의 정보도 함께 수정해야 합니다 . 그 외에 다른 사항을 슬레이브 Administration Server 에서 변경하려면 클러스터에서 서버를 제거한 후 , 변경 사항이 적용된 해당 서버를 클러스터에 다시 추가해야 합니다 .

클러스터에 있는 서버에 관한 정보를 수정하려면 다음을 수행합니다 .

1. 마스터 Administration Server 에 액세스하고 Cluster 탭을 누릅니다 .
2. Modify Server 링크를 누릅니다 . 서버 목록이 고유 서버 식별자별로 표시됩니다 .
3. 수정할 서버를 선택하고 원하는 부분을 변경한 다음 OK 를 누릅니다 .

클러스터에서 서버 제거

클러스터에서 서버를 제거하려면 다음을 수행합니다 .

1. 마스터 Administration Server 에 액세스하고 Cluster 탭을 누릅니다 .
2. Remove Server 링크를 누릅니다 .
3. 클러스터에서 제거할 원격 서버를 선택하고 OK 를 누릅니다 . 제거한 서버는 더 이상 클러스터를 통해 액세스할 수 없으며 자체 Administration Server 를 통해서만 액세스할 수 있습니다 .

서버 클러스터 제어

Proxy Server 에서는 다음과 같이 클러스터에 있는 원격 서버를 제어할 수 있습니다 .

- 서버 시작 및 정지
- 액세스 및 오류 로그 확인
- 구성 파일 전송 (마스터 Administration Server 에 Proxy Server 인스턴스가 두 개 이상 있는 경우 이 중 한 서버에서 클러스터에 추가된 슬레이브로 파일을 전송할 수 있습니다 .) 클러스터는 반드시 같은 종류여야 합니다 . 클러스터에 포함된 모든 서버는 같은 유형 (UNIX 또는 Windows) 이어야 합니다 . 서로 다른 플랫폼으로 구성 파일을 전송하면 서버가 중단되거나 손상됩니다 . 구성 파일은 다음과 같습니다 .
  - `server.xml`
  - `magnus.conf`
  - `obj.conf`
  - `mime.types`
  - `socks5.conf`
  - `bu.conf`
  - `icp.conf`
  - `parray.pat`
  - `parent.pat`

클러스터에 있는 서버를 제어하려면 다음을 수행합니다 .

1. 마스터 Administration Server 에 액세스하고 Cluster 탭을 누릅니다 .
2. Control Cluster 링크를 누릅니다 .
3. 제어할 서버를 선택하고 원하는 대로 설정합니다 . 언제든 Reset 버튼을 누르면 변경하기 전에 포함된 값으로 요소를 재설정합니다 .
   - 드롭다운 목록에서 Start, Stop 또는 Restart 를 선택하고 Go 를 누릅니다 . 동작을 수행할 것인지 묻는 메시지가 표시됩니다 .
   - 드롭다운 목록에서 View Access 또는 View Error 를 선택하고 로그 파일에서 보려는 마지막 줄 수를 입력합니다 . Go 를 누르면 정보가 표시됩니다 (표시된 Cluster Execution Report 에서 View 버튼을 누름).
   - 구성 파일을 전송하려면 다음을 수행합니다 .

- 전송하려는 구성 파일을 선택합니다 .
- 파일이 있는 서버를 선택합니다 .
- Go 를 눌러 정보를 전송합니다 .

제 3 부

Proxy Server 구성 및 모니터링

7 장

서버 기본 설정 구성

이 장에서는 Proxy Server 의 시스템 설정 및 구성 방법에 대해 설명합니다 . 시스템 설정은 Proxy Server 전체에 적용됩니다 . 이 설정은 프록시 서버가 사용하는 사용자 계정 및 청취하는 포트와 같은 옵션을 포함합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

Proxy Server 시작

이 절에서는 다양한 플랫폼에서 Proxy Server 를 시작하는 방법에 대해 설명합니다 . 서버가 설치되면 실행되어 요청을 청취하고 허용합니다 .

관리 인터페이스에서 Proxy Server 를 시작하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Start/Stop Server 링크를 누릅니다 . Start/Stop Server 페이지가 표시됩니다 .
3. On 버튼을 누릅니다 .

서버의 상태는 Start/Stop Server 페이지에 표시됩니다 .

UNIX 나 Linux 에서 Proxy Server 를 시작하려면 다음과 같이 합니다 .

- 명령줄에서 *server_root*/proxy-server*id* 로 이동하고 `./start` 를 입력하여 Proxy Server 를 시작합니다 .
- `start` 를 사용합니다 . 이 스크립트를 `init` 와 함께 사용하려면 반드시 start 명령 `prxy:2:respawn:`*server_root*`/proxy-server`*id*`/start -start -i` 를 `/etc/inittab` 에 포함시켜야 합니다 .

Windows 에서 Proxy Server 를 시작하려면 다음과 같이 합니다 .

- 시작 > 프로그램 > Sun Microsystems > Sun Java System Web Proxy Server *버전* > Start Proxy Server 를 사용합니다 .
- 제어판 > 관리 도구 > 서비스 > Sun Java System Web Proxy Server 4.0 (proxy-server*id*) > Start 를 사용합니다 .
- 명령 프롬프트에서 *server_root*`\proxy-server`*id* 로 이동한 후 `startsvr.bat` 를 입력하여 Proxy Server 를 시작합니다 .

SSL 을 사용하는 서버 시작

SSL 을 사용하는 서버를 시작하려면 비밀 번호가 필요합니다 . 비밀 번호를 일반 텍스트 파일에 저장하여 SSL 을 사용하는 서버를 자동으로 시작할 수는 있으나 , 이는 권장하지 않습니다 .

주의 SSL 을 사용하는 서버의 비밀 번호를 서버 시작 스크립트의 일반 텍스트에 보관하면 보안상 위험할 수 있습니다 . 파일에 액세스할 수 있는 사용자는 모두 SSL 을 사용하는 서버의 비밀 번호를 알 수 있습니다 . SSL 을 사용하는 서버의 비밀 번호를 일반 텍스트에 보관하기 전에 보안상 위험 요소에 대해 고려해야 합니다 .

서버의 시작 스크립트 , 키 쌍 파일 및 키 비밀 번호는 루트가 소유해야 하며 (또는 루트가 아닌 서버가 서버를 설치한 경우 해당 사용자 계정) 오직 소유자만이 이에 대한 읽기 및 쓰기 액세스 권한이 부여되어야 합니다 .

SSL 을 사용하는 서버를 UNIX 나 Linux 에서 자동 시작하려면 다음과 같이 합니다 .

1. 텍스트 편집기에서 `start` 파일을 엽니다 .
2. 스크립트에서 `-start` 줄을 찾은 후 다음을 삽입합니다 .

   `echo "`*password*`"|`

   여기에서 *password* 는 선택한 SSL 비밀 번호입니다 .

   예를 들어 SSL 비밀 번호가 `examples` 이면 편집된 행은 다음과 같이 보입니다 .

   ```
   -start)
     echo "examples"|./$PRODUCT_BIN -d $PRODUCT_SUBDIR/config $@
   ```

Proxy Server 중지

이 절에서는 다양한 플랫폼에서 Proxy Server 를 중지하는 여러 방법에 대해 설명합니다 .

관리 인터페이스에서 Proxy Server 를 중지하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Start/Stop Server 링크를 누릅니다 . Start/Stop Server 페이지가 표시됩니다 .
3. Off 버튼을 누릅니다 .

서버의 상태가 Server On/Off 페이지에 표시됩니다 .

UNIX 나 Linux 에서 Proxy Server 를 중지하려면 다음과 같이 합니다 .

- 명령줄에서 *server_root*`/proxy-server`*id* 로 이동하고 `./stop` 을 입력합니다 .

참고 `etc/inittab` 파일을 사용하여 서버를 재시작하는 경우 반드시 `/etc/inittab` 에서 서버를 시작하는 줄을 제거하고 서버를 중지하기 전에 `kill -1 1` 을 입력해야 합니다 . 그렇지 않으면 서버가 중지된 후 자동으로 재시작하게 됩니다 .

- stop 을 사용하면 서버가 완전히 종료되며 서버가 다시 시작할 때까지 서비스가 중단됩니다. etc/inittab 파일이 자동으로 재시작하도록 설정하려면("respawn" 사용) 반드시 서버를 종료하기 전에 etc/inittab 에 있는 프록시 서버 관련 줄을 제거해야 합니다 . 그렇지 않은 경우 서버가 자동으로 재시작합니다 .

서버를 종료하면 서버가 종료 과정을 완료하고 상태를 off 로 변경하는 데 약간의 시간이 걸릴 수 있습니다 .

시스템이 중단되거나 오프라인이 되는 경우에는 서버가 중지하며 서비스하는 모든 요청을 잃게 됩니다 .

참고 서버에 보안 모듈이 설치된 경우 서버를 시작 또는 중지하기 전에 적절한 비밀 번호를 입력해야 합니다 .

Windows 에서 Proxy Server 를 중지하려면 다음과 같이 합니다 .

- 시작 > 프로그램 > Sun Microsystems > Sun Java System Web Proxy Server *버전* > Stop Proxy Server 를 사용합니다 .
- 명령 프롬프트에서 *server_root*\proxy-server*id* 로 이동한 후 stopsvr.bat 를 입력하여 Proxy Server 를 시작합니다 .
- 서비스 창에서 Sun Java System Proxy Server 4.0 (proxy-server*id*) 서비스를 사용합니다 (제어판 > 관리 도구 > 서비스).

Proxy Server 재시작

이 절에서는 다양한 플랫폼에서 Proxy Server 를 재시작하는 여러 방법에 대해 설명합니다 .

서버 재시작 (UNIX 또는 Linux)

다음 중 한 가지 방법으로 서버를 재시작할 수 있습니다 .

- 직접 재시작합니다 .
- inittab 파일에서 자동으로 재시작합니다 .

  SunOS 4.1.3 처럼 System V 에서 파생되지 않은 UNIX 또는 Linux 버전을 사용할 경우 inittab 파일을 사용할 수 없습니다 .
- 시스템을 재부팅할 때 /etc/rc2.d 에 있는 데몬으로 자동 재시작합니다 .

설치 스크립트는 /etc/rc.local 또는 /etc/inittab 파일을 편집할 수 없으므로 반드시 텍스트 편집기에서 이 파일을 편집해야 합니다. 이 파일의 편집 방법을 모르는 경우에는 시스템 관리자에게 문의하거나 시스템 설명서를 참조하십시오.

명령줄에서 Proxy Server 를 재시작하려면 다음과 같이 합니다.

1. 번호가 1024 미만인 포트에서 서버를 실행할 경우 루트로 로그인하고, 그 밖의 경우 루트로 로그인하거나 서버 사용자 계정으로 로그인합니다.
2. 명령줄 프롬프트에서 다음 줄을 입력하고 Enter 를 누릅니다.

   server_root/proxy-server*id*/restart

   여기에서 *server_root* 는 서버를 설치한 디렉토리입니다.

- 줄의 마지막에 선택 매개 변수인 -i 를 사용할 수 있습니다. -i 옵션을 사용하면 서버가 inittab 모드로 실행되므로 서버 프로세스가 중단 또는 중지되는 경우 inittab 가 서버를 자동으로 재시작합니다. 또한 이 옵션을 사용하면 서버가 자체를 배경 프로세스로 전환할 수 없도록 방지합니다.

inittab 를 사용하여 서버를 재시작하려면 다음과 같이 합니다.

/etc/inittab 파일의 한 줄에 다음 텍스트를 추가합니다.

prxy:23:respawn:*server_root*/proxy-server*id*/start -start -i

여기에서 *server_root* 는 서버를 설치한 디렉토리이며 proxy-server*id* 는 서버의 디렉토리입니다.

-i 옵션을 사용하면 서버가 자체를 백그라운드 프로세스로 전환할 수 없도록 방지합니다.

서버를 중지하기 전에 반드시 이 줄을 제거해야 합니다.

System RC Script 를 사용하여 서버를 재시작하려면 다음과 같이 합니다.

/etc/rc.local 또는 이에 상응하는 시스템 파일을 사용할 경우 /etc/rc.local 에 다음 행을 추가합니다.

server_root/proxy-server*id*/start

server_root 를 서버를 설치한 디렉토리로 대체합니다.

서버 재시작 (Windows)

다음과 같이 서버를 재시작할 수 있습니다.

- 서비스 제어판을 사용합니다.

Windows 에서 서버를 재시작하려면 다음과 같이 합니다 .

1. 제어판 > 관리 도구 > 서비스를 사용합니다 .
2. 서비스 목록에서 Sun Java System Web Proxy Server 4.0(proxy-server*id*) 를 선택합니다 .
3. 속성 창에서 시작 유형을 자동으로 변경하여 시스템이 컴퓨터를 시작하거나 재부팅할 때마다 서버를 시작하도록 합니다 .
4. 확인을 누릅니다 .

종료 시간 제한 설정

서버가 중지되면 새 연결을 받지 않습니다 . 또한 기존 연결이 완료되도록 대기합니다 . 서버가 제한 시간이 만료되기 전까지 대기하는 시간은 `magnus.conf` 파일에 구성할 수 있습니다 . 기본값은 30 초입니다 . 이 값을 변경하려면 `magnus.conf` 파일에 다음과 같은 행을 추가합니다 .

`TerminateTimeout` *seconds*

여기서 *seconds* 는 서버가 제한 시간 동안 대기하는 초 단위 시간입니다 .

이 값을 구성하면 서버가 연결이 완료될 때까지 더 긴 시간 동안 대기하는 장점이 있습니다 . 그러나 때로 서버에는 응답하지 않는 클라이언트의 연결이 있으므로 종료 시간 제한을 크게 하면 서버가 종료되는 시간이 더 오래 걸릴 수 있습니다 .

서버 설정 보기

설치 중에 Proxy Server 에 대한 일부 설정을 구성할 수 있습니다 . Server Manager 에서 이러한 설정과 다른 시스템 설정을 확인할 수 있습니다 . View Server Settings 페이지에는 Proxy Server 의 모든 설정이 목록으로 나타납니다 . 또한 Proxy Server 가 새 구성으로 시작되도록 변경 사항을 저장하고 재시작해야 하는 경우에 저장되지 않거나 적용되지 않은 변경 사항이 있으면 사용자에게 알립니다 .

설정 유형에는 기술적 설정과 내용 설정 , 두 가지가 있습니다 . 서버의 내용 설정은 사용자가 서버를 어떻게 구성했는지에 따라 결정됩니다 . 일반적으로 프록시는 모든 템플릿 , URL 매핑 , 액세스 제어를 표시합니다 . 이 페이지는 캐시 설정과 같은 개별 템플릿에 대한 템플릿 이름 , 정규식 및 해당 템플릿의 설정을 표시합니다 .

프록시 서버의 기술 설정은 `magnus.conf` 파일 및 `server.xml` 파일에 있으며 컨텐트 설정은 `obj.conf` 파일에 있습니다 . 이들 파일은 서버 루트 디렉토리의 `proxy-`*id*`/config` 라는 하위 디렉토리에 있습니다 .

Proxy Server 에 대한 설정을 보려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. View Server Settings 링크를 누릅니다 . View Server Settings 페이지가 표시됩니다 .

구성 파일 백업 보기 및 복구

구성 파일의 백업 사본 (`server.xml, magnus.conf, obj.conf, mime types, server.xml.clfilter, magnus.conf.clfilter, obj.conf.clfilter, socks5.conf, bu.conf, icp.conf, parray.pat, parent.pat, proxy-id.acl`)을 보거나 복구할 수 있습니다 . 이 기능을 사용하면 현재 구성에 문제가 있을 경우 이전 구성으로 복구할 수 있습니다 . 예를 들어 프록시 구성을 몇 가지 변경한 후 프록시가 제대로 작동하지 않을 경우 (예 : URL 액세스는 거부당하나 프록시는 요청을 서비스할 경우) 이전 구성으로 복구하여 구성 변경을 다시 수행할 수 있습니다 .

이전 구성을 보려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Restore Configuration 링크를 누릅니다 . Restore Configuration 페이지가 표시됩니다 . 이 페이지는 날짜 및 시간 순서에 따라 모든 이전 구성을 나열합니다 .
3. 특정 버전의 기술적 설정 및 내용 설정 목록을 표시하려면 View 링크를 누릅니다 .

구성 파일의 백업 사본을 복구하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Restore Configuration 링크를 누릅니다 . Restore Configuration 페이지가 표시됩니다 . 이 페이지는 날짜 및 시간 순서에 따라 모든 이전 구성을 나열합니다 .
3. 복구할 버전에 대한 Restore 링크를 누릅니다 .

   특정 시간에 해당하는 모든 파일을 복구하려면 표의 가장 왼쪽에 있는 열에 대한 *time* 링크 (복구할 날짜 및 시간이 될 *시/간*) 를 누릅니다 .

Restore Configuration 페이지에 표시되는 백업 수도 설정할 수 있습니다 .

표시되는 백업 수를 설정하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Restore Configuration 링크를 누릅니다 . Restore Configuration 페이지가 표시됩니다 .
3. Set Number Of Sets Of Backups 필드에 표시할 백업 수를 입력합니다 .
4. Change 버튼을 누릅니다 .

시스템 기본 설정 구성

Configure System Preferences 페이지에서는 서버의 기본적인 측면을 설정하거나 변경할 수 있습니다 . 이 페이지에서 서버 사용자 , 프로세스 수 , 청취 큐 크기 , 프록시 시간 제한 , 프록시 서버 중단 후 시간 제한을 변경할 수 있습니다 . 또한 DNS, ICP, 프록시 배열 , 상위 배열을 사용하도록 설정할 수 있습니다 .

시스템 기본 설정을 수정하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Configure System Preferences 링크를 누릅니다 . Configure System Preferences 페이지가 표시됩니다 .
3. 필요에 따라 옵션을 변경하고 OK 를 누릅니다 .
4. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
5. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

이 옵션은 다음 절에서 설명합니다 .

Server User

Server User 는 프록시가 사용하는 사용자 계정입니다 . 서버 사용자로 입력한 아이디는 이미 정상적인 프록시 서버 사용자 계정으로 존재해야 합니다 . 서버를 시작할 때 이 사용자가 시작한 것처럼 실행됩니다 .

새 사용자 계정을 만들지 않으려면 동일한 호스트에서 실행되는 다른 서버가 사용하는 계정을 선택하거나 , UNIX 프록시를 실행하는 경우 nobody 사용자를 선택할 수 있습니다 . 그러나 시스템에 따라 nobody 사용자가 파일을 소유할 수 있지만 프로그램은 실행할 수 없는 경우가 있습니다 . 이 경우 nobody 는 프록시 사용자 이름으로 적절하지 않습니다 .

UNIX 시스템에서 프록시가 생성하는 모든 프로세스는 서버 사용자 계정에 할당됩니다 .

Processes

Processes 필드는 요청을 서비스하는 데 사용할 수 있는 프로세스 수입니다 . 기본값은 1 이며 , 필요하지 않은 이상 이 설정을 변경하지 마십시오 .

Listen Queue Size

Listen Queue Size 필드는 청취 소켓의 보류된 연결 수의 최대값을 지정합니다 .

DNS

DNS(Domain Name Service) 는 IP 주소를 호스트 이름으로 복원합니다 . 웹 브라우저가 서버에 연결되면 서버는 클라이언트의 IP 주소 (예 : `198.18.251.30`) 만 가져옵니다 . 서버는 `www.example.com.` 과 같은 호스트 이름 정보는 갖고 있지 않습니다 . 서버는 액세스 로그와 액세스 제어를 위해 IP 주소를 호스트 이름으로 변환할 수 있습니다 . Configure System Preferences 페이지에서 서버가 IP 주소를 호스트 이름으로 변환할지 여부를 지정할 수 있습니다 .

ICP

ICP(Internet Cache Protocol) 는 캐시가 서로 통신할 수 있도록 하는 메시지 전달 프로토콜입니다 . 캐시는 ICP 를 사용하여 캐시된 URL 의 존재 및 이러한 URL 을 가져올 최적의 위치에 대한 쿼리와 응답을 보낼 수 있습니다 . Configure System Preferences 페이지에서 ICP 를 사용하도록 설정할 수 있습니다 . ICP 에 대한 자세한 내용은 "ICP 이웃을 통한 라우팅 " (279 페이지) 을 참조하십시오 .

Proxy Array

프록시 배열은 분산 캐시를 위해 하나의 캐시 역할을 하는 여러 프록시의 배열입니다 . Configure System Preferences 페이지에서 프록시 배열 옵션을 사용하도록 설정하면 , 현재 구성하고 있는 프록시 서버는 프록시 배열의 구성원이 되며 , 해당 배열의 다른 모든 구성원과 동급 관계가 됩니다 . 프록시 배열 사용에 대한 자세한 내용은 " 프록시 배열을 통한 라우팅 " (288 페이지) 을 참조하십시오 .

Parent Array

상위 배열은 프록시 또는 프록시 배열이 라우팅하는 프록시 배열입니다 . 따라서 프록시가 원격 서버에 액세스하기 전에 상위의 프록시 배열을 통하여 라우팅할 경우 , 해당 상위 프록시 배열은 상위 배열로 간주됩니다 . 프록시 서버에서의 상위 배열 사용에 대한 자세한 내용은 " 상위 배열을 통한 라우팅 " (300 페이지) 을 참조하십시오 .

Proxy Timeout

프록시 시간 제한은 원격 서버에서 전송되는 연속적인 네트워크 데이터 패킷 간에 허용되는 최대 시간으로 이를 초과할 경우 프록시 서버는 해당 요청을 시간 초과로 처리하게 됩니다 . 프록시 시간 제한의 기본값은 5 분입니다 .

참고	원격 서버가 프록시 시간 제한보다 긴 페이지 사이에 서버 푸시와 지연을 사용하면 전송 시간이 완료되기 전에 연결이 종료될 수 있습니다 . 대신 여러 요청을 프록시로 전송하는 클라이언트 풀을 사용하도록 합니다 .

Proxy Server 조정

Tune Proxy 페이지에서 기본 매개 변수를 변경하여 프록시 서버의 성능을 조정할 수 있습니다 .

기본 조정 매개 변수를 변경하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Tune Proxy 링크를 누릅니다 . Tune Proxy 페이지가 표시됩니다 .

3. 요구 사항에 맞도록 FTP 목록의 너비를 수정할 수 있습니다 . 목록의 너비를 넓히면 더 긴 파일 이름을 사용할 수 있으므로 파일 이름이 잘리는 현상을 줄일 수 있습니다 . 기본 너비는 80 자입니다 .
4. OK 를 누릅니다 .
5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

청취 소켓 추가 및 편집

서버가 요청을 처리하려면 청취 소켓을 통하여 요청을 접수한 후 요청을 올바른 서버로 보내야 합니다 . Proxy Server 를 설치하는 경우 청취 소켓 한 개 (ls1) 가 자동으로 만들어집니다 . 이 청취 소켓은 IP 주소 `0.0.0.0` 과 설치 도중 프록시 서버 포트 번호로 지정한 포트 번호를 사용합니다 . 기본 청취 소켓은 삭제할 수 없습니다 .

청취 소켓은 Server Manager 의 Add Listen Socket 및 Edit Listen Sockets 페이지를 사용하여 추가 , 편집 및 삭제할 수 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 청취 소켓 추가
- 청취 소켓 편집
- 청취 소켓 삭제

청취 소켓 추가

청취 소켓을 추가하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Add Listen Socket 링크를 누릅니다 . Add Listen Socket 페이지가 표시됩니다 .
3. 청취 소켓의 내부 이름을 지정합니다. 청취 소켓을 만든 후에는 이 이름을 변경할 수 없습니다 .
4. 청취 소켓의 IP 주소를 지정합니다 . 이 주소는 점으로 연결된 쌍 (dotted-pair) 또는 IPv6 표기 방식일 수 있습니다 . 또한 `0.0.0.0`, any, `ANY` 또는 `INADDR_ANY`(모든 IP 주소) 일 수 있습니다 .

5. 청취 소켓이 만들어질 포트 번호를 지정합니다 . 유효한 값은 1 - 65535 입니다 . UNIX 에서 포트 1 - 1024 를 청취하는 소켓을 만들려면 수퍼유저 권한이 필요합니다 . SSL 청취 소켓이 포트 443 을 청취하도록 구성하는 것이 좋습니다 .
6. 서버가 클라이언트로 전송하는 모든 URL 의 호스트 이름 부분에 사용되는 서버 이름을 지정합니다 . 이에 따라 서버가 자동으로 생성하는 URL 이 달라지지만 서버에 저장된 디렉토리 및 파일에 대한 URL 에는 영향을 주지 않습니다 . 서버에서 별칭을 사용하는 경우 이 이름은 별칭이어야 합니다 .
7. 드롭다운 목록에서 청취 소켓에 보안이 사용되는지 여부를 지정합니다 .
8. OK 를 누릅니다 .
9. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
10. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

청취 소켓 편집

청취 소켓을 편집하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Edit Listen Sockets 링크를 누릅니다 . Edit Listen Sockets 페이지가 나타납니다 .
3. Configured Sockets 표에서 편집할 청취 소켓에 대한 링크를 누릅니다 . Edit Listen Sockets 페이지가 표시됩니다 .
4. 다음 옵션에 대해 필요한 변경을 수행합니다 .

- **General**
  - **Listen Socket ID.** 청취 소켓용 내부 이름입니다 . 청취 소켓을 만든 후에는 이 이름을 변경할 수 없습니다 .
  - **IP Address.** 청취 소켓의 IP 주소입니다 . 이 주소는 점으로 연결된 쌍 (dotted-pair) 또는 IPv6 표기 방식일 수 있습니다 . 또한 be `0.0.0.0`, any 또는 `ANY` 또는 `INADDR_ANY`(모든 IP 주소) 일 수 있습니다 .
  - **Port.** 청취 소켓이 만들어진 포트 번호입니다 . 유효한 값은 1-65535 입니다 . UNIX 에서 포트 1-1024 를 청취하는 소켓을 만들려면 수퍼유저 권한이 필요합니다 . SSL 청취 소켓이 포트 443 을 청취하도록 구성하는 것이 좋습니다 .
  - **Server Name.** 이 청취 소켓의 기본 서버입니다 .
- **Security**

  보안을 사용하지 않을 경우 다음 매개 변수만 표시됩니다 .

- **Security.** 선택한 청취 소켓용 보안을 사용하거나 사용하지 않도록 설정합니다 .

보안을 사용할 경우 다음 매개 변수가 표시됩니다 .

- **Security.** 선택한 청취 소켓용 보안을 사용하거나 사용하지 않도록 설정합니다 .
- **Server Certificate Name.** 드롭다운 목록에서 설치된 인증서를 선택하여 이 청취 소켓용으로 사용할 수 있습니다 .
- **Client Authentication.** 이 청취 소켓에 클라이언트 인증이 필요한지 여부를 지정합니다 . Optional 로 기본 설정되어 있습니다 .
- **SSL Version 2.** SSL Version 2 를 사용하거나 사용하지 않도록 합니다 . 사용하지 않도록 기본 설정되어 있습니다 .
- **SSL Version 2 Ciphers.** 이 제품군에 있는 모든 암호화 목록이 표시됩니다 . 해당 확인란을 선택하거나 선택 취소하여 편집 중인 청취 소켓에 대해 사용할 암호화를 선택할 수 있습니다 . 암호화를 선택하면 기본 버전은 선택 취소됩니다 .
- **SSL Version 3.** SSL Version 3 을 사용하거나 사용하지 않도록 합니다 . 사용하도록 기본 설정되어 있습니다 .
- **TLS.** 암호화된 통신을 위하여 TLS(Transport Layer Security)를 사용하거나 사용하지 않도록 합니다 . 사용하도록 기본 설정되어 있습니다 .
- **TLS Rollback.** TLS Rollback 을 사용하거나 사용하지 않도록 합니다 . TLS Rollback 을 사용하지 않으면 연결이 버전 롤백 공격에 취약한 상태로 남는다는 점을 유의하십시오 . 사용하도록 기본 설정되어 있습니다 .
- **SSL Version 3 and TLS Ciphers.** 이 제품군에 있는 모든 암호화 목록이 표시됩니다 . 해당 확인란을 선택하거나 선택 취소하여 편집 중인 청취 소켓에 대해 사용할 암호화를 선택할 수 있습니다 . 암호화를 선택하면 기본 버전이 선택됩니다 .

- **Advanced**
  - **Number Of Acceptor Threads.** 청취 소켓용 승인자 스레드의 수입니다 . 권장값은 컴퓨터에 있는 프로세서의 수입니다 . 기본값은 1 이며 유효한 값은 1 -1024 입니다 .

  Protocol Family. 소켓군 유형입니다 . 유효한 값은 `inet`, `inet6` 및 `nca` 입니다 . IPv6 청취 소켓의 경우 `inet6` 을 사용합니다 . Solaris™ Network Cache 와 Accelerator 를 사용할 수 있도록 하려면 `nca` 를 지정합니다 .

5. OK 를 누릅니다 .

6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

청취 소켓 삭제

청취 소켓을 삭제하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Edit Listen Sockets 링크를 누릅니다 .
3. 삭제할 청취 소켓 옆의 확인란을 선택하고 OK 를 누릅니다 . 삭제를 확인하는 메시지가 표시됩니다 . 해당 인스턴스에 대한 유일한 청취 소켓이 아니라면 어떤 청취 소켓이라도 삭제할 수 있습니다 .
4. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
5. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

MIME 유형

MIME(Multi-purpose Internet Mail Extension) 유형은 멀티미디어 전자 메일 및 메시징 표준입니다 . 따라서 해당 MIME 유형에 따라 파일을 필터링할 수 있으며 , 프록시 서버는 서버와 함께 사용할 새 MIME 유형을 만들 수 있는 페이지를 제공합니다 . 프록시는 `mime.types` 파일에 새 유형을 추가합니다 . MIME 유형에 따른 파일 차단에 대한 자세한 내용은 "MIME 유형별 필터링 " (309 페이지) 을 참조하십시오 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 새 MIME 유형 만들기
- MIME 유형 편집
- MIME 유형 제거

새 MIME 유형 만들기

MIME 유형을 만들려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .

2. Create/Edit MIME Types 링크를 누릅니다. Create/Edit MIME Types 페이지는 프록시의 `mime.types` 파일에 있는 모든 MIME 유형을 표시합니다 .
3. 드롭다운 목록에서 MIME 유형 범주를 지정합니다 . 이것은 `type`, enc 또는 `lang` 이 될 수 있습니다 . 여기서 `type` 은 파일 또는 응용 프로그램 유형 , `enc` 는 압축에 사용된 인코딩 , `lang` 은 언어 인코딩입니다 . 범주에 대한 자세한 내용은 온라인 도움말을 참조하십시오 .
4. HTTP 헤더에 표시되는 컨텐트 유형을 지정합니다 .
5. 파일 접미사를 지정합니다 . 파일 접미사는 MIME 유형에 매핑된 파일 확장자를 참조합니다 . 하나 이상의 확장자를 지정하려면 항목을 쉼표로 분리합니다 . 파일 확장자는 고유해야 합니다 . 즉 하나의 파일 확장자를 두 개의 MIME 유형에 매핑해서는 안 됩니다 .
6. New 버튼을 눌러 MIME 유형을 추가합니다 .

MIME 유형 편집

MIME 유형을 편집하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Create/Edit MIME Types 링크를 누릅니다. Create/Edit MIME Types 페이지는 프록시의 `mime.types` 파일에 있는 모든 MIME 유형을 표시합니다 .
3. 해당 MIME 유형에 대한 Edit 링크를 눌러 MIME 유형을 편집할 수 있습니다 .
4. 필요한 사항을 변경하고 Change MIME Type 버튼을 누릅니다 .

MIME 유형 제거

MIME 유형을 제거하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Create/Edit MIME Types 링크를 누릅니다. Create/Edit MIME Types 페이지는 프록시의 `mime.types` 파일에 있는 모든 MIME 유형을 표시합니다 .
3. 해당 MIME 유형에 대한 Remove 링크를 눌러 MIME 유형을 제거할 수 있습니다.

액세스 제어 관리

Administer Access Control List 페이지에서 액세스 제어 목록 (ACL) 을 관리할 수 있습니다 . ACL 을 사용하면 사용하는 서버에 액세스할 수 있는 클라이언트를 통제할 수 있습니다 . ACL 은 특정 사용자 , 그룹 또는 호스트를 검사하여 서버의 일부분에 대한 액세스를 허용 또는 거부할 수 있으며 인증을 설정하여 오직 권한 있는 사용자와 그룹만 서버의 일부분에 액세스할 수 있도록 합니다 . 액세스 제어에 대한 자세한 내용은 제 8 장 , 147 페이지의 " 서버 액세스 제어 " 를 참조하십시오 .

액세스 제어 목록을 관리하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Administer Access Control 링크를 누릅니다 . Administer Access Control 페이지가 표시됩니다 .
3. 리소스 , 기존 ACL 을 선택하거나 ACL 이름을 입력하고 Edit 버튼을 누릅니다 . Access Control Rules 페이지가 표시됩니다 .
4. 원하는 사항을 변경한 다음 Submit을 누릅니다. 액세스 제어에 대한 자세한 내용은 제 8 장 , 147 페이지의 " 서버 액세스 제어 " 의 " 서버 인스턴스용 액세스 제어 설정 " 을 참조하십시오 .

ACL 캐시 구성

Configure ACL Cache 페이지는 프록시 인증 캐시의 사용 여부 결정 , 프록시 인증 캐시 디렉토리 설정 , 캐시 테이블 크기 구성 , 항목 만료시간 설정에 사용됩니다 .

ACL 캐시를 구성하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Configure ACL Cache 링크를 누릅니다 . Configure ACL Cache 페이지가 표시됩니다 .
3. 프록시 인증 캐시의 사용 여부를 결정할 수 있습니다 .
4. Proxy Auth User Cache Size 드롭다운 목록에서 사용자 캐시의 사용자 수를 선택합니다 . 기본값은 200 입니다 .
5. Proxy Auth Group Cache Size 드롭다운 목록에서 한 UID/ 캐시 항목에 캐시될 수 있는 그룹 ID 수를 선택합니다 . 기본값은 4 입니다 .

6. 캐시 항목이 만료되는 초 단위의 시간을 선택합니다. 캐시에 있는 항목이 참조될 때마다 시간이 계산되고 이 값과 비교됩니다 . 항목의 시간이 Proxy Auth Cache Expiration 값과 같거나 크면 해당 항목은 사용되지 않습니다 . 값을 0 으로 설정하면 캐시를 사용하지 않습니다 .

   이 값에 큰 값을 사용하면 LDAP 항목을 변경할 때 Proxy Server 를 다시 시작해야 합니다 . 예를 들어 이 값을 120 초로 설정하는 경우 최대 2 분까지 Proxy Server 가 LDAP 서버와 동기화되지 않을 수 있습니다 . LDAP 항목아 자주 변경되지 않는다면 큰 값을 사용하십시오 . 기본 만료시간은 2 분입니다 .

7. OK 를 누릅니다 .
8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

DNS 캐싱 이해

Proxy Server 는 DNS 호스트 이름을 IP 주소로 전환하는 동안 프록시가 수행하는 DNS 조회 수를 줄이기 위해 DNS 캐싱을 사용합니다 .

DNS 캐시 구성

Configure DNS Cache 페이지를 사용하여 DNS 캐싱 사용 여부를 설정하고 , DNS 캐시 크기와 DNS 캐시 항목의 만료 시간을 설정하고 , 부정 DNS 캐싱 사용 여부를 설정합니다 .

DNS 캐시를 구성하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Configure DNS Cache 링크를 누릅니다 . Configure DNS Cache 페이지가 표시됩니다 .
3. DNS 캐싱을 사용하거나 또는 사용하지 않도록 설정할 수 있습니다 .
4. DNS Cache Size 드롭다운 목록에서 DNS 캐시에 저장할 수 있는 항목 수를 선택합니다 . 기본값은 1024 입니다 .
5. DNS 캐시 만료 시간을 설정할 수 있습니다. Proxy Server는 미리 정해진 만료 시간이 되면 DNS 캐시에서 항목을 삭제합니다 . 기본 DNS 만료 시간은 20 분입니다 .
6. 호스트 이름을 찾을 수 없는 경우에 오류를 캐시하거나 캐시하지 않도록 설정합니다 .

7. OK 를 누릅니다 .
8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

DNS 하위 도메인 구성

일부 URL 에는 하위 도메인 단계가 많은 호스트 이름이 포함되어 있습니다 . 첫 번째 DNS 서버가 호스트 이름을 확인할 수 없는 경우 프록시 서버의 DNS 확인에 많은 시간이 소요될 수 있습니다 . Proxy Server 가 클라이언트에 "host not found" 메시지를 반환하기까지 확인할 단계의 수를 설정할 수 있습니다 .

예를 들어 클라이언트가 `http://www.sj.ca.example.com/index.html` 을 요청한 경우 프록시는 해당 호스트 컴퓨터의 IP 주소를 확보하기 위해 4 개의 DNS 서버를 통과해야 할 수도 있으며 , 따라서 호스트 이름을 IP 주소로 변환하는 데 많은 시간이 소요될 수 있습니다 . 이러한 조회에는 상당한 시간이 소요되므로 프록시가 특정 수보다 많은 DNS 서버를 사용해야 할 경우 프록시 서버가 IP 주소 조회를 종료하도록 구성할 수 있습니다 .

프록시가 거치는 하위 도메인 단계를 설정하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Configure DNS Subdomains 링크를 누릅니다 . Configure DNS Subdomains 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 정규식을 지정합니다 .
4. Local Subdomain Depth 드롭다운 목록에서 수준의 수를 선택합니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

HTTP 연결 유지 구성

Configure HTTP Client 페이지에서 프록시 서버가 연결 유지 기능을 사용하도록 할 수 있습니다 .

프록시는 HTTP 연결 유지 패킷을 지원합니다 . 기본적으로 프록시는 연결 유지 기능을 사용하지 않지만 일부 시스템에서는 연결 유지 기능을 사용하여 프록시의 성능을 향상시킬 수 있습니다 . 연결 유지 기능은 요청이 완료된 이후에도 연결을 개방 상태로 유지하는 TCP/IP 기능입니다 . 이렇게 하면 클라이언트가 개방된 연결을 신속히 재사용할 수 있습니다 .

웹의 일반적인 클라이언트 - 서버 트랜잭션에서 클라이언트는 서버로 여러 개의 문서를 요청하는 다수의 연결을 만들 수 있습니다 . 예를 들어 클라이언트가 다양한 그래픽 이미지가 있는 웹 페이지를 요청하는 경우 클라이언트는 각 그래픽 파일에 대해 별도의 요청을 해야 합니다 . 이런 상황에서 매번 연결을 다시 만드는 것은 시간 낭비입니다 .

HTTP 연결 유지를 구성하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Configure HTTP Client 링크를 누릅니다 . Configure HTTP Client 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택합니다. Proxy Server에 연결 유지 기능을 구성할 HTTP 또는 HTTPS 리소스를 선택하거나 정규식을 지정합니다 .
4. 적절한 Keep Alive 옵션을 선택하여 HTTP 클라이언트가 지속적인 연결을 사용할지 여부를 지정할 수 있습니다 .
5. Keep Alive Timeout 필드에 연결을 유지할 최대 시간(초)을 지정합니다. 기본값은 29 입니다 .
6. 적절한 Persistent Connection Reuse 옵션을 선택하여 HTTP 클라이언트가 모든 유형의 요청에 대해 기존의 지속적인 연결을 재사용할 수 있는지 여부를 지정할 수 있습니다 . 기본값은 Off 이며 GET 이외의 요청과 본문을 포함하는 요청에 대해 지속 연결을 재사용할 수 없도록 합니다 .
7. HTTP Version String 필드에 HTTP 프로토콜 버전 문자열을 지정합니다 . 특정 프로토콜 상호 운용성 문제가 발생하는 경우에만 이 매개 변수를 지정해야 합니다 .
8. Proxy Agent Header 필드에 Proxy Server 제품 이름과 버전을 지정합니다 .
9. SSL Client Certificate Nickname 필드에 원격 서버에 제시할 클라이언트 인증의 별명을 지정합니다 .
10. 적절한 SSL Server Certificate Validation 옵션을 선택하여 Proxy Server 가 원격 서버에서 제시한 인증서를 검증해야 하는지 여부를 표시합니다 .
11. OK 를 누릅니다 .
12. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .

13. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

8 장

서버 액세스 제어

이 장에서는 Administration Server 에 대한 액세스 및 Proxy Server 에 의해 서비스되는 데이터에 대한 액세스를 제어하는 방법에 대해 설명합니다 . 서버에서 서비스하는 모든 데이터 또는 특정 URL 에 대한 액세스를 제어할 수 있습니다 . 예를 들어 지정한 사람만 특정 URL 에 액세스할 수 있게 하거나 , 또는 지정한 사람을 제외한 모든 사람이 파일을 볼 수 있도록 할 수 있습니다 . HTTP 에 대한 URL 에 모든 클라이언트가 액세스할 수 있도록 하면서 FTP 에 대해서는 제한적으로 액세스를 허용할 수도 있습니다 . 또한 예를 들어 Proxy Server 가 많은 내부 웹 서버를 서비스하고 있는 상태에서 이 중 한 서버에 저장되어 있는 비밀 연구 프로젝트에는 특정한 사람들만 액세스하도록 하려면 호스트 이름과 도메인 이름을 기반으로 URL 을 제한할 수 있습니다 .

Administration Server 에서 액세스 제어를 사용하기 전에 반드시 분산 관리를 사용하도록 설정하고 LDAP 데이터베이스에 관리 그룹을 구성해야 합니다 . 이 장에서는 사용자와 이와 같은 작업을 수행한 것으로 가정하고 설명합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

액세스 제어 설명

액세스 제어를 통해 Proxy Server 에 액세스할 수 있는 사용자 및 이들이 서버에서 액세스할 수 있는 부분을 지정할 수 있습니다 . 전체 서버에 대한 액세스를 제어하거나 , 또는 서버의 일부분 (디렉토리 , 파일 , 파일 유형 등) 에 대해서만 액세스를 제어할 수 있습니다 . 수신 요청을 평가하는 경우 ACE(Access Control Entry) 라는 규칙의 계층을 기반으로 액세스가 결정됩니다 . Proxy Server 는 일치하는 항목을 찾아 액세스를 허용할지 거부할지 결정합니다 . 각 ACE 는 서버가 계층의 다음 항목으로 계속할 것인지의 여부를 지정합니다 . ACE 의 컬렉션을 ACL(Access-Control List) 이라고 합니다 . 요청이 수신되면 ACL 에 대한 참조를 위해 `obj.conf` 파일이 확인되며 , ACL 은 액세스 여부를 결정하는 데 사용됩니다 . 기본으로 서버에는 하나의 ACL 파일이 있으며 여기에는 여러 개의 ACL 이 있습니다 .

액세스를 허용 또는 거부하는 기준은 다음과 같습니다 .

- 요청하는 사용자 (사용자 그룹)
- 요청의 출처 (호스트 IP)
- 요청이 발생한 시간 (예 : 하루 중 시간)
- 사용되는 연결의 종류 (SSL)

이 절에서는 다음 항목에 대해 설명합니다 .

사용자 그룹용 액세스 제어

서버에 대한 액세스를 특정 사용자 또는 그룹으로 제한할 수 있습니다 . 사용자 그룹 액세스 제어를 사용하면 사용자는 해당 서버에 액세스하기 전에 아이디와 비밀 번호를 입력해야 합니다 . 서버는 클라이언트 인증서에 있는 정보 , 또는 클라이언트 인증서 자체를 디렉토리 서버 항목과 비교합니다 .

Administration Server 는 오직 기본 인증만 사용합니다 . Administration Server 에서 클라이언트 인증을 요구하려면 반드시 `obj.conf` 의 ACL 파일을 직접 편집하여 SSL 방법으로 변경해야 합니다 .

사용자 그룹 인증은 서버용으로 구성된 디렉토리 서비스가 수행합니다 . 자세한 내용은 " 디렉토리 서비스 구성 " (47 페이지) 을 참조하십시오 .

디렉토리 서비스가 액세스 제어를 구현하는 데 사용하는 정보는 다음 중 한 가지 소스에서 구합니다 .

- 내부 보통 파일 유형 데이터베이스
- 외부 LDAP 데이터베이스

서버가 외부 LDAP 기반 디렉토리 서비스를 사용하는 경우 서버 인스턴스용으로 다음 유형의 사용자 그룹 인증 방법을 지원합니다 .

- Default
- Basic
- SSL
- Digest
- Other

서버가 내부 파일 기반 디렉토리 서비스를 사용하는 경우 서버 인스턴스용으로 다음 유형의 사용자 그룹 인증 방법을 지원합니다 .

- Default
- Basic
- Digest

사용자 그룹 인증에서는 사용자가 액세스를 허용 받으려면 먼저 자신을 인증해야 합니다. 인증 과정에서 사용자는 아이디와 비밀 번호를 입력하거나 클라이언트 인증서 또는 Digest 인증 플러그인을 사용하여 자신의 신분을 증명합니다 . 클라이언트 인증서를 사용하려면 암호화가 필요합니다 .

Default 인증

Default 인증은 가장 많이 사용되는 방법입니다 . Default 설정은 `obj.conf` 파일에 지정한 기본 방법을 사용하거나 , `obj.conf` 에 설정이 없는 경우에는 Basic 을 사용합니다 . Default 를 선택하는 경우 ACL 규칙은 ACL 파일에 방법을 지정하지 않습니다 . Default 를 선택하면 `obj.conf` 파일에서 한 줄만 편집하면 모든 ACL 에 대한 방법을 쉽게 변경할 수 있습니다 .

Basic 인증

Basic 인증의 경우 사용자가 서버에 액세스하려면 아이디와 비밀 번호를 입력해야 합니다 . 이 설정이 기본값입니다 . 반드시 사용자 및 그룹의 목록을 만들고 이를 Sun Java System Directory Server 등의 LDAP 데이터베이스 또는 파일에 저장해야 합니다 . 반드시 Proxy Server 와 다른 서버 루트에 설치된 디렉토리 서버 또는 원격 컴퓨터에 설치된 디렉토리 서버를 사용해야 합니다 .

사용자가 사용자 그룹 인증을 사용하는 리소스에 액세스를 시도하면 아이디와 비밀 번호를 입력하라는 메시지가 표시됩니다 . 서버에서 암호화 기능 (SSL 사용) 이 사용되는지 여부에 따라 이 정보는 암호화 또는 암호화되지 않은 형태로 서버에 입력됩니다 .

주의	SSL 암호화가 없는 Basic 인증을 사용하는 경우 아이디와 비밀 번호가 암호화되지 않은 텍스트로 네트워크에 전송됩니다 . 이 경우 네트워크 패킷이 가로채여 아이디와 비밀 번호가 도용될 수 있습니다 . Basic 인증은 SSL 암호화나 Host-IP 인증 , 또는 이 둘 모두와 함께 사용할 때 가장 효과적입니다 . Digest 인증을 사용하면 이 문제를 해결할 수 있습니다 .

인증 과정이 완료되면 다음이 표시됩니다 .

- 요청된 리스소 (인증에 성공한 경우)
- 아이디나 비밀 번호가 유효하지 않은 경우 액세스를 거부하는 메시지

권한이 없는 사용자가 받는 메시지를 사용자 정의할 수 있습니다 . 자세한 내용은 " 액세스가 거부된 경우의 응답 " (169 페이지) 을 참조하십시오 .

SSL 인증

서버가 보안 인증서가 있는 사용자의 신분을 확인하는 방법은 두 가지입니다 .

- 클라이언트 인증서의 정보를 신분 증명으로 사용
- LDAP 디렉토리에 게시된 클라이언트 인증서 확인 (추가)

클라이언트 인증용으로 인증서 정보를 사용하도록 서버를 구성하면 서버는 다음 작업을 수행합니다 .

- 인증서가 신뢰할 수 있는 CA(인증 기관) 에서 발행한 것인지 확인합니다 . 그렇지 않은 경우 인증이 실패하고 트랜잭션이 종료됩니다 . 클라이언트 인증을 사용하도록 설정하는 방법은 " 보안 기본 설정 " (88 페이지) 을 참조하십시오 .

- 인증서가 신뢰 CA 에서 발행된 경우 certmap.conf 파일을 사용하여 인증서를 사용자 항목과 매핑합니다 . 인증서 매핑 파일의 구성 방법은 "certmap.conf 파일 사용 " (106 페이지) 을 참조하십시오 .
- 인증서가 올바로 매핑된 경우 해당 사용자에 대해 설정된 ACL 규칙을 확인합니다 . 인증서가 올바르게 매핑된 경우라도 ACL 규칙에 따라 사용자 액세스를 거부할 수 있습니다 .

특정 리소스에 대한 액세스를 제어하는 데 필요한 클라이언트 인증은 서버에 대한 모든 연결에 대해 클라이언트 인증을 요구하는 것과 다릅니다 . 모든 연결에 대해 서버가 클라이언트 인증을 요구하도록 구성한 경우 클라이언트는 신뢰 CA 가 발행한 유효한 인증서만 제시해야 합니다 . 서버가 SSL 방법을 사용하여 사용자 및 그룹을 인증하도록 구성된 경우 다음 사항이 수행되어야 합니다 .

- 클라이언트는 신뢰 CA 가 발행한 유효한 인증서를 제시해야 합니다 .
- 인증서는 반드시 LDAP 의 유효한 사용자와 매핑되어야 합니다 .
- 액세스 제어 목록이 반드시 적절히 평가해야 합니다 .

액세스 제어와 함께 클라이언트 인증을 요구하는 경우 Proxy Server 용 SSL 암호를 사용하도록 설정해야 합니다 . SSL 사용에 대한 자세한 내용은 제 5 장 , 75 페이지의 " 인증서 및 키 사용 " 을 참조하십시오 .

SSL 인증이 요구되는 리소스에 성공적으로 액세스하려면 클라이언트 인증서가 Proxy Server 가 신뢰하는 CA 에서 발행한 것이어야 합니다 . Proxy Server 의 certmap.conf 파일이 브라우저에 있는 클라이언트 인증서를 디렉토리 서버에 있는 클라이언트 인증서와 비교하도록 구성된 경우에는 클라이언트 인증서가 디렉토리 서버 내에 게시되어야 합니다 . 인증서의 선택된 정보만 디렉토리 서버 항목과 비교하도록 certmap.conf 파일을 구성할 수 있습니다 . 예를 들어 브라우저 인증서에 있는 사용자 아이디와 전자 메일 주소만 디렉토리 서버 항목과 비교하도록 certmap.conf 파일을 구성할 수 있습니다 . certmap.conf 와 인증서 매핑에 대한 자세한 내용은 제 5 장 , 75 페이지의 " 인증서 및 키 사용 " 및 Proxy Server Configuration File Reference 를 참조하십시오 .

Digest 인증

LDAP 기반 또는 파일 기반 디렉토리 서버스를 사용하여 Digest 인증을 수행하도록 Proxy Server 를 구성할 수 있습니다 .

Digest 인증을 사용하면 아이디와 비밀 번호를 보통 텍스트로 보내지 않고 아이디 및 비밀 번호를 기반으로 인증할 수 있습니다 . 브라우저는 MD5 알고리즘을 사용하여 Proxy Server 가 제공하는 사용자의 비밀 번호 및 일부 정보를 사용하는 다이제스트 값을 만듭니다 .

서버가 LDAP 기반 디렉토리 서비스를 사용하여 Digest 인증을 수행하는 경우 이 다이제스트 값은 Digest 인증 플러그인을 사용하는 서버에서도 계산되고 클라이언트가 제공하는 다이제스트 값과 비교됩니다 . 다이제스트 값이 일치하면 사용자가 인증됩니다 . 이렇게 하려면 디렉토리 서버가 보통 텍스트의 사용자 비밀 번호에 액세스해야 합니다 . Sun Java System Directory Server 에는 역변환 가능 비밀 번호 플러그인이 있으며 , 이는 데이터를 암호화된 형태로 저장하여 나중에 원래의 형태로 해독할 수 있는 대칭 암호화 알고리즘을 사용합니다 . 오직 Directory Server 만이 데이터의 키를 보유합니다 .

LDAP 기반 인증의 경우 Proxy Server 에 포함된 역변환 가능 비밀 번호 플러그인과 Digest 인증별 플러그인을 사용하도록 설정해야 합니다 . Proxy Server 가 Digest 인증을 처리하도록 구성하려면 *server_root*`/userdb/` 에 있는 `dbswitch.conf` 파일에서 데이터베이스 정의의 `digestauth` 등록 정보를 설정해야 합니다 .

서버는 표 8-1 에서와 같이 지정된 ACL 방법에 기반하여 LDAP 데이터베이스에 대한 인증을 시도합니다 . ACL 방법을 지정하지 않은 경우 서버는 인증이 요구되는 경우 Digest 또는 Basic 을 사용하며 , 인증이 요구되지 않는 경우에는 Basic 을 사용합니다 .

인증 데이터베이스에서 지원 및 지원하지 않는 Digest 인증은 다음 표의 목록을 참조하십시오 .

표 8-1 Digest 인증 챌린지 생성

ACL 방법	인증 데이터베이스에서 지원	인증 데이터베이스에서 지원되지 않음
Default 지정된 사항 없음	Digest 및 Basic	Basic
Basic	Basic	Basic
Digest	Digest	ERROR

`method=digest` 로 설정된 ACL 을 처리하는 경우 서버는 다음과 같이 인증을 시도합니다 .

- 인증 요청 헤더 확인 . 없는 경우 Digest 챌린지를 포함하는 401 응답이 생성되며 프로세스는 정지합니다 .
- 인증 유형 확인 . 인증 유형이 Digest 인 경우 서버는 다음을 수행합니다 .

- nonce 를 확인합니다 . 유효하지 않은 경우 이 서버가 새 nonce 를 생성하며 , 401 응답이 생성되고 프로세스가 정지됩니다 . 오래된 경우 `stale=true` 로 설정된 401 응답이 생성되며 프로세스가 정지됩니다 .

  server_root`/proxy-server_name/config/` 에 있는 `magnus.conf` 파일의 `DigestStaleTimeout` 매개 변수 값을 변경하여 nonce 가 새로운 상태를 유지하는 시간을 구성할 수 있습니다 . 이 값을 설정하려면 `magnus.conf` 에 다음과 같은 줄을 추가합니다 .

  `DigestStaleTimeout` *seconds*

  여기에서 *seconds* 는 nonce 가 새로운 상태를 유지하는 초 단위 시간입니다 . 지정된 시간이 경과하면 nonce 가 만기되며 사용자에 대한 새로운 인증이 요구됩니다 .

- 영역을 확인합니다 . 일치하지 않는 경우 401 응답이 생성되며 프로세스가 정지됩니다 .
- 인증 디렉토리가 LDAP 기반인 경우 LDAP 디렉토리에 사용자가 있는지 확인하며 인증 디렉토리가 파일 기반인 경우 파일 데이터베이스에 사용자가 있는지 확인합니다 . 없는 경우 401 응답이 생성되며 프로세스가 정지됩니다 .
- 디렉토리 서버 또는 파일 데이터베이스에서 `request-digest` 값을 가져오고 클라이언트의 `request-digest` 와 일치하는지 확인합니다 . 일치하지 않는 경우 401 응답이 생성되며 프로세스가 정지됩니다 .
- `Authorization-Info` 헤더를 만들고 이를 서버 헤더에 삽입합니다 .

Digest 인증 플러그인 설치

LDAP 기반 디렉토리 서비스를 사용하는 Digest 인증의 경우 Digest 인증 플러그인을 설치해야 합니다 . 이 플러그인은 서버 측의 다이제스트 값을 계산하고 이를 클라이언트가 제공한 다이제스트 값과 비교합니다 . 다이제스트 값이 일치하면 사용자가 인증됩니다 .

파일 기반 인증 데이터베이스를 사용하는 경우 Digest 인증 플러그인을 설치할 필요가 없습니다 .

UNIX 에 Digest 인증 플러그인 설치

Digest 인증 플러그인은 다음 공유 라이브러리로 구성됩니다 .

- `libdigest-plugin.lib`
- `libdigest-plugin.ldif`

UNIX 에 Digest 인증 플러그인을 설치하려면 다음을 수행합니다 .

1. 이 공유 라이브러리가 Sun Java System Directory Server 를 설치한 동일한 서버 컴퓨터에 있는지 확인합니다 .
2. Directory Manager 비밀 번호가 올바른지 확인합니다 .
3. `libdigest-plugin.ldif` 파일을 수정해서 `/path/to` 에 대한 모든 참조를 다이제스트 플러그인 공유 라이브러리를 설치한 위치로 변경합니다 .
4. 플러그인을 설치하려면 다음 명령을 입력합니다 .

```
% ldapmodify -D "cn=Directory Manager" -w password -a <
libdigest-plugin.ldif
```

Windows 에 Digest 인증 플러그인 설치

Directory Server 가 Digest 플러그인과 함께 적절히 시작되려면 여러 개의 `.dll` 파일을 Proxy Server 설치 위치에서 Sun Java System Directory Server 서버 컴퓨터로 복사해야 합니다 .

Windows 에 Digest 인증 플러그인을 설치하려면 다음을 수행합니다 .

1. 다음 위치에 있는 Proxy Server 의 공유 라이브러리에 액세스합니다 .

   server_root`\bin\proxy\bin`
2. 다음 파일을 복사합니다 .
   - `nsldap32v50.dll`
   - `libspnr4.dll`
   - `libplds4.dll`
3. 복사한 파일을 다음 중 한 곳에 붙여넣습니다 .
   - `\Winnt\system32`
   - Sun Java System Directory Server 설치 디렉토리 : *server_root*`\bin\sldap\server`

DES 알고리즘 사용을 위한 Sun Java System Directory Server 설정

다이제스트 비밀 번호가 저장된 위치의 속성을 암호화하려면 DES 알고리즘이 필요합니다 .

DES 알고리즘 사용을 위해 Directory Server 를 설정하려면 다음을 수행합니다 .

1. Sun Java System Directory Server 콘솔을 시작합니다 .
2. iDS 5.0 인스턴스를 엽니다 .
3. Configuration 탭을 선택합니다 .

4. 플러그인 옆의 + 기호를 누릅니다 .
5. DES 플러그인을 선택합니다 .
6. Add 를 선택하여 새 속성을 추가합니다 .
7. `iplanetReversiblePassword` 를 입력합니다 .
8. Save 를 누릅니다 .
9. Sun Java System Directory Server 인스턴스를 다시 시작합니다 .

참고 사용자에 대한 `iplanetReversiblePassword` 속성의 Digest 인증 비밀 번호를 설정하려면 항목에 `iplanetReversiblePasswordobject` 개체를 포함시켜야 합니다 .

Other 인증

액세스 제어 API 를 사용하여 사용자 정의 인증 방법을 만들 수 있습니다 .

Host-IP 용 액세스 제어

Administration Server 및 해당 파일과 디렉토리를 특정 컴퓨터를 이용하는 클라이언트만 사용할 수 있도록 설정하여 이에 대한 액세스를 제한할 수 있습니다 . 액세스를 허용 또는 거부하려는 컴퓨터의 호스트 이름 또는 IP 주소를 지정합니다 . Host-IP 인증을 사용하는 파일 또는 디렉토리에 대한 액세스는 사용자도 모르게 진행됩니다 . 사용자는 아이디 또는 비밀 번호를 입력하지 않고 즉시 파일과 디렉토리에 액세스할 수 있습니다 .

여러 사람이 특정 컴퓨터를 사용할 수 있으므로 Host-IP 인증은 사용자 그룹 인증과 함께 사용할 때 더욱 효과적입니다 . 두 가지 인증 방법이 모두 사용되는 경우 액세스할 때 아이디와 비밀 번호가 필요합니다 .

Host-IP 인증의 경우 서버에서 DNS(Domain Name Service) 를 구성할 필요가 없습니다 . Host-IP 인증을 선택한 경우 반드시 DNS 가 네트워크에서 실행되어야 하며 서버가 이를 사용하도록 구성되어야 합니다 . DNS³¶ 사용하도록 설정하려면 서버의 Server Manager 에 액세스한 다음 Preferences 탭을 누르고 Configure System Preferences 를 누릅니다 . DNS 설정이 표시됩니다 .

DNS 를 사용하면 서버가 DNS 조회를 수행해야 하므로 Proxy Server 의 성능이 낮아집니다 . DNS 조회가 서버 성능에 미치는 영향을 낮추려면 모든 요청의 IP 주소를 변환하는 대신 오직 액세스 제어 및 CGI 용 IP 주소만 변환합니다 . 이렇게 하려면 `obj.conf` 에서 다음을 지정합니다 .

```
AddLog fn="flex-log" name="access" iponly=1
```

액세스 제어 파일 사용

Administration Server 또는 서버의 파일이나 디렉토리에서 액세스 제어를 사용하는 경우 해당 설정은 확장자가 .acl 인 파일에 저장됩니다 . 액세스 제어 파일은 *server_root*/httpacl 디렉토리에 저장됩니다 (*server_root*: 서버가 설치된 위치). 예를 들어 서버를 /usr/Sun/Servers 에 설치한 경우 Administration Server 와 서버에 구성된 각 서버 인스턴스용 ACL 파일의 위치는 /usr/Sun/Servers/httpacl/ 입니다 .

기본 ACL 파일은 generated-proxy-*serverid*.acl 입니다 . 임시 작업 파일은 genwork-proxy-*serverid*.acl 입니다 . Administration Server 를 사용하여 액세스를 구성하는 경우 이 두 파일이 만들어집니다 . 그러나 제한을 더욱 복잡하게 하려면 여러 개의 파일을 만들고 server.xml 파일에서 이들 파일을 참조합니다 . 또한 하루 중 시간 또는 주중 요일을 기준으로 서버에 대한 액세스를 제한하는 등 , 파일을 편집한 경우에만 사용할 수 있는 몇 가지 기능이 있습니다 .

액세스 제어 파일 및 구문에 대한 자세한 내용은 부록 A, 373 페이지의 "ACL 파일 구문 " 을 참조하십시오 . server.xml 에 대한 자세한 내용은 Proxy Server Configuration File Reference 를 참조하십시오 .

ACL 사용자 캐시 구성

기본적으로 Proxy Server 는 사용자 및 그룹 인증 결과를 ACL 사용자 캐시에 캐시합니다 . magnus.conf 파일의 ACLCacheLifetime 지시문을 사용하여 ACL 사용자 캐시의 유효 시간을 조정할 수 있습니다 . 캐시에 있는 항목이 참조될 때마다 시간이 계산되고 ACLCacheLifetime 과 비교됩니다 . 항목의 시간이 ACLCacheLifetime 과 같거나 크면 해당 항목은 사용되지 않습니다 . 기본값은 120 초입니다 . 값을 0 으로 설정하면 캐시가 Off 로 설정됩니다 . 이 값에 큰 값을 사용하면 LDAP 항목을 변경할 때마다 Proxy Server 를 다시 시작해야 할 수도 있습니다 . 예를 들어 이 값을 120 초로 설정하는 경우 최대 2 분까지 Proxy Server 가 LDAP 디렉토리와 동기화되지 않을 수 있습니다 . LDAP 디렉토리가 자주 변경되지 않는 경우에만 큰 값을 사용하십시오 .

ACLUserCacheSize 의 magnus.conf 매개 변수를 사용하여 캐시에 유지할 항목의 최대 수를 구성할 수 있습니다 . 이 매개 변수의 기본값은 200 입니다 . 새 항목은 목록의 앞에 추가되며 목록의 끝에 있는 항목은 캐시가 최대 크기에 도달하면 재활용되어 새로운 항목이 됩니다 .

또한 magnus.conf 매개 변수 ACLGroupCacheSize 를 사용하여 각 사용자 항목마다 캐시될 수 있는 그룹 구성원의 최대 수를 설정할 수 있습니다 . 이 매개 변수의 기본값은 4 입니다 . 유감스럽게도 그룹에 있는 사용자가 구성원이 아닌 경우 캐시되지 않으며 , 요청마다 여러 LDAP 디렉토리 액세스가 발생하게 됩니다 .

클라이언트 인증서로 액세스 제어

서버에서 SSL 을 사용하도록 설정한 경우 액세스 제어와 관련하여 클라이언트 인증서를 사용할 수 있습니다 . 이를 위해서는 특정 리소스에 액세스하려면 클라이언트 인증서가 필요하도록 지정해야 합니다 . 서버에서 이 기능을 사용하도록 설정하면 인증서가 있는 사용자는 제한된 리소스에 처음 액세스를 시도할 때만 아이디와 비밀 번호를 입력하면 됩니다 . 신분이 설정되면 서버는 사용자의 로그인 이름과 비밀 번호를 해당 특정 인증서에 매핑합니다 . 이후 사용자는 클라이언트 인증이 필요한 리소스에 액세스할 때 로그인 이름이나 비밀 번호를 입력할 필요가 없습니다 . 사용자가 제한된 리소스에 대한 액세스를 시도하면 클라이언트는 서버에 클라이언트 인증서를 보내고 , 서버는 이 인증서를 매핑 목록과 비교 확인합니다 . 이 인증서가 액세스 권한이 허용된 사용자에게 속한 경우 리소스를 해당 사용자에게 서비스합니다 .

참고 특정 리소스에 대한 액세스를 제어하는 데 필요한 클라이언트 인증은 서버에 대한 모든 연결에 대해 클라이언트 인증을 요구하는 것과 다릅니다 . 또한 모든 SSL 연결에 대해 클라이언트 인증서를 요구한다고 해서 인증서가 자동으로 데이터베이스에 있는 사용자에게 매핑되지는 않는다는 점을 유의하십시오 . 이를 위해서는 클라이언트 인증서가 특정 리소스에 액세스하는 데 필요하도록 지정해야 합니다 .

액세스 제어 작동 원리

서버에서 페이지에 대한 요청을 수신하면 서버는 ACL 파일에 있는 규칙을 사용하여 액세스를 허용할지 여부를 결정합니다. 규칙은 요청을 보내는 컴퓨터의 호스트 이름 또는 IP 주소를 참조할 수 있습니다 . 또한 LDAP 디렉토리에 저장된 사용자 및 그룹을 참조할 수 있습니다 .

다음 예는 ACL 파일의 내용 예와 액세스 제어 규칙 예를 보여 줍니다 .

```
version 3.0;
# The following "es-internal" rules protect files such
# as icons and images related to Sun Java System Web Proxy Server.
```

```
# These "es-internal" rules should not be modified.
  acl "es-internal";
  allow (read, list, execute,info) user = "anyone";
  deny (write, delete) user = "anyone";

# The following rules deny access to the directory "web"
# to everyone not in the directory server and deny everyone
# in the directory server who is not in GroupB.
# Only the users in GroupB are allowed read, execute, list,
# and info permissions. GroupA cannot gain access to the
# directory "web" even though (in the ACL rule below) they
# can access the directory "my_stuff". Furthermore, members
# of GroupB cannot write or delete files.
  acl "path=/export/user/990628.1/docs/my_stuff/web/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny(all)
  (user = "anyone");

  allow (read,execute,list,info)
  (group = "GroupB");

# The following rule denies everyone not in the directory
# server and denies everyone in the directory server except
# users with the ID of "SpecificMemberOfGroupB". The ACL rule
# in this setting also has a requirement that the user
# connect from a specific IP address. The IP address setting
# in the rule is optional, and has been added for extra
# security. Also, this ACL rule has a Customized prompt
# of "Presentation Owner". This Customized prompt appears
# in the username and password dialog box in the client's
# browser.

  acl "path=/export/user/990628.1/docs/my_stuff/web/presentation.html";
  authenticate (user,group) {
     database = "default";
     method = "basic";
     prompt = "Presentation Owner";
  };
  deny(all)
  (user = "anyone" or group = "my_group");
  allow (all)
  (user = "SpecificMemberOfGroupB") and
  (ip = "208.12.54.76");

# The following ACL rule denies everyone not in the directory
# server and everyone in the directory server except for
# GroupA and GroupB access to the directory "my_stuff"
  acl "path=/export/user/990628.1/docs/my_stuff/";
  authenticate (user,group) {
     database = "default";
     method = "basic";
  };
  deny(all)
```

```
(user = "anyone");
allow (read,execute,list,info)
(group = "GroupA,GroupB");
```

예를 들어 사용자가 다음 URL 을 요청하는 경우 ,
`http://server_name/my_stuff/web/presentation.html`

Proxy Server는 우선 전체 서버에 대한 액세스 제어를 확인합니다. 전체 서버용 ACL 이 계속으로 설정된 경우 서버는 `my_stuff` 디렉토리용 ACL 을 확인합니다 . ACL 이 존재하면 서버는 ACL 에 있는 ACE 를 확인한 후 , 다음 디렉토리로 이동합니다 . 이 프로세스는 액세스를 거부하는 ACL 이 발견되거나 요청된 URL 에 대한 마지막 URL(이 경우에는 `presentation.html` 파일) 에 도달할 때까지 계속됩니다 .

