

WebSphere Application Server



Caching Proxy 관리 안내서

버전 6.1

WebSphere Application Server



Caching Proxy 관리 안내서

버전 6.1

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 317 페이지의 『주의사항』의 일반 정보를 반드시 읽으십시오.

초판(2006년 5월)

이 책은 다음에 적용됩니다.

WebSphere Application Server, 버전 6.1

새 개정판에 달리 명시되지 않는 한 후속 릴리스 및 수정판에 적용됩니다.

IBM 담당자 또는 해당 지역의 IBM 지사를 통해 책을 주문할 수 있습니다.

© Copyright International Business Machines Corporation 2006. All rights reserved.

목차

그림	xi
이 책에 대한 정보.	xiii
이 책의 사용자.	xiii
책에 사용된 규약 및 용어	xiii
내게 필요한 옵션	xiv
고객 의견을 보내는 방법.	xiv
관련 정보.	xiv

제 1 부 Caching Proxy 시작하기 1

제 1 장 개요	3
기본 Caching Proxy 구성.	3
역방향 프록시(기본값)	3
정방향 프록시	3
새로운 기능 지원.	5
제 2 장 구성 및 관리 양식 사용법.	7
브라우저 요구사항	7
구성 및 관리 양식 액세스.	8
관리자 암호 설정	10
제 3 장 구성 마법사 사용	11
제 4 장 ibmproxy.conf 파일 직접 편집	13
제 5 장 Caching Proxy 시작 및 정지.	15
Linux 및 UNIX 시스템에서의 자동 시동 및 시스템 종료.	15
Linux 및 UNIX 시스템에서의 수동 시동	16
AIX의 경우:	16
HP-UX의 경우:	17
Linux의 경우.	17
Solaris의 경우	17
Windows 서비스로 시동	18
Windows 응용프로그램으로 시동	19
시작 메뉴 사용	19
명령 프롬프트 사용.	19
복수 프록시 서버 시작.	20
Linux 및 UNIX 시스템 수동 시스템 종료.	20
시스템 종료 명령의 한계	21
Windows 시스템에서의 수동 시스템 종료	22
구성 변경 후 재시작	23

제 2 부 Caching Proxy 프로세스 구성 및 조정 25

제 6 장 서버 정의.	27
연관된 지시문	28
구성 및 관리 양식	28
제 7 장 프로세스 소유권 설정	29
연관된 지시문	29
구성 및 관리 양식	30
제 8 장 연결 관리.	31
연관된 지시문	32
구성 및 관리 양식	33
제 9 장 프록시 서버 프로세서 조정	35
성능 관련 지시문 설정.	35
기타 응용프로그램 검토	35
페이징 영역 확인	36
파일 시스템 조정	36
TCP/IP 구성 조정	36
상위 로드 환경을 위한 TCP 시간 대기 간격 조정 (HP-UX, Linux, Solaris, Windows).	36
Linux 커널 조정	37
AIX 스레드 조정 변수 조정.	38

제 3 부 Caching Proxy 작동 구성 39

제 10 장 요청 처리 관리.	41
HTTP/FTP 메소드 사용 가능	41
연관된 지시문	42
구성 및 관리 양식	43
WebDAV 메소드, MS Exchange 메소드, 사용자 정의 메소드 사용 가능	43
연관된 지시문	44
맵핑 규칙 정의	44
맵핑 규칙	45
대리 서버 구성	46
연관된 지시문	46
구성 및 관리 양식	46
결합 재작성 사용 가능(선택적)	47
JunctionPrefix 옵션 없이 결합 정의	47

JunctionPrefix 옵션으로 결합 정의(권장하는 메소드)	48
연관된 지시문	49
구성 및 관리 양식	49
JunctionRewrite에 대한 대안으로서의 UseCookie	49
JunctionRewrite 기능성의 확장을 위한 샘플 transmogriifier 플러그인	50
제 11 장 로컬 콘텐츠 전달 관리	53
문서 루트 디렉토리 정의	53
연관된 지시문	53
구성 및 관리 양식	54
기본 환영 페이지 정의	54
연관된 지시문	55
구성 및 관리 양식	55
제 12 장 FTP 연결 관리	57
FTP 파일 보호	57
FTP 서버 로그인 관리	57
FTP 디렉토리 경로 관리	58
FTP 체인 관리	59
제 13 장 서버 처리 사용자 정의	61
정보 포함	61
정보 포함에 대한 고려사항	61
정보 포함의 구성	61
정보 포함의 형식	62
정보 포함을 위한 지시문	62
오류 메시지 사용자 정의	69
RTSP(Real Time Streaming Protocol) 경로 재지정	69
RTSP 경로 재지정 정보	70
RTSP 한계	70
RTSP 향상	70
RTSP 경로 재지정 구성	70
제 14 장 헤더 옵션 구성	73
연관된 지시문	73
구성 및 관리 양식	74
제 15 장 API(application programming interface) 정보	75
연관된 지시문	75
구성 및 관리 양식	75
제 4 부 프록시 서버 캐시 구성	77
제 16 장 프록시 서버 캐시 개요	79

캐시 저장영역	79
캐시 색인	79
FTP 캐시	80
DNS 캐시	81
캐시 제외	81
캐시 관리	81
제 17 장 기본 캐시 구성	83
1. 캐시 사용 가능	83
2. 캐시 저장영역 구성	83
선택적 사용자 정의	85
캐시 메모리 설정	85
디스크에 캐시 메모리 저장 또는 로드	86
캐시 필터 설정	86
조회 결과 및 동적으로 생성된 파일에 대한 캐시 구성	86
파일 만기 구성 및 가비지 콜렉션	86
자동 사전 로드 구성	86
캐시 공유 구성	86
로깅 구성	87
제 18 장 캐시되는 내용 제어	89
URL 기반 캐시 필터 구성	89
캐시 조회 응답	90
조회 응답 캐시에 대한 추가 요구사항	90
로컬로 제공된 파일 캐시	91
부분 URL로 파일 캐시	91
관련 구성 파일 지시문	91
제 19 장 캐시 콘텐츠 유지	93
파일 만기	93
캐시 최신 정보에 대한 추가 정보	94
FTP의 날짜 정보	95
캐시 최신 정보 구성	96
가비지 콜렉션	98
가비지 콜렉션 구성	98
제 20 장 자동 새로 고침 및 사전 로드 에 대한 캐시 에이전트 구성	99
서버 호스트 이름 설정	100
캐시를 고유한 파일로 사전 로드	101
캐시를 자주 캐시된 파일로 사전 로드	101
링크 캐시(delving)	102
관련 프록시 구성 파일 지시문	104
수동으로 캐시 에이전트 시작	105
제 21 장 공유 캐시 사용	107
원격 캐시 액세스	107

원격 캐시 액세스 구성	108
인터넷 캐시 프로토콜 플러그인 구성	108
ICP 플러그인 구성	108
제 22 장 동적 생성 콘텐츠 캐시.	111
프록시 캐시에 대한 IBM WebSphere Application Server 구성.	112
Application Server에 동적 캐시 구성	112
Application Server 어댑터 구성	112
동적 캐시에 대한 Caching Proxy 구성	113
동적 캐시 플러그인을 사용 가능하게 하도록 Service 지시문 설정	113
파일 원본을 지정하도록 ExternalCacheManager 지시문 설정	114
제 23 장 프록시 서버 캐시 조정.	115
캐시 저장 매체 선택	115
디스크 캐시 성능 최대화	115
캐시 가비지 콜렉션	115
플랫폼 고유 최적화	116
AIX	116
HP-UX 및 Solaris	116
Windows	116
제 5 부 Caching Proxy 보안 구성. . . .	117
제 24 장 프록시 서버 보안 정보.	119
제 25 장 서버 보호 설정	121
구성 및 관리 양식을 사용하여 보호 설정	121
구성 파일 지시문을 사용하여 보호 설정	122
기본 보호 설정.	123
제 26 장 SSL(Secure Sockets Layer)	125
SSL 핸드셰이크	125
SSL 성능 조정.	126
SSL 터널링.	127
SSL 터널링 구성	128
보안 원격 관리 구성	130
키 및 인증 관리	130
인증 기관	131
IBM Key Manager 유틸리티 사용	132
새 키 데이터베이스, 암호, 숨김 파일 작성.	134
인증 기관 인증 받기.	138
인증 기관 인증 저장.	139
지원된 암호 스펙	140
제 27 장 암호 하드웨어 지원 사용 가능	143

제 28 장 Tivoli Access Manager 플러그인 사용	145
구성	145
구성 스크립트를 사용하기 전에 수행할 단계	145
구성 스크립트 사용	145
Caching Proxy 및 Access Manager 플러그인 시 작	146
제 29 장 PAC-LDAP 권한 부여 모듈 사용. . . .	147
개요	147
인증	147
권한 부여	147
LDAP(Lightweight Directory Access Protocol).	148
설치	148
보안 PACD-LDAP 서버 연결의 추가 요구사항 및 제한사항	149
LDAP 패키지에서 필요한 GSKit.	149
Linux 시스템에는 LD_PRELOAD 환경 변수를 설정해야 합니다.	149
Linux 시스템에서 IBM Tivoli Directory Server (ITDS) 6.0 LDAP 클라이언트를 사용하는 경우, PACD 프로세스의 시작 실패.	150
AIX 시스템에서 IBM Tivoli Directory Server(ITDS) LDAP 클라이언트를 사용하는 경 우, PAC-LDAP 모듈은 로드할 수 없음	150
ibmproxy.conf 파일을 편집하여 PAC-LDAP 권한 부여 모듈을 사용 가능하게 하기	150
PAC-LDAP 권한 부여 모듈 구성 파일 편집.	152
paccp.conf	152
pac.conf	153
pacpolicy.conf	153
pac_ldap.cred 작성	154
pacd 시작 및 정지	155
제 6 부 Caching Proxy 모니터링	157
제 30 장 로그 구성	159
로그 정보	159
로그 파일 이름 및 기본 옵션	159
액세스 로그 필터	160
로그된 콘텐츠를 제어하려는 이유.	161
액세스 로그 필터 구성	161
기본 로그 설정.	162
로그 유지보수 및 보존	163
로그 파일 시나리오	164
제 31 장 서버 활동 모니터 사용.	167

부록 A. Caching Proxy 명령 사용	171
cgiparse 명령	172
cgiutils 명령	175
htadm 명령	178
htcformat 명령	181
ibmproxy 명령	183
부록 B. 구성 파일 지시문	185
재시작 시 변경되지 않는 지시문	185
지시문 개요	185
허용 가능 값	186
구성 파일 레코드 구문	187
Caching Proxy 지시문	187
AcceptAnything — 모든 파일 제공	187
AccessLog — 액세스 로그 파일에 대한 경로 이름 지정	188
AccessLogExcludeMethod — 지정된 메소드에 서 요청한 파일 또는 디렉토리에 대한 로그 항목 억제	189
AccessLogExcludeMimeType — 특정 MIME 유형에 대한 프록시 액세스 로그 입력 억제	189
AccessLogExcludeReturnCode — 고유한 리턴 코드에 대한 로그 입력 항목 억제	190
AccessLogExcludeURL — 고유한 파일 또는 디렉토리에 대한 로그 입력 항목 억제	190
AccessLogExcludeUserAgent — 고유한 브라 우저에서 로그 입력 항목 억제	191
AddBlankIcon — 디렉토리 목록의 표제 정렬에 사용된 아이콘의 URL 지정	191
AddDirIcon — 디렉토리 목록의 디렉토리에 대 한 아이콘 URL 지정	192
AddEncoding — 특정 접미부가 있는 파일의 MIME 콘텐츠 인코딩 지정	193
AddIcon — MIME 콘텐츠 유형이나 인코딩 유 형에 아이콘 바인드	193
AddParentIcon — 디렉토리 목록의 상위 디렉토 리를 표시하는 아이콘에 URL 지정	194
AddType — 특정 접미부가 있는 파일의 데이터 유형 지정	195
AddUnknownIcon — 디렉토리 목록의 알 수 없는 파일 유형에 대한 아이콘 URL 지정	196
AdminPort — 관리 페이지나 양식을 요청하기 위한 포트 지정	197
AggressiveCaching — 캐시할 수 없는 파일에 대한 캐시 지정	197
AlwaysWelcome — 환영 파일의 요청된 디렉토 리 탐색 여부 지정	198

appendCRLFtoPost — CRLF를 POST 요청에 추가	198
ArrayName — 원격 캐시 배열 이름 지정	199
Authentication — 인증 단계 사용자 정의	199
Authorization — 권한 부여 단계 사용자 정의	200
AutoCacheRefresh — 캐시 새로 고침을 사용할 지 여부 지정	200
BindSpecific — 서버가 하나 또는 모든 IP 주 소로 바인드 여부 지정	201
BlockSize — 캐시 내 블록 크기 지정	201
CacheAccessLog — 캐시 액세스 로그 파일에 대한 경로 지정	201
CacheAlgorithm — 캐시 알고리즘 지정	202
CacheByIncomingUrl — 캐시 파일 이름 생성 을 위한 기초 지정	202
CacheClean — 캐시 파일 보존 기간 지정	203
CacheDefaultExpiry — 파일에 대한 기본 만기 시간 지정	203
CacheDev — 캐시 저장영역 장치 지정	204
CacheExpiryCheck — 서버가 만기된 파일을 리턴할지 여부 지정	204
CacheFileSizeLimit — 캐시될 파일의 최대 크 기 지정	205
CacheLastModifiedFactor — 만기 날짜 판별을 위한 값 지정	205
CacheLocalDomain — 로컬 도메인을 캐시할지 여부 지정	206
CacheMatchLanguage — 리턴된 캐시 콘텐츠의 언어 환경 설정을 지정	206
CacheMaxExpiry — 캐시 파일의 최대 수명 지 정	207
CacheMemory — 캐시 RAM 지정	208
CacheMinHold — 파일을 사용 가능하게 보존 하는 기간 지정	209
CacheNoConnect — 독립형 캐시 모드 지정	209
CacheOnly — 템플리트와 일치하는 URL이 있 는 파일만 캐시	209
CacheQueries — 물음표(?)를 포함하는 URL에 캐시 응답 지정	210
CacheRefreshInterval — 캐시된 오브젝트의 유효 성을 재확인하기 위한 시간 간격 지정	211
CacheRefreshTime — 캐시 에이전트를 시작할 시기 지정	211
CacheTimeMargin — 파일 캐시를 위한 최소 수명 지정	211

CacheUnused — 사용되지 않는 캐시 파일 보존 기간 지정	212
Caching — 프록시 캐시 사용 가능.	212
CompressAge — 로그 압축 시기 지정	213
CompressCommand — 압축 명령 및 매개변수 지정	213
CompressDeleteAge — 로그 삭제 시기 지정	214
CompressionFilterAddContentType — 압축하려는 HTTP 응답의 콘텐츠 유형을 지정	215
CompressionFilterEnable — 압축 필터를 사용하여 HTTP 응답을 압축	216
ConfigFile — 추가 구성 파일의 이름 지정	216
ConnThreads — 연결 관리에 사용되는 연결 스레드의 수를 지정	216
ContinueCaching — 캐시에 필요한 파일 수 지정	216
DefinePicsRule — 콘텐츠 필터링 규칙 공급	217
DefProt — 템플릿과 일치하는 요청에 대한 기본 보호 설정 지정	217
DelayPeriod — 요청 간 일시정지 지정	220
DelveAcrossHosts — 도메인을 통한 캐시 지정	220
DelveDepth — 캐시하는 동안 연결을 따르는 범위 지정	220
DelveInto — 캐시 에이전트가 연결을 따르는지를 지정	220
DirBackgroundImage — 디렉토리 목록에 배경라운드 이미지 지정	221
DirShowBytes — 디렉토리 목록에 작은 파일에 대한 바이트 수 표시	221
DirShowCase — 디렉토리 목록에 파일을 분류할 때 대소문자 사용	222
DirShowDate — 디렉토리 목록에 최종 변경 날짜 표시	222
DirShowDescription — 디렉토리 목록에 파일에 대한 설명 표시	222
DirShowHidden — 디렉토리 목록에 숨겨진 파일 표시	222
DirShowIcons — 디렉토리 목록에 아이콘 표시	223
DirShowMaxDescrLength — 디렉토리 목록에서 설명의 최대 길이 지정	223
DirShowMaxLength — 디렉토리 목록에서 파일 이름의 최대 길이 지정	223
DirShowMinLength — 디렉토리 목록에 파일 이름의 최소 길이 지정	224
DirShowSize — 디렉토리 목록에 파일 크기 표시	224

Disable — HTTP 메소드 사용 불가능	224
DisInheritEnv — CGI 프로그램이 계승하지 않는 환경 변수 지정.	225
DNS-Lookup — 서버가 클라이언트 호스트 이름을 조회할지 여부 지정	225
Enable — HTTP 메소드 사용 가능	226
EnableTcpNodelay — TCP NODELAY 소켓 옵션을 사용 가능하게 함	226
Error — 오류 단계 사용자 정의.	227
ErrorLog — 서버 오류를 로그할 파일 지정	227
ErrorPage — 특정 오류 조건에 대해 사용자 정의된 메시지 지정	228
EventLog — 이벤트 로그 파일에 대한 경로 지정	229
Exec — 요청을 일치시키기 위한 CGI 프로그램 실행	230
ExportCacheImageTo — 디스크로 캐시 메모리 내보내기	232
ExternalCacheManager — IBM WebSphere Application Server의 동적 캐시에 대한 Caching Proxy 구성.	232
Fail — 일치하는 요청 거부	233
FIPSEnable — SSLV3 및 TLS에 대해 FIPS(Federal Information Processing Standard) 승인 암호를 사용 가능하게 함	234
flexibleSocks — 융통성있는 SOCKS 구현 사용 가능	234
FTPDDirInfo — 디렉토리에 대한 환영 또는 설명 메시지 생성.	235
ftp_proxy — FTP 요청에 대한 다른 프록시 서버 지정	235
FTPUrlPath — FTP URL이 해석되는 방법 지정	235
Gc — 가비지 콜렉션 지정.	236
GCAdvisor — 가비지 콜렉션 프로세스 사용자 정의	236
GcHighWater — 가비지 콜렉션 시작 시간 지정	237
GcLowWater — 가비지 콜렉션 종료 시기 지정	237
gopher_proxy — Gopher 요청에 대한 다른 프록시 서버 지정.	237
GroupId — 그룹 ID 지정.	238
HeaderServerName — HTTP 헤더에 리턴되는 프록시 서버 이름 지정	238
Hostname — 서버에 대한 정식 도메인 이름 또는 IP 주소 지정	239

http_proxy — HTTP 요청에 대한 다른 프록시 서버 지정	239
HTTPSCheckRoot — HTTPS 요청 필터.	239
ICP_Address — ICP 조회의 IP 주소 지정	240
ICP_MaxThreads — ICP 조회를 위한 최대 스레드 지정	240
Occupier — ICP 클러스터의 구성원 지정	241
ICP_Port — ICP 조회의 포트 번호 지정.	241
ICP_Timeout — ICP 조회의 최대 대기 시간 지정	242
IgnoreURL — 새로 고치지 않을 URL 지정	242
imbeds — 정보 포함 처리가 사용될지 여부 지정	242
ImportCacheImageFrom — 파일에서 캐시 메모리 가져오기	244
InheritEnv — CGI 프로그램이 계승할 환경 변수 지정	244
InputTimeout — 입력 시간 종료 지정.	244
JunctionReplaceUrlPrefix — JunctionRewrite 플러그인과 사용되는 경우, 접두부 삽입 대신 URL 바꾸기	245
JunctionRewrite — URL 재작성 사용 가능	245
JunctionRewriteSetCookiePath — JunctionRewrite 플러그인과 사용되는 경우, Set-Cookie 헤더에 경로 옵션을 재작성.	246
JunctionSkipUrlPrefix — JunctionRewrite 플러그인과 사용되는 경우, 이미 접두부를 포함하는 URL 재작성을 건너뛰기.	246
KeepExpired — 자원이 프록시에서 갱신되는 경우, 자원의 만기된 사본을 리턴하도록 지정	247
KeyRing — 키 링 데이터베이스에 대한 파일 경로 지정	247
KeyRingStash — 키 링 데이터베이스 암호 파일에 대한 파일 경로 지정	248
LimitRequestBody — PUT 또는 POST 요청의 최대 본문 크기 지정.	248
LimitRequestFields — 클라이언트 요청에서 헤더의 최대 수 지정.	248
LimitRequestFieldSize — 최대 헤더 길이 및 요청 행 지정	249
ListenBacklog — 서버가 수행할 수 있는 인식 백로그 클라이언트 연결 수 지정	249
LoadInlineImages — 삽입된 이미지 새로 고침 제어	249
LoadTopCached — 새로 고칠 즐겨찾기 페이지 수 지정	250

LoadURL — 새로 고칠 URL 지정	250
Log — 로그 단계 사용자 정의	250
LogArchive — 로그 보존 작동 지정	251
LogFileFormat — 액세스 로그 형식 지정	252
LogToGUI (Windows only) — 서버 창에 로그 입력 항목 표시	252
LogToSyslog — 액세스 정보를 시스템 로그에 전송할지 여부 지정(Linux 및 UNIX 전용)	252
Map — 규칙을 일치시키는 요청 경로 문자열을 사용하여 새 요청 문자열에 일치하는 요청 변경	253
MapQuery — 규칙을 일치시키는 요청 경로 및 조회 문자열을 사용하여 일치하는 요청을 새 요청 문자열로 변경	255
MaxActiveThreads — 최대 활성 스레드 수 지정	256
MaxContentLengthBuffer — 동적 데이터에 대한 버퍼 크기 지정.	256
MaxLogFileSize — 각 로그 파일의 최대 크기 지정	257
MaxPersistRequest — 지속적인 연결에서 수신할 요청의 최대수 지정	258
MaxQueueDepth — 대기열에 넣을 URL의 최대수 지정	258
MaxRuntime — 캐시 에이전트의 최대 실행 시간 지정	258
MaxSocketPerServer — 서버의 개방형 대기 소켓 최대수 지정.	259
MaxUrls — 새로 고칠 URL의 최대수 지정	259
Member — 배열의 구성원 지정.	259
Midnight — 로그 보존에 사용되는 API 플러그인 지정	260
NameTrans — 이름 변환 단계 사용자 정의	261
NoBG — Caching Proxy 프로세스를 포그라운드에 실행	262
NoCaching — 템플리트와 일치하는 URL이 있는 파일을 캐시하지 않도록 지정	262
NoLog — 템플리트와 일치하는 고유한 호스트나 도메인에 대한 로그 입력 항목 압축.	263
no_proxy — 도메인에 직접 연결하기 위한 템플리트 지정	263
NoCacheOnRange — 범위 요청에 대해 캐시를 지정하지 않음	264
NoProxyHeader — 차단시킬 클라이언트 헤더 지정	265
NumClients — 사용할 캐시 에이전트 스레드의 수 지정	265

ObjectType — 오브젝트 유형 단계 사용자 정의	266
OptimizeRuleMapping — 규칙의 수가 증가하는 경우, 수신 요청에 대한 규칙 맵핑 프로세스를 최적화	266
OutputTimeout — 출력 시간 종료 지정	267
PacFilePath — PAC 파일이 들어 있는 디렉토리 지정	267
Pass — 요청을 승인하기 위한 템플릿 지정	267
PersistTimeout — 클라이언트가 다른 요청을 전송하기 위한 대기 시간 지정.	269
PICSDbLookup — PICS 레이블 검색 단계 사용자 정의	270
PidFile(Linux 및 UNIX 전용) — Caching Proxy의 프로세스 ID를 저장할 파일 지정.	270
PKCS11DefaultCert, PKCS11DriverPath, PKCS11TokenPassword — IBM 4960 PCI 암호화 액셀러레이터 카드(AIX 전용) 지원	271
플러그인 모듈 지시문.	271
Port — 서버가 요청을 인식하는 포트 지정	272
PostAuth — PostAuth 단계 사용자 정의.	273
PostExit — PostExit 단계 사용자 정의	273
PreExit — PreExit 단계 사용자 정의.	274
Protect — 템플릿과 일치하는 요청에 대한 보호 설정 활성화.	274
Protection — 구성 파일 내에 명명된 보호 설정	279
Protection subdirectives — 자원 세트가 보호되는 방법 지정.	280
Proxy — 프록시 프로토콜 또는 역방향 프록시 지정	282
ProxyAccessLog — 프록시 액세스 로그 파일에 대한 경로 이름 지정.	284
ProxyAdvisor — 프록시 요청의 서비스 사용자 정의	285
ProxyForwardLabels — PICS 필터링 지정	285
ProxyFrom — 클라이언트를 From: 헤더로 지정	285
ProxyIgnoreNoCache — 재로드 요청 무시	286
ProxyPersistence — 지속적인 연결 허용	286
ProxySendClientAddress — Client IP Address: 헤더 생성	287
ProxyUserAgent — 사용자 에이전트 문자열 수정	287
ProxyVia — HTTP 헤더의 형식 지정.	288
ProxyWAS — WebSphere Application Server로 요청 전송 지정	288

PureProxy — 전용 프록시 사용 불가능	288
PurgeAge — 로그의 유효 기간 한계 지정	289
PurgeSize — 로그 보존 크기의 한계 지정	289
RCAConfigFile — ConfigFile의 별명 지정	290
RCAThreads — 포트당 스레드 수 지정	290
ReadTimeout — 연결의 시간 한계 지정	290
Redirect — 다른 서버에 전송된 요청에 대한 템플릿 지정	291
RegisterCacheIdTransformer — 쿠키 헤더를 기반으로 하는 자원에 하나 이상 변환을 개시	292
ReversePass — 자동으로 재지정된 요청 교차	293
RewriteSetCookieDomain — 재작성해야 할 도메인 패턴 지정.	294
RTSPEnable — RTSP 경로 재지정 사용 가능	294
rtsp_proxy_server - 경로 재지정 서버 지정	294
rtsp_proxy_threshold — 캐시로의 경로 재지정 이전에 요청 수 지정	295
rtsp_url_list_size — 프록시 메모리에서 URL 수 지정	295
RuleCaseSense — 대소문자를 구분하지 않는 응용프로그램 URL에서 요청을 맵핑.	296
ScriptTimeout - 스크립트의 시간 종료 설정 지정	296
SendHTTP10Outbound — 프록시된 요청에 대한 프로토콜 버전 지정	296
SendRevProxyName — HOST 헤더에 Caching Proxy 호스트 이름 지정	297
ServerConnGCRun — 가비지 콜렉션 스레드 실행 간격 지정.	298
ServerConnPool — 기점 서버 연결 풀링 지정	298
ServerConnTimeout — 최대 비활성 기간 지정	298
ServerInit — 서버 초기설정 단계 사용자 정의	299
ServerRoot — 서버 프로그램이 설치될 디렉토리 지정	299
ServerTerm — 서버 종료 단계 사용자 정의	300
Service — 서비스 단계 사용자 정의	300
SignificantURLTerminator — URL 요청의 종료 코드 지정	301
SMTPServer(Windows 전용) — sendmail 루틴에 대한 SMTP 서버 설정	301
SNMP — SNMP 지원 사용 가능 및 사용 불가능	302
SNMPCommunity — SNMP에 대한 보안 암호 제공	302
SSLCaching — 보안 요청에 대한 캐시 사용 가능	302

SSLCertificate — 인증서에 대한 키 레이블 지정	303
SSLCryptoCard — 설치된 암호 카드 지정	304
SSLEnable — 보안 요청에 대한 포트 443의 인식 지정	304
SSLForwardPort — HTTP SSL 업그레이드에 대해 주소 지정할 포트 지정	305
SSLOnly — HTTP 요청의 리스너 스레드 사용 불가능.	305
SSLPort — 기본값 외 HTTPS 리스닝 포트 지정	305
SSLTunneling — SSL 터널링 사용 가능	306
SSLVersion — SSL 버전 지정	306
SSLV2Timeout — SSLV2 세션이 만기되기 전에 대기할 시간 지정	307
SSLV3Timeout — SSLV3 세션이 만기되기 전에 대기할 시간 지정	307
SuffixCaseSense — 접미부 정의가 대소문자 구분되는지 여부 지정	307
SupportVaryHeader — HTTP Vary 헤더를 기반으로 하는 자원에 하나 이상의 변형을 캐시	308
TLSV1Enable — 전송 계층 보안 프로토콜 사용 가능	309

Transmogrifier — 데이터 조작 단계 사용자 정의	310
TransmogrifiedWarning — 클라이언트에게 경고 메시지 전송.	310
TransparentProxy — Linux에서 투명 프록시를 사용 가능하게 함	310
UpdateProxy — 캐시 대상 지정.	311
UserId — 기본 사용자 ID 지정.	312
V2CipherSpecs — SSL 버전 2에 지원되는 암호 스펙 나열	313
V3CipherSpecs — SSL 버전 3에 지원되는 암호 스펙 나열	313
WebMasterEMail — 서버 선택 보고서를 수신할 전자 우편 주소 설정.	314
WebMasterSocksServer(Windows 전용) — sendmail 루틴에 대한 socks 서버 설정	315
Welcome — 환영 파일의 이름 지정	315
주의사항	317
상표	319

그림

- | | | | |
|----------------------------|-----|----------------------|-----|
| 1. 링크 캐시(delving). | 103 | 2. SSL 터널링 | 128 |
|----------------------------|-----|----------------------|-----|

이 책에 대한 정보

이 서론에서는 이 책의 독자 및 목적, 조직, 액세스할 수 있는 기능, 규약 및 용어, 관련 문서에 대해 설명합니다.

이 책의 사용자

Caching Proxy 관리 안내서는 운영 체제 및 인터넷 서비스 제공에 익숙하고 경력있는 네트워크 및 시스템 관리자를 위한 것입니다. *Caching Proxy*를 사용하기 전에는 필요하지 않습니다.

이 책은 *Caching Proxy* 이전 릴리스를 지원하지 않습니다.

책에 사용된 규약 및 용어

이 문서에는 다음과 같은 서체와 키 규약을 사용합니다.

표 1. 이 책에 사용된 규약

규약	의미
굵은체	굵은체는 GUI(Graphical User Interface)와 관련하여 메뉴, 메뉴 항목, 레이블, 단추, 아이콘 및 폴더를 나타냅니다. 또한 주위의 텍스트와 혼동될 수 있는 명령 이름을 강조하는 데에도 사용될 수 있습니다.
모노스페이스	명령 프롬프트에 입력해야 할 텍스트를 표시합니다. 모노스페이스는 화면 텍스트, 코드 예제 및 파일 발췌 부분을 표시하기도 합니다.
기울임꼴	지정해야 할 변수값을 표시합니다(예: <i>fileName</i> 에 파일 이름을 지정합니다). 강조 및 책 제목도 표시합니다.
Ctrl-x	여기서 x는 키 이름으로, 제어 문자 순서를 표시합니다. 예를 들어, Ctrl-c는 Ctrl 키를 누른 상태에서 c 키를 누르는 것을 의미합니다.
Return	Return, Enter 또는 왼쪽 화살표로 표시된 키를 나타냅니다.
%	루트 특권이 필요없는 명령에 대한 Linux™ 및 UNIX® 명령 셸 프롬프트를 표시합니다.
#	루트 특권이 필요한 명령에 대한 Linux 및 UNIX 명령 셸 프롬프트를 표시합니다.
C:\	Windows® 명령 프롬프트를 표시합니다.
명령 입력	명령을 “입력” 또는 “실행”할 때 명령을 입력하고 Return을 누릅니다. 예를 들어, “Enter the ls command”라는 명령은 명령 프롬프트에 ls 를 입력하고 Return을 누르는 것을 의미합니다.
[]	구문 설명에 선택적 항목을 넣습니다.
{ }	선택할 항목이 있는 목록을 구문 설명에 넣습니다.
	구문 설명에서 { }(중괄호)에 있는 선택사항 목록의 항목을 구분합니다.
...	구문 설명에서 줄임표는 앞의 항목을 한 번 이상 반복할 수 있다는 것을 의미합니다. 예제에서 줄임표는 간결하게 하기 위해 예제에서 정보를 생략했음을 의미합니다.

내게 필요한 옵션

내게 필요한 옵션 기능은 지체 부자유나 시각 장애와 같은 신체 장애를 가진 사용자가 소프트웨어 제품을 잘 사용할 수 있도록 도와줍니다. 이 기능은 WebSphere® Application Server, 버전 6.1에 주요한 내게 필요한 옵션 기능입니다.

- 화면 판독기 소프트웨어와 디지털 음성 신시사이저를 사용하면 화면에 표시된 내용을 들을 수 있습니다. IBM ViaVoice와 같은 음성 인식 소프트웨어를 사용하여 데이터를 입력하고 사용자 인터페이스를 탐색할 수도 있습니다.
- 모든 기능은 마우스 대신 키보드를 사용하여 작동할 수 있습니다.
- 제공된 그래픽 인터페이스 대신 표준 문서 편집기나 명령행 인터페이스를 사용하여 Application Server 기능을 구성하고 관리할 수 있습니다. 특정 기능의 액세스에 필요한 옵션에 대한 자세한 정보는 해당 기능 문서를 참조하십시오.

고객 의견을 보내는 방법

고객의 피드백은 가장 정확하고 최고의 정보를 제공하는 데 매우 중요합니다. 이 책 또는 WebSphere Application Server Edge Components의 기타 문서에 대한 의견이 있는 경우 다음의 방법에 따라 보내 주시기 바랍니다.

- 여러분의 의견을 전자 우편으로 fsdoc@us.ibm.com에 보내 주십시오. 책 이름, 부품 번호, WebSphere Application Server 버전 그리고 가능하면 의견 관련 특정 텍스트 위치(예: 페이지 번호 또는 테이블 번호)를 명시하십시오.

관련 정보

- *Edge Components*용 개념, 계획 및 설치, GC31-6918-00
- *Edge Components*용 프로그래밍 안내서, GC31-6919-00
- *Load Balancer* 관리 안내서, GC31-6921-00
- *IBM WebSphere* 에지 서비스 구조
- IBM 홈 웹 사이트: www.ibm.com/
- IBM® WebSphere Application Server 제품 웹 사이트:
www.ibm.com/software/webservers/appserv/
- IBM WebSphere Application Server 라이브러리 웹 사이트:
www.ibm.com/software/webservers/appserv/library.html
- IBM WebSphere Application Server 지원 웹 사이트:
www.ibm.com/software/webservers/appserv/support.html
- IBM WebSphere Application Server Information Center:
www.ibm.com/software/webservers/appserv/infocenter.html

- IBM WebSphere Application Server Edge Components Information Center:
www.ibm.com/software/webservers/appserv/ecinfocenter.html

제 1 부 Caching Proxy 시작하기

이 파트에서는 Caching Proxy 컴포넌트의 개요, 구성 및 관리 양식 및 구성 마법사 사용에 필요한 명령, ibmproxy.conf 파일을 수동으로 편집하는 데 필요한 명령, 프록시 서버 시작 및 정지 프로시저를 제공합니다.

이 파트에는 다음과 같은 장이 들어 있습니다.

3 페이지의 제 1 장 『개요』

7 페이지의 제 2 장 『구성 및 관리 양식 사용법』

11 페이지의 제 3 장 『구성 마법사 사용』

13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』

15 페이지의 제 5 장 『Caching Proxy 시작 및 정지』

제 1 장 개요

역방향 프록시 또는 정방향 프록시로 작동. Caching Proxy는 클라이언트의 데이터 요청을 차단하고 콘텐츠 호스팅 시스템에서 요청 정보를 검색하며, 해당 콘텐츠를 클라이언트로 다시 전달합니다. 일반적으로, 대부분의 요청은 웹 서버 시스템(기점 서버 또는 콘텐츠 호스트라고도 부름)에 저장되어 있는 문서에 대한 것으로 HTTP(하이퍼텍스트 전송 프로토콜)를 통하여 전달됩니다. 그러나 Caching Proxy를 구성하면 FTP(파일 전송 프로토콜) 및 Gopher와 같은 다른 프로토콜을 처리할 수 있습니다.

Caching Proxy는 캐시 가능한 콘텐츠를 요청자에게 전달하기 전에 로컬 캐시에 저장합니다. 캐시 가능한 콘텐츠의 예에는 정적 웹 페이지와 동적으로 생성했으나 드물게 변하는 단편 JSP(JavaServer Pages) FILES가 포함됩니다. 캐시를 사용하면 Caching Proxy가 콘텐츠 호스트에서 다시 검색하는 것보다 훨씬 빨리, 로컬 캐시에서 직접 전달하여 동일한 콘텐츠에 대한 후속 요청을 충족시킬 수 있습니다.

중요: Caching Proxy는 다음 예외 사항을 포함하여 모든 Edge component 설치에서 사용 가능합니다.

- Caching Proxy는 Itanium 2 또는 AMD Opteron 64비트 프로세서에서 실행하는 Edge component 설치에서 사용 불가능합니다.
- Caching Proxy는 IPv4 및 IPv6용 Load Balancer의 Edge component 설치에서 사용 불가능합니다.

기본 Caching Proxy 구성

두 가지 기본 프록시 구성은 역방향 프록시와 정방향 프록시입니다.

역방향 프록시(기본값)

기본적으로, Caching Proxy는 역방향 프록시 서버로 구성됩니다. 역방향 프록시 서버 구성에서, 프록시 서버는 하나 이상의 콘텐츠 서버 및 인터넷 사이에 있습니다. 이는 프록시 서버의 홈 사이트에 저장된 콘텐츠에 대한 인터넷 클라이언트의 요청을 승인합니다. 프록시 서버는 기점(콘텐츠) 서버가 되는 클라이언트로 나타납니다. 클라이언트는 다른 서버로 전송되는 요청을 인지하지 않습니다.

정방향 프록시

또는 Caching Proxy를 정방향 프록시 서버로 구성할 수 있습니다. 그러나, 클라이언트 브라우저를 개별적으로 구성해야 프록시를 사용할 수 있습니다. 정방향 프록시 서버 구

성에서, 프록시 서버는 클라이언트 및 인터넷 사이에 있습니다. Caching Proxy는 클라이언트의 요청을 인터넷에 위치한 콘텐츠 서버에 전달하고, 검색된 데이터를 캐시하여, 클라이언트에 검색된 데이터를 전달합니다.

정방향 프록시 구성을 사용하려면, ibmproxy.conf 구성 파일을 다음과 같이 변경해야 합니다.

- 다음 행의 주석을 제거하여 Caching Proxy를 전달하는 프로토콜을 지정하십시오.

```
Proxy http:*  
Proxy ftp:*  
Proxy gopher:*
```

- SSL 터널링을 사용 가능하게 하여 SSL 요청이 정방향 프록시 구성에서 핸들되도록 하십시오.

```
SSLTunneling On
```

SSL 터널링에 대한 자세한 정보는 128 페이지의 『SSL 터널링 구성』을 참조하십시오.

- Enable 지시문을 사용하여 CONNECT 메소드를 사용 가능하게 하십시오.

```
Enable CONNECT OutgoingPorts All
```

또는

```
Enable CONNECT OutgoingPorts 443
```

Enable CONNECT 메소드에 대한 형식 및 사용 가능한 옵션에 대한 정보는 128 페이지의 『SSL 터널링 구성』을 참조하십시오.

이렇게 변경하면, 정방향 프록시가 다음을 수행할 수 있습니다.

- 하이퍼텍스트 전송 프로토콜 또는 파일 전송 프로토콜에서 클라이언트의 요청에 응답합니다.
- Gopher 검색 엔진의 요청에 응답합니다.
- 트랜잭션의 지속 기간에 대해 클라이언트 및 해당 현재 서버 간 유사성을 유지보수합니다.

투명 프록시(Linux 시스템 전용)

정방향 Caching Proxy 변형은 투명 Caching Proxy입니다. 이 역할에서 Caching Proxy는 기본적인 정방향 Caching Proxy로 동일한 기능을 수행하나, 존재를 인지하는 클라이언트가 있어야 수행합니다. 투명 Caching Proxy 구성은 Linux 시스템에서만 지원됩니다.

일반적인 정방향 Caching Proxy와 같이 투명 Caching Proxy는 인터넷/게이트웨이와 가까이 있는 시스템에 설치되나, 클라이언트 브라우저 프로그램은 정방향 Caching

Proxy에 요청을 지정하도록 구성되지 않습니다. 클라이언트는 프록시가 구성에 있는 것을 인지하지 않습니다. 대신, 라우터가 클라이언트 요청을 가로채어, 투명 Caching Proxy로 설정하도록 구성합니다.

이 구성을 위한 지시문에 대한 정보는 310 페이지의 『TransparentProxy — Linux에서 투명 프록시를 사용 가능하게 함』을 참조하십시오.

새로운 기능 지원

버전 6.1 *Caching Proxy* 관리 안내서에서는 새로 문서화된 기능 및 수정 서비스가 포함되어 있습니다.

가장 중요한 기능은 다음과 같습니다.

- 정방향 프록시 지원

정방향 프록시 구성에 대한 정보는 3 페이지의 『정방향 프록시』를 참조하십시오.

- 투명 프록시 지원(Linux 시스템 전용)

투명(정방향) 프록시 지시문에 대한 정보는 310 페이지의 『TransparentProxy — Linux에서 투명 프록시를 사용 가능하게 함』을 참조하십시오.

- WebDAV 메소드, Microsoft Exchange Server 메소드 및 사용자 정의 메소드 지원

이러한 메소드에 대한 정보는 43 페이지의 『WebDAV 메소드, MS Exchange 메소드, 사용자 정의 메소드 사용 가능』을 참조하십시오.

- HTTP 압축 필터 지시문

이러한 지시문에 대한 정보는 215 페이지의 『CompressionFilterAddContentType — 압축하려는 HTTP 응답의 콘텐츠 유형을 지정』 및

216 페이지의 『CompressionFilterEnable — 압축 필터를 사용하여 HTTP 응답을 압축』을 참조하십시오.

- 범위 요청 지시문에 캐시 없음

이 지시문에 대한 정보는 264 페이지의 『NoCacheOnRange — 범위 요청에 대해 캐시를 지정하지 않음』을 참조하십시오.

- 규칙 맵핑 지시문 최적화

이 지시문에 대한 정보는 266 페이지의 『OptimizeRuleMapping — 규칙의 수가 증가하는 경우, 수신 요청에 대한 규칙 맵핑 프로세스를 최적화』를 참조하십시오.

- MapQuery 지시문

맵 지시문과 유사하게, MapQuery는 경로 및 조회 문자열을 사용하여 규칙을 일치 시킵니다.

이 지시문에 대한 정보는 255 페이지의 『MapQuery — 규칙을 일치시키는 요청 경로 및 조회 문자열을 사용하여 일치하는 요청을 새 요청 문자열로 변경』을 참조하십시오.

- RuleCaseSense 지시문

이 지시문에 대한 정보는 296 페이지의 『RuleCaseSense — 대소문자를 구분하지 않는 응용프로그램 URL에서 요청을 맵핑』을 참조하십시오.

- AIX 시스템의 경우, IBM 4960 PCI 암호화 액셀러레이터 카드를 지원하도록 추가 지시문이 제공됩니다.

이러한 지시문에 대한 정보는 271 페이지의 『PKCS11DefaultCert, PKCS11DriverPath, PKCS11TokenPassword — IBM 4960 PCI 암호화 액셀러레이터 카드(AIX 전용) 지원』을 참조하십시오.

- SSLCertificate 지시문의 논리식 옵션

이 지시문의 논리식 옵션에 대한 정보는 303 페이지의 『SSLCertificate — 인증서에 대한 키 레이블 지정』을 참조하십시오.

- 프록시 또는 ProxyWAS 규칙에서 사용 가능한 추가 옵션

Caching Proxy에서는 런타임 시 추가 패턴 일치가 필요한 지시문을 제공합니다. Caching Proxy 성능을 향상시키기 위해, 이 지시문을 프록시 또는 ProxyWAS 규칙의 옵션으로 사용할 수 있습니다. Proxy 또는 ProxyWAS 규칙의 추가 옵션에 대한 자세한 정보는 282 페이지의 『Proxy — 프록시 프로토콜 또는 역방향 프록시 지정』을 참조하십시오.

제 2 장 구성 및 관리 양식 사용법

Caching Proxy는 요청 중인 클라이언트에 제공되어 프록시 서버를 구성하는 데 사용될 수 있는 HTML 양식과 함께 제공됩니다. 이 양식은 로컬 프록시 서버 구성 파일인 `ibmproxy.conf`를 편집하는 CGI 프로그램을 실행합니다. 이 양식을 사용하려면 프록시 서버가 실행 중이어야 하고, 양식이 상주하는 로컬 디렉토리에서 양식을 전달하도록 구성되어야 합니다.

기본적으로 Caching Proxy는 구성 및 관리 양식에 대한 액세스를 사용 가능하게 하는 `ibmproxy.conf` 파일에 포함된 Pass 지시문과 함께 설치됩니다. 클라이언트가 이 프록시 서버에서 기본 홈 페이지를 요청하면 `Frntpage.html`이 제공됩니다. 이 페이지에는 구성 및 관리 양식 시작 페이지 `wte.html`에 대한 하이퍼텍스트 링크가 포함됩니다.

구성 및 관리 양식은 보호되고 클라이언트 인증이 있어야 제공할 수 있습니다. 관리자의 ID 및 암호 설정에 대한 지시사항은 10 페이지의 『관리자 암호 설정』을 참조하십시오.

브라우저 요구사항

구성 및 관리 양식에 액세스하는 데 사용되는 웹 브라우저는 다음 사항을 지원해야 합니다.

- **HTML 4.0:** 모든 양식은 HTML 4.0 스펙에 기록됩니다. 웹 브라우저는 HTML 4.0 및 프레임 세트를 지원해야 합니다.
- **Java 1.1 및 JavaScript:** 애플릿은 Java 1.1 스펙에 기록됩니다. 웹 브라우저는 Java 1.1과 호환 가능한 JVM(Java Virtual Machine)을 지원해야 합니다. 애플릿은 Java 2.0 스펙과 호환 가능한 JVM(Java Virtual Machines)과는 호환되지 않습니다. JavaScript와 Java가 모두 사용 가능해야 합니다.
- **256색:** 웹 브라우저가 실행되는 워크스테이션은 최소 256색을 지원해야 합니다.

권장하는 브라우저는 Mozilla, Firefox(Linux, UNIX 및 Windows 시스템의 경우) 및 Internet Explorer(Windows 시스템의 경우)입니다. Mozilla, Firefox 및 Internet Explorer 브라우저의 특정 버전의 경우, 다음 웹 사이트를 참조하여 지원되는 소프트웨어 웹 페이지의 링크로 연결하십시오. <http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921>.

주:

1. 64비트 PowerPC Linux 시스템에서, Mozilla 브라우저로 구성 및 관리 양식에 액세스할 수 없습니다. 이 아키텍처에 사용 가능한 SDK가 없기 때문입니다. 또한 지원되는 웹 브라우저로 다른 시스템에서 구성 및 관리 양식에 액세스할 수 있습니다.
2. 관리 콘솔을 시작할 때, 로그인을 2번 수행하도록 프롬프트되는 경우, Internet Explorer에서 Java 설정이 올바르게 수행되지 않을 수 있습니다. Internet Explorer에서 이 작업을 올바르게 수행하려면, 도구 > 인터넷 옵션 > 고급을 선택하여 **Java 2 v1.4.X** 사용 선택란을 선택 취소하십시오.

구성 및 관리 양식 액세스

구성 및 관리 양식을 액세스하려면 다음을 수행하십시오.

1. 프록시 서버가 실행 중인지 확인하십시오. 프록시 서버 시작에 대한 명령은 15 페이지의 제 5 장 『Caching Proxy 시작 및 정지』를 참조하십시오.
2. HTTP 브라우저가 프록시 서버의 홈 페이지(Frontpage.html)나 구성 및 관리 시작 페이지(welcome.html)를 요청하도록 지정하십시오.

주: 이 페이지는 프록시 서버의 실제 �핑 규칙에 따라 다르며, 괄호안에 표시된 기본 페이지와 다를 수 있습니다.

`http://your.server.name[:port]/[directory]/[page.html]`

여기서

- *your.server.name*은 호스트의 전체 경로 이름입니다
(예: `http://www.ibm.com/`).
 - *[:port]* 프록시 서버가 80 이외의 포트에서 관리 요청을 인식하는 경우, 다음과 같이 서버 이름 다음에 포트 번호를 포함시키십시오.
`http://your.server.name:port`
 - */[directory]* URL 안에 디렉토리를 추가하는 것은 �핑 규칙에 따라 다릅니다.
 - */[page.html]* HTML 페이지는 환영 페이지로 나열되지 않은 경우에만 지정해야 합니다. 환영 페이지에 대한 정보는 54 페이지의 『기본 환영 페이지 정의』를 참조하십시오.
3. 구성 및 관리 양식을 눌러 서버 구성 약식으로 가십시오. 관리자의 사용자 이름 및 암호가 프롬프트됩니다. 허가된 사용자 이름 및 암호를 입력하십시오. Caching Proxy 구성 클라이언트 창이 열립니다.

주:

- a. 기본 페이지가 표시된 후 탐색 프레임 내용을 로드하는 데 몇 초가 걸립니다.

- b. Windows 2003 시스템에서, 관리 양식(CGI 스크립트)을 요청하는 연결이 연결이 완료되기 전에 재설정을 수신할 수 있습니다. 그 결과로, 브라우저는 수신된 데이터 없음 또는 사용 불가능한 메시지가 있는 페이지를 표시함을 보고합니다. 이 문제점을 해결하려면, MaxActiveThreads의 값을 200보다 더 크게 늘리고, ConnThreads의 값을 50보다 크게 설정하여 재설정 연결을 분석하십시오. 이러한 지시문에 대한 자세한 정보는 256 페이지의 『MaxActiveThreads — 최대 활성 스레드 수 지정』 및 216 페이지의 『ConnThreads — 연결 관리에 사용되는 연결 스레드의 수를 지정』을 참조하십시오.
4. 왼쪽의 탐색 프레임은 다섯 개의 주요 구성 양식 카테고리를 표시합니다.
 - 프록시 구성
 - 캐시 구성
 - 서버 구성
 - 서버 활동 모니터
 - 플러그인 구성

표제 왼쪽의 삼각형 포인터를 눌러 해당 카테고리의 구성 양식 목록을 펼치십시오. 양식을 눌러 포인터를 여십시오. 양식은 입력 필드에 현재 구성값(구성값이 있는 경우)을 표시합니다. 설치 후 구성값을 변경하지 않았으면, 기본값이 표시됩니다.
5. 어느 양식이든 해당 특정 기능에 대한 구성 정보를 입력하십시오. 각 양식은 변경 사항에 도움을 주는 명령을 제공합니다. 추가 정보는 각 양식의 맨 위에 있는 도움말 아이콘인 물음표(?)를 누르십시오. 제공되는 링크는 다음과 같습니다.
 - 필드 도움말—각 화면 패널의 필드에 대한 설명
 - 사용법...—양식을 사용하여 특정 작업을 수행하는 세부 단계
 - 색인—도움말 정보 색인
6. 양식을 채운 후 제출을 눌러 작성한 변경사항으로 서버 구성을 갱신하십시오. 제출 단추는 각 양식의 입력 필드 아래 있습니다. 양식 표시 내용을 변경하지 않으려는 경우, 재설정을 누르면 원래 값으로 양식 필드가 리턴됩니다.
7. 제출을 눌러 입력이 승인되면 다음 메시지가 맨 위 프레임에 표시됩니다.

요청한 구성 변경이 완료되었습니다.

입력이 승인되지 않으면, 맨 위 프레임에 설정을 승인할 수 없다는 오류 메시지가 표시됩니다.
8. 프록시 서버를 재시작하려면, 맨 위 프레임에서 서버 재시작 아이콘(I)을 누르십시오. 프록시 서버가 재시작 명령을 수신하면 클라이언트의 요청 승인을 정지시키지만, 이미 처리 중인 요청은 완료됩니다. 변경된 구성 파일을 재로드한 후, 프록시가 다시 클라이언트 요청을 승인하기 시작합니다.

주: 구성 및 관리 양식을 사용하거나 `ibmproxy.conf` 파일을 편집하여 특정 지시문을 변경하면, 서버를 재시작하는 대신 완전히 정지시킨 다음 시작해야 변경사항을 적용할 수 있습니다. 이 지시문은 185 페이지의 표 6에 나열되어 있습니다.

관리자 암호 설정

Caching Proxy 패키지를 설치하였으면 구성 및 관리 양식에 액세스하기 위한 관리자 ID 및 암호를 작성해야 합니다. 기본 프록시 서버 구성은 `webadmin.passwd` 암호 파일(Linux 및 UNIX 시스템의 `/opt/ibm/edge/cp/server_root/protect/` 디렉토리 또는 Windows 시스템의 `\Program Files\IBM\edge\cp\etc\` 디렉토리에 있음)을 사용하여 구성 및 관리 양식을 요청하는 사용자를 인증합니다. 패키지 설치의 기존 `webadmin.passwd` 파일에 겹쳐쓰기되지 않습니다.

다음 명령을 사용하여 `webadmin.passwd` 파일에 관리 항목을 추가하십시오.

- Linux 및 UNIX 시스템의 경우:

```
# htadm -adduser /opt/ibm/edge/cp/server_root/protect/webadmin.passwd
```

프롬프트가 표시되면 **htadm** 프로그램에서 사용자 이름, 암호 및 관리자의 실제 이름을 입력합니다.

- Windows 시스템의 경우:

```
cd "\Program Files\IBM\edge\cp\server_root\protect\"
htadm -adduser webadmin.passwd"
```

프롬프트가 표시되면 **htadm** 프로그램에 사용자 이름, 암호 및 관리자의 실제 이름을 입력합니다.

주: 관리자 사용자 이름 및 암호는 운영 체제가 대소문자를 구분하지 않더라도 대소문자를 구분합니다. 구성 및 관리 양식에 액세스할 때, **htadm** 명령을 사용하여 정확한 사용자 이름 및 암호를 입력하십시오.

htadm 명령의 자세한 설명에 대해서는 178 페이지의 『**htadm** 명령』을 참조하십시오.

제 3 장 구성 마법사 사용

Caching Proxy 구성 마법사를 사용하면 설치된 Caching Proxy를 신속하게 구성할 수 있습니다. 이 프로그램은 Caching Proxy 작동이 대리로 기능하도록 수정하는 데 필요한 필수 지시문만을 설정합니다. 프록시 서버는 추가 구성을 필요로 합니다.

Caching Proxy 구성 마법사를 사용하려면 다음을 수행하십시오.

1. 구성 마법사를 시작하십시오.

Windows 시스템의 경우: 시작 -> 프로그램 -> IBM WebSphere -> Edge Components -> Caching Proxy -> 구성 마법사를 누르십시오.

Linux 및 UNIX 시스템의 경우: `/opt/ibm/edge/cp/cpwizard/cpwizard.sh` 명령을 입력하십시오.

2. 프록시 서버가 HTTP 요청을 인식할 네트워크 포트를 선택하십시오.
3. 대상 콘텐츠 서버의 이름을 입력하십시오.
4. 프록시 서버 관리자의 사용자 ID 및 암호를 입력하십시오.

주:

1. 구성 마법사는 다음과 같은 지시문을 설정합니다.

```
Port port
Proxy /* http://content server :port
```

2. 구성 마법사를 사용하여 프록시 서버를 구성한 다음 SSL을 사용 가능하게 하려면, 포트 443에서 수신한 프록시 요청에 대해 맵핑 규칙을 작성해야 합니다. 자세한 정보는 44 페이지의 『맵핑 규칙 정의』를 참조하십시오.

예를 들면, 다음과 같습니다.

```
Proxy /* http://content server :443
```

또는

```
Proxy /* https://content server :443
```

Linux 시스템의 제한사항: 키보드 단축키가 Caching Proxy 구성 마법사에서 작동하지 않습니다.

제 4 장 ibmproxy.conf 파일 직접 편집

Caching Proxy는 관리 및 구성 양식을 통해 수동으로 구성되고, ibmproxy 구성 파일을 편집할 수 있습니다.

- Linux 및 UNIX 시스템에서 ibmproxy.conf 파일은 /etc/ 디렉토리에 있습니다.
- Windows 시스템에서 ibmproxy.conf 파일은 C:\Program Files\IBM\edge\cp\etc\en_US\에 있습니다.

구성 파일은 지시문이라고 하는 명령문으로 구성됩니다. 구성을 변경하려면, 지시문을 수정하여 구성 파일을 편집한 다음, 변경사항을 저장하십시오. emacs 및 vi와 같은 구성 파일을 편집하기 위해 대부분의 텍스트 편집기를 사용할 수 있습니다.

주: Solaris CDE(Common Desktop Environment)에 포함된 텍스트 파일 편집기는 사용하지 마십시오. Solaris 편집기는 때로는 파일 자체의 그룹을 수정하고 파일 연결의 등록 정보를 변경하므로, 구성 및 관리 양식은 구성 파일에 기록할 수 없습니다.

185 페이지의 표 6에서 식별된 지시문 중 하나를 변경하지 않은 경우, 서버를 재시작해야 구성 파일에 대한 변경사항이 적용됩니다. 위 목록의 지시문을 변경한 경우에는 서버를 정지시킨 다음 재시작해야 합니다. 명령에 대해서는 15 페이지의 제 5 장 『Caching Proxy 시작 및 정지』를 참조하십시오.

185 페이지의 부록 B 『구성 파일 지시문』에서는 각 구성 파일 지시문을 설명하고 구문에 관한 자세한 내용을 제공합니다.

제 5 장 Caching Proxy 시작 및 정지

Caching Proxy는 운영자의 개입을 최소화하면서 연속적으로 백그라운드 프로세스로서 실행되도록 설계되었습니다. 일반적으로 프록시 서버는 시스템 부팅 주기 중에 시작하며 유지보수가 필요한 경우에만 정지됩니다. 필요한 경우, 프록시 서버를 직접 시작할 수 있습니다. 프록시 서버에 재시작 명령을 전달할 수도 있습니다. 그러면 활성 클라이언트 연결에 영향을 미치지 않고 프록시 서버를 효율적으로 정지한 후 시작할 수 있습니다.

Linux 및 UNIX 시스템에서의 자동 시동 및 시스템 종료

Linux 및 UNIX 시스템의 경우, **ibmproxy** 초기설정 스크립트 및 관련된 기호 링크는 Caching Proxy가 설치될 때 적절한 `/etc/` 디렉토리에 위치합니다. 그런 다음 이 스크립트는 운영 체제의 시동 및 시스템 종료 루틴에 통합됩니다. **ibmproxy** 스크립트를 편집하고 **ibmproxy** 명령 옵션을 변경하여 자동 재시작에 대한 구성 설정을 변경할 수 있습니다.

주: Solaris 파일 설명자 한계

Caching Proxy 초기설정 스크립트는 Solaris 시스템 차원의 파일 설명자에 대한 한계 때문에 원하는 최대수의 파일 설명자를 설정하지 못할 수도 있습니다. 시스템 차원의 최대수가 Caching Proxy 초기설정 스크립트의 설정보다 작으면 시스템 차원의 한계가 사용됩니다. 값이 너무 작으면(1024 미만) 발생할 수 있는 프록시 성능 문제를 피하기 위해 파일 설명자 한계를 변경할 수 있습니다. **ulimit** 명령을 발행하여 현재 사용할 수 있는 설명자의 수를 검토하십시오. 값이 1024 미만이면 파일 설명자 한계를 늘리십시오. 파일 설명자 한계를 1024로 늘리려면 `/etc/system` 파일에 다음 행을 추가하십시오.

```
set rlim_fd_cur=0x400
```

자동 시동 및 시스템 종료 사용 불가능

자동 시동 및 시스템 종료를 사용 불가능하게 하려면 다음을 수행하십시오.

- AIX 시스템에서는 초기설정 파일에서 **ibmproxy** 명령을 제거하십시오.
- HP-UX 시스템에서는 **ibmproxy**에 대한 다음 연결을 제거하십시오.
 - `/sbin/rc1.d/K154ibmproxy`
 - `/sbin/rc2.d/S880ibmproxy`
- Linux 시스템의 경우, 실행 레벨 하위 디렉토리의 `/etc/rc.d/init.d/ibmproxy`에 대한 기호 연결을 제거하십시오.

SUSE Linux의 경우, **ibmproxy**에 대한 다음 연결을 제거하십시오.

- /etc/rc.d/rc3.d/S20ibmproxy
- /etc/rc.d/rc3.d/K20ibmproxy
- /etc/rc.d/rc4.d/S20ibmproxy
- /etc/rc.d/rc4.d/K20ibmproxy
- /etc/rc.d/rc5.d/S20ibmproxy
- /etc/rc.d/rc5.d/K20ibmproxy

Red Hat Linux의 경우, **ibmproxy**에 대한 다음의 연결을 제거하십시오.

- /etc/rc.d/rc0.d/K54ibmproxy
- /etc/rc.d/rc1.d/K54ibmproxy
- /etc/rc.d/rc2.d/K54ibmproxy
- /etc/rc.d/rc6.d/K54ibmproxy
- /etc/rc.d/rc3.d/S88ibmproxy
- /etc/rc.d/rc5.d/S88ibmproxy

- Solaris 시스템의 경우, **ibmproxy start** 명령과 다음 두 개의 종료(kill) 스크립트를 제거하십시오.
 - S88ibmproxy를 /etc/rc2.d 디렉토리에서 삭제하십시오.
 - K54ibmproxy를 /etc/rc0.d 디렉토리에서 삭제하십시오.
 - K54ibmproxy를 /etc/rc1.d 디렉토리에서 삭제하십시오.

Linux 및 UNIX 시스템에서의 수동 시동

시동 메소드에 관계 없이 명령 프롬프트 또는 스크립트 내에서 **ibmproxy** 명령을 호출할 수 있습니다. **ibmproxy** 명령의 자세한 설명에 대해서는 183 페이지의 『ibmproxy 명령』을 참조하십시오. 가장 많이 사용되는 인수를 예로 들면 다음과 같습니다.

AIX의 경우:

- 기본 로케일의 프록시 서버를 시작하려면 **startsrc** 명령을 사용하여 다음을 입력하십시오.

```
startsrc -s ibmproxy
```

- 기본이 아닌 다른 로케일의 프록시 서버를 시작하려면 **startsrc** 명령을 사용하여 다음을 입력하십시오.

```
startsrc -s ibmproxy -e "LC_ALL=locale"
```

- 기본 런타임 설정으로 프록시 서버를 시작하려면 **startsrc** 명령을 사용하지 않고 다음을 입력하십시오.

```
ibmproxy
```

HP-UX의 경우:

- 초기설정 스크립트를 실행하여 프록시 서버를 시작하려면 다음을 루트 프롬프트에 입력하십시오.

```
/sbin/init.d/ibmproxy start
```

- 초기설정 스크립트를 실행하지 않고 프록시 서버를 시작하려면 루트 프롬프트에서 다음 사항을 입력하십시오.

```
/usr/sbin/ibmproxy
```

- 초기설정 스크립트를 실행하지 않고 프록시 서버를 시작하려면 루트 프롬프트에서 다음 사항을 입력하십시오.

```
/usr/sbin/ibmproxy -nobg
```

Linux의 경우

- 초기설정 스크립트를 실행하여 프록시 서버를 시작하려면 다음을 루트 프롬프트에 입력하십시오.

```
/etc/rc.d/init.d/ibmproxy start
```

- 초기설정 스크립트를 실행하지 않고 프록시 서버를 시작하려면 루트 프롬프트에서 다음 사항을 입력하십시오.

```
/usr/sbin/ibmproxy
```

- 초기설정 스크립트를 실행하지 않고 프록시 서버를 시작하려면 루트 프롬프트에서 다음 사항을 입력하십시오.

```
/usr/sbin/ibmproxy -nobg
```

- 기존의 SQUID 구성 파일(squidConfig.file)을 사용하여 프록시 서버를 시작하려면 루트 프롬프트에서 다음 명령을 입력하십시오.

```
squidConfig.file -r /etc/errors_icons.conf
```

errors_icons.conf 파일은 디렉토리를 찾아보기할 때 지정된 파일 유형에 사용할 아이콘을 식별합니다.

Solaris의 경우

- 초기설정 스크립트를 실행하여 프록시 서버를 시작하려면 다음을 루트 프롬프트에 입력하십시오.

```
/etc/init.d/ibmproxy start
```

- 초기설정 스크립트를 실행하지 않고 프록시 서버를 시작하려면 루트 프롬프트에서 다음 사항을 입력하십시오.

```
/usr/sbin/ibmproxy
```

- 초기설정 스크립트를 실행하지 않고 프록시 서버를 시작하려면 루트 프롬프트에서 다음 사항을 입력하십시오.

Windows 서비스로 시동

Caching Proxy가 Windows 서비스로 설치되는 경우, 이는 다른 모든 Windows 서비스와 같이 시작됩니다.

1. 시작 -> 설정(Windows 2000의 경우) -> 제어판을 누르십시오.
2. 제어판 창에서 관리 도구 -> 서비스를 두 번 누르십시오.
3. 서비스 창에서 **Caching Proxy**를 강조표시하십시오.
4. 시작을 눌러 Caching Proxy 서비스를 시작하십시오.

Caching Proxy가 서비스로서 설치되는 경우, 이는 Windows가 시작될 때 자동으로 시작하도록 구성될 수 있습니다. 이 경우에는 프록시가 요청에 응답할 수 있기 전에 로그인할 필요가 없습니다. 프록시를 자동으로 시작하려면 다음을 수행하십시오.

1. 시작 -> 설정(Windows 2000의 경우) -> 제어판을 누르십시오.
2. 제어판 창에서 관리 도구 -> 서비스를 두 번 누르십시오.
3. 서비스 창에서 **Caching Proxy**를 강조표시하십시오.
4. 자동 단일 선택 단추를 누른 후에 시작을 눌러서 Windows가 시작될 때 Caching Proxy 서비스를 자동으로 시작하십시오.

PATH 환경 변수 새로 고침

Caching Proxy가 서비스 창에 시작됨으로 표시되어 있으나 프록시가 작동하지 않는 경우에는 프록시 설치 후 시스템을 재시작하지 않은 것입니다. Caching Proxy 서비스가 데스크탑과 상호작용하도록 설정된 경우, 재시작하지 않으면 팝업 상자에 다음의 오류 메시지가 표시될 수도 있습니다.

Message catalog error: the message catalog could not be loaded or is invalid

PATH 환경 변수의 값이 Windows 레지스트리에서 새로 고쳐지도록 시스템을 다시 시작해야 합니다. 레지스트리가 새로 고쳐지지 않은 경우, PATH 변수가 올바른 Caching Proxy 및 GSK7 경로를 표시하지만 올바르게 작동하지 않을 수 있습니다.

주: Caching Proxy 및 다른 응용프로그램(예: 네트워크 파일 시스템)이 모두 서비스로서 실행되는 경우에는 Windows 시스템에 대해 잠재적 충돌이 존재합니다. Caching Proxy는 때로, 서비스로 또한 실행되고 있는 파일 시스템 응용프로그램이 소유하는 원격 드라이브를 포함하는 경로를 해석할 수 없습니다.

파일 시스템 서비스에 대한 경로가 Windows PATH 환경 변수의 Caching Proxy 서비스에 대한 경로 이전에 나타나는 경우에는 문제점이 발생할 수 있습니다. 설정 종료 근처에 파일 시스템 서비스가 위치하도록 PATH 변수를 변경하면 이 문제점을 해결할 수 있습니다.

이 문제점은 Windows 서비스로 실행되지 않는 응용프로그램이 제어하는 원격 드라이브에 영향을 주지 않습니다. 예를 들어, Caching Proxy는 LAN(Local Area Network)을 통해 가시적인 다른 Windows 시스템의 공유 드라이브를 액세스할 수 있습니다.

Windows 응용프로그램으로 시동

시작 메뉴 사용

Caching Proxy가 Windows 응용프로그램으로 설치된 경우, 설치 프로시저는 **Caching Proxy** 항목을 시작 메뉴의 서브메뉴로 작성합니다. Caching Proxy를 응용프로그램으로 시작하는 경우, 시작 -> 프로그램 -> IBM WebSphere -> Edge Components -> **Caching Proxy**를 누르십시오.

시동 프로시저는 현재 구성 설정을 사용하여 프록시 서버를 실행합니다. 시동 시간에 다른 설정을 지정하려면 명령 시동 프로시저(다음 섹션 참조)를 사용하십시오.

명령 프롬프트 사용

임의의 Windows 또는 DOS 명령 프롬프트에서 서버를 시작하려면 **ibmproxy** 명령을 사용하십시오. 서버를 설치한 이후에 Windows를 시스템 종료하고 재시작하지 않은 경우, 다음과 같이 이 명령에 대한 전체 경로 이름을 입력하십시오(기본값으로).

```
c:\Program Files\IBM\edge\cp\bin\ibmproxy.exe
```

ibmproxy 명령은 현재 구성 설정으로 서버를 시작합니다. 설치 이후 서버 구성을 변경하지 않으면, 현재 구성은 설치 중에 입력한 정보와 기본 옵션을 기반으로 합니다.

ibmproxy 명령을 이용하여 Caching Proxy가 서비스로 실행하도록 설치한 경우에도 서버를 응용프로그램으로 시작할 수 있습니다. 서버를 강제로 응용프로그램으로 실행하도록 하려면, **-noservice** 명령 옵션도 지정해야 합니다. 기타 명령 옵션이 실시간으로 구성 설정을 변경합니다.

복수 프록시 서버 시작

프록시 서버의 여러 인스턴스는 동시에 실행할 수 있지만, 각 인스턴스는 다른 포트에서 대기해야 합니다. AIX 시스템에서는 단지 하나의 인스턴스만 SRC(System Resource Controller)로 시작할 수 있습니다. 구성 파일은 특정 시스템의 서버마다 달라야 하는 포트 번호를 식별하기 때문에, 모든 서버 인스턴스에 고유한 구성 파일을 지정해야 합니다. 추가 서버 인스턴스를 시작하려면(최소 한 개의 인스턴스를 이미 실행하고 있는 경우) 다음 명령을 입력하십시오.

- Linux 및 UNIX의 경우:

```
ibmproxy -r other_config_file
```

- Windows의 경우:

```
ibmproxy -noservice -r other_config_file
```

여기서 *other_config_file*은 고유한 구성 파일입니다.

서버의 여러 인스턴스를 시작할 때, 각 인스턴스에 표시되는 프로세스 ID를 기록하십시오. 프로세스 ID는 서버의 고유 인스턴스를 정지하는 데 필요합니다.

주: 서버의 여러 인스턴스를 실행하는 Linux 시스템의 경우, **/etc/rc.d/init.d/ibmproxy stop** 명령은 마지막으로 시작된 서버만을 정지시킵니다. 다른 인스턴스는 각각 따로 정지되어야 합니다. 『Linux 및 UNIX 시스템 수동 시스템 종료』에서 관련 정보를 참조하십시오.