Server Manager 를 이용하여 이 예제의 액세스 제어를 설정하려면 파일 전용 또는 파일로 유도되는 각 리소스용 ACL 을 만들 수 있습니다 . 즉 , 전체 서버용 1 개 , `my_stuff` 디렉토리용 1 개 , `my_stuff/web` 디렉토리용 1 개 및 해당 파일용 1 개를 만들 수 있습니다 .

참고 일치하는 ACL 이 하나 이상인 경우 서버는 일치하는 항목이 있는 마지막 ACL 문을 사용합니다 .

액세스 제어 설정

이 절에서는 액세스를 제한하는 프로세스에 대해 설명합니다 . 모든 서버에 대한 전역 액세스 제어 규칙 및 특정 서버에 대한 개별 규칙을 설정할 수 있습니다 . 예를 들어 인력 관리 부서에서는 모든 인증된 사용자가 자신의 급여 데이터를 볼 수 있도록 허용하면서 데이터 업데이트를 위한 액세스는 인력 관리 부서의 급여 담당 직원만 가능하도록 제한하는 ACL 을 만들 수 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 전역 액세스 제어 설정
- 서버 인스턴스용 액세스 제어 설정

참고 전역 액세스 제어를 설정하기 전에 반드시 분산 관리를 구성하고 사용해야 합니다 .

전역 액세스 제어 설정

모든 서버에 대한 액세스 제어를 설정하려면 다음을 수행합니다 .

1. Administration Server 에 액세스하고 Global Settings 탭을 누릅니다 .
2. Administer Access Control 링크를 누릅니다 .
3. 드롭다운 목록에서 administration server(`proxy-admserv`) 를 선택하고 Go 를 눌러 데이터를 로드한 다음 New ACL 또는 Edit ACL 을 누릅니다 .
4. 메시지가 표시되면 인증합니다 . Access Control Rules For 페이지가 표시됩니다 . Administration Server 에는 편집할 수 없는 기본 액세스 제어 규칙이 두 줄 있습니다 .
5. Access Control Is On 이 아직 선택되지 않았으면 선택합니다 .
6. 표의 하단에 기본 ACL 규칙을 추가하려면 New Line 버튼을 누릅니다 . 액세스 제어 제한의 위치를 변경하려면 위 / 아래 화살표를 누릅니다 .
7. Users/Groups 열에서 Anyone 을 누릅니다 . User/Group 페이지가 아래 창에 표시됩니다 .
8. 액세스를 허용할 사용자 및 그룹을 선택하고 Update 를 누릅니다 . Group 또는 User 에 대한 List 버튼을 누르면 선택할 수 있는 목록이 표시됩니다 . 설정에 대한 자세한 내용은 온라인 도움말 및 " 사용자 및 그룹 지정 " (164 페이지) 을 참조하십시오 .
9. From Host 열에서 Anyplace 를 누릅니다 . From Host 페이지가 아래 창에 표시됩니다 .
10. 액세스가 허용된 호스트 이름과 IP 주소를 지정하고 Update 를 누릅니다 . 설정에 대한 자세한 내용은 온라인 도움말 및 "From Host 지정 " (166 페이지) 을 참조하십시오 .
11. Programs 열에서 All 을 누릅니다 . Programs 페이지가 아래 창에 표시됩니다 .
12. 액세스를 허용할 Program Groups 를 선택하거나 Program Items 필드에 특정 파일 이름을 입력한 다음 Update 를 누릅니다 . 설정에 대한 자세한 내용은 온라인 도움말 및 " 프로그램에 대한 액세스 제한 " (167 페이지) 을 참조하십시오 .
13. (선택 사항) 사용자 정의 ACL 표현식을 추가하려면 Extra 열에서 X 를 누릅니다 . Customized Expressions 페이지가 아래 창에 표시됩니다 . 자세한 내용은 " 사용자 정의 표현식 작성 " (168 페이지) 을 참조하십시오 .
14. Continue 열에 확인란이 선택되지 않았으면 선택합니다 . 서버는 사용자의 액세스가 허용되었는지 결정하기 전에 다음 줄을 확인합니다 . 여러 줄을 만드는 경우에는 가장 일반적인 제한에서 가장 구체적인 제한으로 진행합니다 .

15. (선택 사항) 휴지통 아이콘을 누르면 액세스 제어 규칙에서 해당 줄을 삭제합니다 .

16. (선택 사항) 액세스가 거부되었을 때 사용자가 수신하는 응답을 지정하려면 Response When Denied 링크를 누릅니다 . Access Deny Response 페이지가 아래 창에 표시됩니다 . 원하는 응답을 선택한 다음 필요한 경우 추가 정보를 지정하고 Update 를 누릅니다 . 설정에 대한 자세한 내용은 " 액세스가 거부된 경우의 응답 " (169 페이지) 을 참조하십시오 .

17. Submit 를 눌러 ACL 파일에 새 액세스 제어 규칙을 저장하거나 Revert 를 눌러 페이지의 요소를 변경하기 전에 포함된 값으로 재설정합니다 .

서버 인스턴스용 액세스 제어 설정

Server Manager 를 사용하여 특정 서버 인스턴스용 액세스 제어를 만들거나 편집 또는 삭제할 수 있습니다 . 삭제하는 경우 ACL 파일의 모든 ACL 규칙을 삭제하면 안 됩니다 . 서버를 시작하려면 ACL 규칙을 한 개 이상 포함하는 ACL 파일이 적어도 하나 이상 있어야 합니다 . ACL 규칙을 모두 삭제하고 서버를 재시작하면 구문 오류가 발생합니다 .

서버 인스턴스에 대한 액세스 제어를 설정하려면 다음을 수행합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .

2. Administer Access Control 링크를 누릅니다 .

3. 다음 중 한 가지 방법을 사용하여 ACL 을 선택합니다 .

   - Select A Resource 는 ACL 을 사용하여 액세스를 제한하는 리소스를 표시합니다 . 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 을 눌러 정규식을 지정합니다 . 자세한 내용은 Proxy Server 관리자 설명서의 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 를 참조하십시오 .

   - Select An Existing ACL은 사용하도록 설정된 모든 ACL을 표시합니다. 사용하도록 설정되지 않은 기존 ACL 은 이 목록에 표시되지 않습니다 . 드롭다운 목록에서 선택합니다 .

   - Type In The ACL Name을 사용하여 named ACL을 만들 수 있습니다. 이 옵션은 ACL 파일에 익숙한 경우에만 사용하십시오 . named ACL 을 리소스에 적용하려는 경우에는 `obj.conf` 를 직접 편집해야 합니다 . 자세한 내용은 부록 A, 373 페이지의 "ACL 파일 구문 " 을 참조하십시오 .

4. 해당 Edit 버튼을 누릅니다 . Access Control Rules For 페이지가 표시됩니다 .

5. Access Control Is On 이 아직 선택되지 않았으면 선택합니다 .

6. 표의 하단에 기본 ACL 규칙을 추가하려면 New Line 버튼을 누릅니다 . 액세스 제어 제한의 위치를 변경하려면 위 / 아래 화살표를 누릅니다 .

7. 이 서버 인스턴스용 ACL 을 편집하려면 Action 열에서 해당 작업을 누릅니다 . Allow/Deny 페이지가 아래 창에 표시됩니다 .

8. Allow 가 아직 기본값으로 선택되지 않았으면 선택하고 Update 를 누릅니다 . Allow 또는 Deny 에 대한 자세한 내용은 " 작동 설정 " (163 페이지) 을 참조하십시오 .

9. Users/Groups 열에서 Anyone 을 누릅니다 . User/Group 페이지가 아래 창에 표시됩니다 .

10. 액세스를 허용할 사용자 및 그룹을 선택하고 인증 정보를 지정한 다음 Update를 누릅니다 . Group 또는 User 에 대한 List 버튼을 누르면 선택할 수 있는 목록이 표시됩니다 . 설정에 대한 자세한 내용은 온라인 도움말 및 " 사용자 및 그룹 지정 " (164 페이지) 을 참조하십시오 .

11. From Host 열에서 Anyplace 를 누릅니다 . From Host 페이지가 아래 창에 표시됩니다 .

12. 액세스가 허용된 호스트 이름과 IP 주소를 지정하고 Update 를 누릅니다 . 설정에 대한 자세한 내용은 온라인 도움말 및 "From Host 지정 " (166 페이지) 을 참조하십시오 .

13. Rights 열에서 All 을 누릅니다 . Access Rights 페이지가 아래 창에 표시됩니다 .

14. 이 사용자에 대한 액세스 권한을 지정한 다음 Update 를 누릅니다 . 자세한 내용은 " 프로그램에 대한 액세스 제한 " (167 페이지) 을 참조하십시오 .

15. (선택 사항) 사용자 정의 ACL 표현식을 추가하려면 Extra 열 아래의 X 를 누릅니다 . Customized Expressions 페이지가 아래 창에 표시됩니다 . 자세한 내용은 " 사용자 정의 표현식 작성 " (168 페이지) 을 참조하십시오 .

16. Continue 열에서 확인란이 선택되지 않았으면 선택합니다 . 서버는 사용자의 액세스가 허용되었는지 결정하기 전에 다음 줄을 확인합니다 . 여러 줄을 만드는 경우에는 가장 일반적인 제한에서 가장 구체적인 제한으로 진행합니다 .

17. (선택 사항) 휴지통 아이콘을 누르면 액세스 제어 규칙에서 해당 줄을 삭제합니다 . ACL 파일의 모든 ACL 규칙을 삭제하면 안 됩니다 . 서버를 시작하려면 ACL 규칙을 최소한 한 개 이상 포함하는 ACL 파일이 적어도 하나 이상 있어야 하기 때문입니다 . ACL 파일에서 ACL 규칙을 모두 삭제하고 서버를 재시작하면 구문 오류가 발생합니다 .

18. (선택 사항) 액세스가 거부되었을 때 사용자가 수신하는 응답을 지정하려면 Response When Denied 링크를 누릅니다 . Access Deny Response 페이지가 아래 창에 표시됩니다 . 원하는 응답을 선택한 다음 필요한 경우 추가 정보를 지정하고 Update 를 누릅니다 . 설정에 대한 자세한 내용은 " 액세스가 거부된 경우의 응답 " (169 페이지) 을 참조하십시오 .
19. Submit 를 눌러 ACL 파일에 새 액세스 제어 규칙을 저장하거나 Revert 를 눌러 페이지의 요소를 변경하기 전에 포함된 값으로 재설정합니다 .

액세스 제어 옵션 선택

다음 항목에서는 액세스 제어를 설정할 때 선택할 수 있는 다양한 옵션에 대해 설명합니다 . Administration Server 의 경우 첫 두 줄은 기본으로 설정되며 편집할 수 없습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

작동 설정

요청이 액세스 제어 규칙과 일치할 때 서버의 작동을 지정할 수 있습니다 .

- **Allow**: 사용자 또는 시스템이 요청된 리소스에 액세스할 수 있습니다 .
- **Deny**: 사용자 또는 시스템이 리소스에 액세스할 수 없습니다 .

서버는 ACE(Access Control Entry) 목록 전체를 확인하여 액세스 권한을 판단합니다 . 예를 들어 첫 번째 ACE 는 보통 모든 사용자를 거부합니다 . 첫 번째 ACE 가 Continue 로 설정된 경우 서버는 목록의 두 번째 ACE 를 확인하며 , 일치되는 경우 다음 ACE 를 사용합니다 . Continue 가 선택되지 않은 경우 모든 사용자의 리소스에 대한 액세스를 거부합니다 . 서버는 일치되지 않는 ACE 가 발견되거나 , 일치되지만 Continue 가 설정되지 않은 ACE 를 발견할 때까지 목록을 계속 확인합니다 . 마지막으로 일치되는 ACE 에 따라 액세스의 허용 또는 거부가 결정됩니다 .

사용자 및 그룹 지정

사용자 및 그룹 인증을 사용하면 사용자가 액세스 제어 규칙에 지정된 리소스에 액세스하기 전에 아이디 및 비밀 번호를 입력하라는 메시지가 표시됩니다 .

Proxy Server 는 Sun Java System Directory Server 등의 LDAP 서버 또는 내부 파일 기반 인증 데이터베이스에 저장된 사용자 및 그룹 목록을 확인합니다 .

데이터베이스에 있는 모든 사용자의 액세스를 허용 또는 거부할 수 있으며 , 와일드카드 패턴을 사용하여 특정 사용자를 허용 또는 거부할 수 있습니다 . 또한 사용자 및 그룹 목록에서 허용 또는 거부할 사용자를 선택할 수 있습니다 .

사용자 인터페이스의 Access Control Rules For 페이지에서 Users/Groups 에 대해 다음 요소가 표시됩니다 .

- **Anyone(No Authentication)** 은 기본값으로 모든 사용자가 아이디 및 비밀 번호를 입력하지 않고 리소스에 액세스할 수 있습니다 . 그러나 호스트 이름 또는 IP 주소 등의 기타 설정에 따라 액세스를 거부할 수 있습니다 . Administration Server 의 경우 분산 관리에 대해 지정한 관리 그룹의 모든 사용자가 페이지에 액세스할 수 있습니다 .
- **Authenticated People Only**
  - **All In The Authentication Database** 는 데이터베이스에 항목이 있는 임의 사용자를 일치시킵니다 .
  - **Only The following People**은 일치할 사용자 및 그룹을 지정합니다. 사용자 또는 사용자 그룹의 목록을 만들 수 있으며 각 항목을 쉼표로 분리하거나 , 와일드카드 패턴을 사용할 수 있습니다 . 또는 데이터베이스에 저장된 사용자 및 그룹 목록에서 선택할 수 있습니다 . **Group** 은 지정한 그룹의 모든 사용자를 일치시킵니다 . **User** 는 지정한 개별 사용자를 일치시킵니다 . Administration Server 의 경우 사용자는 반드시 분산된 관리용으로 지정한 관리 그룹에 속해야 합니다 .

- **Prompt For Authentication**은 인증 대화 상자에 표시되는 메시지 텍스트를 지정합니다 . 이 텍스트를 사용하여 사용자가 입력해야 할 것을 설명할 수 있습니다 . 운영 체제에 따라 사용자는 프롬프트의 첫 40 자 정도만 보게 될 수 있습니다 . 대부분의 브라우저는 사용자 이름과 암호를 캐시하여 프롬프트 텍스트로 연결하기 때문에 사용자가 동일한 프롬프트가 있는 서버의 영역 (파일 및 디렉토리) 에 액세스하는 경우 사용자 이름과 비밀 번호를 다시 입력하지 않아도 됩니다 . 반대로 영역마다 사용자가 인증하도록 하려면 반드시 해당 리소스의 ACL 용 프롬프트를 변경해야 합니다 .
- **Authentication Methods** 는 서버가 클라이언트에서 인증 정보를 가져올 때 사용하는 방법을 지정합니다 . Administration Server 의 경우 오직 Basic 인증 방법만 제공됩니다 . Server Manager 에서는 다음을 제공합니다 .
  - **Default** 는 `obj.conf` 에 지정한 기본 방법을 사용하거나 , `obj.conf` 에 설정이 없는 경우에는 Basic 을 사용합니다 . Default 를 선택하는 경우 ACL 규칙은 ACL 파일에 방법을 지정하지 않습니다 . Default 를 선택하면 `obj.conf` 파일에서 한 줄만 편집하여 모든 ACL 에 대한 방법을 쉽게 변경할 수 있습니다 .
  - **Basic**은 HTTP 메소드를 사용하여 클라이언트에서 인증 정보를 가져옵니다. 서버에 대해 암호화를 사용하는 경우 (SSL 사용 설정) 에만 아이디와 비밀 번호가 암호화됩니다 . 그렇지 않은 경우 아이디와 비밀 번호는 보통 텍스트로 전송되기 때문에 가로채여 노출될 수 있습니다 .
  - **SSL**은 클라이언트 인증서를 사용하여 사용자를 인증합니다. 이 방법을 사용하려면 반드시 서버용에 SSL 을 사용해야 합니다 . 암호화를 사용하는 경우 Basic 과 SSL 방법을 결합하여 사용할 수 있습니다 .
  - **Digest** 는 아이디와 비밀 번호를 보통 텍스트로 전송하지 않고서도 브라우저에서 아이디와 비밀 번호를 기반으로 사용자를 인증할 수 있는 인증 방법을 사용합니다 . 브라우저는 MD5 알고리즘을 사용하여 Proxy Server 가 제공하는 사용자의 비밀 번호 및 일부 정보를 사용하는 다이제스트 값을 만듭니다 . 다이제스트 값은 또한 Digest 인증 플러그인을 사용하는 서버 측에서도 계산되며 이 값은 클라이언트가 제공하는 다이제스트 값과 비교됩니다 .
  - **Other**는 액세스 제어 API 를 사용하여 만든 사용자 정의 방법을 사용합니다.
- **Authentication Database** 는 서버가 사용자를 인증하는 데 사용하는 데이터베이스를 지정합니다 . 이 옵션은 오직 Server Manager 를 통하여만 사용할 수 있습니다 . Default 를 선택하는 경우 서버는 기본으로 구성된 디렉토리 서비스에서 사용자 및 그룹을 찾습니다 . 개별 ACL 을 구성하여 서로 다른 데이터베이스에 사용하려면 Other 를 선택하고 데이터베이스를 지정합니다 . 기본이 아닌 데이터베이스와 LDAP 디렉토리는 *server_root*`/userdb/dbswitch.conf` 에 지정되어야 합니다 . 사용자 정의 데이터베이스용 액세스 제어 API 를 사용하는 경우 Other 를 선택하고 데이터베이스 이름을 입력합니다 .

From Host 지정

요청을 보내는 컴퓨터를 기준으로 Administration Server 에 대한 액세스를 제한할 수 있습니다 .

사용자 인터페이스의 Access Control Rules For 페이지에서 From Host 에 대해 다음 요소가 표시됩니다 .

- **Anyplace**: 모든 사용자 및 시스템의 엑세스를 허용
- **Only From**: 특정 호스트 이름 또는 IP 주소에 대한 액세스 제한

Only From 옵션을 선택한 경우 Host Names 또는 IP Address 필드에 와일드카드 패턴 또는 쉼표로 분리된 목록을 입력합니다 . 호스트 이름을 기준으로 제한하는 것이 IP 주소를 기준으로 제한하는 것 보다 더 유연합니다 . 사용자의 IP 주소가 변경되어도 목록을 업데이트할 필요가 없습니다 . 그러나 IP 주소를 기준으로 제한하는 것이 더욱 안전합니다 . 연결된 클라이언트에 대한 DNS 조회가 실패하는 경우 호스트 이름 제한을 사용할 수 없습니다 .

컴퓨터의 호스트 이름 또는 IP 주소와 일치하는 와일드카드 패턴에는 오직 * 와일드카드만 사용할 수 있습니다 . 예를 들어 , 특정 도메인에 있는 모든 컴퓨터를 허용 또는 거부하려면 `*.example.com` 과 같이 해당 도메인의 모든 호스트에 일치하는 와일드카드 패턴을 입력합니다 . Administration Server 에 액세스하는 수퍼유저용으로 다른 호스트 이름 및 IP 주소를 설정할 수 있습니다 .

호스트 이름의 경우 * 는 반드시 호스트 이름의 전체 구성 요소를 대신해야 합니다 . 즉 , `*.example.com` 은 사용 가능하지만 `*users.example.com` 은 사용할 수 없습니다 . 호스트 이름에 * 가 있는 경우 반드시 문자의 가장 왼쪽에 있어야 합니다 . 예를 들어 `*.example.com` 은 사용 가능하지만 `users.*.com` 은 사용할 수 없습니다 .

IP 주소의 경우 * 는 반드시 IP 주소의 전체 바이트를 대신해야 합니다 . 예를 들어 `198.95.251.*` 는 사용 가능하지만 `198.95.251.3*` 는 사용할 수 없습니다 . IP 주소에 * 가 있는 경우 반드시 문자의 가장 오른쪽에 있어야 합니다 . 예를 들어 `198.*` 는 사용 가능하지만 `198.*.251.30` 은 사용할 수 없습니다 .

프로그램에 대한 액세스 제한

프로그램에 대한 액세스는 오직 Administration Server 에 의하여 제한될 수 있습니다 . 프로그램에 대한 액세스를 제한하면 오직 지정된 사용자만 Server Manager 페이지를 볼 수 있도록 하고 , 이 사용자가 해당 서버를 구성할 수 있는지 여부를 결정할 수 있습니다 . 예를 들어 일부 관리자가 Administration Server 의 Users 및 Groups 섹션을 구성할 수 있도록 허용하면서 Global Settings 섹션은 액세스하지 못하도록 할 수 있습니다 .

서로 다른 사용자가 서로 다른 기능 영역에 액세스하도록 구성할 수 있습니다 . 사용자를 몇 가지 선택된 기능 영역에 액세스하도록 지정한 경우 해당 사용자는 로그인한 다음 지정된 기능 영역의 Administration Server 페이지만 사용할 수 있습니다 .

사용자 인터페이스의 Access Control Rules For 페이지에서 Programs 에 대해 다음 요소가 표시됩니다 .

- **All Programs** 는 모든 프로그램에 대한 액세스를 허용 또는 거부합니다 . 기본적으로 관리자는 서버의 모든 프로그램에 액세스할 수 있습니다 .
- **Only The Following** 은 사용자가 액세스할 수 있는 프로그램을 지정할 수 있도록 합니다 .
  - **Program Groups** 는 예를 들면 Preferences 및 Global Settings 등과 같은 Administration Server 의 탭들을 말하며 해당 페이지에 대한 액세스를 나타냅니다 . 관리자가 Administration Server 에 액세스하면 서버는 아이디 , 호스트 및 IP 주소를 사용하여 관리자가 볼 수 있는 페이지를 결정합니다 .
  - **Program Items** 는 필드에 페이지 이름을 입력하여 프로그램 내의 특정 페이지에 대한 액세스를 제어할 수 있도록 합니다 .

액세스 권한 설정

액세스 권한은 오직 Server Manager 가 서버 인스턴스에 대해 설정합니다 . 액세스 권한은 서버의 파일 및 디렉토리에 대한 액세스를 제한합니다 . 액세스 권한의 허용 또는 거부와 함께 부분적인 액세스 권한을 허용 또는 거부하는 규칙을 지정할 수 있습니다 . 예를 들어 사용자에게 파일에 대한 읽기 전용 액세스 권한을 부여하여 정보를 볼 수 있으나 파일을 변경할 수 없도록 할 수 있습니다 .

사용자 인터페이스의 Access Control Rules For 페이지에서 Rights 에 대해 다음 요소가 표시됩니다 .

- **All Access Rights** 는 기본값으로 모든 권한을 허용 또는 거부합니다 .

- **Only The following Rights** 에서는 허용 또는 거부할 권한을 결합하여 선택할 수 있습니다 .
  - **Read** 는 HTTP 메소드인 GET, HEAD, POST 및 INDEX 를 포함하여 파일을 볼 수 있도록 허용합니다 .
  - **Write** 는 HTTP 메소드 PUT, DELETE, MKDIR, RMDIR 및 MOVE 를 포함하여 파일을 변경하거나 삭제할 수 있도록 허용합니다 . 파일을 삭제하려면 사용자에게 반드시 쓰기 및 삭제 권한이 있어야 합니다 .
  - **Execute** 는 사용자가 CGI 프로그램 , Java 애플릿 및 에이전트 등의 서버측 응용 프로그램을 실행할 수 있도록 허용합니다 .
  - **Delete** 는 쓰기 권한을 가진 사용자가 파일 또는 디렉토리를 삭제할 수 있도록 허용합니다 .
  - **List** 는 `index.html` 파일을 포함하지 않은 디렉토리의 파일 목록에 액세스할 수 있도록 허용합니다 .
  - **Info** 는 사용자가 `http_head` 등의 URI 에 대한 정보를 받을 수 있도록 허용합니다 .

사용자 정의 표현식 작성

ACL 용 사용자 정의 표현식을 입력할 수 있습니다 . ACL 파일의 구문과 구조에 익숙한 경우에만 이 옵션을 선택하십시오 . 몇 가지 기능은 ACL 파일을 편집하거나 사용자 정의 표현식을 만든 경우에만 사용할 수 있습니다 . 예를 들어 하루 중 시간 , 주중 요일 또는 이 둘 모두를 기준으로 서버에 대한 액세스를 제한할 수 있습니다 .

하루 중 시간 및 주중 요일을 기준으로 액세스를 제한하는 사용자 정의 표현식의 예는 다음과 같습니다 . 이 예에서는 LDAP 디렉토리에 두 개의 그룹이 있는 것으로 가정합니다 . Regular 그룹은 월요일에서 금요일까지 8:00am 에서 5:00pm 사이에 액세스할 수 있습니다 . Critical 그룹은 항상 액세스할 수 있습니다 .

```
allow (read)
{
    (group=regular and dayofweek=?on,tue,wed,thu,fri?;
    (group=regular and (timeofday>=0800 and timeofday<=1700));
    (group=critical)
}
```

유효한 구문과 ACL 파일에 대한 자세한 내용은 부록 A, 373 페이지의 "ACL 파일 구문 " 을 참조하십시오 .

엑세스 제어 사용 중지

Access Control Rules For 페이지에서 Access Control Is On 옵션의 선택을 취소하면 ACL 의 기록을 삭제할 것인지 묻는 메시지가 표시됩니다 . OK 를 누르면 해당 리소스에 대한 ACL 항목이 ACL 파일에서 삭제됩니다 .

ACL 을 사용 중지하려면 `generated-proxy-`*`serverid`*`.acl` 파일에서 각 줄의 앞에 # 기호를 삽입하여 이를 주석으로 만듭니다 .

Administration Server 에서 특정 서버 인스턴스에 대한 액세스 제어를 만들어 사용하며 기타 서버에 대해는 사용하지 않도록 (기본값) 할 수 있습니다 . 예를 들어 Administration Server 의 Server Manager 페이지에서 모든 액세스를 거부할 수 있습니다 . 기타 서버는 기본적으로 분산 관리를 사용하고 액세스 제어는 사용하지 않으므로 관리자는 다른 서버에 액세스하여 구성할 수 있으나 Administration Server 는 구성할 수 없습니다 .

액세스가 거부된 경우의 응답

Proxy Server 는 액세스가 거부된 경우 기본 메시지를 제공하며 원하는 경우 이 기본 응답 메시지를 사용자 정의할 수 있습니다 . 또한 각 액세스 제어 개체마다 서로 다른 메시지를 만들 수 있습니다 .

Administration Server 의 경우 사용자에게 표시되는 기본 메시지는 *server_root*`/httpacl/admin-denymsg.html` 에 있는 Permission Denied 입니다 .

Access Denied 메시지를 변경하려면 다음을 수행합니다 .

1. Access Control Rules For 페이지에서 Response When Denied 링크를 누릅니다 .
2. 원하는 응답을 선택하고 필요한 경우 추가 정보를 입력 (사용자에게 리디렉션된 응답에 대한 액세스 권한이 있는지 확인해야 함) 한 다음 Update 를 누릅니다 .
3. Submit 를 눌러 변경 사항을 저장하거나 Revert 를 눌러 페이지의 요소를 변경하기 전에 포함된 값으로 재설정합니다 .

서버의 영역에 대한 액세스 제한

이 절에서는 서버 및 해당 컨텐트에 대해 일반적으로 사용되는 제한에 대해 설명합니다 . 각 절차에 대한 단계에서는 필요한 작업을 세부적으로 설명합니다 . 그러나 " 서버 인스턴스용 액세스 제어 설정 " (161 페이지) 에 설명된 단계를 반드시 완료해야 합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

전체 서버에 대한 액세스 제한

하위 도메인의 컴퓨터에서 서버에 액세스하는 그룹의 사용자에게 액세스를 허용해야 하는 경우가 있습니다 . 예를 들어 회사 부서의 서버의 경우 사용자가 오직 네트워크의 특정 도메인의 컴퓨터에서 액세스하도록 할 수 있습니다 .

전체 서버에 대한 액세스를 제한하려면 다음을 수행합니다 .
서버 인스턴스에 대한 액세스 제어를 설정하는 방법을 사용하여 (" 서버 인스턴스용 액세스 제어 설정 " (161 페이지) 참조) 다음과 같이 설정합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스합니다 .
2. Preferences 탭에서 Administer Access Control 링크를 누릅니다 .
3. 드롭다운 목록에서 전체 서버를 선택하고 Select 를 누른 다음 해당 Edit 버튼을 누릅니다 . Access Control Rules For 페이지가 표시됩니다 .
4. 모두의 액세스를 거부할 새 규칙을 추가합니다 .
5. 특정 그룹의 액세스를 허용하는 다른 규칙을 새로 추가합니다 .
6. From Host 를 사용하여 제한하려는 호스트 이름과 IP 주소를 지정합니다 .
7. Submit 를 눌러 변경 사항을 저장합니다 .

디렉토리 (경로) 에 대한 액세스 제한

그룹의 사용자가 디렉토리, 또는 그룹의 소유자가 제어하는 하위 디렉토리 및 파일에서 응용 프로그램을 읽거나 실행하도록 허용할 수 있습니다 . 예를 들어 프로젝트 관리자는 프로젝트 팀이 검토할 수 있도록 상태 정보를 업데이트할 수 있습니다 .

디렉토리에 대한 액세스를 제한하려면 다음을 수행합니다 .
서버 인스턴스에 대한 액세스 제어를 설정하는 방법을 사용하여 (" 서버 인스턴스용 액세스 제어 설정 " (161 페이지) 참조) 다음과 같이 설정합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스합니다 .
2. Preferences 탭에서 Administer Access Control 링크를 누릅니다 .
3. 드롭다운 목록에서 원하는 리소스를 선택하고 Edit 를 누릅니다 .
4. 새 규칙을 만들고 기본값을 유지하여 기타 위치로부터의 사용자 액세스를 거부합니다 .
5. 특정 그룹의 사용자에게 오직 읽기 및 실행 권한만 허용하는 새 규칙을 만듭니다 .
6. 특정 사용자에게 모든 권한을 허용하는 세 번째 새 규칙을 만듭니다 .
7. 마지막 두 규칙에 대해 Continue 선택을 취소합니다 .
8. Submit 를 눌러 변경 사항을 저장합니다 .

파일 유형에 대한 액세스 제한

파일 유형에 대한 액세스를 제한할 수 있습니다 . 예를 들어 오직 지정된 사용자만 서버에서 실행되는 프로그램을 만들 수 있도록 허용할 수 있습니다 . 모든 사람이 프로그램을 실행할 수 있으나 오직 그룹의 지정된 사용자만 프로그램을 만들거나 삭제할 수 있습니다 .

파일 유형에 대한 액세스를 제한하려면 다음을 수행합니다 .
서버 인스턴스에 대한 액세스 제어를 설정하는 방법을 사용하여 (" 서버 인스턴스용 액세스 제어 설정 " (161 페이지) 참조) 다음과 같이 설정합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스합니다 .
2. Preferences 탭에서 Administer Access Control 링크를 누릅니다 .
3. Select A Resource 섹션에서 Regular Expression 을 누르고 정규식을 지정합니다 . 예 : `*.cgi`.
4. Edit 를 누릅니다 .

5. 모든 사용자에게 읽기 액세스를 허용하는 새 규칙을 만듭니다 .
6. 오직 지정된 그룹에게 쓰기 및 삭제 액세스를 허용하는 다른 규칙을 만듭니다 .
7. Submit 를 눌러 변경 사항을 저장합니다 .

파일 유형 제한의 경우 두 Continue 확인란을 모두 선택한 상태로 둡니다 . 파일에 대한 요청이 수신되면 서버는 우선 해당 파일 유형에 대한 ACL 을 확인합니다 .

Pathcheck 기능이 obj.conf 에 만들어지며 , 여기에는 파일 또는 디렉토리용 와일드카드 패턴이 포함될 수 있습니다 . ACL 파일의 항목은 다음과 같이 표시됩니다 .

```
acl"*.cgi";
```

하루 중 시간을 기준으로 액세스 제한

지정된 시간 또는 일 동안 서버에 대한 쓰기 및 삭제 액세스를 제한할 수 있습니다 .

하루 중 시간을 기준으로 액세스를 제한하려면 다음을 수행합니다 .
서버 인스턴스에 대한 액세스 제어를 설정하는 방법을 사용하여 (" 서버 인스턴스용 액세스 제어 설정 " (161 페이지) 참조) 다음과 같이 설정합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스합니다 .
2. Preferences 탭에서 Administer Access Control 링크를 누릅니다 .
3. Select A Resource 섹션의 드롭다운 목록에서 전체 서버를 선택하고 Edit 를 누릅니다 .
4. 모든 사용자에게 읽기 및 실행 권한을 허용하는 새 규칙을 만듭니다 . 이렇게 하면 사용자가 파일이나 디렉토리를 추가 , 업데이트 또는 삭제하려 할 때 이 규칙이 적용되지 않으며 서버는 일치하는 다른 규칙을 검색합니다 .
5. 모든 사용자의 쓰기 및 삭제 권한을 거부하는 다른 규칙을 만듭니다 .
6. X 링크를 눌러 사용자 정의 표현식을 만듭니다 .
7. 허용할 주중 요일과 하루 중 시간을 입력합니다 . 예 :

```
user = "anyone" and
dayofweek = "sat,sun" or
(timeofday >= 1800 and
timeofday <= 600)
```

8. Submit 를 눌러 변경 사항을 저장합니다 . 사용자 정의 표현식에 오류가 있는 경우 오류 메시지가 생성됩니다 . 오류를 수정하고 다시 제출하십시오 .

보안을 기준으로 액세스 제한

동일한 서버 인스턴스에 대해 SSL 청취 소켓과 SSL 이 아닌 청취 소켓을 구성할 수 있습니다 . 보안을 기준으로 액세스를 제한하면 오직 보안 채널을 통하여 전송되어야 하는 리소스를 보호할 수 잇습니다 .

보안을 기준으로 액세스를 제한하려면 다음을 수행합니다 .
서버 인스턴스에 대한 액세스 제어를 설정하는 방법을 사용하여 (" 서버 인스턴스용 액세스 제어 설정 " (161 페이지) 참조) 다음과 같이 설정합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스합니다 .
2. Preferences 탭에서 Administer Access Control 링크를 누릅니다 .
3. Select A Resource 섹션의 드롭다운 목록에서 전체 서버를 선택하고 Edit 를 누릅니다 .
4. 모든 사용자에게 읽기 및 실행 권한을 허용하는 새 규칙을 만듭니다 . 이렇게 하면 사용자가 파일이나 디렉토리를 추가 , 업데이트 또는 삭제하려 할 때 이 규칙이 적용되지 않으며 서버는 일치하는 다른 규칙을 검색합니다 .
5. 모든 사용자의 쓰기 및 삭제 권한을 거부하는 다른 규칙을 만듭니다 .
6. X 링크를 눌러 사용자 정의 표현식을 만듭니다 .
7. `ssl="on"` 을 입력합니다 . 예 :

```
user = "anyone" and ssl="on"
```

8. Submit 를 눌러 변경 사항을 저장합니다 . 사용자 정의 표현식에 오류가 있는 경우 오류 메시지가 생성됩니다 . 오류를 수정하고 다시 제출하십시오 .

리소스에 대한 액세스 보안

이 절에서는 분산 관리를 사용하도록 설정한 후 Proxy Server 의 액세스 제어 보안을 위해 수행해야 하는 추가 작업에 대해 설명합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 서버 인스턴스에 대한 액세스 보안
- IP 기반 액세스 제어 사용

서버 인스턴스에 대한 액세스 보안

Proxy Server 가 서버 인스턴스에 대한 액세스를 제어하도록 구성하려면 *server_root*/httpacl/*.proxy-admserv.acl 파일을 편집하여 액세스 제어 권한을 부여할 사용자를 지정합니다 . 예 :

```
acl "proxy-server_instance";
authenticate (user,group) {
database = "default";
method = "basic";
};
deny absolute (all) user != "UserA";
```

IP 기반 액세스 제어 사용

ip 속성을 참조하는 액세스 제어 항목이 Administration Server 와 관련된 ACL 파일 (gen*.proxy-admserv.acl) 에 있는 경우 단계 1 과 단계 2 를 완료합니다 .

ip 속성을 참조하는 액세스 제어 항목이 서버 인스턴스에 관련된 ACL 파일 안에 있는 경우 해당 ACL 에 대해 단계 1 만 완료합니다 .

IP 기반 액세스 제어를 사용하도록 설정하려면 다음을 수행합니다 .

1. 아래와 같이 *server_root*/httpacl/gen*.proxy-admserv.acl 파일을 편집하여 user 및 group 에 추가하여 인증 목록에 ip 를 추가합니다 .

   ```
   acl "https-admserv";
   authenticate (user,group,ip) {
   database = "default";
   method = "basic";
   };
   ```

2. 다음 액세스 제어 항목을 추가합니다 .

```
deny absolute (all) ip !="ip_for_which_access_is_allowed";
```

예 :

```
acl "https-admserv";
authenticate (user,group,ip) {
database = "default";
method = "basic";
};
deny absolute (all) ip !="205.217.243.119";
```

파일 기반 인증용 ACL 생성

Proxy Server 는 파일 기반 인증 데이터베이스를 사용할 수 있으며 , 이 데이터베이스에서는 텍스트 형식으로 보통 파일에 사용자 및 그룹 정보를 저장합니다 . ACL 프레임워크는 파일 인증 데이터베이스와 함께 작동하도록 디자인되었습니다 .

참고 Proxy Server 는 동적 보통 파일을 지원하지 않습니다 . 보통 파일 데이터베이스는 서버가 시작할 때 로드됩니다 . 파일이 변경되는 경우 오직 서버가 재시작되어야 적용됩니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 파일 인증 기반 디렉토리 서비스용 ACL 생성
- Digest 인증 기반 디렉토리 서비스용 ACL 생성

ACL 항목은 `database` 키워드를 사용하여 사용자 데이터베이스를 참조할 수 있습니다 . 예 :

```
acl "default";
    authenticate (user) {
...
    database="myfile";
...
};
```

server_root`/userdb/dbswitch.conf` 파일은 파일 인증 데이터베이스 및 해당 구성을 정의하는 항목을 포함하고 있습니다 . 예 :

```
directory myfiledb file
myfiledb:syntax keyfile
myfiledb:keyfile /path/to/config/keyfile
```

파일 인증 데이터베이스에서 지원하는 매개 변수는 다음 표의 목록을 참조하십시오 .

표 8-2 파일 인증 데이터베이스가 지원하는 매개 변수

매개 변수	설명
구문	(선택 사항) 값은 `keyfile` 또는 `digest` 입니다 . 지정하지 않는 경우 기본값은 `keyfile` 입니다 .
`keyfile`	(`syntax=keyfile` 인 경우 필수) 사용자 데이터가 있는 파일 경로입니다
`digestfile`	(`syntax=digest` 인 경우 필수) Digest 인증용 사용자 데이터가 있는 파일 경로

주의 파일 인증 데이터베이스 파일에서 줄의 최대 길이는 255 입니다 . 이 한계를 초과하는 줄이 있는 경우 서버는 시작할 수 없으며 로그 파일에 오류가 기록됩니다 .

참고 파일 기반 인증 데이터베이스를 사용하여 ACL 을 설정하기 전에 파일 기반 인증 디렉토리 서비스가 이미 구성되어 있는지 확인하십시오. 자세한 내용은 " 디렉토리 서비스 구성 " (47 페이지) 을 참조하십시오 .

파일 인증 기반 디렉토리 서비스용 ACL 생성

파일 인증 기반 디렉토리 서비스용 ACL 을 생성하려면 다음을 수행합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스합니다 .
2. Preferences 탭에서 Administer Access Control 링크를 누릅니다 .
3. 드롭다운 목록에서 ACL 파일을 선택하고 Edit 를 누릅니다 .
4. Access Control Rules For 페이지에서 편집하려는 ACL 항목에 대한 Users/Groups 링크를 누릅니다 . User/Group 페이지가 아래 창에 표시됩니다 .
5. Authentication Database 아래의 드롭다운 목록에서 키 파일 데이터베이스를 지정합니다 .
6. Update 를 누른 다음 Submit 를 눌러 변경 사항을 저장합니다 .

키 파일 기반 파일 인증 데이터베이스에 대한 ACL 을 설정하는 경우 dbswitch.conf 파일이 아래의 예제와 같은 ACL 항목을 포함하여 업데이트됩니다 .

```
version 3.0;
acl "default";
authenticate (user) {
     prompt = "Sun Java System Proxy Server 4.0";
     database = "mykeyfile";
     method = "basic";
};
deny (all) user = "anyone";
allow (all) user = "all";
```

Digest 인증 기반 디렉토리 서비스용 ACL 생성

파일 인증 데이터베이스는 또한 RFC 2617 을 통한 Digest 인증을 사용하기에 적합한 파일 형식을 지원합니다 . 비밀 번호와 영역 기반의 해시가 저장됩니다 . 보통 텍스트 비밀 번호는 보관되지 않습니다 .

Digest 인증 기반 디렉토리 서비스용 ACL 을 생성하려면 다음을 수행합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스합니다 .
2. Preferences 탭에서 Administer Access Control 링크를 누릅니다 .
3. 드롭다운 목록에서 ACL 파일을 선택하고 Edit 를 누릅니다 .
4. Access Control Rules For 페이지에서 편집하려는 ACL 에 대한 Users/Groups 링크를 누릅니다 . User/Group 페이지가 아래 창에 표시됩니다 .
5. Authentication Database 아래의 드롭다운 목록에서 다이제스트 데이터베이스를 지정합니다 .
6. Update 를 누른 다음 Submit 를 눌러 변경 사항을 저장합니다 .

Digest 인증 기반 파일 인증 데이터베이스에 대한 ACL 을 설정하는 경우 dbswitch.conf 파일이 아래의 예제와 같은 ACL 항목을 포함하여 업데이트됩니다 .

```
version 3.0;
acl "default";
authenticate (user) {
     prompt = "filerealm";
     database = "mydigestfile";
     method = "digest";
};
```

```
deny (all) user = "anyone";
allow (all) user = "all";
```

9 장

로그 파일 사용

다양한 방법으로 서버의 작동을 모니터할 수 있습니다 . 이 장에서는 로그 파일을 기록하고 확인하여 서버를 모니터하는 방법에 대해 설명합니다 . 내장 성능 모니터 서비스 또는 SNMP 에 대한 내용은 제 10 장 , 205 페이지의 " 서버 모니터 " 를 참조하십시오 .

이 장은 다음 내용으로 구성되어 있습니다 .

로그 파일 설명

서버 로그 파일은 서버의 작동을 기록합니다 . 이 로그를 서버를 모니터와 문제 해결에 사용할 수 있습니다 . 서버에 발생한 모든 오류 목록은 서버 루트 디렉토리의 `proxy-server_name/logs/errors` 에 있는 오류 로그 파일에 있습니다 . 액세스 로그는 서버 루트 디렉토리의 `proxy-server_name/logs/access` 에 있으며 , 서버로의 요청과 서버로부터의 응답에 대한 정보가 기록됩니다 . 프록시 서버 `access` 로그 파일에 기록된 정보를 구성할 수 있습니다 . 서버 통계를 생성하려면 로그 분석기를 사용합니다 . 서버 오류 및 액세스 로그를 백업하려면 해당 파일을 보관합니다 .

참고	운영 체제의 한계로 인하여 Linux 의 경우 프록시 서버는 2GB 를 초과하는 로그 파일을 사용할 수 없습니다 . 최대 파일 크기가 초과되면 기록이 중단됩니다 .

UNIX 및 Windows 플랫폼에서의 로깅

여기에서는 로그 파일이 만들어지는 방법에 대해 설명합니다 . 또한 다음 항목에 대해 설명합니다 .

- 기본 오류 로깅
- syslog 를 사용하여 로깅
- Windows eventlog 를 사용하여 로깅

기본 오류 로깅

UNIX 및 Windows 플랫폼 모두의 경우 Administration Server 의 로그는 관리 `proxy-admserv/logs/` 디렉토리에서 수집됩니다 . 서버 인스턴스로부터의 로그는 `proxy-server_name/logs/` 디렉토리에 수집됩니다 .

전체 서버용 기본 로그 수준을 설정할 수 있습니다 . `stdout` 및 `stderr` 을 서버의 이벤트 로그로 재지정할 수 있으며 로그 출력을 운영 체제의 시스템 로그로 지정할 수 있습니다 . 또한 stdout 및 stderr 내용을 서버의 이벤트 로그로 지정할 수 있습니다 . 기본적으로 로그 메시지는 지정된 서버 로그 파일뿐 아니라 stderr 로 또한 전송됩니다 .

syslog 를 사용하여 로깅

중앙 집중식 로깅이 필요한 안정된 운영 환경의 경우 syslog 를 사용하는 것이 더 좋습니다 . 진단 및 디버깅용으로 로그 출력이 자주 필요한 환경의 경우 개별 서버 인스턴스 로그가 더 관리하기 쉽습니다 .

참고

- 하나의 파일에 기록되는 모든 서버 인스턴스 및 관리 서버용 데이터는 읽고 디버깅하기 어려운 것이 될 수 있습니다 . 오직 문제 없이 실행되는 응용 프로그램에 대해 syslog 마스터 로그 파일을 사용하는 것이 좋습니다 .
- 기록된 메시지는 Solaris 데몬 응용 프로그램으로부터의 모든 기타 로그와 혼합됩니다 .

syslogd 및 시스템 로드 데몬과 함께 syslog 로그 파일을 사용하면 `syslog.conf` 파일을 다음과 같이 구성할 수 있습니다 .

- 적절한 시스템 로그로 메시지 기록
- 시스템 콘솔에 메시지 표시
- 기록된 메시지를 전달하여 모든 사용자 목록을 표시하거나 네트워크를 통하여 다른 호스트의 다른 syslog 로 기록된 메시지 전달

syslog 로의 로깅은 프록시 서버로부터의 로그를 의미하며 기타 데몬 응용 프로그램이 동일한 파일에 수집되므로 , 기록된 메시지는 다음 정보를 포함하여 특정 서버 인스턴스로부터의 프록시 서버 특정 메시지를 구분합니다 .

- Unique message ID
- Timestamp
- Instancename
- Program name(proxyd 또는 proxyd-wdog)
- Process ID(proxyd 프로세스의 PID)
- Thread ID(선택)
- Server ID

LOG 요소는 `server.xml` 파일에서 관리 서버와 서버 인스턴스 모두에 대해 구성할 수 있습니다 .

UNIX 운영 체제에서 사용되는 syslog 로깅에 대한 자세한 내용은 단말기 프롬프트에서 다음의 man 명령을 사용하십시오 .

```
man syslog
man syslogd
man syslog.conf
```

Windows eventlog 를 사용하여 로깅

Windows 운영 체제에서 사용하는 이벤트 로그 기법에 대한 자세한 내용은 Windows 도움말에서 Event Logging 키워드로 검색하십시오 .

로그 수준

Proxy Server 의 로그 수준과 메시지는 중요도의 순서에 따라 다음 표에 정의된 것과 같습니다 .

표 9-1 로그 수준

로그 수준	설명
finest finer fine	메시지는 디버그 메시지의 다변화 수준을 표시합니다 . finest 의 경우 다변화가 최대입니다 .
info	원래 정보를 제공하는 메시지이며 , 보통 서버 구성 또는 서버 상태에 관련된 메시지입니다 . 즉각적인 조치가 필요한 오류를 표시하는 메시지는 아닙니다 .
warning	경고를 표시하는 메시지입니다 . 이 메지시에는 예외가 포함될 수 있습니다 .
failure	정상의 응용 프로그램 실행을 방해할 수 있는 중요한 이상을 표시하는 메시지입니다 .
config	다양한 정적 구성 정보에 관련된 메시지로 특정 구성에 관련된 문제를 해결하는데 도움이 됩니다 .
security	보안 문제를 표시하는 메시지입니다 .
catastrophe	중요한 오류를 표시하는 메시지입니다 .

로그 파일 보관

액세스 및 오류 로그 파일이 자동 보관되도록 설정할 수 있습니다 . 특정 시간이나 지정된 시간이 경과하면 로그가 교체됩니다 . Proxy Server 는 이전 로그 파일을 저장하고 파일이 저장된 일자 및 시간이 포함된 이름을 파일에 지정합니다 .

예를 들어 액세스 로그 파일이 매시간 교체되도록 설정하면 Proxy Server 는 파일을 "access.200505160000" 이라는 이름으로 저장합니다 . 여기에서 로그 파일 이름 , 년 , 월 , 일 및 24 시간 형식 시간은 단일 문자열로 합쳐집니다 . 로그 보관 파일의 정확한 형식은 설정한 로그 교체 유형에 따라 달라집니다 .

Proxy Server 는 파일 보관용으로 두 가지의 로그 교체 유형을 제공합니다 . 바로 내부 데몬 로그 교체 및 Cron 기반 로그 교체입니다 .

내부 데몬 로그 교체

이러한 형태의 로그 교체는 HTTP 데몬에서 수행되며 오직 시작할 때 구성될 수 있습니다 . 내부 데몬 로그 교체를 사용하면 서버가 서버 재시작 없이 내부적으로 로그를 교체할 수 있습니다 . 이 방법으로 교체한 로그는 다음의 형식으로 저장됩니다 .

```
access.<YYYY><MM><DD><HHMM>
```

```
errors.<YYYY><MM><DD><HHMM>
```

로그 파일을 교체하고 새 로그 파일을 시작할 기준으로 사용할 시간을 지정할 수 있습니다 . 예를 들어 교체 시작 시간이 오전 12:00 이고 교체 간격이 1440 분 (하루) 이면 , 현재 시간에 상관없이 변경 사항을 저장 및 적용할 때 새 로그 파일이 만들어집니다 . 로그 파일은 매일 오전 12:00 에 교체되며 액세스 로그 파일은 오전 12:00 으로 스탬프되고 access.200505172400 으로 저장됩니다 . 마찬가지로 간격을 240 분 (4 시간) 으로 설정하고 간격이 오전 12:00 에 시작하면 액세스 로그 파일에는 오전 12:00 에서 오전 4:00 까지 , 오전 4:00 에서 오전 8:00 까지 등의 순서로 정보를 수집됩니다 .

로그 교체를 사용하는 경우 로그 교체는 서버가 시작할 때 시작됩니다 . 첫 로그 파일은 현재 시간부터 다음 교체 시간까지 정보를 수집합니다 . 앞의 예에서 시작 시간을 오전 12:00 으로 , 교체 간격을 240 분으로 설정하며 현재 시간이 오전 6:00 이라면 , 교체의 첫 번째 로그 파일에는 오전 6:00 에서 오전 8:00 까지 수집된 정보가 포함되며 다음 로그 파일에는 오전 8:00 에서 오후 12:00(정오) 까지의 정보가 포함됩니다 .

스케줄 기반 로그 교체

이러한 유형의 로그 교체는 *server_root*/proxy-*server_name*/config/ 디렉토리의 server.xml 파일에 저장된 날짜 및 시간에 따라 수행됩니다. 이 방법을 사용하면 로그 파일을 즉시 보관하거나 특정 일자의 특정 시간에 서버가 로그 파일을 보관하도록 할 수 있습니다 . 서버의 스케줄러 구성 옵션은 server_root/proxy-*server_name*/config/ 디렉토리의 server.xml 에 저장됩니다 . 스케줄 기반 방법으로 교체된 로그는 다음의 형식으로 저장됩니다 .

<*original_filename*>.<YYYY><MM><DD><HHMM>

예를 들어 access 가 오후 4:30 에 교체되면 access.200505171630 이 됩니다 .

로그 교체는 서버가 시작할 때 초기화됩니다 . 교체를 사용하는 경우 Proxy Server 는 시간 스탬프 액세스 로그 파일을 만들고 서버가 시작할 때 교체가 시작됩니다 .

교체가 시작되면 Proxy Server 는 액세스 또는 오류 로그 파일에 기록해야 할 요청 또는 오류가 있는 경우 , 새로운 시간 스탬프 로그 파일을 만들며 , 또한 이 작업은 미리 설정된 " 다음 교체 시간 " 이 경과하면 수행됩니다 .

참고 로그 분석기를 실행하기 전에 서버 로그를 보관해야 합니다 .

로그 파일을 보관하고 내부 데몬 방법 또는 스케줄 기반 방법을 사용할 것인지 지정하려면 Server Manager 의 Archive Log 페이지를 사용합니다 .

액세스 로그 기본 설정

설치할 때 access 라는 이름의 액세스 로그 파일이 해당 서버용으로 만들어집니다 . 액세스를 기록할 것인지의 여부 , 기록에 사용할 형식 및 리소스에 액세스할 때 서버가 클라이언트의 도메인 이름을 조회할 것인지의 여부를 지정하여 모든 리소스에 대한 액세스 로깅을 사용자 정의할 수 있습니다 .

Server Manager 의 Set Access Log Preferences 페이지를 사용하거나 obj.conf 파일에서 다음 지시문을 직접 구성하여 로깅 기본 설정을 지정할 수 있습니다 . obj.conf 에서 서버는 flex-init 함수를 호출하여 유연한 로깅 시스템을 초기화하며 flex-log 함수를 호출하여 요청에 대한 데이터를 유연한 로그 형식으로 기록합니다 . 요청을 공통 로그 파일 형식으로 기록하려면 서버가 init-clf 를 호출하여 obj.conf 에서 사용되는 Common Log 하위 시스템을 초기화하고 common-log 를 호출하여 요청에 대한 데이터를 공통 로그 형식 (대부분의 HTTP 서버에서 사용) 으로 기록합니다 .

리소스용 액세스 로그가 일단 만들어지면 해당 로그를 보관하거나 해당 리소스용으로 새 액세스 로그 파일을 만들지 않는 한 , 이 로그를 변경할 수 없습니다 .

기존 로그 파일의 형식을 변경하는 경우 우선 기존 로그 파일을 삭제 / 이름 변경하거나 다른 파일 이름을 사용해야 합니다 .

Administration Server 의 액세스 로그 기본 설정을 구성하려면 다음과 같이 합니다 .

1. Administration Server 에 액세스하고 Preferences 탭을 누릅니다 .
2. Set Access Log Preferences 링크를 누릅니다 . Set Access Log Preferences 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 클릭하고 정규식을 입력한 다음 OK 를 누릅니다 .
4. 클라이언트 액세스를 로그 파일에 기록할지 지정합니다 . 이 요소를 사용하려면 DNS(Domain Name Service) 를 사용하도록 설정해야 합니다 .
5. 액세스 로그 파일의 절대 경로를 지정합니다 . 기본적으로 로그 파일은 서버 루트의 logs 디렉토리에 저장됩니다 . 부분적인 경로를 지정하면 서버는 이를 서버 루트의 logs 디렉토리에 대한 상대 경로로 가정합니다 .

   전체 서버를 편집하는 경우 이 필드의 기본값은 `$accesslog` 이며 , 이 변수는 구성 파일에서 해당 서버용 액세스 로그 파일을 나타냅니다 .
6. 서버에 액세스하는 시스템의 도메인 이름 또는 IP 주소를 액세스 로그에 기록할 것인지 선택합니다 .
7. 액세스 로그에 사용할 로그 파일 형식을 선택합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .
   - **Use Common LogFile Format.** 클라이언트의 호스트 이름 , 인증된 사용자 이름 , 요청 일자 및 시간 , HTTP 헤더 , 클라이언트에 반환된 상태 코드 및 클라이언트에 전송된 문서의 내용 길이 등이 포함됩니다 .
   - **Only Log.** 로그할 정보를 선택할 수 있습니다 . 다음의 유연한 로그 형식 항목을 선택할 수 있습니다 .
     - **Client Hostname**. 액세스를 요청하는 클라이언트의 호스트 이름 (또는 DNS 를 사용하지 않는 경우 IP 주소)
     - **Authenticate User Name.** 인증이 필요한 경우 , 인증된 아이디 목록이 액세스 로그에 기록되도록 합니다 .
     - **System Date.** 클라이언트 요청의 일자 및 시간
     - **Full Request.** 클라이언트가 수행한 그대로의 요청
     - **Status.** 서버가 클라이언트에게 반환한 상태 코드

- **Content Length.** 클라이언트에게 송신한 문서의 길이 (바이트 단위)
- **HTTP Header, "referer".** 참조자 (referer) 는 클라이언트가 현재 액세스한 페이지의 상위 페이지에 해당합니다 . 예를 들어 사용자가 텍스트 검색 쿼리의 결과를 보고 있다면 참조자는 사용자가 검색할 텍스트를 입력한 페이지가 됩니다 . 서버는 참조자를 사용하여 역방향 추적 링크를 만듭니다 .
- **HTTP Header, "user-agent".** 사용자 에이전트 정보에는 클라이언트가 사용하는 브라우저의 종류 , 버전 및 실행되는 운영 체제 등이 포함되며 , 이 정보는 클라이언트가 서버로 보내는 HTTP 헤더 정보의 User-agent 필드에서 가져옵니다 .
- **Method.** 사용된 HTTP 요청 메소드 (GET, PUT, POST, 등)
- **URI.** Universal Resource Identifier. 서버에 있는 리소스의 위치입니다 . 예를 들어 `http://www.a.com:8080/special/docs` 의 경우 URI 는 `special/docs` 입니다 .
- **Query String Of The URI.**URI 의 의문 부호 뒤에 이어지는 내용 . `http://www.a.com:8080/special/docs?find_this` 의 경우 URI 의 쿼리 문자열은 `find_this` 입니다 .
- **Protocol.** 사용된 전송 프로토콜 및 버전 .

○ 사용자 정의 형식을 선택하려면 Custom Format 필드에 입력합니다 .

8. OK 를 누릅니다 .
9. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
10. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

Server Instance 의 액세스 로그 기본 설정을 구성하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Server Status 탭을 누릅니다 .
2. Set Access Log Preferences 링크를 누릅니다 . Set Access Log Preferences 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 클릭하고 정규식을 입력한 다음 OK 를 누릅니다 .
4. 클라이언트 액세스를 로그 파일에 기록할지 지정합니다 . 이 요소를 사용하려면 DNS(Domain Name Service) 를 사용하도록 설정해야 합니다 .

5. 액세스 로그 파일의 절대 경로를 지정합니다 . 기본적으로 로그 파일은 서버 루트의 logs 디렉토리에 저장됩니다 . 부분적인 경로를 지정하면 서버는 이를 서버 루트의 logs 디렉토리에 대한 상대 경로로 가정합니다 .

   전체 서버를 편집하는 경우 이 필드의 기본값은 `$accesslog` 이며 , 이 변수는 구성 파일에서 해당 서버용 액세스 로그 파일을 나타냅니다 .

6. 서버에 액세스하는 시스템의 도메인 이름 또는 IP 주소를 액세스 로그에 기록할 것인지 선택합니다 .

7. 로그 파일 형식을 공통 , 확장 , 확장 2, 지정한 정보만 ("Only log" 선택 버튼) 또는 사용자 정의로 선택합니다 . Only log 를 누르면 다음의 유연한 로그 형식 항목을 사용할 수 있습니다 .

8. 액세스 로그에 사용할 로그 파일 형식을 선택합니다 . 서버 액세스 로그는 Common Logfile Format, Extended Logfile Format, Extended2 Logfile 형식 , 유연한 로그 형식 또는 사용자 정의 형식을 사용할 수 있습니다 . Common LogFile Format 은 흔히 지원되는 형식으로 서버에 대한 고정된 양의 정보를 제공합니다 . 유연한 로그 형식을 사용하면 로그할 내용을 선택 (Proxy Server 에서) 할 수 있습니다 . 사용자 정의 형식의 경우 로그할 사항을 조정하는 매개 변수 블록을 사용합니다 .

   - **Use Common LogFile Format.** 클라이언트의 호스트 이름 , 인증된 사용자 이름 , 요청 일자 및 시간 , HTTP 헤더 , 클라이언트에 반환된 상태 코드 및 클라이언트에 전송된 문서의 내용 길이 등이 포함됩니다 .

   - **Use Extended LogFile Format.** 일반적인 로그 파일 형식의 모든 필드를 포함하고 , 추가적으로 원격 상태 , 프록시에서 클라이언트로의 내용 길이 , 원격에서 프록시로의 내용 길이 , 프록시에서 원격으로의 내용 길이 , 클라이언트에서 프록시로의 헤더 길이 , 프록시에서 클라이언트로의 헤더 길이 , 프록시에서 원격으로의 헤더 길이 , 원격에서 프록시로의 헤더 길이 및 전송 시간과 같은 필드를 기록합니다 .

   - **Use Extended2 LogFile Format.** 확장형 로그 파일 형식의 모든 필드를 포함하고 , 추가적으로 클라이언트 상태 , 서버 상태 , 원격 상태 , 캐시 완료 상태 , 실제 라우팅 방식과 같은 필드를 기록합니다 .

   - **Only Log.** 로그할 정보를 선택할 수 있습니다 . 다음의 유연한 로그 형식 항목을 선택할 수 있습니다 .

     - **Client Hostname**. 액세스를 요청하는 클라이언트의 호스트 이름 (또는 DNS 를 사용하지 않는 경우 IP 주소).

     - **Authenticate User Name.** 인증이 필요한 경우 , 인증된 사용자 이름 목록이 액세스 로그에 기록되도록 합니다 .

     - **System Date.** 클라이언트 요청의 일자 및 시간

- **Full Request.** 클라이언트가 수행한 그대로의 요청
- **Status.** 서버가 클라이언트에게 반환한 상태 코드
- **Content Length.** 클라이언트에게 송신한 문서의 내용 길이 (바이트 단위)
- **HTTP Header, "referer".** 참조자 (referer) 는 클라이언트가 현재 액세스한 페이지의 상위 페이지에 해당합니다 . 예를 들어 사용자가 텍스트 검색 쿼리의 결과를 보고 있다면 참조자는 사용자가 검색할 텍스트를 입력한 페이지가 됩니다 . 서버는 참조자를 사용하여 역방향 추적 링크를 만듭니다 .
- **HTTP Header, "user-agent".** 사용자 에이전트 정보에는 클라이언트가 사용하는 브라우저의 종류 , 버전 및 실행되는 운영 체제 등이 포함되며 , 이 정보는 클라이언트가 서버로 보내는 HTTP 헤더 정보의 User-agent 필드에서 가져옵니다 .
- **Method.** 사용된 HTTP 요청 메소드 (GET, PUT, POST 등)
- **URI.** Universal Resource Identifier. 서버에 있는 리소스의 위치입니다 . 예를 들어 `http://www.a.com:8080//special/doc` 의 경우 URI 는 `special/docs` 입니다 .
- **Query String Of The URI.**URI 의 의문 부호 뒤에 이어지는 내용 . `http://www.a.com:8080//special/docs?find_this` 의 경우 URI 의 쿼리 문자열은 `find_this` 입니다 .
- **Protocol.** 사용된 전송 프로토콜 및 버전 .
- **Cache finish status.** 이 필드는 캐시 파일이 쓰여졌는지 , 새로 고쳤는지 , 또는 최신인지의 검사를 통해 반환되었는지를 나타냅니다 .
- **Remote Server Finish Status.** 이 필드는 원격 서버로의 요청이 성공적으로 전달되었는지 , Netscape Navigator 에서 클라이언트가 중지 버튼을 눌러 중단되었는지 , 또는 오류 조건에 의해 취소되었는지를 나타냅니다 .
- **Status Code From Server.** 서버에서 반환한 상태 코드
- **Route To Proxy (PROXY, SOCKS, DIRECT).** 리소스를 얻기 위해 사용된 라우팅 방식 . 문서는 직접 , 프록시를 통해 , 또는 SOCKS 서버를 통해 얻어질 수 있습니다 .
- **Transfer Time.** 전송 시간의 길이 (초나 밀리초 단위)
- **Header-length From Server Response.** 서버 응답의 헤더 길이
- **Request Header Size From Proxy To Server.** 프록시에서 서버로의 요청 헤더 길이

- **Response Header Size Sent To Client.** 클라이언트로 송신된 응답 헤더의 길이
- **Request Header Size Received From Client.** 클라이언트로부터 수신된 요청 헤더의 길이
- **Content-length From Proxy To Server Request.** 프록시에서 서버로 보내진 문서의 길이 (바이트 단위).
- **Content-length Received From Client.** 클라이언트로부터 수신된 문서의 길이 (바이트 단위)
- **Content-length From Server Response.** 서버로부터 수신된 문서의 길이 (바이트 단위)
- **Unverified User From Client.** 인증 과정에서 원격 서버에 주어진 사용자 이름

❍ 사용자 정의 형식을 선택하려면 Custom Format 필드에 입력합니다 .

9. 특정 호스트 이름이나 IP 주소로부터의 클라이언트 액세스를 기록하지 않으려면 이를 호스트 이름 및 IP 주소 필드에 입력합니다 . 액세스를 기록하지 않을 호스트의 와일드카드 패턴을 입력할 수 있습니다 . 예를 들어 `*.example.com` 을 입력하면 도메인이 example.com 인 사용자의 액세스는 로그에 기록하지 않습니다 . 호스트 이름 , IP 주소 또는 호스트 이름과 IP 주소 모두에 대한 와일드카드 패턴을 입력할 수 있습니다 .

10. 로그 파일에 형식 문자열을 포함할 것인지 선택합니다 . Proxy Server 의 로그 분석기를 사용할 경우 , 형식 문자열을 포함해야 합니다 . 타사 분석기를 사용할 경우 , 로그 파일에 형식 문자열이 필요하지 않은 경우가 많습니다 .

11. OK 를 누릅니다 .

12. Restart Required 를 누릅니다 . Apply Changes 페이지가 나타납니다 .

13. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

용이한 쿠키 로깅

Proxy Server 에서는 flexlog 기능을 사용하여 특정 쿠키를 쉽게 기록할 수 있습니다 . 구성 파일 `obj.conf` 에서 `flex-log` 하위 시스템을 초기화하는 줄에 "`Req->headers.cookie.cookie_name`" 을 추가합니다 . 이렇게 하면 쿠키 변수가 요청의 헤더에 있는 경우 쿠키 변수 `cookie_name` 의 값을 기록하며 , 쿠키 변수가 없는 경우에는 "-" 을 기록합니다 .

오류 로깅 옵션 설정

Proxy Server 에서 서버의 오류 로그에 기록될 정보를 구성할 수 있습니다 .

오류 로깅 옵션을 설정하려면 다음과 같이 합니다 .