Linux 및 UNIX 시스템 수동 시스템 종료

서버를 정지시키려면 다음을 수행하십시오.

- 프로세스를 시작한 사용자거나 수퍼유저인 루트여야 합니다.
- 서버를 시작한 것과 동일한 방법을 사용해야 합니다. 다음 테이블은 시작 방법 및 이와 연관된 정지 방법을 나열합니다.

표 2. Linux 및 UNIX 시스템의 시작 및 정지 방법

시작 방법	정지 방법
/etc/inittab에서(AIX의 경우)	stopsrc -s ibmproxy 입력
/sbin/init.d에서(HP-UX의 경우)	/sbin/init.d/ibmproxy stop 입력
Linux의 경우, /etc/rc.d/init.d에서	/etc/rc.d/init.d/ibmproxy stop 입력

표 2. Linux 및 UNIX 시스템의 시작 및 정지 방법 (계속)

ibmproxy	<ol style="list-style-type: none"> 1. ibmproxy 프로세스 ID 찾기: AIX에서는 <code>ps -aef grep "ibmproxy"</code>를 입력하십시오. Linux의 경우, <code>ps -aux grep ibmproxy grep server_ID</code> 입력. Solaris 및 HP-UX에서는 <code>ps -ef grep "ibmproxy"</code>를 입력하십시오. 2. ibmproxy 프로세스 정지: <code>kill process_id</code> 입력 <p>이 시스템의 모든 서버를 정지시키려면 <code>killall ibmproxy</code>를 입력하십시오.</p>
ibmproxy -nbg	ctrl-c 입력
ibmproxy -r -other_config_file(AIX의 경우)	stopsrc -s ibmproxy -p process_id 입력
Linux의 경우, ibmproxy -r -other_config_file	<ol style="list-style-type: none"> 1. ibmproxy 프로세스 ID 찾기: <code>ps aux grep ibmproxy grep process_id</code> 입력 2. ibmproxy 프로세스 정지: <code>kill process_id</code> 입력

주: 투명 프록시를 시작했을 경우, Caching Proxy 서버를 정지한 후 투명 프록시 커널 확장자 및 연관 방화벽 규칙을 로드 해제해야 합니다. 루트로서 다음 명령을 입력하십시오.

```
ibmproxy -unload
```

루트 프롬프트에서 서버를 정지하려면 다음을 입력하십시오.

- AIX의 경우: `stopsrc -s ibmproxy`
- HP-UX의 경우: `/sbin/init.d/ibmproxy stop`
- Linux의 경우: `/etc/rc.d/init.d/ibmproxy stop`
- Solaris의 경우: `/etc/init.d/ibmproxy stop`

시스템 종료 명령의 한계

시스템 종료 명령 사용 시 다음 한계를 경험할 수 있습니다.

- **AIX, HP-UX 및 Linux**

AIX, HP-UX 및 Linux 시스템의 경우, Caching Proxy 시스템을 정지하기 위한 명령은 종종 Caching Proxy 프로세스만 시스템 종료합니다. 이러한 작동을 야기하는 AIX 명령은 `stopsrc -s ibmproxy` 명령입니다. 이러한 작동을 야기하는 HP-UX 및 Linux 명령은 `ibmproxy -stop` 명령입니다.

LDAP 서버가 사용하는 PACD 프로세스는 프록시 서버 종료 후에도 실행 중일 수 있습니다. PACD 프로세스는 다음과 같이 **kill** 명령을 사용하여 안전하게 종료할 수 있습니다.

```
kill -15 PACD_process_ID
```

- **Solaris**

Solaris 시스템에서 **ibmproxy -stop** 명령의 발행 결과는 다른 운영 체제에서의 결과와 동일하지 않습니다. Solaris 코드 상의 한계로 인해 Solaris 플랫폼에서 **ibmproxy -stop**을 사용할 때는 서버 종료 플러그인 단계가 실행되지 않습니다.

이 한계는 고객 구현 플러그인 뿐만 아니라 프록시 서버 소프트웨어에서도 유효합니다.

종료 후 프록시 서버의 계속 실행은, LDAP 서버가 사용하는 PACD 프로세스의 경우에도 가능합니다. PACD 프로세스는 다음과 같이 **kill** 명령을 사용하여 안전하게 종료할 수 있습니다.

```
kill -15 PACD_process_ID
```

Windows 시스템에서의 수동 시스템 종료

다른 Windows 프로그램을 정지하는 방법과 동일한 방법으로 Caching Proxy 서버를 정지할 수 있습니다.

프록시가 서비스로 설치된 경우,

1. 시작 -> 설정(Windows 2000의 경우) -> 제어판을 누르십시오.
2. 제어판 창에서 관리 도구 -> 서비스를 두 번 누르십시오.
3. 서비스 창에서 **Caching Proxy**를 강조표시하십시오.
4. 정지를 눌러 Caching Proxy 서비스를 정지하십시오.

프록시를 서비스로 설치하지 않은 경우에는 다음 중 하나를 수행하여 Caching Proxy를 정지하십시오.

- 오른쪽 위 모서리의 **x** 아이콘을 누르십시오.
- 파일 메뉴에서 나감을 누르십시오.
- **Alt + F4**를 누르십시오.

구성 변경 후 재시작

구성 및 관리 양식을 사용하거나 `ibmproxy.conf` 파일을 편집하여 서버 구성을 변경한 후, 변경사항을 적용하기 위해서는 서버를 재시작해야 합니다. 대부분의 경우, 먼저 서버를 정지하지 않고 재시작할 수 있습니다. 그러나 일부 설정은 단순히 재시작하는 것으로 새로 고쳐지지 않는 경우가 있습니다. 자세한 정보는 185 페이지의 표 6을 참조하십시오.

먼저 정지하지 않고 서버를 재시작하려면 임의의 구성 및 관리 양식에서 **재시작** 단추를 누르거나 `ibmproxy -restart`를 입력하십시오.

제 2 부 Caching Proxy 프로세스 구성 및 조정

이 파트에서는 Caching Proxy 컴포넌트가 운영 체제, 컴퓨터 하드웨어 및 네트워크와 상호 교류하는 방법에 대해 설명합니다. 또한 이 상호 교류를 구성하기 위한 프로시저도 제공합니다. 이 프록시 서버 구성 요소는 일반적으로 시스템 관리자가 관리하며 시스템 자원(예: 사용 가능한 메모리 및 CPU 주기)뿐만 아니라 네트워크 자원(예: IP 주소 및 호스트 이름)에 맞도록 주의해서 조정되어야 합니다.

이 파트에는 다음과 같은 장이 들어 있습니다.

27 페이지의 제 6 장 『서버 정의』

29 페이지의 제 7 장 『프로세스 소유권 설정』

31 페이지의 제 8 장 『연결 관리』

35 페이지의 제 9 장 『프록시 서버 프로세서 조정』

제 6 장 서버 정의

Caching Proxy는 일반적으로 네트워크 서버로 수행하도록 구성된 호스트 컴퓨터 시스템에서 백그라운드 프로세스로 실행됩니다. 이 프로세스는 호스트 컴퓨터 시스템의 하나 또는 모든 활성 IP(Internet Protocol) 주소에 연관(바인드)되어 있습니다. 이 프로세스는 지정된 포트에서 여러 IP(예: FTP 또는 HTTP)를 인식하여 작동 구성에 따라 이 요청에 대한 조치를 수행합니다. (자세한 정보는 39 페이지의 제 3 부 『Caching Proxy 작동 구성』을 참조하십시오.)

기본적으로 Caching Proxy는 호스트 컴퓨터 시스템의 이름을 사용합니다. 신중하게 프록시 서버의 호스트 이름을 지정하여 이 기본 작동을 덮어쓸 수 있습니다. Caching Proxy를 특정 IP 주소에 바인드하려면 프록시 서버의 호스트 이름을 해당 IP 주소와 동일하게 변경해야 합니다.

주: 프록시 서버가 IP 주소에 바인드하고자 할 때 호스트 이름을 사용 가능한 IP 주소로 설정하지 않은 경우, 바인드는 실패하며 프록시 서버는 모든 사용 가능한 IP 주소를 인식합니다.

프록시 서버의 호스트 이름은 클라이언트 통신을 해석하는 방법에 영향을 미치지 않습니다. 프록시 서버는 자신의 호스트 이름을 HTTP 요청 헤더에 있는 호스트 이름 인수의 값과 비교하지 않습니다. 간혹 프록시 서버의 호스트 이름은 동적으로 생성된 로컬 콘텐츠 페이지(예: 오류 메시지)에 통합됩니다. 이는 HTTP 헤더의 Via 인수값으로서 요청된 클라이언트로 다시 전달될 수 있습니다.

요청을 대상 서버로 전달하기 전에 요청 클라이언트의 호스트 이름을 프록시 서버의 호스트 이름으로 바꾸도록 프록시 서버를 구성할 수 있습니다. 이렇게 하면 대상 서버가 클라이언트와의 직접 연결을 설정하지 않고도 프록시 서버를 통해 통신 채널을 유지할 수 있도록 합니다.

ServerRoot, Hostname 및 port 지시문에 대한 값으로서 호스트 컴퓨터 시스템에 있는 프록시 서버 파일의 실제 위치, 프록시 서버가 자신을 참조할 때 사용하는 이름, 인식하는 포트를 지정하여 프록시 서버 프로세스를 정의하십시오. 호스트에 여러 개의 IP 주소가 있으면 BindSpecific 지시문의 값을 설정하고 Hostname 지시문의 값을 IP 주소와 동일하게 설정하여 특정 주소로 프록시 서버를 바인드할 수 있습니다.

관리 포트는 구성 및 관리 양식에 액세스하고 서버를 유지하는 메소드를 제공합니다. 관리 포트를 통해 프록시 서버에 대한 액세스를 제공하려면 AdminPort 지시문의 값을 지정하십시오. 관리 포트에서 받은 요청은 표준 포트에서 받은 요청과 함께 대기열에 넣여지지 않습니다. 이 포트를 통해 구성 및 관리 양식에 액세스할 수 있게 하는 맵핑 규칙을 기록할 수 있습니다.

BindSpecific 지시문이 사용 가능한 경우, Caching Proxy는 Hostname 지시문의 값에서 파생된 IP 주소와 함께 Port 지시문으로 지정된 포트로 바인드됩니다. AdminPort 지시문으로 지정된 포트는 시스템에서 사용 가능한 모든 IP 주소로 바인드됩니다.

실행 중인 서버의 기본 이름(예: IBM-PROXY 또는 IBM_HTTP_SERVER)을 덮어쓰려면 HeaderServerName 지시문에 대한 값을 지정하십시오. 이 값은 HTTP 응답 서버 필드를 채웁니다.

프록시 성능을 향상시키기 위해 PureProxy 지시문의 값을 온(On)으로 설정할 수 있습니다. 이는 모든 캐시 기능을 완전히 사용 불가능하게 합니다.

연관된 지시문

다음 지시문은 프록시 서버 프로세스를 정의합니다.

- 239 페이지의 『Hostname — 서버에 대한 정식 도메인 이름 또는 IP 주소 지정』
- 299 페이지의 『ServerRoot — 서버 프로그램이 설치될 디렉토리 지정』
- 238 페이지의 『HeaderServerName — HTTP 헤더에 리턴되는 프록시 서버 이름 지정』
- 201 페이지의 『BindSpecific — 서버가 하나 또는 모든 IP 주소로 바인드 여부 지정』
- 272 페이지의 『Port — 서버가 요청을 인식하는 포트 지정』
- 197 페이지의 『AdminPort — 관리 페이지나 양식을 요청하기 위한 포트 지정』
- 288 페이지의 『PureProxy — 전용 프록시 사용 불가능』

자세한 정보는 13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』을 참조하십시오.

구성 및 관리 양식

다음 구성 및 관리 양식은 연관된 지시문의 값을 편집합니다.

- 서버 구성 -> 기본 설정값 -> 호스트 이름
- 서버 구성 -> 기본 설정값 -> 서버 루트
- 서버 구성 -> 기본 설정값 -> 기본 포트 번호
- 서버 구성 -> 기본 설정값 -> 관리 포트 번호
- 서버 구성 -> 기본 설정값 -> 바인드 옵션
- 프록시 구성 -> 프록시 성능 -> 순수 프록시로 실행

주: 구성 및 관리 양식을 사용하여 HeaderServerName 지시문을 편집할 수 없습니다.

자세한 정보는 7 페이지의 제 2 장 『구성 및 관리 양식 사용법』을 참조하십시오.

제 7 장 프로세스 소유권 설정

슈퍼유저 루트가 아닌 사용자가 Caching Proxy를 시작할 경우 해당 사용자가 프록시 서버에 연관된 모든 프로세스의 소유권을 유지합니다. 그러나 슈퍼유저 루트가 Caching Proxy를 시작할 경우, 프록시 서버의 사용자 ID 설정 기능이 ibmproxy.conf 파일의 UserId 및 GroupId 지시문을 읽어 프로세스 소유권을 지정된 사용자 및 그룹으로 변경합니다. 이는 파일 액세스를 제한하고 컴퓨터 시스템을 보호하기 위해 수행됩니다. UserId 또는 GroupId 지시문을 변경할 경우, 프록시 서버가 사용하는 로그 디렉토리 및 기타 파일(예: ACL(액세스 제어 목록))에 대한 소유권 및 사용 권한을 갱신해야 합니다.

UserID, GroupID 및 PidFile 지시문의 값으로서 사용자 ID, 그룹 ID 및 프로세스 ID를 기록하는 파일의 위치를 지정하여 프록시 서버 프로세스의 소유권을 설정하십시오.

강제로 프록시 서버 프로세스가 포그라운드 프로세스로 실행되게 하려면 NoBG 지시문의 값을 온(On)으로 설정하십시오.

Linux 시스템의 경우:

Linux 시스템에서는 연결 인식을 담당하는 프로세스 및 스레드의 소유권만을 변경합니다. 워크플로우 내의 다른 활동을 담당하는 프로세스 및 스레드는 계속 루트가 소유합니다. 모든 프로세스 및 스레드는 PID(프로세스 ID) 번호를 받습니다. **ps** 명령은 프로세스 또는 스레드와의 연관 여부에 관계 없이 모든 프로세스 ID를 나열합니다.

주: 일부 Linux 커널에서, Caching Proxy는 오류 로그에 다음 오류 메시지를 생성합니다.

```
Cannot init groups for user nobody, errno: 1
```

Caching Proxy의 정상적인 운영에 영향을 미치지 않으므로, 오류 메시지를 무시할 수 있습니다. Caching Proxy를 시작하기 전에 다음 환경 변수를 내보내서, 오류 메시지를 방지할 수 있는 해결책이 있습니다.

```
export RPM_FORCE_NPTL=1
export LD_ASSUME_KERNEL=2.4.19:
```

연관된 지시문

다음 지시문은 프록시 서버 프로세스 소유권을 설정합니다.

- 312 페이지의 『UserId — 기본 사용자 ID 지정』
- 238 페이지의 『GroupId — 그룹 ID 지정』
- 262 페이지의 『NoBG — Caching Proxy 프로세스를 포그라운드에 실행』

- 270 페이지의 『PidFile(Linux 및 UNIX 전용) — Caching Proxy의 프로세스 ID를 저장할 파일 지정』

자세한 정보는 13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』을 참조하십시오.

구성 및 관리 양식

다음 구성 및 관리 양식은 연관된 지시문의 값을 편집합니다.

- 서버 구성 -> 기본 설정값 -> 사용자 ID
- 서버 구성 -> 기본 설정값 -> 그룹 ID
- 서버 구성 -> 기본 설정값 -> 프로세스 ID 파일 위치

주: 구성 및 관리 양식을 사용하여 NoBG 지시문을 편집할 수 없습니다.

자세한 정보는 7 페이지의 제 2 장 『구성 및 관리 양식 사용법』을 참조하십시오.

제 8 장 연결 관리

Caching Proxy는 각 클라이언트 요청을 처리하기 위해 새 스레드를 생성합니다. 사용 가능한 스레드가 없으면, 프록시 서버는 더 많은 스레드가 사용 가능해질 때까지 요청을 보유합니다. 활성 스레드의 수가 증가함에 따라 프록시 서버는 더 많은 메모리를 소비합니다. MaxActiveThreads 지시문의 값에 최대 활성 스레드 수를 지정하십시오.

인식 백로그는 서버가 새 클라이언트와의 연결을 거부하기 전에 서버가 로그하는 클라이언트 연결에 대해 보류 중인 요청 수입니다. 이 설정은 몇 초 안에 서버가 처리할 수 있는 요청의 수에 기초해야 합니다. 서버는 클라이언트 연결 시간 종료 전에 응답해야 합니다. ListenBacklog 지시문에 대한 값으로 백로그에 보유할 수 있는 최대 연결 수를 지정하십시오.

프록시 서버는 지속적인 클라이언트/서버 연결을 유지할 수 있습니다. 서버는 지속적인 연결로 클라이언트의 여러 요청을 승인하여, 동일한 TCP/IP 연결을 통해 응답을 전송합니다. 지속적인 연결을 사용하면, 클라이언트의 대기 시간이 줄어들고 프록시 서버의 CPU 로드도 줄어들며 서버 메모리의 증가량이 적으므로 비용이 줄어듭니다. 서버가 각 요청 및 응답에 개별 TCP/IP 연결을 설정하지 않으면 전체 처리량이 증가하고, 연결을 지속하면 TCP/IP 연결을 최대 효율로 사용할 수 있습니다.

서버측 연결 풀링은 프록시 서버와 기점 서버 간 기존의 연결을 재사용할 수 있도록 하여 서버측에 지속적인 연결의 이점을 제공합니다. 각각의 재사용된 연결은 세 개의 TCP 패킷을 저장합니다(두 개는 연결을 설정하기 위한 세 방향 핸드셰이크 패킷이고, 나머지 하나는 연결을 닫기 위한 패킷). 서버측 연결 풀링의 장점은 다음과 같습니다.

- (연결의 열기 및 닫기를 최소화하여) 네트워크 연결 정체가 적습니다.
- 라우터, 클라이언트 및 서버에 사용된 CPU 시간이 적습니다.
- 클라이언트 및 서버의 메모리 사용이 적습니다.
- (연결의 열기 및 닫기를 최소화하여) 캐시가 누락되어 프록시 응답이 빨라집니다.

주: 연결 풀링은 제어된 환경에서만 사용하는 것이 좋습니다. 기점 서버가 HTTP 1.1을 따르지 않는 경우에 성능이 저하될 수 있습니다. 또한 기점 서버를 올바르게 설정하는 것이 중요합니다. 다음은 Apache 1.3.19 구성 파일의 간단한 예제입니다.

- #KeepAlive: 지속적인 연결을 허용할지 여부(#연결당 하나 이상의 요청). Off로 설정하여 비활성화하십시오#
- KeepAlive On
- #MaxKeepAliveRequests: 지속적인 연결에서 허용되는 요청의 최대수 0으로 설정하여 무제한 수량을 허용하십시오. 최상의 성능을 위해 이 숫자를 높게 유지하십시오#

- Max KeepAliveRequests 0
- #KeepAliveTimeout: 동일한 연결에서 동일한 클라이언트의 다음 요청을 기다리는 초 수#
- KeepAliveTimeout 240

웹 서버를 사용 중인 경우, 이 설정으로 웹 서버에 대한 연결을 유지하여 기점 서버보다는 프록시가 연결을 관리할 수 있도록 합니다. 따라서 연결이 필요한 범위까지만 풀립니다.

서버측 연결 풀링이 사용 가능하면 기점 서버에 대한 HTTP 연결이 풀립니다. SSL 연결 또한 프록시의 SSLEnable 지시문을 커짐으로 설정하는 구성에 풀립니다.

1회 연결에서 서버당 보유할 최대 대기 소켓 수를 지정하여 연결 풀링을 유지하는 방법, 서버가 대기 중인 지속적인 연결을 종료하기 전에 기다리는 시간, 가비지 콜렉션 스레드가 시간이 지난 연결을 확인하는 시간 간격(기본값 2분)을 구성하십시오.

InputTimeout, OutputTimeout, PersistTimeout, ReadTimeout 및 ScriptTimeout 지시문에 대한 값으로 여러 연결이 열린 상태로 남아 있는 시간을 정의하십시오.

연관된 지시문

다음 지시문은 프록시 서버 프로세스와의 연결을 관리합니다.

- 256 페이지의 『MaxActiveThreads — 최대 활성 스레드 수 지정』
- 216 페이지의 『ConnThreads — 연결 관리에 사용되는 연결 스레드의 수를 지정』
- 249 페이지의 『ListenBacklog — 서버가 수행할 수 있는 인식 백로그 클라이언트 연결 수 지정』
- 286 페이지의 『ProxyPersistence — 지속적인 연결 허용』
- 258 페이지의 『MaxPersistRequest — 지속적인 연결에서 수신할 요청의 최대수 지정』
- 298 페이지의 『ServerConnPool — 기점 서버 연결 풀링 지정』
- 259 페이지의 『MaxSocketPerServer — 서버의 개방형 대기 소켓 최대수 지정』
- 298 페이지의 『ServerConnTimeout — 최대 비활성 기간 지정』
- 298 페이지의 『ServerConnGCRun — 가비지 콜렉션 스레드 실행 간격 지정』
- 269 페이지의 『PersistTimeout — 클라이언트가 다른 요청을 전송하기 위한 대기 시간 지정』
- 244 페이지의 『InputTimeout — 입력 시간 종료 지정』
- 290 페이지의 『ReadTimeout — 연결의 시간 한계 지정』
- 267 페이지의 『OutputTimeout — 출력 시간 종료 지정』
- 296 페이지의 『ScriptTimeout — 스크립트의 시간 종료 설정 지정』

자세한 정보는 13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』을 참조하십시오.

구성 및 관리 양식

다음 구성 및 관리 양식은 연관된 지시문의 값을 편집합니다.

- 서버 구성 -> **System Management** -> 성능 -> 최대 활성 스레드 수
- 서버 구성 -> **System Management** -> 성능 -> 인식 백로그 크기
- 프록시 구성 -> 프록시 성능 -> 지속적인 연결 허용
- 서버 구성 -> **System Management** -> 성능 -> 최대 요청 수
- 서버 구성 -> **System Management** -> 성능 -> 지속 시간 종료
- 서버 구성 -> **System Management** -> 시간 종료 -> 입력 시간 종료
- 서버 구성 -> **System Management** -> 시간 종료 -> 읽기 시간 종료
- 서버 구성 -> **System Management** -> 시간 종료 -> 출력 시간 종료
- 서버 구성 -> **System Management** -> 시간 종료 -> 스크립트 시간 종료
- 서버 구성 -> **System Management** -> 시간 종료 -> 지속 시간 종료

주:

1. 구성 및 관리 양식을 사용하여 ServerConnPool, MaxsocketPerServer, ServerConnTimeout 또는 ServerConnGCRun 지시문을 편집할 수 있습니다.
2. PersistTimeout은 서버 구성 -> **System Management** -> 성능 양식 또는 서버 구성 -> **System Management** -> 시간 종료 양식에서 편집할 수 있습니다.

자세한 정보는 7 페이지의 제 2 장 『구성 및 관리 양식 사용법』을 참조하십시오.

제 9 장 프록시 서버 프로세서 조정

시스템을 올바르게 설정 및 조정하여 Caching Proxy 성능을 현저히 향상시킬 수 있습니다. 설정 및 조정 향상을 위한 제안사항은 다음과 같습니다.

성능 관련 지시문 설정

다음 지시문은 프록시 서버 프로세스의 성능에 많은 영향을 미칩니다.

- 288 페이지의 『PureProxy — 전용 프록시 사용 불가능』. 이 기능은 캐시를 완전히 사용 불가능하게 함으로써 시스템 성능을 향상시킵니다.
- 286 페이지의 『ProxyPersistence — 지속적인 연결 허용』. 이 기능은 클라이언트 및 서버가 열린 연결을 유지할 수 있게 합니다. 지속적인 연결은 프록시 서버에서의 문서 요청에 대한 대기 시간을 감소시키지만, 각 연결에 대한 전용 스레드 및 증가된 네트워크 대역폭을 필요로 합니다. 설정이 사용 가능한 스레드의 수를 제한하면 지속적인 연결을 허용하지 마십시오.

다음 구성 및 관리 양식 필드는 연관된 지시문의 값을 편집합니다.

- 프록시 구성 -> 프록시 성능: 순수 프록시로 실행
- 프록시 구성 -> 프록시 성능: 지속적인 연결 허용

기타 응용프로그램 검토

시스템에서 실행 중인 서비스 또는 디먼을 검토하여 필요하지 않은 서비스나 디먼을 제거하여 사용 가능한 메모리 및 CPU 주기를 늘리십시오. 예를 들어, 시스템이 소수의 웹 페이지만 제공하는 웹 서버를 실행하고 있다면, Caching Proxy를 웹 서버로만 사용할 수 있습니다. 다른 웹 서버를 다음과 같이 사용 불가능하게 하십시오.

- AIX의 경우: /etc/inittab을 점검하십시오.
- Linux: /etc/rc.d/rcx.d에서 시스템의 기본 실행 레벨(일반적으로 2)을 점검하십시오.
- HP-UX 및 Solaris의 경우: /etc/rcx.d에서 시스템의 기본 실행 레벨(일반적으로 2)을 점검하십시오.
- Windows 시스템의 경우:
 1. 시작 -> 설정(Windows 2000의 경우) -> 제어판 -> 관리 도구 -> 서비스를 누르십시오.
 2. 검토 서비스는 꼭 필요하지는 않지만 자동으로 설정되어 있습니다.
 3. 이러한 서비스의 시동 유형을 자동에서 수동으로 변경하십시오.

페이징 영역 확인

올바른 조작에 필요한 만큼 페이징 공간이 충분한지 확인해 봅니다. 시스템에는 물리적 메모리 두 배의 페이징 공간이 필요합니다. 가능한 경우 복수의 물리적 드라이브에 페이징 영역을 분산하십시오. 예를 들어, 512MB의 메모리 및 5개의 SCSI 드라이브가 있는 Netfinity 5000 서버는 각각의 드라이브에 대략 200MB가 있는 1GB의 총 페이징 영역이 필요합니다.

파일 시스템 조정

Caching Proxy는 조작 중 많은 파일을 작성하고 파괴합니다. 프록시 서버가 액세스를 기록(액세스 로그, 프록시 액세스 로그 또는 캐시 액세스 로그 사용)하는 경우에는, 자체 파일 시스템으로 로그를 지정하여, 로그가 현저히 증가하는 경우 다른 기능(예: 캐시)을 위해 유보해 둔 공간을 사용하지 않도록 합니다.

TCP/IP 구성 조정

Caching Proxy는 TCP/IP 구성의 변화에 민감합니다. 운영 체제에서 TCP/IP 값을 낮추면 프록시 서버가 예상하지 않은 방법으로 수행될 수도 있습니다. 보다 특수하게 TCP/IP 값을 너무 낮게 설정한 경우, 프록시 서버에 연결하는 클라이언트에 의해 또는 프록시가 연결된 원래 서버에 의해 연결이 재설정될 수도 있습니다. 이는 낮은 대역폭 연결(56700bps 이하)을 통해 프록시 서버에 연결하는 클라이언트에 특히 적용됩니다. TCP/IP 매개변수의 값을 낮추어야 할 경우, 주의해서 처리하십시오.

상위 로드 환경을 위한 TCP 시간 대기 간격 조정(HP-UX, Linux, Solaris, Windows)

TCP 시간 대기 간격은 연결을 강제로 닫기 전에 소켓이 전송자로부터 FIN 패킷을 대기하는 시간을 지정합니다. 대용량 로드 환경에서, 연결을 닫은 후에 많은 수의 소켓이 TIME_WAIT 상태에서 일시중단된 경우 프록시 서버가 정지된 것처럼 보일 수도 있습니다. TCP 시간 대기 간격을 줄이면 일시중단된 소켓의 수가 감소되며, 대용량 로드 환경에서 프록시 서버가 정지된 것처럼 보이는 것을 방지할 수 있습니다. 이 간격을 5초로 설정하도록 권장합니다.

시간 대기 간격을 5초로 설정하려면

- HP-UX의 경우:

다음 명령을 발행하십시오.

```
ndd /dev/tcp -set tcp_time_wait_interval 5000
```

"sam" 유틸리티를 사용하여 커널 매개변수 max_thread_proc를 최소 2048로 설정할 수 있습니다.

주: 또한 커널 매개변수 maxfiles, maxfiles_lim, maxproc, shmem, tcp_conn_request_max, tcp_ip_abort_interval, tcp_keepalive_interval, tcp_rexmit_interval_initial, tcp_rexmit_interval_max, tcp_rexmit_interval_min, tcp_xmit_hiwater_def, tcp_recv_hiwater_def에 대한 조정도 고려하십시오.

- Linux의 경우:

다음 명령을 발행하십시오.

```
echo "1024 61000" > /proc/sys/net/ipv4/ip_local_port_range
echo "5" > /proc/sys/net/ipv4/tcp_fin_timeout
```

- Solaris의 경우:

다음 명령을 발행하십시오.

```
ndd /dev/tcp -set tcp_time_wait_interval 5000
```

다음과 같이 /etc/system 파일을 편집하십시오.

```
set tcp:tcp_conn_hash_size=8129
```

- Windows의 경우:

TCP 시간 대기 간격이 설정된 레지스트리 항목을 작성해야 합니다. 자세한 정보는 Windows 문서를 참조하십시오.

Linux 커널 조정

Linux 커널의 몇 가지 한계는 낮으며, 수정할 수 있습니다. 일부는 /proc 파일 시스템을 통해 변경할 수 있으며 나머지는 커널 재컴파일이 필요합니다.

주: /proc 파일 시스템은 가상입니다. 즉, 디스크에 물리적으로 존재하지 않습니다. 대신, Linux 커널로의 인터페이스로 이용할 수 있습니다. 존재하지 않기 때문에 재시작 시 입력값이 손실됩니다. 따라서 원하는 변경사항을 RedHat의 /etc/rc.d/rc.local 파일 또는 SUSE의 /etc/rc.config 파일의 /proc 파일 시스템에 위치시키십시오. 그러면 재시작 시 변경사항이 항상 활성화됩니다.

일부 권고사항은 다음과 같습니다.

- 파일 설명자 최대값은 기본적으로 4096입니다. 다음 사항을 rc.local 파일에 추가하여 이 값을 변경할 수 있습니다.

```
echo 32768 > /proc/sys/fs/file-max
```

- inode 최대값은 기본적으로 16384입니다. 다음 사항을 rc.local 파일에 추가하여 이 값을 변경할 수 있습니다.

```
echo 65536 > /proc/sys/fs/inode-max
```

- TCP와 UDP 포트 범위는 기본적으로 1024 – 4999입니다. rc.local 파일에 다음을 추가하여 해당 범위를 32768 – 61000으로 변경할 수 있습니다.

```
echo 32768 61000 > /proc/sys/net/ipv4/ip_local_port_range
```

- 허용된 TASK 수의 기본값은 512입니다. TASK가 너무 많이 실행 중이면, 이 값은 프로세스에 대해 스레드의 최대수에 영향을 줍니다. 이 한계값은 *YourKernelSource/include/linux/tasks.h* 파일의 NR_TASKS에 대한 값을 수정하여 2048로 증가시킬 수 있습니다.
- 추가적으로 MIN_TASKS_LEFT_FOR_ROOT 값을 24로 변경하십시오. 이 변경사항에 대한 커널을 재컴파일해야 변경사항이 적용됩니다.

커널을 다시 작성하려면, 꼭 필요한 옵션만 사용 가능하게 하십시오. 고유한 디먼이 필요없는 경우에는 이것을 실행하지 마십시오.

AIX 스레드 조정 변수 조정

AIX 시스템의 경우, Caching Proxy 성능은 시스템 범위 스레드를 사용하고 스레드에 의해 다중 힙이 사용되도록 허용함으로써 개선될 수 있습니다. 성능은 운영 체제의 다중 처리 기능 및 기본 운영 체제의 스레드 스케줄과 관련되어 있습니다. 성능은 다음과 같이 다음의 AIX 스레드 조정 변수를 설정하여 개선될 수 있습니다.

```
export AIXTHREAD_SCOPE=S
export SPINLOOPTIME=500
export YIELDLOOPTIME=100
export MALLOCMULTIHEAP=1
```

이 환경 변수를 `/usr/sbin/ibmproxy`를 시작하기 이전에 설정하거나 이를 `/etc/rc.ibmproxy`에 추가할 수 있습니다(`startsrc -s ibmproxy`를 사용하여 프록시 서버를 시작하는 경우). 이 스레드 조정 변수를 조정하고 나면 성능 개선은 SMP 시스템에서 보다 두드러집니다. 그러나 일부 경우에 개선은 단일 프로세서 시스템에서도 나타날 수 있습니다.

주: 자세한 정보는 AIX 운영 체제 문서의 스레드 조정 변수에 대한 세부사항을 참조하십시오.

제 3 부 Caching Proxy 작동 구성

이 파트에서는 Caching Proxy 컴포넌트가 클라이언트 요청에 응답하는 방법을 설명하고 이 작동을 구성하는 프로시저를 제공합니다. 이 프록시 서버 구성 요소는 일반적으로 웹 관리자가 관리하며 호스트 컴퓨터 시스템 또는 네트워크 내의 다른 컴퓨터 시스템에 있는 다른 프로세스에는 영향을 미치지 않습니다.

이 파트에는 다음과 같은 장이 들어 있습니다.

41 페이지의 제 10 장 『요청 처리 관리』

53 페이지의 제 11 장 『로컬 콘텐츠 전달 관리』

57 페이지의 제 12 장 『FTP 연결 관리』

61 페이지의 제 13 장 『서버 처리 사용자 정의』

73 페이지의 제 14 장 『헤더 옵션 구성』

75 페이지의 제 15 장 『API(application programming interface) 정보』

제 10 장 요청 처리 관리

Caching Proxy가 클라이언트 요청을 받을 경우, 요청된 방법이 사용 가능하면 URL 필드에 지정된 오브젝트에 대해 메소드 필드에 지정된 조치를 수행합니다. 프록시 서버는 관리자가 정의한 맵핑 규칙 세트에 따라 URL을 해석합니다. 이 맵핑 규칙은 Caching Proxy에게 웹 서버로 작동하여 로컬 파일에서 오브젝트를 검색하거나 프록시 서버로 작동하여 기점 서버로부터 오브젝트를 검색하도록 지시할 수 있습니다.

이 장에서는 메소드 사용 방법, 맵핑 규칙 정의 방법 및 대리 프록시 서버 구성 방법을 설명합니다.

HTTP/FTP 메소드 사용 가능

서버에 대한 클라이언트 요청에는 서버가 지정된 오브젝트에서 수행할 조치를 지시하는 메소드 필드가 있습니다.

다음은 프록시 서버가 지원하는 메소드 목록 및 메소드가 사용 가능한 경우, 프록시 서버가 이 메소드를 포함하는 클라이언트 요청에 응답하는 방법에 대한 설명입니다.

주: 일부 메소드는 HTTP 및 FTP의 요청과 동일합니다. 다음 메소드를 HTTP에 사용 가능하게 하면 FTP에 대해서도 사용 가능해집니다.

CONNECT

CONNECT 메소드는 프록시 서버를 통해 요청 및 응답을 터널링하도록 허용합니다. 이는 정방향 프록시 구성에만 적용합니다.