1. Administration Server 에서 오류 로깅 옵션을 설정하려면 Preferences 탭을 선택하고 Set Error Log Preferences 링크를 누릅니다 .

   Server Manager 의 서버 인스턴스에 대한 오류 로깅 옵션을 설정하려면 Server Status 탭을 선택하고 Set Error Log Preferences 링크를 누릅니다 .
2. 서버 메시지를 저장할 파일을 Error Log File Name 필드에 지정합니다 .
3. Log Level 드롭다운 목록에서 오류 로그에 기록될 정보의 양을 지정합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .
4. stdout 출력을 오류 로그로 재지정하려면 Log Stdout 확인란을 선택합니다 .
5. stderr 출력을 오류 로그로 재지정하려면 Log Stderr 확인란을 선택합니다 .
6. 로그 메시지를 콘솔로 재지정하려면 Log To Console 확인란을 선택합니다 .
7. UNIX syslog 서비스 또는 Windows Event Logging 이 로그를 생성하고 관리하게 하려면 Use System Logging 확인란을 선택합니다 .
8. OK 를 누릅니다 .
9. Restart Required 를 누릅니다 . Apply Changes 페이지가 나타납니다 .
10. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

LOG 요소 구성

`server.xml` 파일에서 구성할 수 있는 LOG 요소용 속성은 다음 표와 같습니다 .

표 9-2 LOG 속성

속성	기본 모드	설명
`file`	`errors`	서버에서 보내는 메시지를 저장할 파일을 지정합니다 .
`loglevel`	`info`	다른 요소가 오류 로그에 기록한 메시지의 기본 유형을 제어합니다 . 최고에서 최저까지 허용되는 값은 다음과 같습니다 . `finest`, `fine`, `fine`, `info`, `warning`, `failure`, `config`, `security` 및 `catastrophe`

표 9-2 LOG 속성

속성	기본 모드	설명
logstdout	true	(선택) true 인 경우 stdout 출력을 오류 로그로 보냅니다. 유효한 값은 on, off, yes, no, 1, 0, true, false 입니다 .
logstderr	true	(선택) true 인 경우 stderr 출력을 오류 로그로 보냅니다. 유효한 값은 on, off, yes, no, 1, 0, true, false 입니다 .
logtoconsole	true	(선택 , UNIX 전용) true 인 경우 로그 메시지를 콘솔로 보냅니다 .
createconsole	false	(선택 , Windows 전용) true 인 경우 stderr 출력용 Windows 콘솔을 만듭니다 . 유효한 값은 on, off, yes, no, 1, 0, true, false 입니다 .
usesyslog	false	(선택) true 인 경우 UNIX syslog 서비스 또는 Windows Event Logging 을 사용하여 로그를 생성하고 관리합니다 . 유효한 값은 on, off, yes, no, 1, 0, true, false 입니다 .

액세스 로그 파일 확인

서버의 사용 중인 로그 파일과 보관된 로그 파일을 볼 수 있습니다 .

Administration Server 에서 Administration Server 의 액세스 로그를 보려면 Preferences 탭을 선택한 후 View Access Log 페이지를 누릅니다 .

Server Manager 에서 서버 인스턴스에 대한 액세스 로그를 보려면 Server Status 탭을 선택한 후 View Access Log 링크를 누릅니다 .

다음은 Common Logfile Format 의 액세스 로그 예입니다 . 형식은 Log Preferences 창에서 지정합니다 . 자세한 내용은 " 액세스 로그 기본 설정 " (184 페이지) 을 참조하십시오 .

```
198.18.17.222 - - [20/May/2005:14:15:49 +0530] "GET http://www.example.com/
HTTP/1.1" 504 622
198.18.17.222 - abc [20/May/2005:14:16:09 +0530] "GET
http://www.test.com/report.zip HTTP/1.1" 504 630
```

표 9-3 은 이 예제 액세스 로그 마지막 줄에 대한 설명입니다 .

표 9-3 예제 액세스 로그 파일의 마지막 줄 필드

액세스 로그 필드	예
클라이언트의 호스트 이름 또는 IP 주소	198.18.17.222 (이 경우 프록시 서버의 DNS 조회 설정이 사용 안 함으로 설정되어 있으므로 클라이언트의 IP 주소가 표시됩니다. DNS 조회를 사용하도록 설정된 경우 클라이언트의 호스트 이름이 나타납니다).
IFC 931 정보	- (RFC 931 ID 는 구현되지 않음)
아이디	abc(클라이언트가 인증용으로 입력한 아이디)
요청 일자 / 시간	20/May/2005:14:16:09 +0530
요청	GET
프로토콜	HTTP/1.1
상태 코드	504
전송된 바이트	630

오류 로그 파일 확인

오류 로그 파일에는 로그 파일이 만들어진 후부터 서버에 발생한 오류가 기록되며, 서버의 시작 시간 등 서버에 대한 정보 메시지가 들어 있습니다. 성공하지 못한 사용자 인증 또한 오류 로그에 기록됩니다. 오류 로그를 사용하여 끊어진 URL 경로나 누락된 파일을 찾을 수 있습니다.

Administration Server 의 오류 로그 파일을 보려면 Administration Server 에서 Preferences 탭을 선택한 후 View Error Log 링크를 누릅니다.

서버 인스턴스 오류 로그 파일을 보려면 Server Manager 에서 Server Status 탭을 선택한 후 View Error Log 링크를 누릅니다.

다음은 오류 로그 항목의 세 가지 예제입니다.

```
20/May/2005:14:08:37] info ( 6141): CORE1116: Sun Java System Web Proxy
Server 4.0 B05/10/2005 01:26
20/May/2005:14:08:37] info ( 6142): CORE3274: successful server startup
20/May/2005:14:08:37] security (23246): for host 198.18.148.89 trying to GET
/, deny-service reports: denying service of /
```

로그 분석기 작업

server_root/extras/log_anly 디렉토리에는 Server Manager 사용자 인터페이스를 통하여 실행할 수 있는 로그 분석 도구가 있습니다 . 이 로그 분석기는 오직 공통 로그 형식의 파일만 분석합니다 . 도구의 매개 변수를 설명하는 log_anly 디렉토리 내의 HTML 문서입니다 . server-install/extras/flexanlg 디렉토리에는 유연한 로그 파일 형식용 명령줄 로그 분석기가 있습니다 . 그러나 Server Manager 는 선택한 로그 파일 형식과 상관 없이 기본적으로 유연한 로그 파일 보고 도구를 사용하도록 설정합니다 .

로그 분석기를 사용하여 작동 요약 , 가장 많이 액세스된 URL, 하루 중 서버에 대한 액세스가 가장 많은 시간 , 등의 기본 서버에 대한 통계를 생성합니다 . 로그 분석기는 Proxy Server 나 명령줄에서도 실행할 수 있습니다 .

flexanlg 명령줄 유틸리티를 실행하기 전에 반드시 라이브러리 경로를 설정해야 합니다 . 다양한 플랫폼용 설정은 다음과 같습니다 .

Solaris 및 Linux:

```
LD_LIBRARY_PATH=server_root/bin/proxy/lib:$LD_LIBRARY_PATH
```

AIX:

```
LIBPATH=server_root/bin/proxy/lib:$LIBPATH
```

HP-UX:

```
SHLIB_PATH=server_root/bin/proxy/lib:$SHLIB_PATH
```

Windows:

```
path=server_root\bin\proxy\bin;%path%
```

참고 로그 분석기를 실행하기 전에 서버 로그를 보관해야 합니다 . 서버 로그 보관에 대한 자세한 내용은 " 로그 파일 보관 " (183 페이지) 을 참조하십시오 .

라이브러리 경로를 설정하는 대신 server_root/proxy-server*id* 디렉토리로 변경한 후 명령 프롬프트에 **./start -shell** 을 입력할 수도 있습니다 .

확장 또는 확장 2 로깅 형식을 사용할 경우 로그 분석기는 보고하도록 지정한 정보 외에도 출력 파일에서 여러 보고서를 생성합니다 . 다음 절에서는 이러한 보고서에 대해 설명합니다 .

Transfer Time Distribution Report

전송 시간 배포 보고서는 프록시 서버가 요청을 전송하는 데 걸린 시간을 표시합니다. 이 보고서에서는 정보를 서비스 시간 및 완료된 퍼센트별로 구분하여 표시합니다 . 다음은 전송 시간 배포 보고서의 예입니다 .

By service time category:

```
< 1 sec [644%] ....................................
< 2 sec [33.3%] ...................
< 3 sec [ 2.7%] .
< 4 sec [ 1.7%] .
< 5 sec [ 0.6%]
< 6 sec [ 0.4%]
< 7 sec [ 0.2%]
< 8 sec [ 0.0%]
< 9 sec [ 0.0%]
```

By percentage finished:

```
< 1 sec [64.4%] ....................................
< 2 sec [97.7%] ................................
< 3 sec [100.4%].........................................
```

Status Code Report

상태 코드 보고서는 프록시 서버가 원격 서버로부터 받고 클라이언트로 전송한 상태 코드와 상태 코드 수를 표시합니다 . 또한 이러한 모든 상태 코드에 대한 설명도 제공합니다 . 다음은 상태 코드 보고서의 예입니다 .

Code	-From remote-	-To client-	-Explanation-
200	338 [70.7%]	352 [73.6%]	OK
302	33 [6.9%]	36 [7.5%]	Redirect
304	90 [18.8%]	99 [20.7%]	Not modified
404	3 [0.6%]	3 [0.6%]	Not found
407		5 [1.0%]	Proxy authorization required

Code	-From remote-	-To client-	-Explanation-
500		2 [0.4%]	Internal server error
504		6 [1.3%]	Gateway timeout

Data Flow Report

데이터 흐름 보고서는 클라이언트에서 프록시로 , 프록시에서 클라이언트로 , 프록시에서 원격 서버로 , 원격 서버에서 프록시로 데이터 흐름 (전송된 바이트 수) 을 표시합니다 . 이 보고서에서는 각 시나리오에 대해 전송된 바이트 수를 헤더 및 컨텐츠 형식으로 표시합니다 . 데이터 흐름 보고서에서는 캐시에서 클라이언트로의 데이터 흐름도 표시합니다 . 다음은 데이터 흐름 보고서의 예입니다 .

	Headers	Content	Total
- Client -> Proxy.....	0 MB	0 MB	0 MB
- Proxy -> Client.....	0 MB	2 MB	3 MB
- Proxy -> Remote.....	0 MB	0 MB	0 MB
- Remote -> Proxy.....	0 MB	2 MB	2 MB
Approx:			
- Cache -> Client.....	0 MB	0 MB	0 MB

Requests and Connections Report

요청 및 연결 보고서는 프록시 서버가 클라이언트로부터 받은 요청 수 , 프록시의 원격 서버 연결 시도 수 (초기 검색 , 최신 여부 확인 및 새로 고침) 및 프록시 서버가 캐시된 문서를 사용하여 거부한 원격 연결 수를 표시합니다 . 다음은 요청 및 연결 보고서의 예입니다 .

- Total requests.....	478
- Remote connections.....	439
- Avoided remote connects....	39 [8.2%]

Cache Performance Report

캐시 성능 보고서는 클라이언트 캐시 , 프록시 서버 캐시 및 직접 연결의 성능을 표시합니다 .

Client Cache

참고	클라이언트 캐시 적중은 클라이언트가 문서에서 최신 여부 확인을 수행하고 원격 서버가 클라이언트 문서가 수정되지 않았음을 알려주는 304 메시지를 반환한 경우 발생합니다 . 클라이언트가 시작한 최신 여부 확인은 클라이언트가 캐시에 고유의 문서 사본을 가지고 있음을 표시합니다 .

이 보고서에서는 클라이언트 캐시에 대해 다음을 표시합니다 .

- **client and proxy cache hits:** 프록시 서버 및 클라이언트가 모두 요청한 문서의 사본을 가지고 있으며 원격 서버가 프록시 서버의 사본과 관련한 최신 여부 확인 쿼리를 받은 후 클라이언트의 요청이 프록시의 사본과 관련하여 평가되는 클라이언트 캐시 적중 횟수입니다. 캐시 성능 보고서는 프록시가 서비스한 이 유형의 요청 수와 , 이러한 요청을 서비스하는 데 소요된 평균 시간을 표시합니다 .
- **proxy shortcut no-check:** 프록시 서버와 클라이언트가 모두 요청한 문서의 사본을 가지고 있고 프록시 서버가 클라이언트에게 원격 서버 확인 없이 클라이언트 캐시의 문서가 최신이라고 알리는 클라이언트 캐시 적중입니다 . 캐시 성능 보고서는 프록시가 서비스한 이 유형의 요청 수와 , 이러한 요청을 서비스하는 평균 시간을 표시합니다 .
- **client cache hits only:** 클라이언트만 요청 문서의 캐시된 사본을 가지고 있는 클라이언트 캐시 적중입니다 . 이러한 유형의 요청에서는 프록시 서버가 직접 클라이언트의 If-modified-since GET 헤더를 터널링합니다 . 캐시 성능 보고서는 프록시가 서비스한 이 유형의 요청 수와 , 이러한 요청을 서비스하는 평균 시간을 표시합니다 .
- **total client cache hits:** 총 클라이언트 캐시 적중 횟수와 이러한 요청을 서비스하는 데 걸린 평균 시간입니다 .

Proxy Cache

프록시 캐시 적중은 클라이언트가 프록시 서버로부터 문서를 요청하고 프록시 서버가 이미 캐시에 이 문서를 가지고 있는 경우 발생합니다 . 이 보고서에서는 프록시 서버의 캐시 적중에 대해 다음을 표시합니다 .

- **proxy cache hits with check:** 프록시 서버가 원격 서버에 문서에 대한 최신 여부 확인을 쿼리하는 프록시 캐시 적중입니다 . 캐시 성능 보고서는 프록시가 서비스한 이 유형의 요청 수와 이러한 요청을 서비스하는 데 걸린 평균 시간을 표시합니다 .
- **proxy cache hits without check:** 프록시 서버가 원격 서버에 문서에 대한 최신 여부 확인을 쿼리 *하지 않은* 프록시 캐시 적중입니다 . 캐시 성능 보고서는 프록시가 서비스한 이 유형의 요청 수와 이러한 요청을 서비스하는 데 걸린 평균 시간을 표시합니다 .
- **pure proxy cache hits:** 클라이언트가 요청 문서의 캐시된 사본을 가지고 있지 않은 프록시 캐시 적중입니다 . 캐시 성능 보고서는 프록시가 서비스한 이 유형의 요청 수와 이러한 요청을 서비스하는 데 걸린 평균 시간을 표시합니다 .

Proxy Cache Hits Combined

보고서에서는 통합된 프록시 캐시 적중에 대해 다음을 표시합니다 .

- **total proxy cache hits:** 총 프록시 서버 캐시 적중 횟수와 이러한 요청을 서비스하는 데 걸린 평균 시간입니다 .

Direct Transactions

직접 트랜잭션은 캐시 적중 없이 원격 서버에서 프록시 서버 및 클라이언트로 직접 이동하는 트랜잭션입니다 . 보고서에서는 직접 트랜잭션에 대해 다음을 표시합니다 .

- **retrieved documents:** 원격 서버에서 직접 검색한 문서입니다 . 캐시 성능 보고서는 프록시가 서비스한 이 유형의 요청 수와 이러한 요청을 서비스하는 데 걸린 평균 시간 및 총 트랜잭션의 퍼센트를 표시합니다 .
- **other transactions:** 200 또는 304 가 아닌 상태 코드로 반환된 트랜잭션입니다 . 캐시 성능 보고서는 프록시가 서비스한 이 유형의 요청 수와 이러한 요청을 서비스하는 데 걸린 평균 시간을 표시합니다 .
- **total direct traffic:** 클라이언트에서 원격 서버로 직접 전달된 요청입니다 (실패한 요청 및 성공한 문서 검색 모두 해당). 캐시 성능 보고서는 프록시가 서비스한 이 유형의 요청 수와 이러한 요청을 서비스하는 데 걸린 평균 시간 및 총 트랜잭션의 퍼센트를 표시합니다 .

다음은 캐시 성능 보고서의 예입니다 .

```
CLIENT CACHE:
- Client & proxy cache hits... 86 reqs [18.0%] 0.21 sec/req
- Proxy shortcut no-check........ 13 reqs [ 2.7%] 0.00 sec/req
- Client cache hits only.....
- TOTAL client cache hits.......... 99 reqs [20.7%] 0.18 sec/req
PROXY CACHE:
- Proxy cache hits w/check........ 4 reqs [ 0.8%] 0.50 sec/req
- Proxy cache hits w/o check.. 10 reqs [ 2.1%] 0.00 sec/req
- Pure proxy cache hits...... 14 reqs [ 2.9%] 0.14 sec/req
PROXY CACHE HITS COMBINED:
- TOTAL proxy cache hits....... 113 reqs [23.6%] 0.18 sec/req
DIRECT TRANSACTIONS:
- Retrieved documents..313 reqs [65.5%]  0.90 sec/req 2 MB
- Other transactions.. 52 reqs [10.9%] 7.79 sec/req
- TOTAL direct traffic..365 reqs [76.4%] 1.88 sec/req 2 MB
```

Transfer Time Report

전송 시간 보고서는 프록시 서버가 트랜잭션을 처리하는 데 걸린 시간 정보를 표시합니다 . 이 보고서에서는 다음 범주에 따라 값을 표시합니다 .

average transaction time: 로그에 기록된 모든 전송의 평균 시간입니다 .

average transfer time without caching: 캐시에서 반환되지 않은 트랜잭션에 대한 평균 전송 시간입니다 (원격 서버로부터 200 응답).

average with caching, without errors: 모든 비오류 트랜잭션 (2xx 및 3xx 상태 코드) 에 대한 평균 전송 시간입니다 .

average transfer time improvement: 오류 없이 캐시된 평균 전송 시간을 평균 트랜잭션에서 제한 값입니다 .

다음은 전송 시간 보고서의 예입니다 .

```
- Average transaction time... 1.48 sec/req
- Ave xfer time w/o caching.. 0.90 sec/req
- Ave w/caching, w/o errors.. 0.71 sec/req
- Ave xfer time improvement.. 0.19 sec/req
```

Hourly Activity Report

시간별 작동 보고서는 각 분석 시간에 대해 다음을 표시합니다 .

- 평균 로드
- 원격 서버에 대한 최신 여부 확인이 없는 캐시 적중 횟수
- 문서가 최신이고 문서가 클라이언트 캐시에 있음을 증명하는 원격 서버에 대한 최신 여부 확인이 있는 프록시 서버 캐시의 적중 횟수
- 문서가 최신이고 문서가 클라이언트 캐시에 있지 *않음*을 증명하는 원격 서버에 대한 최신 여부 확인이 있는 프록시 서버 캐시의 적중 횟수
- 문서 일부를 업데이트하게 한 원격 서버에 대한 최신 여부 확인이 있는 프록시 서버 캐시의 적중 횟수
- 요청된 문서의 새로운 사본을 200 상태 코드로 반환한 원격 서버에 대한 최신 여부 확인이 있는 프록시 서버 캐시의 적중 횟수

프록시 서버 캐시에 대한 적중 없이 원격 서버로부터 문서를 직접 검색한 요청 수

Server Manager 에서 로그 분석기를 실행하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Server Status 탭을 누릅니다 .
2. Generate Report 링크를 누릅니다 . Generate Report 페이지가 표시됩니다 .
3. 서버의 이름을 입력합니다 . 이 이름은 생성된 보고서에 나타납니다 .
4. 보고서를 HTML 또는 ASCII 형식으로 표시할지 선택합니다 .
5. 분석할 로그 파일을 선택합니다 .
6. 결과를 파일로 저장하려면 출력 파일 이름을 Output File 필드에 입력합니다 . 필드를 입력하지 않으면 보고서는 화면 상에 출력됩니다 . 로그 파일이 큰 경우 출력을 화면에 표시하는 것이 시간이 많이 걸릴 수 있으므로 결과를 저장하는 것이 좋습니다 .
7. 특정 서버 통계에 대한 총계 생성 여부를 선택합니다 . 다음 총계가 생성됩니다 .
   - **Total Hits.** 액세스 로그를 사용한 후부터 서버가 수신한 총 적중 횟수입니다 .
   - **304 (Not Modified) Status Codes.** 페이지를 반환한 서버가 아니라 요청 문서의 로컬 사본이 사용된 횟수입니다 .
   - **302 (Redirects) Status Codes.** 원래 URL 이 이동하여 서버가 새 URL 로 재지정된 횟수입니다 .
   - **404 (Not Found) Status Codes.** 서버가 요청된 문서를 찾을 수 없거나 클라이언트가 인증된 사용자가 아니므로 문서를 서비스하지 않은 횟수입니다 .
   - **500 (Server Error) Status Codes.** 서버 관련 오류가 발생한 횟수입니다 .
   - **Total Unique URLs.** 액세스 로그를 사용한 후 액세스된 고유 URL 수입니다 .

- **Total Unique Hosts.** 액세스 로그를 사용한 후 서버에 액세스한 고유 호스트의 수입니다 .
- **Total Kilobytes Transferred.** 액세스 로그를 사용한 후부터 서버가 전송한 데이터 양 (KB) 입니다 .

8. 일반 통계 생성 여부를 선택합니다 . 일반 통계를 만드는 방식은 다음 중에서 선택할 수 있습니다 .
9. **Find Top** *숫자* **Seconds Of Log.** 가장 최근 몇 초의 정보를 기반으로 통계를 생성합니다 .
10. **Find Top** *숫자* **Minutes Of Log.** 가장 최근 몇 분의 정보를 기반으로 통계를 생성합니다 .
11. **Find Top** *숫자* **Hours Of Log.** 가장 최신 시간 단위 수의 정보를 기반으로 통계를 생성합니다 .
12. **Find** *숫자* **Users (If Logged).** . 사용자 수의 정보에 기반하여 통계를 생성합니다 .
13. **Find Top** *숫자* **Referers (If Logged).** 참조자 수의 정보에 기반하여 통계를 생성합니다 .
14. **Find Top** *숫자* **User Agents (If Logged)**. 브라우저 종류 , 버전 , 운영 체제 등의 사용자 에이전트에 대한 정보를 기반으로 통계를 생성합니다 .
15. **Find Top** *숫자* **Miscellaneous Logged Items (If Logged).** 사용자 수의 정보에 기반하여 통계를 생성합니다 .
16. 목록 생성 여부를 선택합니다 . 목록을 만들도록 선택한 경우에는 목록을 생성하려는 대상 항목을 다음 목록에서 지정합니다 .
    - **URLs Accessed.** 액세스된 URL 을 표시합니다 .
      - ***숫자*** **Most Commonly Accessed URL.** 가장 많이 액세스된 URL 또는 지정된 회수 이상 액세스된 URL 을 표시합니다 .
      - **URLs That Were Accessed More Than** *숫자* **Times.** 지정된 횟수보다 많이 액세스된 URL 을 표시합니다 .
    - **Hosts Accessing Your Server.**Proxy Server 에 액세스한 호스트를 표시합니다 .
      - ***숫자*** **Hosts Most Often Accessing Your Server.** 서버에 가장 자주 액세스한 호스트 또는 지정된 횟수보다 많이 서버에 액세스한 호스트를 표시합니다 .
      - **Hosts That Accessed Your Server More Than** *Number* **Times.** 지정된 횟수보다 많이 서버에 액세스한 호스트를 표시합니다 .

17. 결과를 확인할 순서를 지정합니다 . 항목의 우선 순위를 1 에서 3 까지 지정하여 각 세션이 보고서에 표시될 순서를 정합니다 . 이 중 생성하려는 것이 없는 경우 해당 부분은 자동으로 무시됩니다 . 다음 옵션에서 선택하십시오 .
    - Find Totals
    - General Statistics.
    - Make Lists
18. OK 를 누릅니다 . 새 창에 보고서가 표시됩니다 .

명령줄에서 로그 분석기를 실행하려면 다음과 같이 합니다 .

명령줄에서 액세스 로그 파일을 분석하려면 `flexanlg` 도구를 실행합니다 . 이 도구는 `server-install/extras/flexanlg` 디렉토리에 있습니다 .

`flexanlg` 를 실행하려면 명령 프롬프트에서 다음 명령과 옵션을 입력합니다 .

```
./flexanlg [-n name] [-x] [-r] [-p order] [-i file]* [ -m metafile ]*
[-o file][-c opts] [-t opts] [-l opts]
```

* 표시된 옵션은 반복할 수 있습니다 .

구문은 다음의 설명과 같습니다 . (./flexanlg -h 를 입력하여 이 정보를 온라인으로 구할 수 있습니다).

```
-P: proxy log format                                    Default: no
-n servername: The name of the server
-x : Output in HTML                                     Default: no
-r : Resolve IP addresses to hostnames                  Default: no
-p [c,t,l]: Output order (counts, time stats, lists)  Default: ctl
-i filename: Input log file                             Default: none
-o filename: Output log file                            Default: stdout
-m filename: Meta file                                  Default: none
-c [h,n,r,f,e,u,o,k,c,z]: Count these item(s) -       Default: hnreuokc
    h: total hits
    n: 304 Not Modified status codes (Use Local Copy)
    r: 302 Found status codes (Redirects)
    f: 404 Not Found status codes (Document Not Found)
    e: 500 Server Error status codes (Misconfiguration)
    u: total unique URL's
    o: total unique hosts
    k: total kilobytes transferred
    c: total kilobytes saved by caches
    z: Do not count any items.
-t [sx,mx,hx, xx,z]: Find time stats -      Default:s5m5h10u10a10r10x10
    s(number): Find top (number) seconds of log
    m(number): Find top (number) minutes of log
    h(number): Find top (number) hours of log
    u(number): Find top (number) users of log
    a(number): Find top (number) user agents of log
    r(number): Find top (number) referers of log
    x(number): Find top (number) for miscellaneous keywords
    z: Do not find any time stats.
-l [cx,hx]: Make a list of -                            Default: c+3h5
    c(x,+x): Most commonly accessed URL's
             (x: Only list x entries)
             (+x: Only list if accessed more than x times)
    h(x,+x): Hosts (or IP addresses) most often accessing your server
             (x: Only list x entries)
             (+x: Only list if accessed more than x times)
    z: Do not make any lists.
```

이벤트 보기 (Windows)

서버 오류 로그에 오류를 기록하는 것 외에 Proxy Server 는 Event Viewer 에 심각한 시스템 오류를 기록합니다 . Event Viewer 를 사용하여 시스템의 이벤트를 모니터할 수 있습니다 . Event Viewer 를 사용하여 기능적 구성 문제로 인한 오류를 볼 수 있습니다 . 이 오류는 오류 로그가 열리기 전에 발생할 수 있습니다 .

Event Viewer 를 사용하려면 다음과 같이 합니다 .

1. 시작 메뉴에서 모든 프로그램을 선택한 후 관리 도구를 선택합니다 . 관리 도구 프로그램 그룹에서 Event Viewer 를 선택합니다 .
2. Log 메뉴에서 Application 을 선택합니다 .

   Event Viewer 에 Application 로그가 표시됩니다 . Proxy Server 오류에는 `proxy-server`*id* 의 소스 레이블이 포함됩니다 .
3. View 메뉴에서 Find 를 선택하여 로그에서 이들 레이블 중 한 가지를 검색합니다 . 로그 항목을 업데이트하려면 Veiw 메뉴에서 Refresh 를 선택합니다 .

   Event Viewer 에 대한 자세한 내용은 시스템 설명서를 참조하십시오 .

서버 모니터

이 장에서는 내장 모니터 도구 및 SNMP(Simple Network Management Protocol) 등을 포함하여 서버를 모니터하는 방법에 대해 설명합니다 .

SNMP 를 Sun Java System MIB(관리 정보 베이스) 및 HP OpenView 등의 네트워크 소프트웨어와 함께 사용하여 네트워크에서 기타 장치를 모니터하는 것과 마찬가지로 서버를 실시간으로 모니터할 수 있습니다 .

참고 Windows 의 경우 Proxy Server 4 를 설치하기 전에 컴퓨터에 Windows SNMP 구성 요소가 이미 설치되었는지 확인해야 합니다 .

통계 기능 또는 SNMP 를 사용하여 실시간으로 서버의 상태를 확인할 수 있습니다 . UNIX 또는 Linux 를 사용하는 경우 SNMP 를 사용하려면 반드시 Proxy Server 를 SNMP 용으로 구성해야 합니다 . 이 장에서는 UNIX 또는 Linux 에서 Proxy Server 와 함께 SNMP 를 사용하는 데 필요한 정보를 제공합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

- SNMP 메시지 이해

통계를 사용하여 서버 모니터

통계 기능을 사용하여 서버의 현재 작동을 모니터할 수 있습니다 . 통계에는 서버가 처리하는 요청의 수와 해당 요청을 처리하는 상태 등이 표시됩니다 . 대화형 서버 모니터 보고서에 서버가 많은 수의 요청을 처리하는 것으로 표시되는 경우 요청을 수용하도록 서버 구성 또는 시스템의 네트워크 커널을 조정할 수 있습니다 . 통계를 수집하면 Proxy Server 에 오버헤드가 추가되므로 통계는 사용하지 않도록 기본 설정되어 있습니다 . 통계를 사용하면 서버가 통계 정보를 수집하여 저장하기 시작합니다 .

통계를 사용하도록 설정하면 다음 영역에 대한 통계를 볼 수 있습니다 .

- 연결
- DNS
- KeepAlive
- 캐시
- 서버 요청

대화형 서버 모니터 보고서의 다양한 서버 통계에 대한 전체적인 설명은 온라인 도움말의 Monitor Current Activity 페이지를 참조하십시오 .

Proxy Server Statistics 처리

내장 함수 `stats-xml` 을 사용하여 Proxy Server 통계를 수집합니다 . Server Manager 에서 통계를 보거나 `perfdump` 함수를 사용하여 보고서를 생성하려면 이 함수를 사용하도록 설정해야 합니다 . `stats-xml` 함수로도 프로파일 작성을 사용하도록 설정할 수 있습니다 . 이 함수는 사용자 정의 NSAPI 함수를 사용한 모니터링 통계에 필요합니다 . 서버에서 통계와 프로필 작성을 사용하도록 설정하면 `obj.conf` 파일의 서버 함수 `stats-init` 를 초기화하여 통계 수집을 시작합니다 .

```
Init profiling="on" fn="stats-init"
```

또한 브라우저 창을 통해 통계에 액세스할 수 있도록 하는 `NameTrans` 지시문을 만듭니다 .

```
NameTrans fn="assign-name" name="stats-xml"
from="(/stats-xml|/stats-xml/.*)
```

마지막으로 , 통계를 사용하도록 설정하면 Service 지시문을 추가하여 NameTrans 지시문이 선택되었을 때 stats-xml 함수를 처리합니다 .

```
<Object name="stats-xml">

Service fn="stats-xml"

</Object>
```

참고 통계 수집은 obj.conf 에서 Init 함수를 업데이트하므로 이러한 변경 사항을 적용하려면 서버를 중지하고 다시 시작해야 합니다 .

다음 URL 을 사용하여 stats-xml 출력을 검색할 수 있습니다 .

http://*computer_name:proxyport*/stats-xml/proxystats.xml

이 요청은 Proxy Server 통계가 들어 있는 XML 페이지를 반환합니다 . 일부 브라우저에서는 브라우저 창 내에서 데이터를 볼 수 있지만 어떤 브라우저에서는 외부 파일로 데이터를 저장하여 외부 뷰어를 사용해 봐야 합니다 . 분석을 위해 다양한 데이터 보기 통계를 파싱할 수 없으면 이 정보가 유용하지 않을 수도 있습니다 . 이 과정에서 타사 도구를 사용하는 것이 도움이 될 수 있습니다 . 파싱 도구가 없으면 stats-xml 출력은 Server Manager 나 perfdump SAF 를 통해 가장 잘 확인할 수 있습니다 .

stats-xml 출력 액세스 제한

브라우저에서 서버의 stats-xml 통계를 볼 수 있는 사용자를 제한하려면 /stats-xml URI 용 ACL 을 만들어야 합니다 .

ACL 파일은 obj.conf 파일의 stats-xml 개체 정의에서도 참조할 수 있습니다 . 예를 들어 /stats-xml URI 에 대한 명명된 ACL 을 만든 경우 , 다음과 같이 개체 정의의 PathCheck 문의 ACL 파일을 참조해야 합니다 .

```
<Object name="stats-xml">

PathCheck fn="check-acl" acl="stats.acl"

Service fn="stats-xml"

</Object>
```

통계 사용 설정

성능을 모니터하려면 Proxy Server 에서 통계를 활성화해야 합니다 . Server Manager 나 , obj.conf 및 magnus.conf 파일 편집을 통해 통계를 활성화할 수 있습니다 . 자동화된 도구를 만들거나 모니터 및 조정용 사용자 정의 프로그램을 작성하는 사용자는 stats-xml 에서 직접 작업하는 것을 선호할 수 있습니다 .

주의 통계 / 프로필을 작성하면 서버의 모든 사용자가 통계 정보를 볼 수 있습니다 .

Server Manager 에서 통계를 사용하도록 설정하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Monitor Current Activity 링크를 누릅니다 . Monitor Current Activity 페이지가 표시됩니다 .
3. 통계를 사용하려면 Activate Statistics/Profiling? 항목에서 Yes 옵션을 선택합니다 .
4. OK 를 누릅니다 .
5. Restart Required 를 누릅니다 . Apply Changes 페이지가 나타납니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

stats-xml 을 통해 통계 사용을 설정하려면 다음과 같이 합니다 .

1. obj.conf 의 기본 개체 아래에 다음 행을 추가합니다 .

```
NameTrans fn="assign-name" name="stats-xml"
from="(/stats-xml|/stats-xml/.*)"
```

2. 다음 서비스 함수를 obj.conf 에 추가합니다 .

```
<Object name="stats-xml">

Service fn="stats-xml"

</Object>
```

3. stats-init SAF 를 magnus.conf 에 추가합니다 .

obj.conf 의 stats-init 예는 다음과 같습니다 .

```
Init profiling="on" fn="stats-init" update-interval="5"
```

위의 예를 통해 다음을 지정하는 방법도 알 수 있습니다 .

- **update-interval.** 통계 업데이트 사이의 간격을 초 단위로 나타냅니다 . 설정값이 높을수록 빈도수가 작으며 더 나은 성능을 낼 수 있습니다. 최소값은 1, 기본값은 5 입니다 .
- **profiling.** NSAPI 성능 프로필 작성을 활성화합니다 . 기본값은 "no" 이며 서버 성능이 약간 향상됩니다 . 그러나 사용자 인터페이스를 통해 통계를 활성화하면 기본적으로 프로필 작성은 사용하도록 설정됩니다 .

통계 사용

통계를 사용하도록 설정하면 서버 인스턴스 작동에 대한 다양한 정보를 얻을 수 있습니다 . 통계는 기능적 영역으로 나누어집니다 .

통계에 액세스하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Monitor Current Activity 링크를 누릅니다 .
3. Select Refresh Interval 드롭다운 목록에서 새로 고침 간격을 선택합니다 .

   새로 고침 간격은 표시되는 통계 정보를 업데이트하는 초 단위 간격입니다 .
4. Select Statistics To Be Displayed 드롭다운 목록에서 표시할 통계 유형을 선택합니다 . 통계 유형에 대한 자세한 내용은 "Server Manager 에 통계 표시 " (209 페이지) 를 참조하십시오 .
5. Submit 를 누릅니다 .

   서버 인스턴스가 실행 중이며 Statistics/Profiling 을 사용하는 경우 선택한 종류의 통계를 표시하는 페이지가 나타납니다 . 이 페이지는 새로 고침 간격에서 선택한 값에 따라 5-15 초마다 업데이트됩니다 .
6. 드롭다운 목록에서 프로세스 ID 를 선택합니다 .

   Server Manager 에서 현재 작동을 확인할 수 있으나 이러한 범주가 서버 조정에 전적으로 해당하는 것은 아닙니다 . 서버 조정에는 perfdump 통계를 사용하는 것이 좋습니다 .

Server Manager 에 통계 표시

이 절에서는 Server Manager 에서 `proxystats.xml` 데이터 하위 집합을 확인하는 방법을 설명합니다 .

Proxy Server 연결 , DNS 처리 , 연결 유지 값 , 캐시 및 서버 요청과 관련한 총계 , 최대 값 , 최고 수 , 정보 막대 그래프 등을 확인할 수 있습니다 .

다음 절에서는 이러한 각 항목에 대해 구할 수 있는 정보 유형을 설명합니다 .

Connection Statistics

Server Manager 에서 사용할 수 있는 연결 통계는 다음과 같습니다 .

- 총 연결 수
- 큐에 저장된 최대 연결 수
- 큐에 저장된 최고 연결 수
- 큐에 저장된 현재 연결 수
- 프로세스 수

DNS Statistics

Server Manager 에서 사용할 수 있는 DNS 통계는 다음과 같습니다 .

- 최대 DNS 캐시 항목 수
- 프로세스 수
- DNS 캐시 적중 횟수 (막대 그래프로도 표시됨)
- DNS 캐시 누락 횟수 (막대 그래프로도 표시됨)

Keep-Alive Statistics

Server Manager 에서 사용할 수 있는 연결 유지 통계는 다음과 같습니다 .

- 최대 연결 유지 연결 수
- 연결 유지 시간 제한
- 프로세스 수
- 연결 유지 적중 수 (막대 그래프로도 표시됨)
- 연결 유지 플러시 수 (막대 그래프로도 표시됨)
- 연결 유지 거절 수 (막대 그래프로도 표시됨)
- 연결 유지 시간 제한 수 (막대 그래프로도 표시됨)

Cache Statistics

Server Manager 에서 사용할 수 있는 캐시 통계는 다음과 같습니다 .

- 캐시의 최대 지속 시간 (초)
- 최대 힙 캐시 크기
- 최대 메모리 캐시 맵 크기

- 프로세스 수
- 캐시 적중 횟수 (막대 그래프로도 표시됨)
- 캐시 누락 횟수 (막대 그래프로도 표시됨)
- 정보 캐시 적중 횟수 (막대 그래프로도 표시됨)
- 정보 캐시 누락 횟수 (막대 그래프로도 표시됨)
- 컨텐츠 캐시 적중 횟수 (막대 그래프로도 표시됨)
- 컨텐츠 캐시 누락 횟수 (막대 그래프로도 표시됨)

Server Request Statistics

Server Manager 에서 사용할 수 있는 서버 통계는 다음과 같습니다 .

- 총 요청 수
- 수신된 바이트 수
- 전송된 바이트 수
- 프로세스 수
- HTTP 서버 코드당 요청 분석 (막대 그래프로도 표시됨) 예를 들어 HTTP 서버 코드 200 은 이행된 요청을 나타냅니다 .

perfdump 유틸리티를 사용한 현재 작동 모니터

perfdump 유틸리티는 Proxy Server 내부 통계로부터 다양한 성능 데이터를 수집하여 ASCII 텍스트로 표시하는 Proxy Server 에 내장된 SAF(Server Application Function) 입니다 . perfdump 유틸리티를 사용하면 Server Manager 에서보다 더욱 다양한 통계를 모니터할 수 있습니다 .

perfdump 로 통계를 통합할 수 있습니다 . 단일 프로세스에 대한 모니터가 아니라 통계에 프로세스 수가 곱해져 서버 전반을 보다 정확하게 확인할 수 있습니다 .

perfdump 유틸리티 사용 설정

`stats-xml` 함수를 사용하도록 설정한 후에만 perfdump SAF 를 사용할 수 있으며 `obj.conf` 파일을 직접 편집해야만 사용하도록 설정할 수 있습니다 .

perfdump SAF 를 사용하도록 설정하려면 다음과 같이 합니다 .

1. obj.conf 파일의 기본 개체 뒤에 다음 개체를 추가합니다 .

   ```
   <Object name="perf">
   Service fn="service-dump"
   </Object>
   ```

2. 다음을 기본 개체에 추가합니다 .

   ```
   NameTrans fn=assign-name from="/.perf" name="perf"
   ```

3. 서버 소프트웨어를 재시작합니다 .
4. 다음 URL 을 입력하여 perfdump 에 액세스합니다 .

   http://*computer_name:proxyport*/.perf

   perfdump 통계를 요청하고 브라우저가 자동으로 새로 고침을 수행할 간격을 초 단위로 지정할 수 있습니다 . 다음 예에서는 5 초 간격으로 새로 고침을 수행합니다 .

   http://*computer_name:proxyport*/.perf?refresh=5

perfdump 출력 예

다음은 perfdump 출력의 예입니다 .

```
proxyd pid: 6751

Sun Java System Web Proxy Server 4.0 B05/02/2005 15:32 (SunOS DOMESTIC)

Server started Thu May 19 13:15:14 2005
Process 6751 started Thu May 19 13:15:14 2005

ConnectionQueue:
----------------------------------------
Current/Peak/Limit Queue Length           0/1/4096
Total Connections Queued                  1
Average Queue Length (1, 5, 15 minutes)   0.00, 0.00, 0.00
Average Queueing Delay                    0.09 milliseconds

ListenSocket ls1:
-----------------------
Address                  http://0.0.0.0:8081
Acceptor Threads         1

KeepAliveInfo:
--------------------
KeepAliveCount        0/256
KeepAliveHits         0
```

```
KeepAliveFlushes      0
KeepAliveRefusals     0
KeepAliveTimeouts     0
KeepAliveTimeout      30 seconds

SessionCreationInfo:
------------------------
Active Sessions           1
Keep-Alive Sessions       0
Total Sessions Created    48/128

CacheInfo:
------------------
enabled             yes
CacheEntries        0/1024
Hit Ratio           0/0 (  0.00%)
Maximum Age         0

Native pools:
---------------------------
NativePool
Idle/Peak/Limit               1/1/128
Work Queue Length/Peak/Limit  0/0/0

Server DNS cache disabled

Async DNS disabled

Performance Counters:
-----------------------------------------------
                           Average         Total      Percent

Total number of requests:                      1
Request processing time:    0.2559        0.2559

default-bucket (Default bucket)
Number of Requests:                            1    (100.00%)
Number of Invocations:                         7    (100.00%)
Latency:                    0.2483        0.2483    ( 97.04%)
Function Processing Time:   0.0076        0.0076    (  2.96%)
Total Response Time:        0.2559        0.2559    (100.00%)

Sessions:
---------------------------
Process  Status     Function

6751     response   service-dump
```

이 매개 변수에 대한 자세한 내용은 다음에서 제공하는 Sun ONE Web Server 6.1 Performance Tuning, Sizing, and Scaling Guide 의 제 2 장에 있는 "Using Statistics to Tune Your Server" 를 참조하십시오 .

http://docs.sun.com/source/817-6249/index.html

perfdump 출력 액세스 제한

브라우저에서 서버의 perfdump 통계를 볼 수 있는 사용자를 제한하려면 `/.perf` URI 용 ACL 을 만들어야 합니다 .

ACL 파일은 `obj.conf` 파일의 perf 개체 정의에서도 참조할 수 있습니다 . 예를 들어 `/.perf` URI 에 대한 명명된 ACL 을 만든 경우 , 다음과 같이 개체 정의에 있는 `PathCheck` 문의 ACL 파일을 참조해야 합니다 .

```
<Object name="perf">
PathCheck fn="check-acl" acl="perf.acl"
Service fn="service-dump"
</Object>
```

성능 버킷 사용

성능 버킷을 통해 버킷을 정의하고 버킷과 여러 서버 함수를 연결할 수 있습니다 . 이 함수 중 하나를 호출할 때마다 서버는 통계 데이터를 수집하여 버킷에 추가합니다 . 예를 들어 `send-cgi` 및 `NSServletService` 는 각각 CGI 및 Java 서블릿 요청을 서비스하는 데 사용하는 함수입니다 . 두 버킷을 정의하여 CGI 및 서블릿 요청에 대한 카운터를 각각 관리하거나 , 두 가지 동적 컨텐츠 유형에 대한 요청을 모두 계산하는 하나의 버킷을 만들 수 있습니다 . 이 정보를 수집하는 데는 거의 부하가 발생하지 않으며 서버 성능에 미치는 영향도 미미합니다 . 차후 `perfdump` 유틸리티를 사용하여 이 정보에 액세스할 수 있습니다 . 버킷에 저장되는 정보는 다음과 같습니다 .

- **Name of the bucket.** 버킷을 함수에 연결하는 데 사용할 이름입니다 .
- **Description.** 버킷이 연결된 함수의 설명입니다 .
- **Number of requests for this function.** 함수 호출을 발생시킨 총 요청 수입니다 .
- **Number of times the function was invoked.** 어떤 함수는 한 요청에서 여러 번 실행되기도 하므로 이 숫자는 함수 요청 수와 일치하지 않을 수 있습니다 .
- **Function latency or the dispatch time.** 서버가 함수를 호출하는 데 걸린 시간입니다 .

- **Function time.** 함수 자체에서 소모한 시간입니다 .

default-bucket 은 서버에서 미리 정의됩니다 . 사용자 정의 버킷과 연결되지 않은 함수에 대한 통계를 기록합니다 .

구성

magnus.conf 및 obj.conf 파일에 성능 버킷에 대한 모든 구성 정보를 지정해야 합니다 . 기본 버킷만 자동으로 사용 설정되어 있습니다 .

우선 "perfdump 유틸리티를 사용한 현재 작동 모니터 " (211 페이지) 에서 설명하는 성능 측정을 사용하도록 설정해야 합니다 .

다음은 magnus.conf 에서 새 버킷을 정의하는 방법을 설명하는 예입니다 .

```
Init fn="define-perf-bucket" name="acl-bucket" description="ACL bucket"

Init fn="define-perf-bucket" name="file-bucket" description="Non-cached
responses"

Init fn="define-perf-bucket" name="cgi-bucket" description="CGI Stats"
```

위의 예에서는 acl-bucket, file-bucket 및 cgi-bucket 등의 세 가지 버킷을 만듭니다 . 이 버킷을 함수와 연결하려면 bucket=*bucket-name* 을 성능을 측정할 obj.conf 함수에 추가합니다 .

예

```
PathCheck fn="check-acl" acl="default" bucket="acl-bucket"

...

Service method="(GET|HEAD|POST)" type="*~magnus-internal/*"
fn="send-file" bucket="file-bucket"

...

<Object name="cgi">

ObjectType fn="force-type" type="magnus-internal/cgi"

Service fn="send-cgi" bucket="cgi-bucket"

</Object>
```

성능 보고서

버킷의 서버 통계는 perfdump 유틸리티를 사용하여 액세스할 수 있습니다 . 성능 버킷 정보는 perfdump 가 반환한 보고서의 끝 부분에 있습니다 .

이 보고서는 다음과 같은 정보를 포함합니다 .

- Average, Total 및 Percent 열은 각 요청 통계에 대한 데이터를 표시합니다 .
- Request Processing Time 은 서버가 지금까지 수신한 모든 요청을 처리하기 위해 필요한 총 시간을 나타냅니다 .
- Number of Requests 는 함수에 대한 총 요청 수입니다 .
- Number of Invocations는 함수를 호출한 총 횟수입니다. 한 요청을 처리하는 동안 함수를 여러 번 호출할 수 있으므로 이 숫자는 요청 수와 다릅니다 . 모든 버킷에 대한 총 호출 수를 참조하여 이 행에 대한 퍼센트 열을 계산합니다 .
- Latency는 Proxy Server가 함수 호출을 준비하는 데 소요한 시간(초 단위)입니다.
- Function Processing Time은 Proxy Server가 함수 내에서 소요한 시간(초 단위)입니다 . Function Processing Time 퍼센트와 Total Response Time 은 총 Request Processing Time 을 참조하여 산출됩니다 .
- Total Response Time 은 Function Processing Time 과 Latency 의 합 (초 단위) 입니다 .

perfdump 를 통해 사용할 수 있는 성능 버킷 정보의 예는 다음과 같습니다 .

```
Performance Counters:
------------------------------------------------
                           Average         Total      Percent

Total number of requests:                      1
Request processing time:    0.2559        0.2559

default-bucket (Default bucket)
Number of Requests:                            1    (100.00%)
Number of Invocations:                         7    (100.00%)
Latency:                    0.2483        0.2483    ( 97.04%)
Function Processing Time:   0.0076        0.0076    (  2.96%)
Total Response Time:        0.2559        0.2559    (100.00%)
```

SNMP 기초

SNMP 는 네트워크 작동에 대한 데이터를 교환하는 데 사용하는 프로토콜입니다 . SNMP 를 사용하면 데이터가 관리된 장치와 네트워크 관리 스테이션 (NMS) 사이에서 전송됩니다 . 관리된 장치는 SNMP 를 실행하는 장치로 네트워크 상의 호스트 , 라우터 , 프록시 서버 및 다른 서버 등이 될 수 있습니다 . NMS 는 네트워크를 원격으로 관리하는 시스템입니다 . 보통 NMS 소프트웨어는 수집된 데이터를 표시하는 그래프를 제공하거나 해당 데이터를 사용하여 서버가 특정 임계치 내에서 작동하는지 확인합니다 .

NMS 는 보통 하나 이상의 네트워크 관리 응용 프로그램이 설치된 고기능 워크스테이션입니다. HP OpenView 등의 네트워크 관리 응용 프로그램은 웹 서버 등의 관리된 장치에 대한 정보를 그래픽으로 표시합니다 . 예를 들어 기업에서 작동 중이거나 중지된 서버를 표시할 수 있으며 수신된 오류 메시지의 수와 유형을 표시할 수 있습니다 . 프록시 서버에서 SNMP 를 사용하는 경우 이 정보는 하위 에이전트와 마스터 에이전트 등 두 종류의 에이전트를 통하여 NMS 와 서버 사이에 전송됩니다 .

하위 에이전트는 서버에 대한 정보를 수집하며 해당 정보를 서버의 마스터 에이전트로 전달합니다 . Administration Server 를 제외한 모든 서버에는 하위 에이전트가 있습니다 .

참고 SNMP 구성을 변경한 경우에는 반드시 Apply Required 를 눌러 SNMP 하위 에이전트가 다시 시작되도록 해야 합니다 .

마스터 에이전트는 NMS 와 통신합니다 . 마스터 에이전트는 Administration Server 에 설치됩니다 .

호스트 컴퓨터에 여러 개의 하위 에이전트가 있을 수 있으나 마스터 에이전트는 하나만 있어야 합니다 . 예를 들어 동일한 호스트에 Directory Server, Proxy Server 및 Messaging Server 가 설치된 경우 각 서버의 하위 에이전트가 동일한 마스터 에이전트와 통신합니다 .

Management Information Base

Proxy Server 에는 네트워크 관리에 관련된 변수가 저장됩니다 . 마스터 에이전트가 액세스할 수 있는 변수는 관리된 개체라고 합니다 . 이 개체는 MIB(Management Information Base) 라고 하는 트리 형식의 구조로 정의됩니다 . MIB 은 서버의 네트워크 구성 , 상태 및 통계에 대한 액세스를 제공합니다 . SNMP 를 사용하면 NMS 에서

이 정보를 볼 수 있습니다 . MIB 트리의 최상위 수준에는 인터넷 개체 아이디가 표시되며 , 여기에는 directory(1), mbmt(2), experimental(3) 및 private(4) 의 네 가지 하위 트리가 있습니다 . private(4) 하위 트리에는 enterprise(1) 노드가 있습니다 . enterprises(1) 의 각 하위 트리는 개별 엔터프라이즈에 지정되며 , 해당 엔터프라이즈는 자체의 MIB 확장자를 등록한 조직입니다 . 따라서 엔터프라이즈는 자체의 하위 트리에 제품 특정 하위 트리를 만들 수 있습니다 . 회사가 만든 MIB 은 enterprises(1) 노드 아래에 위치합니다 . Sun Java System 서버 MIB 도 enterprises(1) 노드 아래에 위치합니다 . 각 Sun Java System 서버 하위 에이전트는 SNMP 통신에서 사용할 MIB 을 제공합니다 . 서버는 이러한 변수가 포함된 메시지 또는 트랩을 전송하여 NMS 에 중요한 이벤트를 보고합니다 . NMS 는 서버 MIB 에서 데이터 쿼리를 수행하거나 MIB 의 변수를 원격으로 변경할 수 있습니다 . 각 Sun Java System 서버는 고유의 MIB 를 갖습니다 . 모든 Sun Java System 서버 MIB 는 다음 위치에 있습니다 .

```
server_root/plugins/snmp
```

Proxy Servers MIB 는 proxyserv40.mib 파일입니다 . 이 MIB 에는 Proxy Server 용 네트워크 관리에 관련된 다양한 변수의 정의가 포함됩니다 . Proxy Server MIB 를 사용하여 실시간으로 Proxy Server 에 대한 관리 정보를 볼 수 있으며 서버를 모니터할 수 있습니다 .

SNMP 설정

일반적으로 SNMP 를 사용하려면 반드시 시스템에 마스터 에이전트와 최소한 하나의 하위 에이전트가 설치되고 실행되어야 합니다 . 하위 에이전트를 사용하기 전에 마스터 에이전트를 설치해야 합니다 .

SNMP 를 설정하는 방법은 시스템에 따라 다릅니다 .

시작하기 전에 두 가지를 확인해야 합니다 .

- 시스템에 이미 SNMP 에이전트 (운영 체제의 원시 에이전트) 가 실행되고 있는가 ?
- 있는 경우 SNMP 에이전트가 SMUX 통신을 지원하는가 ? AIX 플랫폼을 사용하는 경우에는 시스템이 SMUX 를 지원합니다 .

이 정보를 확인하는 방법은 시스템 설명서를 참조하십시오 .

참고 Administration Server 에서 SNMP 설정을 변경하거나 , 새 서버를 설치하거나 , 기존 서버를 삭제한 후에는 반드시 다음과 같이 해야 합니다 .

- (Windows) Windows SNMP 서비스를 재시작하거나 시스템을 재부팅합니다 .
- (UNIX) Administration Server 를 사용하여 SNMP 마스터 에이전트를 재시작합니다 .

표 1 SNMP 마스터 에이전트 및 하위 에이전트 사용 설정을 위한 절차 개요

시스템의 전제 조건	수행할 작업 (다음 절에서 자세히 설명)
• 현재 실행되는 원시 에이전트 없음	1. 마스터 에이전트를 시작합니다 . 2. 서버에 설치된 각 서버용 하위 에이전트를 사용하도록 설정합니다 .
• 현재 원시 에이전트 실행 • SMUX 없음 • 원시 에이전트를 사용하여 계속할 필요 없음	1. Administration Server 용 마스터 에이전트를 설치할 때 원시 에이전트를 중지합니다 . 2. 마스터 에이전트를 시작합니다 . 3. 서버에 설치된 각 서버용 하위 에이전트를 사용하도록 설정합니다 .
• 현재 원시 에이전트 실행 • SMUX 없음 • 원시 에이전트를 사용하여 계속	1. 프록시 SNMP 에이전트를 설치합니다 . 2. 마스터 에이전트를 시작합니다 . 3. 해당 프록시 SNMP 에이전트를 시작합니다 . 4. 마스터 에이전트 포트 번호가 아닌 포트 번호를 사용하여 원시 에이전트를 재시작합니다 . 5. 서버에 설치된 각 서버용 하위 에이전트를 사용하도록 설정합니다 .
• 현재 원시 에이전트 실행 • SMUX 지원	1. SNMP 원시 에이전트를 재구성합니다 . 2. 서버에 설치된 각 서버용 하위 에이전트를 사용하도록 설정합니다 .

프록시 SNMP 에이전트 사용 (UNIX)

이미 원시 에이전트가 실행 중이며 이를 Proxy Server 마스터 에이전트와 동시에 사용하려는 경우 프록시 SNMP 에이전트를 사용해야 합니다 . 시작하기 전에 원시 마스터 에이전트가 중단되었는지 확인합니다 . 자세한 내용은 시스템 설명서를 참조하십시오 .

참고 프록시 에이전트를 사용하려면 이를 설치한 후 시작해야 합니다 . 또한 Proxy Server 마스터 에이전트가 실행되는 포트 번호가 아닌 다른 포트 번호를 사용하여 원시 SNMP 마스터 에이전트를 재시작해야 합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- Proxy SNMP Agent 시작
- 프록시 SNMP 에이전트 시작
- 원시 SNMP 데몬 재시작

Proxy SNMP Agent 시작

SNMP 에이전트가 시스템에서 실행되며 원시 SNMP 데몬을 계속 사용하려면 다음과 같이 합니다 .

1. SNMP 마스터 에이전트를 설치합니다 . "SNMP 마스터 에이전트 설치 " (222 페이지) 를 참조하십시오 .
2. 프록시 SNMP 를 설치 및 시작하고 원시 SNMP 데몬을 재시작합니다 . " 프록시 SNMP 에이전트 사용 (UNIX)" (220 페이지) 을 참조하십시오 .
3. SNMP 마스터 에이전트를 시작합니다 . "SNMP 마스터 에이전트 사용 설정 및 시작 " (223 페이지) 을 참조하십시오 .
4. 하위 에이전트를 사용하도록 설정합니다 . " 하위 에이전트 사용 설정 " (228 페이지) 을 참조하십시오 .

SNMP 프록시 에이전트를 설치하려면 서버 루트 디렉토리의 `plugins/snmp/sagt` 에 있는 `CONFIG` 파일 (다른 이름을 지정할 수 있음) 을 편집하여 SNMP 데몬이 수신할 포트를 포함시킵니다 . 또한 MIB 트리와 프록시 SNMP 에이전트가 전달할 트랩을 포함해야 합니다 .

`CONFIG` 파일의 예는 다음과 같습니다 .

```
AGENT AT PORT 1161 WITH COMMUNITY public
SUBTREES        1.3.6.1.2.1.1,
                   1.3.6.1.2.1.2,
                   1.3.6.1.2.1.3,
                   1.3.6.1.2.1.4,
                   1.3.6.1.2.1.5,
                   1.3.6.1.2.1.6,
                   1.3.6.1.2.1.7,
                   1.3.6.1.2.1.8
FORWARD ALL TRAPS;
```

프록시 SNMP 에이전트 시작

프록시 SNMP 에이전트를 시작하려면 명령 프롬프트에서 다음을 입력합니다 .

```
# sagt -c CONFIG&
```

원시 SNMP 데몬 재시작

프록시 SNMP 에이전트를 시작한 후 `CONFIG` 파일에서 지정한 포트에서 원시 SNMP 데몬을 재시작해야 합니다 . 원시 SNMP 에이전트를 재시작하려면 명령 프롬프트에서 다음을 입력합니다 .

`# snmpd -P` *port_number*

여기에서 *port_number* 는 `CONFIG` 파일에 지정된 포트 번호입니다 . 예를 들어 Solais 플랫폼의 경우 앞에서 언급한 예제의 `CONFIG` 파일의 포트를 사용하는 경우 다음을 입력합니다 .

```
# snmpd -P 1161
```

SNMP 원시 에이전트 재구성

SNMP 데몬이 AIX 에서 실행되는 경우 SMUX 가 지원됩니다 . 따라서 마스터 에이전트를 설치할 필요는 없습니다 . 그러나 AIX SNMP 데몬 구성을 변경해야 합니다 .

AIX 는 여러 구성 파일을 사용하여 통신을 검사합니다 . 이 중 한 가지인 `snmpd.conf` 를 변경하여 SNMP 데몬이 SMUX 하위 에이전트에서 들어오는 메시지를 받도록 해야 합니다 . 자세한 내용은 `snmpd.conf` 용 온라인 설명서 페이지를 참조하십시오 . 각 하위 에이전트를 정의하는 줄을 추가해야 합니다 .

예를 들어 다음 줄은 `snmpd.conf` 에 추가할 수 있습니다 .

```
smux 1.3.6.1.4.1.1.1450.1 "" IP_address net_mask
```

IP_address 는 하위 에이전트가 실행되는 호스트의 IP 주소이며 *net_mask* 는 이 호스트의 네트워크 마스크입니다 .

참고	루프백 주소 127.0.0.1 을 사용하면 안 되며 , 실제의 IP 주소를 사용해야 합니다 .

SNMP 마스터 에이전트 설치

SNMP 마스터 에이전트를 구성하려면 반드시 Administration Server 인스턴스를 `root` 사용자로 설치해야 합니다 . 그러나 `root` 가 아닌 사용자라도 웹 서버 인스턴스에서 SNMP 하위 에이전트가 마스터 에이전트와 함께 작동하도록 구성하여 MIB 찾아 보기 등의 기본적인 SNMP 작업을 수행할 수 있습니다 .

마스터 SNMP 에이전트를 설치하려면 다음과 같이 합니다 .

1. 루트로 로그인합니다 .
2. 포트 161 에 SNMP 데몬 (`snmpd`) 이 실행되는지 확인합니다 .

   실행되는 SNMP 데몬이 없으면 단계 4 로 계속합니다 .

   SnMP 데몬이 실행 중이면 데몬을 시작하는 방법과 지원하는 MIB 을 확인합니다 .
3. SNMP 데몬이 실행 중이면 해당 프로세스를 종료합니다 .
4. Administration Server 에서 Global Settings 탭의 Set SNMP Master Agent Trap 페이지를 선택합니다 .
5. 네트워크 관리 소프트웨어를 실행하는 시스템의 이름을 입력합니다 .
6. 네트워크 관리 시스템이 트랩을 청취할 포트 번호를 입력합니다 . 주로 사용하는 포트는 162 입니다 . 트랩에 대한 자세한 내용은 " 트랩 대상 구성 " (228 페이지) 을 참조하십시오 .
7. 트랩에서 사용할 커뮤니티 문자열을 입력합니다 . 커뮤니티 문자열에 대한 자세한 내용은 " 커뮤니티 문자열 구성 " (227 페이지) 을 참조하십시오 .
8. OK 를 누릅니다 .
9. Administration Server 에서 Global Settings 탭의 Set SNMP Master Agent Community 페이지를 선택합니다 .

10. 마스터 에이전트용 커뮤니티 문자열을 입력합니다 .
11. 커뮤니티용 작업을 선택합니다 .
12. New 를 누릅니다 .

SNMP 마스터 에이전트 사용 설정 및 시작

마스터 에이전트 작업은 `CONFIG` 라는 에이전트 구성 파일에 정의됩니다 . Server Manager 를 사용하여 `CONFIG` 파일을 편집하거나 , 파일을 직접 편집할 수 있습니다 . SNMP 하위 에이전트를 사용 설정하기 전에 반드시 마스터 SNMP 에이전트를 설치해야 합니다 .

마스터 에이전트를 시작할 때 "System Error: Could not bind to port" 와 유사한 바인드 오류가 발생하면 `ps -ef | grep snmp` 를 사용하여 `magt` 가 실행 중인지 확인합니다 . 실행되는 경우 `kill -9` *`pid`* 명령을 사용하여 해당 프로세스를 종료합니다 . SNMP 용 CGI 가 다시 작동을 시작할 것입니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 다른 포트에서 마스터 에이전트 시작
- SNMP 마스터 에이전트 직접 구성
- 마스터 에이전트 CONFIG 파일 편집
- sysContact 및 sysLocation 변수 정의
- SNMP 하위 에이전트 구성
- SNMP 마스터 에이전트 시작

다른 포트에서 마스터 에이전트 시작

Administration Interface 는 161 이 아닌 다른 포트에서 SNMP 에이전트를 시작하지 않을 것입니다 .

다른 포트에서 마스터 에이전트를 직접 시작하려면 다음과 같이 합니다 .

1. */server_root*`/plugins/snmp/magt/CONFIG` 를 편집하여 원하는 포트를 지정합니다 .

2. 다음과 같은 시작 스크립트를 실행합니다 .

   ```
   cd /server_root/proxy-admserv
   ./start -shell /server_root/plugins/snmp/magt/magt
   /server_root/plugins/snmp/magt/CONFIG
   /server_root/plugins/snmp/magt/INIT
   ```

마스터 에이전트가 원하는 포트에서 시작될 것입니다 . 그러나 사용자 인터페이스는 해당 에이전트가 실행되는 것을 감지할 수 있습니다 .

SNMP 마스터 에이전트 직접 구성

마스터 SNMP 에이전트를 직접 구성하려면 다음과 같이 합니다 .

1. 수퍼유저로 로그인합니다 .
2. 포트 161 에 SNMP 데몬 (snmpd) 이 실행되는지 확인합니다 .

   SnMP 데몬이 실행 중이면 데몬을 시작하는 방법과 지원하는 MIB 을 확인합니다 . 해당 프로세스를 종료합니다 .
3. 서버 루트 디렉토리의 plugins/snmp/magt 에 있는 CONFIG 파일을 편집합니다 .
4. (선택) CONFIG 파일에 sysContact 및 sysLocation 변수를 정의합니다 .

마스터 에이전트 CONFIG 파일 편집

마스터 SNMP 에이전트를 직접 구성하려면 다음과 같이 합니다 .

1. 수퍼유저로 로그인합니다 .
2. 포트 161 에 SNMP 데몬 (snmpd) 이 실행되는지 확인합니다 .

   SnMP 데몬이 실행 중이면 데몬을 시작하는 방법과 지원하는 MIB 을 확인합니다 . 해당 프로세스를 종료합니다 .
3. 서버 루트 디렉토리의 plugins/snmp/magt 에 있는 CONFIG 파일을 편집합니다 .
4. (선택) CONFIG 파일에 sysContact 및 sysLocation 변수를 정의합니다 .

sysContact 및 sysLocation 변수 정의

CONFIG 파일을 편집하여 sysContact 및 sysLocation MIB-II 변수를 지정하는 sysContact 및 sysLocation 의 초기값을 추가할 수 있습니다 . 이 예의 sysContact 및 sysLocation 문자열은 인용 부호 안에 넣습니다 . 공백 , 줄바꿈 , 탭 등을 포함하는 문자열은 인용 부호 안에 넣어야 합니다 . 또한 16 진수 표기법으로 값을 지정할 수 있습니다 .

sysContract 및 sysLocation 변수가 정의된 CONFIG 파일의 예는 다음과 같습니다 .

```
COMMUNITY          public
                   ALLOW ALL OPERATIONS

MANAGER            nms2
                   SEND ALL TRAPS TO PORT 162
                   WITH COMMUNITY public

INITIAL            sysLocation ?erver room
987 East Cannon Road
Mountain View, CA 94043
USA

INITIAL            sysContact "Jill Dawson
email: jdawson@example.com"
```

SNMP 하위 에이전트 구성

SNMP 하위 에이전트를 구성하여 서버를 모니터할 수 있습니다 .

SNMP 하위 에이전트를 구성하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Configure SNMP Subagent 링크를 누릅니다 . Configure SNMP Subagent 페이지가 표시됩니다 .
3. Master Host 필드에 서버의 이름과 도메인을 입력합니다 .
4. 운영 체제 정보를 포함하여 서버의 설명을 입력합니다 .
5. 서버를 담당하는 조직을 입력합니다 .
6. Location 필드에 서버의 절대 경로를 입력합니다 .
7. Contact 필드에 서버를 담당하는 담당자의 이름과 연락처 정보를 입력합니다 .

8. Enable the SNMP Statistics Collection 에 On 을 선택합니다 .
9. OK 를 누릅니다 .
10. Restart Required 를 누릅니다 . Apply Changes 페이지가 나타납니다 .
11. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

SNMP 마스터 에이전트 시작

SNMP 마스터 에이전트를 설치하면 에이전트를 직접 시작하거나 Administration Server 를 이용하여 시작할 수 있습니다 .

SNMP 마스터 에이전트 직접 시작

마스터 에이전트를 직접 시작하려면 명령 프롬프트에서 다음을 입력합니다 .

```
# magt CONFIG INIT&
```

`INIT` 파일은 MIB-II 시스템 그룹으로부터의 정보를 포함하는 비휘발성 파일로 여기에는 시스템 위치와 연락처 정보가 있습니다 . `INIT` 파일이 없는 경우 마스터 에이전트를 처음 시작하면 파일이 만들어집니다 . `CONFIG` 파일에 잘못된 관리자 이름이 있는 경우 마스터 에이전트가 시작할 수 없습니다 .

비표준 포트에서 마스터 에이전트를 시작하려면 다음 중 한 가지 방법을 사용합니다 .

방법 1:`CONFIG` 파일에서 마스터 에이전트가 관리자로부터의 SNMP 요청을 수신할 각 인터페이스에 대한 전송 매핑을 지정합니다 . 전송 매핑을 사용하면 마스터 에이전트가 표준 포트뿐 아니라 비표준 포트의 연결을 수락합니다 . 마스터 에이전트는 또한 비표준 포트의 SNMP 트래픽을 수락합니다 . 대상 시스템의 한계에 의하여 정해진 동시 SNMP 의 최대 수에 따라 각 프로세스에 대한 개방 소켓 또는 파일 기술자의 수가 제한됩니다 . 전송 매핑 항목의 예는 다음과 같습니다 .

```
TRANSPORT          extraordinary   SNMP
                   OVER UDP SOCKET
                    AT PORT 11161
```

`CONFIG` 파일을 직접 편집한 후 명령 프롬프트에서 다음을 입력하여 마스터 에이전트를 직접 시작해야 합니다 .

```
# magt CONFIG INIT&
```

방법 2:`/etc/services` 파일을 편집하여 마스터 에이전트가 표준 포트뿐 아니라 비표준 포트의 연결을 수락하도록 합니다 .

Administration Server 를 사용하여 SNMP 마스터 에이전트 시작

Administration Server 를 사용하여 SNMP 마스터 에이전트를 시작하려면 다음과 같이 합니다 .

1. Administration Server 에 로그인합니다 .
2. Administration Server 에서 Global Settings 탭의 Control SNMP Master Agent 페이지를 선택합니다 .
3. Start 를 누릅니다 .

또한 Control SNMP Master Agent 페이지에서 SNMP 마스터 에이전트를 중지하고 재시작할 수 있습니다 .

SNMP 마스터 에이전트 구성

마스터 에이전트를 사용 설정하고 호스트 컴퓨터의 하위 에이전트를 사용 설정하면 호스트의 Administration Server 를 구성해야 합니다 . 여기에는 커뮤니티 문자열과 트랩 대상을 지정해야 합니다 .

커뮤니티 문자열 구성

커뮤니티 문자열은 SNMP 가 권한 부여에 사용하는 텍스트 문자열입니다 . 따라서 네트워크 관리 스테이션은 에이전트에 보내는 각 메시지에 커뮤니티 문자열을 함께 보냅니다 . 에이전트는 비밀번호를 받아 네트워크 관리 스테이션이 정보를 얻을 자격이 있는지 인증을 확인합니다 . 커뮤니티 문자열이 SNMP 패킷과 보내질 때에는 감추어지지 않으며 , 문자열은 ASCII 텍스트로 보내집니다 .

Administration Server 의 Set SNMP Master Agent Community 페이지에서 SNMP 마스터 에이전트용 커뮤니티 문자열을 구성할 수 있습니다 . 또한 특정 커뮤니티가 수행할 수 있는 SNMP 관련 작업을 정의합니다 . 또한 Administration Server 에서 이미 구성한 커뮤니티를 확인 , 편집 및 제거할 수 있습니다 .

트랩 대상 구성

SNMP 트랩은 SNMP 에이전트가 네트워크 관리 스테이션으로 송신하는 메시지입니다 . 예를 들어 SNMP 에이전트는 인터페이스의 상태가 가동 (up) 에서 중지 (down) 로 변경될 때 트랩을 송신합니다 . SNMP 에이전트는 반드시 네트워크 관리 스테이션의 주소를 알고 있어야 트랩을 보낼 위치를 알 수 있습니다 . Proxy Server 에서 SNMP 마스터 에이전트용 트랩 대상을 구성할 수 있습니다 . 또한 이미 구성한 트랩 대상을 확인 , 편집 및 제거할 수 있습니다 . Proxy Server 를 사용하여 트랩 대상을 구성하는 경우 실제로는 `CONFIG` 파일을 편집하는 것입니다 .

하위 에이전트 사용 설정

Administration Server 와 함께 제공되는 마스터 에이전트를 설치한 후에는 반드시 에이전트를 시작하기 전에 서버 인스턴스용 하위 에이전트를 사용하도록 설정해야 합니다 . 마스터 에이전트 설치에 대한 자세한 내용은 "SNMP 마스터 에이전트 설치 " (222 페이지) 를 참조하십시오 . Server Manager 를 사용하여 하위 에이전트를 사용하도록 설정할 수 있습니다 .

UNIX 나 Linux 플랫폼에서 SNMP 기능을 중지하려면 반드시 우선 하위 에이전트를 중지한 후 마스터 에이전트를 중지합니다. 마스터 에이전트를 먼저 중지시키는 경우 하위 에이전트를 중지시킬 수 없게 될 수 있습니다 . 이러한 경우 마스터 에이전트를 다시 시작하고 하위 에이전트를 중지한 후 , 마스터 에이전트를 중지시킵니다 .

SNMP 하위 에이전트를 사용하도록 설정하려면 Server Manger 의 Configure SNMP Subagent 페이지를 사용합니다. Control SNMP Subagent 페이지에서 하위 에이전트를 시작합니다 . 자세한 내용은 온라인에서 도움말의 해당 부분을 참조하십시오 .

하위 에이전트를 사용하도록 설정한 후에는 Control SNMP Subagent 페이지나 Windows 의 Services Control Panel 에서 에이전트를 중지 또는 재시작할 수 있습니다 .

참고 SNMP 구성을 변경한 경우에는 반드시 Apply Required 버튼을 눌러 SNMP 하위 에이전트가 다시 시작되도록 해야 합니다 .

SNMP 메시지 이해

GET 과 SET 은 SNMP 에 의하여 정의되는 두 가지 유형의 메시지입니다 . GET 과 SET 메시지는 NMS(Network Management Station) 가 마스터 에이전트로 보내는 메시지입니다 . 이 중 한가지 또는 둘 모두를 Administration Server 에서 사용할 수 있습니다 .

SNMP 는 PDU(protocol data unit) 의 형태로 네트워크 정보를 교환합니다 . 이 단위에는 웹 서버 등의 관리된 장치에 저장된 변수에 대한 정보가 들어 있습니다 . 이들 변수는 관리된 개체라고도 하며 , 필요한 경우 NMS 로 보고하는 값과 제목이 포함됩니다 . 서버가 NMS 로 보내는 프로토콜 데이터 단위는 " 트랩 " 이라고도 합니다 . GET, SET 및 " 트랩 " 메시지를 사용하는 방법은 다음 예에서 설명합니다 .

NMS 가 시작한 통신 NMS 는 서버에서의 정보를 요청하거나 서버의 MIB 에 저장된 변수의 값을 변경합니다 . 예 :

1. NMS 는 Administration Server 마스터 에이전트에 메시지를 보냅니다 . 메시지는 데이터에 대한 요청 (GET 메시지) 일 수 있으며 MIB 의 변수를 설정하는 지시문 (SET 메시지) 일 수 있습니다 .
2. 마스터 에이전트는 메시지를 적절한 하위 에이전트로 전달합니다 .
3. 하위 에이전트는 데이터를 수신하거나 MIB 의 변수를 변경합니다 .
4. 하위 에이전트는 데이터 또는 상태를 마스터 에이전트에 보고하고 , 마스터 에이전트는 해당 메시지 (GET 메시지) 를 다시 NMS 로 전달합니다 .
5. NMS 는 네트워크 관리 응용 프로그램을 통하여 데이터를 텍스트 또는 그래픽으로 표시합니다 .

서버가 시작한 통신 . 중요한 이벤트가 발생하면 서버 하위 에이전트는 메시지 또는 " 트랩 " 을 NMS 로 보냅니다 . 예 :

1. 하위 에이전트는 마스터 에이전트에게 서버가 중지되었음을 알립니다 .
2. 마스터 에이전트는 메시지 또는 " 트랩 " 을 보내어 NMS 에 해당 이벤트를 보고합니다 .
3. NMS 는 네트워크 관리 응용 프로그램을 통하여 정보를 텍스트 또는 그래픽으로 표시합니다 .

제 4 부

Proxy Server 관리

11 장

URL 프록시 및 라우팅

이 장에서는 프록시 서버가 요청을 처리하는 방법에 대해 설명합니다 . 또한 특정 리소스에 대해 프록시를 사용하고 프록시 서버가 URL 을 다른 URL 이나 서버로 라우팅하도록 구성하는 방법에 대해서도 설명합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

리소스에 대한 프록시 사용 설정

리소스에 대해 프록시를 사용하거나 사용하지 않도록 설정할 수 있습니다 . 리소스는 개별 URL, 공통점이 있는 URL 그룹 , 전체 프로토콜일 수 있습니다 . 프록시를 전체 서버에서 사용할지 , 또는 다양한 리소스나 템플릿 파일에서 지정한 리소스에 대해 사용할지 여부를 제어할 수 있습니다 . 즉 하나 이상의 URL 에 대한 액세스를 거부하려

면 해당 리소스에 대한 프록시를 사용하지 않도록 하면 됩니다 . 이 기능은 리소스에 대한 모든 액세스를 거부하거나 허용하는 방식으로 활용될 수 있습니다 . 또한 URL 필터를 사용하여 액세스를 허용 또는 거부할 수도 있습니다 . 로그 파일에 대한 더 자세한 내용은 "URL 필터링 " (304 페이지) 을 참조하십시오 .

리소스에 대해 프록시를 사용하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Routing 탭을 누릅니다 .
2. Enable/Disable Proxying 링크를 누릅니다. Enable/Disable Proxying 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 눌러 정규식을 입력한 후 OK 를 누릅니다 .
4. 지정한 리소스에 대해 기본 설정을 선택할 수 있습니다. 리소스에 대해 프록시를 선택하지 않거나 (프록시 사용 안 함) 사용하도록 선택할 수 있습니다 . 선택할 수 있는 옵션은 다음과 같습니다 .

- **Use Default Setting Derived From A More General Resource.** 이 리소스를 포함한 더 일반적인 리소스에 대한 설정이 사용됩니다 .
- **Do Not Proxy This Resource.** 프록시를 통하여 이 리소스에 도달할 수 없도록 합니다 .
- **Enable Proxying Of This Resource.** 다른 보안 및 인증 확인을 통과한 클라이언트에 한해 이 리소스에 액세스할 수 있도록 프록시를 구성합니다. 리소스에 대해 프록시를 사용하도록 하면 *모든 메소드를 사용할 수 있습니다* . 해당 리소스에 대해 GET, HEAD, INDEX, POST 등의 읽기 메소드 , SSL 터널링용 CONNECT, 그리고 PUT, MKDIR, RMDIR, MOVE, DELETE 등의 쓰기 메소드를 모두 사용합니다 . 다른 보안 확인이 없다면 클라이언트는 모두 읽기와 쓰기 액세스 권한을 갖게 됩니다 .

5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

다른 프록시를 통한 라우팅

Set Routing Preferences 페이지는 프록시 서버가 파생된 기본 구성이나 직접 연결을 사용하는 특정 리소스를 라우팅하도록 구성하는 데 사용됩니다 . 또는 프록시 배열 , ICP 환경 , 다른 프록시 서버나 SOCKS 서버를 통하여 라우팅하도록 구성할 수도 있습니다 .

리소스에 대한 라우팅 구성

리소스에 대해 라우팅을 구성하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Routing 탭을 누릅니다 .
2. Set Routing Preferences 링크를 누릅니다 . Set Routing Preferences 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 눌러 정규식을 입력한 후 OK 를 누릅니다 .
4. 구성하는 리소스에 대한 라우팅 유형에 해당하는 선택 버튼을 누릅니다 . 옵션은 다음과 같습니다 .

- **Derived Default Configuration.** 프록시 서버는 더 일반적인 템플릿 (더 짧고 정규식과 일치하는 템플릿) 을 사용하여 원격 서버와 다른 프록시 중 어떤 것을 사용할지 결정합니다 . 예를 들어 프록시가 모든 `http://.*` 요청을 다른 Proxy Server 로 라우팅하고 모든 `http://www.*` 요청을 원격 서버로 라우팅하는 경우 , `http://www.example.*` 요청에 대해 유도된 기본 구성 라우팅을 생성할 수 있습니다 . 이렇게 하면 이 요청은 `http://www.*` 템플릿용 설정이 적용되어 원격 서버로 직접 이동하게 됩니다 .
- **Direct Connections.** 요청이 프록시를 통하지 않고 항상 원격 서버로 직접 이동하도록 합니다 .
- **Route Through A SOCKS Server.** 특정 리소스에 대한 요청이 SOCKS 서버를 통해 라우팅됩니다 . 이 옵션을 선택한 경우 프록시 서버가 통과하여 라우팅할 SOCKS 서버의 이름 (또는 IP 주소) 과 포트 번호를 지정해야 합니다 .
- **Route Through.** 프록시 배열 , ICP 환경 , 상위 배열 및 / 또는 프록시 서버를 통해 라우팅할지 여부를 지정합니다 . 라우팅 방법을 여러 개 선택하면 프록시는 양식에 표시된 계층을 따라가게 됩니다 (즉 , 프록시 배열 , 재지정 , ICP, 상위 배열 , ICP, 다른 프록시). 프록시 서버에 대한 자세한 내용은 " 프록시 서버 체인 " (236 페이지) 을 참조하십시오 .

  SOCKS 서버를 통한 라우팅에 대한 자세한 내용은 "SOCKS 서버를 통한 라우팅 " (236 페이지) 을 참조하십시오 . 프록시 배열 , 상위 배열 또는 ICP 환경을 통한 라우팅에 관한 자세한 내용은 제 12 장 , 251 페이지의 " 캐시 " 를 참조하십시오 .

참고 기본이 아닌 포트 (443 이외) 에서 연결 요청 라우팅을 사용하려면 `obj.conf` 파일의 `ppath` 매개변수를 `connect://.*` 로 변경해야 합니다 .

5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

프록시 서버 체인

일부 리소스에 대해 프록시가 원격 서버에 액세스하지 않고 다른 프록시에 액세스하도록 설정할 수 있습니다 . 즉 프록시를 하나의 체인으로 연결할 수 있습니다 . 체인은 방화벽 뒤의 여러 프록시를 정리하는 좋은 방법입니다 . 체인을 사용하면 계층적 캐싱을 구축할 수 있습니다 .

다른 프록시 서버를 통해 라우팅하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Routing 탭을 누릅니다 .
2. Set Routing Preferences 링크를 누릅니다 . Set Routing Preferences 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 눌러 정규식을 입력한 후 OK 를 누릅니다 .
4. 이 페이지의 Routing Through Another Proxy 부분에서 Route Through 옵션을 선택합니다 .
5. Another Proxy 확인란을 선택합니다 .
6. Another Proxy 필드에서 라우팅에 사용할 프록시 서버의 IP 주소나 이름을 입력합니다 .
7. OK 를 누릅니다 .
8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

SOCKS 서버를 통한 라우팅

네트워크에서 실행 중인 원격 SOCKS 서버가 이미 있을 경우 프록시가 특정 리소스에 대해 이 서버에 연결하도록 설정할 수 있습니다 .

SOCKS 서버를 통해 라우팅하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Routing 탭을 누릅니다 .

2. Set Routing Preferences 링크를 누릅니다 . Set Routing Preferences 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 눌러 정규식을 입력한 후 OK 를 누릅니다 .
4. 이 페이지의 Routing Through Another Proxy 부분에서 Route Through 옵션을 선택합니다 .
5. Route Through SOCKS Server 옵션을 선택합니다 .
6. 프록시 서버가 라우팅에 사용할 SOCKS 서버의 이름 (또는 IP 주소) 과 포트 번호를 지정합니다 .
7. OK 를 누릅니다 .

참고 SOCKS 서버를 통한 라우팅을 사용하도록 설정한 후에는 SOCKS v5 Routing 페이지에서 프록시 라우팅을 만들어야 합니다 . 프록시 라우팅은 프록시가 라우팅에 사용하는 SOCKS 서버를 통해 액세스할 수 있는 IP 주소를 식별합니다 . 또한 SOCKS 서버가 호스트에 직접 연결할지 여부를 지정합니다 .

8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

서버에 클라이언트 IP 주소 전달

Forward Client Credentials 페이지는 프록시가 클라이언트 인증서를 원격 서버로 전송하도록 구성하는 데 사용됩니다 .

프록시가 클라이언트 IP 주소를 전송하도록 구성하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Forward Client Credentials 링크를 누릅니다 . Forward Client Credentials 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 눌러 정규식을 입력한 후 OK 를 누릅니다 .
4. 필요에 따라 다음 옵션을 변경합니다 .

- **Client IP Addressing Forwarding.** 문서에 대한 요청을 만드는 경우 Proxy Server 에서 원격 서버로 클라이언트의 IP 주소를 전송하지 않습니다 . 대신 프록시가 클라이언트 역할을 하여 자신의 IP 주소를 원격 서버로 전송합니다 . 그러나 다음과 같은 경우 클라이언트의 IP 주소를 전달할 수도 있습니다 .
  - 프록시가 내부 프록시의 체인에 속해 있는 경우
  - 클라이언트에서 액세스해야 하는 서버가 해당 클라이언트의 IP 주소를 알아야 하는 경우 템플릿을 사용하여 특정 서버에만 클라이언트의 IP 주소를 전송할 수 있습니다 .

  다음 옵션 중 하나를 선택하여 클라이언트의 IP 주소를 전송하도록 프록시를 구성합니다 .
  - **Default.** Proxy Server 가 클라이언트의 IP 주소를 전달하도록 합니다 .
  - **Blocked.** 프록시가 클라이언트의 IP 주소를 전달하지 않도록 합니다 .
  - **Enabled Using HTTP Header.** 프록시가 IP 주소를 전달하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 . 기본 HTTP 헤더의 이름은 `Client-ip` 이지만 어떤 헤더를 선택해도 IP 주소를 전송할 수 있습니다 .
- **Client Proxy Authentication Forwarding.** 다음 옵션 중 하나를 선택하여 클라이언트의 인증 세부 내용을 전송하도록 프록시를 구성합니다 .
  - **Default.** Proxy Server 가 클라이언트의 인증 세부 내용을 전달하도록 합니다 .
  - **Blocked.** 프록시가 클라이언트의 인증 세부 내용을 전달하지 않도록 합니다 .
  - **Enabled Using HTTP Header.** 프록시가 인증 세부 내용을 전달하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 .
- **Client Cipher Forwarding**. 다음 옵션 중 하나를 선택하여 클라이언트 SSL/TLS 암호 제품군의 이름을 원격 서버로 전송하도록 프록시를 구성합니다 .
  - **Default.** Proxy Server 가 클라이언트 SSL/TLS 암호 제품군의 이름을 원격 서버로 전달하도록 합니다 .
  - **Blocked.** 프록시가 클라이언트 SSL/TLS 암호 제품군의 이름을 원격 서버로 전달하지 않도록 합니다 .
  - **Enabled Using HTTP Header.** 프록시가 클라이언트 SSL/TLS 암호 제품군의 이름을 원격 서버로 전달하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 . 기본 HTTP 헤더는 `Proxy-cipher` 이지만 어떤 헤더를 선택해도 클라이언트 SSL/TLS 암호 제품군의 이름을 전송할 수 있습니다 .

- **Client Keysize Forwarding.** 다음 옵션 중 하나를 선택하여 클라이언트 SSL/TLS 키의 크기를 원격 서버로 전송하도록 프록시를 구성합니다 .
  - **Default.** Proxy Server가 클라이언트 SSL/TLS 키의 크기를 원격 서버로 전달하도록 합니다 .
  - **Blocked.** 프록시가 클라이언트 SSL/TLS 키의 크기를 원격 서버로 전달하지 않도록 합니다 .
  - **Enabled Using HTTP Header.** 프록시가 클라이언트 SSL/TLS 키의 크기를 원격 서버로 전달하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 . 기본 HTTP 헤더는 `Proxy-keysize` 이지만 어떤 헤더를 선택해도 클라이언트 SSL/TLS 키의 크기를 전송할 수 있습니다 .
- **Client Secret Keysize Forwarding.** 다음 옵션 중 하나를 선택하여 클라이언트 SSL/TLS 비밀 키의 크기를 원격 서버로 전송하도록 프록시를 구성합니다 .
  - **Default.** Proxy Server 가 클라이언트 SSL/TLS 비밀 키의 크기를 원격 서버로 전달하도록 합니다 .
  - **Blocked.** 프록시가 클라이언트 SSL/TLS 비밀 키의 크기를 원격 서버로 전달하지 않도록 합니다 .
  - **Enabled Using HTTP Header.** 프록시가 클라이언트 SSL/TLS 비밀 키의 크기를 원격 서버로 전달하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 . 기본 HTTP 헤더는 `Proxy-secret-keysize` 이지만 어떤 헤더를 선택해도 클라이언트 SSL/TLS 보안 키의 크기를 전송할 수 있습니다 .
- **Client SSL Session ID Forwarding.** 다음 옵션 중 하나를 선택하여 클라이언트의 SSL/TLS 세션 ID 를 원격 서버로 전송하도록 프록시를 구성합니다 .
  - **Default.** Proxy Server가 클라이언트의 SSL/TLS 세션 ID를 원격 서버로 전달하도록 합니다 .
  - **Blocked.** Proxy Server 가 클라이언트의 SSL/TLS 세션 ID 를 원격 서버로 전달하지 않도록 합니다 .
  - **Enabled Using HTTP Header.** ㅌㄴ 프록시가 클라이언트의 SSL/TLS 세션 ID 를 원격 서버로 전달하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 . 기본 HTTP 헤더는 `Proxy-ssl-id` 이지만 어떤 헤더를 선택해도 클라이언트의 SSL/TLS 세션 ID 를 전송할 수 있습니다 .
- **Client Issuer DN Forwarding.** 다음 옵션 중 하나를 선택하여 클라이언트 SSL/TLS 인증서 발행 기관의 고유한 이름을 원격 서버로 전송하도록 프록시를 구성합니다 .
  - **Default.** Proxy Server 가 클라이언트 SSL/TLS 인증서 발행 기관의 고유한 이름을 원격 서버로 전달하도록 합니다 .

- **Blocked.** Proxy Server 가 클라이언트 SSL/TLS 인증서 발행 기관의 고유한 이름을 원격 서버로 전달하지 않도록 합니다 .
- **Enabled Using HTTP Header.** 프록시가 클라이언트 SSL/TLS 인증서 발행 기관의 고유한 이름을 원격 서버로 전달하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 . 기본 HTTP 헤더의 이름은 `Proxy-issuer-dn` 이지만 어떤 헤더를 선택해도 클라이언트 SSL/TLS 인증서 발행 기관의 고유한 이름을 전송할 수 있습니다 .

❍ **Client User DN Forwarding.** 다음 옵션 중 하나를 선택하여 클라이언트 SSL/TLS 인증서 주체의 고유한 이름을 원격 서버로 전송하도록 프록시를 구성합니다 .

- **Default.** Proxy Server 가 클라이언트 SSL/TLS 인증서 주체의 고유한 이름을 원격 서버로 전달하도록 합니다 .
- **Blocked.** Proxy Server 가 클라이언트 SSL/TLS 인증서 주체의 고유한 이름을 원격 서버로 전달하지 않도록 합니다 .
- **Enabled Using HTTP Header.** 프록시가 클라이언트 SSL/TLS 인증서 주체의 고유한 이름을 원격 서버로 전달하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 . 기본 HTTP 헤더는 `Proxy-user-dn` 이지만 어떤 헤더를 선택해도 클라이언트 SSL/TLS 인증서 주체의 고유한 이름을 전송할 수 있습니다 .

❍ **Client SSL/TLS Certificate Forwarding.** 다음 옵션 중 하나를 선택하여 클라이언트의 SSL/TLS 인증서를 원격 서버로 전송하도록 프록시를 구성합니다 .

- **Default.** Proxy Server 가 클라이언트의 SSL/TLS 인증서를 원격 서버로 전달하도록 합니다 .
- **Blocked.** Proxy Server가 클라이언트의 SSL/TLS 인증서를 원격 서버로 전달하지 않도록 합니다 .
- **Enabled Using HTTP Header.** 프록시가 클라이언트의 SSL/TLS 인증서를 원격 서버로 전달하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 . 기본 HTTP 헤더는 `Proxy-auth-cert` 이지만 어떤 헤더를 선택해도 클라이언트의 SSL/TLS 인증서를 전송할 수 있습니다 .

❍ **Client Cache Information Forwarding.** 다음 옵션 중 하나를 선택하여 로컬 캐시 적중 횟수에 대한 정보를 원격 서버로 전송하도록 프록시를 구성합니다 .

- **Default.** Proxy Server 가 로컬 캐시 적중 횟수에 대한 정보를 원격 서버로 전달하도록 합니다 .
- **Blocked.** Proxy Server 가 로컬 캐시 적중 횟수에 대한 정보를 원격 서버로 전달하지 못하도록 합니다 .

- **Enabled Using HTTP Header.** 프록시가 로컬 캐시 적중 횟수에 대한 정보를 원격 서버로 전달하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 . 기본 HTTP 헤더의 이름은 `Cache-info` 이지만 어떤 헤더를 선택해도 로컬 캐시 적중 횟수에 대한 정보를 전송할 수 있습니다 .

❍ **Set Basic Authentication Credentials.** 다음 옵션 중 하나를 선택하여 HTTP 요청을 전송하도록 프록시를 구성합니다 .

- **User.** 인증할 사용자를 지정합니다 .
- **Password.** 사용자 비밀 번호를 지정합니다 .
- **Using HTTP Header.** 프록시가 인증서를 통신하는 데 사용할 HTTP 헤더를 지정할 수 있습니다 .

5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

클라이언트의 IP 주소 확인 허용

네트워크 보안을 유지하기 위해 클라이언트에 특정 IP 주소에만 액세스하도록 제한하는 기능을 사용할 수도 있습니다 . 클라이언트가 이 기능을 사용할 때는 프록시 서버가 Java IP Address 확인을 지원합니다 . 이러한 지원을 통해 클라이언트가 리소스를 검색하는 데 사용하는 IP 주소에 대한 프록시 서버를 쿼리할 수 있습니다 . 이 기능을 사용하면 클라이언트는 원본 서버의 IP 주소를 전송하도록 프록시 서버에 요청할 수 있으며 , 프록시 서버는 헤더에 IP 주소를 첨부하게 됩니다 . 클라이언트는 원본 서버의 IP 주소를 파악하면 이후 연결에서 동일한 IP 주소가 사용되도록 명시적으로 지정할 수 있습니다 .

Java IP 주소를 확인하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Check Java IP Address 링크를 누릅니다 . Check Java IP Address 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 눌러 정규식을 입력한 후 OK 를 누릅니다 .

4. 라디오 버튼을 눌러 Java IP 주소 확인을 사용 또는 사용하지 않거나, 기본 구성을 사용하도록 선택합니다 .

참고 기본 옵션은 더 일반적인 템플릿 (더 짧고 정규식과 일치하는 템플릿) 에서 가져온 기본 구성을 사용하여 Java IP 주소 확인 기능의 사용 여부를 결정합니다 .

5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

클라이언트 자동 구성

프록시 서버가 많은 클라이언트를 지원할 경우 클라이언트 자동 구성 파일을 사용하여 모든 브라우저 클라이언트를 구성할 수 있습니다 . 자동 구성 파일에는 Navigator 가 여러 URL 에 액세스할 때 사용하는 프록시가 있으면 이를 결정하는 JavaScript 함수가 들어 있습니다 . 이 기능에 대한 자세한 내용은 제 17 장 , 349 페이지의 " 클라이언트 자동 구성 파일 사용 " 을 참조하십시오 .

네트워크 연결 모드 설정

네트워크에서 프록시 서버 컴퓨터로 연결하거나 연결을 해제할 수 있습니다 . 이 기능을 사용하여 데모용으로 사용할 수 있는 휴대용 컴퓨터에 프록시를 편리하게 설치할 수 있습니다 .

프록시와 네트워크의 연결을 해제하면 문서는 직접 캐시로 반환되고 프록시는 최신 여부 확인을 수행할 수 없으므로 문서가 매우 신속하게 검색됩니다 (문서가 최신이 아닐 수 있음). 캐싱에 대한 자세한 내용은 제 12 장 , 251 페이지의 " 캐시 " 를 참조하십시오 .

또한 사용자가 네트워크에 연결되어 있지 않은 경우 프록시 서버는 네트워크 부재를 인식하고 원격 서버로 접속을 시도하지 않기 때문에 연결 대기가 없습니다 . 네트워크가 다운된 상태에서 프록시 서버 컴퓨터가 실행되고 있는 경우 이러한 네트워크 부재 설정을 사용할 수 있습니다 .

참고 네트워크에 연결되지 않은 프록시를 실행하면 때때로 캐시의 지난 데이터에 액세스하는 경우가 있을 수 있습니다 . 또한 네트워크 없이 실행하면 프록시 보안 기능이 필요하지 않습니다 .

Proxy Server 는 4 가지 연결 모드를 제공합니다 .

- Default 모드는 일치하는 가장 일반적인 개체의 구성에서 파생된 모드입니다 .
- Normal 모드는 프록시 정상 작동 모드입니다 . 프록시는 문서가 캐시에 없으면 컨텐트 서버에서 해당 문서를 가져옵니다 . 문서가 캐시에 있으면 컨텐트 서버와 대조하여 문서가 최신 상태인지 여부를 확인합니다 . 캐시 파일이 변경된 경우 현재 문서로 교체됩니다 .
- Fast-demo 모드는 네트워크를 사용할 수 있는 경우 데모를 원활하게 제공하기 위한 모드입니다 . 캐시에서 문서를 찾으면 컨텐트 서버로 연결하지 않습니다 . 따라서 문서의 변경 여부도 확인하지 않습니다 . 이 모드에는 컨텐트 서버의 응답을 대기하면서 발생되는 지연 현상이 없습니다 . 캐시에 문서가 없으면 컨텐트 서버에서 가져온 다음 캐시에 저장합니다 . Fast-demo mode 는 Normal mode 보다 지연 현상이 덜하지만 일단 캐시에 문서가 있으면 이 문서의 최신 여부는 확인하지 않기 때문에 때때로 지난 데이터를 반환하는 경우가 있습니다 .
- No-network 모드는 휴대용 컴퓨터가 네트워크에 연결되지 않은 동안 사용하는 모드입니다 . 프록시는 캐시에 문서가 있으면 반환하고 , 없으면 오류를 반환합니다 . 프록시는 어떤 경우에도 컨텐트 서버로 연결을 시도하지 않으며 , 따라서 존재하지 않는 연결을 제한 시간까지 대기하는 일이 발생하지 않습니다 .

프록시 서버 실행 모드를 변경하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Set Connectivity Mode 링크를 누릅니다 . Set Connectivity Mode 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 눌러 정규식을 입력한 후 OK 를 누릅니다 .
4. 원하는 모드를 선택합니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

기본 FTP 전송 모드 변경

FTP 에는 FTP 서버와 클라이언트 (클라이언트 역할을 하는 프록시) 간 데이터 연결을 설정하는 두 가지 방법이 있습니다 . 이 두 모드를 PASV 와 PORT 모드 FTP 라고 합니다 .

- **Passive Mode (PASV).** 프록시 서버에서 데이터 연결을 시작하고 , FTP 서버가 이 연결을 수락합니다 . 이 전송 모드는 외부에서 내부로의 연결을 받아들이지 않기 때문에 프록시 서버를 실행하는 사이트에서는 이 방법이 더 안전합니다 .
- **Active Mode (PORT).** 원격 FTP 서버가 데이터 연결을 시작하고 , Proxy Server 가 이 연결을 수락합니다 . 프록시 서버가 방화벽 내에 있으면 방화벽이 FTP 서버에서 들어오는 FTP 데이터 연결을 차단할 수 있으며 , 이 경우 PORT 모드가 작동하지 않을 수 있습니다 .

일부 FTP 사이트는 방화벽을 이용하는데 , 이 경우 Proxy Server 에서 PASV 모드가 동작하지 않습니다 . 따라서 프록시 서버가 PORT mode FTP 를 사용하도록 구성할 수 있습니다 . 서버 전체에 PORT 모드를 사용하거나 , 아니면 특정 FTP 서버에 대해서만 사용할 수 있습니다 .

참고 PASV 모드를 사용 중인 경우에도 원격 FTP 서버가 PASV 모드를 지원하지 않으면 프록시 서버는 PORT 모드를 사용합니다 .

프록시 서버가 방화벽 뒤에 있어 PORT 모드 FTP 가 작동하지 않는 경우에는 PORT 모드를 사용할 수 없습니다 . 리소스에 대해 기본값을 선택하면 프록시 서버는 보다 일반적인 리소스 모드를 사용합니다 . 아무것도 지정하지 않으면 PASV 모드가 사용됩니다 .

FTP 모드를 설정하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Set FTP Mode 링크를 누릅니다 . Set FTP Mode 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 눌러 정규식을 입력한 후 OK 를 누릅니다 .
4. FTP 전송 모드를 선택합니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

SOCKS 이름 서버 IP 주소 지정

프록시가 SOCKS 서버를 통하여 내부에서 외부로의 연결을 설정하도록 구성된 경우 , SOCKS 에 사용될 네임 서버에 대한 IP 주소를 명확히 지정해야 합니다 .

방화벽 내의 내부 DNS 서비스를 제외한 DNS 서버로 외부 호스트 이름을 확인하는 경우에는 네임 서버의 IP 주소를 지정해야 합니다 .

SOCKS 이름 서버 IP 주소를 지정하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Set SOCKS Name Server 링크를 누릅니다 . Set SOCKS Name Server 페이지가 표시됩니다 .
3. 텍스트 필드에 DNS 이름 서버의 IP 주소를 입력합니다 .
4. OK 를 누릅니다 .

참고 SOCKS 이름 서버의 IP 주소를 지정하는 기능은 SOCKS_NS 환경 변수를 통해서 한번만 액세스할 수 있습니다 . SOCKS_NS 환경 변수를 설정하고 SOCKS Name Server Setting 양식을 사용하여 네임 서버의 IP 주소를 지정하면 , 프록시는 환경 변수가 아닌 양식에 지정된 IP 주소를 사용하게 됩니다 .

5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

HTTP 요청 로드 밸런싱 구성

Configure HTTP Request Load Balancing 페이지는 지정된 원본 서버로 부하를 분산시키는 데 사용됩니다 .

HTTP 요청 로드 밸런싱을 구성하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Configure HTTP Request Load Balancing 페이지 링크를 누릅니다 . Configure HTTP Request Load Balancing 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 눌러 정규식을 입력한 후 OK 를 누릅니다 .

4. Server 필드에 원본 서버의 URL을 지정합니다. 서버 매개 변수가 여러 개 주어지면 Proxy Server 는 부하를 지정된 원본 서버로 분산합니다 .
5. 요청에 쿠키가 있는 경우 Sticky Cookie 필드에 이 쿠키의 이름을 지정하면 이후의 요청들이 해당 원본 서버에 고착됩니다 . 기본값은 `JSESSIONID` 입니다 .
6. 라우팅 정보를 확인하려면 Sticky Parameter 필드의 URI 매개 변수의 이름을 지정합니다 . 요청 URI 에 URI 매개 변수가 존재하고 그 값에 콜론과 라우팅 ID 가 포함되어 있는 경우 , 해당 요청은 라우팅 ID 에 의하여 확인되는 원본 서버에 " 고착 " 됩니다 . 기본값은 `jsessionid` 입니다 .
7. 원본 서버에 라우팅 ID 를 전달하는 데 사용되는 HTTP 요청 헤더의 Route Header 필드에 이름을 지정합니다 . 기본값은 `proxy-jroute` 입니다 .
8. Proxy Server 가 응답에서 고정 쿠키를 발견한 경우 생성하는 쿠키의 이름을 Route Cookie 필드에 지정합니다 . 기본값은 `JROUTE` 입니다 .
9. 적절한 Rewrite Host 옵션을 선택하여 호스트 HTTP 요청 헤더가 서버 매개 변수에서 지정한 호스트와 일치하도록 다시 작성된 것인지 여부를 표시합니다 .
10. 적절한 Rewrite Location 옵션을 선택하여 서버 매개 변수와 일치하는 Location HTTP 응답 헤더를 다시 작성해야 하는지 여부를 표시합니다 .
11. 적절한 Rewrite Content Location 옵션을 선택하여 서버 매개 변수와 일치하는 Content-Location HTTP 응답 헤더를 다시 작성해야 하는지 여부를 표시합니다 .
12. 확인란을 선택하여 서버 매개 변수와 일치하는 *headername* HTTP 응답 헤더를 다시 작성해야 하는지 여부를 표시합니다 . 여기서 *headername* 은 사용자 정의 헤더 이름입니다 . Headername 필드에 헤더 이름을 지정합니다 .
13. OK 를 누릅니다 .
14. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
15. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

URL 및 URL 매핑 관리

Server Manager 를 통해 URL 을 다른 서버에 매핑할 수 있습니다 . 이를 간혹 미러 서버라고도 합니다 . 클라이언트가 미러링된 URL 이 있는 프록시에 액세스하면 프록시는 URL 에서 지정한 서버가 아닌 미러링된 서버에서 요청한 문서를 검색합니다 . 클라이언트는 이 요청이 다른 서버로 전달된다는 것을 인식할 수 없습니다 . URL 을 재전송할 수도 있으며 이 경우에는 프록시가 문서가 아닌 재전송된 URL 만 클라이언트로 반환하므로 클라이언트가 새 문서를 요청할 수 있습니다 . 매핑을 통해 PAC 및 PAT 매핑에서처럼 URL 을 파일에 매핑할 수도 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- URL 매핑 만들기
- 기존 URL 매핑 확인 , 편집 또는 제거
- URL 재지정

URL 매핑 만들기

URL 을 매핑하려면 URL 접두사와 매핑 위치를 지정합니다 . 다음 절에서는 다양한 URL 매핑 유형에 대해 설명합니다 . 4 가지 유형의 URL 매핑을 만들 수 있습니다 .

- 정상 매핑은 URL 접두사를 다른 URL 접두사와 매핑합니다 . 예를 들어 `http://www.example.com` 으로 시작하는 요청을 받으면 특정 URL 로 이동하도록 프록시를 구성할 수 있습니다 .
- 역방향 매핑은 재전송된 URL 접두사를 다른 URL 접두사와 매핑합니다 . 이들 옵션은 내부 서버에서 문서가 아닌 재지정된 응답을 프록시로 전송하는 경우 역방향 프록시와 함께 사용됩니다 . 자세한 내용은 제 14 장 , 313 페이지의 " 역방향 프록시 사용 " 을 참조하십시오 .
- 정규식은 해당 식에 일치하는 모든 URL 을 하나의 URL 로 매핑합니다 . 예를 들어 , *job.* 와 일치하는 모든 URL 을 특정 URL(프록시 서버가 특정 URL 에 대한 접속을 허용하지 않는 이유를 설명하는 내용일 수 있음) 로 매핑합니다 .
- 클라이언트 자동 구성은 URL 을 Proxy Server 에 저장되어 있는 특정 `.pac` 파일로 매핑합니다 . 자동 구성 파일에 대한 자세한 내용은 제 17 장 , 349 페이지의 " 클라이언트 자동 구성 파일 사용 " 을 참조하십시오 .
- PAT(Proxy Array Table) 는 URL 을 Proxy Server 에 저장되어 있는 특정 `.pat` 파일로 매핑합니다 . 이 유형의 매핑은 마스터 프록시에서만 만들어야 합니다 . PAT 파일 및 프록시 배열 사용에 대한 자세한 내용은 " 프록시 배열을 통한 라우팅 " (288 페이지) 을 참조하십시오 .

URL 에 액세스하는 클라이언트가 같은 서버의 다른 위치나 다른 서버로 전달됩니다 . 이것은 리소스가 옮겨진 경우나 , 끝에 오는 슬래시 없이 디렉토리에 액세스한 경우 관련 링크의 무결성을 유지하는 데 유용합니다 .

예를 들어 로드가 상당히 심한 웹 서버 hi.load.com 을 다른 서버인 mirror.load.com 으로 미러링하고자 할 경우 , 프록시 서버가 hi.load.com 컴퓨터로 이동하는 URL 에 대해 mirror.load.com 컴퓨터를 사용하도록 구성할 수 있습니다 .

소스 URL 접두사는 이스케이프하면 안 됩니다 . 대상 (미러) URL 에서는 HTTP 요청에 적합하지 않는 문자만 이스케이프해야 합니다 .

주의 접두사에는 끝에 오는 슬래시를 사용하지 않습니다 .

URL 매핑을 만들려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 URLs 탭을 누릅니다 .
2. Create Mapping 링크를 누릅니다 . Create Mapping 페이지가 나타납니다 .
3. 만들 매핑의 유형을 선택합니다 .
   - **Regular Mappings**. URL 접두사를 다른 URL 접두사에 매핑합니다 . 예를 들어 `http://www.example.com` 으로 시작하는 요청을 받으면 특정 URL 로 이동하도록 프록시를 구성할 수 있습니다 . 이 옵션을 선택하면 페이지 아래 부분에 다음 옵션이 표시됩니다 .
     - **Rewrite Host.** 적절한 옵션을 선택하여 호스트 HTTP 헤더가 `to` 매개 변수에서 지정한 호스트와 일치하도록 다시 작성된 것인지 여부를 표시합니다 .
   - **Reverse Mappings.** 재지정된 URL 접두사를 다른 URL 접두사에 매핑힙니다 . 이들 옵션은 내부 서버에서 문서가 아닌 재지정된 응답을 프록시로 전송하는 경우 역방향 프록시와 함께 사용됩니다 . 역방향 프록시에 대한 자세한 내용은 제 14 장 , 313 페이지의 " 역방향 프록시 사용 " 을 참조하십시오 . 이 옵션을 선택하면 페이지 아래 부분에 다음 옵션이 표시됩니다 .
     - **Rewrite Location.** 적절한 옵션을 선택하여 Location HTTP 응답 헤더를 다시 작성해야 하는지 여부를 표시합니다 .
     - **Rewrite Content Location.** 적절한 옵션을 선택하여 Content-location HTTP 응답 헤더를 다시 작성해야 하는지 여부를 표시합니다 .
     - **Rewrite Headername.** 확인란을 선택하여 *headername* HTTP 응답 헤더를 다시 작성해야 하는지 여부를 표시합니다 . 여기서 *headername* 은 사용자 정의 헤더 이름입니다 .
   - **Regular Expressions.** 이 정규식에 일치하는 모든 URL 을 하나의 URL 로 매핑합니다 . 정규식에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 를 참조하십시오 .
   - **Client Autoconfiguration.** URL 을 Proxy Server 에 저장되어 있는 특정 .pac 파일로 매핑합니다 . 자동 구성 파일에 대한 자세한 내용은 제 17 장 , 349 페이지의 " 클라이언트 자동 구성 파일 사용 " 을 참조하십시오 .
   - **Proxy Array Table (PAT).** URL 을 Proxy Server 에 저장되어 있는 특정 `.pat` 파일로 매핑합니다 . 이 유형의 매핑은 마스터 프록시에서만 만들어야 합니다 . PAT 파일 및 프록시 배열에 대한 자세한 내용은 제 12 장 , 251 페이지의 " 캐시 " 의 "Routing through Proxy Arrays" 부분을 참조하십시오 .

4. 매핑 소스 접두사를 입력합니다 . 정규식 및 역방향 매핑에서 대체할 URL 부분입니다 .

   정규식 매핑에서 URL 접두사는 일치하는 모든 URL 에 대한 정규식이어야 합니다 . 매핑에 대한 템플릿을 선택한 경우 이 정규식은 템플릿의 정규식에 포함된 URL 에 대해서만 동작합니다 .

   클라이언트 자동 구성 매핑과 배열 테이블 매핑에서 URL 접두사는 클라이언트가 액세스하는 전체 URL 이어야 합니다 .

5. 매핑 대상을 입력합니다 .

   클라이언트 자동 구성 및 프록시 배열 테이블을 제외한 모든 매핑 유형에서 이 값은 매핑하려는 전체 URL 이어야 합니다 . 클라이언트 자동 구성 매핑에서 이 값은 프록시 서버의 하드 디스크에 저장되어 있는 .pac 파일의 절대 경로여야 합니다 . 프록시 배열 테이블 매핑에서 이 값은 마스터 프록시의 로컬 디스크에 저장되어 있는 .pat 파일의 절대 경로여야 합니다 .

6. 드롭다운 목록에서 템플릿 이름을 선택하거나 , 템플릿을 적용하지 않으려면 NONE 값을 그대로 둡니다 .

7. OK 를 눌러 매핑을 만듭니다 .

8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .

9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

기존 URL 매핑 확인 , 편집 또는 제거

기존 매핑을 변경하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 URLs 탭을 누릅니다 .

2. View/Edit Mappings 링크를 누릅니다 . View/Edit Mappings 페이지가 표시됩니다 .

3. 매핑을 편집하려면 해당 매핑 옆에 표시되는 Edit 링크를 누릅니다 . 해당 매핑이 적용되는 접두사 , 매핑된 URL 및 템플릿을 편집할 수 있습니다 . OK 를 눌러 변경 사항을 확인합니다 .

4. 매핑을 제거하려면 제거할 매핑을 누르고 해당 매핑 옆의 Remove 링크를 누릅니다 .

5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .

6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

URL 재지정

프록시 서버가 문서를 가져와 반환하는 대신 재지정된 URL 을 클라이언트에 반환하도록 구성할 수 있습니다 . 재지정하면 클라이언트는 원래 요청한 URL 이 다른 URL 로 재지정되었음을 인식합니다 . 보통 클라이언트는 이 경우 즉시 재지정된 URL 을 요청합니다 . Netscape Navigator 는 재지정된 URL 을 자동으로 요청하므로 사용자가 문서를 다시 명시적으로 요청할 필요가 없습니다 .

URL 재지정은 어느 부분에 대한 액세스를 거부하는 경우 유용합니다 . 액세스가 거부된 이유를 재지정한 URL 에서 사용자에게 설명할 수 있습니다 .

하나 이상의 URL 을 재지정하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 URLs 탭을 누릅니다 .
2. Redirect URLs 링크를 누릅니다 . Redirect URLs 페이지가 표시됩니다 .
3. URL 접두사인 소스 URL 을 입력합니다 .
4. 재지정할 URL을 입력합니다. 이 URL은 URL 접두사이거나 고정 URL일 수 있습니다 .

   URL 재지정에 URL 접두사 사용을 선택한 경우 , URL 접두사 필드 옆의 선택 버튼을 선택한 다음 URL 접두사를 입력합니다 . 고정 URL 사용을 선택한 경우 Fixed URL 필드 옆의 선택 버튼을 선택한 다음 고정 URL 을 입력합니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

12 장

캐시

이 장에서는 Sun Java™ System Web Proxy Server 가 문서를 캐시하는 방법을 설명합니다 . 또한 온라인 페이지를 사용하여 캐시를 구성하는 방법에 대해서도 설명합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

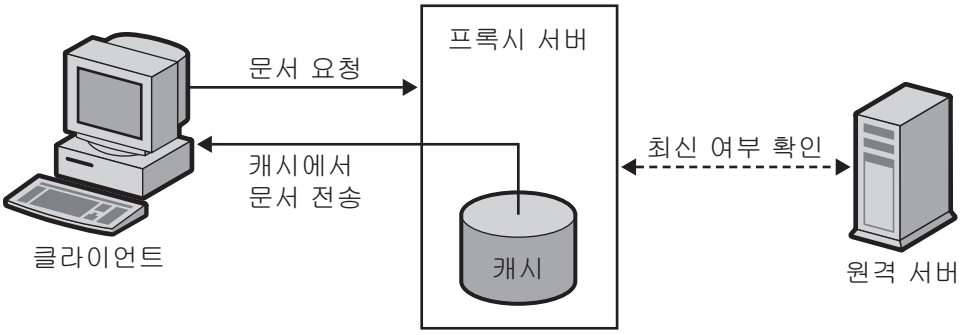
캐시 동작 원리

캐시는 네트워크 트래픽을 줄여 주며 원격 서버로 직접 이동하지 않고 프록시 서버를 사용하는 클라이언트에게 빠른 응답 시간을 제공합니다 .

클라이언트가 프록시 서버에 웹 페이지나 문서를 요청하면 프록시 서버는 문서를 클라이언트로 보내는 동안 원격 서버에서 로컬 캐시 디렉토리로 문서를 복사합니다 .

클라이언트가 이전에 요청하여 프록시 캐시에 복사된 문서를 다시 요청하는 경우 프록시는 이 문서를 원격 서버에서 다시 검색하지 않고 캐시에서 반환하게 됩니다 (그림 12-1 참조). 프록시는 해당 파일이 최신 상태가 아니라고 판단하면 클라이언트에 문서를 보내기 전에 원격 서버로부터 문서를 새로 고치고 해당 캐시를 업데이트합니다 .

그림 12-1 프록시 문서 검색



캐시의 파일은 Sun Java™ System Web Proxy Server 가비지 수집 유틸리티 (CacheGC) 가 자동으로 관리합니다 . CacheGC 는 일정한 간격에 따라 자동으로 캐시를 비워 캐시가 시한이 지난 문서로 혼잡해지지 않도록 합니다 .

Cache 구조 이해

캐시는 하나 이상의 파티션으로 구성됩니다. 파티션은 캐시를 위해 예비된 디스크의 저장 영역입니다 . 캐시를 여러 디스크에 두려면 각 디스크에 대해 적어도 하나의 캐시 파티션을 구성해야 합니다 . 각 파티션은 독립적으로 관리할 수 있습니다 . 즉 한 파티션을 다른 모든 파티션에 독립적으로 설정 , 해제 및 구성할 수 있습니다 .

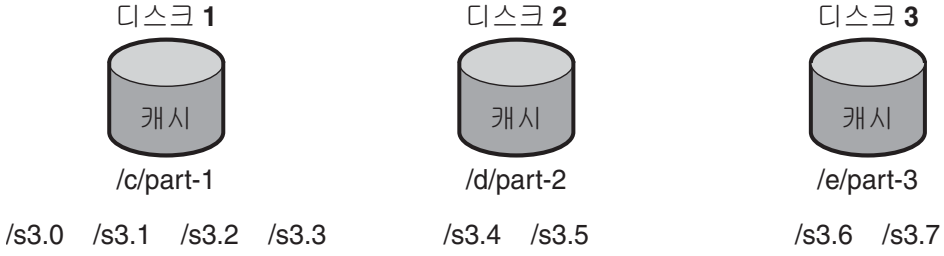
한 위치에 많은 수의 파일을 캐시하면 성능이 저하될 수 있으므로 각 파티션에 여러 개의 디렉토리 또는 구역을 만드는 것이 좋습니다 . 구역은 캐시 구조에서 파티션 아래 단계입니다 . 캐시에는 모든 파티션에 걸쳐 최대 256 개의 구역을 설정할 수 있습니다 . 캐시 구역의 수는 2 의 제곱수여야 합니다 (예 : 1, 2, 4, 8, 16, ..., 256).

캐시 구조 계층에서 마지막 단계는 하위 구역입니다 . 하위 구역은 구역 내의 디렉토리입니다 . 각 구역에는 64 개의 하위 구역이 있습니다 . 캐시된 파일은 캐시에서 가장 낮은 단계인 하위 구역에 저장됩니다 .

그림 12-2 는 파티션과 구역이 있는 캐시 구조의 예를 보여 줍니다 . 이 그림에서 캐시 디렉토리 구조는 전체 캐시를 3 개의 파티션으로 나누고 있습니다 . 첫 번째 파티션에는 4 개의 캐시 구역이 있으며 다른 두 파티션에는 2 개의 구역이 있습니다 .

각 캐시 구역은 구역 (section) 을 나타내는 s 와 구역 번호로 표시되어 있습니다 . s3.4 로 표시된 구역에서 3 은 캐시 구역의 수에 대한 2 의 제곱수 ($2^3 = 8$) 를 나타내며 4 는 구역 번호를 나타냅니다 (8 개의 구역은 0 부터 7 까지로 표시됨). 따라서 s3.4 는 8 개의 구역 중 5 번째 구역을 의미합니다 .

그림 12-2 캐시 구조의 예



캐시에 파일 분산

Proxy Server 는 특정 알고리즘을 사용하여 문서를 저장할 디렉토리를 결정합니다 . 이 알고리즘은 문서가 디렉토리에 균등하게 분산되도록 합니다 . 디렉토리에 문서의 수가 많아지면 성능이 저하될 수 있기 때문에 균등한 분산은 중요합니다 .

Proxy Server 는 RSA MD5 알고리즘 (Message Digest 5) 을 사용하여 URL 을 16 바이트의 이진 데이터로 줄이고 이 중 8 바이트는 문서를 캐시에 저장하는 데 사용되는 16 자의 16 진수 파일 이름을 계산하는 데 사용합니다 .

캐시 특성 설정

캐시 특성을 설정하여 캐시를 사용하도록 하고 Proxy Server 가 캐시할 프로토콜 유형을 제어할 수 있습니다 . 캐시 특성에는 다음 항목이 포함됩니다 .

- 캐시의 사용 여부
- 임시 파일을 저장하는 작업 디렉토리
- 캐시된 URL 을 기록할 디렉토리의 이름
- 캐시 크기
- 캐시 용량
- 캐시할 프로토콜 유형
- 캐시된 문서를 새로 고침하는 시기
- 프록시가 문서 액세스 횟수를 추적하여 원격 서버에 보고하는지 여부

캐시 특성을 설정하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Set Cache Specifics 링크를 누릅니다 . Set Cache Specifics 페이지가 표시됩니다 .
3. 적절한 옵션을 선택하여 캐시를 사용 또는 사용하지 않도록 설정할 수 있습니다. 기본으로 캐시는 사용함으로 설정되어 있습니다 . 자세한 내용은 " 캐시 사용 " (255 페이지) 을 참조하십시오 .
4. 작업 디렉토리를 입력합니다 . 기본적으로 작업 디렉토리는 프록시 인스턴스 아래에 있지만 이 디렉토리를 다른 위치에 두고 싶은 경우 변경할 수 있습니다 . 자세한 내용은 " 캐시 작업 디렉토리 만들기 " (256 페이지) 를 참조하십시오 .
5. Partition Configuration 링크를 누릅니다 . Add/Edit Cache Partitions 페이지가 표시됩니다 . 새 캐시 파티션을 추가하거나 기존 캐시 파티션을 편집할 수 있습니다 . 캐시 크기는 캐시가 커질 수 있는 최대 크기입니다 . 최대 캐시 크기는 32GB 입니다 . 자세한 내용은 " 캐시 크기 설정 " (256 페이지) 을 참조하십시오 .
6. Cache Capacity Configuration 링크를 누릅니다 . Set Cache Capacity 페이지가 표시됩니다 . Set Cache Capacity 페이지에서 캐시 용량을 설정할 수 있습니다 . 자세한 내용은 " 캐시 용량 편집 " (256 페이지) 을 참조하십시오 .
7. HTTP 문서 캐시를 사용하려면 Cache HTTP 확인란을 선택합니다 . 프록시 서버가 HTTP 문서를 캐시하도록 한 경우 캐시에 저장된 문서에 대해 항상 최신 여부를 확인하도록 할 것인지 또는 일정 간격으로 확인하도록 할 것인지 결정해야 합니다 . 또한 Proxy Server 에서 캐시 적중을 원격 서버로 보고할 것인지 여부를 설정할 수 있습니다 . 자세한 내용은 "HTTP 문서 캐시 " (257 페이지) 를 참조하십시오 . 다음 옵션에서 선택하십시오 .

- HTTP 를 항상 최신 상태로 유지하려면 Always Check That The Document Is Up To Date 옵션을 선택합니다 .
- Check Only If Last Check More Than 드롭다운 목록에서 프록시 서버의 새로 고침 간격을 시간 단위로 선택합니다 . 최신 여부 확인은 다음 옵션 중 하나를 사용하여 수행됩니다 .
  - **Use Last-modified Factor.** 문서와 함께 원본 서버에서 전송된 Last-modified 헤더입니다 .
  - **Use Only Explicit Expiration Information.** 프록시 서버는 Expires 헤더를 사용하여 해당 캐시 항목이 새로운 것인지 오래된 것인지 판단합니다 .
- 프록시 서버가 원격 서버에 대한 액세스 횟수를 보고하지 않도록 하려면 Never Report Accesses To Remote Server 옵션을 선택합니다 .
- 문서가 액세스된 횟수를 추적하여 원격 서버에 보고하도록 하려면 Report Cache Hits To Remote Server 옵션을 선택합니다 .

8. 캐시된 FTP 문서에 대한 새로 고침 간격을 설정할 수 있습니다 . Yes; Reload If Older Than 확인란을 선택한 다음 드롭다운 목록에서 값을 선택하여 시간 간격을 설정합니다 . 자세한 내용은 "FTP 및 Gopher 문서 캐시 " (259 페이지) 를 참조하십시오 .
9. 캐시된 Gopher 문서에 대한 새로 고침 간격을 설정할 수 있습니다 . Yes; Reload If Older Than 확인란을 선택한 다음 드롭다운 목록에서 값을 선택하여 시간 간격을 설정합니다 . 자세한 내용은 "FTP 및 Gopher 문서 캐시 " (259 페이지) 를 참조하십시오 .
10. OK 를 누릅니다 .
11. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
12. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

다음 절에서는 Set Cache Specifics 페이지에 표시되는 항목에 대한 자세한 내용을 제공하며 사용자의 요구에 가장 적합한 설정을 결정할 수 있도록 도움을 줍니다 .

캐시 사용

캐시는 프록시 서버 사용자의 네트워크 트래픽을 감소시키는 효과적인 방법입니다 . 캐시는 또한 원격 서버에서 문서를 검색할 필요가 없도록 하여 클라이언트에 더 빠른 응답 시간을 제공합니다 . 프록시 서버는 캐시를 사용할 때 가장 효과적으로 작동합니다 .

캐시 작업 디렉토리 만들기

캐시 파일은 캐시 파티션 아래에 있습니다 . 대개의 경우 사용자는 Set Cache Specifics 페이지에서 캐시의 상위 디렉토리를 작업 디렉토리로 지정합니다 . 캐시된 모든 파일은 캐시 디렉토리 아래에 구성된 디렉토리 구조에 표시됩니다 . 캐시 디렉토리의 이름을 변경하거나 다른 위치로 이동시키면 프록시에게 새 위치를 알려주어야 합니다 .

캐시 디렉토리 구조는 여러 파일 시스템으로 확장할 수 있습니다 . 이렇게 하면 대용량 디스크 하나에 모든 파일을 둘 필요 없이 여러 개의 작은 디스크로 분할된 캐시 구조를 만들 수 있습니다 . 각 프록시 서버는 자체 캐시 디렉토리 구조를 갖고 있어야 합니다 . 즉 , 여러 프록시 서버에서 캐시 디렉토리를 동시에 공유할 수 없습니다 .

캐시 크기 설정

캐시 크기는 파티션 크기를 나타냅니다 . 캐시 용량은 허용된 최대 크기이므로 캐시 크기는 캐시 용량보다 항상 작아야 합니다 . 모든 파티션 크기의 합은 캐시 크기와 같거나 작아야 합니다 .

프록시 캐시가 사용할 수 있는 디스크 공간의 크기는 캐시 성능에 중요한 영향을 미칩니다 . 캐시가 너무 작으면 Cache GC 가 디스크 공간 확보를 위해 그만큼 더 자주 캐시된 문서를 제거해야 하며 , 문서를 컨텐트 서버에서 검색하는 일도 더 잦아집니다 . 따라서 성능이 저하됩니다 .

캐시 크기는 크게 하는 것이 좋은데 , 더 많은 문서가 캐시될수록 네트워크 트래픽 부하는 줄어들고 프록시의 응답 시간은 더 빨라지기 때문입니다 . Cache GC 는 또한 사용자에게 더 이상 필요 없는 캐시된 문서를 제거합니다 . 다른 시스템 제한이 없다면 캐시 크기는 크게 할수록 좋습니다 . 초과되는 공간은 사용되지 않은 상태로 남아 있게 됩니다 .

또한 캐시를 여러 디스크 파티션에 나누어 둘 수도 있습니다 .

주의 캐시 구조 변경에는 많은 시간이 소요됩니다 .

캐시 용량 편집

Set Cache Capacity 페이지 및 Set Cache Specifics 페이지에서 캐시 용량을 편집할 수 있습니다 . 캐시 용량 편집에 대한 자세한 내용은 "Cache 용량 설정 " (261 페이지) 을 참조하십시오 .

HTTP 문서 캐시

HTTP 문서 캐시와 FTP 및 Gopher 문서 캐시는 내부적으로 다릅니다 . HTTP 문서는 다른 프로토콜의 문서에는 없는 캐시 기능을 제공합니다 . 캐시를 적절히 설정 및 구성하면 Proxy Server 가 HTTP, FTP 및 Gopher 문서를 효과적으로 캐시하도록 할 수 있습니다 .

모든 HTTP 문서에는 설명 헤더 섹션이 있어 Proxy Server 에서 이를 사용하여 프록시 캐시와 원격 서버에 있는 문서를 비교 및 평가할 수 있습니다 . 프록시는 HTTP 문서의 최신 여부를 확인할 때 캐시에 있는 문서가 오래된 버전일 경우 서버에 문서를 반환하라는 요청을 보냅니다 . 마지막 요청 이후 문서가 변경되지 않아 서버에서 문서를 전송하지 않는 경우가 많습니다 . 이 방법으로 HTTP 문서의 최신 상태를 확인함으로써 대역폭을 절약하고 지연을 줄일 수 있습니다 .

Proxy Server 에서는 HTTP 문서의 Cache Expiration 을 설정하여 원격 서버와의 트랜잭션을 줄일 수 있습니다 . Cache Expiration 설정에 따라 프록시는 서버에 요청을 보내기 전에 HTTP 문서의 최신 여부 확인이 필요한지 판단합니다 . 프록시는 HTTP 문서의 Last-Modified 헤더에 있는 최종 수정 날짜를 기반으로 최신 여부 확인의 필요 여부를 판단합니다 .

HTTP 문서에는 Cache Refresh 설정도 사용할 수 있습니다 . 이 옵션은 프록시가 Expiration 설정을 무시하고 항상 최신 여부 확인을 수행할지 , 아니면 최신 여부를 확인하기 전에 지정한 기간 동안 기다릴지 지정합니다 . 표 12-1 은 Expiration 및 Refresh 설정을 모두 지정한 경우 프록시의 동작을 보여 줍니다 . Refresh 설정을 사용하면 지연을 줄이고 대역폭을 상당히 절약할 수 있습니다 .

표 12-1 HTTP Cache Expiration 및 Cache Refresh 설정 사용

Refresh 설정	Expiration 설정	Results
Always do an up-to-date check	(해당 사항 없음)	항상 최신 여부 확인
User-specified interval	문서의 "expires" 헤더 사용	간격이 만료되었는지 최신 여부 확인
	문서의 Last-Modified 헤더로 평가	평가한 값과 expires 헤더 값 중 작은 값 *

* 두 값 중 작은 값을 사용함으로써 자주 변경되는 문서에 대해 캐시에서 오래된 데이터를 가져오는 문제를 방지할 수 있습니다 .

HTTP 캐시 새로 고침 간격 설정

Proxy Server 가 HTTP 문서를 캐시하도록 한 경우 캐시에 저장된 문서에 대해 항상 최신 여부를 확인하도록 할 것인지 또는 Cache Refresh 설정 (최신 여부 확인 간격) 을 기반으로 확인하도록 할 것인지 결정해야 합니다 . 예를 들어 HTTP 문서의 경우 적절한 새로 고침 간격은 4 ~ 8 시간입니다 . 새로 고침 간격이 길수록 프록시가 원격 서버에 연결하는 횟수는 줄어듭니다 . 프록시가 새로 고침 간격 도중에 최신 여부 확인을 하지 않는 경우에도 클라이언트의 Reload 버튼을 눌러 프록시가 원격 서버에 연결해 최신 여부를 확인하도록 함으로써 새로 고침을 강제할 수 있습니다 .

HTTP 문서에 대한 새로 고침 간격은 Set Cache Specifics 페이지 또는 Set Caching Configuration 페이지에서 설정할 수 있습니다 . Set Cache Specifics 페이지에서는 전역 캐시 절차를 구성할 수 있으며 Set Caching Configuration 페이지에서는 특정 URL 및 리소스에 대한 캐시 절차를 제어할 수 있습니다 .

HTTP 캐시 만료 정책 설정

또한 최종 수정 요인 또는 명시적인 만료 정보만 사용하여 캐시된 문서의 최신 여부를 확인하도록 서버를 설정할 수 있습니다 .

명시적인 만료 정보는 일부 HTTP 문서에서 파일의 시한이 만료되는 날짜 및 시간을 지정하는 헤더입니다 . 명시적인 Expires 헤더를 사용하는 HTTP 문서는 많지 않기 때문에 Last-modified 헤더를 기반으로 판단하는 것이 좋습니다 .

Last-modified 헤더를 기반으로 HTTP 문서를 캐시하도록 결정한 경우 만료 판단에 사용할 분수를 선택해야 합니다 . LM 요인이라고 하는 이 분수에 마지막 수정 시간과 문서에서 마지막으로 최신 여부를 확인한 시간의 간격을 곱합니다 . 결과 수와 최신 여부 확인을 마지막으로 수행한 이후 지난 시간을 비교합니다 . 결과 수가 시간 간격보다 작으면 문서가 만료되지 않은 것입니다 . 분수가 작을수록 프록시는 더 자주 문서를 확인합니다 . 예를 들어 10 일 전에 마지막으로 변경된 문서가 있습니다 . 최종 수정 요인을 0.1 로 설정하면 프록시는 문서가 1 일 (10 * 0.1 = 1) 동안 변경되지 않을 것이라는 의미로 해석합니다 . 프록시는 문서를 마지막으로 확인한 후 1 일 이내에는 이 문서를 캐시에서 반환합니다 .

이 예에서 HTTP 문서에 대한 캐시 새로 고침 설정이 1 일 미만으로 설정되어 있다면 프록시는 하루에 한 번 이상 최신 여부를 확인합니다 . 프록시는 항상 캐시 새로 고침 또는 캐시 만료 값 중 더 빈번한 업데이트를 요구하는 값을 사용합니다 .

HTTP 문서에 대한 만료 설정은 Set Cache Specifics 페이지 또는 Set Caching Configuration 페이지에서 할 수 있습니다 . Set Cache Specifics 페이지에서는 전역 캐시 절차를 구성할 수 있으며 Set Caching Configuration 페이지에서는 특정 URL 및 리소스에 대한 캐시 절차를 제어할 수 있습니다 .

원격 서버에 HTTP 액세스 보고

Sun Java™ System Web Proxy Server 에서 캐시한 문서는 새로 고치기 전에 여러 번 액세스될 수 있습니다 . 원격 서버에서 프록시가 캐시할 한 개의 사본을 보내는 것은 한 번의 액세스 , 또는 " 적중 " 을 의미합니다 . Sun Java™ System Web Proxy Server 는 지정한 문서가 최신 여부 확인 사이에 프록시 캐시에서 액세스된 횟수를 계산하고 문서의 다음 새로 고침 시 이 적중 수를 추가 HTTP 요청 헤더 (Cache-Info) 를 통해 원격 서버로 보낼 수 있습니다 . 원격 서버가 이 유형의 헤더를 인식하도록 구성된 경우 더 정확한 문서 액세스 횟수를 받을 수 있습니다 .

FTP 및 Gopher 문서 캐시

FTP 및 Gopher 는 문서가 최신 상태인지 확인할 수 있는 방법을 포함하고 있지 않습니다 . 따라서 FTP 및 Gopher 문서에서 캐시를 최적화하는 유일한 방법은 Cache Refresh 간격을 설정하는 것입니다 . Cache Refresh 간격은 Proxy Server 가 원격 서버에서 최신 버전의 문서를 검색하기 전까지 기다리는 시간입니다 . Cache Refresh 간격을 설정하지 않으면 프록시는 캐시에 있는 문서가 최신 상태인 경우에도 이러한 문서를 검색하게 됩니다 .