Enable CONNECT 메소드에 대한 형식 및 사용 가능한 옵션에 대한 정보는 128 페이지의 『SSL 터널링 구성』을 참조하십시오.

DELETE

프록시 서버가 URL로 식별된 오브젝트를 삭제합니다. DELETE는 클라이언트가 Caching Proxy에서 파일을 지울 수 있도록 허용합니다. 서버 보호 설정을 사용하여 DELETE를 사용할 수 있는 사용자와 해당 파일을 정의하십시오. 자세한 내용은 121 페이지의 제 25 장 『서버 보호 설정』을 참조하십시오.

GET 프록시 서버는 URL이 식별한 데이터를 모두 리턴합니다. URL이 실행 가능 프로그램을 참조하면, 프록시가 그 프로그램의 출력을 리턴합니다. 이 메소드는 지속적인 연결을 통해 처리될 수 있습니다.

HEAD

프록시 서버는 문서 본문 없이 URL이 식별한 HTTP 문서 헤더만 리턴합니다.

OPTIONS

프록시 서버가 URL로 구별되는 요청/응답 연결의 통신 옵션에 대한 정보를 리턴합니다. 이 메소드를 사용하여, 클라이언트는 오브젝트를 실행하거나 검색하지 않고 오브젝트와 연관된 옵션 및 요구사항 또는 서버의 성능을 판별할 수 있습니다.

POST 요청에는 데이터와 URL이 있습니다. 프록시 서버는 요청에 포함된 데이터를 이를 처리하는 URL에서 식별하는 자원의 새 하위 자원으로 승인합니다. 이 자원은 일부 다른 프로토콜에 대한 게이트웨이인 데이터 승인 프로그램이나 주석을 승인하는 분리된 프로그램이 될 수 있습니다.

POST 메소드는 기존 자원에 대한 주석을 처리하도록 설계되었습니다. 예를 들어, 게시판, 뉴스그룹, 메일 목록, 또는 유사한 자원 그룹으로 메시지를 전달하거나, 데이터 블록을 양식에서 데이터 처리 프로그램으로 제공하거나, 추가 조작을 통해 데이터베이스를 확장합니다. Caching Proxy의 경우, POST 메소드는 구성 및 관리 양식을 처리하는 데 사용합니다.

이 메소드는 지속적인 연결을 통해 처리될 수 있습니다.

PUT 요청에는 데이터와 URL이 있습니다. 프록시 서버가 URL로 식별된 자원의 데이터를 저장합니다. 이미 자원이 있으면, PUT이 요청에 포함된 데이터로 자원을 대체합니다. 자원이 없으면, PUT은 자원을 작성하여, 요청에 포함된 데이터로 자원을 채웁니다. 이 메소드는 지속적인 연결을 통해 처리될 수 있습니다.

PUT 메소드를 사용 가능하게 하면, 파일이 HTTP 및 FTP를 사용하여 Caching Proxy에 기록됩니다. 클라이언트가 PUT을 사용하여 Caching Proxy에 기록할 수 있으므로, 서버 보호 설정을 사용하여 PUT을 사용할 수 있는 사람과 PUT을 사용할 수 있는 파일을 정의하십시오. (121 페이지의 제 25 장 『서버 보호 설정』을 참조하십시오.)

TRACE

프록시 서버는 클라이언트가 전송한 요청 메시지를 되돌리합니다. 이 메소드로 클라이언트가 요청 연결의 다른 끝에서 수신되는 내용을 알고, 그 정보를 검사 및 진단으로 사용할 수 있습니다. 프록시 응답의 내용 유형은 message/http입니다.

연관된 지시문

다음 지시문은 HTTP/FTP 메소드를 사용 가능하게 합니다.

- 226 페이지의 『Enable — HTTP 메소드 사용 가능』
- 224 페이지의 『Disable — HTTP 메소드 사용 불가능』

자세한 정보는 13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』을 참조하십시오.

구성 및 관리 양식

다음 구성 및 관리 양식은 연관된 지시문의 값을 편집합니다.

- 서버 구성 -> 요청 처리 -> HTTP 메소드 -> GET
- 서버 구성 -> 요청 처리 -> HTTP 메소드 -> HEAD
- 서버 구성 -> 요청 처리 -> HTTP 메소드 -> POST
- 서버 구성 -> 요청 처리 -> HTTP 메소드 -> PUT
- 서버 구성 -> 요청 처리 -> HTTP 메소드 -> DELETE
- 서버 구성 -> 요청 처리 -> HTTP 메소드 -> OPTIONS
- 서버 구성 -> 요청 처리 -> HTTP 메소드 -> TRACE
- 서버 구성 -> 요청 처리 -> HTTP 메소드 -> CONNECT

주: POST 메소드를 사용 불가능하게 하면 Caching Proxy를 구성하는 데 구성 및 관리 양식을 사용할 수 없습니다.

자세한 정보는 7 페이지의 제 2 장 『구성 및 관리 양식 사용법』을 참조하십시오.

WebDAV 메소드, MS Exchange 메소드, 사용자 정의 메소드 사용 가능

이는 역방향 프록시 구성에만 적용합니다.

표준 HTTP 메소드 지원 뿐만 아니라, Caching Proxy는 RFC에서 정의되거나 일부 응용프로그램에서 사용되는 다른 메소드 전달을 지원합니다. Caching Proxy는 사용자 정의 메소드를 지원하고, 프록시 서버를 통해 전달되도록 허용합니다.

WebDAV(Web-based Distributed Authoring and Versioning)는 원격 웹 서버에서 파일을 협업적으로 편집, 관리할 수 있게 하는 HTTP 프로토콜의 확장 세트입니다. Caching Proxy는 WebDAV 메소드, Microsoft Exchange Server에서 사용되는 메소드, 사용자 정의 메소드를 지원합니다.

이 메소드는 하드 코드되며, 지시문 사용 가능 및 사용 불가능으로 관리됩니다. 관리자는 PROTECT 지시문에서 정의된 해당 메소드 마스크를 사용하여 이 메소드의 사용에 권한을 부여할 수 있습니다.

지원되는 WebDAV 메소드(RFC 2518): PROPFIND , PROPPATCH , MKCOL, COPY, MOVE, LOCK, UNLOCK, SEARCH

지원되는 MS Exchange 메소드: BMOVE, BCOPY, BDELETE, BPROPFIND, BPROPPATCH, POLL, NOTIFY, SUBSCRIBE, UNSUBSCRIBE, ACL, SUBSCRIPTIONS, X_MS_ENUMATTS

WebDAV 또는 MS Exchange Server 메소드가 사용 가능한 경우, Caching Proxy는 요청을 대상 서버에만 전달하고, 요청 본문에서 자원 링크를 재작성하지 않습니다.

또한 Caching Proxy는 사용자 정의 메소드를 백엔드 서버에 전달합니다. `ibmproxy.conf` 파일의 `Enable` 지시문에 대한 다음 구문을 사용하여 사용자 정의된 메소드를 사용 가능하게 하십시오.

```
Enable user-defined-method [WithBody | WithoutBody]
```

`WithBody` 또는 `WithoutBody` 값을 지정하여 사용자 정의 메소드에 요청 본문이 필요한 경우, 프록시에 나타냅니다.

다음 예를 통해 사용자 정의 메소드 `My_METHOD`를 사용 가능하게 하고, 메소드에 요청 본문이 필요한 프록시에 나타냅니다.

```
Enable MY_METHOD WithBody
```

연관된 지시문

다음 지시문은 WebDAV 메소드, MS Exchange 메소드 및 사용자 정의 메소드를 사용 가능하게 합니다.

- 226 페이지의 『Enable — HTTP 메소드 사용 가능』
- 224 페이지의 『Disable — HTTP 메소드 사용 불가능』

자세한 정보는 13 페이지의 제 4 장 『`ibmproxy.conf` 파일 직접 편집』을 참조하십시오.

맵핑 규칙 정의

맵핑 규칙은 Caching Proxy에 대한 클라이언트 요청을 처리(예를 들어, 기점 서버로 전달(프록시됨), 경로재지정, 거부)하는 구성 지시문입니다. Caching Proxy가 제대로 작동하기 위해서는 맵핑 규칙을 올바르게 설정하는 것이 중요합니다. 맵핑 규칙은 다음에 영향을 미칩니다.

- 기본 프록시 기능
- 브라우저 기반 구성 및 관리 양식에 액세스
- servlet 결과 및 기타 동적으로 생성된 콘텐츠를 캐시하는 기능

맵핑 규칙 지시문은 다음 양식을 사용합니다.

```
rule template target [IP_address | host_name]:[port]
```

제공된 템플릿 및 IP 포트 조합에 일치하는 요청만이 해당 규칙에 적용됩니다. 템플릿은 와일드카드(예: `https://**/*.asp`)를 포함할 수 있습니다.

구성 파일에 규칙이 나타나는 순서는 매우 중요합니다. 맵 지시문을 제외하고 요청이 템플릿과 일치하면 바로 처리되고 후속 규칙은 평가되지 않습니다. 맵 지시문은 요청의 URL을 바꿉니다. 이 새 요청을 나머지 맵핑 규칙과 계속 비교합니다.

맵핑 규칙

다음 맵핑 규칙은 제공된 템플리트와 일치하는 클라이언트 요청에 적용됩니다.

- **Map, MapQuery** — 요청을 다시 기록합니다. Map 및 MapQuery 규칙은 요청 URL(템플리트)을 다른 URL 문자열(대상)로 바꿉니다. 대체한 후, 새 문자열을 포함하는 요청을 계속해서 다른 맵핑 규칙과 비교합니다.
- **RuleCaseSense** — 대소문자를 구분하지 않은 응용프로그램 URL의 요청 맵핑을 허용합니다. off로 조정된 경우, RuleCaseSense 지시문은 프록시가 대소문자를 구분하지 않고, ibmpoxy.conf 파일에서 정의된 규칙에 대해 요청을 맵핑하도록 허용합니다.
- **Pass, Exec** — 로컬로 요청을 제공합니다. Pass 및 Exec 규칙은 프록시 서버에서 요청을 처리합니다. Pass 규칙은 요청 URL(템플리트)를 프록시 서버(대상)에서 제공하는 파일로 맵핑합니다. Exec 규칙은 요청 URL을 프록시 서버에서 실행하는 CGI 프로그램으로 맵핑합니다.
- **Fail** — 요청을 거부합니다. Fail 규칙은 프록시 서버에서의 요청(템플리트)을 거부합니다. Fail 규칙의 템플리트와 일치하는 요청은 더 이상 처리되지 않습니다. Fail 규칙에는 대상 인수가 없습니다.
- **Redirect** — 요청을 전달합니다. Redirect 규칙은 요청(템플리트)을 다른 웹 서버(대상)로 전달합니다. 통신 프로토콜을 포함한 완전한 URL이 이 규칙의 대상이므로 경로 재지정 중에 프로토콜을 변경(예: HTTP 요청에 SSL 암호화 추가)하는 것이 가능합니다. 경로 재지정은 요청을 만족시키기 전에 캐시를 확인하지 않습니다.
- **Proxy, ProxyWAS** — 요청을 프록시합니다. Proxy 및 ProxyWAS 규칙은 요청(템플리트)을 다른 서버(대상)로 전달합니다. 단순한 Redirect 규칙과는 달리 Proxy 규칙은 프록시 서버가 캐시를 확인하여 요청을 만족시키고 기점 서버의 콘텐츠를 캐시하며 고급 기능이 사용 가능한 HTTP 헤더를 기록할 수 있게 합니다. 기점 서버가 WebSphere Application Server인 경우에는 Proxy 규칙 대신에 ProxyWAS 규칙을 사용하십시오.

다음 맵핑 규칙이 기점 서버 응답에 대해 적용됩니다.

- **ReversePass** — 경로 재지정된 요청을 자동으로 가로챍니다. ReversePass 규칙은 프록시 서버를 통해 클라이언트로 향하는 중에 기점 서버로부터의 응답을 템플리트에 일치시킵니다. ReversePass 지시문은 클라이언트가 직접 기점 서버에 연결하게 하는 경로 재지정 상태 코드를 발견하도록 설계되었습니다. 클라이언트에게 대상 인수에 정의된 서버에 접속하도록 명령합니다.

다음과 같은 맵핑 규칙이 API 응용프로그램에 적용됩니다.

- **nameTrans** — 요청 처리의 이름 변환 단계 동안, 요청을 승인하여 대체 파일 경로가 정의한 API 응용프로그램을 실행합니다.
- **service** — 요청 처리의 서비스 단계 동안, 요청을 승인하여 대체 파일 경로가 정의한 API 응용프로그램을 실행합니다.

대리 서버 구성

표준 대리 서버를 구성하려면 다음을 수행하십시오.

- 프록시 서버 포트를 80으로 설정하십시오.

Port 80

- 다른 규칙에 앞서 포트 80에서 받은 모든 요청을 기점 서버로 프록시하는 프록시 규칙을 추가하십시오.

Proxy /* http://our.content.server.com/* :80

- 포트 80이 아닌 관리 포트를 사용 가능하게 하십시오.

AdminPort 8080

그러면 포트 80의 모든 HTTP 통신을 기점 서버로 프록시할 수 있습니다. 관리 포트에 입력된 통신은 초기 와일드 카드 프록시 규칙과 일치하지 않으므로 영향을 받지 않습니다. 나머지 맵핑 규칙은 요청을 처리하는 데 사용됩니다.

연관된 지시문

다음 지시문은 맵핑 규칙을 정의합니다.

- 253 페이지의 『Map — 규칙을 일치시키는 요청 경로 문자열을 사용하여 새 요청 문자열에 일치하는 요청 변경』
- 255 페이지의 『MapQuery — 규칙을 일치시키는 요청 경로 및 조회 문자열을 사용하여 일치하는 요청을 새 요청 문자열로 변경』
- 296 페이지의 『RuleCaseSense — 대소문자를 구분하지 않는 응용프로그램 URL에서 요청을 맵핑』
- 267 페이지의 『Pass — 요청을 승인하기 위한 템플릿 지정』
- 230 페이지의 『Exec — 요청을 일치시키기 위한 CGI 프로그램 실행』
- 291 페이지의 『Redirect — 다른 서버에 전송된 요청에 대한 템플릿 지정』
- 282 페이지의 『Proxy — 프록시 프로토콜 또는 역방향 프록시 지정』
- 288 페이지의 『ProxyWAS — WebSphere Application Server로 요청 전송 지정』
- 293 페이지의 『ReversePass — 자동으로 재지정된 요청 교차』

자세한 정보는 13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』을 참조하십시오.

구성 및 관리 양식

다음 구성 및 관리 양식은 연관된 지시문의 값을 편집합니다.

- 서버 구성 -> 요청 처리 -> 요청 경로 지정

주: 구성 및 관리 양식은 포트 번호 인수를 지원하지 않습니다.

결합 재작성 사용 가능(선택적)

이는 역방향 프록시 구성에만 적용합니다.

JunctionRewrite 지시문은 서버의 상대 URL이 결합 사용 시 해당하는 기점 서버로 반드시 맵핑되도록 기점 서버의 응답을 재작성하는 결합 재작성 루틴을 Caching Proxy 내에서 사용 가능하게 합니다. 또한 결합 재작성 플러그인이 사용 가능해야 합니다. 결합은 프록시 맵핑 규칙에 의해 정의됩니다.

프록시 맵핑 규칙을 사용하여 결합을 정의하는 경우, JunctionPrefix 옵션에 상관없이 Proxy 지시문을 사용할 수 있습니다.

JunctionPrefix 옵션 없이 결합 정의

다음은 결합 재작성 루틴이 작동할 수 있는 유효한 결합의 예제입니다.

- Proxy /shop/* http://shopserver.acme.com/*
- Proxy /auth/* http://authserver.acme.com/*

다음은 결합 재작성 루틴이 작동하지 않는 유효한 결합의 예제입니다.

- Proxy /* http://defaultserver.acme.com/*

다음은 유효하지 않은 결합의 예제입니다.

- Proxy /images/*.gif http://imageserver.acme.com/images/*.gif
- Proxy /cgi-bin/* http://cgiserver.acme.com/cgi/perl/*

이들 맵핑 규칙이 shopserver, authserver 및 b2bserver에 대한 결합을 작성합니다. shopserver는 해당하는 HTML 태그 안에 다음과 같은 URL을 포함하여 HTML 문서를 리턴합니다.

- /index.html(서버 상대 참조)
- /images/shop.gif(서버 상대 참조)
- buy/buy.jsp(디렉토리 상대 참조)
- http://ebay.com(절대 참조)

결합 재작성 루틴은 프록시 맵핑 규칙을 사용하여 서버 상대 참조를 다음과 같이 재작성합니다.

- /shop/index.html(변경됨)
- /shop/images/shop.gif(변경됨)
- buy/buy.jsp(변경되지 않음)
- http://ebay.com(변경되지 않음)

JunctionPrefix 옵션으로 결합 정의(권장하는 메소드)

Proxy 지시문에서 JunctionPrefix 옵션을 사용하는 경우, 프록시 규칙의 첫 URL 패턴에서 JunctionPrefix를 추론하는 대신, 다음 형식을 사용하여 프록시 규칙에서 결합 접두부를 선언할 수 있습니다.

```
Proxy url_pattern1 url_pattern2 JunctionPrefix:url_prefix
```

JunctionPrefix를 사용하는 경우, 첫 URL 패턴의 형식에는 제한이 없습니다. JunctionPrefix 옵션을 사용하지 않는 경우, 프록시 URL에는 다음 형식이 있어야 합니다. Proxy /market/* http://b2bserver/*. 그러나, JunctionPrefix를 사용하는 경우, 다음 프록시 규칙이 결합 재작성에 유효합니다.

```
Proxy /market/partners/*.html http://b2bserver.acme.com/*.html
junctionprefix:/market/partners
```

결합 재작성 루틴은 다음과 같은 태그에 영향을 미칩니다.

표 3. 결합 재작성 루틴의 영향을 받는 태그

태그	속성
!—	URL
A	href
applet	archive, codebase
area	href
base	href
body	background
del	cite
embed	pluginspage
form	action
input	src
frame	src, longdesc
iframe	src, longdesc
ilayer	src, background
img	src, usemap, lowsrc, longdesc, dynsrc
layer	src, background
연결	href
meta	url
object	data, classid, codebase, codepage
script	src
table	background
td	background
th	background
tr	background

주: 결합 재작성 루틴은 JavaScript 또는 브라우저 내의 플러그인에 의해 생성된 태그에는 영향을 미치지 않습니다.

연관된 지시문

다음과 같은 지시문을 사용하여 결합 재작성 루틴 및 플러그인을 사용 가능하게 합니다.

- 299 페이지의 『ServerInit — 서버 초기설정 단계 사용자 정의』
- 310 페이지의 『Transmogriifier — 데이터 조작 단계 사용자 정의』
- 245 페이지의 『JunctionRewrite — URL 재작성 사용 가능』
- 246 페이지의 『JunctionRewriteSetCookiePath — JunctionRewrite 플러그인과 사용되는 경우, Set-Cookie 헤더에 경로 옵션을 재작성』
- 245 페이지의 『JunctionReplaceUrlPrefix — JunctionRewrite 플러그인과 사용되는 경우, 접두부 삽입 대신 URL 바꾸기』
- 246 페이지의 『JunctionSkipUrlPrefix — JunctionRewrite 플러그인과 사용되는 경우, 이미 접두부를 포함하는 URL 재작성을 건너뛰기』

자세한 정보는 13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』을 참조하십시오.

구성 및 관리 양식

다음과 같은 구성 및 관리 양식을 사용하여 결합 재작성 플러그인을 사용 가능하게 합니다.

- 서버 구성 -> 요청 처리 -> API 요청 처리

주: 구성 및 관리 양식은 JunctionRewrite 지시문을 지원하지 않습니다.

자세한 정보는 7 페이지의 제 2 장 『구성 및 관리 양식 사용법』을 참조하십시오.

JunctionRewrite에 대한 대안으로서의 UseCookie

쿠키를 사용하여 다음과 같이 백엔드 서버 정보를 저장할 수 있습니다. 쿠키는 클라이언트 브라우저로 전송됩니다. 브라우저가 HTML 페이지의 자원에 대한 요청을 전송하는 경우, 이는 Caching Proxy가 요청을 올바른 백엔드 서버로 전달할 수 있도록 쿠키를 첨부합니다.

JunctionRewrite에 대한 대안으로 쿠키를 사용하려면 다음과 같이 ibmproxy.conf 파일을 수정하십시오.

1. **JunctionRewrite on**을 **JunctionRewrite on UseCookie**로 변경하십시오.
2. JunctionRewrite 플러그인을 설명 처리하십시오.

다음은 JunctionRewrite 플러그인 및 쿠키 구현의 비교입니다.

- JunctionRewrite 플러그인
 - HTML 페이지가 재작성됩니다.
 - transmogrifier 플러그인을 사용해야 스크립트 언어 및 애플릿의 재작성을 지원합니다. 『JunctionRewrite 기능성의 확장을 위한 샘플 transmogrifier 플러그인』을 참조하십시오.
 - 성능이 떨어집니다.
 - 백엔드 서버 구성에 제한이 없습니다. 클라이언트는 세션의 백엔드 서버를 상호 액세스할 수 있습니다.
- 쿠키 구현
 - HTML 페이지가 재작성되지 않습니다. 쿠키가 클라이언트로 전송됩니다.
 - 클라이언트 브라우저가 쿠키 지원을 사용 가능하게 해야 합니다.
 - 성능이 증가합니다.
 - 백엔드 서버 구성에 일부 제한이 있습니다. 클라이언트가 세션의 백엔드 서버에서 액세스하는 경우에만 사용할 수 있습니다.

주: UseCookie 옵션과 함께 JunctionRewrite를 사용하는 경우 알려진 제한사항이 있습니다. 쿠키 응용프로그램이 호스트의 한 서브디렉토리에만 적용되더라도, 모든 요청에 대해 URL을 올바르게 변환하지 않습니다. 다음은 결합이 불필요한 ROOT 아래의 URL을 올바르게 핸들하는 2가지 방법입니다.

- ibmproxy.conf 파일에서 JunctionRewrite 지시문 앞에 프록시 규칙을 위치 시키십시오. (JunctionRewrite 지시문 앞의 모든 프록시 규칙은 재작성되지 않습니다.)
- 와일드 카드(*)를 사용하는 대신, 각 URL을 명시적으로 맵핑합니다. 예제는 다음과 같습니다.

```
Proxy /no-junction.jpg http://login-server/no-junction.jpg
```

JunctionRewrite 기능성의 확장을 위한 샘플 transmogrifier 플러그인

HTML 파일의 JavaScript™ (SCRIPT) 및 애플릿 (APPLET) 태그 블록을 재작성하고 구문 분석하는 사용자 정의 가능한 예제 코드가 제공됩니다. 독자적으로 JunctionRewrite 플러그인은 Javascript로 또는 Java™의 매개변수 값으로 자원 연결을 처리할 수 없습니다.

Caching Proxy 설치 이후에는 동일 코드를 컴파일할 수 있으며 JunctionRewrite로 실행하도록 이를 구성할 수 있습니다.

다음의 예제 파일은 픽스팩을 다운로드한 디렉토리 아래의 ...samples/cp/ 하위 디렉토리에 있습니다.

- Makefile(이 예제 플러그인에 대한 Makefile)

- junctionRewrite2.h(사용자 정의된 구문 분석기 처리기에 대한 인터페이스)
- junctionRewrite2.c(위의 인터페이스에 대한 구현)
- scriptHandler.c(예제 JavaScript 재작성 처리기)
- appletHandler.c(예제 애플릿 블록 처리기)
- junctionRewrite2.def(Windows 플러그인 def 파일)
- junctionRewrite2.exp(Linux 및 UNIX 플러그인 내보내기 파일)

제 11 장 로컬 콘텐츠 전달 관리

Pass 및 Exec 맵핑 규칙은 로컬 콘텐츠를 요청한 클라이언트로 전달하는 데 사용됩니다. 기본적으로 와일드 카드 템플리트를 사용하는 Pass 규칙은 최종 맵핑 규칙으로 배치됩니다. 이 규칙은 이전 템플리트와 일치하지 않는 모든 요청에게 대상 디렉토리(일반적으로 문서 루트 디렉토리라고 함)에서 파일을 검색하도록 명령합니다.

파일 이름을 포함하지 않는 URL을 받을 경우, Caching Proxy는 지정된 디렉토리를 검색하거나(제공될 경우) 구성 파일에 지정된 환영 페이지 목록과 일치하는 파일을 검색하여 (디렉토리가 지정되지 않은 경우) 요청을 만족시킵니다. 하나 이상의 환영 페이지가 정의되어 있는 경우, 프록시 서버는 환영 페이지가 정의되어 있는 순서대로 페이지를 검색합니다. 첫 번째 찾은 환영 페이지를 제공합니다.

서버 홈 페이지는 디렉토리나 파일 이름 없이 서버의 URL만 있는 요청을 수신할 때 기본적으로 제공되는 웹 페이지입니다. 앞에서 설명한 대로 기본 와일드 카드 맵핑 규칙에서는 서버 홈 페이지가 문서 루트 디렉토리에 저장되어 있고 홈 페이지의 파일 이름이 정의된 환영 페이지와 일치해야 합니다.

주: 일부 웹 브라우저에서는 브라우저 시작 시 로드되는 첫 번째 페이지를 지칭하는 용어로 홈 페이지를 사용합니다. 이 문서는 서버 홈 페이지 전용 용어를 사용합니다.

이 장에서는 문서 루트 디렉토리 및 환영 페이지를 정의하는 방법에 대해 설명합니다.

문서 루트 디렉토리 정의

기본 문서 루트 디렉토리는 다음과 같습니다.

- Linux 및 UNIX의 경우: `/opt/ibm/edge/cp/server_root/pub/lang/`
- Windows의 경우: `drive:\Program Files\IBM\edge\cp\server_root\pub\lang\` 또는 설치 중에 HTML 디렉토리로 지정한 디렉토리

연관된 지시문

다음 지시문은 문서 루트 디렉토리를 정의합니다.

- 267 페이지의 『Pass — 요청을 승인하기 위한 템플리트 지정』

자세한 정보는 13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』을 참조하십시오.

구성 및 관리 양식

구성 및 관리 양식에서 문서 루트 디렉토리를 변경하려면 다음 프로시저를 사용하십시오.

1. 서버 구성 -> 요청 처리 -> 경로 지정 요청을 선택하십시오.
2. 요청 경로 지정 테이블에서 요청 템플릿 컬럼에 /*(슬래시 별표)가 들어 있는 행을 찾으십시오. 이는 문서 루트 디렉토리를 나타냅니다. 테이블 아래의 색인 상자에서 그 행의 색인 컬럼에 있는 숫자에 해당하는 숫자를 누르십시오.
3. 바꾸기를 누르십시오.
4. 조치 확장 목록에서 전달을 누르십시오.
5. URL 요청 템플릿 필드에 /*를 입력하십시오.
6. 대체 파일 경로 필드에 새 문서 루트 디렉토리를 입력하십시오.
7. 제출을 누르십시오.
8. 변경사항이 승인된 후, 맨 위 프레임에서 서버 재시작 아이콘(I)을 누르십시오. 재시작 후, 서버는 새 문서 루트 디렉토리를 사용하기 시작합니다.

자세한 정보는 7 페이지의 제 2 장 『구성 및 관리 양식 사용법』을 참조하십시오.

기본 환영 페이지 정의

서버는 문서 루트 디렉토리에서 홈 페이지를 찾지만, 서버가 돌려보낸 고유한 파일이 환영 페이지 목록에 의해 정의됩니다.

환영 페이지 정보

서버는 파일 이름을 지정하지 않는 URL 요청을 수신할 때, 서버 구성 파일에 설정된 환영 페이지 목록에 따라 요청을 만족시키려고 합니다. 이 목록은 기본 홈 페이지로 사용할 파일을 정의합니다. 서버는 문서 루트 디렉토리에 있는 파일과 환영 페이지 목록을 일치시켜 사용자 홈 페이지를 판별합니다. 처음 일치된 것은 홈 페이지로 리턴되는 파일입니다. 일치하는 것이 없다면, 서버는 문서 루트 디렉토리 목록을 표시합니다.

특정 파일을 서버의 홈 페이지로 사용하거나 디렉토리나 파일 이름을 요청이 지정하지 않았을 때 리턴시키려면, 그 파일을 문서 루트 디렉토리에 위치시키고 파일의 이름이 환영 페이지 목록에 나열된 파일 이름 중 하나와 일치하도록 해야 합니다.

기본 구성 파일은 다음의 파일 이름을 정의하며, 이 순서에 따라 환영 페이지로 사용됩니다.

1. welcome.html 또는 welcome.htm
2. index.html 또는 index.htm
3. Frntpage.html

서버는 이 목록에서 파일 이름이 일치하는 첫 번째 파일을 리턴합니다. welcome.html 이나 index.html 파일을 작성하여 문서 루트 디렉토리에 배치할 때까지 서버는 Frntpage.html을 홈 페이지로 사용합니다.

예를 들어, 기본 구성을 사용하고 있거나 문서 루트 디렉토리에 이름이 welcome.html인 파일은 없지만 이름이 index.html이나 FrntPage.html인 파일이 있을 경우에는 index.html 파일이 홈 페이지로 사용됩니다.

홈 페이지가 없으면, 문서 루트 디렉토리의 내용이 디렉토리로 표시됩니다.

연관된 지시문

다음 지시문은 환영 페이지를 정의합니다.

- 315 페이지의 『Welcome — 환영 파일의 이름 지정』

자세한 정보는 13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』을 참조하십시오.

구성 및 관리 양식

다음 구성 및 관리 양식은 환영 페이지를 정의합니다.

- 서버 구성 -> 디렉토리 및 환영 페이지 -> 환영 페이지

자세한 정보는 7 페이지의 제 2 장 『구성 및 관리 양식 사용법』을 참조하십시오.

제 12 장 FTP 연결 관리

이는 정방향 프록시 구성에만 적용합니다.

Caching Proxy는 FTP URL에 대한 요청을 적절한 FTP 서버로 프록시하지만, FTP 클라이언트의 요청을 프록시하는 데 사용할 수 없습니다. HTTP 클라이언트로부터 받은 FTP 요청만 지원할 수 있습니다(ftp:// 프로토콜 설계 사용).

FTP 파일에 대한 요청에 대해서는 GET, PUT, DELETE 메소드만 지원됩니다. FTP 디렉토리 목록에 대한 요청에 대해서는 GET 메소드만 지원됩니다. 기본적으로 Caching Proxy에서는 PUT, DELETE를 사용할 수 없습니다. 자세한 정보는 41 페이지의 『HTTP/FTP 메소드 사용 가능』을 참조하십시오.

이 장에서는 FTP 파일을 보호하고 FTP 서버 로그인, 디렉토리 경로 및 체인을 관리하는 방법에 대해 설명합니다.

FTP 파일 보호

FTP 파일 업로드에 PUT 메소드를 또는 FTP 파일 삭제에 DELETE 메소드를 사용 가능하게 했으면, 권한이 없는 파일이 FTP 서버에서 갱신하는 것을 막기 위해 최소한 PUT 및 DELETE 요청에 대해서 FTP 프록시 보호를 정의해야 합니다.

FTP 요청의 프록시를 보호하려면, 구성 및 관리 양식에서 서버 구성 -> 문서 보호를 선택하십시오. FTP 파일 요청에 대한 보호 설정을 작성하려면, 요청 템플리트를 시작할 때 ftp://를 포함해야 합니다. 예를 들어, exams 디렉토리의 파일을 보호하려면 ftp://exams/* 템플리트를 사용하십시오.

보호 설정 작성에 대한 정보는 121 페이지의 제 25 장 『서버 보호 설정』을 참조하십시오.

FTP 서버 로그인 관리

요청 URL에 사용자 ID 및 암호가 지정되지 않으면, Caching Proxy는 ANONYMOUS 사용자 ID를 사용하여, 요청받은 FTP 서버에 익명으로 로그인을 시도합니다. 많은 FTP 서버에서는 anonymous FTP의 암호로 전자 우편 주소를 필요로 합니다. FTP 서버가 anonymous 로그인으로 암호를 요청한 경우, Caching Proxy는 구성 파일의 WebmasterEmail 지시문이 지정한 전자 우편 주소를 전송합니다.

구성 및 관리 양식에 웹 마스터 전자 우편 주소를 설정하려면 서버 구성 -> System Management -> SNMP MIB를 선택하십시오. 전자 우편 주소는 WebmasterEmail

지시문을 사용해서 설정할 수도 있습니다. 자세한 내용은 314 페이지의 『WebMasterEMail — 서버 선택 보고서를 수신할 전자 우편 주소 설정』 참조 섹션을 참조하십시오.

요청 URL의 FTP 서버가 로그인할 고유한 사용자 ID와 암호를 요구하면, 사용자는 요청 URL에 사용자 ID와 암호를 입력할 수 있습니다. 다음은 그 예제입니다.

```
ftp://userid:password@ftpserverhost/
```

요청 URL의 FTP 사용자 ID에 대한 암호를 지정하지 않으려면, 사용자는 `ftp://userid@ftpserverhost` URL에 사용자 ID만 입력할 수 있습니다. Caching Proxy는 우선 암호 없이 지정된 사용자 ID로 FTP 서버에 로그인하려고 합니다. 암호 없이 시도한 로그인에 실패하면, 브라우저는 지정된 사용자 ID와 연관된 암호를 요청하는 프롬프트를 표시합니다.

anonymous 로그인이 아닌 경우에는 최소한 사용자 ID가 URL에 지정되어야 합니다. 사용자 ID가 지정되지 않으면, anonymous 로그인이 시도되고 클라이언트에 사용자 ID를 요청하는 프롬프트가 표시되지 않습니다.

FTP 디렉토리 경로 관리

FTP URL의 경로 이름을 사용자의 작업 디렉토리와 관련하여 해석할지 또는 루트 디렉토리와 관련하여 해석할지 여부를 Caching Proxy에 지정해야 합니다. 예를 들어, FTP 서버에 로그인한 사용자가 `/export/home/user1`의 기본 작업 디렉토리를 가지고 있고, `test`라는 하위 디렉토리에서 `test1.exe`라는 파일을 검색하려고 하는 경우, 프록시는 FTP URL이 해석되는 방법에 따라 다음 URL을 사용하여 FTP 서버에서 파일을 검색할 수 있습니다.

- 절대 경로 이름이 설정된 경우, `ftp://user1:user1pw@FTPhost/export/home/user1/test/test1.exe`
- 상대 경로 이름이 설정된 경우, `ftp://user1:user1pw@FTPhost/test/test1.exe`

상대 FTP URL 경로가 설정되면, 사용자는 루트 디렉토리를 표시하는 `%2F`가 있는 처음의 슬래시 문자(/)를 이탈시키는 규칙을 사용하여 절대 경로를 계속 지정할 수 있습니다. 예를 들어, 작업 디렉토리가 `/export/home/user1`인 `user1`이 `user2`의 작업 디렉토리인 `/export/home/user2`에 액세스하려는 경우, 상대 FTP URL 경로 이름이 선택된 경우에도 `ftp://user1:user1pw@FTPhost/%2Fexport/home/user2/ file` 요청을 루트 디렉토리 /(즉, 절대 경로 이름으로)로 올바르게 해석합니다.

FTP URL 해석 방법을 설정하려면, 구성 및 관리 양식에서 **프록시 구성** -> **프록시 성능**을 선택하십시오. **FTP URL paths should be:** 밑의 양식 아랫 부분에서, 경로 시작으로 절대 경로를 선택하여 서버의 루트 디렉토리를 지정하거나, 상대 경로를 선택하여 사용자의 작업 디렉토리를 지정하십시오.