FTP 및 Gopher Cache Refresh 간격 설정

FTP 및 Gopher 에 대한 캐시 새로 고침 간격을 설정하는 경우 프록시에서 가져오는 문서에 대해 안심할 수 있는 간격을 선택합니다 . 예를 들어 거의 변경되지 않는 정보를 저장하는 경우 높은 수 (며칠) 를 사용합니다 . 계속 변경되는 데이터인 경우 적어도 몇 시간 간격으로 검색되도록 설정합니다 . 새로 고침 시간 중에는 오래된 파일을 클라이언트로 보낼 위험이 있습니다 . 간격을 몇 시간 정도로 충분히 짧게 하면 현저히 빠른 응답 시간을 확보하면서 이러한 위험을 거의 없앨 수 있습니다 .

FTP 및 Gopher 문서에 대한 캐시 새로 고침 간격은 Set Cache Specifics 페이지 또는 Set Caching Configuration 페이지에서 설정할 수 있습니다 . Set Cache Specifics 페이지에서는 전역 캐시 절차를 구성할 수 있으며 Set Caching Configuration 페이지에서는 특정 URL 및 리소스에 대한 캐시 절차를 제어할 수 있습니다 . Set Cache Specifics 페이지와 Set Caching Configuration 페이지 사용에 대한 자세한 내용은 각각 " 캐시 특성 설정 " (254 페이지) 과 " 캐시 구성 " (263 페이지) 을 참조하십시오 .

참고 FTP 및 Gopher 문서가 일부는 자주 변경되고 일부는 거의 변경되지 않는 등 종류가 다양한 경우에는 Set Caching Configuration 페이지를 사용하여 각 문서 종류에 대해 별도의 템플릿을 만들어 (예 : `ftp://.*.gif` 리소스로 템플릿 생성) 해당 리소스에 대해 적절한 새로 고침 간격을 설정합니다 .

Cache 생성 및 수정

캐시 파티션은 디스크 및 메모리에서 캐시에 사용하기 위해 예비된 부분입니다 . 캐시 용량이 변경된 경우 Add/Edit Cache Partitions 페이지에서 파티션을 변경하거나 추가할 수 있습니다 . 이 페이지에서 파티션의 위치 , 연상 기억 이름 , 최대 및 최소 크기를 편집할 수 있으며 , 해당 파티션의 캐시 구역 테이블을 볼 수 있습니다 .

캐시 파티션을 추가하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Add/Edit Cache Partitions 링크를 누릅니다 . Add/Edit Cache Partitions 페이지가 표시됩니다 .
3. Add Cache Partition 버튼을 누릅니다 . Cache Partition Configuration 페이지가 표시됩니다 .
4. 새 파티션에 대해 적절한 값을 입력합니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

cache 파티션을 수정하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Add/Edit Cache Partitions 링크를 누릅니다 . Add/Edit Cache Partitions 페이지가 표시됩니다 .
3. 변경하려는 파티션의 이름을 누릅니다 .
4. 정보를 수정합니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

Cache 용량 설정

캐시 용량 값은 캐시 디렉토리 구조를 유도하는 데 사용됩니다 . 캐시 디렉토리에 있을 수 있는 구역의 수는 캐시 용량에 따라 결정됩니다 . Set Cache Capacity 페이지가 나타나며 , 이 페이지에서 캐시 용량을 설정할 수 있습니다 . 용량이 클수록 계층도 커집니다 . 캐시 용량은 캐시 크기와 같거나 더 커야 합니다 . 이후 외부 디스크를 추가하는 등의 방법으로 캐시 크기를 늘릴 계획이 있는 경우 캐시 용량을 캐시 크기보다 크게 설정하는 것이 좋습니다 . 최대 캐시 용량은 32GB 이며 이 경우 256 개의 구역이 생성됩니다 .

캐시 용량을 설정하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Set Cache Capacity 링크를 누릅니다 . Set Cache Capacity 페이지가 표시됩니다 .
3. New Capacity Range 드롭다운 목록에서 용량을 선택합니다 .
4. OK 를 누릅니다 .
5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

Cache 구역 관리

프록시 캐시는 하나 이상의 캐시 구역으로 분할되어 있습니다 . 구역은 최대 256 개입니다 . 캐시 구역의 수는 2 의 제곱수여야 합니다 (예 : 1, 2, 4, 8, 16, ..., 256). 최대 용량은 32GB(최적) 로 , 캐시 구역은 256 개입니다 .

캐시 용량을 500MB 로 설정하면 설치 프로그램은 4 개의 캐시 구역을 만들며 (500 / 125 = 4), 2GB 로 선택하면 16 개의 구역을 만듭니다 (2000 / 125 = 16). 각 구역에 대한 최적 값이 125MB 로 선택되어 이와 같은 구역 수가 나오게 됩니다 . 구역의 수가 많을수록 저장되고 구역으로 분산되는 URL 의 수도 많아집니다 .

캐시 구역을 관리하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Manage Sections 링크를 누릅니다 . Manage Sections 페이지가 표시됩니다 .
3. 테이블의 정보를 변경합니다 . 파티션 간에 구역을 이동할 수 있습니다 .
4. OK 를 누릅니다 .
5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .

6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

가비지 수집 기본 설정 지정

Set Garbage Collection Preferences 페이지는 가비지 수집 모드를 설정하는 데 사용됩니다 .

캐시 가비지 수집기를 사용하여 캐시에서 파일을 제거할 수 있습니다 . 가비지 수집은 자동 모드 또는 명시적 모드로 수행됩니다 . 명시적 모드는 관리자가 외부에서 Schedule Garbage Collection 페이지를 사용하여 예약합니다 . 두 모드 중 하나를 선택한 다음 OK 를 누릅니다 . Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 . Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

가비지 수집 일정

Schedule Garbage Collection 페이지에서 가비지 수집이 실행되는 날짜와 시간을 지정할 수 있습니다 .

가비지 수집 일정을 지정하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Schedule Garbage Collection 링크를 누릅니다 . Schedule Garbage Collection 페이지가 표시됩니다 .
3. Schedule Garbage Collection At 목록에서 가비지 수집이 실행되는 시간을 선택합니다 .
4. 가비지 수집이 실행되는 주중 요일을 지정합니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

캐시 구성

Set Caching Configuration 페이지를 사용하여 특정 리소스에 대해 사용하려는 캐시의 종류를 구성할 수 있습니다 . 지정한 정규식 패턴에 일치하는 URL 에 대한 몇 가지의 구성 매개 변수를 지정할 수 있습니다 . 이 기능을 사용하면 캐시된 문서의 유형에 따라 프록시 캐시를 미세하게 제어할 수 있습니다 . 캐시를 구성하는 과정에서 다음 항목들을 확인할 수 있습니다 .

- 캐시 기본값
- 인증이 필요한 페이지를 캐시하는 방법
- 쿼리를 캐시하는 방법
- 캐시 파일의 최소 , 최대 크기
- 캐시된 문서를 새로 고침하는 시기
- 캐시 만료 정책
- 클라이언트 중단에 대한 캐시 동작
- 원래 서버에 대해 실패한 연결에 대한 캐시 동작

참고 특정 리소스에 대한 캐시 기본값을 Derived Configuration, 또는 Don’ t Cache 로 설정한 경우 Set Caching Configuration 페이지에 캐시 구성 옵션이 표시되지 않습니다 . 그러나 리소스에 대한 캐시 기본값을 캐시 사용함으로 선택하면 몇 가지 다른 구성 항목을 지정할 수 있습니다 .

캐시를 구성하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Set Caching Configuration 페이지를 누릅니다 . Set Caching Configuration 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 누르고 정규식을 입력한 다음 OK 를 누릅니다 .
4. 구성 정보를 변경합니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

캐시 구성 요소

다음 절에서는 Set Caching Configuration 페이지에 표시되는 항목에 대해 설명합니다 . 이 절은 사용자의 요구에 가장 적합한 구성을 결정하는 데 도움이 되는 정보를 포함하고 있습니다 .

캐시 기본값 설정

프록시 서버는 특정 리소스에 대한 캐시 기본값을 확인할 수 있도록 합니다 . 리소스는 사용자가 지정한 특정 기준에 일치하는 파일 유형입니다 . 예를 들어 도메인 company.com 의 모든 문서를 자동으로 캐시하도록 서버를 구성하려고 합니다 . 이 경우 Set Caching Configuration 페이지의 상단에 있는 Regular Expression 버튼을 누르고 표시되는 필드에 다음을 입력합니다 .

```
[a-z] *://[^/:]\.company\.com.*
```

기본적으로 캐시 옵션이 선택되어 있습니다 . 이렇게 하면 서버는 해당 도메인의 캐시 가능한 모든 문서를 자동으로 캐시합니다 . 정규식에 대한 자세한 내용은 Understanding Regular Expressions 를 참조하십시오 .

참고	특정 리소스에 대한 캐시 기본값을 Derived Configuration, 또는 Don’ t Cache 로 설정한 경우 해당 리소스에 대한 캐시를 구성할 필요가 없습니다 . 그러나 리소스에 대한 캐시 기본값을 캐시 사용함으로 선택하면 몇 가지 다른 구성 항목을 지정할 수 있습니다 . 이러한 항목 목록은 " 캐시 구성 " (263 페이지) 을 참조하십시오 .

HTTP, FTP, Gopher 에 대한 캐시 기본값도 Set Cache Specifics 페이지에서 설정할 수 있습니다 .

Caching Pages That Require Authentication

서버가 사용자 인증이 필요한 파일을 캐시하도록 할 수 있습니다 . 이러한 파일을 캐시하도록 선택하면 Proxy Server 는 캐시 파일에 표시를 남겨 사용자가 원격 서버의 인증이 필요한 파일을 요청하는 경우 이를 알 수 있도록 합니다 .

프록시 서버는 원격 서버의 인증 방법과 사용자 ID 및 비밀번호를 알 수 없기 때문에 , 인증이 필요한 문서 요청이 들어올 때마다 단순히 원격 서버를 통한 최신 여부 확인을 강제합니다 . 따라서 해당 파일에 액세스하려면 사용자는 ID 와 암호를 입력해야 합니다 . 사용자가 Navigator 세션에서 이전에 이미 해당 서버에 액세스한 경우 Navigator 는 사용자에게 입력을 요구하지 않고 자동으로 인증 정보를 전송합니다 .

인증이 필요한 페이지의 캐시를 사용하도록 설정하지 않은 경우 프록시는 기본값으로 간주하여 이러한 페이지를 캐시하지 않습니다 .

Caching Queries

캐시된 쿼리는 HTTP 문서에서만 동작합니다 . 캐시된 쿼리의 길이를 제한하거나 쿼리 캐시를 완전히 금지할 수 있습니다 . 쿼리가 길수록 캐시 내에서 중복될 가능성은 줄어들지만 , 그만큼 캐시의 유용성도 작아집니다 .

이러한 캐시 제한은 다음과 같은 조건의 쿼리에 적용됩니다 . 액세스 메소드는 GET 이어야 하며 , 문서 보호가 적용되지 않아야 하고 (인증된 페이지 캐시를 사용하지 않는 경우), 응답에는 최소한 Last-modified 헤더가 있어야 합니다 . 따라서 쿼리 엔진은 해당 쿼리 결과 문서가 캐시될 수 있음을 표시해야 합니다 . Last-Modified 헤더가 있는 경우 쿼리 엔진은 캐시 효과를 위해 조건부 GET 메소드 (If-Modified-Since 헤더와 함께 사용) 를 지원해야 합니다 . 그렇지 않은 경우 Expires 헤더를 반환해야 합니다 .

캐시 파일 최소 및 최대 크기 설정

Proxy Server 에서 캐시하는 파일의 최소 및 최대 크기를 설정할 수 있습니다 . 네트워크 연결 속도가 빠른 경우라면 최소 크기를 설정할 것입니다 . 네트워크 속도가 빠르면 용량이 작은 파일은 빠르게 검색되므로 서버에서 캐시할 필요가 없습니다 . 이 경우 용량이 비교적 큰 파일만 캐시할 것입니다 . 최대 파일 크기를 설정하면 용량이 큰 파일이 프록시의 디스크 공간을 너무 많이 차지하지 않도록 할 수 있습니다 .

최신 여부 확인 정책 설정

이 옵션을 사용하면 HTTP 문서가 항상 최신 상태가 되도록 할 수 있습니다 . 또한 Proxy Server 의 새로 고침 간격을 지정할 수 있습니다 .

만료 정책 설정

마지막으로 수정된 요인 또는 명시적인 만료 정보를 사용하여 만료 정책을 설정할 수 있습니다 .

클라이언트 중단에 대한 캐시 동작 설정

문서의 일부분만 가져온 상황에서 클라이언트가 데이터 전송을 중단한 경우에도 프록시는 캐시를 위해 해당 문서 전체를 가져올 수 있습니다 . 프록시 기본값은 문서의 25% 이상을 이미 가져온 경우 문서 전체를 가져와 캐시에 저장하도록 합니다 . 25% 미만이면 프록시는 해당 원격 서버와의 연결을 종료하고 가져온 파일 일부분을 제거합니다 . 클라이언트 중단 비율을 높이거나 낮출 수 있습니다 .

Behaviour On Failure To Connect To Server

원본 서버에 연결되지 않아 지난 문서의 최신 여부를 확인할 수 없는 경우 프록시가 캐시의 지난 문서를 전송할 것인지 지정할 수 있습니다 .

로컬 호스트 캐시

로컬 호스트에서 요청한 URL 에 도메인 이름이 없는 경우 Proxy Server 는 중복 캐시를 막기 위해 이를 캐시하지 않습니다 . 예를 들어 사용자가 로컬 서버에서 `http://machine/filename.html` 과 `http://machine.example.com/filename.html` 을 요청하면 두 URL 은 모두 캐시에 표시됩니다 . 이와 같은 파일들은 로컬 서버에서 전송되기 때문에 빠르게 가져올 수 있으며 , 따라서 캐시할 필요가 없습니다 .

그러나 많은 원격 위치에 서버를 두고 있는 기업에서는 모든 호스트의 문서를 캐시하도록 하여 네트워크 트래픽을 감소시키고 이 파일들에 액세스하는 데 필요한 시간을 줄일 수 있습니다 .

로컬 호스트의 캐시를 사용하도록 설정하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Cache Local Hosts 링크를 누릅니다 . Cache Local Hosts 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 누르고 정규식을 입력한 다음 OK 를 누릅니다 . 정규식에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 를 참조하십시오 .
4. Enabled 버튼을 누릅니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

파일 캐시 구성

기본적으로 파일 캐시는 사용하도록 설정됩니다 . 파일 캐시 설정은 `server.xml` 파일에 포함됩니다 . Server Manager 를 사용하여 파일 캐시 설정을 변경할 수 있습니다 .

참고 Configure File Cache 페이지는 사용자 인터페이스에 표시되지만 Proxy Server 4 릴리스에서 구현되지는 않았습니다 .

파일 캐시를 구성하려면 다음을 수행합니다 .

1. Server Manager 에서 Preferences 탭을 누릅니다 .
2. File Cache Configuration 링크를 누릅니다 . File Cache Configuration 페이지가 표시됩니다 .
3. Enable File Cache 가 선택되지 않았으면 선택합니다 .
4. 파일 전송 여부를 선택합니다 .

   Transmit File 을 사용하면 서버는 파일 내용이 아닌 파일 캐시에서 파일의 열린 파일 설명자 (descriptor) 를 캐시하고 PR_TransmitFile 은 파일 내용을 클라이언트로 전송하는 데 사용됩니다 . Transmit File 을 사용하면 열린 파일 설명자만 캐시되므로 일반적인 파일 캐시의 대 , 중 , 소 규모의 파일 구분이 더 이상 적용되지 않습니다 . 기본적으로 Transmit File 은 Windows 에서는 사용하도록 설정되며 UNIX 에서는 사용하지 않도록 설정됩니다 . UNIX 의 경우 PR_TransmitFile 을 운영 체제에서 원시 지원하는 플랫폼에서만 Transmit File 을 사용합니다 . 현재 이러한 플랫폼에는 HP-UX 와 AIX 가 있습니다 . 이 외의 UNIX/Linux 플랫폼에서는 Transmit File 사용을 권장하지 않습니다 .
5. 해시 테이블의 크기를 입력합니다 . 기본 크기는 파일의 최대 수의 두 배 더하기 1 입니다 . 예를 들어 , 파일의 최대 수가 1024 로 설정되었으면 기본 해시 테이블 크기는 2049 입니다 .
6. 유효한 캐시 항목의 최대 지속 시간을 입력합니다 (초 단위). 기본값은 30 입니다 . 이 설정에 따라 파일이 캐시된 후 캐시된 정보가 계속하여 사용되는 시간이 달라집니다 . 동일한 파일이 캐시를 통하여 참조된 경우 , MaxAge 값보다 오래된 항목은 동일한 파일의 새 항목으로 대체됩니다 . 컨텐트가 일정한 스케줄에 의하여 업데이트 (기존 파일이 변경) 되는지의 여부에 따라 최대 지속 시간을 설정합니다 . 예를 들어 컨텐트가 하루에 네 번 일정 간격으로 업데이트 된다면 Maximum Age 를 21600 초 (6 시간) 로 설정할 수 있습니다 . 그렇지 않은 경우 , 컨텐트 파일의 이전 버전을 수정 후 최대 얼마동안 서비스할지에 따라 Maximum Age 를 설정하는 것이 좋습니다 .
7. 캐시할 최대 파일 개수를 입력합니다 . 기본값은 1024 입니다 .
8. Medium File Size Limit 및 Small File Size Limit 값을 입력합니다 (바이트 단위). Medium File Size Limit 기본값은 537600, Small File Size Limit 기본값은 2048 입니다 .

   캐시는 대 , 중 , 소형의 파일을 각기 다르게 처리합니다 . 중간 크기 파일의 내용은 파일을 가상 메모리 (현재 UNIX/Linux 플랫폼에만 적용) 에 매핑하여 캐시합니다 . 작은 파일의 내용은 힙 공간 (heap space) 을 할당하고 파일을 이 공간으로 읽어 캐시합니다 . 큰 파일 (중간 크기 파일보다 큰) 의 경우 파일에 대한 정보

는 캐시하지만 내용은 캐시하지 않습니다 . 작은 파일과 중간 크기 파일을 구별하면 작은 파일이 많은 경우 가상 메모리의 페이지에서 많은 부분이 낭비되지 않도록 할 수 있는 장점이 있습니다 . 따라서 Small File Size Limit 값은 일반적으로 VM 페이지 크기보다 약간 작게 설정합니다 .

9. 중간 파일 공간 및 작은 파일 공간을 설정합니다 . 중간 파일 공간은 모든 중간 크기 파일을 매핑하는 데 사용되는 가상 메모리의 크기 (바이트 단위) 입니다 . 기본값은 10485760 입니다 . 작은 파일 공간은 캐시에 사용되는 힙 공간의 크기 (바이트 단위) 이며 , 작은 파일을 캐시하는 데 사용되는 힙 공간을 포함합니다 . UNIX/Linux 플랫폼에서 기본값은 1048576 입니다 .
10. OK 를 누릅니다 .
11. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
12. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

URL 데이터베이스 확인

캐시된 모든 URL 의 기록에서 이름과 속성을 확인할 수 있습니다 . URL 정보에서는 액세스 프로토콜과 사이트 이름에 따라 그룹화된 캐시된 문서 목록이 표시됩니다 . 목록에서 제한된 URL 만 보려면 Search 필드에 도메인 이름을 입력합니다 . 이 정보를 통하여 캐시에서 문서를 제거하거나 기간을 만료하는 것과 같은 다양한 캐시 관리 기능을 수행할 수 있습니다 .

데이터베이스에서 URL 을 확인하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. View URL Database 링크를 누릅니다 . View URL Database 페이지가 표시됩니다 .
3. Regenerate 버튼을 누르면 캐시된 URL 의 현재 목록을 생성합니다 . 특정 URL 에 대한 정보를 보려면 URL 또는 정규식을 Search 필드에 입력한 다음 Search 버튼을 누릅니다 .
4. 도메인 이름과 호스트에 따라 그룹화된 캐시 데이터베이스 정보를 보려면 목록에서 도메인 이름을 선택합니다 . 해당 도메인의 호스트 목록이 표시됩니다 . 호스트 이름을 누르면 URL 목록이 표시됩니다 .
5. URL 이름을 누릅니다 . 해당 URL 에 대한 자세한 정보가 표시됩니다 .

캐시에서 파일 만료 및 제거

View URL Database 페이지를 사용하여 캐시에서 문서의 기간을 만료하거나 문서를 제거할 수 있습니다 .

캐시된 URL 의 기간을 만료하거나 제거하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. View URL Database 링크를 누릅니다 . View URL Database 페이지가 표시됩니다 .
3. Regenerate 버튼을 누릅니다 . 캐시 데이터베이스의 스냅숏이 생성됩니다 . 나머지 단계는 이 스냅숏을 기반으로 합니다 .
4. 기간을 만료하거나 제거하려는 특정 URL 을 아는 경우 Search 필드에 이 URL 또는 이 URL 과 일치하는 정규식을 입력하고 Search 버튼을 누릅니다 . 도메인 이름과 호스트에 따라 그룹화된 URL 로 작업하려면 목록에서 도메인 이름을 선택합니다 . 해당 도메인의 호스트 목록이 표시됩니다 . 호스트 이름을 누르면 URL 목록이 표시됩니다 .
5. 개별 파일의 기간을 만료하려면 해당 파일에 대한 URL 옆에 있는 Ex 옵션을 선택하고 Exp/Rem Marked 버튼을 누릅니다 . 목록에 있는 모든 파일의 기간을 만료하려면 양식 하단의 Exp All 버튼을 누릅니다 . 캐시에서 개별 파일을 제거하려면 해당 파일에 대한 URL 옆에 있는 Rm 옵션을 선택하고 Exp/Rem Marked 버튼을 누릅니다 . 목록에 있는 모든 파일을 제거하려면 Rem All 버튼을 누릅니다 .
6. 스냅숏을 다시 생성하려면 Regenerate 버튼을 누릅니다 .

참고 Ex 또는 Rm 옵션을 사용하면 해당 파일은 처리되지만 스냅숏에는 변경 사항이 적용되지 않습니다 . 변경 사항을 보려면 스냅숏을 다시 생성해야 합니다 .

Cache Batch Updates 사용

Cache Batch Updates 기능을 사용하면 프록시 서버가 사용 중이 아닐 때 특정 웹 사이트의 파일을 미리 로드하거나 이미 캐시에 있는 문서의 최신 여부를 확인할 수 있습니다 . Set Cache Batch Updates 페이지에서 URL 을 일괄적으로 작성 , 편집 , 제거할 수 있으며 일괄 업데이트를 사용 또는 사용하지 않도록 설정할 수 있습니다 .

일괄 업데이트 생성

파일을 일괄 업데이트하도록 지정하면 필요에 따라 캐시하는 것과 달리 활발하게 파일을 캐시할 수 있습니다 . 프록시 서버는 현재 캐시에 있는 여러 파일의 최신 여부를 확인하거나 특정 웹 사이트에 있는 여러 파일을 미리 로드할 수 있도록 합니다 .

일괄 업데이트를 만들려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Set Cache Batch Updates 링크를 누릅니다 . Set Cache Batch Updates 페이지가 표시됩니다 .
3. Create/Select a Batch Update Configuration 옆에 있는 드롭다운 목록에서 New and Create 를 선택합니다 .
4. OK 를 누릅니다 . Set Cache Batch Updates 페이지가 표시됩니다 .
5. Name 섹션에 새로운 일괄 업데이트 항목의 이름을 입력합니다 .
6. 페이지의 Source 섹션에서 새로 만들 일괄 업데이트 유형에 해당하는 라디오 버튼을 누릅니다 . 캐시에 있는 모든 문서의 최신 여부 확인을 수행하려면 첫 번째 라디오 버튼을 누릅니다 . 주어진 원본 URL 로 시작하는 URL 을 반복적으로 캐시하려면 두 번째 라디오 버튼을 누릅니다 .
7. Source 섹션 필드에 일괄 업데이트에서 사용할 문서를 지정합니다 .
8. Exceptions 섹션에 일괄 업데이트에서 제외할 파일을 지정합니다 .
9. Resources 섹션에 최대 동시 연결 수와 통과할 최대 문서 수를 입력합니다 .
10. Timing 섹션에 일괄 업데이트의 생성 시작과 종료 시간을 입력합니다 . 일괄 업데이트는 한 번에 하나만 활성화할 수 있으므로 다른 일괄 업데이트 구성과 겹치지 않도록 하는 것이 좋습니다 .
11. OK 를 누릅니다 .

참고 일괄 업데이트를 사용하지 않는 경우에도 일괄 업데이트 구성을 작성 , 편집 , 제거할 수 있습니다 . 그러나 Set Cache Batch Updates 페이지에서 설정한 시간에 따라 일괄 업데이트를 수행하려면 업데이트를 사용해야 합니다 .

12. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
13. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

일괄 업데이트 구성 편집 및 삭제

Set Cache Batch Updates 페이지에서 일괄 업데이트를 편집하거나 제거할 수 있습니다 . 일괄 업데이트를 편집하여 특정 파일을 제외하거나 일괄 업데이트 간격을 더 작게 할 수 있습니다 . 일괄 업데이트를 완전히 제거할 수도 있습니다 .

일괄 업데이트 구성을 편집 또는 삭제하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Set Cache Batch Updates 링크를 누릅니다 . Set Cache Batch Updates 페이지가 표시됩니다 .
3. 일괄 업데이트를 편집하려면 일괄 업데이트의 이름을 선택하고 Create/Select a Batch Update Configuration 옆에 있는 드롭다운 목록에서 "Edit" 를 선택합니다 . 일괄 업데이트를 삭제하려면 일괄 업데이트의 이름을 선택하고 드롭다운 목록에서 "Delete" 를 선택합니다 .
4. OK 를 누릅니다 . Set Cache Batch Updates 페이지가 표시됩니다 .
5. 원하는 대로 정보를 수정합니다 .
6. OK 를 누릅니다 .
7. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
8. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

캐시 명령줄 인터페이스 사용

프록시 서버에는 캐시 디렉토리 구조를 구성 , 변경 , 생성 및 수정할 수 있는 여러 개의 명령줄 유틸리티가 포함되어 있습니다 . 이러한 유틸리티의 대부분은 Server Manager 페이지와 기능이 중복되지만 크론 작업과 같은 유지 보수 일정을 예약하는 경우 이 유틸리티를 사용할 수 있습니다 . 모든 유틸리티는 `extras` 디렉토리에 있습니다 .

명령줄 유틸리티를 실행하려면 다음을 수행합니다 .

1. 명령줄 프롬프트에서 *server_root*`/proxy-server`*id* 디렉토리로 이동합니다 .
2. **`./start -shell`** 을 입력합니다 .

   다음 절에서는 다양한 유틸리티에 대해 설명합니다 .

캐시 디렉토리 구조 구축

프록시에 포함된 cbuild 유틸리티는 오프라인 캐시 데이터베이스 관리자입니다 . 이 유틸리티로 명령줄 인터페이스를 사용하여 새 캐시 구조를 만들거나 기존 캐시 구조를 수정할 수 있습니다 . Server Manager 페이지를 사용하여 프록시에서 새로 만든 캐시를 사용하도록 할 수 있습니다 . 이 유틸리티는 server.xml 파일을 업데이트하지 않습니다 . cbuild 는 여러 파티션이 있는 캐시의 크기를 조정할 수 없습니다 . server.xml 파일에 있는 CACHE 라는 요소에는 cachecapacity 매개 변수가 있습니다 . cbuild 에서 캐시를 만들거나 수정한 경우 cachecapacity 매개 변수를 server.xml 파일에서 직접 업데이트해야 합니다 .

```
<PARTITION partitionname="part1" partitiondir="/home/build/install9
/proxy-server1/cache" maxsize="1600" minspace="5" enabled="true"/>
<CACHE enabled="true" cachecapacity="2000" cachedir="/tmp/cache">
```

cbuild 유틸리티는 두 가지 모드로 실행할 수 있습니다 . 첫 번째 모드는 다음과 같습니다 .

cbuild -d *conf-dir* -c *cache-dir* -s *cache size*

cbuild -d *conf-dir* -c *cache-dir* -s *cache size* -r

예 :

cbuild -d *server_root*/proxy-server*id*/config -c *server_root*/proxy-server*id*/cache -s 512

cbuild -d *server_root*/proxy-server*id*/config -c *server_root*/proxy-server*id*/cache -s 512 -r

각 부분의 의미는 다음과 같습니다 .

- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*/proxy-server*id*/config 입니다 .
- *cache-dir* 은 캐시 구조에 대한 디렉토리입니다 .
- *cache size* 는 캐시가 커질 수 있는 최대 크기입니다 . 이 옵션은 *cache-dim* 매개 변수와 함께 사용할 수 없습니다 . 최대 크기는 65135MB 입니다 .
- -r 은 기존 캐시 구조의 크기를 조정합니다 (캐시에 파티션이 하나만 있는 경우). 캐시를 새로 만드는 경우에는 필요하지 않습니다 .

cbuild 를 실행하는 두 번째 모드는 다음과 같습니다 .

cbuild -d *conf-dir* -c *cache-dir* -n *cache-dim*

cbuild -d *conf-dir* -c *cache-dir* -n *cache-dim* -r

예 :

```
cbuild -d server_root/proxy-serverid/config -c server_root/proxy-serverid/cache
-n 3
```

```
cbuild -d server_root/proxy-serverid/config -c server_root/proxy-serverid/cache
-n 3 -r
```

각 부분의 의미는 다음과 같습니다 .

- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*/proxy-server*id*/config 입니다 .
- *cache-dir* 은 캐시 구조에 대한 디렉토리입니다 .
- *cache-dim* 은 구역의 수를 결정합니다 . 예를 들어 그림 12-1 의 s3.4 로 표시된 구역에서 3 은 캐시 구역의 수를 나타냅니다 . *cache-dim 의 기본값은* 0 이며 최대값은 8 입니다 .
- -r 은 기존 캐시 구조의 크기를 조정합니다 (캐시에 파티션이 하나만 있는 경우). 캐시를 새로 만드는 경우에는 필요하지 않습니다 .

캐시 URL 목록 관리

프록시에 포함된 urldb 유틸리티는 캐시의 URL 목록을 관리합니다 . 이 유틸리티를 사용하여 캐시된 URL 을 나열할 수 있습니다 . 또한 선택적으로 캐시 데이터베이스에서 캐시된 개체의 기간을 만료하거나 제거할 수 있습니다 .

urldb 명령은 -o 옵션을 기반으로 다음의 세 그룹으로 분류할 수 있습니다 .

- 도메인
- 사이트
- URL

도메인을 나열하려면 명령줄에서 다음을 입력합니다 .

```
urldb -o matching_domains -e reg_exp -d conf-dir
```

예 :

```
urldb -o matching_domains -e ".*phoenix.*" -d
server_root/proxy-serverid/config
```

각 부분의 의미는 다음과 같습니다 .

- matching_domains 는 정규식에 일치하는 도메인을 나열합니다 .

- *reg_exp* 는 사용되는 정규식입니다 .
- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*/proxy-server*id*/config 입니다 .

도메인에서 일치하는 모든 사이트를 나열하려면 명령줄에서 다음을 입력합니다 .

```
urldb -o matching_sites_in_domain -e reg_exp -m domain_name -d conf-dir
```

예 :

```
urldb -o matching_sites_in_domain -e ".*atlas" -m phoenix.com -d
server_root/proxy-serverid/config
```

각 부분의 의미는 다음과 같습니다 .

- matching_sites_in_domain 은 도메인에서 정규식에 일치하는 모든 사이트를 나열합니다 .
- *reg_exp* 는 사용되는 정규식입니다 .
- *domain_name* 은 도메인의 이름입니다 .
- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*/proxy-server*id*/config 입니다 .

일치하는 모든 사이트를 나열하려면 명령줄에서 다음을 입력합니다 .

```
urldb -o all_matching_sites -e reg_exp -d conf-dir
```

예 :

```
urldb -o all_matching_sites -e ".*atlas.*" -d server_root/proxy-serverid/config
```

각 부분의 의미는 다음과 같습니다 .

- all_matching_sites 는 정규식에 일치하는 모든 사이트를 나열합니다 .
- *reg_exp* 는 사용되는 정규식입니다 .
- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*/proxy-server*id*/config 입니다 .

사이트에서 일치하는 URL 을 나열하려면 명령줄에서 다음을 입력합니다 .

```
urldb -o matching_urls_from_site -e reg_exp -s site_name -d conf-dir
```

예 :

```
urldb -o matching_urls_from_site -e "http://.*atlas.*" -s atlas.phoenix.com
-d server_root/proxy-serverid/config
```

각 부분의 의미는 다음과 같습니다 .

- `matching_urls_from_site` 는 사이트에서 정규식에 일치하는 모든 URL 을 나열합니다 .
- *reg_exp* 는 사용되는 정규식입니다 .
- *site_name* 은 사이트의 이름입니다 .
- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*`/proxy-server`*id*`/config` 입니다 .

사이트에서 일치하는 URL 의 기간을 만료하거나 제거하려면 명령줄에서 다음을 입력합니다 .

`urldb -o matching_urls_from_site -e` *reg_exp* `-s` *site_name* `-x e -d` *conf-dir*

`urldb -o matching_urls_from_site -e` *reg_exp* `-s` *site_name* `-x r -d` *conf-dir*

예 :

`urldb -o matching_urls_from_site -e "http://.*atlas.*" -s atlas.phoenix.com -x e -d` *iserver_root*`/proxy-server`*id*`/config`

각 부분의 의미는 다음과 같습니다 .

- `matching_urls_from_site` 는 사이트에서 정규식에 일치하는 모든 URL 을 나열합니다 .
- *reg_exp* 는 사용되는 정규식입니다 .
- *site_name* 은 사이트의 이름입니다 .
- `-x e` 는 캐시 데이터베이스에서 일치하는 URL 의 기간을 만료하기 위한 옵션입니다 . 이 옵션은 도메인 및 사이트 모드에서 사용할 수 없습니다 .
- `-x r` 은 캐시 데이터베이스에서 일치하는 URL 을 제거하기 위한 옵션입니다 .
- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*`/proxy-server`*id*`/config` 입니다 .

일치하는 모든 URL 을 나열하려면 명령줄에서 다음을 입력합니다 .

`urldb -o all_matching_urls -e` *reg_exp* `-d` *conf-dir*

예 :

`urldb -o all_matching_urls -e ".*cgi-bin.*" -d` *server_root*`/proxy-server`*id*`/config`

각 부분의 의미는 다음과 같습니다 .

- `all_matching_urls` 는 정규식에 일치하는 모든 URL 을 나열합니다 .

- *reg_exp* 는 사용되는 정규식입니다 .
- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*/proxy-server*id*/config 입니다 .

일치하는 모든 URL 의 기간을 만료하거나 제거하려면 명령줄에서 다음을 입력합니다 .

```
urldb -o all_matching_urls -e reg_exp -x e -d conf-dir
```

```
urldb -o all_matching_urls -e reg_exp -x r -d conf-dir
```

예 :

```
urldb -o all_matching_urls -e ".*cgi-bin.*" -x e -d
server_root/proxy-serverid/config
```

각 부분의 의미는 다음과 같습니다 .

- all_matching_urls 는 정규식에 일치하는 모든 URL 을 나열합니다 .
- *reg_exp* 는 사용되는 정규식입니다 .
- -x e 는 캐시 데이터베이스에서 일치하는 URL 의 기간을 만료하기 위한 옵션입니다 .
- -x r 은 캐시 데이터베이스에서 일치하는 URL 을 제거하기 위한 옵션입니다 .
- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*/proxy-server*id*/config 입니다 .

URL 목록의 기간을 만료하거나 제거하려면 명령줄에서 다음을 입력합니다 .

```
urldb -l url-list -x e -e reg_exp -d conf-dir
```

```
urldb -l url-list -x r -e reg_exp -d conf-dir
```

예 :

```
urldb -l url.lst -x e -e ".*cgi-bin.*" -d server_root/proxy-serverid/config
```

각 부분의 의미는 다음과 같습니다 .

- url-list 는 기간을 만료해야 하는 URL 목록입니다 . 이 옵션은 URL 목록을 제공하는 데 사용할 수 있습니다 .
- -x e 는 캐시 데이터베이스에서 일치하는 URL 의 기간을 만료하기 위한 옵션입니다 .
- -x r 은 캐시 데이터베이스에서 일치하는 URL 을 제거하기 위한 옵션입니다 .
- *reg_exp* 는 사용되는 정규식입니다 .

- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*/proxy-server*id*/config 입니다 .

캐시 가비지 수집 관리

캐시 크기 제한으로 인해 필요한 경우 cachegc 유틸리티를 사용하여 만료되었거나 디렉토리에 캐시하기에는 지나치게 오래된 개체의 데이터베이스 캐시를 비울 수 있습니다 .

참고 cachegc 유틸리티를 사용할 때 CacheGC 가 프록시 인스턴스에서 실행 중이 아닌지 확인하십시오 .

cachegc 유틸리티는 다음과 같은 방법으로 사용할 수 있습니다 .

cachegc -f *leave-fs-full-percent* -u *gc-high-margin-percent* -l *gc-low-margin-percen*t -e *extra-margin-percen*t -d *conf-dir*

예 :

cachegc -f 50 -u 80 -l 60 -e 5 -d *server_root*/proxy-server*id*/config

각 부분의 의미는 다음과 같습니다 .

- *leave-fs-full-percent* 는 캐시 파티션 크기의 비율을 지정해 그 이하인 경우 가비지 수집을 실행하지 않습니다 .
- *gc-high-margin-percent* 는 최대 캐시 크기의 비율을 조정해 비율에 도달하면 가비지 수집을 실행합니다 .
- *gc-low-margin-percen*t 는 가비지 수집기가 목표로 하는 최대 캐시 크기의 비율을 조정합니다 .
- *extra-margin-percen*t 는 가비지 수집기가 제거할 캐시 조각을 결정하는 데 사용됩니다 .
- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*/proxy-server*id*/config 입니다 .

일괄 업데이트 관리

bu 유틸리티는 두 가지 모드에서 동작하며 캐시를 업데이트합니다 . 첫 번째 모드에서는 캐시 데이터베이스를 반복하여 거치면서 캐시에 있는 모든 URL 에 대해 각각 HTTP 요청을 보내 업데이트합니다 . 두 번째 모드에서는 지정된 URL 로 시작하여 이 URL 의 모든 링크에 대해 지정한 깊이까지 너비 우선 반복을 수행하고 페이지를 캐시로 가져옵니다 . bu 는 RFC 호환 로봇입니다 .

bu -n *hostname* -p *port* -t *time-lmt* -f *contact-address* -s *sleep-time* -o *object* -r n -d *conf-dir*

예 :

bu -n phoenix -p 80 -t 3600 -f admin@phoenix.com -s 60 -o nova -r n -d *server_root*/proxy-server*id*/config

각 부분의 의미는 다음과 같습니다 .

- *hostname* 은 프록시가 실행 중인 컴퓨터의 호스트 이름입니다 . 기본값은 localhost 입니다 .
- *port* 는 프록시 서버가 실행 중인 포트입니다 . 기본 포트는 8080 입니다 .
- *time-lmt* 는 유틸리티 실행 제한 시간입니다 .
- *contact-address* 는 bu **가 보낸 HTTP 요청을 통해 전송되는 연락처 주소입니다 .** 기본값은 worm@proxy-name 입니다 .
- *sleep-time* 은 연속되는 두 요청 사이의 정지 시간입니다 . 기본값은 5 초입니다 .
- *object* 는 현재 실행 중인 bu.conf 에서 지정된 개체입니다 .
- -r n 옵션은 robot.txt 정책을 준수할 것인지 여부를 결정합니다 . 기본값은 y 입니다 .
- *conf-dir* 은 프록시 인스턴스의 구성 디렉토리입니다 . 경로는 *server_root*/proxy-server*id*/config 입니다 .

ICP(Internet Cache Protocol) 사용

ICP 정보

ICP(Internet Cache Protocol) 는 캐시가 서로 통신할 수 있도록 하는 개체 위치 프로토콜입니다 . 캐시는 ICP 를 사용하여 캐시된 URL 의 존재 및 이러한 URL 을 가져올 최적의 위치에 대한 쿼리와 응답을 보낼 수 있습니다 . 일반적인 ICP 교환에서 캐시는 이웃한 모든 캐시로 특정 URL 에 대한 ICP 쿼리를 전송합니다 . 쿼리를 받은 캐시는 해당 URL 을 포함하고 있는지 여부에 대한 ICP 응답을 전송합니다 . 해당 URL 이 없으면 "MISS" 를 , 있으면 "HIT" 를 전송합니다 .

ICP 이웃을 통한 라우팅

ICP 를 사용하면 서로 다른 관리 도메인에 위치한 프록시들 간에 통신을 할 수 있습니다 . ICP 는 한 관리 도메인에 있는 프록시 캐시가 다른 관리 도메인에 있는 프록시 캐시와 통신할 수 있도록 합니다 . 이는 여러 프록시 서버가 통신을 원하는 상황에서 효과적이지만 하나의 마스터 프록시 (프록시 배열에 있는 경우) 에서 모두 구성할 수는 없습니다 . 그림 12-3 은 다른 관리 도메인에 있는 프록시 간의 ICP 교환을 보여 줍니다 .

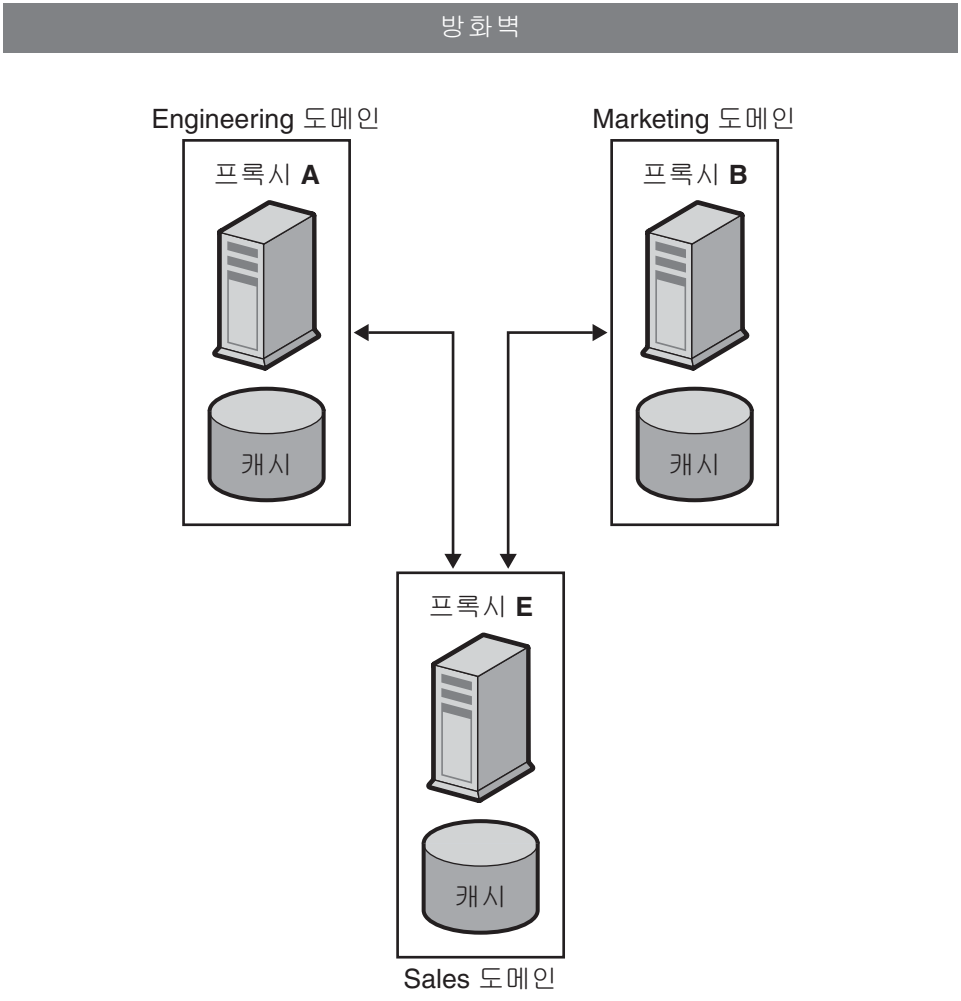
ICP 를 통해 서로 통신하는 프록시를 *이웃 (neighbor)* 이라고 합니다 . ICP 이웃은 최대 64 개입니다 . ICP 이웃에는 *상위 (parent)* 와 *동급 (sibling)*, 두 가지 유형이 있습니다 . 요청된 URL 이 다른 이웃에 없으면 상위 프록시만 원격 서버에 액세스할 수 있습니다 . ICP 상위 이웃은 없을 수 있으며 , 하나 이상일 수도 있습니다 . 상위가 *아닌* ICP 이웃은 모두 동급으로 간주됩니다 . 동급 프록시는 ICP 의 기본 경로로 표시되고 , ICP 가 이 기본 경로를 사용하는 경우에만 원격 서버에서 문서를 가져올 수 있습니다 .

*폴링 라운드*를 사용하여 쿼리를 받는 이웃의 순서를 결정할 수 있습니다 . 폴링 라운드는 ICP 쿼리 주기입니다 . 각 이웃에 대해 폴링 라운드를 할당해야 합니다 . 모든 이웃을 폴링 라운드 1 에 포함되도록 구성하면 한 주기에 모든 이웃이 쿼리를 받게 됩니다 . 즉 , 동시에 쿼리를 받게 됩니다 . 일부 이웃을 폴링 라운드 2 에 포함되도록 구성하면 폴링 라운드 1 의 모든 이웃이 먼저 쿼리를 받고 , 이중 "HIT" 를 반환한 이웃이 없으면 폴링 라운드 2 의 모든 프록시가 쿼리를 받게 됩니다 . 최대 폴링 라운드 수는 2 입니다 .

ICP 상위 이웃이 네트워크 병목 지점이 될 수 있기 때문에 폴링 라운드를 사용하여 로드를 줄일 수 있습니다 . 일반적으로 모든 동급 이웃을 폴링 라운드 1 로 구성하고 모든 상위 이웃을 폴링 라운드 2 로 구성합니다 . 이렇게 하면 로컬 프록시가 URL 을 요청한 경우 이 요청은 동급 이웃으로 먼저 전송됩니다 . 요청한 URL 을 동급 이웃에서 찾지 못하면 이 요청은 상위 이웃으로 이동합니다 . 상위 이웃에도 URL 이 없는 경우 원격 서버에서 가져옵니다 .

ICP 이웃의 각 이웃은 실행 중인 ICP 서버가 적어도 하나 이상 있어야 합니다 . 실행 중인 ICP 서버가 없는 이웃은 다른 이웃의 ICP 요청에 응답할 수 없습니다 . 프록시 서버에서 ICP 를 사용하도록 설정하면 ICP 서버가 시작됩니다 (이미 실행 중이 아닌 경우).

그림 12-3 ICP 교환



ICP 를 설정하려면 다음을 수행합니다 .

1. ICP 이웃에 상위 이웃을 추가합니다 . (이 단계는 ICP 이웃에 상위 이웃이 있도록 하려는 경우에만 필요합니다 .) ICP 이웃에 상위 이웃을 추가하는 방법에 대한 자세한 내용은 "ICP 이웃에 상위 이웃 추가 " (282 페이지) 를 참조하십시오 .
2. ICP 이웃에 동급 이웃을 추가합니다 . ICP 이웃에 동급 이웃을 추가하는 방법에 대한 자세한 내용은 "ICP 이웃에 동급 프록시 추가 " (284 페이지) 를 참조하십시오 .

3. ICP 이웃의 각 이웃을 구성합니다 . ICP 이웃 구성에 대한 자세한 내용은 " 개별 ICP 이웃 구성 " (286 페이지) 을 참조하십시오 .
4. ICP 를 사용하도록 설정합니다 . ICP 사용 설정에 대한 자세한 내용은 "ICP 사용 " (287 페이지) 을 참조하십시오 .
5. 프록시의 ICP 이웃에 동급 또는 상위 이웃이 있는 경우 ICP 이웃을 통한 라우팅을 사용하도록 설정합니다 . ICP 이웃을 통한 라우팅을 사용하도록 설정하는 방법에 대한 자세한 내용은 "ICP 이웃을 통한 라우팅 사용 " (287 페이지) 을 참조하십시오 .

ICP 이웃에 상위 이웃 추가

ICP 이웃에 상위 프록시를 추가하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure ICP 링크를 누릅니다 . Configure ICP 페이지가 표시됩니다 .
3. 페이지의 Parent List 섹션에서 Add 버튼을 누릅니다 . ICP Parent 페이지가 표시됩니다 .
4. Machine Address 필드에 ICP 이웃에 추가할 상위 프록시의 호스트 이름 또는 IP 주소를 입력합니다 .
5. ICP Port 필드에 상위 프록시가 ICP 메시지를 청취할 포트 번호를 입력합니다 .
6. Multicast Address 필드에 상위 프록시가 청취할 멀티캐스트 주소를 입력할 수 있습니다 . 멀티캐스트 주소는 여러 서버가 청취할 수 있는 IP 주소입니다 . 멀티캐스트 주소를 사용하면 Proxy Server 에서 해당 멀티캐스트 주소를 청취하는 모든 이웃이 볼 수 있는 네트워크로 쿼리를 전송할 수 있으므로 각 이웃에 별도로 쿼리를 전송할 필요가 없습니다 . 멀티캐스트 사용은 선택 사항입니다 .

참고	다른 폴링 라운드가 있는 이웃이 동일한 멀티캐스트 주소를 청취하도록 하면 안 됩니다 .

7. TTL 필드에 멀티캐스트 메시지를 전달할 서브넷의 수를 입력합니다 . TTL 이 1 로 설정되어 있으면 멀티캐스트 메시지는 로컬 서브넷에만 전달됩니다 . TTL 을 2 로 설정하면 메시지는 한 단계 이동한 범위 내의 모든 서브넷으로 전달됩니다 . 지정된 숫자에 따라 계속 마찬가지 방법이 적용됩니다 .

참고 멀티캐스트를 사용하면 관련되지 않은 두 이웃이 서로 ICP 메시지를 주고 받을 수 있습니다 . 따라서 ICP 이웃에 속한 프록시가 전송하는 ICP 메시지를 관련되지 않은 이웃이 받지 않도록 하려면 TTL 값을 낮게 설정해야 합니다 .

8. Proxy Port 필드에 상위 프록시 서버용 포트를 입력합니다 .
9. Polling Round 드롭다운 목록에서 상위 프록시를 포함할 폴링 라운드를 선택합니다 . 기본 폴링 라운드는 1 입니다 .
10. OK 를 누릅니다 .
11. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
12. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

ICP 이웃의 상위 프록시 구성 편집

상위 프록시 구성을 편집하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure ICP 링크를 선택합니다 . Configure ICP 페이지가 표시됩니다 .
3. 편집하려는 상위 프록시 옆에 있는 라디오 버튼을 누릅니다 .
4. Edit 버튼을 누릅니다 .
5. 정보를 적절히 수정합니다 .
6. OK 를 누릅니다 .
7. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
8. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

ICP 이웃에서 상위 프록시 제거

ICP 이웃에서 상위 프록시를 제거하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure ICP 링크를 선택합니다 . Configure ICP 페이지가 표시됩니다 .

3. 제거하려는 상위 프록시 옆에 있는 라디오 버튼을 누릅니다 .
4. Delete 버튼을 누릅니다 .
5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

ICP 이웃에 동급 프록시 추가

ICP 이웃에 동급 프록시를 추가하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure ICP 링크를 선택합니다 . Configure ICP 페이지가 표시됩니다 .
3. 페이지의 Sibling List 섹션에서 Add 버튼을 누릅니다 . ICP Sibling 페이지가 표시됩니다 .
4. Machine Address 필드에 ICP 이웃에 추가할 동급 프록시의 호스트 이름 또는 IP 주소를 입력합니다 .
5. Port 필드에 동급 프록시가 ICP 메시지를 청취할 포트 번호를 입력합니다 .
6. Multicast Address 필드에 동급 프록시가 청취할 멀티캐스트 주소를 입력합니다 . 멀티캐스트 주소는 여러 서버가 청취할 수 있는 IP 주소입니다 . 멀티캐스트 주소를 사용하면 Proxy Server 에서 해당 멀티캐스트 주소를 청취하는 모든 이웃이 볼 수 있는 네트워크로 쿼리를 전송할 수 있으므로 각 이웃에 별도로 쿼리를 전송할 필요가 없습니다 .

참고 다른 폴링 라운드가 있는 이웃이 동일한 멀티캐스트 주소를 청취하도록 하면 안 됩니다 .

7. TTL 필드에 멀티캐스트 메시지를 전달할 서브넷의 수를 입력합니다 . TTL 이 1 로 설정되어 있으면 멀티캐스트 메시지는 로컬 서브넷에만 전달됩니다 . TTL 을 2 로 설정하면 메시지는 한 단계 이동한 범위 내의 모든 서브넷으로 전달됩니다 .

참고 멀티캐스트를 사용하면 관련되지 않은 두 이웃이 서로 ICP 메시지를 주고 받을 수 있습니다 . 따라서 ICP 이웃에 속한 프록시가 전송하는 ICP 메시지를 관련되지 않은 이웃이 받지 않도록 하려면 TTL 값을 낮게 설정해야 합니다 .

8. Proxy Port 필드에 동급 프록시 서버용 포트를 입력합니다 .
9. Polling Round 드롭다운 목록에서 동급 프록시를 포함할 폴링 라운드를 선택합니다 . 기본 폴링 라운드는 1 입니다 .
10. OK 를 누릅니다 .
11. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
12. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

ICP 이웃의 동급 프록시 구성 편집

동급 프록시 구성을 편집하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure ICP 링크를 선택합니다 . Configure ICP 페이지가 표시됩니다 .
3. 편집하려는 동급 프록시 옆에 있는 라디오 버튼을 누릅니다 .
4. Edit 버튼을 누릅니다 .
5. 정보를 적절히 수정합니다 .
6. OK 를 누릅니다 .
7. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
8. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

ICP 이웃에서 동급 프록시 제거

ICP 이웃에서 동급 프록시를 제거하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure ICP 링크를 선택합니다 . Configure ICP 페이지가 표시됩니다 .
3. 제거하려는 동급 프록시 옆에 있는 라디오 버튼을 누릅니다 .
4. Delete 버튼을 누릅니다 .
5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

개별 ICP 이웃 구성

ICP 이웃 관계에 있는 각 이웃 또는 로컬 프록시를 구성해야 합니다 .

ICP 이웃에 있는 로컬 프록시 서버를 구성하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure ICP 링크를 선택합니다 . Configure ICP 페이지가 표시됩니다 .
3. Binding Address 필드에 이웃 서버가 바인드할 IP 주소를 입력합니다 .
4. Port 필드에 이웃 서버가 ICP 를 청취할 포트 번호를 입력합니다 .
5. Multicast Address 필드에 이웃이 청취할 멀티캐스트 주소를 입력합니다 . 멀티캐스트 주소는 여러 서버가 청취할 수 있는 IP 주소입니다 . 멀티캐스트 주소를 사용하면 Proxy Server 에서 해당 멀티캐스트 주소를 청취하는 모든 이웃이 볼 수 있는 네트워크로 쿼리를 전송할 수 있으므로 각 이웃에 별도로 쿼리를 전송할 필요가 없습니다 .

   멀티캐스트 주소와 바인드 주소가 모두 지정된 이웃은 응답 전송에는 바인드 주소를 , 청취에는 멀티캐스트 주소를 사용합니다 . 멀티캐스트 주소와 바인드 주소를 모두 지정하지 않으면 데이터 전송에 사용하는 주소는 운영 체제가 결정하게 됩니다 .
6. Default Route 필드에 "HIT" 응답을 한 이웃 프록시가 없을 때 이웃에서 요청을 라우팅할 프록시의 IP 주소 또는 이름을 입력합니다 . 이 필드에 "origin" 을 입력하거나 아무것도 입력하지 않으면 기본 경로는 원본 서버가 됩니다 .

참고 No Hit Behavior 드롭다운 목록에서 "first responding parent" 를 선택하면 Default Route 필드에 입력한 경로는 효력을 잃게 됩니다 . No Hit Behavior 기본값을 선택하면 프록시는 이 경로만 사용합니다 .

7. 두 번째 Port 필드에는 Default Route 필드에 입력한 기본 경로 컴퓨터의 포트 번호를 입력합니다 .
8. On No Hits, Route Through 드롭다운 목록에서 ICP 이웃에 요청된 URL 을 캐시에 저장하고 있는 동급 프록시가 없는 경우 이웃의 동작을 선택합니다 . 다음 중 선택할 수 있습니다 .
   - **first responding parent.** 이웃은 처음 "MISS" 응답을 전송한 상위 프록시를 통하여 요청된 URL 을 검색합니다 .
   - **default route**. 이웃은 Default Route 필드에 지정된 컴퓨터를 통하여 요청된 URL 을 검색합니다 .

9. Server Count 필드에 ICP 요청을 서비스할 프로세스의 수를 입력합니다 .
10. Timeout 필드에 이웃이 각 시도에서 ICP 응답을 대기하는 최대 시간을 입력합니다 .
11. OK 를 누릅니다 .
12. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
13. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

ICP 사용

ICP 를 사용하도록 설정하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Configure System Preferences 링크를 누릅니다 . Configure System Preferences 페이지가 표시됩니다 .
3. ICP 에 대한 Yes 라디오 버튼을 선택합니다 .
4. OK 를 누릅니다 .
5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

ICP 이웃을 통한 라우팅 사용

ICP 이웃을 통한 라우팅을 사용하도록 설정하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Routing 탭을 누릅니다 .
2. Set Routing Preferences 링크를 누릅니다 . Set Routing Preferences 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 누르고 정규식을 입력한 다음 OK 를 누릅니다 .
4. Route Through 텍스트 옆에 있는 라디오 버튼을 선택합니다 .
5. ICP 옆에 있는 확인란을 선택합니다 .
6. 클라이언트가 다른 이웃을 거치지 않고 문서를 갖고 있는 ICP 이웃에서 직접 문서를 가져오도록 하려면 Redirect 텍스트 옆에 있는 확인란을 선택합니다 .

7. OK 를 누릅니다 .

주의 현재 리디렉션을 지원하는 클라이언트는 없으므로 이 기능을 사용하지 마십시오 .

참고 프록시의 ICP 이웃에 동급 또는 상위 이웃이 있는 경우에만 ICP 이웃을 통한 라우팅을 사용하도록 설정해야 합니다 . 프록시가 다른 프록시의 상위 이웃이면서 동급 이웃이나 상위 이웃이 없는 경우에는 해당 프록시에 대해서만 ICP 를 사용하도록 설정해야 합니다 . ICP 이웃을 통한 라우팅은 사용하도록 설정할 필요가 없습니다 .

8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

프록시 배열 사용

프록시 배열 정보

분산 캐시를 위한 프록시 배열을 통해 여러 프록시가 하나의 캐시 역할을 할 수 있습니다 즉 , 배열 내의 각 프록시는 브라우저 또는 다운스트림 프록시 서버에서 가져올 수 있는 서로 다른 캐시된 URL 을 포함하게 됩니다 . 프록시 배열은 프록시 서버가 여러 개인 경우 자주 발생하는 캐시 중복 문제를 방지합니다 . 프록시 배열은 해시 기반 라우팅을 통해 프록시 배열에서 올바른 캐시로 요청을 라우팅합니다 .

프록시 배열은 또한 증분 확장성을 제공합니다 . 즉 , 프록시 배열에 다른 프록시를 추가해도 각 구성원의 캐시는 무효화되지 않습니다 . 각 구성원의 캐시에서 $1/n$(n: 배열에 있는 프록시의 수) 만큼의 URL 만 다른 구성원에게 재할당됩니다 .

프록시 배열을 통한 라우팅

프록시 배열을 통과하는 각 요청에 대해 해시 함수는 요청된 URL, 프록시 이름 , 프록시의 로드 요인을 기반으로 한 점수를 배열 내의 각 프록시에 할당합니다 . 요청은 가장 점수가 높은 프록시로 라우팅됩니다 .

URL 요청은 클라이언트와 프록시 모두 할 수 있으므로 프록시 배열을 통한 라우팅에는 두 가지 유형이 있습니다 .

클라이언트에서 프록시로의 라우팅인 경우 클라이언트는 PAC(Proxy Auto Configuration) 기법을 사용하여 통과할 프록시를 결정합니다 . 그러나 클라이언트는 표준 PAC 파일을 사용하는 대신 특수 PAC 파일로 해시 알고리즘을 계산하여 요청된 URL 에 대한 적절한 경로를 결정합니다 . 그림 12-4 는 클라이언트에서 프록시로의 라우팅을 보여 줍니다 .

그림 12-4 에서 프록시 배열의 각 구성원은 PAT 파일 업데이트에 대해 마스터 프록시를 로드 및 폴링합니다 . PAC 파일을 가진 클라이언트는 구성이 변경된 경우에만 파일을 다시 다운로드하면 됩니다 . 일반적으로 클라이언트는 재시작 시 PAC 파일을 다운로드합니다 .

프록시 서버는 관리 인터페이스를 통해 만들어진 PAT(Proxy Array Membership Table) 형식에서 자동으로 특수 PAC 파일을 생성할 수 있습니다 .

프록시에서 프록시로의 라우팅에서 프록시는 클라이언트에서 사용하는 PAC 파일 대신 PAT(Proxy Array Table) 파일을 사용하여 해시 알고리즘을 계산합니다 . PAT 파일은 프록시의 컴퓨터 이름 , IP 주소 , 포트 , 로드 요인 , 캐시 크기 등의 프록시 배열에 대한 정보를 포함한 ASCII 파일입니다 . 서버에서 해시 알고리즘을 계산하는 데 있어 런타임시 해석해야 하는 JavaScript 파일인 PAC 파일보다 PAT 파일을 사용하는 것이 훨씬 더 효율적입니다 . 하지만 대부분의 클라이언트는 PAT 파일 형식을 인식하지 못하므로 PAC 파일을 사용해야 합니다 . 그림 12-5 는 프록시에서 프록시로의 라우팅을 보여 줍니다 .

PAT 파일은 프록시 배열의 마스터 프록시에서 생성됩니다 . 프록시 관리자는 마스터 프록시 역할을 할 프록시를 결정해야 합니다 . 관리자는 마스터 프록시 서버에서 PAT 파일을 변경할 수 있으며 , 해당 프록시 배열의 다른 모든 구성원은 이 변경 사항에 대해 직접 또는 자동으로 마스터 프록시를 폴링할 수 있습니다 . 각 구성원이 이러한 변경에서 자동으로 PAC 파일을 생성하도록 구성할 수 있습니다 .

또한 계층적 라우팅을 위해 여러 프록시 배열을 체인으로 연결할 수 있습니다 . 프록시 서버가 업스트림 프록시 배열을 통해 수신 요청을 라우팅하는 경우 업스트림 프록시 배열은 상위 배열로 간주됩니다 . 상위 배열은 프록시 서버가 통과하는 프록시 배열입니다 . 즉 , 클라이언트가 Proxy X 에 문서를 요청하고 Proxy X 에 해당 문서가 없는 경우 Proxy X 는 이 요청을 원격 서버로 직접 보내는 대신 Proxy Array Y 로 전송합니다 . 따라서 Proxy Array Y 는 상위 배열입니다 . 그림 12-5 에서 Proxy Array 1 은 Proxy Array 2 의 상위 배열입니다 . Proxy Array 2 의 구성원은 상위 배열의 PAT 파일에 대한 업데이트를 로드 및 폴링합니다 . 일반적으로 상위 배열의 마스터 프록

시를 폴링합니다 . 다운로드한 PAT 파일을 사용하여 요청된 URL 에 대한 해시 알고리즘이 계산되면 Proxy Array 2 의 구성원은 Proxy Array1 에서 가장 점수가 높은 프록시에서 요청된 URL 을 가져옵니다 . 그림 12-5 에서 Proxy B 는 클라이언트에서 요청한 URL 에 대해 가장 높은 점수를 가진 프록시입니다 .

그림 12-4 클라이언트에서 프록시로의 라우팅

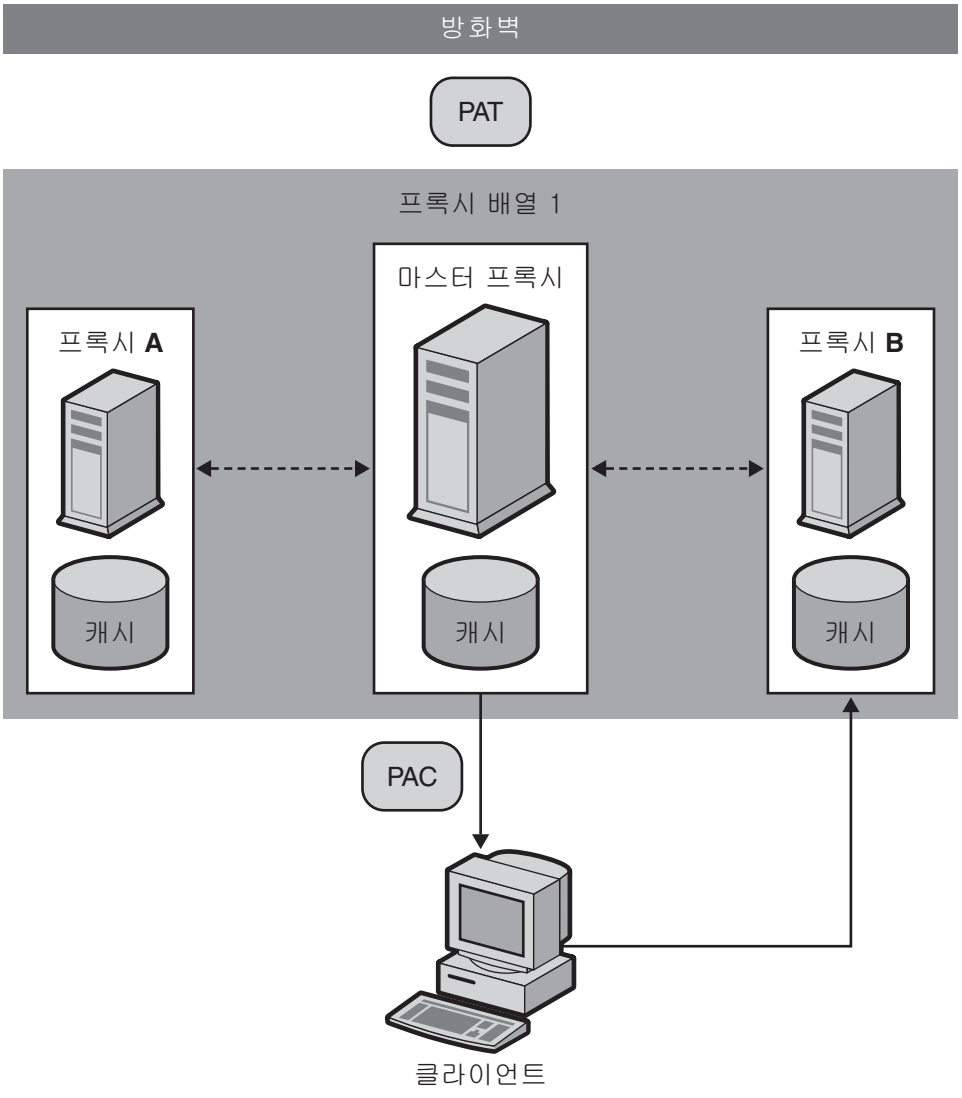
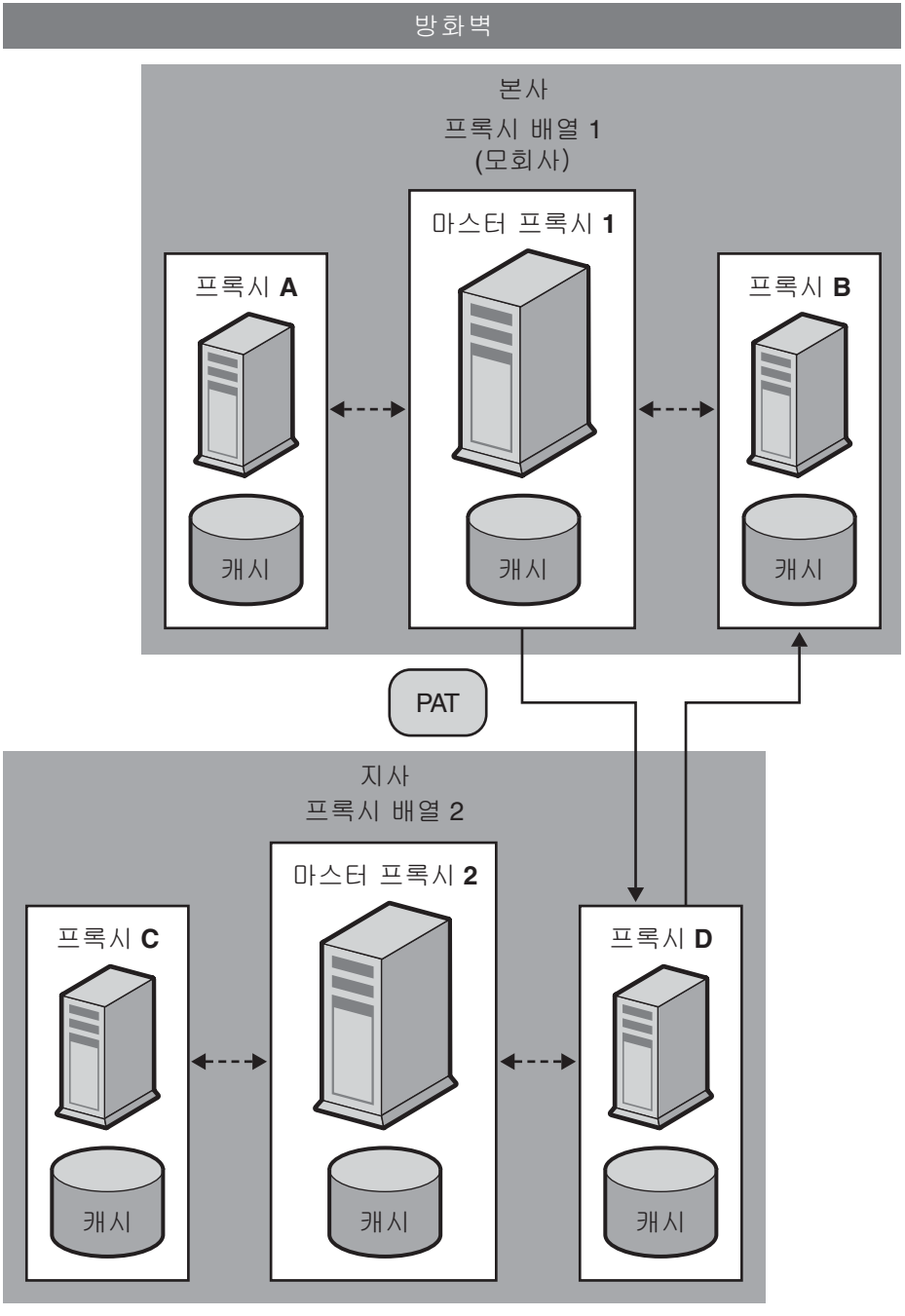


그림 12-5 프록시에서 프록시로의 라우팅



프록시 배열을 설정하려면 다음을 수행합니다 .

1. 마스터 프록시에서 다음 단계를 수행합니다 .

   a. 프록시 배열을 만듭니다 . 구성원 목록을 만드는 방법에 대한 자세한 내용은 " 프록시 배열 구성원 목록 만들기 " (293 페이지) 를 참조하십시오 .

   b. PAT 파일에서 PAC 파일을 생성합니다 . PAC 파일은 클라이언트에서 프록시로의 라우팅을 사용하는 경우에만 만들면 됩니다 . PAT 파일에서 PAC 파일을 만드는 방법에 대한 자세한 내용은 "PAT 파일에서 PAC 파일 생성 " (298 페이지) 을 참조하십시오 .

   c. 배열의 마스터 구성원을 구성합니다 . 마스터 구성원 구성에 대한 자세한 내용은 " 프록시 배열 구성원 구성 " (295 페이지) 을 참조하십시오 .

   d. 프록시 배열을 통한 라우팅을 사용하도록 설정합니다 . 프록시 배열을 통한 라우팅을 사용하도록 설정하는 방법에 대한 자세한 내용은 " 프록시 배열을 통한 라우팅 사용 " (296 페이지) 을 참조하십시오 .

   e. "/pat" URL 을 PAT 파일에 매핑하기 위한 PAT 매핑을 만듭니다 .

   f. 프록시 배열을 사용하도록 설정합니다 . 프록시 배열을 사용하도록 설정하는 방법에 대한 자세한 내용은 " 프록시 배열 사용 " (297 페이지) 을 참조하십시오 .

2. 마스터 프록시가 아닌 각 프록시에서 다음 단계를 수행합니다 .

   a. 배열의 마스터가 아닌 구성원을 구성합니다 . 마스터가 아닌 구성원 구성에 대한 자세한 내용은 " 프록시 배열 구성원 구성 " (295 페이지) 을 참조하십시오 .

   b. 프록시 배열을 통한 라우팅을 사용하도록 설정합니다 . 프록시 배열을 통한 라우팅을 사용하도록 설정하는 방법에 대한 자세한 내용은 " 프록시 배열을 통한 라우팅 사용 " (296 페이지) 을 참조하십시오 .

   c. 프록시 배열을 사용하도록 설정합니다 . 프록시 배열을 사용하도록 설정하는 방법에 대한 자세한 내용은 " 프록시 배열 사용 " (297 페이지) 을 참조하십시오 .

참고 프록시 배열이 상위 배열을 통해 라우팅하려면 상위 배열을 사용하도록 설정하고 각 구성원을 원하는 URL 에 대해 상위 배열을 통해 라우팅하도록 구성해야 합니다 . 상위 배열에 대한 자세한 내용은 " 상위 배열을 통한 라우팅 " (300 페이지) 을 참조하십시오 .

프록시 배열 구성원 목록 만들기

배열의 마스터 프록시에서만 프록시 배열 구성원 목록을 작성하고 업데이트해야 합니다 . 프록시 배열 구성원 목록은 한 번만 만들면 되며 언제든 수정할 수 있습니다 . 프록시 배열 구성원 목록을 만들면 PAT 파일이 생성되어 배열의 모든 프록시와 다운스트림 프록시에 배포됩니다 .

주의 배열의 마스터 프록시를 통해서만 프록시 배열 구성원 목록을 변경하거나 구성원을 추가해야 합니다 . 배열의 다른 구성원은 구성원 목록을 읽기만 할 수 있습니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure Proxy Array 링크를 누릅니다 . Configure Proxy Array 페이지가 표시됩니다 .
3. Array 이름 필드에 배열의 이름을 입력합니다 .
4. Reload Configuration Every 필드에 PAT 파일에 대한 폴링 간격을 분 단위로 입력합니다 .
5. Array Enabled 확인란을 누릅니다 .
6. Create 버튼을 누릅니다 .

참고 구성원 목록에 구성원을 추가하기 전에 반드시 OK 를 누르십시오 .

참고 프록시 배열이 생성되면 Create 버튼이 OK 버튼으로 바뀝니다 .

7. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
8. 프록시 배열의 각 구성원에 대해 다음 사항을 입력한 다음 OK 를 누르십시오 .
   - **Name.** 구성원 목록에 추가하려는 Proxy Server 의 이름입니다 .
   - **IP Address.** 구성원 목록에 추가하려는 Proxy Server 의 IP 주소입니다 .
   - **Port.** 구성원이 PAT 파일을 폴링하는 포트입니다 .
   - **Load Factor.** 구성원에서 라우팅해야 하는 상대적인 로드를 나타내는 정수 값입니다 .

- **Status.** 구성원의 상태를 나타냅니다. 값은 on 또는 off입니다. off로 설정된 프록시 배열 구성원의 요청은 다른 구성원을 통하여 다시 라우팅됩니다.

참고 다른 구성원을 추가하기 전에 마스터 구성원을 가장 먼저 추가해야 합니다.

참고 프록시 배열 구성원을 추가할 때 각 구성원의 정보를 입력한 다음 반드시 OK 를 누르십시오.

9. Restart Required 를 누릅니다. Apply Changes 페이지가 표시됩니다.
10. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다.

프록시 배열 구성원 목록 정보 편집

프록시 배열 구성원 목록에 있는 구성원의 정보는 언제든 변경할 수 있습니다. 마스터 프록시에서만 프록시 배열 구성원 목록을 편집할 수 있습니다.

주의 배열의 마스터 프록시를 통해서만 프록시 배열 구성원 목록을 변경하거나 구성원을 추가해야 합니다. 배열의 다른 구성원에서 목록을 수정하는 경우 변경 사항이 적용되지 않습니다.

프록시 배열 구성원에 대한 구성원 목록 정보를 편집하려면 다음을 수행합니다.

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다.
2. Configure Proxy Array 링크를 누릅니다. Configure Proxy Array 페이지가 표시됩니다.
3. Member List 에서 편집하려는 구성원 옆에 있는 라디오 버튼을 선택합니다.
4. Edit 버튼을 누릅니다. Configure Proxy Array Member 페이지가 표시됩니다.
5. 적절한 정보를 편집합니다.
6. OK 를 누릅니다.
7. Restart Required 를 누릅니다. Apply Changes 페이지가 표시됩니다.
8. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다.

참고 변경 사항을 적용하고 프록시 배열 구성원들에게 배포하려면 Configure Proxy Array 페이지에서 Configuration ID를 업데이트한 다음 OK 를 눌러야 합니다 . 기존 Configuration ID 값에 1 을 더하여 입력하면 업데이트됩니다 .

프록시 배열 구성원 삭제

프록시 배열에서 프록시 배열 구성원을 삭제합니다 . 마스터 프록시에서만 프록시 배열 구성원을 삭제할 수 있습니다 .

주의 배열의 마스터 프록시를 통해서만 프록시 배열 구성원 목록을 변경하거나 구성원을 추가해야 합니다 . 배열의 다른 구성원에서 목록을 수정하는 경우 변경 사항이 적용되지 않습니다 .

프록시 배열의 구성원을 삭제하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure Proxy Array 링크를 누릅니다 . Configure Proxy Array 페이지가 표시됩니다 .
3. Member List 에서 삭제하려는 구성원 옆에 있는 라디오 버튼을 선택합니다 .
4. Delete 버튼을 누릅니다 .

참고 변경 사항을 적용하고 프록시 배열 구성원들에게 배포하려면 Configure Proxy Array 페이지에서 Configuration ID를 업데이트한 다음 OK 를 눌러야 합니다 . 기존 Configuration ID 값에 1 을 더하여 입력하면 업데이트됩니다 .

5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

프록시 배열 구성원 구성

프록시 배열의 각 구성원은 한 번만 구성하면 되며 , 구성원에서 직접 구성해야 합니다 . 배열의 구성원에서 다른 구성원을 구성할 수 없습니다 . 또한 마스터 프록시를 구성해야 합니다 .

프록시 배열의 각 구성원을 구성하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure Proxy Array Member 링크를 누릅니다 . Configure Proxy Array Member 페이지가 표시됩니다 .
3. Proxy Array 섹션에서 적절한 라디오 버튼을 선택하여 구성원이 PAT 파일을 폴링할 것인지 여부를 표시합니다 . 다음과 같이 선택할 수 있습니다 .
   - **Non-master Member.** 구성하고 있는 구성원이 마스터 프록시가 *아니면* 이 옵션을 선택해야 합니다 . 마스터 프록시 외의 프록시 배열 구성원이 마스터 프록시에서 PAT 파일을 가져오려면 PAT 파일을 폴링해야 합니다 .
   - **Master Member.** 마스터 프록시를 구성하는 경우 이 옵션을 선택해야 합니다 . 마스터 프록시를 구성하고 있는 경우에는 PAT 파일이 로컬에 있으므로 따라서 폴링할 필요가 없습니다 .
4. Poll Host 필드에 PAT 파일을 폴링하려는 마스터 프록시의 이름을 입력합니다 .
5. Port 필드에 HTTP 요청을 받을 마스터 프록시의 포트를 입력합니다 .
6. URL 필드에 마스터 프록시에 있는 PAT 파일의 URL 을 입력합니다 . 예를 들어 , 마스터 프록시에서 PAT 파일을 /pat URL 에 맵핑하는 PAT 매핑을 만든 경우 URL 필드에 **`/pat`** 를 입력해야 합니다 .
7. Headers File 필드에는 PAT 파일에 대한 HTTP 요청 (예 : 인증 정보) 과 함께 전송되어야 하는 특수 헤더를 갖고 있는 파일의 전체 경로 이름을 입력합니다 . 이 필드는 선택 사항입니다 .
8. OK 를 누릅니다 .
9. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
10. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

프록시 배열을 통한 라우팅 사용

프록시 배열을 통한 라우팅을 사용하도록 설정하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Routing 탭을 누릅니다 .
2. Set Routing Preferences 링크를 누릅니다 . Set Routing Preferences 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 누르고 정규식을 입력한 다음 OK 를 누릅니다 .
4. Route Through 옵션을 선택합니다 .

5. 프록시 배열 및 상위 배열의 확인란을 선택합니다 .
6. 프록시 배열을 통한 라우팅을 선택한 상태에서 요청을 다른 URL 로 리디렉션하려면 Redirect 확인란을 선택합니다 . 리디렉션이란 프록시 배열의 구성원이 서비스할 수 없는 요청을 받은 경우 이 요청에 대해 연결할 프록시를 클라이언트에게 알려주는 것을 의미합니다 .
7. OK 를 누릅니다 .

참고 프록시 배열을 통한 라우팅은 현재 구성 중인 프록시 서버가 프록시 배열의 구성원인 경우에만 사용하도록 설정할 수 있습니다 . 상위 라우팅은 상위 배열이 있는 경우에만 사용하도록 설정할 수 있습니다 . 두 라우팅 옵션은 서로 독립적입니다 .

주의 현재 리디릭션을 지원하는 클라이언트는 없으므로 이 기능을 사용하지 마십시오 .

8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

프록시 배열 사용

프록시 배열을 사용 설정하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Configure System Preferences 링크를 누릅니다 . Configure System Preferences 페이지가 표시됩니다 .
3. 사용 설정하려는 배열 유형 또는 배열 (일반 프록시 배열 또는 상위 배열) 에 대해 Yes 옵션을 누릅니다 .
4. OK 를 누릅니다 .

참고 프록시 배열을 통해 라우팅하지 않는 경우 프록시 배열 옵션을 사용하지 않도록 설정하기 전에 모든 클라이언트가 특수 PAC 파일을 사용하여 제대로 라우팅하고 있는지 확인하십시오 . 프록시 배열 옵션을 사용하지 않도록 설정한 경우 Set Routing Preferences 페이지에서 명시적 프록시나 직접 연결 등의 유효한 다른 라우팅 옵션을 설정해야 합니다 .

5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

프록시 배열에서 요청 리디렉션

프록시 배열을 통해 라우팅하도록 선택한 경우 요청을 다른 URL 로 리디렉션할 것인지 여부를 지정해야 합니다 . 리디렉션이란 프록시 배열의 구성원이 서비스할 수 없는 요청을 받은 경우 이 요청에 대해 연결할 프록시를 클라이언트에게 알려주는 것을 의미합니다 .

주의 현재 리디렉션을 지원하는 클라이언트는 없으므로 이 기능을 사용하지 마십시오 .

PAT 파일에서 PAC 파일 생성

대부분의 클라이언트는 PAT 파일 형식을 인식하지 못하므로 클라이언트에서 프록시로의 라우팅에서 클라이언트는 PAC(Proxy Auto Configuration) 기법을 사용하여 통과할 프록시에 대한 정보를 받습니다 . 클라이언트는 표준 PAC 파일을 사용하는 대신 PAT 파일에서 파생된 특수 PAC 파일을 사용합니다 . 이 특수 PAC 파일은 해시 알고리즘을 계산하여 요청된 URL 에 대한 적절한 경로를 결정합니다 .

PAT 파일에서 직접 또는 자동으로 PAC 파일을 생성할 수 있습니다 . 프록시 배열의 특정 구성원에서 PAC 파일을 직접 생성하는 경우 해당 구성원은 현재 PAT 파일에 있는 정보를 기반으로 즉시 PAC 파일을 재생성합니다 . PAC 파일을 자동으로 생성하도록 프록시 배열 구성원을 구성하는 경우 구성원은 PAT 파일의 수정된 버전을 감지할 때마다 자동으로 PAC 파일을 재생성합니다 .

참고 프록시 서버에서 프록시 배열 기능을 사용하지 않는 경우 Create / Edit Autoconfiguration File 페이지를 사용하여 PAC 파일을 생성해야 합니다 . 자세한 내용은 제 17 장 , 349 페이지의 " 클라이언트 자동 구성 파일 사용 " 을 참조하십시오 .

PAT 파일에서 PAC 파일 직접 생성

참고 PAC 파일은 마스터 프록시에서만 생성할 수 있습니다 .

PAT 파일에서 PAC 파일 직접 생성하려면 다음을 수행합니다 .

1. 마스터 프록시의 Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure Proxy Array 링크를 누릅니다 . Configure Proxy Array 페이지가 표시됩니다 .
3. Generate PAC 버튼을 누릅니다 . PAC Generation 페이지가 표시됩니다 .
4. PAC 파일에서 사용자 지정 로직을 사용하려면 PAC 파일 생성에 포함할 사용자 지정 로직이 포함된 파일의 이름을 Custom logic file 필드에 입력합니다 . 이 로직은 `FindProxyForURL` 함수의 프록시 배열 선택 로직 앞에 삽입됩니다 . 이 함수는 보통 프록시 배열을 통과할 필요가 없는 로컬 요청에 사용됩니다 .

   Configure Proxy Array Member 페이지에서 사용자 지정 로직 파일을 이미 입력한 경우 , 이 필드에 자동으로 입력됩니다 . 사용자 지정 로직 파일 이름은 편집할 수 있습니다 . 이 경우 변경 사항은 Configure Proxy Array Member 페이지로도 전송됩니다 .
5. Default Route 필드에 배열의 프록시를 사용할 수 없는 경우 클라이언트가 이동해야 하는 경로를 입력합니다 .

   Configure Proxy Array Member 페이지에서 기본 경로를 이미 입력한 경우 이 필드에 자동으로 입력됩니다 . 기본 경로는 편집할 수 있습니다 . 이 경우 변경 사항은 Configure Proxy Array Member 페이지로도 전송됩니다 .
6. OK 를 누릅니다 .
7. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
8. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

PAT 파일에서 PAC 파일 자동 생성

변경 사항이 감지될 때마다 PAT 파일에서 PAC 파일을 자동으로 생성하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. Configure Proxy Array Member 링크를 누릅니다 . Configure Proxy Array Member 페이지가 표시됩니다 .
3. Auto-generate PAC File 확인란을 선택합니다 .

4. PAC 파일에서 사용자 지정 로직을 사용하려면 PAC 파일 생성에 포함할 사용자 지정 로직이 포함된 파일의 이름을 Custom Logic File 필드에 입력합니다 . 이 로직은 `FindProxyForURL` 함수의 프록시 배열 선택 로직 앞에 삽입됩니다 .

   Configure Proxy Array 페이지에서 사용자 지정 로직 파일을 이미 입력하고 저장한 경우 , 이 필드에 자동으로 입력됩니다 . 사용자 지정 로직 파일 이름은 편집할 수 있습니다 . 이 경우 변경 사항은 Configure Proxy Array 페이지로도 전송됩니다 .

5. Default Route 필드에 배열의 프록시를 사용할 수 없는 경우 클라이언트가 이동해야 하는 경로를 입력합니다 .

6. Configure Proxy Array 페이지에서 기본 경로를 이미 입력하고 저장한 경우 , 이 필드에 자동으로 입력됩니다 . 기본 경로는 편집할 수 있습니다 . 이 경우 변경 사항은 Configure Proxy Array 페이지로도 전송됩니다 .

7. OK 를 누릅니다 .

8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .

9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

상위 배열을 통한 라우팅

프록시 또는 프록시 배열 구성원이 원격 서버로 직접 이동하지 않고 업스트림 상위 배열을 통해 라우팅하도록 구성할 수 있습니다 .

프록시 또는 프록시 배열 구성원이 상위 배열을 통해 라우팅하도록 구성하려면 다음을 수행합니다 .

1. 상위 배열을 사용하도록 설정합니다 . 배열을 사용하도록 설정하는 방법에 대한 자세한 내용은 " 프록시 배열 사용 " (297 페이지) 을 참조하십시오 .

2. 상위 배열을 통한 라우팅을 사용하도록 설정합니다 . 배열을 통한 라우팅을 사용하도록 설정하는 방법에 대한 자세한 내용은 " 프록시 배열을 통한 라우팅 사용 " (296 페이지) 을 참조하십시오 .

3. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .

4. Configure Proxy Array Member 링크를 누릅니다 . Configure Proxy Array Member 페이지가 표시됩니다 .

5. 페이지의 Parent Array 섹션에 있는 Poll Host 필드에 PAT 파일을 폴링할 상위 배열에 있는 프록시의 호스트 이름을 입력합니다 . 보통 해당 상위 배열의 마스터 프록시입니다 .

6. 페이지의 Parent Array 부분에 있는 Port 필드에 PAT 파일을 폴링할 상위 배열에 있는 프록시의 포트 번호를 입력합니다 .

7. URL 필드에 마스터 프록시에 있는 PAT 파일의 URL 을 입력합니다 . 마스터 프록시에서 PAT 매핑을 만든 경우 이 URL 필드에 매핑을 입력해야 합니다 .
8. 양식의 Parent Array 섹션에 있는 Headers File 필드에는 PAT 파일에 대한 HTTP 요청 (예 : 인증 정보) 과 함께 전송되어야 하는 특수 헤더를 갖고 있는 파일의 전체 경로 이름을 입력합니다 . 이 필드는 선택 사항입니다 .
9. OK 를 누릅니다 .
10. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
11. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

상위 배열 정보 확인

프록시 배열이 상위 배열을 통과하여 라우팅하는 경우 , 상위 배열의 구성원에 대한 정보가 있어야 합니다 . 이 정보는 상위 배열에서 PAT 파일 형식으로 전송됩니다 . PAT 파일에 포함된 정보가 View Parent Array Configuration 페이지에 표시됩니다.

상위 배열 정보를 확인하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Caching 탭을 누릅니다 .
2. View Parent Array Configuration 링크를 누릅니다 . View Parent Array Configuration 페이지가 표시됩니다 .
3. 정보를 확인합니다 .

13 장

프록시를 통한 컨텐트 필터링

이 장에서는 URL 을 필터링하여 프록시 서버에서 해당 URL 에 대한 액세스 또는 클라이언트로 반환하는 HTML 과 JavaScript 컨텐트의 수정을 허용하지 않도록 하는 방법을 설명합니다 . 또한 이 장에서는 클라이언트가 사용하는 웹 브라우저 (사용자 에이전트) 를 기반으로 프록시를 통한 액세스를 제한하는 방법을 설명합니다 .

프록시 서버에서는 URL 필터 파일을 사용하여 서버가 지원하는 URL을 결정할 수 있습니다 . 예를 들어 지원할 URL 을 와일드카드 패턴으로 직접 입력하는 대신 제한하려는 URL 이 포함된 텍스트 파일을 하나 만들거나 구매할 수 있습니다 . 이 기능으로 하나의 URL 파일을 만들어 서로 다른 다양한 프록시 서버에서 사용할 수 있습니다 .

MIME 유형을 기반으로 URL 을 필터링할 수도 있습니다 . 예를 들어 프록시에서 HTML 및 GIF 파일을 캐시하고 보낼 수 있지만 컴퓨터 바이러스의 위험이 있는 이진 파일 또는 실행 파일은 가져오지 못하게 할 수 있습니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

URL 필터링

URL 파일을 사용하여 프록시 서버에서 가져올 내용을 구성할 수 있습니다 . URL 목록을 설정하면 Proxy Server 가 항상 지원할 부분과 지원하지 않을 부분을 지정할 수 있습니다 .

예를 들어 어린이용 컨텐트를 제공하고 프록시 서버를 실행하는 인터넷 서비스 제공자는 어린이가 볼 수 있도록 승인된 URL 의 목록을 설정할 수 있습니다 . 이 경우 프록시 서버가 승인된 URL 만 가져오도록 할 수 있습니다 . 클라이언트가 지원되지 않는 URL 로 이동하려고 하면 프록시에서 기본 "Forbidden" 메시지를 반환하도록 하거나 또는 클라이언트가 해당 URL 을 액세스할 수 없는 이유를 설명하는 사용자 지정 메시지를 만들 수 있습니다 .

URL 을 기반으로 액세스를 제한하려면 허용 또는 제한하려는 URL 파일을 만들어야 합니다 . Server Manager 에서 이 작업을 수행할 수 있습니다 . 일단 파일이 만들어지면 이러한 제한 사항을 설정할 수 있습니다 . 다음 절에서 이 절차를 설명합니다 .

URL 필터 파일 만들기

필터 파일은 URL 목록을 포함한 파일입니다 . 프록시 서버가 사용하는 필터 파일은 다음과 같은 패턴의 URL 이 행으로 나열된 일반 텍스트 파일입니다 .

프로토콜 :// 호스트 : 포트 / 경로 / 파일이름

URL 을 이루는 세 부분인 프로토콜 , 호스트 : 포트 , 경로 / 파일이름에서 모두 정규식을 사용할 수 있습니다 . 예를 들어 netscape.com 도메인으로 가는 모든 프로토콜에 대한 URL 패턴을 만들려면 파일에 다음과 같은 행을 넣으면 됩니다 .

```
.*://.*\.example\.com/.*
```

이 행은 포트 번호를 지정하지 않은 경우에만 작동합니다 . 정규식에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 에 있는 " 정규식에 대한 이해 " 항목을 참조하십시오 .

Server Manager 를 사용하지 않고 직접 파일을 만들 때는 Server Manager 페이지를 사용해서 빈 파일을 만들고 텍스트를 추가하거나 정규식을 포함하는 텍스트로 대체하는 것이 편리합니다 .

필터 파일을 만들려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Restrict URL Filter Access 링크를 누릅니다 . Restrict URL Filter Access 페이지가 표시됩니다 .

3. Create/Edit 버튼 옆의 드롭다운 목록에서 New Filter 를 선택합니다 .
4. 드롭다운 목록 오른쪽에 있는 텍스트 입력란에 필터의 이름을 입력하고 Create/Edit 버튼을 누릅니다 . Filter Editor 페이지가 표시됩니다 .
5. 스크롤 가능한 Filter Content 텍스트 입력란에 URL 이나 URL 의 정규식을 입력합니다 . Reset 버튼을 누르면 이 필드의 모든 텍스트를 지웁니다 .

   정규식에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 에 있는 " 정규식에 대한 이해 " 항목을 참조하십시오 .
6. OK 를 누릅니다 .

프록시 서버에서 파일을 생성하고 Restrict URL Filter Access 페이지로 돌아갑니다 . 필터 파일은 `proxy-server`*id*`/conf_bk` 디렉토리에 만들어집니다 .

필터 파일에 대한 기본 액세스 설정

사용할 URL 이 담긴 필터 파일이 만들어지면 , 이 URL 에 대한 기본 액세스를 설정할 수 있습니다 .

필터 파일에 대한 기본 액세스를 설정하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Restrict URL Filter Access 링크를 누릅니다 . Restrict URL Filter Access 페이지가 표시됩니다 .
3. 필터와 함께 사용할 템플릿을 선택합니다 .

   일반적으로 전체 프록시 서버에 사용할 필터 파일을 만들지만 HTTP 용과 FTP 용 필터 파일을 구분할 때도 있습니다 .
4. URL Filter To Allow 목록을 사용하여 프록시 서버에서 지원할 URL 이 포함된 필터 파일을 선택합니다 .
5. URL Filter To Deny 목록을 사용하여 프록시 서버에서 액세스를 거부할 URL 이 포함된 필터 파일을 선택합니다 .
6. 거부된 URL 을 요청한 클라이언트에게 프록시 서버가 반환할 텍스트를 선택합니다 . 선택할 수 있는 옵션은 다음 두 가지 입니다 .
   - Proxy Server 가 생성하는 기본 "Forbidden" 응답을 전송합니다 .
   - 사용자 지정 텍스트를 담은 HTML 파일이나 다른 텍스트를 전송합니다 . 텍스트 입력란에 이 파일의 절대 경로를 입력합니다 .
7. OK 를 누릅니다 .

8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

컨텐트 URL 재작성

Proxy Server 4 에는 클라이언트로 반환되는 컨텐트를 확인하고 패턴 (예 : URL) 을 다른 문자열로 대체할 수 있는 기능이 있습니다 . 구성할 수 있는 매개 변수는 원본 문자열과 대상 문자열 , 두 가지입니다 . Proxy Server 는 원본 문자열과 일치하는 텍스트를 찾아 대상 문자열에 있는 텍스트로 대체합니다 . 이 기능은 역방향 프록시 모드에서만 작동합니다 .

URL 재작성 패턴을 만들려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Set Content URL Rewriting 링크를 누릅니다 . Set Content URL Rewriting 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 정규식을 지정합니다 . 정규식에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 에 있는 " 정규식에 대한 이해 " 항목을 참조하십시오 .
4. Source Pattern 텍스트 입력란에 원본 문자열을 지정합니다 .
5. Destination Pattern 텍스트 입력란에 대상 문자열을 지정합니다 .
6. MIME Pattern 텍스트 입력란에 컨텐트 유형을 지정합니다 .
7. OK 를 누릅니다 .
8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

URL 재작성 패턴을 편집하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Set Content URL Rewriting 링크를 누릅니다 . Set Content URL Rewriting 페이지가 표시됩니다 .
3. 편집하려는 URL 재작성 패턴 옆에 있는 Edit 링크를 누릅니다 .
4. OK 를 누릅니다 .
5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

URL 재작성 패턴을 삭제하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Set Content URL Rewriting 링크를 누릅니다 . Set Content URL Rewriting 페이지가 표시됩니다 .
3. 삭제하려는 URL 재작성 패턴 옆에 있는 Remove 링크를 누릅니다 . OK 를 눌러 삭제를 확인합니다 .
4. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
5. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

특정 웹 브라우저에 대한 액세스 제한

클라이언트의 웹 브라우저 유형과 버전에 기반하여 프록시 서버에 대한 액세스를 제한할 수 있습니다 . 제한 사항은 모든 웹 브라우저가 요청을 할 때 사용하는 user-agent 헤더를 기반으로 적용됩니다 .

클라이언트의 웹 브라우저에 기반하여 프록시에 대한 액세스를 제한하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Set User-Agent Restriction 링크를 누릅니다 . Set User-Agent Restriction 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Proxy Server 에서 지원하도록 하려는 브라우저에 대한 user-agent 문자열에 일치하는 정규식을 입력합니다 . 두 개 이상의 클라이언트를 지정하려면 정규식을 괄호 안에 넣고 | 문자를 사용하여 여러 항목을 분리합니다 . 정규식에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 에 있는 " 정규식에 대한 이해 " 항목을 참조하십시오 .
4. Allow Only User-Agents Matching 옵션을 선택합니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

요청 차단

업로드 컨텐트 유형에 기반하여 파일 업로드 및 다른 요청을 차단해야 하는 경우가 있습니다 .

MIME 유형에 기반하여 요청을 차단하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Set Request Blocking 링크를 누릅니다. Set Request Blocking 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 Regular Expression 버튼을 누르고 정규식을 입력한 다음 OK 를 누릅니다 .
4. 차단하고자 하는 요청 유형에 해당하는 라디오 버튼을 누릅니다 . 다음 옵션에서 선택하십시오 .
   - Disabled - 요청 차단을 사용하지 않도록 합니다 .
   - Multipart MIME (File Upload) - 모든 파일 업로드를 차단합니다 .
   - MIME Types Matching Regular Expression - 입력한 정규식과 일치하는 MIME 유형에 대한 요청을 차단합니다 . 정규식에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 에 있는 " 정규식에 대한 이해 " 항목을 참조하십시오 .
5. 모든 클라이언트의 요청을 차단할지, 입력한 정규식과 일치하는 user-agent만을 차단할지 선택합니다 .
6. 요청을 차단할 메소드를 라디오 버튼에서 선택합니다 . 옵션은 다음과 같습니다 .
   - Any Method With Request Body - 메소드에 관계없이 요청 본문이 있는 모든 요청을 차단합니다 .
   - Only For:
     POST - POST 메소드를 사용한 파일 업로드 요청을 차단합니다 .
     PUT - PUT 메소드를 사용한 파일 업로드 요청을 차단합니다 .
   - Methods Matching Regular Expression- 입력한 메소드를 사용한 모든 파일 업로드 요청을 차단합니다 .
7. OK 를 누릅니다 .
8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

송신 헤더 제거

보안 상의 이유로 요청에서 송신 헤더를 제거하도록 프록시 서버를 구성할 수 있습니다 . 예를 들어 , From 헤더를 제거하여 사용자의 전자 메일 주소가 유출되지 않도록 하거나 , User-Agent 헤더를 필터링하여 외부 서버가 사용자의 회사에서 사용하는 웹 브라우저의 종류를 알 수 없도록 할 수 있습니다 . 또한 요청이 인터넷으로 전달되기 전에 로깅이나 클라이언트 관련 헤더를 제거하여 인트라넷 내에서만 사용되도록 할 수 있습니다 .

이 기능은 프록시 자체에서 특수하게 처리 되거나 생성되는 헤더 또는 프로토콜의 적절한 작동을 위해 필요한 헤더 (예 : If-Modified-Since 나 Forwarded) 에 대해서는 효과가 없습니다 .

전달된 헤더가 프록시에서 외부로 유출되는 것을 막을 방법은 없지만 , 이는 보안상의 문제는 되지 않습니다 . 원격 서버가 연결되는 프록시 호스트를 감지하고 , 프록시 체인에서 , 내부 프록시에서 전달된 헤더는 외부 프록시에 의해 표시되지 않을 수 있기 때문입니다 . 내부 프록시나 호스트 이름이 원격 서버에 공개되지 않도록 하려면 이 방식으로 서버를 설정하면 됩니다 .

송신 헤더를 제거하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Suppress Outgoing Headers 링크를 누릅니다 . Suppress Outgoing Headers 페이지가 표시됩니다 .
3. 제거할 요청 헤더 목록을 쉼표로 구분하여 Suppress Headers 텍스트 입력란에 입력합니다 . 예를 들어 , From 과 User-Agent 헤더를 제거하려면 텍스트 입력란에 **`from,user-agent`** 를 입력합니다 . 헤더를 입력할 때는 대소문자를 구분할 필요가 없습니다 . 정규식에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 에 있는 " 정규식에 대한 이해 " 항목을 참조하십시오 .
4. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
5. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

MIME 유형별 필터링

MIME 유형에 일치하는 특정 파일을 차단하도록 프록시 서버를 구성할 수 있습니다. 예를 들어 , 프록시 서버에서 실행 파일이나 이진 파일을 차단하도록 설정하여 프록시 서버를 사용하는 모든 클라이언트가 혹시 있을지 모르는 컴퓨터 바이러스를 다운받을 수 없도록 할 수 있습니다 .

프록시 서버가 새 MIME 유형을 지원하도록 하려면 Server Manager 에서 Preferences > Create/Edit MIME Types 를 선택하고 해당 유형을 추가합니다 . MIME 유형을 만드는 방법에 대한 자세한 내용은 " 새 MIME 유형 만들기 " (140 페이지) 를 참조하십시오 .

필터링할 MIME 형식을 템플릿과 결합하여 특정한 URL 에서의 특정한 MIME 형식을 차단할 수 있습니다 . 예를 들어 .edu 도메인 내의 컴퓨터에서 유입되는 실행 파일을 차단할 수 있습니다 .

MIME 유형별로 필터링하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Set MIME Filters 링크를 누릅니다 . Set MIME Filters 페이지가 표시됩니다 .
3. MIME 유형을 필터링하는 데 사용할 템플릿을 선택하거나 또는 전체 서버를 편집 중인지 확인합니다 .
4. Current filter 텍스트 입력란에 차단하려는 MIME 유형과 일치하는 정규식을 입력할 수 있습니다 .

   예를 들어 , 모든 응용 프로그램을 필터링하려면 정규식으로 **`application/.*`** 을 입력합니다 . 이는 모든 응용 프로그램 유형에 대해 각 MIME 형식을 검사하는 것보다 빠른 방법입니다 . 정규식은 대소문자를 구별하지 않습니다 . 정규식에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 에 있는 " 정규식에 대한 이해 " 항목을 참조하십시오 .
5. 필터링하려는 MIME 유형을 선택합니다 . 클라이언트가 차단된 파일에 대한 액세스를 시도하면 프록시 서버는 "403 Forbidden" 메시지를 반환합니다 .
6. OK 를 누릅니다 .
7. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
8. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

HTML 태그별 필터링

프록시 서버에서 파일을 클라이언트로 전달하기 전에 필터링할 HTML 태그를 지정할 수 있습니다 . 이 기능을 사용하면 HTML 파일에 포함된 Java 애플릿이나 JavaScript 와 같은 개체를 필터링할 수 있습니다 . HTML 태그를 필터링하려면 HTML 태그의 시작과 끝을 지정합니다 . 이렇게 하면 프록시는 파일을 클라이언트로 보내기 전에 해당 태그 사이의 모든 개체와 텍스트를 빈칸으로 대체합니다 .

참고 만약 프록시가 해당 리소스를 캐시하도록 구성되었을 경우 프록시는 편집되지 않은 원본 파일을 캐시에 저장합니다 .

HTML 태그를 필터링하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Set HTML Tag Filters 링크를 누릅니다 . Set HTML Tag Filters 페이지가 표시됩니다 .
3. 수정하려는 템플릿을 선택합니다. HTTP를 선택할 수도 있고 특정 URL(예: `.edu` 도메인 내의 호스트에 있는 URL) 만 지정하는 템플릿을 선택할 수도 있습니다 .
4. 필터링할 모든 기본 HTML 태그의 필터 확인란을 선택합니다 . 기본 태그에는 다음이 포함됩니다 .
   - Java 애플릿에 사용되는 APPLET 태그
   - JavaScript 코드의 시작을 나타내는 SCRIPT 태그
   - 인라인 이미지 파일을 지정하는 IMG 태그
5. 이 밖에도 필터링할 HTML 태그를 직접 입력할 수 있습니다. 입력할 때는 시작과 끝 HTML 태그를 모두 입력해야 합니다 .

   예를 들어 형식을 필터링할 때는 Start Tag 텍스트 상자에 **`FORM`** 을 (대소문자를 구별하지 않습니다), End Tag 텍스트 상자에 **`/FORM`** 을 입력합니다 . 필터링할 태그가 `OBJECT` 및 `IMG` 와 같은 끝 태그가 없는 경우 , End Tag 텍스트 상자는 비워둡니다 .
6. OK 를 누릅니다 .
7. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
8. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

내용 압축으로 서버 구성

Proxy Server 는 HTTP 내용 압축을 지원합니다 . 내용 압축을 사용하면 클라이언트로 전송 속도가 빨라지고 하드웨어 비용을 증가시키지 않고 더 큰 용량의 내용을 서비스할 수 있습니다 . 내용 압축은 내용 다운로드 시간을 떨어뜨리지만 전화 접속 및 높은 수준의 트래픽 연결 사용자는 더 많은 혜택을 누릴 수 있습니다 .

내용 압축을 사용하여 Proxy Server 는 압축된 데이터를 전송하고 브라우저에게 전송 중에 데이터를 압축 해제할 것을 지시하여 전송된 데이터 양을 감소시키고 페이지 표시 속도를 높입니다 .

필요 시 내용 압축으로 서버 구성

전송 중에 전송 데이터를 압축하도록 Proxy Server 를 구성할 수 있습니다 . 동적으로 생성된 HTML 페이지는 사용자가 요청할 때까지는 존재하지 않습니다 .

필요 시 내용을 압축하도록 서버를 구성하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Filters 탭을 누릅니다 .
2. Compress Content on Demand 링크를 누릅니다 . Compress Content on Demand 페이지가 표시됩니다 .
3. 드롭다운 목록에서 리소스를 선택하거나 정규식을 지정합니다 . 정규식에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 에 있는 " 정규식에 대한 이해 " 항목을 참조하십시오 .
4. 다음 정보를 지정합니다 .
   - **Activate Compress Content on Demand?** 서버가 선택된 리소스에 대해 미리 압축된 내용을 서비스해야 하는지 선택합니다 .
   - **Vary Header.** Vary:Accept-encoding 헤더를 삽입할지 여부를 지정합니다 . yes 또는 no 를 선택합니다 . yes 로 설정하면 압축된 버전의 파일이 선택되는 경우 항상 Vary:Accept-encoding 헤더가 삽입됩니다 .

     no 로 설정하면 Vary:Accept-encoding 헤더가 삽입되지 않습니다 .

     기본값은 yes 로 설정됩니다 .
   - **Fragment Size.** 압축 라이브러리 (`zlib`) 가 한 번에 압축할 양을 제어하는 데 사용하는 메모리 조각의 크기를 바이트 단위로 지정합니다 . 기본값은 8096 입니다 .
   - **Compression Level.** 압축의 수준을 지정합니다 . 1 ~ 9 사이 값을 선택합니다 . 값 1 은 속도가 최고이고 9 는 압축율이 최고입니다 . 기본값은 6 으로 속도와 압축율이 조화된 값입니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

14 장

역방향 프록시 사용

이 장에서는 Proxy Server 를 역방향 프록시로 사용하는 방법에 대해 설명합니다 . 역방향 프록시는 프록시 서버의 특정한 대체 사용 이름입니다 . 방화벽 외부에서 외부 클라이언트에게 보안 컨텐트 서버를 표시하는 데 사용할 수 있습니다 . 이를 통해 기업 외부로부터 서버의 데이터에 대한 직접적이고 모니터되지 않는 액세스를 방지합니다 . 또한 복제용으로 사용할 수도 있습니다 . 즉 로드 밸런싱을 위해 사용량이 많은 서버의 전면에 여러 프록시를 연결할 수 있습니다 . 이 장에서는 Proxy Server 를 방화벽 내부 또는 외부에서 대체 사용하는 방법에 대해 설명합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

역방향 프록시 작동 원리

역방향 프록시에는 두 가지 모델이 있습니다 . 한 가지 모델은 Proxy Server 의 보안 기능을 활용하여 트랜잭션을 처리하고 , 다른 하나는 캐싱 기능을 사용하여 사용량이 많은 서버에서 로드 밸런싱을 수행합니다 . 두 모델은 모두 엄격히 방화벽에서 작동하지 않는다는 점에서 기존 프록시 사용과는 다릅니다 .

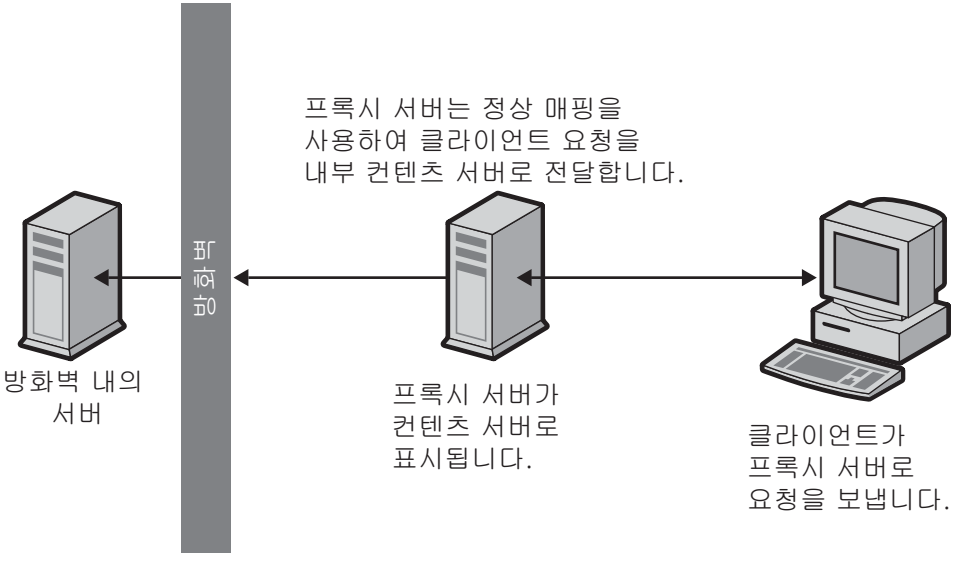
서버 대용로서의 프록시

신용 카드 번호 데이터베이스와 같이 보안을 유지해야 하는 중요 정보가 들어 있는 컨텐트 서버가 있을 경우 방화벽 외부에 프록시를 컨텐트 서버의 대용으로 설정할 수 있습니다. 외부 클라이언트가 컨텐트 서버에 액세스를 시도하면 컨텐트 서버가 아닌 프록시 서버로 전달됩니다 . 컨텐트 서버의 실제 컨텐츠는 방화벽 내에서 안전하게 유지됩니다 . 프록시 서버는 방화벽 외부에 상주하므로 클라이언트에게는 컨텐트 서버처럼 보입니다 .

클라이언트가 해당 사이트에 전달한 요청은 프록시 서버로 갑니다 . 그러면 프록시 서버는 방화벽의 특정 통로를 통해 클라이언트의 요청을 컨텐트 서버로 전달합니다. 컨텐트 서버는 다시 이 통로를 통해 프록시로 결과를 전달합니다 . 프록시는 마치 실제 컨텐트 서버인 것처럼 검색된 정보를 클라이언트에 전달합니다 (그림 14-1 참조). 컨텐트 서버가 오류 메시지를 반환하면 프록시 서버는 메시지를 클라이언트에 전송하기 전에 메시지를 가로채 헤더에 열거된 URL 을 변경합니다 . 이를 통해 외부 클라이언트가 내부 컨텐트 서버로의 재지정 URL 을 알지 못하도록 합니다 .

프록시는 이런 방식으로 보안 데이터베이스와 악성 공격 가능성 간을 차단하는 장벽 역할을 수행합니다 . 공격 성공 가능성이 희박한 상황에서 , 공격자는 전체 데이터베이스에 대한 액세스를 수행하는 것이 아니라 단일 트랜잭션과 관련이 있는 정보에만 제한적으로 액세스하게 될 가능성이 높습니다 . 방화벽 통로에는 프록시 서버만 액세스할 수 있으므로 권한 없는 사용자는 실제 컨텐트 서버에 가까이 갈 수 없습니다 .

그림 14-1 역방향 프록시가 실제 컨텐트 서버처럼 보임



다른 시스템이 들어오거나 나가는 것을 허용하지 않으면서 특정 포트의 특정 서버 (이 경우 할당된 포트의 프록시) 가 방화벽을 통해 액세스하는 것을 허용하도록 방화벽 라우터를 구성할 수 있습니다 .

보안 역방향 프록시

보안 역방향 프록시는 프록시 서버와 다른 시스템 간의 하나 이상의 연결이 SSL(Secure Sockets Layer) 프로토콜을 사용하여 데이터를 암호화할 때 발생합니다 .

보안 역방향 프록시는 여러 용도로 사용할 수 있습니다 .

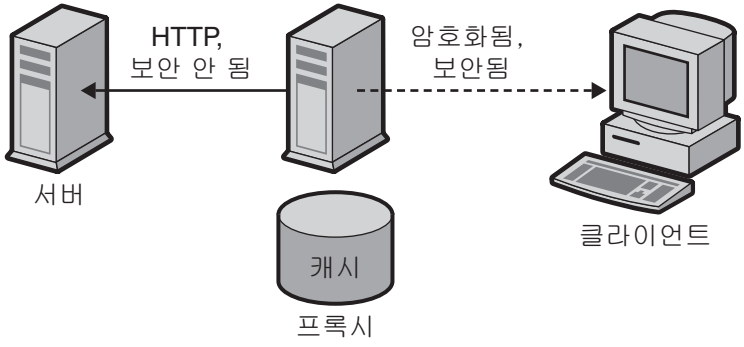
- 방화벽 외부의 프록시 서버에서 방화벽 내부의 컨텐트 서버로의 암호화된 연결을 제공합니다 .
- 클라이언트가 프록시 서버에 안전하게 연결할 수 있으므로 신용 카드 번호와 같은 정보를 안전하게 전달할 수 있습니다 .

보안 역방향 프록시를 사용하면 데이터 암호화와 관련한 오버헤드가 있으므로 각 보안 연결이 더 느려집니다 . 그러나 SSL 이 캐시 메커니즘을 제공하므로 두 연결 당사자는 이후 연결에서 이전에 협상한 보안 매개 변수를 재활용하여 오버헤드를 상당히 줄일 수 있습니다 .

세 가지 방법으로 보안 역방향 프록시를 구성할 수 있습니다 .

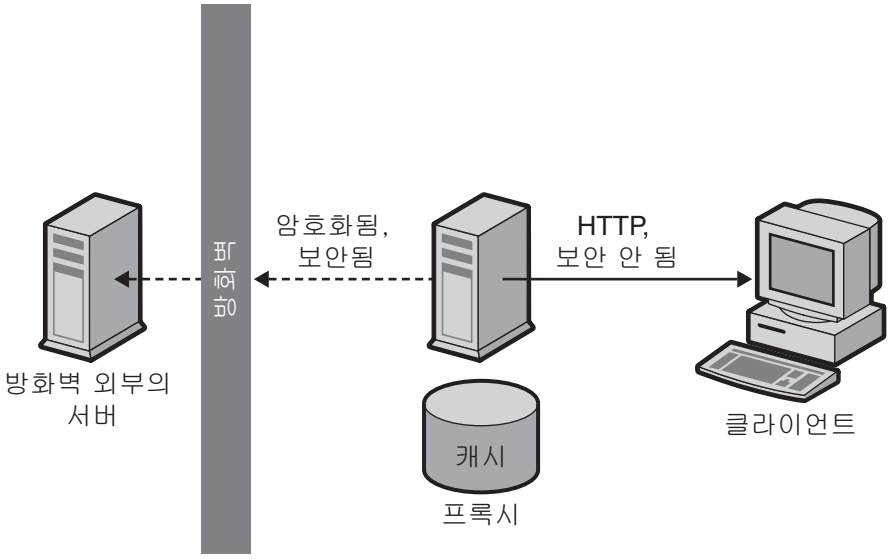
보안 클라이언트에서 프록시 . 이 시나리오는 프록시와 컨텐트 서버 간에 교환하는 정보를 권한 없는 사용자가 액세스할 수 있는 가능성이 거의 없는 경우에 효과적입니다 (그림 14-2 참조).

그림 14-2 보안 클라이언트에서 프록시 연결



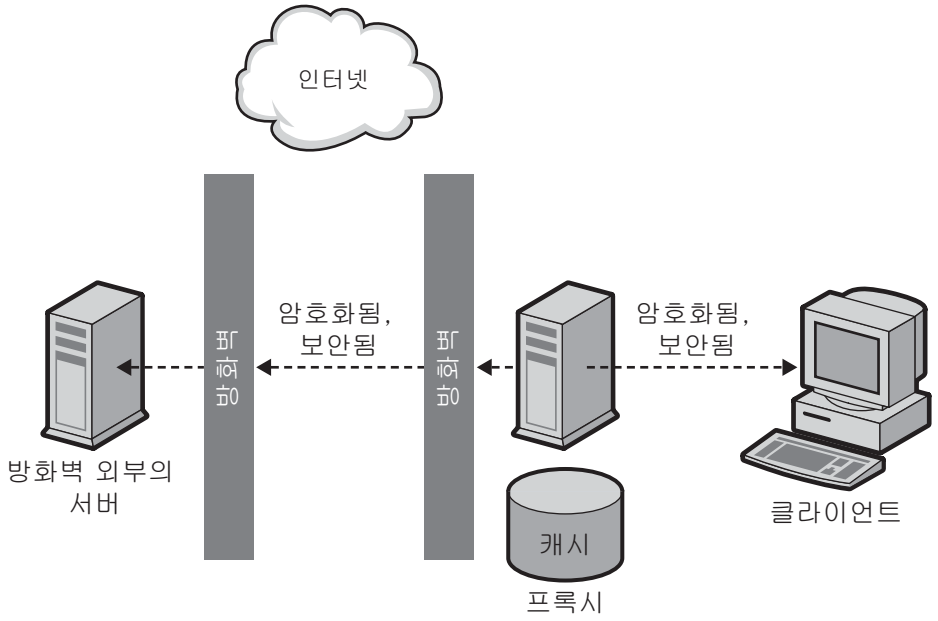
보안 프록시에서 컨텐트 서버 . 이 시나리오는 방화벽 내부에 클라이언트가 있고 방화벽 외부에 컨텐트 서버가 있을 때 효과적입니다 . 이 시나리오에서는 프록시 서버가 사이트 간의 보안 채널 역할을 수행합니다 (그림 14-3 **참조).**

그림 14-3 보안 프록시에서 컨텐트 서버 연결



- **보안 클라이언트에서 프록시 , 보안 프록시에서 컨텐트 서버** . 이 시나리오는 서버 간에 교환하는 정보 , 프록시 및 클라이언트에 보안이 필요한 경우에 효과적입니다 . 이 시나리오에서는 프록시 서버가 클라이언트 인증의 추가 보안이 있는 사이트 간의 보안 채널 역할을 수행합니다 (그림 14-4 참조).

그림 14-4 보안 클라이언트에서 프록시 연결 및 보안 프록시에서 컨텐트 서버 연결



각각의 구성을 설정하는 방법에 대한 자세한 내용은 " 역방향 프록시 설정 " (319 페이지) 을 참조하십시오 .

SSL 처럼 프록시도 클라이언트 인증을 사용할 수 있습니다 . 이를 위해 프록시에 대한 요청을 수행하는 컴퓨터가 신분 확인을 위한 인증서 (인증 양식) 를 제공해야 합니다 .

로드 밸런싱용 프록시

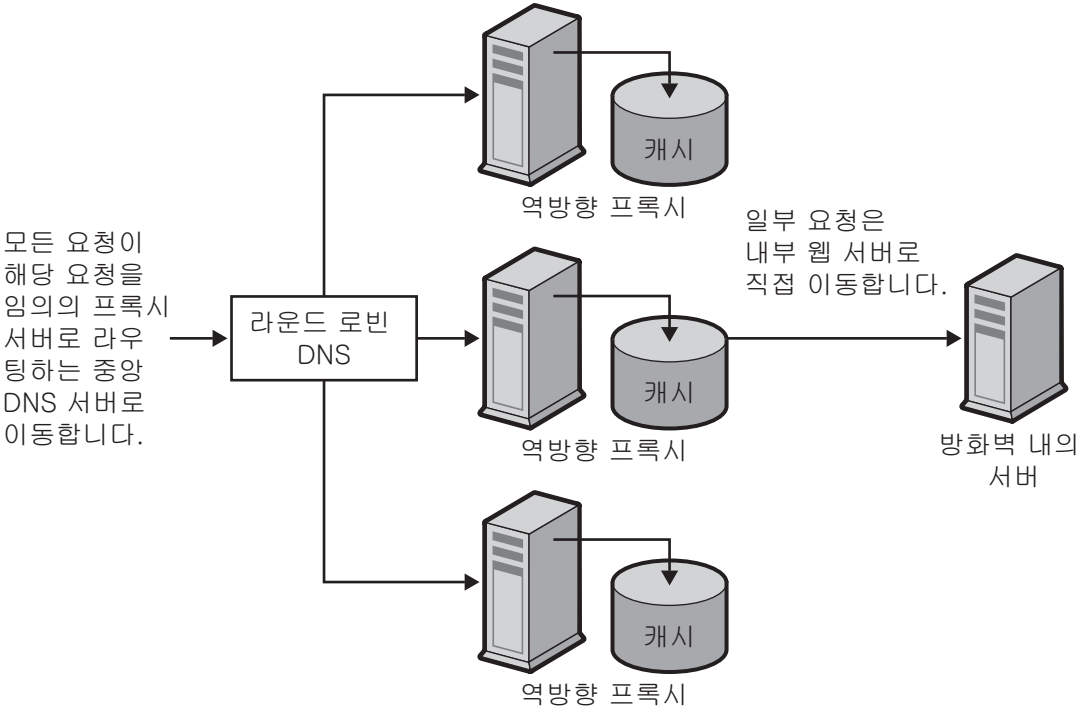
한 조직 내에서 여러 프록시 서버를 사용하여 웹 서버 간의 네트워크 로드를 밸런싱할 수 있습니다 . 이 모델을 사용하면 프록시 서버의 캐시 키능을 활용하여 로드 밸런싱용 서버 풀을 만들 수 있습니다 . 이 경우 프록시 서버는 방화벽의 어느 쪽에나 위치할 수 있습니다 . 날마다 많은 요청을 받는 웹 서버가 있는 경우 프록시 서버를 활용하여 웹 서버의 로드를 줄이고 보다 효율적인 네트워크 액세스가 가능합니다 .

프록시 서버는 클라이언트 요청과 실제 서버 간의 중개인 역할을 수행합니다 . 프록시 서버는 요청된 문서를 캐시합니다 . 프록시 서버가 여럿인 경우 DNS 는 IP 주소를 " 라운드 로빈 " 방식으로 임의 선택하여 요청을 라우팅할 수 있습니다 . 클라이언트는 매번 같은 URL 을 사용하지만 요청 라우팅은 매번 다른 프록시를 통해 수행됩니다 .

사용량이 많은 컨텐트 서버로의 요청을 여러 프록시를 사용하여 처리하면 서버가 더 높은 부하를 처리하고 단독 처리 시보다 더 효율적이라는 장점이 있습니다 . 프록시가 최초로 컨텐트 서버에서 문서를 검색하는 초기 시작 후 컨텐트 서버 요청 수는 급격히 줄어듭니다 .

CGI 요청과 비정기적인 새 요청만 컨텐트 서버로 계속 들어갑니다 . 나머지는 프록시에서 처리합니다 . 다음을 예로 들 수 있습니다 . 서버에 대한 요청 중 90% 가 CGI 요청이 아니며 (즉 , 캐시될 수 있으며) 컨텐트 서버가 매일 2 백만 건의 요청을 받는다고 가정합니다 . 이 상황에서 3 대의 역방향 프록시를 연결하고 각 프록시가 매일 2 백만 건의 요청을 처리하도록 하면 매일 6 백만 건의 요청을 처리할 수 있게 됩니다 . 컨텐트 서버에 도착하는 10% 의 요청을 합하면 매일 각 프록시에서 200,000 건 , 전체 합해서 600,000 건을 처리하게 되며 전에 비해 상당히 효율적입니다 . 요청 수는 2 백만에서 6 백만까지 증가할 수 있고 , 이에 따라 컨텐트 서버의 로드는 2 백만에서 600,000 만으로 격감할 수 있습니다 . 실제 결과는 상황에 따라 달라질 수 있습니다 .

그림 14-5 로드 밸런싱에 사용되는 프록시



역방향 프록시 설정

역방향 프록시를 설정하려면 정상 매핑과 역방향 매핑 등 , 두 가지 매핑이 필요합니다 .

- 정상 매핑은 요청을 컨텐트 서버로 재지정합니다 . 클라이언트가 프록시 서버에서 문서를 요청하면 프록시 서버는 정상 매핑을 사용하여 실제 문서 위치를 알립니다 .

주의 자동 구성 파일을 서비스하는 프록시에서는 역방향 프록시를 사용해서는 안 됩니다 . 프록시가 잘못된 결과를 반환할 수 있습니다 .

- 역방향 매핑은 컨텐트 서버로부터의 재지정을 위한 프록시 서버 트랩을 만듭니다 . 프록시는 재지정을 가로챈 후 재지정된 URL 을 변경하여 프록시 서버에 매핑합니다 . 예를 들어 클라이언트가 옮겨졌거나 찾을 수 없는 문서를 요청한 경우 컨텐트 서버는 요청한 URL 에서 문서를 찾을 수 없음을 설명하는 메시지를 클라이언트에게 반환합니다 . 컨텐트 서버는 옮겨진 파일을 찾을 수 있는 URL 을 나열하는 HTTP 헤더를 반환된 메시지에 추가합니다 . 내부 컨텐트 서버의 비밀을 보장하기 위해 프록시는 역방향 매핑을 사용하여 URL 을 재지정할 수 있습니다 .

웹 서버 `http://http.site.com/` 이 있고 이에 대한 역방향 프록시를 설정한다고 가정합니다 . 역방향 프록시를 `http://proxy.site.com/` 이라고 합니다 .

정상 매핑과 *역방향* 매핑을 다음과 같이 만듭니다 .

1. Server Manager 에 액세스하고 URLs 탭을 누릅니다 .
2. Create Mapping 링크를 누릅니다 . Create Mapping 페이지가 나타납니다 .
3. 나타난 페이지에서 한 매핑에 대한 정보를 입력합니다 . 예 :

   Regular mapping:

   Source prefix: `http://proxy.site.com`

   Source destination: `http://http.site.com/`
4. OK 를 누릅니다 . 페이지로 돌아가서 두 번째 매핑을 만듭니다 .

   Reverse mapping:

   Source prefix: `http://http.site.com/`

   Source destination: `http://proxy.site.com/`

5. 변경 후 OK 를 누릅니다 .

   OK 버튼을 누르면 프록시 서버가 하나 이상의 추가 매핑을 추가합니다 . 매핑을 보려면 View/Edit Mappings 링크를 누릅니다 . 추가 매핑의 형식은 다음과 같습니다 .

   from: /

   to: `http://http.site.com/`

   이러한 추가 자동 매핑은 보통 서버로 역방향 프록시에 연결하는 사용자에 대한 것입니다 . 첫 번째 매핑은 정상 프록시로 역방향 프록시에 연결하는 사용자에 대한 것입니다 . 설치에 따라 보통 두 번째 매핑만 필요하지만 둘 다 있다고 해서 프록시에서 문제가 발생하지는 않습니다 .

참고 웹 서버 별칭이 여럿이면 각 별칭마다 해당하는 정상 매핑이 있어야 합니다 . 웹 서버가 여러 DNS 별칭을 통해 자기 자신에 대한 재지정을 생성할 경우 각 별칭마다 해당하는 역방향 매핑이 있어야 합니다 .

프록시 서버는 자체에서 CGI 응용 프로그램을 실행하지 않으므로 CGI 응용 프로그램은 계속 원본 서버에서 실행됩니다. 그러나 CGI 스크립트에서 결과값이 캐시될 수 있음을 나타낼 경우 (Last-modified 나 Expires 헤더를 발행하여 0 이 아닌 TTL(Time-to-live) 포함), 프록시가 결과를 캐시합니다 .

주의 웹 서버용 컨텐츠를 제작할 때는 컨텐츠가 역방향 프록시에서 서비스되고 웹 서버의 파일에 대한 모든 링크가 상대 링크여야 함을 기억하십시오 . HTML 파일에 호스트 이름에 대한 참조가 있어서는 *안 됩니다* . 즉 모든 링크는 페이지에 대한 것이어야 합니다 .

`/abc/def`

다음과 같이 유효한 호스트 이름과 반대입니다 .

`http://http.site.com/abc/def`

보안 역방향 프록시 설정

보안 역방향 프록시를 설정하기 전에 디지털 인증서 , 인증 기관 (CA) 및 인증에 대해 이해해야 합니다 .

보안 역방향 프록시 설정은 비보안 역방향 프록시 설정과 거의 유사합니다 . 유일한 차이점은 암호화할 파일에 대한 프로토콜로 HTTPS 를 지정해야 한다는 점입니다 .

다음 지침은 선택한 구성 시나리오에 따라 보안 역방향 프록시를 설정하는 방법을 설명합니다 . 이 지침에서는 매핑 설정 방법을 설명하기 위해 `http.site.com` 이라는 웹 서버가 있으며 `proxy.site.com` 이라는 보안 역방향 프록시 서버를 설정한다고 가정합니다 . 다음 단계를 수행할 때는 지침에서 사용하는 예제 이름을 실제 웹 서버와 프록시 서버의 이름으로 변경하십시오 .

보안 클라이언트에서 프록시

1. Server Manager 에 액세스하고 URLs 탭을 누릅니다 .
2. Create Mapping 링크를 누릅니다 . Create Mapping 페이지가 나타납니다 .
3. 나타나는 페이지에서 다음과 같은 방법으로 정상 및 역방향 매핑을 설정합니다 .

   정상 매핑 :

   Source prefix: `https://proxy.mysite.com`

   Source destination: `http://http.mysite.com/`

   역방향 매핑 :

   Source prefix: `http://http.mysite.com/`

   Source destination: `https://proxy.mysite.com/`

4. 변경 사항을 저장 및 적용합니다 .

   만든 매핑을 보려면 View/Edit Mappings 링크를 누릅니다 .

참고 프록시 서버가 보안 모드에서 실행될 때만 이 구성이 적용됩니다 . 즉 암호화를 사용하도록 설정하고 명령줄에서 프록시를 재시작해야 합니다 . 명령줄에서 프록시를 재시작하려면 프록시 디렉토리로 이동한 후 **`./start`** 를 입력합니다 .

보안 프록시에서 컨텐트 서버

1. Server Manager 에 액세스하고 URLs 탭을 누릅니다 .
2. Create Mapping 링크를 누릅니다 . Create Mapping 페이지가 나타납니다 .

3. 나타나는 페이지에서 다음과 같은 방법으로 정상 및 역방향 매핑을 설정합니다 .

   정상 매핑 :

   Source prefix: `http://proxy.mysite.com`

   Source destination: `https://http.mysite.com/`

   역방향 매핑 :

   Source prefix: `https://http.mysite.com/`

   Source destination: `http://proxy.mysite.com/`

4. 변경 사항을 저장 및 적용합니다 . 만든 매핑을 보려면 View/Edit Mappings 링크를 누릅니다 .

참고 컨텐트 서버가 보안 모드에서 실행될 때만 이 구성이 적용됩니다 .

보안 클라이언트에서 프록시 , 보안 프록시에서 컨텐트 서버

1. Server Manager 에 액세스하고 URLs 탭을 누릅니다 .
2. Create Mapping 링크를 누릅니다 . Create Mapping 페이지가 나타납니다 .
3. 나타나는 페이지에서 다음과 같은 방법으로 정상 및 역방향 매핑을 설정합니다 .

   정상 매핑 :

   Source prefix: `https://proxy.mysite.com`

   Source destination: `https://http.mysite.com/`

   역방향 매핑 :

   Source prefix: `https://http.mysite.com/`

   Source destination: `https://proxy.mysite.com/`

4. 변경 사항을 저장 및 적용합니다 . 만든 매핑을 보려면 View/Edit Mappings 링크를 누릅니다 .

참고 프록시 서버와 컨텐트 서버가 보안 모드에서 실행될 때만 이 구성이 적용됩니다 . 즉 프록시에서 암호화를 사용하도록 설정하고 명령줄에서 프록시를 재시작해야 합니다 . 명령줄에서 프록시를 재시작하려면 프록시 디렉토리로 이동한 후 **`./restart`** 를 입력합니다 .