이 설정은 프록시 구성 파일에서도 변경될 수 있습니다. 자세한 내용은 235 페이지의 『FTPUrlPath — FTP URL이 해석되는 방법 지정』을 참조하십시오.

FTP 체인 관리

다중 웹 프록시 서버에 함께 연결되어 있으면, FTP URL이 있는 요청을 FTP 서버에 직접 전송하지 않고 연결된 웹 프록시 서버로 전송하도록 지정할 수 있습니다. FTP 요청에 체인된 프록시 서버를 지정하려면, 구성 및 관리 양식에서 **프록시 구성 -> 프록시 체인 및 비프록시 도메인**을 선택하십시오. http:// 프로토콜 설계는 이것이 ftp:// 프로토콜 설계 요청을 체인할 때에도 체인된 프록시 URL을 지정하는 데 사용됩니다.

프록시 구성 파일을 사용하여 FTP 연결하기를 구성하려면, 235 페이지의 『ftp_proxy — FTP 요청에 대한 다른 프록시 서버 지정』의 참조 섹션을 참조하십시오.

제 13 장 서버 처리 사용자 정의

이 장에서는 CGI 프로그램과 클라이언트에게 전달된 HTML 문서에 정보를 삽입하기 위한 정보 포함 방법에 대해 설명합니다. 서버의 오류 메시지 및 자원 맵핑을 사용자 정의하는 방법도 설명합니다.

정보 포함

정보 포함 기능으로 서버가 기점 서버(즉, 프록시 또는 캐시된 오브젝트가 아닌 서버)로 작동할 때, 서버가 클라이언트로 전송하는 CGI 프로그램 및 HTML 문서에 정보를 추가할 수 있습니다. 현재 날짜, 파일 크기, 파일의 최종 변경 날짜 등이 클라이언트로 전송할 수 있는 정보 종류의 예입니다. 이 섹션에서는 정보 포함을 위한 명령 형식을 설명하고, 정보 포함을 CGI 프로그램 및 HTML 문서에서 작동하기 위한 방법을 설명합니다. 또한 정보 포함을 사용하여 오류 페이지를 사용자 정의할 수 있습니다.

정보 포함에 대한 고려사항

서버에서 정보 포함을 사용하기 전에 성능, 보안 및 위험 문제를 고려합니다.

- 서버가 파일을 전송하는 중 파일을 처리하면 성능이 크게 떨어질 수 있습니다.
- 일반 사용자가 서버에서 명령을 실행하도록 내버려두면 보안이 약해질 수 있습니다. 정보 포함을 배치할 디렉토리 및 **exec** 명령을 배치할 디렉토리를 결정할 때 주의하십시오. **exec** 명령을 사용할 수 없게 하면 보안 위험을 줄일 수 있습니다.
- 정보 포함을 사용하면 일부 문제점을 유발할 수 있습니다. 예를 들면, 파일의 반복적으로 참조할 수 없습니다. `sleepy.html` 파일을 실행 중이고 프로그램이 `<--!#include file="sleepy.html" -->`을 발견하면, 서버는 오류를 발견하지 못하고 실패할 수 있습니다. (다른 파일 내에서 반복하지 않고 파일을 참조하는 것은 문제가 되지 않습니다.)

정보 포함의 구성

정보 포함을 사용 가능하게 하려면, 구성 및 관리 양식에서 서버 구성 -> 기본 설정값을 선택하십시오. 이 양식을 사용하여 다음 정보 포함 유형 중 승인할 수 있는 유형을 지정하십시오.

- CGI 스크립트
- 파일
- **exec** 명령을 사용하는 CGI 스크립트를 제외한 모든 것
- 없음

이 양식을 사용하여 텍스트 또는 HTML 문서 및 다른 파일 유형에 대한 정보 포함 처리 수행 여부도 지정하십시오.

또한 포함에 사용하는 파일 확장자를 인식했는지 확인하십시오. 구성 및 관리 양식에서 서버 구성 -> MIME 유형 및 인코딩을 선택한 후, MIME 유형 양식을 사용하십시오..shtml 및 .htmls 확장자는 기본값으로 인식함에 주의하십시오.

프록시 구성 파일의 지시문을 편집하여 정보 포함의 서버를 구성하려면, 다음 지시문에 대한 참조 섹션을 참조하십시오.

- 195 페이지의 『AddType — 특정 접미부가 있는 파일의 데이터 유형 지정』
- 242 페이지의 『imbeds — 정보 포함 처리가 사용될지 여부 지정』

정보 포함의 형식

정보 포함 명령은 HTML 문서 또는 CGI 프로그램에 설명으로 포함되어야 합니다. 명령 형식은 다음과 같습니다.

```
<!--#directive tag=value ... -->
또는
<!--#directive tag="value" ... -->
```

값 옆의 따옴표는 선택적이지만, 공백을 임베드하기 위해서는 필수적입니다.

정보 포함을 위한 지시문

이 절에서는 서버가 정보 포함을 위해 승인하는 지시문을 설명합니다. (이 지시문을 185 페이지의 부록 B 『구성 파일 지시문』에 문서화된 프록시 구성 파일의 지시문과 혼동하지 마십시오.)

config—파일 처리 제어

이 지시문을 사용하여 파일 처리의 특정 측면을 제어하십시오. 유효한 태그는 cmntmsg, errmsg, sizeofmt, timefmt입니다.

cmntmsg

이 태그를 사용하여 다른 지시문이 추가한 설명의 시작 부분 앞에 나올 메시지를 지정하십시오. 지시문 스펙과 "-->" 사이의 텍스트를 포함하는 임의 지시문의 경우, 이 텍스트는 설명으로 처리되고 서버가 클라이언트로 전송하는 파일에 추가됩니다.

예제:

```
<!--#config cmntmsg="[This is a comment]" -->
<!-- #echo var=" " extra text -->
```

결과: <!--[This is a comment] extra text -->

기본값: [the following was extra in the directive]

errmsg

이 태그를 사용하여 파일이 처리될 때 오류가 발생하는 경우 클라이언트로 전송되는 메시지를 지정하십시오. 메시지는 서버의 오류 로그에 기록됩니다.

예제:

```
<!-- #config errmsg="[An error occurred]" -->
```

기본값: "[An error occurred while processing this directive]"

sizefmt

이 태그를 사용하여, 파일 크기를 표시할 형식을 지정하십시오. 다음 예제에서, bytes는 바이트 수를 표시하는 데 사용되는 값이고, abbrev는 킬로바이트나 메가바이트를 표시하는 데 사용되는 값입니다.

예제 1:

```
<!--#config sizefmt=bytes -->
<!--#fsize file=foo.html -->
```

결과: 1024

예제 2:

```
<!--#config sizefmt=abbrev -->
<!--#fsize file=foo.html -->
```

결과: 1K

기본값: "abbrev"

timefmt

이 태그를 사용하여, 날짜를 제공하는 데 사용할 형식을 지정하십시오.

예제:

```
<!--#config timefmt="%D %T" -->
<!--#flastmod file=foo.html -->
```

결과: "10/18/95 12:05:33"

기본값: "%a, %d %b %Y %T %Z"

다음의 strftime() 형식은 timefmt 태그와 함께 유효합니다

지정자	의미
%%	%(으)로 바꾸기
%a	축약형 요일 이름으로 바꾸기
%A	전체 요일 이름으로 바꾸기
%b	축약형 월 이름으로 바꾸기
%B	전체 월 이름으로 바꾸기

지정자	의미
%c	날짜 및 시간으로 바꾸기
%C	세기 숫자로 바꾸기(년을 100으로 나누고 축약함)
%d	월의 날짜(01-31)로 바꾸기
%D	날짜를 %m/%d/%y로 삽입
%e	1년의 월을 10진수(01-12)로 삽입(C POSIX에서만, 2개의 문자, 오른 쪽 자리 맞춤으로, 공백 채워진 필드입니다).
%E[cCxyY]	대체 날짜/시간 형식을 사용할 수 없으면, %E 설명자가 확장되지 않은 형식에 대응됩니다(예를 들어, %EC는 %C에 대응됩니다).
%Ec	대체 날짜 및 시간 표현으로 바꾸기
%EC	대체 표현에서 기본 년(기간)의 이름으로 바꾸기
%Ex	대체 날짜 표현으로 바꾸기
%EX	대체 시간 표현으로 바꾸기
%Ey	대체 표현에서 %EC(년만)에서 오프셋으로 바꾸기
%EY	전체 대체 년 표현으로 바꾸기
%h	축약형 월 이름(%b와 동일한)으로 바꾸기
%H	10진수(00-23)인 시간(23시간 시계)으로 바꾸기
%I	10진수(00-12)인 시간(12시간 시계)으로 바꾸기
%j	년의 날짜(001-366)로 바꾸기
%m	월(01-12)로 바꾸기
%M	분(00-59)으로 바꾸기
%n	새 행으로 바꾸기
%O[deHImMSUwWy]	대체 날짜/시간 형식을 사용할 수 없으면, %E 설명자가 확장되지 않은 형식에 대응됩니다(예를 들어, %Od는 %d에 대응됩니다).
%Od	대체 숫자 기호를 사용하여 0에 대한 대체 문자가 있으면 0을, 그렇지 않으면 공백을 앞에 두도록 채워진 월의 날짜로 바꾸기
%Oe	대체 숫자 기호를 사용하여 공백을 앞에 두도록 채워진 월의 날짜로 바꾸기
%OH	대체 숫자 기호를 사용하여 시간(24시간 시계)으로 바꾸기
%OI	대체 숫자 기호를 사용하여 시간(12시간 시계)으로 바꾸기
%Om	대체 숫자 기호를 사용하여 월로 바꾸기
%OM	대체 숫자 기호를 사용하여 월로 바꾸기
%OS	대체 숫자 기호를 사용하여 초로 바꾸기
%OU	대체 숫자 기호를 사용하여 년의 주 수(일요일을 주의 첫 번째 날로 하며 %U에 해당하는 규칙)로 바꾸기
%Ow	대체 숫자 기호를 사용하여 요일(일요일=0)로 바꾸기
%OW	대체 숫자 기호를 사용하여 년의 주 수(월요일을 주의 첫 번째 날로 함)로 바꾸기
%Oy	대체 표현에서 년(%C로부터 오프셋)으로 바꾸기, 대체 숫자 기호 사용
%p	로컬에서 AM 또는 PM에 해당하는 것으로 바꾸기
%r	%I:%M:%S%p에 해당하는 문자열로 바꾸기
%R	24시간 표시법(%H:%M)의 시간으로 바꾸기
%S	초(00-61)로 바꾸기

지정자	의미
%t	탭으로 바꾸기
%T	%H:%M:%S에 해당하는 문자열로 바꾸기
%u	12진수(1 - 7)로 1이 월요일을 표시하는 요일로 바꾸기
%U	일요일이 주의 첫 번째 날인 년의 주 수(00-53)로 바꾸기
%V	월요일이 주의 첫 번째 날인 년의 주 수(00-53)로 바꾸기
%w	일요일이 0인 요일(0-6)로 바꾸기
%W	월요일이 주의 첫 번째 날인 년의 주 수(00-53)로 바꾸기
%x	적절한 날짜 표현으로 바꾸기
%X	적절한 시간 표현으로 바꾸기
%y	세기의 두 자리 년 수로 바꾸기
%Y	전체 네 자리 년 수로 바꾸기
%Z	표준 시간대의 이름으로 바꾸거나, 표준 시간대를 알 수 없으면 문자 없음

운영 체제 구성은 전체 및 축약 월 이름 및 년을 판별합니다.

echo—변수값 표시

이 지시문을 사용하여 `var` 태그로 지정된 환경 변수 값을 표시하십시오. 변수가 발견되지 않으면, (없음)이 표시됩니다. 또한 **echo**는 **set**이나 **global** 지시문이 설정한 값을 표시할 수 있습니다. 다음의 환경 변수를 표시할 수 있습니다.

DATE_GMT

그리니치 표준 시간대의 현재 날짜 및 시간. 이 변수의 형식은 **config timefmt** 지시문을 사용하여 정의됩니다.

DATE_LOCAL

현재 날짜 및 로컬 시간. 이 변수 형식은 **config timefmt** 지시문을 사용하여 정의됩니다.

DOCUMENT_NAME

최상의 문서 이름. HTML이 CGI에 의해 생성된 경우, 이 변수는 CGI 이름이 들어 있습니다.

DOCUMENT_URI

클라이언트가 조회 문자열 없이 요청한 전체 URL.

LAST_MODIFIED

현재 문서가 최종 변경된 날짜 및 시간. 이 변수의 형식은 **config timefmt** 지시문을 사용하여 정의됩니다.

QUERY_STRING_UNESCAPED

클라이언트가 전송한 탐색 조회. 이 조회는 CGI가 HTML을 생성하지 않은 경우 정의되지 않습니다.

SSI_DIR

SSI_ROOT와 관련된 현재 파일 경로. 현재 파일이 SSI_ROOT에 있으면, 이 값은 "/"입니다.

SSI_FILE

현재 파일의 이름.

SSI_INCLUDE

현재 파일을 검색한 포함 명령에 사용된 값. 이 값은 최상의 파일로 정의되지 않습니다.

SSI_PARENT

SSI_ROOT와 관련하여 현재 파일에서 검색된 정보 포함 명령을 포함하고 있는 파일의 경로 및 파일 이름.

SSI_ROOT

최상의 파일 경로. 모든 포함 요청은 이 디렉토리나 이 디렉토리의 하위 디렉토리에 있어야 합니다.

예제:

```
<!--#echo var=SSI_DIR -->
```

exec—CGI 프로그램 지정

이 지시문을 사용하여, CGI 프로그램 출력을 포함하십시오. exec 지시문은 다음 사항을 제외하고 CGI가 출력하는 HTTP 헤더를 버립니다.

컨텐츠 유형

다른 정보에 대한 출력 본문을 구문 분석할지 여부를 판별합니다.

컨텐츠 암호화

EBCDIC로부터 ASCII로의 변환을 완료해야 하는지 여부를 판별합니다.

최종 변경

현재 값이 지정된 값보다 새로운 값이면 현재의 최종 변경 헤더 값을 바꿉니다.

cgi—CGI 프로그램 URL 지정

이 지시문을 사용하여, CGI 프로그램의 URL을 지정하십시오.

이 예제에서, **program**은 실행될 CGI 프로그램이고 **path_info** 및 **query_string**은 환경 변수로 프로그램에 전달된 하나 이상의 매개변수입니다.

```
<!--#exec cgi="/cgi-bin/program/path_info?query_string" -->
```

다음 예제는 변수의 사용을 표시합니다.

```
<!--#exec cgi="%path;&cgiprogram;%pathinfo;%querystring;" -->
```

flastmod—문서가 최종 변경된 날짜 및 시간 표시

이 지시문을 사용하여, 문서가 변경된 최종 날짜 및 시간을 표시하십시오. 이 변수의 형식은 **config timefmt** 지시문을 사용하여 정의됩니다. **file** 및 **virtual** 태그는 이 지시문과 더불어 유효하며, 의미는 다음과 같이 정의됩니다.

지시문 형식은 다음과 같습니다.

```
<!--#flastmod file="/path/file" -->
<!--#flastmod virtual="/path/file" -->
```

file 이 태그를 사용하여 파일의 이름을 지정하십시오. **flastmod**, **fsize**, **include**에 대해서, **file**이 ‘/’ 다음에 오는 경우 **SSI_ROOT**에 상대적인 것으로 추정됩니다. 그렇지 않으면, **SSI_DIR**에 상대적입니다. 지정된 파일은 **SSI_ROOT**나 하위 중 하나에 있어야 합니다. 예제는 다음과 같습니다.

```
<!--#flastmod file="/path/file" -->
```

virtual

이 태그를 사용하여 문서에 대한 가상 경로의 URL을 지정하십시오. **flastmod**, **fsize**, **include**에 대해서, **virtual**은 항상 서버의 맵핑 지시문을 통해 전달됩니다. 예제는 다음과 같습니다.

```
<!--#flastmod virtual="/path/file" -->
```

예제:

```
<!--#flastmod file="foo.html" -->
```

결과: 12May96

fsize—파일 크기 표시

이 지시문을 사용하여 지정된 파일의 크기를 표시하십시오. 이 변수를 형식 지정하는 것은 **config sizefmt** 지시문을 사용하여 지정됩니다. **file** 및 **virtual** 태그는 이 지시문과 더불어 유효하며, 의미는 이전의 **flastmod** 지시문에 정의된 것과 동일합니다.

예제:

```
<!--#fsize file="/path/file" -->
<!--#fsize virtual="/path/file" -->
```

결과: 1K

global—글로벌 변수 정의

이 지시문을 사용하여, 이 파일이나 포함된 파일이 나중에 에코할 수 있는 전체 변수를 정의하십시오.

예제:

```
<!--#global var=VariableName value="SomeValue" -->
```

예를 들면, 가상 경계를 가로질러 상위 문서를 참조하려면, 글로벌 변수 DOCUMENT_URI를 설정해야 합니다. 또한 하위 문서의 전체 변수를 참조해야 합니다. 이 예제는 상위 문서에 삽입해야 하는 HTML 코딩을 표시합니다.

```
<!--#global var="PARENT_URI" value=&DOCUMENT_URI; -->
```

이 예제는 하위 문서에 삽입해야 하는 HTML 코딩을 표시합니다.

```
<!--#flastmod virtual=&PARENT_URI; -->
```

include—출력에 문서 포함

이 지시문을 사용하여 출력에 문서의 텍스트를 포함하십시오. **file** 및 **virtual** 태그는 이 지시문과 더불어 유효하며, 의미는 위의 **flastmod** 지시문에 정의된 것과 동일합니다.

set—에코할 변수 설정

이 지시문을 사용하여, 이 파일만 나중에 에코할 수 있는 변수를 설정하십시오.

예제:

```
<!--#set var="Variable 2" value="AnotherValue" -->
```

지시문을 정의하는 동안, value 중간에서 문자열을 반복할 수 있습니다. 예제는 다음과 같습니다.

```
<!--#include file="&filename;" -->
```

변수: 서버측 설정 지시문 뒤에는 일반적으로 echo 지시문이 따라와서, 설정 변수를 탐색하고, 변수가 발견된 장소에서 에코하고, 이 기능을 진행합니다. 변수에 대한 여러 참조사항이 포함될 수 있습니다. 서버측 설정으로 이미 설정된 변수를 에코할 수 있습니다. set 변수가 발견되지 않으면, 없음이 표시됩니다

정보 포함이 서버측 설정에서 변수 참조사항을 발견하면, 서버측에서 해석하려고 합니다. 다음 예제의 두 번째 행에서 서버측 변수 &index;는 var 문자열과 함께 사용되어 변수 이름 var1을 구성합니다. 그런 다음 ê에서 &를 뺀 값을 변수 &var1;에 지정하여 변수로 인식되지 않게 합니다. 대신 이는 frêd 또는 e 위에 곡절 액센트가 붙은 fred를 작성하는 문자열로 사용됩니다. ê 변수는 클라이언트측 변수입니다.

```
<!--#set var="index" value="1" -->
<!--#set var="var&index;" value="fr&ecirc;d" -->
<!--#echo var="var1" -->
```

이스케이프된 문자(이스케이프된 변수라고 불림)는 백슬래시(\)를 앞에 두고 다음을 포함합니다.

문자	의미
\a	경보(벨)
\b	백스페이스
\f	폼 피드(새 페이지)
\n	새 행
\r	캐리지 리턴
\t	수평 탭
\v	수직 탭
\'	작은 따옴표
\"	큰 따옴표
\?	물음표
\\	백슬래시
\-	하이픈
\.	마침표
\&	앰퍼샌드

오류 메시지 사용자 정의

Caching Proxy가 리턴하는 오류 메시지를 사용자 정의할 수 있으며 특정 오류 조건에 대한 특정 메시지를 정의할 수 있습니다. 구성 및 관리 양식에서 서버 구성 -> 오류 메시지 사용자 정의를 선택하십시오. 이 양식을 사용하여 오류 조건을 선택하고 해당 조건에 사용할 특정 HTML을 지정하십시오.

프록시 구성 파일의 지시문을 편집하여 오류 메시지를 사용자 정의하려면, 228 페이지의 『ErrorPage — 특정 오류 조건에 대해 사용자 정의된 메시지 지정』 지시문에 대한 참조 섹션을 참조하십시오.

RTSP(Real Time Streaming Protocol) 경로 재지정

이는 역방향 프록시 구성에만 적용합니다.

WebSphere Application Server, 버전 6.1은 RTSP 경로 재지정기 양식으로 스트리밍 매체 지원을 소개합니다. RTSP를 사용하면 Caching Proxy가 매체 재생기와 접속하는 첫 지점으로 작동하고, 요청을 해당 프록시 서버나 요청한 매체 내용을 제공하는 콘텐츠 서버로 경로 재지정할 수 있습니다.

실시간 스트리밍 프로토콜인 RTSP로 RFC 2326에 정의되어 있습니다. RTSP는 데이터 스트림을 제어하기 위한 인터넷 표준 프로토콜입니다. 스트림 전달 기술은 없지만, 융통성이 있어서 비디오나 오디오 재생장치와 관련되지 않은 데이터 스트림을 제어하는데 사용할 수 있습니다.

RTSP 경로 재지정 정보

RTSP 경로 재지정 기능을 사용하여, Caching Proxy는 RTSP에 의해 제어된 스트리밍 매체 세션에 대한 요청을 경로 재지정할 수 있습니다. 다음과 같은 매체 유형이 포함됩니다.

- RealNetworks 레코드 오디오
- RealNetworks 레코드 비디오
- RealNetworks 라이브 스트림(오디오 및 비디오)
- Microsoft Media Player 파일
- Apple Quicktime 매체 파일

RTSP 포트(주로 554)의 프록시 서버와 접속하도록 구성될 수 있는 재생기는 Caching Proxy에서 프레임워크를 사용하여, RTSP 경로 재지정기가 요청을 처리하도록 할 수 있습니다.

RTSP 경로 재지정기는 매체 프리젠테이션을 캐시하거나 직접 프록시하지 않습니다. RTSP 경로 재지정기가 이 두 가지 기능 중 하나 또는 두 가지 기능을 모두 제공하려면 타사의 스트리밍 매체 서버와 연계하여 사용되어야 합니다. RTSP 경로 재지정기가 있는 Caching Proxy는 하나 이상의 RTSP 프록시 서버에 네트워크 액세스 가능해야 합니다.

RTSP 한계

이 기능에는 다음과 같은 한계가 있습니다.

현재는 RealNetworks 기술만 지원됩니다. 여기에는 RealProxy 프록시 서버, RealServer 기점 서버 및 RealPlayer 매체 재생기가 포함됩니다.

RTSP 향상

이전에 RTSP 경로 재지정기에는 모든 URL의 동일한 기점 서버에 대한 모든 요청이 동일한 방법으로 경로 재지정되는 한계가 있었습니다. 파일 이름이나 요청된 URL에 대한 기타 부분에 기초한 경로 재지정이 불가능했습니다. 이러한 제한사항은 더이상 적용되지 않습니다. 이제 RTSP 경로 재지정기는 수신된 요청의 완전한 URL을 Caching Proxy 구성 파일에 설정된 임계치 값(`rtsp_proxy_threshold`)과 함께 사용하여 클라이언트 요청을 기점 서버로 경로 재지정할지 프록시 서버로 경로 재지정할지를 판별합니다. 동일한 기점 서버에 대한 요청이 이제 개별적으로 처리됩니다.

RTSP 경로 재지정 구성

다음의 구성 파일 지시문은 RTSP 경로 재지정을 제어하는 데 사용됩니다. 이러한 지시문에 대한 설정은 서버를 재시작해도 새로 고쳐지지 않습니다. 서버를 완전히 정지시킨 다음 재시작해야 지시문에 대한 변경사항이 적용됩니다.

- 294 페이지의 『RTSPEnable — RTSP 경로 재지정 사용 가능』

- 294 페이지의 『rtsp_proxy_server - 경로 재지정 서버 지정』
- 295 페이지의 『rtsp_proxy_threshold — 캐시로의 경로 재지정 이전에 요청 수 지정』
- 295 페이지의 『rtsp_url_list_size — 프록시 메모리에서 URL 수 지정』

제 14 장 헤더 옵션 구성

문서를 요청할 때, 웹 클라이언트는 브라우저나 요청에 대한 추가 정보를 제공하는 헤더를 전송합니다. 헤더는 요청 전송 시 자동으로 생성됩니다.

Caching Proxy에는 사용자 정의한 헤더 정보를 대상 서버로부터 보호할 수 있는 몇 가지 옵션이 있습니다. 실제적인 헤더 대신 보다 일반적인 헤더를 쓰면 클라이언트의 익명성이 커지는 이점이 있지만, 반면에 일부 웹 페이지에 기록된 헤더 기반의 페이지 사용자 정의를 사용 불가능하게 하는 단점이 있습니다.

헤더는 일반적으로 다음 형식을 사용합니다.

사용자-에이전트: Mozilla 2.02/OS2
클라이언트-IP: 45.37.192.3
참조자: <http://www.bigcompany.com/WebTrafficExpress/main.html>

이 헤더에는 다음 필드가 있습니다.

- **사용자-에이전트:** 브라우저 및 운영 체제 정보를 제공합니다.
- **클라이언트-IP:** URL을 요청하는 클라이언트의 IP 주소를 제공합니다.
- **참조자:** 이 페이지에 대한 참조 연결의 URL과 함께 대상 서버를 제공합니다.

대부분의 헤더는 해당 프록시 구성 설정으로 차단될 수 있습니다. 그러나 일부 헤더 필드는 기점 서버에 필수이므로 헤더를 차단하면 웹 페이지가 올바르게 표시되지 않을 수 있는데 예를 들어, 특정 상황에서 "호스트" 헤더 필드를 차단하면 사용자가 잘못된 웹 페이지를 보게 될 수 있습니다. 헤더 필드에 대한 자세한 내용은 HTTP 버전 1.1 스펙을 참조하십시오.

연관된 지시문

프록시 구성 파일을 편집하여 헤더 옵션을 변경하려면, 다음 지시문에 대한 참조 섹션을 참조하십시오.

- 264 페이지의 『NoCacheOnRange — 범위 요청에 대해 캐시를 지정하지 않음』
- 265 페이지의 『NoProxyHeader — 차단시킬 클라이언트 헤더 지정』
- 285 페이지의 『ProxyFrom — 클라이언트를 From: 헤더로 지정』
- 286 페이지의 『ProxyIgnoreNoCache — 재로드 요청 무시』
- 287 페이지의 『ProxySendClientAddress — Client IP Address: 헤더 생성』
- 287 페이지의 『ProxyUserAgent — 사용자 에이전트 문자열 수정』
- 288 페이지의 『ProxyVia — HTTP 헤더의 형식 지정』

자세한 정보는 13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』을 참조하십시오.

구성 및 관리 양식

두 가지 구성 및 관리 양식을 사용하여 헤더 옵션을 지정할 수 있습니다.

- 프록시 구성 -> 프라이버시 설정을 선택하십시오. 프라이버시 설정에서 다음을 설정하십시오.

- 클라이언트 IP 주소를 대상 서버에 전달

요청하는 클라이언트의 IP 주소가 대상(컨텐츠) 서버에 전달되게 하려면 이 상자를 체크하십시오. 이 상자를 체크하지 않으면, 대상 서버가 프록시 서버의 IP 주소를 수신합니다. 이 상자를 체크하지 않으면, 웹을 서핑하는 동안 클라이언트의 익명성이 증가합니다.

- 사용자-에이전트 문자열

헤더에서 대상 서버로 전송할 문자열을 입력하여 클라이언트가 사용 중인 브라우저와 운영 체제의 유형을 바꾸십시오. 예를 들어, Caching Proxy 4.0을 지정하면 다음 헤더에서 Mozilla 2.02/OS2를 변경합니다.

내용-유형:MIME

사용자-에이전트: Mozilla 2.02/OS2

참조자: <http://www.ics.raleigh.ibm.com/WebTrafficExpress/main.html>

Pragma: 캐시하지 않음

- From:

"Form:" 헤더를 분석할 때 대상 서버가 읽을 전자 우편 주소를 입력하십시오. 관리자가 문제점 보고서를 수신할 사람이기 때문에, 프록시 관리자의 전자 우편 주소를 지정할 수 있습니다.

- 제출을 눌러 구성 파일에 변경사항을 작성하십시오.

- 프록시 구성 -> 프록시 헤더 필터링을 선택하십시오. 이 양식을 사용하여 HTTP 헤더를 블록에 나열하십시오.

1. 추가 또는 제거를 누르고 차단된 헤더의 색인 위치를 지시하십시오.
2. 차단할 클라이언트 HTTP 헤더를 입력하십시오. (헤더의 완료 목록과 설명은 HTTP 1.1 스펙을 참조하십시오.)
3. 제출을 눌러 구성 파일에 변경사항을 작성하십시오.

자세한 정보는 7 페이지의 제 2 장 『구성 및 관리 양식 사용법』을 참조하십시오.

제 15 장 API(application programming interface) 정보

API(application programming interface)는 *Edge Components*용 프로그래밍 안내서에 자세히 설명되어 있습니다. 구성 파일 내의 API 지시문에서 요청 처리 워크플로우 내의 특정 단계 중에서 호출되는 플러그인 루틴이 사용 가능합니다. 내장 루틴뿐 아니라 이 플러그인 루틴도 바꾸거나 실행할 수 있습니다.

연관된 지시문

다음은 API 지시문입니다.

- 199 페이지의 『Authentication — 인증 단계 사용자 정의』
- 200 페이지의 『Authorization — 권한 부여 단계 사용자 정의』
- 227 페이지의 『Error — 오류 단계 사용자 정의』
- 250 페이지의 『Log — 로그 단계 사용자 정의』
- 260 페이지의 『Midnight — 로그 보존에 사용되는 API 플러그인 지정』
- 261 페이지의 『NameTrans — 이름 변환 단계 사용자 정의』
- 266 페이지의 『ObjectType — 오브젝트 유형 단계 사용자 정의』
- 273 페이지의 『PostAuth — PostAuth 단계 사용자 정의』
- 273 페이지의 『PostExit — PostExit 단계 사용자 정의』
- 274 페이지의 『PreExit — PreExit 단계 사용자 정의』
- 299 페이지의 『ServerInit — 서버 초기설정 단계 사용자 정의』
- 300 페이지의 『ServerTerm — 서버 종료 단계 사용자 정의』
- 300 페이지의 『Service — 서비스 단계 사용자 정의』
- 310 페이지의 『Transmogriifier — 데이터 조작 단계 사용자 정의』
- 310 페이지의 『TransmogriifiedWarning — 클라이언트에게 경고 메시지 전송』

자세한 정보는 13 페이지의 제 4 장 『ibmproxy.conf 파일 직접 편집』을 참조하십시오.

구성 및 관리 양식

다음 구성 및 관리 양식은 연관된 지시문의 값을 편집합니다.

- 서버 구성 -> 요청 처리 -> API 요청 처리

자세한 정보는 7 페이지의 제 2 장 『구성 및 관리 양식 사용법』을 참조하십시오.

제 4 부 프록시 서버 캐시 구성

이 섹션에서는 프록시 캐시 및 이를 구성하는 방법을 검토합니다. 파일을 메모리(메모리 캐시) 또는 둘 이상의 저장영역 장치(디스크 캐시)에 저장하도록 캐시를 설정할 수 있습니다. 캐시 새로 고침 에이전트는 캐시로 자주 요청되는 파일을 사전 로드하도록 구성할 수 있습니다. 또한 다양한 URL 필터를 캐시에 적용할 수 있습니다. 이 섹션에서는 원격 캐시 액세스 또는 ICP(Internet Caching Protocol) 플러그인을 사용한 캐시 공유, 캐시 가비지 콜렉션을 사용하여 오래된 파일 제거, 동적 생성 파일 캐시에 대해서도 설명합니다.

이 파트에는 다음과 같은 장이 들어 있습니다.

- 79 페이지의 제 16 장 『프록시 서버 캐시 개요』
- 83 페이지의 제 17 장 『기본 캐시 구성』
- 89 페이지의 제 18 장 『캐시되는 내용 제어』
- 93 페이지의 제 19 장 『캐시 콘텐츠 유지』
- 99 페이지의 제 20 장 『자동 새로 고침 및 사전 로드에 대한 캐시 에이전트 구성』
- 107 페이지의 제 21 장 『공유 캐시 사용』
- 111 페이지의 제 22 장 『동적 생성 콘텐츠 캐시』
- 115 페이지의 제 23 장 『프록시 서버 캐시 조정』

제 16 장 프록시 서버 캐시 개요

캐시는 프록시 서버가 클라이언트가 요청하는 파일의 로컬 사본을 저장하여, 동일한 클라이언트나 기타 클라이언트로부터 파일을 다시 요청받을 때 캐시에서 직접 제공할 수 있는 기능입니다.

Caching Proxy는 HTTP 1.1을 따르며, 일반적으로 캐시를 하고 문서의 최신 정보를 판별하는 데 HTTP 1.1 프로토콜을 따릅니다.

이 장에서는 일부 프록시 서버 캐시의 역할을 다룹니다. 구성할 수 있는 기능의 경우, 적절한 값을 설정하는 방법에 대한 세부사항은 다음 장에 포함되어 있습니다.

캐시 저장영역

프록시 서버는 물리적 저장영역 장치 또는 시스템 메모리에 캐시를 저장할 수 있습니다. 시스템에 어떤 캐시 저장영역 유형이 더 좋은지는 하드웨어의 성능과 빠른 캐시 응답이 중요한지 아니면 캐시에 많은 수의 항목을 저장하는 것이 중요한지에 따라 다릅니다. 일반적으로 메모리 캐시에 대한 응답 시간은 디스크 캐시 응답 시간보다 빠르지만 메모리 캐시의 크기는 프록시 서버 시스템의 RAM 용량으로 제한되어 있습니다. 디스크 캐시의 크기는 저장영역 장치의 크기로 제한되며 일반적으로 RAM 용량보다는 훨씬 큼니다.

디스크 캐시의 경우, Caching Proxy는 공 디스크 캐시를 사용하는데 이는 프록시 서버가 운영 체제의 읽기 및 쓰기 프로토콜을 사용하지 않고 직접 캐시 장치에 기록함을 의미합니다. 디스크 캐시용 저장영역 장치는 **htcformat** 명령을 사용하여 준비되어 있어야 합니다. **htcformat**에 대한 정보는 83 페이지의 제 17 장『기본 캐시 구성』 섹션에 들어 있습니다.

캐시 색인

Caching Proxy는 메모리 또는 디스크 캐시 모두에서 시스템 메모리 영역을 사용하여 캐시 색인을 보유합니다. 이로써 캐시된 파일을 찾는 데 걸리는 처리 시간을 줄일 수 있습니다.

Caching Proxy의 캐시 디렉토리 구조 및 조회 방법은 다른 프록시 서버와 다릅니다. Caching Proxy는 캐시에 있는 파일에 대한 정보와 함께 색인을 메모리 속에 유지합니다. 조회할 때 디스크나 다른 매체 대신 RAM을 사용하면 보다 빨리 파일을 조회하고 검색할 수 있습니다.

색인에는 URL, 캐시 위치 및 캐시된 오브젝트에 대한 만기 정보가 들어 있습니다. 이러한 이유로, 색인을 보유하는 데 필요한 메모리 용량은 캐시의 오브젝트 수에 비례합니다.

클라이언트에서 요청이 수신될 때, 프록시가 해당 URL에 대한 메모리에서 캐시 색인을 확인합니다.

- 파일이 색인에 없는 경우, 대상 서버로 요청합니다.
 - 그런 다음 검색된 파일이 캐시될 수 있는지 여부를 판별하기 위해 URL을 확인합니다. 허용된 경우, 프록시 서버는 검색된 파일을 캐시합니다.
 - 그러면 캐시 색인이 URL, 위치 및 새로 캐시된 오브젝트에 대한 만기 정보로 갱신됩니다.
- 파일이 색인에 있는 경우,
 - 만기 정보를 확인하여 캐시된 파일이 새로운 파일인지 판별합니다.
 - 오브젝트가 만기된 경우, 대상 서버가 접속되며 만기 오브젝트는 새로 검색된 문서로 바꿉니다. 만기 정보가 캐시 색인에서 갱신됩니다.
 - 오브젝트가 만기되지 않았으면, 프록시 캐시에서 문서가 제공됩니다.

FTP 캐시

프록시가 요청을 캐시하도록 구성되면, FTP 파일 요청과 HTTP 파일 요청을 캐시할 수 있습니다. 그러나 FTP 파일에는 HTTP 파일과 동일한 유형의 헤더 정보가 들어있지 않기 때문에, 캐시된 FTP 파일의 만기 날짜는 기타 캐시된 파일과 다르게 계산됩니다.

파일을 검색하기 위해서 FTP 서버에 요청이 이루어지면, 프록시는 우선 파일에 대한 FTP 디렉토리 정보를 얻기 위해서 FTP 서버에 파일의 LIST 요청을 전송합니다. FTP 서버가 LIST 요청에 대해서 긍정적인 완료 응답과 파일에 대한 디렉토리 정보를 보내오면, 프록시는 FTP 디렉토리 정보에서 구문 분석된 날짜로 HTTP 최종 변경 헤더를 작성합니다. 그러면 프록시 캐시 기능은 구성 파일의 CacheLastModifiedFactor 지시문 설정 값과 최종 변경 헤더를 사용하여 FTP 파일이 만기 전에 캐시에 남아있을 기간을 판별합니다.

최종 변경 헤더와 CacheLastModifiedFactor 지시문이 파일이 캐시에 남아있을 기간을 판별하는 데 사용되는 방법에 관한 자세한 정보는 93 페이지의 제 19 장 『캐시 콘텐츠 유지』를 참조하십시오.

anonymous 로그인인 아니라 고유한 사용자 ID로 검색된 FTP 파일은 개인용 파일로 간주되며 캐시되지 않습니다.

DNS 캐시

웹 콘텐츠의 캐시 이외에 프록시 서버는 DNS(Domain Name Server) 캐시를 수행합니다. 예를 들어, 클라이언트가 `www.myWebsite.com`에서 URL을 요청하면, 프록시는 DNS 서버에 `www.myWebsite.com` 호스트 이름을 IP 주소로 해석하도록 요청합니다. 그러면 IP 주소가 캐시되어 이 호스트 이름에 대한 후속 요청의 응답 시간을 줄여줍니다. DNS 캐시는 자동으로 실행되며 재구성할 수 없습니다.

캐시 제외

일부 파일과 문서는 캐시되지 않습니다. 캐시되지 않는 파일은 다음과 같습니다.

- GET 이외의 HTTP 메소드를 사용하여 요청에서 리턴된 파일(예: POST 및 PUT)
- 기점 서버가 문서 캐시를 특정하게 허용하지 않는 경우, 인증이 필요한 문서
- CGI 스크립트의 동적 출력(이는 요청될 때마다 고유하기 때문임) 동적 캐시가 사용 가능한 경우 IBM WebSphere Application Server가 실행한 Servlet 및 JSP(JavaServer Pages)로부터 동적으로 생성된 결과를 캐시할 수 있습니다. 자세한 내용은 111 페이지의 제 22 장 『동적 생성 콘텐츠 캐시』를 참조하십시오.
- SSL 터널링 연결로 전달된 정보(프록시가 이 연결을 통해 전달된 데이터를 암호 해독할 수 없기 때문임)
- 조회 캐시를 특정하게 허용하지 않는 경우, 물음표(?)를 포함하는 URL에서 모든 파일이 리턴됩니다. (조회 결과 캐시 구성 정보는 89 페이지의 제 18 장 『캐시되는 내용 제어』를 참조하십시오.)

캐시 필터를 설정하여 캐시된 항목을 좀더 제한할 수 있습니다. 예를 들어, 프록시 서버가 프록시에서 로컬로 제공하는 파일을 캐시하지 못하게 할 수 있습니다. 자세한 내용은 89 페이지의 제 18 장 『캐시되는 내용 제어』를 참조하십시오.

캐시 관리

캐시 관리에는 많은 요소가 연관되어 있습니다. 서버 관리자로서 다음 사항을 지정할 수 있습니다.

- 캐시되는 문서(자세한 정보는 89 페이지의 제 18 장 『캐시되는 내용 제어』 참조)
- 캐시될 수 있는 문서의 수(자세한 정보는 83 페이지의 제 17 장 『기본 캐시 구성』 참조)
- 캐시된 문서가 현재로 인정되는 기간(자세한 정보는 93 페이지의 제 19 장 『캐시 콘텐츠 유지』 참조)
- 캐시를 제거하는 빈도(가비지 콜렉션) 및 보존하고자 하는 파일의 유형(자세한 정보는 93 페이지의 제 19 장 『캐시 콘텐츠 유지』 참조).

- 캐시된 문서를 색인화하는 방법(자세한 정보는 83 페이지의 제 17 장 『기본 캐시 구성』 참조).
- 캐시를 새로 고치는 시기(자세한 정보는 99 페이지의 제 20 장 『자동 새로 고침 및 사전 로드』에 대한 캐시 에이전트 구성』 참조).
- 원격 캐시 액세스(자세한 정보는 107 페이지의 제 21 장 『공유 캐시 사용』 참조)
- 로그를 보존 및 아카이브하는 방법(자세한 정보는 83 페이지의 제 17 장 『기본 캐시 구성』 참조).

또한 Caching Proxy의 전체 성능을 향상시키기 위해 캐시 구성을 조정할 수 있습니다. 성능 조정에 대한 자세한 정보는 115 페이지의 제 23 장 『프록시 서버 캐시 조정』을 참조하십시오.

제 17 장 기본 캐시 구성

Edge Components 제품 설치 프로그램에서 기본 설정을 사용하여 Caching Proxy 캐시를 설치하면, 캐시가 사용 가능하게 되며 캐시는 메모리에 저장됩니다. 시스템의 요구에 맞게 캐시를 사용자 정의하기 위해 다음과 같은 기본 캐시 설정을 조정하고자 할 수 있습니다.

설치 프로그램을 사용하지 않을 경우 캐시가 사용 가능하도록 이 설정을 구성하십시오.

캐시를 구성하는 데 필요한 기본 단계는 다음과 같습니다.

1. 캐시 사용 가능
2. 캐시 저장영역 구성

기본 캐시 설정을 구성한 후 다음 기능에 대한 설정을 추가하거나 변경할 수 있습니다.

- 캐시 메모리 사용자 정의
- 디스크에 캐시 메모리 저장 또는 로드
- URL 필터를 사용하여 캐시될 내용 제한
- 조회 결과 또는 동적으로 생성된 파일에 대해 캐시를 사용 가능하게 하여 캐시될 내용 펼치기
- 캐시된 파일 만기 또는 가비지 콜렉션 구성
- 자동 캐시 새로 고침 및 사전 로드 구성
- RCA(Remote Cache Access) 또는 ICP(Internet Caching Protocol)와 공유하는 캐시 구성
- 로깅 구성

이 장에 각 설정 변경에 대한 명령이 제공되거나 참조됩니다.

1. 캐시 사용 가능

캐시를 사용 가능하게 하려면, Caching 지시문을 on으로 설정하거나 또는 캐시 구성 -> 캐시 설정 구성 양식에서 프록시 캐시 사용 가능 상자를 선택하십시오. 캐시 장치를 지정하지 않으면 캐시가 메모리에 저장됩니다. 디스크 캐시를 작성하려면 『2. 캐시 저장영역 구성』의 단계를 따르십시오.

2. 캐시 저장영역 구성

캐시 저장영역 구성 타스크는 메모리 캐시를 사용하는지 또는 디스크 캐시를 사용하는지에 따라 다릅니다.

메모리 캐시를 사용하려면 캐시의 콘텐츠를 저장하기에 충분한 메모리를 포함하도록 캐시 메모리 설정을 사용자 정의하십시오. 권장 캐시 메모리 크기는 85 페이지의 『캐시 메모리 설정』을 참조하십시오.

디스크 캐시를 사용하려면 다음 사항을 수행하십시오.

1. 캐시를 보유할 저장영역을 준비하십시오.

캐시는 특별히 포맷된 장치가 필요합니다. 전체 장치나 디스크 파티션을 캐시 전용으로 사용하는 것이 권장됩니다. 캐시의 최소 크기는 16392KB입니다.

캐시 장치를 포맷하려면 다음을 수행하십시오.

- a. 캐시를 보유할 장치를 선택하십시오. 다른 프로그램이 이 저장영역을 사용할 수 없으며, 미처리(또는 문자 포맷된)된 장치로서 장치에 액세스할 수 있습니다.
- b. **htcformat** 명령을 사용하여 장치를 포맷하십시오. 구문은 다음과 같습니다.

```
htcformat raw_device_path [-blocksize block_size] [-blocks number_of_blocks]
```

-blocksize 및 -blocks 인수는 선택적입니다. 기본 블록 크기는 8192바이트입니다. 블록 수를 지정하지 않으면, 디스크 파티션은 포함할 수 있는 블록의 최대 수로 채워집니다.

장치 경로를 지정할 때, 반드시 미처리된 장치의 경로를 지정하십시오.

- AIX 플랫폼에서, /dev/lv02로 정의된 논리 볼륨에 대한 미처리된 장치의 경로는 /dev/rlv02입니다.
- Linux 플랫폼에서, **htcformat**를 실행하기에 앞서 **raw** 명령을 첫번째로 실행해야 합니다.

```
raw /dev/raw/raw1 dev/sdb1
```

- HP-UX 및 Solaris 플랫폼에서 /dev/dsk/c0t0d0s0로 정의된 파티션에 대한 미처리 장치 경로는 /dev/rdisk/c0t0d0s0입니다.
- Windows 플랫폼에서, e:로 정의된 장치의 미처리된 장치의 경로는 \\.\e:입니다.

미처리된 장치에 액세스하기 위한 추가 정보는 파일 시스템에 대한 참조 자료를 참조하십시오.

2. CacheDev 지시문 또는 캐시 설정 구성 양식을 사용하여 캐시 장치를 지정하십시오. 한 개 이상의 장치를 지정할 수 있습니다.

주의:

Windows 시스템에서, **htcformat** 명령은 캐시 장치를 쓰기 불가능으로 자동으로 표시하지 않습니다.

운영 체제에서 캐시 장치에 쓰기를 시도하면, 캐시된 데이터가 손실될 수 있습니다. 이러한 결과를 피하기 위해서 **htcformat** 명령을 사용하기 전에 Windows 디스크 관리자 유틸리티를 사용하여 디스크를 준비할 수 있습니다. 디스크를 준비하려면 디스크 유틸리티를 사용하여 사용할 장치 또는 파티션을 삭제한 후에 이를 형식화하지 않고 재작성하십시오. 이렇게 하여, 시스템 저장영역으로 사용할 수 없는 장치를 검토합니다.

선택적 사용자 정의

캐시 메모리 설정

CacheMemory 지시문(캐시 설정값 구성 양식의 캐시 메모리 필드)에 값을 설정하려면 다음 원칙에 따르십시오. 이 값의 총 메모리 세트는 캐시 색인을 포함한 캐시 기반 구조 지원에 사용되며, 메모리 캐시가 구성된 경우에는 캐시 콘텐츠 저장에 사용됩니다.

최소값

디스크 캐시의 최적의 성능을 위하여, 캐시 색인을 포함한 캐시 하부 구조 지원에 64MB의 최소 캐시 메모리 값을 권장합니다. 캐시 크기가 증가되면, 캐시 색인이 증가되고 색인을 저장하기 위한 추가 캐시 메모리가 필요하게 됩니다. 64MB의 캐시 메모리 값은 캐시 하부 구조 지원을 제공하고 최대 6.4GB의 디스크 캐시에 대한 캐시 색인을 저장하는 데 충분합니다. 보다 큰 디스크 캐시의 경우, 캐시 메모리는 캐시 크기의 1%여야 합니다.

메모리 캐시의 경우, 캐시 메모리 값은 캐시 하부 구조 지원 및 캐시 자체에 대하여 별도로 설정되는 메모리의 양입니다. 64MB의 최소 캐시 메모리 값을 권장합니다.

최대값

메모리 캐시에 실제 메모리를 너무 많이 할당하면 "메모리 부족" 오류 또는 프록시 서버 장애와 같은 바람직하지 못한 조작이 발생할 수 있습니다. 캐시 메모리의 값 제한은 32비트 응용프로그램의 제한 때문입니다. Caching Proxy는 32비트 응용프로그램으로 최대 2GB의 메모리를 사용할 수 있습니다.

Caching Proxy는 CacheMemory 지시문이 정의하는 메모리를 할당하고 이를 캐시로 사용하여 오브젝트를 저장합니다. 메모리 캐시 또는 공 디스크 캐시 여부에 상관없이 캐시와 네트워크 I/O, 연결 버퍼와 세션 버퍼 및 기본 프로세스와 모든 스레드 메모리에 대한 데이터 구조에 추가 메모리를 할당해야 합니다. 또한 일부 클라이언트의 요청에 기본보다 큰 메모리 풀 블록 할당이 필요할 수 있습니다. 따라서 CacheMemory 지시문을 2GB 표시에 근접하게 설정하면 특히 요청 로드가 큰 경우, Caching Proxy의 조작 메모리가 충분하지 않을 수 있습니다.

CacheMemory 지시문의 값은 1600MB 이하로 설정하는 것이 바람직합니다. 1600MB 보다 큰 값을 설정하면 Caching Proxy의 정상 운영에 필요한 메모리와 충돌하여 역효과를 발생시킬 수 있습니다. 이 역효과에는 일반적으로 CPU 사용 증가(100%까지 증가 가능), 메모리 부족 오류 및 성능 저하가 포함되며 다른 역효과도 발생할 수 있습니다. 전체적으로 더 큰 캐시 크기가 필요한 경우에는, 캐시 장치를 사용하거나 RCA 또는 ICP로 공유 캐시 구성을 수행하십시오.

디스크에 캐시 메모리 저장 또는 로드

덤프 파일에 양방향으로 캐시 콘텐츠를 가져오거나 내보낼 수 있습니다. 이는 캐시 메모리가 재시작 중에 유실되거나 다중 프록시에 대해 동일 캐시를 전개하는 경우에 유용합니다.

캐시 필터 설정

필터는 URL 요청 양식을 일치시켜 캐시되는 내용을 제한할 수 있습니다. 필터 설정에 대한 자세한 정보는 89 페이지의 제 18 장 『캐시되는 내용 제어』를 참조하십시오.

조회 결과 및 동적으로 생성된 파일에 대한 캐시 구성

선택적으로 조회 요청 결과를 캐시하도록 프록시 서버를 구성할 수 있습니다. 기본값으로 물음표(?)를 포함하는 URL은 캐시되지 않습니다. 자세한 내용은 90 페이지의 『캐시 조회 응답』을 참조하십시오.

다른 옵션은 IBM WebSphere Application Server로부터의 JSP 실행 또는 servlet 결과를 캐시하는 것입니다. 자세한 내용은 111 페이지의 제 22 장 『동적 생성 콘텐츠 캐시』를 참조하십시오.

파일 만기 구성 및 가비지 컬렉션

캐시 내의 파일 만기 시기 구성 및 오래된 파일 제거 방법에 대한 정보는 93 페이지의 제 19 장 『캐시 콘텐츠 유지』의 내용을 참조하십시오.

자동 사전 로드 구성

요청하기 전에 매일 가장 많이 사용하는 파일을 새로 고치도록 캐시를 구성할 수 있습니다. 정보는 99 페이지의 제 20 장 『자동 새로 고침 및 사전 로드에 대한 캐시 에이전트 구성』을 참조하십시오.

캐시 공유 구성

특정 환경에서 공유 캐시를 사용하면 캐시에서 요청된 파일을 찾을 수 있는 가능성이 증가합니다. 정보는 107 페이지의 제 21 장 『공유 캐시 사용』을 참조하십시오.

로깅 구성

Caching Proxy를 관리하려면 간결하고 정확한 로그를 유지하는 것이 중요합니다. 157 페이지의 제 6 부『Caching Proxy 모니터링』에 프록시 서버 로그 구성 및 사용에 대한 정보가 포함되어 있습니다.

제 18 장 캐시되는 내용 제어

Caching Proxy는 캐시될 파일, 문서 및 기타 오브젝트를 제어하기 위한 몇 가지 필터링 방법을 제공합니다. 다음과 같은 기능이 포함됩니다.

- URL 기반 캐시 필터
- 조회 응답 캐시
- 로컬로 제공된 파일 캐시
- 부분 URL 기반 캐시
- 요청 URL의 파트를 기본으로 파일 캐시
- 동적 생성 파일 캐시 — 111 페이지의 제 22 장 『동적 생성 콘텐츠 캐시』 참조

주: 캐시 구성 -> 캐시 작동 구성 및 관리 양식에는 수신 **URL** 기반 캐시로 레이블된 옵션이 있습니다(해당 구성 파일 지시문의 이름은 CacheByIncomingURL입니다). 이 지시문은 캐시된 파일의 파일 이름을 참조합니다. 이 상자를 선택하여 수신 URL 상에 캐시된 파일의 파일 이름을 기초로 하십시오. 이 상자를 선택하지 않으면 파일 이름은 전송 URL의 이름을 기초로 합니다.

URL 기반 캐시 필터 구성

파일을 캐시할지 여부를 판별하기 위해 요청을 URL 템플리트와 비교하도록 프록시 서버를 구성할 수 있습니다. 이 기능은 파일이 항상 캐시되는 요청에 대한 템플리트와 파일이 캐시되지 않는 요청에 대한 별도의 템플리트를 설정하여 구성됩니다. 여러 개의 템플리트를 사용할 수 있습니다.

조회 응답 캐시를 사용 가능하게 하는 데도 유사한 시스템이 사용됩니다. 정보는 90 페이지의 『캐시 조회 응답』을 참조하십시오.

ibmproxy.conf 파일을 설정하여 URL 캐시 필터를 설정하려면 209 페이지의 『CacheOnly — 템플리트와 일치하는 URL이 있는 파일만 캐시』 및 262 페이지의 『NoCaching — 템플리트와 일치하는 URL이 있는 파일을 캐시하지 않도록 지정』의 내용을 참조하십시오.

구성 및 관리 양식에서 URL 캐시 필터를 설정하려면 캐시 구성 -> 캐시 작동: **URL** 별로 캐시 필터링 필드를 사용하십시오. 이 섹션을 사용하여 파일이 항상 캐시되는 URL을 지정하거나 파일이 캐시되지 않는 URL을 지정하십시오. 항상 캐시해야 할 파일과 캐시하지 말아야 할 파일의 두 목록을 지정하려면, 하나의 목록을 작성한 후 다른 목록을 작성하기 전에 제출을 누르십시오.

캐시 조회 응답

조회(물음표를 포함하는 URL 요청)로부터 리턴된 응답은 캐시 필터링을 사용하여 캐시될 수 있습니다. 이 기능은 많은 클라이언트가 동일한 조회 요청을 작성하는 경우 역방향 프록시(대리) 시나리오에서 유용합니다.

조회 구성은 ibmproxy.conf 구성 파일에서 CacheQueries 지시문을 편집하여 구성할 수 있습니다. CacheQueries 지시문의 옵션은 다음과 같습니다.

- Always — HTTP 1.1 표준으로 캐시할 수 있는 경우, 템플리트와 일치하는 호스트로부터의 모든 조회 응답을 캐시합니다.
- Public — "Cache-control: public" 헤더 또는 강제 유효성 재확인 헤더를 포함하고 있고 HTTP 1.1 표준으로 캐시할 수 있는 경우, 템플리트와 일치하는 호스트로부터의 조회 응답을 캐시합니다.

이 옵션에 대한 추가 정보는 210 페이지의 『CacheQueries — 물음표(?)를 포함하는 URL에 캐시 응답 지정』을 참조하십시오.

구성 및 관리 양식에서 조회 응답 캐시를 구성하려면 캐시 구성 -> 캐시 작동: URL 별로 캐시 조회 응답 필터링 필드를 사용하십시오. 두 개의 목록을 지정하려면, 하나의 목록을 작성한 후 다른 목록을 작성하기 전에 제출을 누르십시오.

조회 응답 캐시에 대한 추가 요구사항

조회 캐시 설정 구성 외에, 다음 설정을 올바르게 구성하여 조회 응답이 캐시되도록 하십시오. 구성 및 관리 양식을 사용하여 이 옵션을 설정하는 방법에 대한 정보는 96 페이지의 『캐시 최신 정보 구성』을 참조하십시오.

- CacheTimeMargin — 이 지시문은 최소 만기 시간을 지정합니다. 최소 시간보다 만기 시간이 짧은 파일은 캐시되지 않습니다. 때로 조회 응답의 만기 시간이 매우 짧기 때문에, 이 지시문을 낮은 설정으로 설정하면 보다 많은 조회 응답을 캐시할 수 있습니다. 211 페이지의 『CacheTimeMargin — 파일 캐시를 위한 최소 수명 지정』의 내용을 참조하거나 96 페이지의 『캐시 최신 정보 구성』에 설명되어 있는 캐시 만기 설정 양식을 사용하십시오.
- CacheDefaultExpiry — 이 지시문은 만기 날짜가 정확하지 않거나 만기 시간을 계산할 최종 변경 날짜가 없는 파일의 만기 시간을 지정합니다. 이 HTTP 요청 설정을 기본값 0에서 증가시키면 보다 많은 조회 응답을 캐시할 수 있습니다. 그러나 이러한 방법으로 설정을 변경하면 만기된 콘텐츠를 캐시에서 제공할 위험도 증가합니다. 203 페이지의 『CacheDefaultExpiry — 파일에 대한 기본 만기 시간 지정』의 내용을 참조하거나 96 페이지의 『캐시 최신 정보 구성』에 설명되어 있는 캐시 만기 설정 양식을 사용하십시오.
- CacheLastModifiedFactor — 이 지시문은 날짜를 최종 변경했으나 만기 날짜가 정확하지 않은 파일의 만기 날짜를 계산하는 데 사용됩니다. HTTP 파일 요소 값을 더 크게 설정하면 유효성 재검증 없이 HTTP 파일이 캐시에 상주하는 시간을 증가시키

게 됩니다. 이러한 방법으로 설정을 변경하면 만기된 콘텐츠를 캐시에서 제공할 위험 또한 증가합니다. 205 페이지의 『CacheLastModifiedFactor — 만기 날짜 판별을 위한 값 지정』의 내용을 참조하거나 96 페이지의 『캐시 최신 정보 구성』에 설명되어 있는 최종 수정 요소 양식을 사용하십시오.

- 선택적으로, SignificantUrlTerminator 지시문 및 AggressiveCaching 지시문을 설정하십시오. 301 페이지의 『SignificantURLTerminator — URL 요청의 종료 코드 지정』 및 197 페이지의 『AggressiveCaching — 캐시할 수 없는 파일에 대한 캐시 지정』을 참조하십시오.

로컬로 제공된 파일 캐시

일반적으로 프록시 서버에서 제공하는 파일을 캐시하는 것은 비효율적이므로 기본적으로 서버의 로컬 도메인이 기점인 파일은 캐시되지 않습니다. 서버의 로컬 도메인이 기점인 오브젝트를 캐시하려면 캐시 구성 → 캐시 작동 구성 및 관리 양식에서 로컬 도메인 파일 캐시 상자를 선택하십시오. 아니면 프록시 구성 파일의 CacheLocalDomain 지시문을 on으로 설정하십시오.

부분 URL로 파일 캐시

전체 URL 대신 수신 URL의 지정된(중요한) 부분만을 기반으로 항목을 캐시할 수 있습니다. 수신 요청 URL의 중요한 부분이 동일한 경우, 수신 요청이 다양해서 동일한 응답이 자주 리턴되기 때문에 이 기능은 트랜잭션 모델 웹 서비스나 동적 캐시에 유용합니다.

구성 및 관리 양식을 사용하여 부분 URL 기반의 캐시를 지정할 수 없습니다. 그 대신, 프록시 구성 파일에 있는 SignificantUrlTerminator 지시문을 사용하여 URL 요청에 대한 종료 코드를 지정하십시오. 종료 코드를 지정하면 요청 처리 및 요청 파일의 캐시 여부 판별 시, 종료 코드 앞의 문자만을 Caching Proxy가 평가할 수 있습니다. 종료 코드가 하나 이상 정의되면, Caching Proxy는 수신 URL을 종료 코드와 비교하여 ibmproxy.conf 파일에서 정의됩니다. 301 페이지의 『SignificantURLTerminator — URL 요청의 종료 코드 지정』에서 자세한 정보를 참조하십시오.

관련 구성 파일 지시문

프록시 구성 파일을 직접 편집하여 캐시 필터를 설정하려면, 참조 섹션에서 다음 지시문을 참조하십시오.

- 262 페이지의 『NoCaching — 템플리트와 일치하는 URL이 있는 파일을 캐시하지 않도록 지정』
- 209 페이지의 『CacheOnly — 템플리트와 일치하는 URL이 있는 파일만 캐시』
- 210 페이지의 『CacheQueries — 물음표(?)를 포함하는 URL에 캐시 응답 지정』

- 206 페이지의 『CacheLocalDomain — 로컬 도메인을 캐시할지 여부 지정』
- 301 페이지의 『SignificantURLTerminator — URL 요청의 종료 코드 지정』

캐시될 수 없는 문서에 대한 정보는 79 페이지의 제 16 장 『프록시 서버 캐시 개요』를 참조하십시오.

제 19 장 캐시 콘텐츠 유지

캐시는 저장된 파일의 사본 작성 및 저장을 포함하므로 캐시가 제대로 작동하기 위해서는 약간의 유지보수가 필요합니다. 캐시된 파일은 이제 기점 서버에 있는 파일과 일치하지 않기 때문에 신규인지와 무효화 여부를 반드시 확인해야 합니다. 이 파일 만기 프로세스는 『파일 만기』에 설명되어 있습니다. 또한 캐시에서 무효화되었거나 사용하지 않는 파일을 제거하여 새 파일에 대한 공간을 만들어야 합니다. 이러한 캐시 제거 프로세스는 98 페이지의 『가비지 콜렉션』에 설명되어 있습니다.

파일 만기

콘텐츠 서버의 원래 오브젝트와 일치하도록 캐시된 오브젝트를 유지하는 것을 캐시 최신 정보 유지라고 합니다. 캐시하는 각 문서 또는 기타 오브젝트의 경우, Caching Proxy는 오브젝트가 만기될 시간을 계산합니다.

HTTP 페이지의 경우, 콘텐츠 서버가 생성한 문서의 헤더에 만기 정보가 있습니다.

FTP 프로토콜에 상응하는 만기 정보가 없기 때문에, Caching Proxy는 각 파일의 FTP 디렉토리 정보를 기반으로 FTP 파일에 대한 자체의 Last-Modified: 헤더를 생성하고, 이 정보를 사용하여 만기 시간을 계산합니다. 프록시 서버가 FTP 서버에서 파일에 대한 디렉토리 정보를 얻을 수 없으면, FTP URL과 일치하는 기본값이 사용됩니다. 또한 FTP 서버의 표준 날짜 형식이 없기 때문에, Caching Proxy가 일부 FTP 서버에서 전송한 날짜 및 시간을 인식하지 못할 수 있습니다. 이런 경우에는 프록시 서버의 기본 만기 시간 값을 사용합니다. 그러면 프록시는 비슷한 방식으로 HTTP 페이지 및 FTP 파일의 캐시를 관리할 수 있습니다.

콘텐츠 서버가 다음과 같은 몇 가지 방법(선호 사항 순서) 중 하나로 만기를 지정할 수 있습니다.

1. 콘텐츠 서버가 Cache-control: s-maxage=*n*이라는 헤더를 지정합니다. 이는 프록시에게 오브젝트가 수신된 이후 *n*초 동안 최신 상태임을 알려줍니다.
2. 콘텐츠 서버가 Cache-control: max-age=*n*이라는 헤더를 지정합니다. 이는 프록시에게 오브젝트가 수신된 이후 *n*초 동안 최신 상태임을 알려줍니다.
3. 콘텐츠 서버가 Expires: *n*이라는 헤더를 지정합니다. 이는 프록시에게 오브젝트가 *n*에 의해 지정된 시간까지 최신 상태임을 알려줍니다.
4. 콘텐츠 서버가 Last-Modified: *n* 헤더를 사용하여 문서가 최종 변경되었음을 알려줍니다. 프록시 서버는 문서가 최종 수정된 이후 경과된 시간을 계산하고, 이 시간을 프록시 구성 파일에 설정된 캐시 최종 변경 요소와 곱하여, 이 기간 동안 문서가 유효하다고 가정합니다. 예를 들어, 콘텐츠 서버에서 문서가 1주일(7일) 전에

최종 수정되어 최종 수정 요소가 0.14라고 보고했으면, 프록시 서버는 문서가 약 1 일 동안 유효할 것으로 받아들입니다. 캐시 최종 수정 요소 설정에 대한 명령은 96 페이지의 『캐시 최신 정보 구성』을 참조하십시오.

5. 콘텐츠 서버에서 위의 정보를 지정하지 않은 경우, Caching Proxy가 현재 URL과 일치하는 캐시 기본 만기 설정을 찾아 만기 시간에 사용합니다. 캐시 기본 만기 값 설정에 대한 명령은 96 페이지의 『캐시 최신 정보 구성』을 참조하십시오.

위에서 설명한 방법으로 만기 시간을 계산하면 Caching Proxy가 이 URL에 적용되는 최소 보유 값의 존재 여부를 확인합니다. 최소 보유 값이 있고 지정한 시간이 계산된 만기 시간보다 길면, 최소 보유 값이 지정한 시간이 오브젝트의 만기 시간으로 사용됩니다. Caching Proxy가 문서에 대한 만기 시간을 0분으로 계산한 경우에도 마찬가지로 적용됩니다. 따라서 만기된 콘텐츠를 제공하지 않게 하려면, 최소 보유 값 설정 시 유의하십시오. (최소 보유 값을 설정하려면, CacheMinHold 지시문을 사용하거나 캐시 구성 -> 캐시 만기 설정: URL 만기 설정을 사용하십시오. 추가 정보는 96 페이지의 『캐시 최신 정보 구성』을 참조하십시오.)

최종 만기 시간이 시간 여유 설정에 지정된 시간과 비교하여 확인됩니다. 시간 여유 값보다 크면 문서가 캐시되고, 그렇지 않으면 캐시에 추가되지 않습니다. (시간 여유 값을 설정하려면 CacheTimeMargin 지시문을 사용하거나 96 페이지의 『캐시 최신 정보 구성』에 있는 명령을 참조하십시오.)

문서가 캐시에 있지만 만기된 경우, Caching Proxy가 *if-modified-since* 요청으로 알려진 특별 요청을 콘텐츠 서버에 발행합니다. 이 요청에 의해 콘텐츠 서버는, 프록시가 마지막으로 문서를 수신한 이후 문서가 변경된 경우에만 문서를 송신합니다. 문서가 변경되지 않은 경우에는 콘텐츠 서버가 이 콘텐츠를 알리는 메시지를 송신하고 페이지는 다시 송신하지 않습니다. 이 경우, 프록시가 캐시된 문서를 제공합니다. FTP 파일의 경우 프록시 서버가 *if-modified-since* 프로세스를 시뮬레이트합니다. 파일이 FTP 서버에서 변경되지 않은 것으로 판별하면, 캐시에서 파일을 제공합니다. 그렇지 않은 경우, FTP 서버에서 새 버전을 획득합니다.

캐시 최신 정보에 대한 추가 정보

- (동적으로 생성된 문서와 반대로) 거의 모든 정적 웹 문서에는 최종 변경 헤더가 있습니다. 이것은 프록시가 문서에 대한 만기 시간을 계산하는 가장 일반적인 방법이며, Caching Proxy가 FTP 파일에 대해 시도하는 첫 번째 방법입니다. 이것이 실패하면, 프록시는 기본 만기 값을 참조합니다.
- Cache-control: s-maxage, Cache-control: max-age 또는 Expires: header를 사용하는 문서는 거의 없습니다.
- 종종 캐시될 수 없는 동적으로 생성된 페이지에는 문서의 즉각적인 만기를 나타내는 Expires: 0 또는 Cache-control: no-cache 헤더가 있습니다. IBM WebSphere

Application Server에서 동적으로 생성된 파일의 캐시에 대한 정보는 111 페이지의 제 22 장 『동적 생성 콘텐츠 캐시』를 참조하십시오.

- HTTP 구문을 사용하여 URL에 대한 기본 만기 값을 0분 이상으로 설정할 때 주의하십시오. 동적으로 생성된 다수의 페이지에는 만기 헤더가 없으며 기본 만기 값을 따릅니다. 기본 만기를 0분 이상으로 설정하면 프록시가 오브젝트를 캐시할 수 있지만, 사용자는 시효가 지난 콘텐츠(또는 CGI 프로그램이나 servlet에서 예상치 못한 결과)를 획득하게 될 수 있습니다.
- 다음과 같은 경우에는, 캐시된 문서의 만기 여부에 관계없이 프록시 서버가 모든 요청에 대한 서버 문서의 유효성을 재검증합니다.
 - 문서가 다음 헤더 중 하나를 포함합니다.
 - Cache-control: s-maxage
 - Cache-control: must-revalidate
 - Cache-control: proxy-revalidate
 - 문서는 사용자 자격사항을 필요로 하지만 서버가 캐시할 수 있습니다.
 - 문서에는 Cache-Control: no-cache 헤더가 들어 있지만, (확장 캐시로 인해) 이와 관계없이 캐시됩니다.

FTP의 날짜 정보

이는 정방향 프록시 구성에만 적용합니다.

FTP 프로토콜은 HTTP 프로토콜과는 달리 날짜 및 시간을 엄격하게 정의하지 않기 때문에, 일부 요소는 FTP 파일에 대해 프록시에서 생성한 최종 변경 헤더가 실제 파일 날짜와 약간 다를 수 있습니다. 이 요소에는 다음 사항이 포함됩니다.

- HTTP 프로토콜과 달리, FTP 프로토콜을 리턴된 날짜가 GMT(그리니치 표준 시간)로 되어 있어야 한다고 지정하지 않습니다. FTP 서버가 돌려보낸 날짜는 FTP 서버의 로컬 시간일 수 있습니다. 프록시는 FTP 서버가 실행 중인 시간대를 판별할 수 없기 때문에, 이 시간을 자체의 시간대로 해석합니다. 이에 대한 예외는 Windows FTP 서버입니다. 이는 GMT의 날짜를 리턴합니다. FTP 서버가 Windows 시스템에서 실행 중임을 프록시가 발견하는 경우, 이는 디렉토리 날짜가 GMT로 되어 있다고 가정합니다.
- 일부 FTP 서버는 돌려보낸 디렉토리 정보의 날짜를 월 일 년 형식으로만 지정하며, 지정된 날짜에 대한 자세한 시간이나 분 정보는 포함하지 않습니다. FTP 서버가 파일에 대한 시간 및 분 정보를 돌려보내지 않으면, 프록시는 FTP 서버가 돌려보낸 날짜의 가장 최근의 시와 분으로 파일이 최종 변경되었다고 추정합니다. 예를 들어, FTP 서버가 파일이 1998년 10월 13일에 최종 변경되었음을 지시하는 파일에 대한 디렉토리 정보를 돌려보내면서 시간이나 분에 관한 정보는 포함하지 않는 경우, 프록시는 이 파일이 1998년 10월 13일 11:59:59 p.m.에 최종 변경되었다고 추정합니다.