역방향 프록시에서의 가상 멀티호스팅

가상 멀티호스팅은 원본 서버 (또는 이 예제에서는 역방향 프록시) 가 각각의 해당 주소에 각기 다른 서버가 설치된 것처럼 여러 DNS 별칭에 응답하도록 하는 기능입니다 . 예를 들어 다음과 같은 DNS 호스트 이름이 있을 수 있습니다 .

- `www`
- `specs`
- `phones`

이 호스트 이름들을 같은 IP 주소 (역방향 프록시의 IP 주소) 에 매핑할 수 있습니다 . 그런 다음 어느 DNS 가 액세스에 사용되었는가에 따라 역방향 프록시의 동작을 다르게 할 수 있습니다 .

또한 가상 멀티호스팅을 사용하면 하나의 역방향 프록시에서 여러 개의 도메인을 호스팅할 수 있습니다 . 예 :

- `www.domain-1.com`
- `www.domain-2.com`
- `www.domain-3.com`

하나의 프록시 서버에서 모두 여러 도메인 및 여러 로컬 호스트 이름의 조합을 사용할 수 있습니다 .

- `www`
- `specs`
- `phones`
- `www.domain-1.com`
- `www.domain-2.com`
- `www.domain-3.com`

이 절에서는 다음 항목에 대해 설명합니다 .

- 가상 멀티호스팅의 기능적 세부 사항
- 가상 멀티호스팅에 대한 중요 참고 정보

가상 멀티호스팅의 기능적 세부 사항

가상 멀티호스팅 기능은 DNS 호스트와 DNS 이름 (또는 별칭) 을 지정한 다음 해당 호스트 이름으로 전송된 요청을 보내야 하는 대상 URL 접두사를 제공하여 작동합니다 . 예를 들어 두 가지 매핑이 있습니다 .

- `engr.domain.com -> http://int-engr.domain.com`
- `mktg.domain.com -> http://int-mktg.domain.com`

매핑은 루트 대 루트일 필요는 없으며 대상 URL 의 추가 URL 접두사를 지정할 수 있습니다 .

- `engr.domain.com -> http://internal.domain.com/engr`
- `mktg.domain.com -> http://internal.domain.com/mktg`

가상 도메인 매핑에서도 마찬가지입니다 . 예를 들어 다음을 사용할 수 있습니다 .

- `www.domain-1.com -> http://int-engr.domain.com`
- `www.domain-2.com -> http://int-mktg.domain.com`

시스템은 HTTP "Host:" 헤더를 찾고 , 헤더에 따라 일치하는 가상 멀티호스팅 매핑을 선택합니다 . 일치하는 멀티호스팅 매핑이 없으면 서버는 구성 파일의 순서에 따라 계속 다른 매핑을 찾거나 , 일치하는 매핑이 없으면 매핑을 수행하지 않습니다 . 일치가 없으면 프록시는 보통 "Proxy denies fulfilling the request(프록시가 요청 수행 거부)" 라는 응답을 반환합니다 .

가상 멀티호스팅을 구성하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 URLs 탭을 누릅니다 .
2. Configure Virtual Multihosting 링크를 누릅니다 . Configure Virtual Multihosting 페이지가 나타납니다 .
3. Source Hostname (alias) 필드에 이 매핑을 적용할 로컬 호스트 이름 또는 DNS 별칭을 지정합니다 .
4. Source Domain Name 필드에 이 매핑을 적용할 로컬 도메인 이름을 입력합니다 . 여러 DNS 도메인을 멀티호스팅하는 경우를 제외하면 보통 네트워크의 도메인 이름을 사용합니다 .
5. Destination URL Prefix 필드에 호스트 및 도메인 이름이 위에 지정한 것과 일치하는 경우에 요청을 보낼 대상 URL 접두사를 입력합니다 .
6. 템플릿을 사용할 경우 Use This Template 드롭다운 목록에서 템플릿 이름을 선택하거나 , 템플릿을 적용하지 않으려면 "NONE" 값을 그대로 둡니다 .
7. OK 를 누릅니다 .

8. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
9. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

   만들 가상 멀티호스팅 매핑 각각에 대해 위의 절차를 반복합니다 .

모든 가상 멀티호스팅 매핑은 Configure Virtual Multihosting 페이지의 하단에 표시됩니다 . 참고로 Source Hostname (alias) 및 Source Domain Name 필드는 프록시 포트 번호와 함께 하나의 정규식으로 병합됩니다 . 병합된 정규식은 Host 헤더와 비교하는 데 사용됩니다 .

예를 들어 호스트가 "www", 도메인이 "example.com", 포트 번호가 "8080" 이면 다음과 같은 정규식으로 표시됩니다 .

```
www(|.example.com)(|:8080)
```

이렇게 하면 사용자가 입력하거나 클라이언트가 전송할 수 있는 아래의 모든 조합과 일치를 보장할 수 있습니다 . 단 , 포트 번호가 80 이 아닌 경우에도 서버의 청취 포트가 명확하다면 일부 클라이언트 소프트웨어는 포트 번호를 전송하지 않을 수도 있습니다 .

- `www`
- `www:8080`
- `www.example.com`
- `www.example.com:8080`

가상 멀티호스팅에 대한 중요 참고 정보

역방향 프록시 매핑을 구성하기 전에 Client autoconfiguration 기능을 사용하지 않도록 설정해야 합니다 . 클라이언트 자동 구성 기능은 역방향이 아닌 전방향 프록시 작업을 위한 것이므로 이를 사용하지 않아야 문제를 예방할 수 있습니다 .

Virtual Multihosting 기능은 자동 역방향 매핑을 만듭니다 . 즉 Virtual Multihosting 페이지에서 입력한 매핑에 대해 역방향 매핑을 만들지 마십시오 .

가상 매핑은 `obj.conf` 에서 `virt-map` 함수로 지정됩니다 .

가상 매핑은 obj.conf 구성 파일에서 지정한 순서로 비교됩니다 . 가상 매핑에 앞서 정상 , 역방향 , 정규식 또는 클라이언트 자동 구성 매핑이 있으면 해당 매핑이 우선 적용됩니다 . 마찬가지로 가상 매핑에서 일치하는 내용이 없으면 `obj.conf` 에서 가상 매핑 부분 이후의 다음 매핑에 대한 변환을 계속합니다 .

프록시 서버의 포트 번호가 변경되면 Virtual Multihosting 매핑의 포트 번호가 틀리게 되므로 Virtual Multihosting 을 다시 만들어야 합니다 .

15 장

SOCKS 사용

이 장에서는 Sun Java System Web Proxy Server 에 포함된 SOCKS 서버의 구성 및 사용 방법에 대해 설명합니다 . Proxy Server 는 SOCKS 버전 4 및 5 를 지원합니다 .

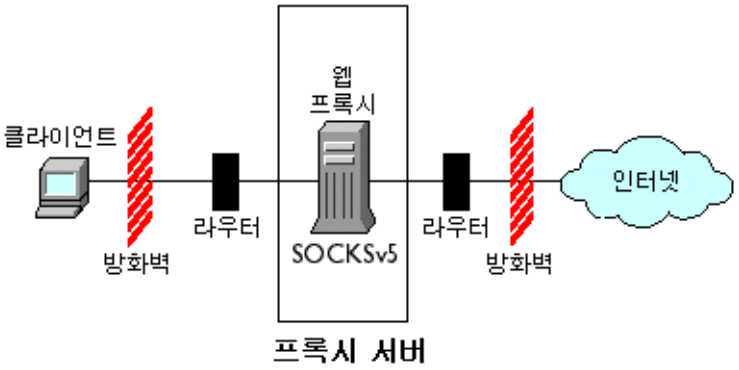
이 장은 다음 내용으로 구성되어 있습니다 .

SOCKS 정보

SOCKS 는 SOCKS 서버의 반대쪽에 있는 호스트의 연결 요청을 재지정하여 직접적인 IP 접근 없이 한쪽의 호스트가 다른 쪽의 호스트에 완전하게 액세스할 수 있도록 하는 네트워킹 프록시 프로토콜입니다 . SOCKS 는 권한 없는 사용자가 인터넷을 통해 내부 호스트에 액세스하는 것은 방지하면서도 SOCKS 서버 뒤의 호스트가 인터넷에 완전하게 액세스할 수 있도록 하는 네트워크 방화벽으로 널리 사용됩니다 .

SOCKS 서버는 지점 간 기반으로 방화벽을 통해 액세스를 제어하는 일반적인 방화벽 데몬입니다 . SOCKS 서버는 요청을 인증 및 승인하고 , 프록시 연결을 수립하고 , 데이터를 중계합니다 . SOCKS 서버는 응용 프로그램 수준이 아닌 네트워크 수준에서 작동하므로 요청 전송에 사용되는 프로토콜이나 방법에 대해 알지 못합니다 . SOCKS 서버가 프로토콜을 알지 못하므로 Proxy Server 가 지원하지 않는 Telnet 과 같은 프로토콜을 전달하는 데 사용될 수 있습니다 .

그림 15-1 네트워크에서의 SOCKS Server 위치



번들로 제공되는 SOCKS v5 Server 사용

Sun Java System Web Proxy Server 는 다른 SOCKS 데몬이 사용하는 표준 `socks5.conf` 파일 형식을 포함합니다 . Proxy Server 가 이 데몬을 사용하여 요청을 라우팅하거나 , Proxy Server 와는 별도로 실행하여 네트워크에 추가 기능을 제공할 수 있습니다 . Proxy Server 가 SOCKS 서버를 통해 요청을 라우팅하도록 구성하는 방법에 대한 자세한 내용은 " 라우팅 항목 구성 " (338 페이지) 을 참조하십시오 .

Proxy Server 에 포함된 SOCKS 데몬은 사용하지 않도록 기본 설정되어 있으며 Server Manager 의 SOCKS 탭이나 명령줄에서 사용하도록 설정할 수 있습니다 . 자세한 내용은 "SOCKS v5 서버 시작 및 중지 " (330 페이지) 를 참조하십시오 .

참고 Proxy Server 4 에서 SOCKS 데몬 이름은 `ns-sockd` 에서 `sockd` 로 변경됩니다 .

다음은 Proxy Server 에 포함된 SOCKS 서버를 사용하기 위해 수행해야 하는 높은 수준의 절차입니다 .

1. SOCKS 서버를 구성합니다 ("SOCKS v5 Server 구성 " (331 페이지) 참조).
2. SOCKS 서버가 여러 인터페이스가 있는 컴퓨터에서 실행될 경우 SOCKS 라우팅 항목을 만듭니다 (" 라우팅 항목 구성 " (338 페이지) 참조).
3. 인증 항목을 만듭니다 ("SOCKS v5 인증 항목 구성 " (332 페이지) 참조).
4. 연결 항목을 만듭니다 ("SOCKS v5 연결 항목 구성 " (335 페이지) 참조).
5. SOCKS 서버를 사용하도록 설정합니다 ("SOCKS v5 서버 시작 및 중지 " (330 페이지) 참조).

socks5.conf 정보

Sun Java System Web Proxy Server 는 `socks5.conf` 파일을 사용하여 SOCKS 서버 및 서비스에 대한 액세스를 제어합니다 . 각 행은 행과 일치하는 요청을 수신했을 때 Proxy Server 가 수행할 작업을 정의합니다 . Server Manager 에서의 선택은 `socks5.conf` 에 저장됩니다 . 파일을 직접 편집할 수도 있습니다 . `socks5.conf` 파일은 다음과 같이 설치 디렉토리 (*server_root*) 에 있습니다 .

server_root`/proxy-`*serverid*`/config` 디렉토리

이 절에서는 `socks5.conf` 에 관한 일반적인 정보를 제공합니다 . 이 파일과 , 이 파일의 지시문 및 구문에 대한 자세한 내용은 Proxy Server Configuration File Reference 를 참조하십시오 .

인증

SOCKS 데몬을 해당 서비스를 사용하는 데 인증을 요청하도록 구성할 수 있습니다 . 인증은 연결하는 클라이언트의 호스트 이름 및 포트에 따라 수행됩니다 . 사용자 이름과 비밀 번호를 요청하도록 선택한 경우 `socks5.conf` 파일이 참조하는 사용자 이름 및 비밀 번호 파일에 대해 정보를 인증합니다 . 입력한 사용자 이름 및 비밀 번호가 비밀 번호 파일의 목록과 일치하지 않으면 액세스가 거부됩니다 . 비밀 번호 파일의 사용자 이름과 비밀 번호 형식은 *사용자 이름 비밀 번호*이며 , 여기서 사용자 이름과 비밀 번호는 공백으로 구분합니다 . 사용자를 금지할 수도 있습니다 . 사용자 이름과 비밀 번호 인증을 요구하려면 `SOCKS5_PWDFILE` 지시문을 `socks5.conf` 에 추가해야 합니다 . 지시문 및 구문에 대한 자세한 내용은 Proxy Server Configuration File Reference 의 `socks5.conf` 부분을 참조하십시오 .

파일뿐 아니라 구성된 LDAP 서버에 대해서도 사용자 이름 및 비밀 번호 인증을 수행할 수 있습니다 .

액세스 제어

액세스 제어는 socks5.conf 파일의 정렬된 행 세트를 사용하여 수행합니다 . 각 행에는 리소스에 대한 액세스를 허용 또는 차단하는 지시문 한 줄을 포함합니다 . 지시문은 구성 파일에 나타나는 순서로 처리됩니다 . 허용 지시문과 일치하지 않는 요청은 액세스가 거부됩니다 .

로깅

SOCKS 데몬은 오류와 액세스 메시지를 모두 SOCKS 로그 파일에 기록합니다 . 로그 파일 위치와 로깅 유형은 socks5.conf 에서 지정할 수 있습니다 .

또한 SOCKS 데몬은 데몬에 대한 통계를 제공하는 시간별 통계 항목을 생성합니다 .

조정

작업자 수와 SOCKS 서버가 사용하는 허용 스레드 수를 결정하는 데 socks5.conf 파일을 사용할 수 있습니다 . 작업자 수는 SOCKS 서버의 성능에 영향을 미칩니다 .

작업자 및 허용 스레드 수와 이들이 성능에 미치는 영향에 대한 자세한 내용은 "SOCKS v5 Server 구성 " (331 페이지) 을 참조하십시오 .

SOCKS v5 서버 시작 및 중지

Server Manager 나 명령줄에서 SOCKS 서버를 시작 및 중지할 수 있습니다 .

Server Manager 에서 SOCKS 서버를 시작 및 중지하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Start/Stop SOCKS Server 링크를 누릅니다 .
3. SOCKS 서버를 시작하거나 중지합니다 .

명령줄에서 SOCKS 서버를 시작 및 중지하려면 다음과 같이 합니다 .

server_root/proxy-*serverid* 디렉토리의 스크립트를 실행합니다 . 여기서 *server_root* 는 설치 루트입니다 .

- start-sockd 는 SOCKS 데몬을 시작합니다 .
- stop-sockd 는 SOCKS 데몬을 중지합니다 .
- restart-sockd 는 SOCKS 데몬을 재시작합니다 .

SOCKS v5 Server 구성

SOCKS 서버를 구성하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Configure SOCKS v5 링크를 누릅니다 .
3. SOCKS Port 필드에 SOCKS 서버가 청취할 포트 번호를 입력합니다 (기본 포트 : 1080).
4. 사용할 SOCKS 옵션을 선택합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .
   - **Disable Reverse DNS Lookup.** SOCKS 서버의 역방향 DNS 조회를 사용하지 않습니다 . 역방향 DNS 는 IP 주소를 호스트 이름으로 변환합니다 . 역방향 DNS 조회를 사용하지 않으면 네트워크 리소스를 보호할 수 있습니다 . 이 옵션은 사용하지 않도록 기본 설정되어 있습니다 . 즉 Disable Reverse DNS Lookup 확인란이 기본 선택되어 있습니다 . 역방향 DNS 조회를 사용하지 않으며 호스트 이름과 URL 이 요청된 경우 , 서버는 호스트 이름을 IP 주소와 매핑하지 않습니다 . 역방향 DNS 조회를 사용할 경우 , 서버는 매핑을 수행하고 SOCKS 로그 파일에 항목을 추가하여 DNS 변환을 나열합니다 .
   - **Use Client-specific Bind Port.** 클라이언트가 BIND 요청에서 포트를 지정할 수 있도록 합니다 . 이 옵션을 사용하지 않으면 SOCKS 는 클라이언트가 요청한 포트를 무시하고 임의 포트를 할당합니다 . 사용하지 않도록 기본 설정되어 있습니다 .
   - **Allow Wildcard As Bind IP Address.** 클라이언트가 BIND 요청에서 모든 IP 주소가 연결할 수 있음을 나타내는 모두 0 으로 된 IP 주소 (0.0.0.0) 를 지정할 수 있도록 합니다 . 이 기능을 사용하지 않으면 클라이언트는 해당 바인드 포트로 연결할 IP 주소를 지정해야 하며 , SOCKS 서버는 0.0.0.0 으로의 바인드 요청을 거부합니다 . 사용하지 않도록 기본 설정되어 있습니다 .
   - **Quench Updates.** 매 시간마다 작성하는 자동 통계 파일을 사용하지 않습니다 . 사용하지 않을 경우 요청 시마다 작성을 수행합니다 (" 로깅 " (330 페이지) 참조).

참고 Quench Updates 요소는 사용자 인터페이스에 표시되기는 하지만 이번 Proxy Server 4 릴리즈에서는 구현되지 않았습니다 .

5. Log File 필드에 SOCKS 로그 파일의 전체 경로 이름을 입력합니다 . 기본값은 *server_root*/proxy-*serverid*/logs/socks5.log 입니다 .

6. Log Level 드롭다운 목록에서 로그 파일이 경고 및 오류만 포함할지 , 모든 요청을 포함할지 또는 디버깅 메시지를 포함할지 지정합니다 .
7. RFC 1413 ident 응답을 선택합니다 . Ident 는 SOCKS 서버에서 클라이언트의 사용자 이름을 결정할 수 있도록 합니다 . 일반적으로 이 기능은 클라이언트가 UNIX 종류를 실행하는 경우에만 작동합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .
   - **Don't Ask.** 클라이언트의 사용자 이름을 결정하는 데 Ident 를 사용하지 않습니다 . 권장 및 기본 설정입니다 .
   - **Ask But Don't Require.** 모든 클라이언트의 사용자 이름을 요청하지만 응답의 유효성 여부는 관계 없습니다 . 이 옵션을 선택하면 Ident 를 로깅 용도로만 사용합니다 .
   - **Require.** 모든 클라이언트의 사용자 이름을 요청하며 , 유효한 응답을 한 경우에만 액세스를 허용합니다 .
8. SOCKS Tuning 부분에서 SOCKS 서버가 사용할 작업자 수 및 승인 스레드 수를 지정하고 (이 숫자는 SOCKS 서버 성능에 영향을 미침) OK 를 누릅니다 .
   - **Number of Worker Threads.** 기본값은 40 입니다 . SOCKS 서버가 너무 느리면 작업자 스레드의 수를 늘리십시오 . SOCKS 서버가 불안정하면 이 수를 줄이십시오 . 이 수를 변경하는 경우 기본값에서 시작하여 필요에 따라 늘리거나 줄입니다 . 보통 작업자 스레드 수는 10 - 150 입니다 . 최대 512 까지 설정할 수 있지만 150 을 초과하면 효율이 떨어지고 불안정해지기 쉽습니다 .
   - **Number of Posted Accepts.** 기본값은 1 입니다 . SOCKS 서버에서 연결이 자꾸 끊어지면 허용 스레드의 수를 늘리십시오 . SOCKS 서버가 불안정하면 이 수를 줄이십시오 . 이 수를 변경하는 경우 기본값에서 시작하여 필요에 따라 늘리거나 줄입니다 . 보통 허용 스레드 수는 1 - 10 입니다 . 최대 512 까지 설정할 수 있지만 60 을 초과하면 효율이 떨어지고 불안정해지기 쉽습니다 . 매우 중요한 설정입니다 . SOCKS 서버에 부하가 걸리고 연결이 끊어져 요청이 실패하는 경우에만 이 설정을 조정하십시오 .

SOCKS v5 인증 항목 구성

SOCKS 인증 항목은 SOCKS 데몬이 연결을 승인해야 하는 호스트와 SOCKS 데몬이 해당 호스트를 인증하는 데 사용할 인증 유형을 표시합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 인증 항목 삭제
- 인증 항목 이동

인증 항목 만들기

SOCKS 인증 항목을 만들려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Authentication 링크를 누릅니다 .
3. Add 버튼을 누릅니다 .
4. Host Mask 필드에 SOCKS 서버가 인증할 호스트의 IP 주소 또는 호스트 이름을 입력합니다 . IP 주소를 입력할 때는 주소 뒤에 슬래시 (/) 와 해당 수신 IP 주소에 적용할 마스크를 함께 입력합니다 . SOCKS 서버는 IP 주소에 이 마스크를 적용하여 호스트의 유효성 여부를 결정합니다 . Host mask 항목에 공백을 입력해서는 안 됩니다 . 호스트 마스크를 입력하지 않으면 모든 호스트에 해당 인증 항목이 적용됩니다 .

   예를 들어 호스트 마스크 필드에 155.25.0.0/255.255.0.0 을 입력했다면 , 호스트의 IP 주소가 155.25.3.5 인 경우 , SOCKS 서버는 이 IP 주소에 마스크를 적용하고 인증 레코드를 적용할 IP 주소 (155.25.0.0) 에 일치하는 것으로 결정합니다 .
5. Port Range 필드에 SOCKS 서버가 인증할 호스트 컴퓨터의 포트를 입력합니다 . 포트 범위 입력에 공백이 있으면 안 됩니다 . 포트 범위를 입력하지 않으면 모든 포트에 해당 인증 항목이 적용됩니다 .

   대괄호 ([]) 를 이용하면 범위의 시작과 끝 지점의 포트를 포함하고 , 괄호를 이용하면 시작과 끝 지점 포트를 제외합니다 . 예를 들어 [1000-1010] 으로 입력하면 이 범위에 1000 과 1010 도 포함되며 , (1000-1010) 으로 입력하면 범위에서 1000 과 1010 을 제외합니다 . 대괄호와 괄호를 혼용할 수 있습니다 . 예를 들어 (1000-1010] 으로 입력하면 범위에서 1000 은 제외되지만 1010 은 포함됩니다 .
6. Authentication Type 드롭다운 목록에서 인증 유형을 선택합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .
   - **Require user-password.** SOCKS 서버에 액세스하려면 사용자 이름과 암호가 필요합니다 .
   - **user-password, if available.** 사용자 이름과 암호가 있는 경우에는 사용해야 하지만 없어도 SOCK 서버에 액세스할 수 있습니다 .
   - **Ban.** SOCK 서버 액세스를 금지합니다 .
   - **None.** 아무런 인증 없이 SOCKS 서버에 액세스할 수 있습니다 .

7. Insert 드롭다운 목록에서 socks5.conf 파일에서의 이 항목 위치를 선택하고 OK 를 누릅니다 . 여러 가지의 인증 메소드가 있기 때문에 평가할 순서를 지정해야 합니다 . 따라서 클라이언트가 첫 번째 인증 메소드를 지원하지 않으면 두 번째 메소드가 대신 사용됩니다 . 클라이언트가 목록에 있는 인증 메소드를 모두 지원하지 않으면 SOCKS 서버는 요청을 수락하지 않고 연결을 끊게 됩니다 .

인증 항목 편집

인증 항목을 편집하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Authentication 링크를 누릅니다 .
3. 편집할 인증 항목을 선택하고 Edit 버튼을 누릅니다 .
4. 원하는 사항을 변경한 다음 OK 를 누릅니다 .

인증 항목 삭제

인증 항목을 삭제하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Authentication 링크를 누릅니다 .
3. 삭제할 인증 항목을 선택하고 Delete 버튼을 누릅니다 .

인증 항목 이동

항목은 socks5.conf 파일에 나타나는 순서대로 평가됩니다 . 이동을 통해 순서를 변경할 수 있습니다 .

인증 항목을 이동하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Authentication 링크를 누릅니다 .
3. 이동할 인증 항목을 선택하고 Move 버튼을 누릅니다 .
4. Move 드롭다운 목록에서 socks5.conf 파일에서의 이 항목 위치를 선택하고 OK 를 누릅니다 .

SOCKS v5 연결 항목 구성

SOCKS 연결 항목은 SOCKS 데몬이 요청을 허용할 것인지 , 또는 거부할 것인지 지정합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

연결 항목 만들기

연결 항목을 만들려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Connections 링크를 누릅니다 .
3. Add 버튼을 누릅니다 .
4. Authentication Type 드롭다운 목록에서 이 액세스 제어 줄에 적용할 인증 메소드를 선택합니다 .
5. Connection Type 드롭다운 목록에서 해당 줄과 일치하는 명령 유형을 지정합니다 . 다음과 같은 명령 유형을 사용할 수 있습니다 .
   - **Connect**
   - **Bind**
   - **UDP**
   - **All**

6. Source Host Mask 필드에 연결 제어 항목에 적용할 호스트의 IP 주소 또는 호스트 이름을 입력합니다 . IP 주소를 입력할 때 주소 뒤에 슬래시 (/) 와 해당 소스 IP 주소에 적용할 마스크를 함께 입력합니다 . SOCKS 서버는 소스 IP 주소에 이 마스크를 적용하여 호스트의 유효성 여부를 결정합니다 . Host mask 항목에 공백을 입력해서는 안 됩니다 . 대상 호스트 마스크를 입력하지 않으면 모든 호스트에 해당 연결 항목이 적용됩니다 .

   예를 들어 호스트 마스크 필드에 155.25.0.0/255.255.0.0 을 입력했다면 , 호스트의 IP 주소가 155.25.3.5 인 경우 , SOCKS 서버는 이 IP 주소에 마스크를 적용하고 연결 제어 항목을 적용할 IP 주소 (155.25.0.0) 에 일치하는 것으로 결정합니다 .

7. Port Range 필드에 연결 제어 항목을 적용할 소스 컴퓨터의 포트를 입력합니다 . 포트 범위에 공백이 있으면 안 됩니다 . 포트 범위를 지정하지 않으면 모든 포트에 해당 연결 항목이 적용됩니다 .

   대괄호 ([]) 를 이용하면 범위의 시작과 끝 지점의 포트를 포함하고 , 괄호를 이용하면 시작과 끝 지점 포트를 제외합니다 . 예를 들어 [1000-1010] 으로 입력하면 이 범위에 1000 과 1010 도 포함되며 , (1000-1010) 으로 입력하면 범위에서 1000 과 1010 을 제외합니다 . 대괄호와 괄호를 혼용할 수 있습니다 . 예를 들어 (1000-1010] 으로 입력하면 범위에서 1000 은 제외되지만 1010 은 포함됩니다 .

8. Destination Host Mask 필드에 연결 항목을 적용할 호스트의 IP 주소 또는 호스트 이름을 입력합니다 . IP 주소를 입력할 때 주소 뒤에 슬래시 (/) 와 해당 수신 IP 주소에 적용할 마스크를 함께 입력합니다 . SOCKS 서버는 대상 컴퓨터의 IP 주소에 이 마스크를 적용하여 대상 호스트의 유효성 여부를 결정합니다 . Host mask 항목에 공백을 입력해서는 안 됩니다 . 대상 호스트 마스크를 입력하지 않으면 모든 호스트에 해당 연결 항목이 적용됩니다 .

   예를 들어 대상 호스트 마스크 필드에 155.25.0.0/255.255.0.0 을 입력했다면 , 대상 호스트의 IP 주소가 155.25.3.5 인 경우 , SOCKS 서버는 이 대상 IP 주소에 마스크를 적용하고 , 대상 호스트의 IP 주소가 프록시 항목을 적용할 IP 주소 (155.25.0.0) 에 일치하는 것으로 결정합니다 .

9. Port Range 필드에 연결 제어 항목을 적용할 대상 호스트 컴퓨터의 포트를 입력합니다 . 포트 범위에 공백이 있으면 안 됩니다 . 포트 범위를 입력하지 않으면 모든 포트에 해당 연결 항목이 적용됩니다 .

참고 대부분의 SOCKS 응용 프로그램은 바인드 요청에 대해 포트 0 을 요청합니다 . 즉 포트 기본 설정이 없습니다 . 따라서 바인드에 대한 대상 포트 범위는 항상 포트 0 을 포함해야 합니다 .

대괄호 ([]) 를 이용하면 범위의 시작과 끝 지점의 포트를 포함하고 , 괄호를 이용하면 시작과 끝 지점 포트를 제외합니다 . 예를 들어 [1000-1010] 으로 입력하면 이 범위에 1000 과 1010 도 포함되며 , (1000-1010) 으로 입력하면 범위에서 1000 과 1010 을 제외합니다 . 대괄호와 괄호를 혼용할 수 있습니다 . 예를 들어 (1000-1010] 으로 입력하면 범위에서 1000 은 제외되지만 1010 은 포함됩니다 .

10. User Group 필드에 액세스를 허용 또는 거부할 그룹을 입력합니다 . 그룹을 지정하지 않으면 연결 항목이 모든 사용자에게 적용됩니다 .
11. Action 드롭다운 목록에서 생성 중인 연결에 대한 액세스 허용 , 또는 거부를 선택합니다 .
12. Insert 드롭다운 목록에서 `socks5.conf` 파일에서의 이 항목 위치를 선택하고 OK 를 누릅니다 . 여러 가지의 연결 지시문이 있기 때문에 평가할 순서를 지정해야 합니다 .

연결 항목 편집

연결 항목을 편집하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Connections 링크를 누릅니다 .
3. 편집할 연결 항목을 선택하고 Edit 버튼을 누릅니다 .
4. 원하는 사항을 변경한 다음 OK 를 누릅니다 .

연결 항목 삭제

연결 항목을 삭제하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Connections 링크를 누릅니다 .
3. 삭제할 연결 항목을 선택하고 Delete 버튼을 누릅니다 .

연결 항목 이동

항목은 socks5.conf 파일에 나타나는 순서대로 평가됩니다 . 이동을 통해 순서를 변경할 수 있습니다 .

연결 항목을 이동하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Connections 링크를 누릅니다 .
3. 이동할 연결 항목을 선택하고 Move 버튼을 누릅니다 .
4. Move 드롭다운 목록에서 , socks5.conf 파일에서의 이 항목 위치를 선택하고 OK 를 누릅니다 .

SOCKS v5 Server 체인 구성

Proxy Server 에서와 같은 방법을 이용하여 SOCKS 서버 여러 대로 체인을 구성할 수 있습니다 . 이렇게 하면 SOCKS 서버가 다른 SOCKS 서버를 통하여 라우팅할 수 있습니다 .

SOCKS 서버 체인을 구성하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Routing 링크를 누릅니다 .
3. Server Chaining 부분에 체인으로 연결된 Proxy Server 를 인증할 사용자 이름과 비밀 번호를 입력하고 (프록시 체인의 하향 프록시가 요청을 서비스하기 위해 인증을 요구할 경우) OK 를 누릅니다 .

라우팅 항목 구성

라우팅 항목을 사용하여 Proxy Server 가 SOCKS 서버를 통해 요청을 라우팅하도록 구성할 수 있습니다 . 라우팅 항목에는 SOCKS v5 라우팅과 SOCKS v5 프록시 라우팅 등 , 두 가지 유형이 있습니다 .

- SOCKS v5 라우팅은 SOCKS 데몬이 특정 IP 주소에 대해 사용할 인터페이스를 확인합니다 .
- SOCKS v5 프록시 라우팅은 다른 SOCKS 서버를 통해 액세스할 수 있는 IP 주소와 SOCKS 서버가 호스트에 직접 연결하는지 여부를 확인합니다 . SOCKS 서버를 통한 라우팅에서는 프록시 라우팅이 중요합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- SOCKS v5 라우팅 항목 만들기
- SOCKS v5 프록시 라우팅 항목 만들기
- 라우팅 항목 편집
- 라우팅 항목 삭제
- 라우팅 항목 이동

SOCKS v5 라우팅 항목 만들기

라우팅 항목을 만들려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Routing 링크를 누릅니다 .
3. Routing 부분에서 Add 버튼을 누릅니다 .
4. Host Mask 필드에 지정된 인터페이스를 통과하도록 할 연결 대상의 IP 주소 또는 호스트 이름을 입력합니다 . 외부에서 들어오는 연결과 내부에서 나가는 연결 모두에 적용됩니다 . IP 주소를 입력할 때 주소 뒤에 슬래시 (/) 와 해당 수신 IP 주소에 적용할 마스크를 함께 입력합니다 . SOCKS 서버는 IP 주소에 이 마스크를 적용하여 호스트의 유효성 여부를 결정합니다 . Host mask 항목에 공백을 입력해서는 안 됩니다 . 호스트 마스크를 입력하지 않으면 모든 호스트에 해당 SOCKS v5 가 적용됩니다 .

   예를 들어 호스트 마스크 필드에 155.25.0.0/255.255.0.0 을 입력했다면 , 호스트의 IP 주소가 155.25.3.5 인 경우 , SOCKS 서버는 이 IP 주소에 마스크를 적용하고 라우팅 항목을 적용할 IP 주소 (155.25.0.0) 에 일치하는 것으로 결정합니다 .
5. Port Range 필드에 지정된 인터페이스를 통과하도록 할 연결의 포트 범위를 입력합니다 . 외부에서 들어오는 연결과 내부에서 나가는 연결 모두에 적용됩니다 . 포트 범위에 공백을 입력해서는 안 됩니다. 포트 범위를 지정하지 않으면 모든 포트에 해당 SOCKS v5 항목이 적용됩니다 .

   대괄호 ([]) 를 이용하면 범위의 시작과 끝 지점의 포트를 포함하고 , 괄호를 이용하면 시작과 끝 지점 포트를 제외합니다 . 예를 들어 [1000-1010] 으로 입력하면 이 범위에 1000 과 1010 도 포함되며 , (1000-1010) 으로 입력하면 범위에서 1000 과 1010 을 제외합니다 . 대괄호와 괄호를 혼용할 수 있습니다 . 예를 들어 (1000-1010] 으로 입력하면 범위에서 1000 은 제외되지만 1010 은 포함됩니다 .
6. Interface/Address 필드에 외부에서 들어오는 연결과 내부에서 나가는 연결이 통과해야 하는 인터페이스의 IP 주소나 이름을 입력합니다 .

7. Insert 드롭다운 목록에서 `socks5.conf` 파일에서의 이 항목 위치를 선택하고 OK 를 누릅니다. 여러 가지의 라우팅 메소드가 있기 때문에 평가할 순서를 지정해야 합니다 .

참고 지정한 인터페이스는 들어오는 연결과 나가는 연결 모두에 사용해야 합니다 . 그렇지 않으면 들어오는 라우팅이 구성된 인터페이스와 달라 오류 메시지가 수신됩니다 .

SOCKS v5 프록시 라우팅 항목 만들기

프록시 라우팅 항목을 만들려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Routing 링크를 누릅니다 .
3. Proxy Routing 부분에서 Add 버튼을 누릅니다 .
4. Proxy Type 드롭다운 목록에서 라우팅에 사용할 Proxy Server 유형을 선택합니다 . 선택할 수 있는 옵션은 다음과 같습니다 .
   - **SOCKS v5**
   - **SOCKS v4**
   - **Direct connection**
5. Destination Host Mask 필드에 연결 항목을 적용할 호스트의 IP 주소 또는 호스트 이름을 입력합니다 . IP 주소를 입력할 때 주소 뒤에 슬래시 (/) 와 해당 수신 IP 주소에 적용할 마스크를 함께 입력합니다 . SOCKS 서버는 대상 컴퓨터의 IP 주소에 이 마스크를 적용하여 대상 호스트의 유효성 여부를 결정합니다 . Host mask 항목에 공백을 입력해서는 안 됩니다 . 대상 호스트 마스크를 입력하지 않으면 모든 호스트에 해당 연결 항목이 적용됩니다 .

   예를 들어 대상 호스트 마스크 필드에 155.25.0.0/255.255.0.0 을 입력했다면 , 대상 호스트의 IP 주소가 155.25.3.5 인 경우 , SOCKS 서버는 이 대상 IP 주소에 마스크를 적용하고 프록시 항목을 적용할 IP 주소 (155.25.0.0) 에 일치하는 것으로 결정합니다 .

6. Destination Port Range 필드에 연결 제어 항목을 적용할 대상 호스트의 포트를 입력합니다 . 포트 범위에 공백이 있으면 안 됩니다 . 포트 범위를 지정하지 않으면 모든 포트에 해당 프록시 항목이 적용됩니다 .

   대괄호 ([]) 를 이용하면 범위의 시작과 끝 지점의 포트를 포함하고 , 괄호를 이용하면 시작과 끝 지점 포트를 제외합니다 . 예를 들어 [1000-1010] 으로 입력하면 이 범위에 1000 과 1010 도 포함되며 , (1000-1010) 으로 입력하면 범위에서 1000 과 1010 을 제외합니다 . 대괄호와 괄호를 혼용할 수 있습니다 . 예를 들어 (1000-1010] 으로 입력하면 범위에서 1000 은 제외되지만 1010 은 포함됩니다 .

7. Destination Proxy Address 필드에 사용할 Proxy Server 의 호스트 이름이나 IP 주소를 입력합니다 .

8. Destination Proxy Port 필드에 Proxy Server 가 SOCKS 요청을 청취할 포트 번호를 입력합니다 .

9. Insert 드롭다운 목록에서 `socks5.conf` 파일에서의 이 항목 위치를 선택하고 OK 를 누릅니다 . 여러 가지의 라우팅 메소드가 있기 때문에 평가할 순서를 지정해야 합니다 .

라우팅 항목 편집

라우팅 항목을 편집하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Routing 링크를 누릅니다 .
3. 편집할 항목을 선택하고 Edit 버튼을 누릅니다 .
4. 원하는 사항을 변경한 다음 OK 를 누릅니다 .

라우팅 항목 삭제

라우팅 항목을 삭제하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Routing 링크를 누릅니다 .
3. 삭제할 항목을 선택하고 Delete 버튼을 누릅니다 .

라우팅 항목 이동

항목은 socks5.conf 파일에 나타나는 순서대로 평가됩니다 . 이동을 통해 순서를 변경할 수 있습니다 .

라우팅 항목을 이동하려면 다음과 같이 합니다 .

1. 서버 인스턴스에 대한 Server Manager 에 액세스하고 SOCKS 탭을 누릅니다 .
2. Set SOCKS v5 Routing 링크를 누릅니다 .
3. 이동할 항목을 선택하고 Move 버튼을 누릅니다 .
4. Move 드롭다운 목록에서 socks5.conf 파일에서의 이 항목 위치를 선택하고 OK 를 누릅니다 .

16 장

템플릿 및 리소스 관리

템플릿을 사용하면 URL 을 그룹화하여 프록시가 해당 URL 을 처리하는 방법을 구성할 수 있습니다 . 클라이언트가 검색을 시도하는 URL 에 따라 프록시가 각기 다르게 작동하도록 할 수 있습니다 . 예를 들어 클라이언트가 특정 도메인의 URL 에 액세스할 때 인증 (사용자 이름 및 비밀 번호 입력) 을 요구할 수 있습니다 . 또는 이미지 파일을 지시하는 URL 에 대한 액세스를 거부할 수도 있습니다 . 파일 유형에 따라 다른 캐시 새로 고침 설정을 구성할 수 있습니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

템플릿 정보

템플릿은 리소스라고 하는 URL 의 모음입니다 . 단일 URL, 공통점이 있는 URL 의 그룹 또는 전체 프로토콜이 리소스가 될 수 있습니다 . 템플릿의 이름을 지정하고 만든 후에는 정규식을 사용하여 URL 을 해당 템플릿에 할당할 수 있습니다 . 즉 프록시 서버가 다양한 URL 요청을 서로 다르게 처리하도록 구성할 수 있습니다 . 정규식으로 만들 수 있는 모든 URL 패턴은 템플릿에 포함할 수 있습니다 . 표 16-1 은 기본 리소스 목록과 , 다른 템플릿에 대한 정보를 제공합니다 .

표 16-1 리소스 정규식 와일드카드 패턴

정규식 패턴	구성 대상
`ftp://.*`	모든 FTP 요청
`http://.*`	모든 HTTP 요청
`https://.*`	모든 보안 HTTP 요청
`gopher://.*`	모든 Gopher 요청
`connect://.*:443`	HTTPS 포트에 대한 모든 SSL(보안) 트랜잭션
`http://home\.example\.com.*`	`home.example.com` 웹 사이트의 모든 문서
`.*\.gif.*`	`.gif` 문자열을 포함하는 모든 URL
`.*\.edu.*`	`.edu` 문자열을 포함하는 모든 URL
`http://.*\.edu.*`	`.edu` 도메인에 있는 컴퓨터로 이동하는 모든 URL

정규식에 대한 이해

Proxy Server 에서는 정규식을 사용하여 리소스를 식별할 수 있습니다 . 정규식은 문자열 패턴을 지정합니다 . 프록시 서버에서는 정규식을 사용하여 URL 에서 일치하는 패턴을 찾습니다 .

정규식 파일의 예는 다음과 같습니다 .

```
[a-z]*://[^:/]*\.abc\.com.*>
```

이 정규식은 `.abc.com` 도메인의 모든 문서와 일치합니다 . 이 문서의 프로토콜과 파일 확장자는 어느 것이든 가능합니다 .

표 16-2 는 정규식과 이에 해당하는 의미를 설명합니다 .

표 16-2 정규식과 의미

식	의미
.	새 행을 제외한 모든 단일 문자와 일치합니다 .
x?	정규식 *x* 가 *0-1* 회 *일치합니다* .
*x**	정규식 *x* 가 0 회 이상 일치합니다 .
x+	정규식 *x* 가 1 회 이상 일치합니다 .
x{n,m}	문자 *x* 가 일치합니다 . 여기서 *x* 는 *n* 회 이상 *m* 회 미만 발생합니다 .
x{n,}	문자 *x* 가 일치합니다 . 여기서 *x* 는 최소 *n* 회 발생합니다 .
x{n}	문자 *x* 가 일치합니다 . 여기서 *x* 는 정확히 *n* 회 발생합니다 .
[abc]	대괄호 안의 모든 문자가 일치합니다 .
[^abc]	대괄호 밖의 모든 문자가 일치합니다 .
[a-z]	대괄호 안의 범위에 해당하는 모든 문자가 일치합니다 .
x	문자 *x* 가 일치합니다 . 여기서 *x* 는 특수 문자가 아닙니다 .
\x	특수 문자 *x* 의 의미를 제거합니다 .
"x"	특수 문자 *x* 의 의미를 제거합니다 .
xy	이후에 정규식 *y* 가 발생하는 정규식 *x* 의 발생이 일치합니다 .
x|*y*	정규식 *x* 또는 정규식 *y* 가 일치합니다 .
^	문자열의 맨 처음이 일치합니다 .
$	문자열의 끝이 일치합니다 .
(x)	정규식을 그룹화합니다 .

이 예는 표 16-2 의 정규식을 사용하는 몇 가지 방법을 보여줍니다 .

```
[a-z]*://([^.:/]*[:/]|.*\.local\.com).*"
```

- `[a-z]*` 는 모든 프로토콜의 문서와 일치합니다 .
- `://` 는 (//) 앞의 (:) 와 일치합니다 .
- `[^.:/]*[:/]` 는 (.), (:) 이나 (/) 를 포함하지 않고 뒤에 (:) 또는 (/) 가 있는 모든 문자열과 일치합니다 . 따라서 유효하지 않은 호스트 이름 및 포트 번호가 있는 호스트 이름과 일치합니다 .
- `|.*\.local\.com` 은 local.com 과 같은 유효한 도메인 이름이 아니지만 .local.com 도메인의 문서와 일치합니다 .

- .*" 는 모든 파일 확장자의 문서와 일치합니다 .

참고 표 16-2 에서 설명한 것처럼 역슬래시는 특수 문자의 의미를 이스케이프하거나 제거하는 데 사용됩니다 . 마침표나 물음표 같은 문자에는 특수한 의미가 있으므로 자체 의미를 나타날 경우에는 이스케이프 처리를 해야 합니다 . 특히 마침표는 많은 URL 에 들어 있습니다 . 따라서 정규식에서 마침표의 특수 의미를 제거하려면 앞에 역슬래시를 추가해야 합니다 .

와일드카드 패턴에 대한 이해

사이트에서 액세스할 수 있는 URL 을 지정할 수 있는 와일드카드 패턴 목록을 만들 수 있습니다 . 와일드카드는 사용 방법에 따라 정규식 또는 쉘 식의 형식이 될 수 있습니다 . 일반적인 규칙은 다음과 같습니다 .

- 대상 URL 에 일치하는 모든 패턴에 정규식을 사용합니다 . 여기에는 <Object ppath=...>, URL filters 및 NameTrans, PathCheck, ObjectType 함수 등이 포함됩니다 .
- 액세스 제어용 사용자 이름 및 그룹 , 들어오는 사용자의 IP 주소 또는 DNS 이름과 같은 들어오는 클라이언트나 사용자 ID 와 일치하는 모든 패턴에 쉘 정규식을 사용합니다 (예 : <Client dns=...>).

정규식 와일드카드 패턴을 사용하여 여러 URL 을 지정할 수 있습니다 . 와일드카드를 통해 도메인 이름별로 또는 URL 에 특정 단어가 들어있는 URL 에 따라 필터링을 수행할 수 있습니다 . 예를 들어 문자열 "careers" 를 포함하는 URL 에 대한 액세스를 차단하고자 할 수 있습니다 . 이를 위해서는 `http://.*careers.*` 를 템플릿에 대한 정규식으로 지정합니다 .

새 템플릿 만들기

정규식 와일드카드 패턴을 사용하여 템플릿을 만들 수 있습니다 . 그런 다음 해당 템플릿에서 지정한 URL 에만 적용되는 측면을 구성할 수 있습니다 . 예를 들어 `.GIF` 이미지나 다른 일반 `.HTML` 파일에 대한 하나의 캐싱 구성 유형을 사용할 수도 있습니다 .

템플릿을 만들려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Templates 탭을 누릅니다 .

   Create Template 링크를 누릅니다 . Create Template 페이지가 나타납니다 .

2. Template Name 필드에 만들 템플릿 이름을 입력하고 OK 를 누릅니다 .

   이 이름은 기억하기 쉬워야 합니다 . Server Manager 가 변경 사항을 저장 및 적용할지 묻습니다 . 템플릿에 대한 정규식을 만든 후에는 나머지 단계에서 설명하는 것처럼 변경 사항을 저장할 수 있습니다 .

템플릿 적용

템플릿을 적용하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Templates 탭을 누릅니다 .
2. Apply Template 링크를 누릅니다 . Apply Template 페이지가 나타납니다 .
3. URL Prefix Wildcard 필드에 템플릿에 포함할 모든 URL 을 포함하는 정규식 와일드카드 패턴을 입력합니다 .
4. Template 목록에서 방금 추가한 새 템플릿 이름을 선택합니다 .
5. OK 를 누릅니다 .
6. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
7. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

템플릿 제거

기존 템플릿을 제거할 수 있습니다 . 템플릿을 제거하면 해당 템플릿에 대한 모든 관련 구성이 삭제됩니다 . 예를 들어 TEST 템플릿에 있는 모든 URL 에 대한 액세스 제어가 설정되어 있는 경우 , TEST 템플릿을 제거하면 해당 템플릿에 포함된 URL 에 대한 액세스 제어도 제거됩니다 .

템플릿을 제거하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Remove Template 링크를 누릅니다 . Remove Template 페이지가 나타납니다 .
3. Remove 목록에서 템플릿을 선택합니다 .
4. OK 를 누릅니다 .
5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

템플릿 보기

Server Manager 에서 만든 템플릿을 보고 편집할 수 있습니다 .

템플릿을 편집하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. View Template 링크를 누릅니다 . View Template 페이지가 화면에 표시됩니다 . 템플릿은 템플릿 및 템플릿 이름에 대한 정규식을 나열하는 표에 표시됩니다 .
3. 기존 템플릿을 편집하려면 Apply Template 페이지로 이동하는 Edit Template Assignment 링크를 누릅니다 .

리소스 제거

Remove Resource 페이지에서 전체 정규식 개체와 해당하는 구성을 삭제할 수 있습니다 . 예를 들어 Gopher 리소스를 제거하여 그에 연결된 모든 설정이 Proxy Server 의 구성 파일에서 삭제되도록 할 수 있습니다 .

리소스를 제거하려면 다음과 같이 합니다 .

1. Server Manager 에 액세스하고 Preferences 탭을 누릅니다 .
2. Remove Resource 링크를 누릅니다 . Remove Resource 페이지가 나타납니다 .
3. Remove 드롭다운 목록에서 제거하려는 리소스를 선택합니다 .
4. OK 를 누릅니다 .
5. Restart Required 를 누릅니다 . Apply Changes 페이지가 표시됩니다 .
6. Restart Proxy Server 버튼을 눌러 변경 사항을 적용합니다 .

17 장

클라이언트 자동 구성 파일 사용

많은 수의 클라이언트를 지원하는 여러 프록시 서버가 있는 경우 클라이언트 자동 구성 파일을 사용하여 모든 브라우저 클라이언트를 구성할 수 있습니다 . 자동 구성 파일에는 브라우저가 다양한 URL 을 액세스하는 데 사용할 프록시를 결정하는 JavaScript 함수가 포함되어 있습니다 .

브라우저는 시작될 때 자동 구성 파일을 로드합니다 . 사용자가 링크를 누르거나 URL 을 입력할 때마다 브라우저는 이 구성 파일을 이용하여 프록시를 사용할지 , 또 사용한다면 어떤 프록시를 사용할지 결정합니다 . 이 기능을 사용하여 조직 내 브라우저의 모든 인스턴스를 쉽게 구성할 수 있습니다 . 자동 구성 파일을 클라이언트에 가져오는 방법은 여러 가지가 있습니다 .

- 자동 구성 파일을 반환하는 웹 서버로 프록시 서버를 사용할 수 있습니다 . 브라우저를 프록시의 URL 로 향하도록 합니다 . 프록시가 웹 서버 역할을 하도록 하면 자동 구성 파일을 한 지점에 둘 수 있기 때문에 업데이트가 필요한 경우 한 파일만 변경하면 됩니다 .
- 파일을 웹 서버 , FTP 서버 또는 브라우저가 액세스할 수 있는 네트워크 디렉토리에 저장할 수 있습니다 . 파일의 URL 을 제공하여 브라우저가 해당 파일을 찾도록 구성하면 일반적인 URL 의 경우 이와 같이 동작합니다 . 복잡한 계산을 해야 하는 경우에는 (예 : 조직에 대규모 프록시 체인이 있는 경우) 파일에 액세스하는 사용자에 따라 다른 파일을 출력하는 웹 서버 CGI 프로그램을 만듭니다 .
- 자동 구성 파일을 각 브라우저의 사본과 함께 로컬에 저장할 수 있습니다 . 그러나 이 경우 파일을 업데이트할 때 파일의 복사본을 각 클라이언트로 배포해야 합니다 .

자동 구성 파일을 만드는 방법은 Server Manager 의 페이지를 사용하거나 파일을 직접 만드는 두 가지 방법이 있습니다 . 파일을 만드는 방법은 이 장의 뒷부분에서 설명합니다 .

이 장은 다음 내용으로 구성되어 있습니다 .

자동 구성 파일 이해

Proxy Server 관리자는 클라이언트 자동 구성 파일을 만들어 클라이언트에 배포할 가능성이 크기 때문에 이를 위해 이 설명서에서 자동 구성 파일 기능에 대해 설명합니다 .

자동 구성 파일의 기능

자동 구성 파일은 클라이언트 및 서버 인터넷 응용 프로그램 개발용 객체 지향 스크립트 언어인 JavaScript 로 작성됩니다 . 브라우저는 JavaScript 파일을 해석합니다 .

브라우저는 처음 로드될 때 자동 구성 파일을 다운로드합니다 . 파일을 저장할 위치는 브라우저가 URL 을 사용하여 가져올 수 있는 곳이면 됩니다 . 예를 들어 파일을 웹 서버에 둘 수 있습니다 . 또한 브라우저가 file:// URL 을 사용하여 가져올 수 있다면 네트워크 파일 시스템에 파일을 저장할 수도 있습니다 .

프록시 구성 파일은 JavaScript 로 작성됩니다 . JavaScript 파일은 브라우저가 각 URL 에 대해 사용해야 하는 프록시 서버를 결정하는 하나의 함수 (**FindProxyForURL**) 를 정의합니다 . 브라우저에서는 이 JavaScript 함수에 두 개의 매개 변수로 브라우저가 실행되는 시스템의 호스트 이름과 브라우저가 얻으려는 URL 을 전송합니다 . JavaScript 함수는 브라우저에 값을 반환하여 진행 방법을 알려줍니다 .

자동 구성 파일을 사용하면 다양한 URL 형식 , 다양한 서버 , 심지어 다양한 하루 중 시간에 대해서까지 서로 다른 프록시를 지정하거나 또는 프록시를 지정하지 않을 수 있습니다 . 즉 , 여러 대의 프록시를 예를 들어 한 서버는 .com 도메인 , 다른 서버는 .edu 도메인 , 나머지 서버는 다른 도메인 전문 서버로 사용할 수 있습니다 . 이 방법을 사용하면 여러 프록시가 모두 동일한 문서를 저장하지 않고 모든 파일의 복사본이 캐시에 한 개만 저장되므로 로드를 분산할 수 있고 프록시의 디스크를 더 효율적으로 사용할 수 있습니다 .

또한 자동 구성 파일은 프록시 장애 복구를 지원하므로 한 프록시 서버를 사용할 수 없게 되면 브라우저는 다른 프록시 서버로 투명하게 전환합니다 .

프록시를 웹 서버로 액세스

프록시 서버에 하나 이상의 자동 구성 파일을 저장하고 프록시 서버가 문서만 자동 구성 파일인 웹 서버 역할을 하도록 할 수 있습니다 . 프록시 관리자는 이 방법을 사용하여 조직 내의 클라이언트에 필요한 프록시 자동 구성 파일을 유지 보수할 수 있습니다 . 또한 중앙 위치에 파일을 보관하기 때문에 파일을 업데이트할 때 한 번만 해주면 모든 브라우저 클라이언트가 자동으로 이 업데이트를 적용합니다 .

프록시 자동 구성 파일은 *server- root*/proxy-server*id*/pac/ 디렉토리에 보관합니다 . 브라우저에서 프록시 자동 구성 파일에 대한 URL 을 입력하려면 Proxies 탭에서 파일에 대한 URL 을 입력합니다 . 프록시에 대한 URL 의 형식은 다음과 같습니다 .

`http://`*/proxy.domain:port/URI*

예를 들어 URL 은 `http://proxy.example.com` 일 수 있습니다 . URI(URL 에서 호스트 : 포트 조합의 뒤에 나오는 부분) 는 지정할 필요 없습니다 . URI 를 사용하는 경우에는 템플릿을 사용하여 다양한 자동 구성 파일에 대한 액세스를 제어할 수 있습니다 . 예를 들어 `/proxy.pac` 이라는 자동 구성 파일을 포함하는 /test 라는 URI 를 만든 경우 `http://proxy.mysite.com:8080/test/.*` 리소스 패턴으로 템플릿을 만들 수 있습니다 . 그 다음 이 템플릿을 사용하여 해당 디렉토리에 특정한 액세스 제어를 설정할 수 있습니다 .

여러 개의 자동 구성 파일을 만들어 서로 다른 URL 을 통해 액세스하도록 할 수 있습니다 . 표 17-1 은 URI 와 클라이언트에서 자동 구성 파일에 액세스하는 데 사용하는 URL 의 예를 보여 줍니다 .

표 17-1 URI 예 및 해당 URL

URI(경로)	프록시에 대한 URL
/	http://proxy.mysite.com
/employees	http://proxy.mysite.com/employees
/group1	http://proxy.mysite.com/group1
/managers	http://proxy.mysite.com/managers

역방향 프록시에서 Pac 파일 사용

역방향 프록시의 동작 방식으로 인해 프록시 서버가 역방향 프록시 역할을 하면서 `.pac` 파일을 서비스하기는 매우 어려울 수 있습니다 . 이는 프록시 서버가 파일에 대한 요청을 받고 이 요청이 로컬 `.pac` 파일에 대한 것인지 원격 문서에 대한 것인지 판단해야 하기 때문입니다 .

프록시 서버가 .pac 파일 유지 보수 및 서비스에 더하여 역방향 프록시 역할을 하도록 설정하려면 obj.conf 파일을 직접 편집하여 NameTrans 함수의 순서를 올바르게 해야 합니다 .

프록시 서버가 역방향 프록시 역할을 하도록 하려면 정규 매핑을 만듭니다 . 이 방법은 일반적으로 프록시가 모든 요청을 원격 컨텐트 서버로 라우팅하도록 합니다 . 프록시 자동 구성 파일을 /pac 와 같은 특정 디렉토리에 추가하고 매핑할 수 있습니다 . 이 경우 .pac 파일을 가져오는 모든 클라이언트는 다음과 같은 URL 을 사용하게 됩니다 .

```
http://proxy.mysite.com/pac
```

주의 이 매핑을 사용할 경우에는 원격 컨텐트 서버에 비슷한 디렉토리가 없도록 해야 합니다 .

obj.conf 파일을 편집하여 프록시 자동 구성 파일에 대한 지시문과 함수가 다른 매핑에 앞서 가장 먼저 표시되도록 합니다 . 프록시 서버는 보통 요청을 서비스하기 전에 모든 NameTrans 함수를 통하여 실행되므로 이 지시문과 함수가 가장 먼저 와야 합니다 . 하지만 자동 구성 파일을 사용하면 프록시는 즉시 경로를 인식하고 .pac 파일을 반환합니다 .

다음은 역방향 프록시를 사용하고 자동 구성 파일을 유지 보수하는 obj.conf 파일이 있는 경우의 예입니다 .

```
<Object name="default">
NameTrans from="file:" fn="map" to="ftp:"
NameTrans from="/pac" fn="pac-map" name="file" to="/ns-home/proxy/pac/proxy.pac"
NameTrans fn="redirect" from="http://foo.*" url="http://www.acme.com"
NameTrans from="/ns-icons" fn="pfx2dir" dir="/ns-home/ns-icons" name="file"
NameTrans fn="reverse-map" from="http://web.acme.com" to="http://proxy.acme.com:8080"
NameTrans fn="map" from="http://proxy.acme.com:8080" to="http://web.acme.com"
NameTrans fn="map" from="/" to="http://web.acme.com"
PathCheck fn="url-check"
Service fn="deny-service"
AddLog fn="flex-log" name="access"
AddLog fn="urldb-record"
</Object>
```

Server Manager 페이지를 사용하여 자동 구성 파일 생성

Server Manager 페이지를 사용하여 자동 구성 파일을 생성하려면 다음을 수행합니다 .

1. Server Manager 에 액세스하고 Routing 탭을 선택합니다 .
2. Create / Edit Autoconfiguration File 링크를 누릅니다 . 프록시 시스템에 있는 모든 자동 구성 파일의 목록이 있는 페이지가 표시됩니다 . 자동 구성 파일을 클릭하면 편집할 수 있습니다 . 나머지 단계는 새 파일을 만드는 방법을 설명합니다 .
3. URL 의 경로 부분인 URI 를 입력하면 (선택 사항) 클라이언트가 프록시에서 자동 구성 파일을 가져올 때 사용할 수 있습니다 . 예를 들어 슬래시 (/) 를 입력하면 클라이언트가 해당 파일을 프록시의 기본 문서로 액세스합니다 (웹 서버의 `index.html` 파일과 유사). 이렇게 하면 클라이언트는 도메인 이름만 사용하여 이 자동 구성 파일에 대한 프록시에 액세스합니다 . 여러 개의 URI 를 사용할 수 있으며 각 URI 에 대해 별도의 자동 구성 파일을 만들 수 있습니다 .
4. `.pac` 확장자를 사용하는 자동 구성 파일의 이름을 입력합니다 . 파일이 하나인 경우에는 간단히 `proxy.pac` 이라는 이름을 사용할 수 있습니다 (pac: proxy autoconfiguration 의 약자). 모든 자동 구성 파일은 하나의 JavaScript 함수가 있는 ASCII 텍스트 파일입니다 .
5. OK 를 누릅니다 . 다른 페이지가 표시됩니다 . 자동 구성 파일을 만들려면 이 페이지를 사용하십시오 . 페이지에 표시되는 항목의 순서는 클라이언트에 따라 달라집니다 . 페이지에 표시되는 항목은 다음과 같습니다 .

- **Never Go Direct To Remote Server** 는 Navigator 가 항상 프록시를 사용하도록 합니다 . 프록시 서버가 실행 중이 아닌 경우에 사용할 보조 프록시 서버를 지정할 수 있습니다 .
- **Go Direct To Remote Server When** 은 특정한 경우 프록시 서버를 우회하도록 합니다 . Navigator 는 페이지에 나열되는 다음 옵션에 따라 이러한 특정한 경우를 결정합니다 .
  - **Connecting To Non-fully Qualified Host Names** 는 사용자가 해당 컴퓨터 이름만 지정한 경우 Navigator 가 서버로 직접 이동하도록 합니다 . 예를 들어 `winternal.mysite.com` 이라는 내부 웹 서버가 있는 경우 사용자는 정규화된 도메인을 입력하는 대신 `http://winternal` 만 입력할 수 있습니다 . 이 경우 Navigator 는 프록시를 거치지 않고 해당 웹 서버로 직접 이동합니다 .
  - **Connecting To A Host In Domain** 은 Navigator 가 직접 액세스할 수 있는 도메인 이름을 최대 3 개까지 지정합니다 . 도메인 이름을 지정하는 경우 점 (.) 으로 시작해야 합니다 . 예를 들어 `.example.com` 과 같이 입력합니다 .

- **Connecting To A Resolvable Host** 는 클라이언트가 호스트를 확인할 수 있는 경우 Navigator 가 해당 서버로 직접 이동하도록 합니다 . 일반적으로 이 옵션은 DNS 가 로컬 (내부) 호스트만 확인하도록 설정된 경우 사용됩니다 . 클라이언트는 로컬 네트워크 외부의 서버로 연결하는 경우에는 Proxy Server 를 사용하게 됩니다 .

주의 이 옵션을 사용하면 클라이언트는 모든 요청에 대해 DNS 를 참조합니다 . 따라서 클라이언트 입장에서 성능이 떨어지게 됩니다 . 이러한 성능 저하가 있으므로 이 옵션의 사용은 피해야 합니다 .

- **Connecting To A Host In Subnet**은 클라이언트가 특정 서브넷에서 서버에 액세스하는 경우 Navigator 가 서버로 직접 이동하도록 합니다 . 이 옵션은 지역적으로 많은 서브넷이 있는 기업에게 유용합니다 . 예를 들어 일부 기업은 전세계 특정 지역에 대한 각 서브넷에 하나의 도메인 이름을 적용할 수 있습니다 .

주의 이 옵션을 사용하면 클라이언트는 모든 요청에 대해 DNS 를 참조합니다 . 따라서 클라이언트 입장에서 성능이 떨어지게 됩니다 . 이러한 성능 저하가 있으므로 이 옵션의 사용은 피해야 합니다 .

- **Except When Connecting To Hosts**는 서버로 직접 이동하는 규칙에 예외를 지정합니다 . 예를 들어 `.example.com` 을 직접 이동할 도메인으로 입력한 경우 `home.example.com` 은 예외로 지정할 수 있습니다 . 이렇게 하면 Navigator 는 `home.example.com` 으로 이동하는 경우에는 프록시를 사용하며 , 그 외 `example.com` 도메인에 있는 모든 서버로는 직접 이동하게 됩니다 .
- **Secondary Failover Proxy** 는 프록시 서버가 실행 중이 아닌 경우에 사용할 보조 프록시를 지정합니다 .
- **Failover Direct** 는 프록시 서버가 실행 중이 아닌 경우 Navigator 가 서버로 직접 이동하도록 합니다 . 보조 장애 복구 프록시를 지정하면 Navigator 는 서버로 직접 이동하기 전에 이 보조 Proxy Server 를 확인합니다 .

6. OK 를 눌러 자동 구성 파일을 생성합니다 . 이 파일은 *server-root*`/proxy-server`*id*`/pac` 디렉토리에 저장됩니다 . 파일이 올바르게 만들어졌다는 확인 메시지가 나타납니다 . 이 단계를 반복하여 필요한 수 만큼 자동 구성 파일을 만듭니다 .

자동 구성 파일을 만든 다음에는 모든 프록시 서버 사용자들에게 자동 구성 파일을 올바르게 가리키도록 하거나 또는 직접 Navigator 사본을 구성하십시오 .

자동 구성 파일 직접 만들기

이 절에서는 자동 구성 파일을 직접 만드는 방법에 대해 설명합니다 .

프록시 자동 구성 파일은 클라이언트측 JavaScript 를 사용하여 작성됩니다 . 각 파일은 **FindProxyForURL** 이라는 하나의 JavaScript 함수를 포함하고 있으며 이 함수는 브라우저가 각 URL 에 대해 사용해야 하는 프록시 서버를 결정합니다 . 브라우저에서는 이 JavaScript 함수에 두 개의 매개 변수로 대상 원본 서버의 호스트 이름과 브라우저가 얻으려는 URL 을 전송합니다 . JavaScript 함수는 Navigator 에 값을 반환하여 진행 방법을 알려줍니다 . 다음 절에서는 함수 구문과 가능한 반환 값에 대해 설명합니다 .

FindProxyForURL 함수

FindProxyFor URL 함수의 구문은 다음과 같습니다 .

```
function FindProxyForURL(url, host)
{
   ...
}
```

브라우저는 액세스하는 모든 URL 에 대해 **url** 과 **host** 매개 변수를 전송하고 다음 방법으로 함수를 호출합니다 .

```
ret = FindProxyForURL(url, host);
```

url 은 브라우저가 액세스하는 전체 URL 입니다 .

host 는 액세스되는 URL 에서 추출한 호스트 이름입니다 . 호스트 이름은 단지 편의를 위한 것으로 , **://** 와 첫 **:** 사이의 문자열 또는 **://** 와 / 사이의 문자열과 같습니다 . 포트 번호는 이 매개 변수에 포함되지 않습니다 . 필요한 경우 URL 에서 추출할 수 있습니다 .

ret(반환 값) 는 구성을 설명하는 문자열입니다 .

함수 반환 값

자동 구성 파일은 **FindProxyForURL** 함수를 포함합니다 . 이 함수는 클라이언트 호스트 이름과 해당 클라이언트가 액세스하는 URL 을 매개 변수로 사용합니다 . 이 함수는 브라우저에 진행 방법을 알려주는 하나의 문자열을 반환합니다 . 문자열이 Null 인 경우 프록시가 사용되지 않습니다 . 문자열은 표 17-2 에 표시된 구성 요소를 수에 제한 없이 세미콜론으로 분리하여 포함할 수 있습니다 .

표 17-2 FindProxyForURL 반환 값

반환 값	브라우저의 결과 동작
`DIRECT`	프록시를 통하지 않고 서버에 직접 연결합니다 .
`PROXY` *host:port*	지정된 프록시와 포트 번호를 사용합니다 . 여러 값이 세미콜론으로 분리되어 있으면 첫 번째 프록시가 사용됩니다 . 해당 프록시가 실패하면 다음 프록시가 사용되며 , 실패 시 계속 마찬가지 방법이 적용됩니다 .
`SOCKS` *host:port*	지정된 SOCKS 서버를 사용합니다 . 여러 값이 세미콜론으로 분리되어 있으면 첫 번째 프록시가 사용됩니다 . 해당 프록시가 실패하면 다음 프록시가 사용되며 , 실패 시 계속 마찬가지 방법이 적용됩니다 .