그리고 FTP 서버가 Windows FTP 서버가 아니면 프록시는 이 날짜를 자체 고유 로컬 시간대에서 대응되는 GMT로 변환합니다.

FTP 파일이 캐시에서 만기될 때, 프록시는 FTP 파일에 대한 HTTP if-modified-since 재확인 프로세스를 시뮬레이트합니다. 이 작업은 요청한 파일에 대한 FTP LIST 명령을 다시 실행하고, FTP 서버가 돌려보낸 응답에서 파일 날짜를 분석하고, 이 날짜를 파일이 처음 검색될 때 최종 변경 헤더가 생성한 프록시 서버 날짜와 비교함으로써 수행합니다. 파일 날짜가 변경되지 않으면 프록시 서버는 재확인 시 캐시된 FTP 파일을 표시하고, 파일에 새로운 만기 시간을 설정하고, FTP 서버에서 파일을 다시 검색하지 않고 캐시에서 제공합니다. 두 파일의 날짜가 일치하지 않으면, 프록시는 FTP 서버에서 파일을 다시 검색하여 새로운 파일 날짜가 있는 새 사본을 캐시합니다.

FTP 서버에서 항상 파일에 대한 디렉토리 정보를 얻을 수 있는 것은 아닙니다. 프록시가 FTP 파일의 파일 날짜를 판별할 수 없으면, 그 파일에 대한 최종 변경 헤더를 생성하지 않습니다. 그 대신 캐시에 파일을 보존하는 기간을 판별하기 위해, URL과 일치하는 CacheDefaultExpiry 지시문에 지정된 값을 사용합니다. 이 기간이 만료되면 프록시는 FTP 서버에서 파일을 항상 재검색합니다. 캐시의 특정 FTP 파일이 CacheDefaultExpiry 지시문을 빈번하게 사용하고 자주 검색되면(대용량의 네트워크 통신량 생성), 이 특정 파일에 보다 세분된 CacheDefaultExpiry값을 지정할 것을 고려하십시오. 이렇게 함으로써 보다 오랜 기간 파일을 캐시에 보존할 수 있습니다.

구성 및 관리 양식에서 캐시 만기 설정을 지정하려면, 캐시 구성 -> 캐시 만기 설정 -> 캐시된 파일의 시간 한계 양식을 사용하십시오. 캐시된 파일 만기 날짜에 대한 자세한 내용은 93 페이지의 『파일 만기』를 참조하십시오.

캐시 최신 정보 구성

캐시된 파일의 만기 시간을 지정하려면, 구성 및 관리 양식에서 캐시 구성 -> 캐시 만기 설정을 선택하십시오. 다음 양식이 유용합니다.

URL 기반 만기

이 양식을 사용하여 해당 URL을 기반으로 캐시에서 파일을 보관하는 최소 시간을 설정하십시오. 다른 URL 요청 템플릿에 다른 캐시 작동을 지정할 수 있습니다.

프록시 구성 파일을 편집하여 URL 기반의 파일 만기를 설정하려면, 185 페이지의 부록 B 『구성 파일 지시문』의 참조 섹션에서 다음 지시문을 참조하십시오.

- 209 페이지의 『CacheMinHold — 파일을 사용 가능하게 보존하는 기간 지정』

기본 만기 설정

캐시 만기 설정 양식으로 사용된 파일 또는 사용되지 않은 파일에 대한 기본 만기 값을 지정할 수 있습니다. HTTP, FTP 및 Gopher 파일에 다른 값을 설정할 수 있으며 파일 사용 여부에 따라 다른 값을 설정할 수 있습니다.

이 양식에는 추가적인 파일 만기 옵션도 있습니다.

- 캐시된 파일 확인 사용 가능 이 선택란은 기본적으로 선택됩니다. 일반적으로 이 옵션을 선택하여 서버가 만기된 콘텐츠를 서브하지 않도록 하는 것이 바람직합니다.
- 원격 서버에서 파일 검색 사용 불가능 원격 서버에서 파일을 서버가 검색하지 못하도록 하려면 이 옵션을 선택하십시오.
- 만기될 파일을 캐시하지 않음 단기간에 만기될 파일의 캐시를 방지하려면 이 옵션을 사용하여 기간을 지정하십시오. 기본적으로 10분 이내에 만기하는 파일은 캐시하지 않습니다.

프록시 구성 파일을 편집하여 기본 만기 설정을 설정하려면, 참조 페이지에서 다음 지시문을 참조하십시오.

- 203 페이지의 『CacheDefaultExpiry — 파일에 대한 기본 만기 시간 지정』
- 204 페이지의 『CacheExpiryCheck — 서버가 만기된 파일을 리턴할지 여부 지정』
- 211 페이지의 『CacheTimeMargin — 파일 캐시를 위한 최소 수명 지정』
- 212 페이지의 『CacheUnused — 사용되지 않는 캐시 파일 보존 기간 지정』
- 209 페이지의 『CacheNoConnect — 독립형 캐시 모드 지정』

최종 변경 요소 설정

최종 변경 요소 양식을 사용하여 헤더에 만기 날짜가 없는 캐시된 파일의 만기 날짜 계산을 위해 프록시가 사용하는 값을 설정하십시오. 서로 다른 요청 템플릿과 일치하는 파일에 다른 값을 설정할 수 있습니다. 첫 번째로 일치하는 템플릿은 만기 날짜를 계산하는 데 사용됩니다.

프록시 구성 파일을 직접 편집하여 최종 변경 요소를 설정하려면, 205 페이지의 『CacheLastModifiedFactor — 만기 날짜 판별을 위한 값 지정』을 참조하십시오.

캐시 시간 한계

캐시된 파일의 시간 한계 구성 양식을 사용하여 파일이 캐시에 머무는 최대 시간을 설정하십시오. 시간 한계는 요청 템플릿을 기초로 하여 설정되며, 시간 한계가 만기될 때 버리거나 유효화해야 할 파일을 지정할 수 있습니다. 이러한 설정은 만기 날짜가 유효하지 않은 파일이나 만기 날짜로 설정된 시간이 지나치게 긴 파일을 유지하는 데 사용될 수 있습니다.

프록시 구성 파일을 편집하여 캐시된 파일에 대한 최대 만기 시간 한계를 설정하려면, 다음 사항을 참조하십시오.

- 207 페이지의 『CacheMaxExpiry — 캐시 파일의 최대 수명 지정』
- 203 페이지의 『CacheClean — 캐시 파일 보존 기간 지정』

가비지 콜렉션

즐겨찾는 URL을 캐시된 상태로 보존하고 시스템 자원의 사용을 최소화하기 위한 노력의 일환으로, Caching Proxy는 가비지 콜렉션이라는 정리 프로세스를 수행합니다.

가비지 콜렉션 프로세스는 캐시 크기를 줄이고 새 파일을 위한 공간을 생성하기 위해 캐시 디렉토리에 있는 파일을 점검하고 만기된 파일을 제거하는 과정입니다. 가비지 콜렉션은 자동으로 실행되지만, 일부 설정을 구성하여 사용자 필요에 맞출 수 있습니다.

가비지 콜렉션 구성

가비지 콜렉션을 구성하려면, 구성 및 관리 양식에서 캐시 구성 -> 가비지 콜렉션 설정을 선택하십시오. 이 양식을 사용하여 최고 수준 및 최저 수준을 설정하여 가비지 콜렉션의 시작 및 정지 시기를 판별하십시오. 캐시에 사용된 총 영역이 최고 수준의 백분율 세트에 근접 또는 초과하면 가비지 콜렉션이 시작됩니다. 가비지 콜렉션은 캐시의 사용 영역 백분율이 최저 수준의 값 세트 이하가 될 때까지 계속 실행됩니다.

두 가지 가비지 콜렉션 알고리즘에서 선택할 수 있습니다. 응답 시간 알고리즘은 캐시에서 대용량 파일을 우선적으로 제거함으로써 사용자 응답에 필요한 시간을 최적화합니다. 대역폭 알고리즘은 캐시에서 소용량 파일을 우선적으로 제거함으로써 네트워크 대역폭 사용을 최적화합니다. 두 가지 중 하나를 선택하거나 두 가지를 혼합하십시오.

프록시 구성 파일을 편집하여 가비지 콜렉션을 구성하려면, 참조 섹션에서 다음 지시문을 참조하십시오.

- 236 페이지의 『Gc — 가비지 콜렉션 지정』
- 237 페이지의 『GcHighWater — 가비지 콜렉션 시작 시간 지정』
- 237 페이지의 『GcLowWater — 가비지 콜렉션 종료 시기 지정』
- 202 페이지의 『CacheAlgorithm — 캐시 알고리즘 지정』

제 20 장 자동 새로 고침 및 사전 로드에 대한 캐시 에이전트 구성

대부분의 Caching Proxy 서버는 사용자가 요청한 후에만 파일을 캐시합니다. Caching Proxy에는 자동 캐시 사전 로드를 제공하는 캐시 에이전트가 있습니다. 캐시 에이전트가 지정 URL 또는 가장 즐겨찾는 URL을 자동 검색하여 요청 이전에 이를 캐시에 위치시키도록 지정할 수 있습니다.

일부 경우, 캐시를 사전 로드하기 전에 프록시 서버의 호스트 이름을 설정하고 캐시 액세스 로그를 식별해야 합니다. 캐시 에이전트를 구성하려면, 구성 및 관리 양식에서 캐시 구성을 선택한 후, 캐시 사전 로드 및 캐시 새로 고침 양식을 사용하십시오. 조회 결과를 표시하는 파일(즉, 물음표(?))가 들어있는 URL의 파일은 조회 캐시를 사용할 수 있는 경우에만 캐시됨을 주의하십시오.

자동 캐시 새로 고침 및 사전 로드의 장점은 다음과 같습니다.

- 사용자가 페이지를 요청하기 전에 지정된 URL에 캐시가 적용됩니다.
- 서버의 활동량이 많아지기 전에 캐시가 자리를 차지합니다.
- 현재 파일을 첫 번째 요청에서 불러오는 것 보다 캐시에서 더 신속하게 공급됩니다.

용통성있는 클라이언트 SOCKS의 단점은 다음과 같습니다.

- 사용자 활동이 적은 시간에도 프록시 서버가 페이지를 캐시합니다.
- 자동으로 로드된 문서는 어느 만큼 제어를 실행해야 합니다. 웹 색인 및 탐색 사이트와 같은 상위 레벨 페이지에서 링크 파일을 로드하는 것은 방대한 페이지 요청을 생성할 수 있습니다.

효율을 최적화하기 위해서는, 서버 활동이 적고 클라이언트 요청으로 서버 사용량이 많아지기 전에 캐시 에이전트가 실행하도록 설정하십시오. 그러면 사용자가 처음 요청할 때 신속한 서비스를 제공하도록 파일이 캐시에 준비될 것입니다. 기본적으로 캐시 에이전트는 현지 시간으로 매일 3 a.m.에 시작됩니다.

역방향 프록시 구성에 대한 특수 고려사항:

보안상 이유로 인해, 역방향 프록시 구성을 사용하는 경우, Proxy http:* 규칙을 기본적으로 사용할 수 없습니다. (이 규칙은 ibmproxy.conf 파일에 주석이 있습니다.) 그러나, 규칙을 사용할 수 없는 경우, 캐시 에이전트가 요청을 전송하고, Caching Proxy의 캐시 콘텐츠를 새로 고칠 수 없습니다. 오류 로그에 "403 규칙에 의해 금지 오류"가 발생하며 캐시 새로 고침이 완료되지 않습니다.

이 문제점을 피하려면, cacheAgentService를 사용하는 데, 이는 Caching Proxy에서 제공하는 내부 서비스입니다. 이 서비스를 사용하려면, ibmproxy.conf 파일의 다른 맵핑 규칙 앞에 Service 지시문을 넣으십시오.

Service /any-valid-string* INTERNAL:cacheAgentService

any-valid-string 변수는 ibmproxy.conf 파일에서 다른 맵핑 규칙과 충돌하지 않는 유효한 문자열입니다.

Caching Proxy 및 캐시 에이전트는 이 Service 지시문에 기반하여 URI를 구문 분석합니다. URI를 직접 Caching Proxy에 전송하는 대신, 캐시 에이전트 유틸리티는 Service 지시문에 /any-valid-string 패턴을 URI에 추가합니다.

예를 들어, 캐시 에이전트는 다음 URI

http://www.ibm.com/

을 다음과 같이 변환합니다.

/any-valid-string/http://www.ibm.com/

캐시 에이전트는 접두부가 있는 URI를 Caching Proxy에 전송합니다. Caching Proxy가 요청을 수신하는 경우, /any-valid-string/ 접두부를 제거합니다. 남아 있는 URI가 완전한 단위인 경우, Caching Proxy는 다른 규칙에 대해 URI를 맵핑하지 않고, 요청을 직접 제공합니다.

추가로, 캐시 에이전트는 상대 URI를 Caching Proxy에 전송할 수 있습니다. 예를 들어, ibmproxy.conf에 있는 이전에 참조된 Service 지시문을 사용하여 LoadURL /abc/를 추가하는 경우, 캐시 에이전트는 /any-valid-string/abc/로 변환하여 Caching Proxy에 전송합니다. Caching Proxy는 URL을 수신하여 접두부를 제거하고 다른 맵핑 규칙에 대해 /abc/를 맵핑하며, 일치하는 경우, 요청을 핸들합니다.

Service 지시문에 대한 정보는 300 페이지의 『Service — 서비스 단계 사용자 정의』를 참조하십시오.

서버 호스트 이름 설정

Linux 및 UNIX 플랫폼에서는 캐시를 사전 로드하거나 새로 고칠 프록시 서버의 호스트 이름을 지정합니다. Windows 플랫폼에서는 새로 고칠 프록시 서버가 로컬 시스템에 없는 경우에만 호스트 이름을 지정합니다(로컬 캐시 에이전트가 원격 서버의 캐시 액세스 로그에 액세스할 수 없기 때문에, 자주 액세스한 파일을 기초로 하여 원격 서버의 캐시를 새로 고칠 수 없습니다).

프록시 서버의 호스트 이름을 설정하려면, 구성 및 관리 양식에서 캐시 구성 -> 캐시 새로 고침: 캐시 대상 서버 식별을 선택하십시오.

캐시를 고유한 파일로 사전 로드

특정 URL에 저장된 콘텐츠로 캐시를 사전 로드하려면, 구성 및 관리 양식에서 캐시 구성 -> 캐시 사전 로드를 사용하십시오. 이 양식에서, 로드할 캐시 에이전트에 대한 URL을 지정할 수 있습니다. 캐시 에이전트가 시작되면, 페이지가 이전에 캐시에 있었는지 여부에 관계없이 프록시가 해당 페이지를 검색합니다(이 URL은 LoadURL 지시문에 의해 프록시 구성 파일에 지정됩니다). 이 양식은 또한 콘텐츠가 캐시되지 않는 URL을 정의하는 데 사용할 수 있습니다. 이 유형의 캐시 사전 로드에는 캐시 액세스 로그에 대한 액세스가 요구되지 않습니다.

캐시 사전 로드 양식을 사용하여 다음 옵션을 구성할 수 있습니다.

- 매일 캐시 새로 고침—캐시 에이전트가 매일 밤 캐시를 새로 고치려면 이 상자를 체크하십시오. 캐시 에이전트를 시작하기를 원하지 않으면, 이 상자를 체크하지 마십시오.
- 캐시 새로 고침 시간—현지 시간으로 3:00 a.m. 이외의 시간에 캐시 에이전트를 실행하려면, 시작할 시간을 지정하십시오.
- 캐시 콘텐츠—URL 또는 IP 주소 필드에서, 로드할 URL을 지정하십시오. URL을 사전 로드하지 않으려면 URL을 지정한 후, 캐시 상태 상자에서 무시를 누르십시오.

캐시를 자주 캐시된 파일로 사전 로드

가장 많이 액세스되는 페이지를 자동으로 사전 로드하려면 캐시 구성 -> 캐시 새로 고침 양식을 사용하십시오. 이 기능에는 프록시 서버에 대한 캐시 액세스 로그가 필요합니다. (로그 위치 및 이름은 변경될 수 있습니다. 자세한 정보는 157 페이지의 제 6 부 『Caching Proxy 모니터링』을 참조하십시오.) 가장 즐겨찾는 URL은 캐시 액세스 로그에서 자동으로 판별합니다. 또한 관리자는 캐시에서 사전 로드할 즐겨찾는 페이지의 수를 지정할 수 있습니다(이 수는 프록시 구성파일에서 LoadTopCached 지시문에 의해 지정됩니다).

캐시 새로 고침 양식을 사용하여 다음 옵션을 구성할 수 있습니다.

- 매일 캐시 새로 고침—캐시 에이전트가 매일 밤 캐시를 새로 고치려면 이 상자를 체크하십시오. 캐시 에이전트를 시작하기를 원하지 않으면, 이 상자를 지우십시오.
- 캐시 새로 고침 시간—3:00 a.m. 이외의 시간에 캐시 에이전트를 실행하려면, 시작할 시간을 지정하십시오.
- 캐시 대상 서버 식별—로컬 시스템 이외의 서버를 새로 고치려면 이 옵션을 사용하십시오. (특정 파일 액세스 빈도에 따라 원격 서버를 새로 고칠 수 없습니다).
- 가장 즐겨찾는 URL 캐시—전날 밤 캐시 액세스 로그에서 캐시한 URL 수를 지정하십시오.
- 링크 페이지 로드—이 설정을 사용하여 링크 캐시(delving)를 구성하십시오(링크 캐시(delving)에 대한 세부사항은 다음 섹션 참조). 링크 캐시할 레벨의 수를 설정하십시오.

고, 모든 페이지에 링크 캐시할지(로드함), 전혀 링크 캐시하지 않을지(로드 안함), 관리자가 지정한 페이지만 링크 캐시할 지(관리자), 즐겨찾는 페이지만 링크 캐시할지(topn)를 설정하십시오. 또한 호스트를 통해 탐색할지, 요청 사이에 연기할지, 인라인 이미지를 캐시할지 여부를 지정하십시오.

- 스프레드 수—캐시 새로 고침에 사용할 스프레드의 최대수를 설정하십시오.
- 최대 작업 대기열 깊이—URL이 요청할 최대 대기열을 설정하십시오.
- 요청할 최대 URL—S로드할 페이지의 최대수를 설정하십시오. 이 수는 링크 캐시(delving) 페이지 검색을 시작하기 전에 확인됩니다.
- 최대 시간—캐시 에이전트를 실행할 최대 시간을 설정하십시오. 시간이 0시간 0분으로 설정되면, 캐시 에이전트가 실행을 완료합니다.

링크 캐시(delving)

링크 캐시(delving)는 자동 캐시 새로 고침 기능의 선택적 부분입니다. 대부분의 웹 페이지는 관련 정보가 있는 다른 페이지로 연결되며, 사용자는 종종 한 페이지에서 다른 페이지로, 한 사이트에서 다른 사이트로 경로를 따라 연결됩니다. 링크 캐시(delving)는 이러한 논리 정보 경로를 캐시하는 방법입니다. 링크 캐시(delving)에서 캐시 에이전트는 로드하고 있는 페이지에서 지정된 레벨의 하이퍼텍스트(HTML) 연결을 따르고, 연결된 모든 페이지를 캐시합니다. 연결된 페이지는 원본 페이지와 동일한 호스트에 상주하거나, 다른 호스트에 상주할 수도 있습니다. 103 페이지의 그림 1에서 볼 수 있습니다.

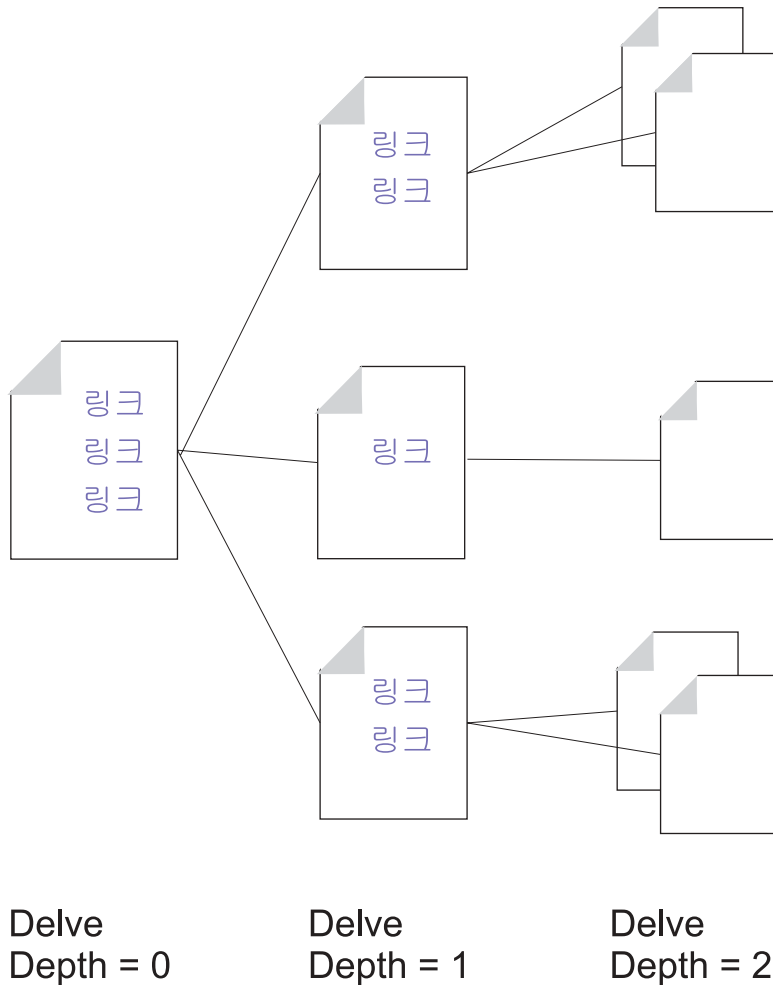


그림 1. 링크 캐시(delving)

링크 캐시(delving) 프로세스를 제어하기 위해서, 관리자는 로드할 수 있는 최대수의 URL(기본 설정 2000), 실행할 수 있는 최대 시간(기본 설정 2시간), 사용할 수 있는 최대 스프레드(기본 설정 4)를 설정합니다. 관리자는 추가적인 제어를 구성할 수 있습니다. 기본적으로 링크 캐시(delving)는 두 레벨의 계층끼리 사용 가능하고 호스트간에는 사용할 수 없습니다. 추가적으로 링크 캐시(delving) 요청 사이에는 시간이 걸립니다. 이러한 설정을 변경하려면, 104 페이지의 『관련 프록시 구성 파일 지시문』을 참조하십시오.

캐시 에이전트는 로드 후 다음 순서로 캐시를 새로 고칩니다.

1. 관리자가 지정한 특정 페이지를 로드합니다.
2. 캐시 액세스 로그에서 즐겨찾는(자주 액세스하는) 페이지를 로드합니다.
3. 페이지의 최대수가 이 지점에 도달하지 않은 경우에는 링크 캐시(delving)가 추가 페이지를 로드합니다.

캐시 에이전트는 연결간의 링크 캐시(delving)를 시작할 때까지 페이지의 최대수에 도달했는지 여부를 확인하지 않습니다. 페이지의 최대수(프록시 구성 파일에서는 MaxURL로 불림)가 1단계 및 2단계에서 검색한 페이지 수보다 적은 경우에는, 연결된 페이지가 검색되지 않습니다.

다음 예는, 캐시 에이전트가 지정된 URL의 최대수와 관계 있는 캐시 새로 고침 우선 순위 및 탐색 정보를 처리하는 방법을 보여줍니다(모든 예에 대해 링크 캐시(delving)가 구성된 것으로 가정).

구성 파일 설정	결과
LoadURL http://www.getthis.com/main.html LoadURL http://www.getmetoo.com/welcome.htm LoadTopCached 30 MaxURLs 50	캐시 액세스 로그의 고유 URL이 30개 이상인 경우에는 캐시 액세스 로그를 기초로 캐시 에이전트가 main.html, welcome.htm 및 상위 30개의 요청된 URL을 검색합니다. MaxURL값에 도달하지 않았기 때문에, 이미 캐시된 페이지에서 최대 18개의 연결된 URL을 검색 및 로드합니다.
LoadURL http://www.joesmith.edu/favorites.html LoadURL http://www.janesmith.edu/dislikes.html LoadTopCached 30 MaxURLs 25	캐시 액세스 로그의 고유 URL이 30개 이상인 경우에는 캐시 에이전트가 favorites.html, dislikes.html 및 캐시 액세스 로그의 상위 30개의 요청된 URL을 검색합니다. MaxURL의 값을 초과했기 때문에 다른 파일을 검색하지 않습니다.
LoadURL http://www.hello.com/hi.htm LoadURL http://www.ballyhoo.com/index.html LoadTopCached 20 MaxURLs 25	캐시 액세스 로그에 20개 이상의 고유한 URL이 있으면, 캐시 에이전트는 hi.htm, index.html, 캐시 액세스 로그에서 상위 20개의 요청된 URL 및 이전 페이지에서 최대 3개의 연결된 URL을 검색합니다. MaxURL의 값에 도달했기 때문에 다른 파일을 검색하지 않습니다.

관련 프록시 구성 파일 지시문

또한 프록시 구성 파일의 해당 지시문을 직접 편집하여 캐시 에이전트를 구성할 수 있습니다. 캐시 에이전트 관련 프록시 구성 파일 지시문은 185 페이지의 부록 B 『구성 파일 지시문』에서 다음 참조 페이지를 참조하십시오.

- 200 페이지의 『AutoCacheRefresh — 캐시 새로 고침을 사용할지 여부 지정』
- 201 페이지의 『CacheAccessLog — 캐시 액세스 로그 파일에 대한 경로 지정』
- 211 페이지의 『CacheRefreshTime — 캐시 에이전트를 시작할 시기 지정』
- 220 페이지의 『DelayPeriod — 요청 간 일시정지 지정』
- 220 페이지의 『DelveAcrossHosts — 도메인을 통한 캐시 지정』
- 220 페이지의 『DelveDepth — 캐시하는 동안 연결을 따르는 범위 지정』
- 220 페이지의 『DelveInto — 캐시 에이전트가 연결을 따르는지를 지정』
- 242 페이지의 『IgnoreURL — 새로 고치지 않을 URL 지정』
- 249 페이지의 『LoadInlineImages — 삽입된 이미지 새로 고침 제어』

- 250 페이지의 『LoadTopCached — 새로 고침 즐겨찾기 페이지 수 지정』
- 250 페이지의 『LoadURL — 새로 고침 URL 지정』
- 259 페이지의 『MaxUrls — 새로 고침 URL의 최대수 지정』

수동으로 캐시 에이전트 시작

자동 캐시 새로 고침이 사용 가능하면, 캐시 에이전트는 지정된 시간에 자동으로 새로 고침을 실행합니다. 그러나 언제라도 명령행에서 캐시 에이전트를 실행할 수도 있습니다.

실행 가능한 파일은 다음과 같습니다.

- Linux 및 UNIX 플랫폼의 경우: `usr/sbin/cacheagt`
- Windows 플랫폼의 경우: `server_root\bin\cacheagt.exe`

여기서 `server_root`는 Caching Proxy를 설치한 드라이브 및 디렉토리(예: `C:\Program Files\IBM\edge\cp`)입니다.

Linux 및 UNIX 플랫폼에서는 **cron** 디먼을 사용하여 다양한 시간에 캐시 에이전트를 자동으로 실행할 수 있습니다. **cron**이 제어하는 작업은 시스템 `crontab` 파일에 행을 추가하여 지정됩니다. Linux 및 UNIX의 명령 파일 입력 항목의 예는 다음과 같습니다.

```
45 16 * * * /usr/sbin/cacheagt
```

이 명령은 현지 시간으로 매일 4:45 p.m.에 캐시 에이전트를 시작합니다. 원할 경우, 여러 개의 항목을 사용하여 한 번 이상 캐시 에이전트를 실행할 수 있습니다. 자세한 정보는 운영 체제의 **cron** 디먼에 관한 문서를 참조하십시오.

캐시 에이전트를 실행하기 위해서 **cron** 디먼을 사용하고 있는 경우, 캐시 구성 -> 캐시 새로 고침 구성 양식을 사용하거나 프록시 구성 파일을 편집하여 자동 새로 고침 옵션을 사용하지 마십시오. 그렇지 않으면, 캐시 에이전트가 매일 한 번 이상 실행됩니다.

제 21 장 공유 캐시 사용

웹상의 한 지점에서 한 대의 서버가 처리할 수 있는 것보다 통신량이 많아지는 경우는 흔합니다. 한 가지 간단한 해결책은 여러 대의 서버를 추가하는 것입니다. 그러나 여러 Caching Proxy 서버가 사용되면, 한 캐시의 콘텐츠가 다른 캐시의 콘텐츠와 중복되는 경우가 있습니다. 자체 캐시에 해당 파일이 없는 프록시 서버로 파일에 대한 요청이 도달하면 캐시된 파일이 기점 서버로부터 다시 페치되기 때문에 저장영역에서의 불필요한 중복은 물론 최대 대역폭 절약도 달성할 수 없습니다. 중복된 캐시는 프록시 서버의 계층적 연결을 사용하여 최소화할 수 있지만, 이 시나리오에서는 여전히 해당 서버를 통해 추가적인 통신량이 발생하는데, 연결된 서버에서 추가 연결로 인해 대기 시간이 늘어납니다.

캐시 공유는 각 캐시가 다른 캐시와 콘텐츠를 공유하게 함으로써 이러한 문제점을 해결합니다. 다음 사항으로 인해 대역폭을 줄일 수 있습니다.

- 오브젝트를 여러 번 페치하지 않습니다.
- 논리 캐시가 더 크게 결합할수록 히트 비율이 높아집니다.

여러 개의 캐시를 하나의 논리 캐시처럼 사용할 수 있도록 하는 두 가지 메소드가 제공됩니다.

- RCA(Remote Cache Access)는 구성원 캐시의 배열을 정의하는 Caching Proxy의 기능입니다. 파일은 내부 논리를 기본으로 이러한 캐시 중 하나에 저장됩니다.
- 프록시 서버가 ICP(Internet Caching Protocol)를 사용할 수 있도록 Caching Proxy 플러그인이 제공됩니다. Caching Proxy 시스템과 비Caching Proxy 캐시 간에 데이터를 공유하고자 할 경우 RCA 대신 ICP 플러그인을 사용할 수 있습니다.

RCA 및 ICP는 함께 사용될 수 있습니다.

원격 캐시 액세스

RCA에 대한 계획을 세울 때, 다음의 권장사항을 고려하십시오.

- 참여 프록시 서버가 서로 인접해야 하며 높은 대역폭 링크로 연결되어야 합니다 (예: FDDI, SP2 버스).
- RCA 배열의 멤버십은 구성이 가능한 안정적인 수 있도록 장기 멤버십이어야 합니다.
- 프록시 서버는 비슷한 성능(예: CPU, 메모리 크기, 캐시 크기)을 가지고 있어야 합니다.
- 네트워크 동력 정지가 자주 발생하지 않아야 합니다.
- 배열의 구성원 수가 100 이하여야 합니다.

- 배열의 모든 구성원은 같은 버전의 Caching Proxy 소프트웨어를 사용하고 있어야 합니다.

주: RCA 배열의 프록시가 다른 Linux 운영 체제(예: SUSE 및 Red Hat)를 사용하고 있으면, "nobody" 사용자가 모든 피어에서 동일한 UID를 갖도록 하십시오. 각 시스템의 /etc/ 디렉토리에서 암호 및 그룹 파일 입력 항목을 확인하고 "nobody"에 같은 UID를 지정하십시오.

이들 조건이 위반된 경우 또는 다른 조직이 배열의 구성원인 다른 서버를 관리하는 경우, 원격 캐시 액세스가 적당하지 않습니다.

원격 캐시 액세스 구성

원격 캐시 액세스를 구성하려면, 구성 및 관리 양식에서 캐시 구성 -> 원격 캐시 액세스를 선택하십시오. 이 양식의 필드는 하나의 논리 캐시를 공유하는 명명된 배열을 정의합니다. 배열의 각 구성원에 대한 필수 정보를 입력하십시오.

프록시 구성 파일을 편집하여 원격 캐시 액세스를 구성하려면, 185 페이지의 부록 B 『구성 파일 지시문』의 참조 섹션에서 다음 지시문을 참조하십시오.

- 199 페이지의 『ArrayName — 원격 캐시 배열 이름 지정』
- 259 페이지의 『Member — 배열의 구성원 지정』

인터넷 캐시 프로토콜 플러그인 구성

인터넷 캐시 프로토콜 플러그인으로 Caching Proxy가 HTML 페이지 및 기타 캐시 가능 자원 검색 시 ICP 준수 캐시를 조회할 수 있습니다. 프록시 서버가 HTTP 요청을 수신하면, 자원에 대한 자체 캐시를 탐색합니다. 자원이 로컬 캐시에 없고 ICP 플러그인이 사용 가능한 경우, 프록시 서버가 URL 요청을 ICP 조회 패킷에 요약한 다음, 식별된 모든 ICP 피어 캐시에 이 패킷을 전달합니다. 피어 캐시가 자원이 있다고 응답하면 프록시 서버는 피어의 캐시에서 자원을 검색합니다. 두 개 이상의 캐시가 긍정적으로 응답하면 첫 번째 응답이 처리됩니다. 히트로 응답하는 피어가 없는 경우, 원래 서버가 작업 흐름에 따라 요청을 계속 처리합니다. 예를 들어, 프록시 서버는 다른 플러그인을 호출하고 RCA(Remote Caching Access) 루틴을 계속하거나(RCA가 사용 가능한 경우) 요청된 자원 자체를 검색할 수 있습니다.

ICP 플러그인 구성

ICP 플러그인은 프록시 구성 파일(ibmproxy.conf)을 편집하여 활성화되고 구성됩니다. ServerInit 지시문, PreExit 지시문 또는 둘 모두가 구성 파일의 API 지시문 섹션에 추가되어야 ICP 플러그인을 초기화할 수 있습니다. 사용되는 지시문은 Caching Proxy가 ICP 시스템에서 맡고 있는 역할에 따라 다릅니다.

- Caching Proxy가 ICP 서버로 작동할 경우, ServerInit 지시문을 사용하여 icpServer 모듈을 호출하십시오.

- Caching Proxy가 ICP 클라이언트로 작동할 경우, PreExit 지시문을 사용하여 icpClient 모듈을 호출하십시오.
- Caching Proxy가 ICP 클라이언트 및 ICP 서버로 작동할 경우, 두 지시문을 사용하십시오.
- 플러그인에서 사용하는 설정을 구성하려면 icpAddress, icpMaxThreads, icpPeer, icpPort 및 icpTimeout 지시문을 사용하십시오.

이러한 지시문을 작성하려면, ibmproxy.conf 파일을 직접 편집하거나 프록시 서버가 이미 실행 중인 경우에는 구성 및 관리 양식 서버 구성 -> 요청 처리 -> **API** 요청 처리에 연결하십시오.