브라우저는 사용할 수 없는 프록시 서버를 만나면 응답하지 않은 이전 프록시를 30 분 후에 자동으로 다시 시도하고 이후 30 분 간격으로 계속 시도합니다 . 따라서 프록시 서버를 임시로 종료한 경우 클라이언트는 프록시가 재시작된 이후 30 분 내에 프록시 사용을 재개하게 됩니다 .

모든 프록시가 다운된 상태고 DIRECT 반환 값이 지정되지 않으면 브라우저는 사용자에게 임시로 프록시를 무시하고 직접 연결을 시도할 것인지 묻습니다 . Navigator 는 20 분 후에 프록시를 재시도할 것인지 물으며 , 이후 20 분 간격으로 계속해서 묻습니다 .

다음 예에서 반환 값은 브라우저에게 8080 포트의 `w3proxy.example.com` 이라는 프록시를 사용하고 , 이 프록시를 사용할 수 없는 경우 8080 포트의 `proxy1.example.com` 이라는 프록시를 사용하라고 알립니다 .

```
PROXY w3proxy.example.com:8080; PROXY proxy1.example.com:8080
```

다음 예에서 기본 프록시는 `w3proxy.example.com:8080` 이며 , 이 프록시를 사용할 수 없는 경우 브라우저는 `proxy1.example.com:8080` 을 사용합니다 . 두 프록시 모두 사용할 수 없는 경우 브라우저는 서버로 직접 이동하며 20 분 후 브라우저는 사용자에게 첫 번째 프록시를 다시 시도할지 묻습니다 .

```
PROXY w3proxy.example.com:8080; PROXY proxy1.example.com:8080; DIRECT
```

JavaScript 함수 및 환경

JavaScript 에는 프록시에서 유용한 여러 가지의 미리 정의된 함수와 환경 조건이 있습니다 . 각 함수는 특정 조건이 충족되는지 여부를 확인하고 true 및 false 값을 반환합니다 . 관련된 유틸리티 함수는 DNS 호스트 이름 또는 IP 주소를 반환하기 때문에 예외입니다 . 기본 **FindProxyForURL** 함수에서 이러한 함수를 사용하여 브라우저로 전송할 반환 값을 결정할 수 있습니다 . 이러한 함수를 사용하는 방법은 이 장의 뒷부분에 나오는 예를 참조하십시오 .

각 함수 또는 환경 조건은 이 절에 설명되어 있습니다 . 프록시와 브라우저 통합에 적용되는 함수 및 환경 조건은 다음과 같습니다 .

호스트 이름 기반 함수

- dnsDomainIs()
- isInNet()
- isPlainhost name()
- isResolvable()
- localHostOrDomainIs()

관련 유틸리티 함수 :

- dnsDomainLevels()
- dnsResolve()
- myIpAddress()

URL/ 호스트 이름 기반 조건

- shExpMatch()

시간 기반 조건

- dateRange()
- timeRange()
- weekdayRange()

호스트 이름 기반 함수

호스트 이름 기반 함수를 통해 호스트 이름 또는 IP 주소를 이용하여 사용할 프록시를 결정할 수 있습니다 .

dnsDomainIs(host, domain)

dnsDomainIs() 함수는 URL 호스트 이름이 지정된 DNS 도메인에 속하는지 여부를 감지합니다 . 이 함수는 " 예제 1: 로컬 호스트를 제외한 모든 서버 프록시 " (366 페이지) 및 " 예제 2: 방화벽 외부의 로컬 서버 프록시 " (366 페이지) 에서 설명한 대로 로컬 도메인에 대해서는 프록시를 사용하지 않도록 브라우저를 구성하는 경우 유용합니다 .

이 함수는 또한 요청을 수신하는 프록시가 URL 이 속한 DNS 도메인을 기반으로 프록시 그룹에서 선택되는 상황에서 로드 밸런싱을 위해 여러 프록시를 사용하는 경우에도 유용합니다 . 예를 들어 `.edu` 가 포함된 URL 을 한 프록시에 , `.com` 이 포함된 URL 을 다른 프록시에 지정하여 로드 밸런싱하는 경우 **dnsDomainIs()** 를 사용하여 URL 호스트 이름을 확인할 수 있습니다 .

매개 변수 :
host 는 URL 의 호스트 이름입니다 .

domain 은 호스트 이름을 시험할 대상 도메인 이름입니다 .

반환 :
true 또는 false

예를 들면 다음과 같습니다 .
다음 문은 true 입니다 .

```
dnsDomainIs("www.example.com", ".example.com")
```

다음 문은 false 입니다 .

```
dnsDomainIs("www", ".example.com")
dnsDomainIs("www.mcom.com", ".example.com")
```

isInNet(host, pattern, mask)

isInNet() 함수를 통해 URL 호스트 이름을 IP 주소로 변환하고 이 주소가 마스크에서 지정한 서브넷에 속하는지 시험할 수 있습니다 . 이는 SOCKS 에서 사용하는 IP 주소 패턴 일치와 같은 유형입니다 . " 예제 4: 서브넷으로 직접 연결 " (368 페이지) 을 참조하십시오 .

매개 변수 :
host 는 DNS 호스트 이름 또는 IP 주소입니다 . 호스트 이름이 전달되면 이 함수는 호스트 이름을 IP 주소로 변환합니다 .

pattern 은 점으로 분리된 형식의 IP 주소 패턴입니다 .

mask 는 IP 주소에서 일치시켜야 하는 부분을 결정하는 IP 주소 패턴 마스크입니다 . 값이 0 이면 무시 , 255 이면 일치를 의미합니다 . 이 함수는 호스트의 IP 주소가 지정된 IP 주소 패턴과 일치하면 true 입니다 .

반환 :
true 또는 false

예를 들면 다음과 같습니다 .
이 문은 호스트의 IP 주소가 정확히 198.95.249.79 와 일치하는 경우에만 true 입니다 .

```
isInNet(host, "198.95.249.79", "255.255.255.255")
```

이 문은 호스트의 IP 주소가 198.95.*.* 와 일치하는 경우에만 true 입니다 .

```
isInNet(host, "198.95.0.0", "255.255.0.0")
```

isPlainhost name(host)

isPlainhost name() 함수는 요청된 URL 의 호스트 이름이 단순한 호스트 이름인지 또는 정규화된 도메인 이름인지 감지합니다 . 이 함수는 " 예제 1: 로컬 호스트를 제외한 모든 서버 프록시 " (366 페이지) 및 " 예제 2: 방화벽 외부의 로컬 서버 프록시 " (366 페이지) 에서 설명한 대로 Netscape Navigator 가 로컬 서버에 직접 연결하도록 하려는 경우 유용합니다 .

매개 변수 :
host 는 호스트 이름에 도메인 이름이 없는 경우에만 (점으로 분리된 구획이 없음) URL 의 호스트 이름 (포트 번호 제외) 입니다 .

반환 :
host 가 로컬인 경우 true, **host** 가 원격인 경우 false

예를 들면 다음과 같습니다 .

```
isPlainhost name("host")
```

host 가 www 와 같은 형태면 true 를 , 호스트가 `www.example.com` 과 같은 형태면 false 를 반환합니다 .

isResolvable(host)

방화벽 내의 DNS 가 내부 호스트만 인식할 경우 **isResolvable()** 함수를 사용하여 호스트 이름이 네트워크 내부인지 외부인지 시험할 수 있습니다 . 이 함수를 사용하여 브라우저가 내부 서버에 대해서는 직접 연결을 사용하고 외부 서버에 대해서만 프록시를 사용하도록 구성할 수 있습니다 . 방화벽 내의 내부 호스트가 다른 내부 호스트

의 DNS 도메인 이름은 변환할 수 있지만 외부 호스트는 모두 변환할 수 없는 사이트에서 이 함수가 유용합니다 . **isResolvable()** 함수는 호스트 이름을 IP 주소로 변환하기 위해 시도하면서 DNS 를 참조합니다 . " 예제 3: 변환되지 않은 호스트만 프록시 " (367 페이지) 를 참조하십시오 .

매개 변수 :
host 는 URL 의 호스트 이름입니다 . 호스트 이름 변환을 시도하며 성공하는 경우 true 를 반환합니다 .

반환 :
호스트 이름을 변환할 수 있으면 true, 변환할 수 없으면 false 를 반환합니다 .

예를 들면 다음과 같습니다 .

```
isResolvable("host")
```

host 가 `www` 와 같은 형태고 DNS 를 통해 변환할 수 있으면 이 함수는 true 를 반환합니다 .

localHostOrDomainIs(host, hostdom)

localHostOrDomainIs() 함수는 정규화된 도메인 이름 또는 단순한 호스트 이름으로 액세스되는 로컬 호스트를 지정합니다 . " 예제 2: 방화벽 외부의 로컬 서버 프록시 " (366 페이지) 를 참조하십시오 .

localHostOrDomainIs() 함수는 호스트 이름이 지정된 호스트 이름과 정확히 일치하거나 또는 호스트 이름에 정규화되지 않은 호스트 이름과 일치하는 도메인 이름 부분이 없는 경우 `true` 를 반환합니다 .

매개 변수 :
host 는 URL 의 호스트 이름입니다 .

hostdom 은 일치시킬 정규화된 호스트 이름입니다 .

반환 :
true 또는 false

예를 들면 다음과 같습니다 .
다음 문은 true 입니다 (정확히 일치).

```
localHostOrDomainIs("www.example.com", "www.example.com")
```

다음 문은 true 입니다 (호스트 이름 일치 , 도메인 이름 지정되지 않음).

```
localHostOrDomainIs("www", "www.example.com")
```

다음 문은 false 입니다 (도메인 이름 불일치).

```
localHostOrDomainIs("www.mcom.com", "www.example.com")
```

다음 문은 false 입니다 (호스트 이름 불일치).

```
localHostOrDomainIs("home.example.com", "www.example.com")
```

관련 유틸리티 함수

관련 유틸리티 함수를 통해 도메인 단계 , Netscape Navigator 가 실행되는 호스트 또는 호스트의 IP 주소를 확인할 수 있습니다 .

dnsDomainLevels(host)

dnsDomainLevels() 함수는 URL 호스트 이름에서 DNS 단계의 수 (점의 수) 를 확인합니다 .

매개 변수 :
host 는 URL 의 호스트 이름입니다 .

반환 :
DNS 도메인 단계의 수 (정수)

예를 들면 다음과 같습니다 .

```
dnsDomainLevels("www")
```

0 을 반환합니다 .

```
dnsDomainLevels("www.example.com")
```

2 를 반환합니다 .

dnsResolve(host)

dnsResolve() 함수는 지정된 호스트 (보통 URL 에 있음) 를 IP 주소로 변환합니다 . JavaScript 함수가 기존 함수로 수행할 수 있는 것보다 더 고급의 패턴 일치를 수행해야 하는 경우 이 함수가 유용합니다 .

매개 변수 :
host 는 변환할 호스트 이름입니다 . 지정된 DNS 호스트 이름을 IP 주소로 변환하고 점으로 분리된 문자열로 반환합니다 .

반환 :
점으로 4 부분으로 분리된 IP 주소의 문자열 값

예를 들면 다음과 같습니다 .
다음 예는 198.95.249.79 문자열을 반환합니다 .

```
dnsResolve("home.example.com")
```

myIpAddress()

myIpAddress() 함수는 브라우저가 실행되는 호스트에 따라 JavaScript 함수가 다르게 작동해야 하는 경우 유용합니다 . 이 함수는 브라우저를 실행 중인 컴퓨터의 IP 주소를 반환합니다 .

반환 :
점으로 4 부분으로 분리된 IP 주소의 문자열 값

예를 들면 다음과 같습니다 .
Navigator 를 `home.example.com` 컴퓨터에서 실행 중인 경우 다음 예는 198.95.249.79 문자열을 반환합니다 .

```
myIpAddress()
```

URL/ 호스트 이름 기반 조건

로드 밸런싱 및 라우팅을 위해 호스트 이름 또는 URL 을 일치시킬 수 있습니다 .

shExpMatch(str, shexp)

shExpMatch() 함수는 URL 호스트 이름 또는 URL 자체를 일치시킵니다 . 이 함수는 주로 로드 밸런싱 및 URL 을 여러 프록시 서버로 지능적으로 라우팅하는 데 사용됩니다 .

매개 변수 :
str 은 비교할 문자열입니다 (예 : URL 또는 호스트 이름).

shexp 는 비교 대상인 쉘 표현식입니다 .

문자열이 지정된 쉘 표현식과 일치하면 이 표현식은 true 입니다 . " 예제 6: shExpMatch() 를 사용한 프록시 로드 밸런싱 " (369 페이지) 을 참조하십시오 .

반환 :
true 또는 false

예를 들면 다음과 같습니다 .
첫 번째 예는 true 를 , 두 번째 예는 false 를 반환합니다 .

```
shExpMatch("http://home.example.com/people/index.html",
           ".*/people/.*")
```

```
shExpMatch("http://home.example.com/people/yourpage/index.html",
           ".*/mypage/.*")
```

시간 기반 조건

FindProxyForURL 함수가 날짜 , 시간 , 주중 요일에 따라 다르게 작동하도록 할 수 있습니다 .

dateRange(년 , 월 , 일 ...)

dateRange() 함수는 특정 날짜나 날짜 범위를 감지할 수 있습니다 (예 : 1996 년 4 월 19 일부터 1996 년 5 월 3 일). 이 함수는 프록시에 유지 보수를 위한 다운 시간이 정기적으로 예약되어 있는 경우 등
FindProxyForURL 함수가 날짜에 따라 다르게 작동하도록 할 때 유용합니다 .

날짜 범위는 다음과 같이 여러 가지 방법으로 지정할 수 있습니다 .

```
dateRange(day)
dateRange(day1, day2)
dateRange(mon)
dateRange(month1, month2)
dateRange(year)
dateRange(year1, year2)
dateRange(day1, month1, day2, month2)
dateRange(month1, year1, month2, year2)
dateRange(day1, month1, year1, day2, month2, year2)
dateRange(day1, month1, year1, day2, month2, year2, gmt)
```

매개 변수 :
day 는 월의 일자에 대한 1 부터 31 까지의 정수입니다 .

month 는 다음 월 문자열 중 하나입니다 .
`JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC`

year 는 연도에 대한 4 자릿수 정수입니다 (예 : 1996).

gmt 는 그리니치 표준시로 시간을 비교하도록 하는 GMT 문자열 또는 빈칸이며 이 경우 로컬 시간대에 있는 것으로 가정합니다 . GMT 매개 변수는 모든 호출 프로파일에서 항상 마지막 매개 변수로 지정될 수 있습니다 . 하나의 값만 지정된 경우 (각 년 , 월 , 일 범주에서) 함수는 지정한 항목과 일치하는 날짜에만 true 를 반환합니다 . 두 값이 지정된 경우 첫 번째 지정 날짜에서 두 번째 지정 날짜 사이면 결과는 true 입니다 .

예를 들면 다음과 같습니다 .
이 문은 로컬 시간대에서 각 월의 첫 번째 일이면 true 입니다 .

```
dateRange(1)
```

이 문은 그리니치 표준시에서 각 월의 첫 번째 일이면 true 입니다 .
```
dateRange(1, "GMT")
```

이 문은 각 월의 전반기에 대해 true 입니다 .
```
dateRange(1, 15)
```

이 문은 각 연도의 12 월 24 일에서 true 입니다 .
```
dateRange(24, "DEC")
```

이 문은 1995 년 12 월 24 일에서 true 입니다 .
```
dateRange(24, "DEC", 1995)
```

이 문은 해당 년도 1 분기 동안 true 입니다 .
```
dateRange("JAN", "MAR")
```

이 문은 각 연도의 6 월 1 일부터 8 월 15 일까지 true 입니다 .
```
dateRange(1, "JUN", 15, "AUG")
```

이 문은 1995 년 6 월 1 일부터 1995 년 8 월 15 일까지 true 입니다 .
```
dateRange(1, "JUN", 15, 1995, "AUG", 1995)
```

이 문은 1995 년 10 월부터 1996 년 3 월까지 true 입니다 .
```
dateRange("OCT", 1995, "MAR", 1996)
```

이 문은 1995 년 내내 true 입니다 .
```
dateRange(1995)
```

이 문은 1995 년 초부터 1997 년 말까지 true 입니다 .
```
dateRange(1995, 1997)
```

timeRange(시간 , 분 , 초 ...)

timeRange 함수는 오후 9 시부터 오전 12 시까지와 같은 하루 중의 특정 시간 또는 시간 범위를 감지합니다 . 이 함수는 시간에 따라 **FindProxyForURL** 함수를 다르게 작동하도록 하려는 경우 유용합니다 .

```
timeRange(hour)
timeRange(hour1, hour2)
timeRange(hour1, min1, hour2, min2)
timeRange(hour1, min1, sec1, hour2, min2, sec2)
```

매개 변수 :

hour 는 시간을 표시하는 0 부터 23 까지의 수입니다 (0 은 자정 , 23 은 오후 11:00).

min 은 분을 표시하는 0 부터 59 까지의 수입니다 .

sec 는 초를 표시하는 0 부터 59 까지의 수입니다 .

gmt 는 GMT 시간대인 경우 GMT 문자열이며 로컬 시간대인 경우 지정되지 않습니다 . 이 매개 변수는 각 매개 변수 프로파일과 함께 사용할 수 있으며 항상 마지막 매개 변수입니다 .

반환 :
true 또는 false

예를 들면 다음과 같습니다 .
이 문은 정오에서 오후 1:00 까지 true 입니다 .
```
timerange(12, 13)
```

이 문은 GMT 정오에서 오후 12:59 까지 true 입니다
```
timerange(12, "GMT")
```

이 문은 오전 9:00 에서 오후 5:00 까지 true 입니다 .
```
timerange(9, 17)
```

자정부터 자정 후 30 초 사이까지 true 입니다 .
```
timerange(0, 0, 0, 0, 0, 30)
```

weekdayRange(wd1, wd2, gmt)

weekdayRange() 함수는 특정 주중 요일 또는 월요일부터 금요일까지 등과 같은 주중 요일 범위를 감지합니다 . 이 함수는 **FindProxyForURL** 함수를 주중 요일에 따라 다르게 작동하도록 하려는 경우 유용합니다 .

매개 변수 :
wd1 및 **wd2** 는 다음 요일 문자열 중 하나입니다 .
```
SUN MON TUE WED THU FRI SAT
```

gmt 는 그리니치 표준시인 경우 GMT 또는 로컬 시간대인 경우 빈칸으로 둡니다 .

첫 번째 매개 변수인 wd1 만 필수 매개 변수입니다 . wd2, gmt, 또는 둘 다 빈칸으로 둘 수 있습니다 .

매개 변수가 하나 뿐이면 함수는 매개 변수가 나타내는 요일에서 true 값을 반환합니다 . 두 번째 매개 변수로 GMT 문자열이 지정된 경우에는 GMT 시간이 , 그렇지 않은 경우에는 로컬 시간대가 적용됩니다 .

wd1 과 wd2 가 모두 정의된 경우 현재 요일이 이 두 요일의 사이에 있으면 조건은 true 입니다 . 경계선도 범위에 포함됩니다 . 매개 변수의 순서가 중요합니다 . "MON," "WED" 는 월요일부터 수요일까지입니다 . 그러나 "WED," "MON" 은 수요일부터 다음주 월요일까지입니다 .

예를 들면 다음과 같습니다 .
다음은 월요일부터 금요일까지 true 입니다 (로컬 시간대).
```
weekdayRange("MON", "FRI")
```

다음은 그리니치 표준시로 월요일부터 금요일까지 true 입니다 .
```
weekdayRange("MON", "FRI", "GMT")
```

다음은 로컬 시간으로 토요일에 true 입니다 .
```
weekdayRange("SAT")
```

다음은 그리니치 표준시로 토요일에 true 입니다 .
```
weekdayRange("SAT", "GMT")
```

다음은 금요일부터 월요일까지 true 입니다 (순서 중요).
```
weekdayRange("FRI", "MON")
```

세부적인 예

예제 1: 로컬 호스트를 제외한 모든 서버 프록시

이 예제에서 Netscape Navigator 는 정규화되지 않고 로컬 도메인에 속하는 모든 호스트에 직접 연결합니다 . 그 외의 모든 연결은 `w3proxy.example.com:8080` 이라는 프록시를 통합니다 .

참고 프록시가 다운되면 자동으로 직접 연결이 됩니다 .

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) ||
        dnsDomainIs(host, ".example.com") ||
        dnsDomainIs(host, ".mcom.com"))
        return "DIRECT";
    else
        return "PROXY w3proxy.example.com:8080; DIRECT";
}
```

예제 2: 방화벽 외부의 로컬 서버 프록시

이 예제는 앞의 예제와 같지만 방화벽 외부의 로컬 서버에 대해 프록시를 사용합니다 . 로컬 도메인에 속하지만 방화벽 외부에 있고 프록시 서버를 통해서만 연결할 수 있는 호스트 (예 : 기본 웹 서버) 인 경우 이와 같은 예외는 **localHostOrDomainIs()** 함수를 사용하여 다음과 같이 처리합니다 :

```
function FindProxyForURL(url, host)
{
    if ((isPlainhost name(host) ||
    dnsDomainIs(host, ".example.com")) &&
    !localHostOrDomainIs(host, "www.example.com") &&
    !localHostOrDoaminIs(host, "merchant.example.com"))
        return "DIRECT";
    else
        return "PROXY w3proxy.example.com:8080; DIRECT";
}
```

이 예에서는 example.com 도메인의 로컬 호스트를 제외한 모든 대상에 대해 프록시를 사용합니다 . www.example.com 과 merchant.example.com 호스트도 프록시를 통하여 연결합니다 .

예외의 순서는 효율성을 높입니다 . **localHostOrDomainIs()** 함수는 모든 URL 이 아니라 로컬 도메인에 있는 URL 에 대해서만 실행됩니다 . 특히 *and* 표현식 앞에 *or* 표현식이 괄호 안에 있음을 유의하십시오 .

예제 3: 변환되지 않은 호스트만 프록시

이 예제는 내부 DNS 를 설정하여 내부 호스트 이름만 변환하는 환경에서 동작하며 , 변환할 수 없는 호스트에 대해서만 프록시를 사용하는 것이 목적입니다 .

```
function FindProxyForURL(url, host)
{
    if (isResolvable(host))
            return "DIRECT";
        else
            return "PROXY proxy.mydomain.com:8080";
}
```

이 예제는 매번 DNS 를 참조해야 하므로 다른 규칙과 그룹화하여 다른 규칙이 결과를 생성하지 않는 경우에만 DNS 를 참조하도록 해야 합니다 .

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) ||
        dnsDomainIs(host, ".mydomain.com") ||
        isResolvable(host))
        return "DIRECT";
    else
        return "PROXY proxy.mydomain.com:8080";
}
```

예제 4: 서브넷으로 직접 연결

이 예제에서 지정된 서브넷의 모든 호스트는 프록시를 통하는 다른 호스트로 직접 연결됩니다 .

```
function FindProxyForURL(url, host)
{
    if (isInNet(host, "198.95.0.0", "255.255.0.0"))
        return "DIRECT";
    else

        return "PROXY proxy.mydomain.com:8080";
}
```

다음과 같이 처음에 리던던트 규칙을 추가하면 이 예제에서 DNS 사용을 최소화할 수 있습니다 .

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) ||
        dnsDomainIs(host, ".mydomain.com") ||
        isInNet(host, "198.95.0.0", "255.255.0.0"))
        return "DIRECT";
    else
        return "PROXY proxy.mydomain.com:8080";
}
```

예제 5: dnsDomainIs() 을 사용한 프록시 로드 밸런싱

이 예제는 더 복잡합니다 . 네 개의 프록시 서버가 있으며 이 중 한 프록시 서버는 다른 서버에 대한 상시 대기 역할을 하여 나머지 세 프록시 서버 중 하나가 다운되면 이를 대신합니다 . 나머지 세 프록시 서버는 URL 패턴을 기반으로 로드를 공유함으로써 캐시의 효율성을 높입니다 (한 문서에 대한 사본이 세 개의 프록시 서버 모두에 있는 것이 아니라 한 프록시 서버에만 있음). 로드는 표 17-3 에서와 같이 분산됩니다 .

표 17-3 프록시 로드 밸런싱

프록시	목적
#1	.com 도메인
#2	.edu 도메인
#3	그 외 모든 도메인
#4	상시 대기

모든 로컬 액세스는 직접 연결됩니다 . 모든 프록시 서버는 8080 포트에서 실행됩니다 . JavaScript 에서 + 연산자를 사용하여 문자열을 연결할 수 있습니다 .

```
function FindProxyForURL(url, host)
{
    if (isPlainhost name(host) || dnsDomainIs(host, ".mydomain.com"))
        return "DIRECT";

    else if (dnsDomainIs(host, ".com"))
        return "PROXY proxy1.mydomain.com:8080; " +
               "PROXY proxy4.mydomain.com:8080";

    else if (dnsDomainIs(host, ".edu"))
        return "PROXY proxy2.mydomain.com:8080; " +
               "PROXY proxy4.mydomain.com:8080";

    else
        return "PROXY proxy3.mydomain.com:8080; " +
               "PROXY proxy4.mydomain.com:8080";
}
```

예제 6: shExpMatch() 를 사용한 프록시 로드 밸런싱

이 예제는 기본적으로 예제 5 와 동일하지만 **dnsDomainIs()** 대신 **shExpMatch()** 를 사용합니다 .

```
function FindProxyForURL(url, host)
{
if (isPlainhost name(host) || dnsDomainIs(host, ".mydomain.com"))
    return "DIRECT";
else if (shExpMatch(host, "*.com"))
    return "PROXY proxy1.mydomain.com:8080; " +
           "PROXY proxy4.mydomain.com:8080";
else if (shExpMatch(host, "*.edu"))
    return "PROXY proxy2.mydomain.com:8080; " +
           "PROXY proxy4.mydomain.com:8080";
else
    return "PROXY proxy3.mydomain.com:8080; " +
           "PROXY proxy4.mydomain.com:8080";
}
```

예제 7: 특정 프로토콜 프록시

프록시를 특정 프로토콜에 대해 설정할 수 있습니다 . 대부분의 표준 JavaScript 기능은 **FindProxyForURL()** 함수에서 사용할 수 있습니다 . 예를 들어 프로토콜을 기반으로 다른 프록시를 설정하려면 다음과 같이 **substring()** 함수를 사용할 수 있습니다 .

```
function FindProxyForURL(url, host)
{
    if (url.substring(0, 5) == "http:") {
        return "PROXY http-proxy.mydomain.com:8080";
    }
    else if (url.substring(0, 4) == "ftp:") {
        return "PROXY ftp-proxy.mydomain.com:8080";
    }
    else if (url.substring(0, 7) == "gopher:") {
        return "PROXY gopher-proxy.mydomain.com:8080";
    }
    else if (url.substring(0, 6) == "https:" ||
            url.substring(0, 6) == "snews:") {
        return "PROXY security-proxy.mydomain.com:8080";
    }
    else {
        return "DIRECT";
    }
}
```

또한 **shExpMatch()** 함수를 사용해서도 마찬가지 기능을 수행할 수 있습니다 . 예를 들면 다음과 같습니다 .

```
...
if (shExpMatch(url, "http:*")) {
    return "PROXY http-proxy.mydomain.com:8080;
}

...
```

제 5 부

부록

부록 A

ACL 파일 구문

ACL(Access Control List) 파일은 Proxy Server 리소스에 액세스할 수 있는 사용자를 정의한 목록을 포함하는 텍스트 파일입니다 . 기본적으로 Proxy Server 에는 서버에 액세스할 수 있는 모든 목록이 포함된 ACL 파일이 하나 있습니다 . 여러 개의 ACL 파일을 만들고 `obj.conf` 파일에서 이를 참조할 수도 있습니다 .

Proxy Server 4 는 Proxy Server 3.x 에서 사용된 구문과 다른 ACL 파일 구문을 사용합니다 . 이 부록에서는 ACL 파일과 해당 구문을 설명합니다 . Proxy Server 와 해당 리소스의 액세스 제어에 대한 자세한 내용은 제 8 장 , 147 페이지의 " 서버 액세스 제어 " 를 참조하십시오 . 리소스 템플릿은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 에서 설명한 대로 Proxy Server 4 릴리스에서 지원됩니다 .

이 부록은 다음과 같은 절로 구성되어 있습니다 .

ACL 파일 및 ACL 파일 구문 정보

모든 ACL 파일은 특정 형식과 구문을 따라야 합니다 . ACL 파일은 하나 이상의 ACL 이 포함된 텍스트 파일입니다 . 모든 ACL 파일은 구문 버전 번호로 시작해야 합니다 . 예 :

```
version 3.0;
```

버전 줄은 하나 뿐이며 그 앞에 원하는 만큼의 주석을 삽입할 수 있습니다 . Proxy Server 는 구문 버전 3.0 을 사용합니다 . 주석은 줄 앞에 # 기호를 사용하여 파일에 포함할 수 있습니다 .

파일의 각 ACL 은 해당 유형을 정의하는 정의문으로 시작합니다 . ACL 유형은 다음 세 가지 중 하나입니다 .

- 경로 ACL 은 영향을 미치는 리소스에 대한 절대 경로를 지정합니다 .
- 리소스 ACL 은 `http://`, `https://`, `ftp://` 등과 같은 영향을 미치는 템플릿을 지정합니다 . 템플릿에 대한 더 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 를 참조하십시오 .
- 이름이 지정된 ACL은 `obj.conf` 파일의 리소스에서 참조되는 이름을 지정합니다. 서버에는 기본 이름이 지정된 리소스가 함께 제공되어 모든 사용자에게 읽기 액세스를 허용하며 LDAP 디렉토리의 사용자에게 쓰기 액세스를 허용합니다 . Proxy Server 사용자 인터페이스에서 이름이 지정된 ACL 을 만들 수 있다 하더라도 반드시 `obj.conf` 파일의 리소스에서 이름이 지정된 ACL 을 직접 참조해야 합니다 .

경로 ACL 및 리소스 ACL 은 와일드카드를 포함할 수 있습니다 . 와일드카드에 대한 자세한 내용은 제 16 장 , 343 페이지의 " 템플릿 및 리소스 관리 " 를 참조하십시오 .

유형 줄은 `acl` 로 시작하며 유형 정보는 인용 부호 안에 포함되고 , 그 뒤에 세미콜론을 넣습니다 . 예 :

```
acl "default";
acl "http://*.*";
```

모든 ACL 의 유형 정보는 서로 다른 ACL 파일이라 할지라도 고유한 이름이어야 합니다 . ACL 의 유형을 정의한 후 , ACL 과 함께 사용할 메소드를 정의하는 줄 (인증문) 과 액세스를 허용 또는 거부할 컴퓨터 또는 사용자를 정의하는 줄 (인증문) 을 하나 이상 추가합니다 . 다음에서는 이러한 줄의 구문에 대해 설명합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- 인증문
- 권한 부여문
- 기본 ACL 파일

인증문

ACL 은 선택적으로 ACL 을 처리할 때 서버가 반드시 사용해야 하는 인증 방법을 지정할 수 있습니다 . 세 가지 방법이 있습니다 .

- Basic(기본값)
- Digest

- SSL

Basic 및 Digest 방법의 경우 사용자가 리소스에 액세스하기 전에 아이디와 비밀 번호를 입력해야 합니다 .

SSL 방법의 경우 사용자에게 클라이언트 인증서가 있어야 합니다 . 인증을 받으려면 Proxy Server 에서 암호화가 사용되어야 하며 사용자의 인증서 발행자가 신뢰 CA 목록에 있어야 합니다 .

기본적으로 서버는 방법이 지정되지 않은 ACL 에 대해 Basic 방법을 사용합니다 . 서버의 인증 데이터베이스가 사용자가 송신한 Digest 인증을 지원해야 합니다 .

각 인증 줄은 반드시 서버가 인증할 속성 (사용자 , 그룹 또는 모두) 을 지정해야 합니다 . 다음 인증문은 ACL 유형 줄 뒤에 표시되며 데이터베이스 또는 디렉토리의 개별 사용자와 사용자가 일치하는 경우 Basic 인증을 지정합니다 .

```
authenticate (user) {
    method = "basic";
};
```

다음 예에서는 SSL 을 사용자 및 그룹용 인증 방법으로 사용합니다 .

```
authenticate (user, group) {
    method = "ssl";
};
```

다음 예에서는 아이디가 `sales` 로 시작하는 모든 사용자를 허용합니다 .

```
allow (all) user = "sales*";
```

마지막 줄을 `group=sales` 로 변경하면 그룹 속성이 인증되지 않으므로 ACL 이 실패하게 됩니다 .

권한 부여문

각 ACL 항목에는 하나 이상의 권한 부여문이 있습니다 . 권한 부여문은 서버 리소스에 대한 액세스를 허용 또는 거부할 사용자를 지정합니다 .

권한 부여문 작성

권한 부여문을 작성하는 경우 다음 구문을 사용합니다 .

```
allow|deny [absolute] (right[,right...]) attribute expression;
```

각 줄은 allow 또는 deny 로 시작합니다 . 보통 첫 번째 규칙에서는 모든 사용자의 액세스를 거부한 다음 이후의 규칙에서 사용자 , 그룹 또는 컴퓨터의 액세스를 구체적으로 허용하는 것이 좋습니다 . 이는 규칙의 계층 때문입니다 . 즉 , /my_files 라는 디렉토리에 대해 모든 사용자의 액세스를 허용한 후 하위 디렉토리인 /my_files/personal 은 일부 사용자에게만 액세스를 허용하는 경우 하위 디렉토리에 대한 액세스 제어가 동작하지 않습니다 . 이는 /my_files 디렉토리에 액세스가 허용된 모든 사용자는 /my_files/personal 디렉토리에 대해서도 액세스가 허용되기 때문입니다 . 이러한 경우를 예방하려면 모든 사용자의 액세스를 거부한 후 일부 필요한 사용자에게 액세스를 허용하는 하위 디렉토리용 규칙을 만듭니다 .

그러나 모든 사용자의 액세스를 거부하도록 기본 ACL 을 설정하는 경우 다른 ACL 규칙에는 "deny all" 규칙이 필요하지 않은 경우가 있습니다 .

다음 줄은 모든 사용자의 액세스를 거부합니다 .

```
deny (all) user = "anyone";
```

권한 부여문의 계층

ACL 에는 리소스에 따른 계층이 있습니다 . 특정 리소스에 대한 요청을 받으면 서버는 해당 리소스에 적용할 ACL 목록을 구축합니다 . 서버는 우선 obj.conf 파일의 check-acl 문에 있는 목록에서 이름이 지정된 ACL 을 추가합니다 . 그런 후 서버는 일치하는 경로 ACL 및 리소스 ACL 을 추가합니다 . 이 목록은 같은 순서로 처리됩니다 . "absolute" ACL 문이 있지 않는 한 모든 문은 순서대로 평가됩니다 . "absolute allow" 또는 "absolute deny" 문이 "true" 인 경우 서버는 처리를 중단하고 이 결과를 받아들입니다 .

일치되는 ACL 이 하나 이상인 경우 서버는 일치하는 마지막 문을 사용합니다 . 그러나 absolute 문을 사용하는 경우 서버는 다른 일치에 대한 조회를 중단하고 absolute 문이 포함된 ACL 을 사용합니다 . 동일한 리소스에 대해 absolute 문이 둘인 경우 서버는 파일의 첫 번째 문을 사용하고 일치하는 다른 리소스에 대한 조회를 중단합니다 .

```
version 3.0;
acl "default";
authenticate (user,group) {
  prompt="Sun Java System Web Proxy Server";
};
allow (read,execute,list,info) user = "anyone";
allow (write,delete) user = "all";

acl "http://*.*";
deny (all) user = "anyone";
allow (all) user = "joe";
```

속성 표현식

속성 표현식은 아이디 , 그룹 이름 , 호스트 이름 또는 IP 주소를 기준으로 허용 또는 거부할 사용자를 정의합니다 . 서로 다른 사용자 또는 컴퓨터에 액세스를 허용하는 방법에 대한 예는 다음과 같습니다 .

- `user = "anyone";`
- `user = "smith*"`
- `group = "sales"`
- `dns = "*.mycorp.com"`
- `dns = "*.mycorp.com,*.company.com"`
- `ip = "198.*"`
- `ciphers = "rc4"`
- `ssl = "on"`

또한 `timeofday` 속성을 사용하여 하루 중 시간 (서버의 로컬 시간 기준) 에 따라 서버에 대한 액세스를 제한할 수 있습니다 . 예를 들어 `timeofday` 속성을 사용하여 특정 시간 동안 특정 사용자의 액세스를 제한할 수 있습니다 .

시간을 지정하려면 24 시간 형식을 사용합니다 . 예를 들어 0400 은 오전 4:00, 2230 은 오후 10:30 을 지정합니다 . `guests` 라는 사용자 그룹의 액세스를 오전 08:00 에서 오후 4:59 까지 제한하려면 다음 예제와 같이 합니다 .

```
allow (read)
    (group="guests") and
    (timeofday<0800 or timeofday=1700);
```

또한 주중 요일에 따라 액세스를 제한할 수 있습니다 . 세 자리 약자 (Sun, Mon, Tue, Wed, Thu, Fri, Sat) 를 사용하여 요일을 지정합니다 .

다음 문은 premium 그룹의 사용자에게 항상 액세스를 허용합니다 . discount 그룹의 사용자는 주말의 모든 시간과 주중 오전 08:00 부터 오후 4:59 까지를 제외한 모든 시간에 액세스할 수 있습니다 .

```
allow (read) (group="discount" and dayofweek="Sat,Sun") or
(group="discount" and (dayofweek="mon,tue,wed,thu,fri" and
(timeofday<0800 or timeofday=1700)))
or
(group="premium");
```

표현식용 연산자

속성 표현식에 다양한 연산자를 사용할 수 있습니다 . 괄호는 연산자의 순서를 변경할 때 사용합니다 . 다음 연산자는 user, group, dns 및 ip 와 사용할 수 있습니다 .

- and
- or
- not
- =(등호)
- !=(부등호)

다음 연산자는 timeofday 및 dayofweek 와 사용할 수 있습니다 .

- greater than
- < less than
- = greater than or equal to
- <= less than or equal to

기본 ACL 파일

설치 후 *server_root*/httpacl/generated.proxy-*serverid*.acl 파일에서 서버에 대한 기본 설정이 제공됩니다 . 사용자 인터페이스에서 설정이 만들어 질 때까지 서버는 작업 파일 genwork.proxy-*serverid*.acl 을 사용합니다 . ACL 을 파일을 편집할 때 genwork 파일을 변경할 수 있으며 , 그런 후 Proxy Server 를 사용하여 변경 사항을 저장 및 적용할 수 있습니다 .

일반 구문 항목

입력 문자열에는 다음 문자를 포함할 수 있습니다 .

- a 에서 z 까지의 문자
- 0 에서 9 까지의 숫자
- 마침표 및 밑줄

다른 문자인 경우 인용 부호 (") 안에 넣어야 합니다 .

단일문은 한 줄에 위치해야 하며 세미콜론으로 끝을 표시합니다 . 복수문은 대괄호 ([]) 안에 넣습니다 . 항목 목록은 반드시 쉼표로 분리해야 하며 인용 부호 (") 안에 넣어야 합니다 .

obj.conf 내의 ACL 파일 참조

이름이 지정된 ACL 이나 별도의 ACL 파일은 obj.conf 파일에서 참조할 수 있습니다 . 이 작업은 check-acl 함수를 사용하는 PathCheck 지시문에서 수행합니다 . 이 줄의 구문은 다음과 같습니다 .

```
PathCheck fn="check-acl" acl="aclname"
```

where *aclname* 은 ACL 파일에 표시되는 ACL 의 고유한 이름입니다 .

예를 들어 ACL named testacl 을 사용하여 디렉토리에 대한 액세스를 제한하려면 다음 줄을 obj.conf 파일에 추가합니다 .

```
<Object ppath="https://"
PathCheck fn="check-acl" acl="testacl"
</Object
```

앞의 예에서 첫 번째 줄은 액세스를 제한하려는 서버 리소스를 표시하는 개체입니다 . 두 번째 줄은 PathCheck 지시문으로 check-acl 함수를 사용하여 이름이 지정된 ACL(testacl) 을 지시문이 나타나는 개체에 바인드합니다 . testacl ACL 은 server.xml 에서 참조하는 모든 ACL 파일에 존재할 수 있습니다 .

부록 B

서버 성능 조정

Proxy Server 환경에서 성능에 영향을 미치는 요소는 프록시 클라이언트 , Proxy Server, 원본 서버 , 네트워크를 포함하여 다양합니다 . 이 부록에서는 Proxy Server 성능이 향상되도록 조정하는 방법에 대해 설명합니다 .

이 부록은 다음과 같은 절로 구성되어 있습니다 .

주의 이 부록은 고급 관리자만을 대상으로 합니다 . 서버를 조정할 때는 세심한 주의를 기울여야 하며 모든 변경 전에 항상 구성 파일을 백업해야 합니다 .

일반 성능 고려 사항

이 절에서는 Proxy Server 성능을 분석할 때 고려해야 하는 일반적인 영역에 대해 설명합니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

액세스 로깅

액세스 로깅을 사용하지 않도록 설정하면 Proxy Server 의 성능을 높일 수 있습니다 . 그러나 이 방법을 사용하는 경우 Proxy Server 를 액세스한 사용자 및 이들이 요청한 페이지를 알 수 없게 된다는 단점이 있습니다 .

`obj.conf` 파일에서 다음 지시문을 주석으로 처리하여 Proxy Server 액세스 로깅을 사용하지 않도록 할 수 있습니다 .

```
Init fn="flex-init" access="$accesslog" format.access="%Ses->client.ip%
- %Req->vars.auth-user% [%SYSDATE%] \"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
...
AddLog fn="flex-log" name="access"
```

ACL 캐시 조정

기본적으로 Proxy Server 는 사용자 및 그룹 인증 결과를 ACL 사용자 캐시에 캐시합니다 . `magnus.conf` 파일의 `ACLCacheLifetime` 지시문을 사용하여 ACL 사용자 캐시의 유효 시간을 조정할 수 있습니다 . 캐시에 있는 항목이 참조될 때마다 지속 시간이 계산되고 `ACLCacheLifetime` 과 비교됩니다 . 항목의 지속 시간이 `ACLCacheLifetime` 과 크거나 같으면 해당 항목은 사용되지 않습니다 .

ACLCacheLifetime 의 기본값은 120 초입니다 . 이 경우 Proxy Server 가 LDAP 서버와 최대 2 분 동안 동기화되지 않을 수 있습니다 . 값을 0 으로 설정하면 캐시를 사용하지 않으며 Proxy Server 는 사용자 인증 때마다 LDAP 서버를 쿼리합니다 . 이는 액세스 제어를 구현하는 경우 Proxy Server 의 성능을 떨어뜨리는 요인이 됩니다 . ACLCacheLifetime 값을 크게 설정한 경우 LDAP 항목을 변경할 때마다 Proxy Server 를 재시작하여 Proxy Server 가 LDAP 서버를 쿼리하도록 해야할 수도 있습니다 . LDAP 디렉토리가 자주 변경되지 않는 경우에만 큰 값을 설정하십시오 .

ACLUserCacheSize 는 캐시에 유지할 항목의 최대 수를 구성하는 magnus.conf 매개변수입니다 . 기본값은 200 입니다 . 새 항목은 목록의 앞에 추가되며 목록의 끝에 있는 항목은 캐시가 최대 크기에 도달하면 새로운 항목이 추가될 수 있도록 순환 처리됩니다 .

또한 ACLGroupCacheSize 매개 변수를 사용하여 각 사용자 항목마다 캐시될 수 있는 그룹 구성원의 최대 수를 설정할 수 있습니다 . 기본값은 4 입니다 . 유감스럽게도 그룹에 있는 사용자가 구성원이 아닌 경우 캐시되지 않으며 , 요청마다 여러 LDAP 디렉토리 액세스가 발생하게 됩니다 .

버퍼 크기

서버의 소켓에서 송신 버퍼의 크기 (SndBufSize) 와 수신 버퍼의 크기 (RcvBufSize) 를 지정할 수 있습니다 . 이들 매개 변수는 magnus.conf 파일 내에서 구성되며 권장 값은 여러 UNIX, Linux 운영 체제에 따라 각기 다릅니다 . 운영 체제의 설명서를 참조하여 이 매개 변수를 적절히 설정하십시오 .

연결 시간 초과

magnus.conf 파일에 있는 AcceptTimeout 지시문을 사용하여 서버가 연결을 종료하기 전까지 클라이언트의 데이터 도착을 대기하는 시간을 초 단위로 지정할 수 있습니다 . 데이터가 지정된 시간까지 도착하지 않으면 연결이 종료됩니다 . 기본값은 30 초로 설정됩니다 . 대부분의 경우 이 설정을 변경할 필요가 없습니다 . 이 값을 기본값보다 낮게 설정하면 그만큼 스레드를 반환할 수 있지만 연결 속도가 느린 사용자의 연결이 끊어질 수 있습니다 .

오류 로그 수준

server.xml 파일의 LOG 태그에서 loglevel 속성 값을 늘리면 서버는 더 많은 오류 로그 정보를 생성하고 저장합니다 . 그러나 이 방법은 이 파일에 항목을 쓸 때 문제가 발생합니다 . 문제를 디버깅하는 경우에만 로깅을 늘리고 문제 해결 모드를 사용하지 않는 경우에는 로깅을 최소화하십시오 .

보안 요구 사항

SSL 을 사용하면 Proxy Server 의 비밀 보호 및 보안을 강화할 수 있지만 패킷의 암호화 및 복호화로 인한 오버헤드가 성능에 영향을 줄 수 있습니다 . 하드웨어 가속 카드를 사용하여 암호화 및 복호화 처리의 로드를 줄이는 방법을 고려해 볼 수 있습니다 .

Solaris 파일 시스템 캐시

Proxy Server 캐시는 RAM(Random Access Memory) 에 저장되지 않습니다 . 문서를 캐시에서 추출할 때 파일 시스템에서 파일에 대한 액세스가 이루어집니다 . Solaris 파일 시스템 캐시를 사용하여 Proxy Server 캐시를 메모리로 미리 로드하는 방법을 고려할 수 있습니다 . 이렇게 하면 캐시된 파일에 대한 참조는 파일 시스템이 아니라 메모리에서 추출됩니다 .

제한 시간 값

제한 시간은 서버 성능에 큰 영향을 미칩니다 . Proxy Server 에 대한 최적 제한 시간 값을 설정하면 네트워크 리소스를 절약할 수 있습니다 .

두 개의 인스턴스별 SAF(Server Application Function) 와 하나의 전역 매개 변수를 사용해 Proxy Server 내에서 제한 시간 값을 구성할 수 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

- init-proxy SAF(obj.conf)
- http-client-config SAF(obj.conf)
- KeepAliveTimeout(magnus.conf)

init-proxy SAF(obj.conf)

`init-proxy` 함수는 Proxy Server 의 내부 설정을 초기화합니다 . 이 함수는 Proxy Server 초기화 과정에서 호출되지만 값이 제대로 초기화되는지 확인하기 위해 `obj.conf` 파일에도 지정되어야 합니다 .

이 함수의 구문은 다음과 같습니다 .

```
Init fn=init-proxy
  timeout=seconds
  timeout-2=seconds
```

앞의 예에서 다음 매개 변수는 `init-proxy` SAF 에 대한 Proxy Server 제한 시간 설정에 직접 적용할 수 있습니다 .

- `timeout`(프록시 제한 시간) - 프록시 제한 시간 매개 변수는 서버가 유휴 상태의 연결을 중단하기까지 대기하는 시간을 지정합니다 . 프록시 제한 시간 값을 높게 설정하면 다운 상태일 수 있는 클라이언트에 귀중한 프록시 스레드가 오랜 시간 동안 묶여 있게 됩니다 . 제한 시간 값을 낮게 설정하면 데이터베이스 쿼리 게이트웨이와 같은 결과가 나오기까지 오랜 시간이 걸리는 CGI 스크립트가 중단됩니다 .

  서버에 대한 최적의 프록시 제한 시간을 결정하려면 다음 사항을 고려하십시오 .

  - Proxy Server 가 많은 수의 데이터베이스 쿼리나 CGI 스크립트를 처리하는가 ?
  - 프로세스에 항상 여유가 있을 만큼 Proxy Server 가 처리하는 요청의 수가 적은가 ?

  두 가지 중 일치하는 조건이 있는 경우 프록시 제한 시간 값을 높게 설정하는 것이 좋습니다 . 최대 프록시 제한 시간 권장 값은 1 시간입니다 . 기본값은 300 초 (5 분) 입니다 .

  Server Manager 의 Preferences 탭에 있는 Configure System Preferences 페이지에 액세스하여 프록시 시간 제한 값을 확인하거나 변경할 수 있습니다 . 이 매개 변수는 Proxy Timeout 으로 표시됩니다 .

- `timeout-2`(중단 후 제한 시간) - 중단 후 제한 시간 값은 클라이언트에서 트랜잭션을 중단한 이후 Proxy Server 가 캐시 파일 쓰기를 지속하는 시간을 지정합니다 . 즉 , Proxy Server 가 문서 캐시를 거의 완료한 상황에서 클라이언트가 연결을 중단한 경우 서버는 중단 후 제한 시간 값에 도달할 때까지 해당 문서의 캐시를 지속할 수 있습니다 .

  중단 후 제한 시간의 최대 권장 값은 5 분입니다 . 기본값은 15 초입니다 .

http-client-config SAF(obj.conf)

`http-client-config` 함수는 Proxy Server 의 HTTP 클라이언트를 구성합니다 .

이 함수의 구문은 다음과 같습니다 .

```
Init fn=http-client-config
  keep-alive=(true|false)
  keep-alive-timeout=seconds
  always-use-keep-alive=(true|false)
  protocol=HTTP Protocol
  proxy-agent="Proxy-agent HTTP request header"
```

설정에 대한 정의는 다음과 같습니다 .

- `keep-alive` - (선택 사항) HTTP 클라이언트가 지속 연결을 사용하도록 시도할 것인지 여부를 나타내는 부울 값입니다 . 기본값은 true 입니다 .
- `keep-alive-timeout` - (선택 사항) 지속 연결을 개방 상태로 유지할 초 단위의 최대 수입니다 . 기본값은 29 입니다 .
- `always-use-keep-alive` - (선택 사항) HTTP 클라이언트가 모든 유형의 요청에 대해 기존의 지속 연결을 재사용할 수 있는지 여부를 나타내는 부울 값입니다 . 기본값은 false 이며 , 이 경우 GET 이외의 요청 또는 본문을 포함하는 요청에 대해 지속 연결을 재사용할 수 없습니다 .
- `protocol` - (선택 사항) HTTP 프로토콜 버전 문자열입니다 . 기본적으로 HTTP 클라이언트는 HTTP 요청의 내용에 따라 HTTP/1.0 또는 HTTP/1.1 을 사용합니다 . 일반적으로 특정 프로토콜 상호 운용성 문제가 발생하는 경우에만 프로토콜 매개 변수를 사용합니다 .
- `proxy-agent` - (선택 사항) Proxy-agent HTTP 요청 헤더의 값입니다 . 기본값은 Proxy Server 제품 이름과 버전을 포함한 문자열입니다 .

KeepAliveTimeout(magnus.conf)

이 매개 변수는 서버가 클라이언트와 Proxy Server 사이의 HTTP 연결 유지 연결 또는 지속 연결을 개방 상태로 유지할 최대 시간 (초 단위) 을 결정합니다 . 기본값은 30 초입니다 . 유휴 상태가 30 초 이상이면 해당 연결은 시간 초과로 처리됩니다 . 최대값은 300 초 (5 분) 입니다 .

참고 magnus.conf 파일의 제한 시간 설정은 클라이언트와 Proxy Server 사이의 연결에 적용됩니다 . obj.conf 파일에서 http-client-config SAF 의 제한 시간 설정은 Proxy Server 와 원본 서버 사이의 연결에 적용됩니다 .

최신 여부 확인

Proxy Server 는 문서를 원본 서버가 아닌 로컬 캐시에서 서비스함으로써 성능을 향상시킵니다 . 이러한 방법론의 한가지 단점은 최신 상태가 아닌 문서를 제공할 가능성이 있다는 점입니다 .

Proxy Server 는 문서의 최신 여부를 확인한 다음 최신 문서가 아니라고 판단한 경우 캐시된 문서를 새로 고칠 수 있습니다 . 문서의 최신 여부 확인은 자주 수행할 경우 Proxy Server 의 전체 성능을 떨어뜨릴 수 있으므로 필요한 경우에만 수행해야 합니다 .

최신 여부 확인은 Caching 탭의 Set Cache Specifics 페이지에서 구성합니다 . 기본적으로 2 시간마다 새 문서를 확인합니다 . 이 정보는 ObjectType 지시문에서 max-uncheck 매개 변수로 구성됩니다 .

문서를 최신 상태로 유지하면서 서버의 성능을 높이려면 아래 설명된 최종 수정 요인과 관련한 적절한 문서 수명 주기를 결정하여 최신 여부 확인을 사용자 정의합니다 .

최종 수정 요인

최종 수정 요인은 문서를 최신 상태로 유지하는 과정을 미세 조정하는 데 사용됩니다 . 이 요인은 문서의 이전 변경 기록을 기반으로 앞으로 변경될 가능성을 판단할 수 있도록 해줍니다 .

최종 수정 요인은 .02 에서 1.0 사이의 분수입니다 . 이 값에 문서의 실제 최종 수정 시간과 최신 여부 확인이 마지막으로 수행된 시간 사이의 간격을 곱합니다 . 결과 수와 최신 여부 확인을 마지막으로 수행한 이후 지난 시간을 비교합니다 . 결과 수가 시간 간격보다 작으면 문서가 만료되지 않은 것입니다 . 결과 수가 시간 간격보다 크면 문서가 만료된 것으로 , 원본 서버에서 새 버전을 가져옵니다 .

최종 수정 요인은 최근에 변경된 문서를 오래된 문서보다 더 자주 확인할 수 있도록 해줍니다 .

최종 수정 요인은 0.1 에서 0.2 사이에서 설정해야 합니다 .

DNS 설정

DNS 는 표준 IP 주소를 호스트 이름과 연결하는 데 사용되는 시스템입니다 . 이 시스템을 신중하게 구성하지 않으면 귀중한 Proxy Server 리소스가 묶여 있을 수 있습니다 . 성능을 최적화하려면 다음 사항을 고려하십시오 .

- DNS 캐시 사용

  DNS 캐시는 Server Manager 의 Preferences 탭에서 Configure DNS Cache 링크를 선택하면 사용할 수 있습니다 . DNS 캐시를 위한 Enabled 라디오 버튼을 선택합니다 .

- 클라이언트 DNS 이름을 기록하지 않고 클라이언트 IP 주소만 기록

  클라이언트 DNS 이름 로깅은 Server Manager 의 Server Status 탭에서 Set Access Log Preferences 링크를 선택하여 사용하지 않도록 설정할 수 있습니다 . IP Addresses 라디오 버튼을 선택해 클라이언트 호스트 이름이 아닌 IP 주소를 기록하도록 합니다 .

- 역방향 DNS 사용하지 않음

  역방향 DNS 는 IP 주소를 호스트 이름으로 변환합니다 . 역방향 DNS 는 Server Manager 의 Preferences 탭에서 Configure System Preferences 링크를 선택하여 사용하지 않도록 설정할 수 있습니다 . No 라디오 버튼을 선택하여 역방향 DNS 를 사용하지 않도록 설정합니다 .

- 클라이언트 호스트 이름에 기반한 액세스 제어 지양

  액세스 제어 문에서 가능한 경우 클라이언트의 호스트 이름 대신 IP 주소를 사용합니다 .

스레드의 수

`magnus.conf` 파일의 `RqThrottle` 매개 변수는 Proxy Server 가 처리할 수 있는 동시 트랜잭션의 최대 수를 지정합니다 . 기본값은 128 입니다 . 이 값을 변경하여 서버의 로드를 조절하고 수행되는 트랜잭션의 지연을 최소화할 수 있습니다 .

서버는 동시 요청의 수를 계산하기 위해 활성 요청의 수를 계산하고 새 요청이 수신되면 이 수에 1 을 더하고 요청을 처리하면 1 을 뺍니다 . 새 요청이 수신되면 서버는 처리하고 있는 요청의 수가 이미 최대값인지 확인합니다 . 최대값에 도달한 경우 진행 중인 요청의 수가 최대값 아래로 떨어질 때까지 새 요청은 보류됩니다 .

perfdump 에서 생성하는 데이터의 SessionCreationInfo 부분 또는 proxystats.xml 데이터를 확인하여 동시 요청의 수를 모니터링할 수 있습니다 . 이 정보를 통해 전체 스레드 수 (제한) 에 비교하여 동시 (최대) 요청의 최대 수를 결정할 수 있습니다 . 다음 정보는 perfdump 출력의 일부입니다 .

```
SessionCreationInfo:
------------------------
Active Sessions          1
Keep-Alive Sessions      0
Total Sessions Created   48/128
```

Active Sessions 는 현재 요청을 서비스하고 있는 세션 (요청 처리 스레드) 의 수를 표시합니다 . Keep-Alive Sessions 는 Active Sessions 와 비슷하지만 클라이언트가 연결 유지 연결을 요청하고 있는 경우에만 국한됩니다 . Total Sessions Created 는 생성된 세션의 수와 허용된 최대 세션 수를 표시합니다 . 이 두 수는 RqThrottle 의 최소 및 최대값입니다 .

참고 RqThrottleMin 은 서버 시작 시 실행되는 스레드의 최소 수입니다 . 기본값은 48 입니다 . 이 매개 변수는 magnus.conf 파일에서도 설정할 수 있지만 기본적으로 표시되지는 않습니다 .

구성된 스레드의 수가 최대 수에 도달했다고 해서 반드시 부적절한 것은 아니며 RqThrottle 값을 무조건 높일 필요도 없습니다 . 한계에 도달했다는 것은 서버가 최대 로드에서 한계 수 만큼의 스레드를 요구했다는 의미이며 , 서버가 요청을 적시에 처리했다면 적절하게 조정된 것입니다 . 그러나 이 시점에서 연결이 연결 큐에 쌓여 오버플로가 발생할 가능성이 있습니다 . 정기적으로 perfdump 출력을 확인하여 생성된 총 세션이 RqThrottle 최대값에 근접하는 경우가 자주 있다면 스레드 한계를 높이는 것을 고려하십시오 .

RqThrottle 값은 로드에 따라 100 에서 500 사이가 적당합니다 .

인바운드 연결 풀

magnus.conf 의 KeepAlive* 및 다음과 같은 관련 설정을 사용하여 인바운드 연결 풀을 조정할 수 있습니다 .

- MaxKeepAliveConnections
- KeepAliveThreads

- KeepAliveTimeout
- KeepAliveQueryMaxSleepTime
- KeepAliveQueryMeanTime
- ConnQueueSize
- RqThrottle
- acceptorthreads

참고 이러한 매개 변수에 대한 자세한 내용은 다음 사이트에서 Sun ONE Web Server 6.1 Performance Tuning, Sizing, and Scaling Guide 의 제 2 장을 참조하십시오 .

http://docs.sun.com/source/817-6249/index.html

아웃바운드 연결 풀 설정은 이 Proxy Server 릴리스에서 구성할 수 없습니다 .

FTP 목록 너비

사용자의 요구 사항에 맞도록 FTP 목록의 너비를 수정할 수 있습니다 . 목록의 너비를 넓히면 더 긴 파일 이름을 사용할 수 있으므로 파일 이름이 잘리는 현상을 줄일 수 있습니다 . 기본 너비는 80 자입니다 .

Server Manager 의 Preferences 탭에서 Tune Proxy 링크를 선택하여 FTP 목록 너비를 수정할 수 있습니다 .

캐시 아키텍처

캐시 아키텍처를 적절히 설정하여 서버의 성능을 높일 수 있습니다 . 캐시 아키텍처를 설정할 때 다음 사항을 유의하십시오 .

- 로드 분산
- 여러 프록시 캐시 파티션 사용
- 여러 디스크 드라이브 사용

- 여러 디스크 컨트롤러 사용

적절한 캐시 설정은 Proxy Server 의 성능에 있어 중요합니다 . 프록시 캐시를 설정할 때 유의해야 할 가장 중요한 규칙은 로드를 분산하는 것입니다 . 캐시는 파티션 당 약 1GB 로 설정해야 하며 여러 개의 디스크와 디스크 컨트롤러에 걸쳐 분산되어야 합니다 . 이런 유형의 배열을 사용하면 하나의 대형 캐시를 사용하는 것보다 파일을 더 빠르게 만들고 검색할 수 있습니다 .

캐시 일괄 업데이트

캐시 일괄 업데이트 기능을 사용하여 특정 웹 사이트의 파일을 미리 로드하거나 이미 캐시에 있는 문서의 최신 여부를 확인할 수 있습니다 . 캐시 일괄 업데이트는 일반적으로 Proxy Server 의 로드가 최저일 때 실행됩니다 . Cache Batch Updates 형식에서 URL 일괄 처리를 작성 , 편집 , 제거할 수 있으며 일괄 업데이트의 사용 여부를 결정할 수 있습니다 .

파일을 일괄 업데이트하도록 지정하면 필요에 따라 캐시하는 것과 달리 활발하게 내용을 캐시할 수 있습니다 . Proxy Server 는 현재 캐시에 있는 여러 파일의 최신 여부를 확인하거나 특정 웹 사이트에 있는 여러 개의 파일을 미리 로드할 수 있도록 합니다 .

서버 및 프록시 네트워크가 있는 대형 사이트에서 관리자는 지정한 웹 영역을 미리 로드하기 위해 일괄 업데이트를 사용할 수 있습니다 . 일괄 처리 프로세스는 문서 내의 링크 전반에 순환적 하향 기능을 수행하고 내용을 로컬에 캐시합니다 . 이 기능은 원격 서버에 부담이 될 수 있으므로 주의하여 사용해야 합니다 . 이 프로세스가 순환 기능을 무한정 수행하지 않도록 하는 조치가 취해지며 `bu.conf` 구성 파일의 매개 변수로 이 프로세스를 일부 제어할 수 있습니다 .

Proxy Server 액세스 로그를 사용하여 가장 활동적인 사이트를 확인한 다음 이 사이트에 일괄 업데이트를 수행하여 성능을 높일 수 있습니다 .

가비지 수집

가비지 수집은 Proxy Server 캐시를 검토하고 오래된 (시한이 지난) 파일을 제거하는 프로세스입니다 . 가비지 수집은 리소스를 많이 사용하는 프로세스이므로 일부 가비지 수집 설정을 조정하여 성능을 향상시킬 수 있습니다 .

다음 매개 변수를 통해 가비지 수집 프로세스를 미세 조정할 수 있습니다 . Server Manager 의 Caching 탭에서 Tune GC 를 선택하여 이동한 Tune Garbage Collection 양식에서 이러한 매개 변수를 확인하거나 수정할 수 있습니다 .

이 절에서는 다음 항목에 대해 설명합니다 .

gc hi margin percent 변수

gc hi margin percent 변수는 가비지 수집이 시작되는 최대 캐시 크기의 비율을 조정합니다 .

이 값은 gc lo margin percent 변수 값보다 높아야 합니다 .

gc hi margin percent 의 유효한 범위는 10 ~ 100% 입니다 . 기본값은 80% 입니다 (캐시가 80% 까지 차면 가비지 수집 시작).

gc lo margin percent 변수

gc lo margin percent 변수는 가비지 수집기가 목표로 하는 최대 캐시 크기의 비율을 조정합니다 .

이 값은 gc hi margin percent 값보다 낮아야 합니다 .

gc lo margin percent 의 유효한 범위는 5 ~ 100% 입니다 . 기본값은 70% 입니다 (가비지 수집 이후 70% 까지 찬 캐시를 대상으로 함).

gc extra margin percent 변수

파티션 크기가 허용된 최대 크기 (gc hi margin percent) 에 근접한 것 외의 다른 이유로 가비지 수집이 시작된 경우 가비지 수집기는 gc extra margin percent 변수에 의해 설정된 비율을 사용하여 제거할 캐시 조각을 결정합니다 .

gc extra margin percent 의 유효한 범위는 0 ~ 100% 입니다 . 기본값은 30% 입니다 (기존 캐시 파일의 30% 를 제거).

gc leave fs full percent 변수

gc leave fs full percent 값은 캐시 파티션 크기의 비율을 결정합니다 . 캐시 파티션 크기 비율이 이 값보다 작을 경우 가비지 수집이 실행되지 않습니다 . 이 값은 다른 응용 프로그램이 디스크 공간을 독점하고 있는 경우 가비지 수집기가 캐시에서 모든 파일을 제거하지 않도록 합니다 .

gc leave fs full percent 의 유효한 범위는 0(전체 제거 허용) ~ 100%(제거하지 않음) 입니다 . 기본값은 60% 입니다 (캐시 크기를 현재의 60% 로 줄이도록 허용).

Solaris 성능 조정

Solaris 커널의 다양한 매개 변수를 사용하여 Proxy Server 성능을 미세 조정할 수 있습니다 . 이러한 매개 변수는 다음 표의 목록을 참조하십시오 .

표 B-1 Solaris 성능 조정 매개 변수

매개 변수	범위	기본값	조정값	설명
rlim_fd_max	/etc/system	1024	8192	열린 파일 설명자 (descriptor) 제한을 처리합니다 . 예상 로드 (연결된 소켓 , 파일 , 파이프 등) 를 계산해야 합니다 .
rlim_fd_cur	/etc/system	64	8192	
sq_max_size	/etc/system	2	0	스트림 드라이버 큐 크기를 제어합니다 . 0 으로 설정하면 크기가 무한대이므로 버퍼 공간 부족으로 인한 성능 저하가 없습니다 . 클라이언트에서도 설정하십시오 .
tcp_close_wait_interval	ndd/dev/tcp	240000	60000	클라이언트에서도 설정하십시오 .
tcp_conn_req_max_q	ndd/dev/tcp	128	1024	
tcp_conn_req_max_q0	ndd/dev/tcp	1024	4096	
tcp_ip_abort_interval	ndd/dev/tcp	480000	60000	
tcp_keepalive_interval	ndd/dev/tcp	7200000	900000	트래픽이 많은 웹 사이트인 경우 이 값을 낮추십시오 .
tcp_rexmit_interval_initial	ndd/dev/tcp	3000	3000	재전송이 30 ~ 40% 이상이면 이 값을 늘리십시오 .

표 B-1 Solaris 성능 조정 매개 변수

매개 변수	범위	기본값	조정값	설명
tcp_rexmit_interval_max	ndd/dev/tcp	240000	10000	
tcp_rexmit_interval_min	ndd/dev/tcp	200	3000	
tcp_smallest_anon_port	ndd/dev/tcp	32768	1024	클라이언트에서도 설정하십시오 .
tcp_slow_start_initial	ndd/dev/tcp	1	2	소량의 데이터 전송 속도가 약간 더 빠릅니다 .
tcp_xmit_hiwat	ndd/dev/tcp	8129	32768	송신 버퍼를 늘리는 데 사용합니다 .
tcp_recv_hiwat	ndd/dev/tcp	8129	32768	수신 버퍼를 늘리는 데 사용합니다 .

이러한 매개 변수에 대한 자세한 내용은 다음 사이트에서 Sun ONE Web Server 6.1 Performance Tuning, Sizing, and Scaling Guide 의 제 5 장을 참조하십시오 .

http://docs.sun.com/source/817-6249/index.html

색인

기호

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

X

ㄱ

ㄴ

ㄷ

ㄹ

ㅁ

ㅂ

ㅅ

ㅇ

ㅈ

ㅊ

ㅋ

E

ㅍ

ㅎ