표준 지시문(설명 양식)이 ibmproxy.conf 파일의 API 섹션에 추가되었습니다. API 지시문은 중요도 순으로 나열되어 있습니다. API 지시문을 추가하여 새로운 기능 및 플러그인 모듈을 사용할 수 있을 때, 구성 파일의 표준 섹션에 표시된 대로 지시문을 나열하십시오. 또는 필요한 경우, 원하는 각 기능이나 플러그인에 대한 지원을 포함할 API 지시문의 설명을 지우거나 편집하십시오.

ServerInit 및 PreExit 지시문에는 (1) 공유 라이브러리의 전체 경로, (2) 함수 호출이라는 두 가지 인수가 있습니다. 두 인수는 콜론(:)으로 구분됩니다. 첫 번째 인수는 시스템에 고유하며, 플러그인 컴포넌트가 설치되는 위치에 따라 다릅니다. 두 번째 인수는 공유 라이브러리에 하드 코드되며, 표시된 대로 정확히 입력해야 합니다.

각 지시문은 프록시 구성 파일의 단일 행에 표시되어야 합니다.

ServerInit *path_of_shared_library*:icpServer

Linux 및 UNIX 예제:

ServerInit /opt/ibm/edge/cp/internet/lib/plugins/icp/libicp_plugin.so:icpServer

Windows 예제:

ServerInit C:\Program Files\IBM\edge\cp\Bin\plugins\icp\icpplugin.dll:icpServer

PreExit *path_of_shared_library*:icpClient

Linux 및 UNIX 예제:

PreExit /opt/ibm/edge/cp/internet/lib/plugins/icp/libicp_plugin.so:icpClient

Windows 예제:

PreExit C:\Program Files\IBM\edge\cp\Bin\plugins\icp\icpplugin.dll:icpClient

플러그인 설정을 구성하려면, 프록시 구성 파일에 제공되는 ICP* 지시문을 추가하거나 수정하십시오. 추가 정보는 다음 지시문의 설명을 참조하십시오.

- 240 페이지의 『ICP_Address — ICP 조회의 IP 주소 지정』
- 240 페이지의 『ICP_MaxThreads — ICP 조회를 위한 최대 스레드 지정』

- 241 페이지의 『Occupier — ICP 클러스터의 구성원 지정』
- 241 페이지의 『ICP_Port — ICP 조회의 포트 번호 지정』
- 242 페이지의 『ICP_Timeout — ICP 조회의 최대 대기 시간 지정』
- 274 페이지의 『PreExit — PreExit 단계 사용자 정의』
- 299 페이지의 『ServerInit — 서버 초기설정 단계 사용자 정의』

제 22 장 동적 생성 콘텐츠 캐시

이는 역방향 프록시 구성에만 적용합니다.

동적 캐시 기능을 사용하면 Caching Proxy가 IBM WebSphere Application Server에 의해 생성된 Servlet 및 JSP(JavaServer Pages)에서의 응답 양식으로 생성된 콘텐츠를 동적으로 캐시할 수 있습니다. Application Server에서 Caching Proxy 어댑터 모듈을 사용하여 응답을 수정함으로써 이를 프록시 서버 및 Application Server의 동적 캐시에서 캐시할 수 있습니다. 이 기능을 사용하면, 동적으로 생성된 콘텐츠를 네트워크 에지에서 캐시할 수 있기 때문에, 하나 이상의 클라이언트가 동일한 콘텐츠를 요청할 때 콘텐츠 호스트가 Application Server에 반복 요청을 할 필요가 없습니다.

주: 동적 캐시 기능은 URL 조회 결과를 캐시하기 위해 프록시 서버를 사용 가능하게 하지 않습니다. 조회 결과를 캐시하려면, 89 페이지의 제 18 장 『캐시되는 내용 제어』 및 210 페이지의 『CacheQueries — 물음표(?)를 포함하는 URL에 캐시 응답 지정』의 지시문 참조 문서에서 설명하는 캐시 필터를 구성하십시오. IBM WebSphere Application Server가 아닌 기점 서버로부터의 조회 결과는 캐시할 수 있습니다.

예를 들어, servlets가 조회 양식에서 URL을 사용할 경우, 동적 캐시 기능이 작동하게 하려면 종종 조회 캐시를 사용 가능하게 해야 합니다. 프록시 서버는 조회할 URL에 물음표(?)가 포함되어 있는지를 고려합니다.

동적으로 생성된 콘텐츠를 캐시하면 다음과 같은 이점이 있습니다.

- 콘텐츠 호스트에 대한 로드가 줄어듭니다.
- Application Server에 대한 로드가 줄어듭니다.
- 요청 자원이 일반 사용자에게 전달되는 속도가 빨라집니다.
- 서버간의 대역폭 사용이 감소합니다.
- 동적으로 생성된 콘텐츠를 작성 또는 제공하는 웹 사이트의 가변성을 향상시킵니다.

Application Server는 프록시 캐시의 완전히 작성된 공용 페이지만을 내보냅니다. 개인용 페이지는 프록시에 의해 캐시되지 않습니다. 예를 들어, 현재의 일기 예보를 나열하는 공용 사이트에서 동적으로 생성된 페이지는 IBM WebSphere Application Server가 내보내고 Caching Proxy가 캐시할 수 있습니다. 그러나 사용자의 장비구내 내용을 나열하는 동적 생성 페이지는 프록시 서버가 캐시할 수 없습니다. 또한 동적으로 작성된 페이지를 캐시하려면 해당 페이지의 모든 하위 컴포넌트 또한 캐시할 수 있어야 합니다.

캐시된 동적 파일은 일반 파일과 동일한 방식으로 만료되지 않고 이 파일을 생성한 Application Server가 무효화해야 합니다.

동적 캐시 항목이 무효화되는 경우는 다음과 같습니다.

- 동적 캐시 가비지 콜렉터가 캐시 정체로 인해 항목을 제거합니다.
- Servlet 항목(servletcache.xml) 또는 프록시의 ExternalCacheManager 지시문의 시간 종료 설정이 만기되었습니다.
- 외부 에이전트 또는 응용프로그램이 동적 캐시 API를 호출하여 캐시 항목을 무효화합니다.

동적 캐시 항목 무효화는 Caching Proxy 동적 캐시 플러그인의 특정 인스턴스에 대한 무효화 메시지를 생성하여 수행할 수 있습니다. Caching Proxy는 무효화 메시지를 /WES_External_Adapter 자원 위치 지정자의 포스트로서 수신합니다. 그러면 Caching Proxy는 캐시에서 유효한 항목을 지웁니다.

동적 캐시는 다음 구성 단계가 필요합니다.

- IBM WebSphere Application Server 구성
 - 각 Application Server가 로컬 동적 캐시를 수행하도록 구성하십시오.
 - 각 Application Server가 외부 캐시 어댑터를 사용하도록 구성하십시오.
 - 각 캐시 가능한 Servlet 및 JSP 파일에 사용할 외부 캐시를 지정하십시오.
- Caching Proxy 구성
 - Caching Proxy가 동적 캐시 플러그인을 사용하도록 하십시오.
 - 동적 콘텐츠가 캐시될 원본을 지정하십시오.

프록시 캐시에 대한 IBM WebSphere Application Server 구성

Application Server에 동적 캐시 구성

로컬 동적 캐시(동적 단편 캐시라고도 함)를 사용하도록 Application Server를 구성하려면 IBM WebSphere Application Server 문서의 지시사항을 수행하십시오. 동적 단편 캐시는 Application Server Caching Proxy의 외부 캐시와 상호작용합니다.

Application Server 어댑터 구성

IBM WebSphere Application Server는 Application Server로 설치된 외부 캐시 어댑터라는 소프트웨어 모듈을 사용하여 Caching Proxy와 통신합니다.

주: 동적 캐시 구성에 대해서는 TechNote용 IBM WebSphere Application Server 지원 웹 사이트를 참조하십시오.

동적 캐시에 대한 Caching Proxy 구성

동적으로 작성된 콘텐츠(Servlet 및 JSP 결과)를 캐시하기 위해 프록시 서버를 사용 가능하게 하려면 프록시 구성 파일인 `ibmproxy.conf`에서 두 가지를 변경해야 합니다. 첫 번째 변경은 동적 캐시 플러그인 모듈을 사용 가능하게 하고, 두 번째 변경은 이 모듈이 캐시 가능한 동적 콘텐츠 원본을 인식하도록 구성합니다.

동적 캐시 플러그인을 사용 가능하게 하도록 Service 지시문 설정

Service 단계에 대한 API 지시문을 사용하여 동적 캐시 플러그인을 사용 가능하게 할 수 있습니다. 이 지시문을 작성하려면 `ibmproxy.conf` 파일을 수동으로 편집하거나, 프록시 서버가 이미 실행되고 있는 경우에는 구성 및 관리 양식에서 서버 구성 -> 요청 처리 -> API 요청 처리를 선택하십시오. 지시문 콘텐츠는 이 섹션 후반부의 예제에 표시됩니다.

동적 캐시를 사용 가능하게 하기 위한 프로토타입 Service 지시문이 `ibmproxy.conf` 파일의 API 섹션에 설명으로 존재합니다. 표제 JSP 플러그인이 포함됩니다. 프로토타입 API 지시문은 중요도 순으로 나열됨에 주의하십시오. API 지시문을 추가하여 새로운 기능 및 플러그인 모듈을 사용할 수 있을 때, 구성 파일의 표준 섹션에 표시된 대로 지시문을 나열하십시오. 선택적으로 프로토타입 API 지시문에서 설명 문자를 제거하여 각 희망 기능 또는 플러그인 지원을 포함하는 데 필요한 대로 편집할 수 있습니다.

Service 지시문을 다음 예제와 같이 설정하십시오. (각 지시문은 프록시 구성 파일의 단일 행에 표시되어야 합니다. 이 예제에는 읽기 편의를 위해 일부 행이 구분되어 있습니다.)

- AIX의 경우

```
Service /WES_External_Adapter /opt/ibm/edge/cp/lib/plugins/  
dynacache/libdyna_plugin.o:exec_dynacmd
```

- Solaris의 경우

```
Service /WES_External_Adapter /opt/ibm/edge/cp/lib/plugins/  
dynacache/libdyna_plugin.so:exec_dynacmd
```

- Linux의 경우

```
Service /WES_External_Adapter /usr/lib/libdyna_plugin.so:exec_dynacmd
```

- Windows의 경우

```
Service /WES_External_Adapter C:\Program Files\IBM\edge\cp\bin\plugins\  
dynacache\dyna_plugin.dll:exec_dynacmd
```

Caching Proxy 소프트웨어를 기본값이 아닌 다른 디렉토리에 설치한 경우에는, 이 예제 경로를 사용자 설치 경로로 대체하십시오.

파일 원본을 지정하도록 ExternalCacheManager 지시문 설정

각 Caching Proxy는 동적으로 생성된 파일의 원본 또한 인식하도록 구성해야 합니다. 이 프록시 서버에서 동적으로 생성된 콘텐츠를 캐시할 각 Application Server의 ibmproxy.conf 파일에 ExternalCacheManager 지시문을 추가하십시오. 이 지시문은 프록시 결과를 캐시할 WebSphere Application Server를 지정하고 해당 서버 콘텐츠의 최대 만기 시간을 설정합니다. 세부사항은 232 페이지의 『ExternalCacheManager — IBM WebSphere Application Server의 동적 캐시에 대한 Caching Proxy 구성』에 나 타납니다.

ExternalCacheManager 지시문에서 사용하는 서버 ID는 Application Server의 dynacache.xml 파일의 외부 캐시 그룹 스탠자에 있는 그룹 ID와 일치해야 합니다.

앞의 예제의 경우, 각 프록시의 ibmproxy.conf 파일에 다음 항목을 추가하십시오.

```
ExternalCacheManager IBM-edge-cp-XYZ-1 20 seconds
```

Caching Proxy는 ibmproxy.conf 파일의 ExternalCacheManager 항목과 일치하는 그룹 ID가 있는 IBM WebSphere Application Server의 콘텐츠만 캐시합니다.

제 23 장 프록시 서버 캐시 조정

캐시가 사용 가능하게 될 때 캐시 저장영역의 속도는 Caching Proxy의 성능에 중요한 영향을 줍니다. 이 절에서는 캐시 저장영역을 선택하고 최상의 성능을 위하여 캐시 저장 장치를 구성하는 데 필요한 암시를 제공합니다.

캐시 저장 매체 선택

Caching Proxy는 캐시 저장 매체의 두 가지 유형을 사용할 수 있습니다.

- 메모리
- 공 디스크 파티션

메모리 캐시로 검색을 빠르게 수행할 수 있지만, 메모리 캐시의 크기는 프록시 서버 시스템의 사용 가능한 메모리 용량에 의해 제한됩니다. 하나 이상의 디스크 파티션으로 구성된 디스크 캐시는 메모리 캐시보다 느리지만, 대체로 캐시할 수 있는 크기는 더 큽니다.

디스크 캐시 성능 최대화

디스크 캐시에 사용될 장치 파티션은 캐시 전용이어야 합니다. 즉, 이 물리적 디스크에 다른 파일 시스템이 있어서는 안되며 프록시 캐시 저장 외의 용도로 사용해서는 안됩니다. 또한 성능을 저하시킬 수 있으므로 프록시 캐시에 사용된 디스크에 데이터 압축을 사용하지 마십시오.

각 캐시 저장 장치(디스크 또는 파일)는 프록시 서버에 메모리 오버헤드를 야기합니다. 일반적으로 물리적인 전체 디스크를 단일 캐시 장치로 사용하는 것이 최상의 성능을 위해 바람직합니다. RAID나 기타 메커니즘을 사용하여 여러 물리적 디스크를 단일 논리적 디스크에 결합하면 성능면에서 좋지 않을 수 있습니다. 복수 디스크를 사용하려는 경우, 캐시 설정 구성 양식을 사용하거나 프록시 구성 파일에서 CacheDev 지시문을 편집하여 복수 캐시 장치로 지정하십시오. 이 방법으로 프록시 서버는 여러 개의 디스크에 대해 병행하여 수행하는 읽기 및 쓰기 기능을 조정할 수 있는데, 이는 운영 체제나 디스크 서브시스템의 성능에 의해 영향을 받지 않습니다.

캐시 가비지 콜렉션

프록시 서버용 캐시 가비지 콜렉션은 캐시에서 만기된 파일을 버려 새 요청의 파일을 캐시할 공간을 늘립니다. 가비지 콜렉션은 캐시에서 사용된 공간의 용량이 최고 수준이라는 관리자 특정 한계에 도달할 때 자동으로 시작되어, 사용된 공간의 용량이 최저 수준에 도달할 때까지 계속 실행합니다.

가비지 콜렉션 루틴은 최소 CPU 자원을 사용하고 만기되지 않은 캐시된 자료의 사용 가능성에 영향을 미치지 않기 때문에, 특정 시간에 실행하도록 가비지 콜렉션을 반드시 구성할 필요가 없습니다.

가비지 콜렉션의 성능을 개선하려면, 최고 수준 및 최저 수준을 설정할 수 있습니다. 또한 가비지 콜렉션에 사용되는 알고리즘의 유형을 구성할 수 있습니다. 가비지 콜렉션을 수정하는 데 필요한 자세한 정보는 98 페이지의 『가비지 콜렉션』을 참조하십시오.

플랫폼 고유 최적화

각 플랫폼에서 캐시 성능을 최적화하기 위해 다음과 같은 추가 제안사항이 있습니다.

AIX

디스크에 단일 논리 볼륨을 작성하십시오. 사용 가능한 모든 물리적 파티션을 사용하는 것이 좋습니다. 예를 들어, 9GB 디스크면 `cpcache1`이라고 불리는 9GB의 논리 볼륨을 작성하십시오. 이것을 포맷하고 `/dev/rcpcache1` 미처리된 논리 볼륨을 사용하여 프록시 캐시 장치로 지정하십시오.

HP-UX 및 Solaris

캐시 장치에서 디스크의 전체 크기를 사용하는 단일 파티션(또는 슬라이스)을 작성하십시오. 예를 들어, 9GB 디스크면 `c1t3d0s0`라고 불리는 9GB의 논리적 볼륨을 작성하십시오. 이것을 포맷하고 `/dev/rdisk/c1t3d0s0` 미처리된 논리 볼륨을 사용하여 프록시 캐시 장치로 지정하십시오.

Windows

전체 디스크 크기를 사용하여 단일 파티션을 작성하십시오. 예를 들어, 9GB 디스크면 `i`라고 불리는 9GB의 논리적 볼륨을 작성하십시오. 이것을 포맷하고 `\\.\i:` 원시 장치를 사용하여 프록시 캐시 장치로 지정하십시오.

프록시 서버의 캐시를 구성하고 캐시 장치를 포맷하고 지정하는 것에 대한 정보는 77 페이지의 제 4 부 『프록시 서버 캐시 구성』에 있습니다.

제 5 부 Caching Proxy 보안 구성

이 파트에서는 Caching Proxy에서 SSL을 사용하고, 암호 하드웨어를 사용 가능하게 하며, IBM Tivoli® Access Manager (이전에는 Tivoli Policy Director라고 함) 플러그인 및 PAC-LDAP 권한 부여 모듈을 사용하는 기본 보안에 대한 정보를 제공합니다.

이 파트에는 다음과 같은 장이 들어 있습니다.

119 페이지의 제 24 장 『프록시 서버 보안 정보』

121 페이지의 제 25 장 『서버 보호 설정』

125 페이지의 제 26 장 『SSL(Secure Sockets Layer)』

143 페이지의 제 27 장 『암호 하드웨어 지원 사용 가능』

145 페이지의 제 28 장 『Tivoli Access Manager 플러그인 사용』

147 페이지의 제 29 장 『PAC-LDAP 권한 부여 모듈 사용』

제 24 장 프록시 서버 보안 정보

인터넷에서 액세스 가능한 서버로 실행되고 있는 시스템이라면 모두 뜻하지 않은 관심에 노출되는 위험이 도사리고 있습니다. 권한이 없는 사람이 암호 추측, 파일 갱신, 파일 실행 또는 기밀 데이터 읽기를 시도할 수 있습니다. WWW(World Wide Web)의 매력은 개방성입니다. 그러나 웹의 사용에는 긍정적인 면과 부정적인 면이 모두 있습니다.

다음 섹션에서는 Caching Proxy 서버에 있는 파일에 액세스하는 사용자를 제어하는 방법에 대해 설명합니다.

Caching Proxy는 SSL(Secure Sockets Layer) 연결을 지원하며, 이 연결에서 암호화 및 암호 해독과 연관된 보안 전송이 클라이언트 브라우저와 대상 서버(컨텐츠 서버 또는 대리 서버) 사이에 설정됩니다.

Caching Proxy가 대리로 구성될 때, 클라이언트, 컨텐츠 서버 또는 모두와 보안 연결을 구축할 수 있습니다. SSL 연결을 사용 가능하게 하려면, 구성 및 관리 양식에서 프록시 구성 -> SSL 설정을 선택하십시오. 이 양식에서 SSL 사용 가능 선택란을 선택하고 키 링 데이터베이스와 키 링 데이터베이스 암호 파일도 지정해야 합니다.

Caching Proxy가 정방향 프록시 서버로 사용될 때, SSL 터널링이라고 하는 전달 프로토콜을 따라서 클라이언트와 컨텐츠 서버 간에 암호화된 요청을 전달합니다. 프록시 서버가 터널을 통과한 요청을 암호 해독하지 않기 때문에, 암호화된 정보는 캐시되지 않습니다. 정방향 프록시 설치에서 SSL 터널링이 사용 가능합니다. 이를 사용 불가능하게 하려면, 구성 및 관리 양식에서 프록시 구성 -> 프록시 설정을 선택한 후 이 양식의 SSL 터널링 선택란을 지우십시오.

몇 가지 기본적인 주의사항에 따라 시스템을 보호할 수 있습니다.

- 공용 액세스에 대한 서버를 로컬 또는 내부 네트워크와 분리된 네트워크에 배치하십시오.
- 원격 사용자가 서버의 내부 프로세스에 액세스할 수 있도록 허용하는 유틸리티를 사용할 수 없게 하십시오. 특히, 서버를 실행하고 있는 시스템의 텔넷, TN3270, rlogin 및 ping 클라이언트를 사용 불가능하게 해야 합니다.
- 패킷 필터링 및 방화벽을 사용하십시오.

패킷 필터링을 사용하여 데이터의 원본과 대상을 정의할 수 있습니다. 시스템이 특정 원본-대상 결합을 거부하도록 구성할 수 있습니다.

방화벽은 내부 네트워크를 인터넷과 같이 대중적으로 액세스 가능한 네트워크에서 분리합니다. 방화벽은 양방향에서 게이트웨이로 작동하면서 이곳을 지나는 통신을 조절하고 추적하는 그룹이나 단일 컴퓨터가 될 수 있습니다. IBM Firewall은 방화벽 소프트웨어의 한가지 예입니다.

- CGI 스크립트를 제어하십시오. CGI 스크립트는 사용자 ID 및 암호와 같은 민감한 데이터를 포함하는 환경 변수를 표시할 수 있기 때문에 웹 서버에서 CGI 스크립트를 사용하면 보안에 위험이 생길 수 있습니다. CGI 프로그램을 서버에서 실행하기 전에 CGI 프로그램에 대해서 정확히 이해하고 서버의 CGI 스크립트에 액세스하는 사람을 제어해야 합니다.

주: 구성 마법사를 사용하여 프록시 서버를 구성한 다음 SSL을 사용 가능하게 하려면, 443 포트에서 수신한 프록시 요청에 대해 맵핑 규칙을 작성해야 합니다. 자세한 정보는 44 페이지의 『맵핑 규칙 정의』를 참조하십시오.

예를 들면, 다음과 같습니다.

```
Proxy /* http://content server :443
```

또는

```
Proxy /* https://content server :443
```

제 25 장 서버 보호 설정

이 장에서는 보호 설정을 사용하여 서버의 데이터와 파일을 보호하는 방법에 대해 설명합니다. 보호 설정은 특히 특정 디렉토리, 파일 또는 요청이 제출하는 파일 유형에서 서버가 수신하는 요청에 기초하여 트리거됩니다. 보호 설정에서 부 지시문은 보호되는 디렉토리 또는 파일의 특성에 기초하여 액세스가 허용 또는 거부되는 방법을 제어합니다.

구성 및 관리 양식을 사용하여 보호 설정

보호 설정 및 적용 방법을 정의하려면 구성 및 관리 양식에서 서버 구성 -> 문서 보호를 선택하십시오. 다음 단계에 이 양식을 사용하십시오.

1. 이 보호 규칙의 순서를 설정하십시오.

보호 규칙은 구성 양식 테이블에 나열된 순서로 적용됩니다. 일반적으로 특정 규칙에서 일반 규칙으로 나열됩니다.

드롭 다운 메뉴 및 단추를 사용하여 보호 규칙의 배치를 지정하십시오.

2. 요청 템플릿 정의

클라이언트가 프록시 서버로 전송한 요청의 내용과 비교되는 요청 템플릿에 기초하여 보호가 활성화됩니다.

요청은 서버 호스트 이름 다음에 오는 전체 URL의 일부입니다. 예를 들어 서버 이름이 fine.feathers.com이고 브라우저 사용자가 `http://fine.feathers.com/waterfowl/schedule.html` URL을 입력하면, 서버가 `/waterfowl/schedule.html` 요청을 수신합니다. 요청 템플릿은 보호가 요구되는 디렉토리나 파일 이름, 또는 이들 모두를 지정합니다. 예를 들어, 방금 설명한 요청 템플릿(`/waterfowl/schedule.html`)에 따라 보호를 활성화시키는 일부 요청에는 `/waterfowl/*` 및 `/*schedule.html`이 포함됩니다.

URL 요청 템플릿 필드에 요청 템플릿을 입력하십시오.

3. 보호 설정 정의

보호 설정은 요청 템플릿과 일치하는 요청에 대한 조치를 Caching Proxy에 알립니다. 이름 지정 보호 설정을 사용하거나 문서 보호 양식에 새 설정을 정의할 수 있습니다.

이름 지정 설정을 사용하려면 이름 지정 보호 단일 선택 단추를 누른 후, 해당 필드에 이름을 입력하십시오. 새 설정을 정의하려면 인라인 단일 선택 단추를 누른 후, 해당 지시사항을 따르십시오(6단계 참조).

4. 요청자 주소 선택(선택적)

다른 서버 주소에서 온 요청에 다른 규칙을 적용할 수 있습니다. 예를 들어, 로그 파일에 대한 요청이 사용자 회사에 지정된 IP 주소에서 수신되면, 이러한 요청에 다른 보호 설정을 적용할 수 있습니다.

주: 차단할 요청자 주소의 경우, DNS 조회를 사용 가능해야 합니다. 225 페이지의 『DNS-Lookup — 서버가 클라이언트 호스트 이름을 조회할지 여부 지정』을 참조하십시오.

요청자 주소를 규칙에 포함시키려는 경우에는 서버 IP 주소 또는 호스트 이름 필드에 이를 입력하십시오.

5. 제출을 누르십시오.

이름 지정 보호 설정을 사용했으면 더 이상 입력할 필요가 없습니다. 인라인 보호 설정을 선택했거나 존재하지 않는 이름 지정 설정을 지정한 경우에는, 시스템이 추가 양식을 엽니다.

6. 보호 세부사항 설정

기존의 이름 지정 보호 설정을 지정하지 않은 경우, 요청 템플릿과 일치하는 문서나 디렉토리에 액세스할 수 있는 사용자 및 사용자에게 허용된 조치를 지정하도록 추가 양식이 열립니다.

- 암호 인증 설정—사용자 인증에 사용할 암호 파일, 그룹 파일 또는 이 둘 모두를 지정하십시오. 또한 요청자 이름 및 암호가 프롬프트되면 서버 식별에 사용되는 이름을 지정하십시오.

주: 일부 브라우저는 사용자 ID 및 암호를 캐시하여 서버 ID와 연관시킵니다. 동일한 암호 파일에 항상 동일한 서버 ID를 사용하면, 보다 편리해 집니다.

- 권한—보호 설정된 파일을 읽거나 쓰거나 삭제할 수 있도록 허가된 사용자나 그룹을 지정하십시오.

7. 제출을 누르십시오.

8. 서버를 재시작하십시오.

구성 파일 지시문을 사용하여 보호 설정

Caching Proxy 구성 파일을 직접 편집하여 보호를 설정하려면, 먼저 다음 사항을 이해해야 합니다.

- Protect, defProt 및 Protection 지시문 간의 차이

- Protect 지시문은 요청 템플리트를 보호 설정에 연결함으로써 보호를 설정합니다. 274 페이지의 『Protect — 템플리트와 일치하는 요청에 대한 보호 설정 활성화』에서 자세한 내용을 참조하십시오.
- defProt 지시문은 특정 요청 템플리트에 대한 기본 보호 설정을 설정합니다. 217 페이지의 『DefProt — 템플리트와 일치하는 요청에 대한 기본 보호 설정 지정』에서 자세한 내용을 참조하십시오.
- Protection 지시문은 이름 지정 보호 설정을 정의하는 데 사용됩니다. 279 페이지의 『Protection — 구성 파일 내에 명명된 보호 설정』에서 자세한 내용을 참조하십시오.

- 보호와 요청 경로 지정이 상호작용하는 방법

Map, Exec, Pass 및 Proxy와 같은 요청 경로 지정 지시문은 서버가 승인할 요청 및 요청을 실제 파일 위치로 경로 재지정하는 방법을 제어하는 데 사용됩니다. 요청 경로 지정 지시문은 보호 지시문과 동일한 유형의 요청 템플리트를 사용합니다. 각 요청에 대하여 처음 일치하는 템플리트와 연관된 방향이 실행되기 때문에 보호 지시문이 구성 파일에서 경로 지정 지시문 앞에 나열되어야만 보호가 올바르게 작동합니다. 자세한 정보는 274 페이지의 『Protect — 템플리트와 일치하는 요청에 대한 보호 설정 활성화』를 참조하십시오.

- 인라인 보호 설정과 이름 지정 보호 설정 간의 차이

보호 지시문은 인라인 보호 설정을 지정하거나 기존의 명명된 설정을 참조하는 데 사용할 수 있습니다. 이 두 가지 유형의 명령문에 대한 구문은 약간 다릅니다. 자세한 내용은 274 페이지의 『Protect — 템플리트와 일치하는 요청에 대한 보호 설정 활성화』를 참조하십시오.

- 보호 설정 기록 방법

보호 설정은 보호 부 지시문을 사용하는 일련의 명령문입니다. 보호 설정 기록에 대한 구문 및 참조 정보는 185 페이지의 부록 B 『구성 파일 지시문』에 나와 있습니다. 다음 참조 섹션을 참조하십시오.

- 274 페이지의 『Protect — 템플리트와 일치하는 요청에 대한 보호 설정 활성화』
- 279 페이지의 『Protection — 구성 파일 내에 명명된 보호 설정』
- 280 페이지의 『Protection subdirectives — 자원 세트가 보호되는 방법 지정』

기본 보호 설정

기본 프록시 구성 파일에는 /admin-bin/ 디렉토리의 파일에 액세스하기 위해서 관리자 ID 및 암호가 필요한 보호 설정이 있습니다. 이 설정은 구성 및 관리 양식에 대한 액세스를 제한합니다.

제 26 장 SSL(Secure Sockets Layer)

SSL(Secure Sockets Layer)은 인터넷을 통해 전송하기 전에 정보를 자동으로 암호화하고, 사용하기 전에 상대방에서 암호 해독하기 위한 시스템입니다. 이것은 인터넷에서 전송되는 동안 신용 카드 번호와 같은 민감한 정보를 보호합니다.

Caching Proxy는 다음 섹션에서 설명한 것처럼 SSL을 사용하여 대리 서버를 보호하고 보안 원격 관리를 제공합니다. SSL은 백엔드 서버와의 연결을 보호하는 데 사용될 뿐만 아니라(예: 콘텐츠 서버 또는 Application Server), 프록시 서버와 클라이언트 간의 통신도 보호할 수 있습니다.

정방향 프록시의 경우, Caching Proxy는 SSL 터널링을 지원하여 SSL을 생략하고 이미 암호화된 데이터를 변경하지 않고 전달합니다.

SSL 핸드셰이크

SSL 보호는 보안 연결 요청이 한 시스템에서 다른 시스템으로 전송될 때—예를 들어, 브라우저가 요청을 대리 프록시 서버로 전송할 때 초기화됩니다. `https://` 요청 구문은 브라우저에게 서버가 보안 연결 요청을 대기하는 장소인 포트 443(루틴 요청의 경우 포트 80 대신)으로 요청을 전송하도록 알려줍니다. 브라우저와 서버 간에 보안 세션을 구축하려면, 두 시스템은 암호 스펙에 동의하고, 정보를 암호화하고 해독하는 데 사용되는 키를 선택하는 *SSL 핸드셰이크*라고 하는 교환을 수행합니다. 키는 자동으로 생성되며 세션이 종료할 때 유효기간이 끝납니다. 일반적인 시나리오(SSL 버전 3으로 가정)는 다음과 같습니다.

1. 클라이언트 준비

클라이언트는 클라이언트의 암호화 성능을 설명하는 클라이언트 준비 메시지를 전송하여 Caching Proxy의 SSL 세션을 시작합니다.

2. 서버 준비

서버는 클라이언트에게 인증을 전송하고 데이터 암호화에 사용할 암호 세트를 선택합니다.

3. 클라이언트 종료

클라이언트는 암호화된 데이터에 대해 대응하는 암호화 키를 작성하는 데 사용되는 암호 키 정보를 전송합니다. 이 키 자료는 *프리마스터 기밀*이라고 하며, 서버의 공용 키(서버의 인증에서 획득, 130 페이지의 『키 및 인증 관리』 참조)로 암호화됩니다. 서버와 클라이언트 모두 읽기 및 쓰기 대칭 암호화 키를 프리마스터 기밀에서 끌어낼 수 있습니다.

4. 서버 종료

서버는 전체 핸드셰이크 프로토콜에 대한 최종 확인 및 MAC(메시지 인증 코드)를 전송합니다.

5. 클라이언트 유효화

클라이언트는 서버 종료 메시지를 유효화하는 메시지를 전송합니다.

6. 데이터 흐름 보안

클라이언트가 서버 종료 메시지를 유효화하면, 암호화된 데이터 흐름이 시작됩니다.

Caching Proxy를 보안 연결에 대한 끝점으로 사용하면, 콘텐츠 서버 또는 Application Server의 로드를 줄일 수 있습니다. Caching Proxy는 보안 연결을 유지할 때 암호화 및 암호 해독을 수행하고 모든 CPU 집중 작업인 키 작성을 수행합니다. 또한 Caching Proxy로 각 키의 사용을 최대화할 수 있도록 SSL 세션 시간 종료를 구성할 수 있습니다.

SSL 한계

다음과 같은 한계사항이 WebSphere Application Server의 Caching Proxy SSL에 적용됩니다.

- Caching Proxy는 인증 권한으로 사용될 수 없습니다(130 페이지의 『키 및 인증 관리』 참조).
- 일부 브라우저는 Caching Proxy에 사용된 모든 암호화 기술을 지원하지는 않습니다.

SSL 성능 조정

높은 HTTPS 통신량으로, Caching Proxy 서버는 높은 CPU 사용을 발생시킵니다. 환경 변수(GSK_V3_SIDCACHE_SIZE) 및 프록시 지시문(SSLV3Timeout)을 조정하면, 프록시 서버가 로드를 핸들하고 CPU 사용을 줄일 수 있습니다.

SSL 세션 ID는 브라우저 및 서버에서 사용되는 암호화 또는 암호 해독 키를 포함하여 재사용 가능한 SSL 세션을 식별하고, 새 연결 시 발생하는 불필요한 SSL 핸드셰이크를 피하기 위해 사용하는데 이는 서버에서 CPU 시간을 많이 소비하기 때문입니다. Caching Proxy 서버에 대한 GSKit 라이브러리는 SSL 세션 ID를 지원하고 SSL 세션 ID 캐시를 포함합니다. 기본적으로, SSL 세션 ID 캐시는 512 항목을 포함하고 있습니다. 항목 한계에 도달하는 경우, 가장 오래된 세션 항목이 제거되고, 새 항목이 캐시에 추가됩니다.

GSK_V3_SIDCACHE_SIZE 환경 변수를 사용하여 SSL 세션 ID 캐시의 기본 크기를 변경하십시오. 변수의 유효값은 1 - 4096입니다. 크기를 늘리면, 캐시 SSL 세션을 찾는 데 필요한 찾아보기 시간이 늘어납니다. 그러나, 늘어난 찾아보기 시간은 SSL 연

결을 하는 데 필요한 오버헤드에 비해 중요하지 않습니다. 캐시 크기를 늘리면, 프록시 서버가 더 많은 동시 SSL 세션을 핸들하고, 프록시 서버에 높은 HTTPS 로드가 있는 경우, CPU 사용을 줄일 수 있습니다.

Caching Proxy에는 조정 가능한 지시문 `SSLV3Timeout`이 있습니다. (307 페이지의 『`SSLV3Timeout` — `SSLV3` 세션이 만기되기 전에 대기할 시간 지정SSLV3Timeout 값을 일반적인 보안 클라이언트 세션의 길이로 설정하는 것을 권장합니다. 제한시간이 너무 짧게 설정된 경우, 프록시의 성능을 저하시킬 수 있습니다. 이는 여러 SSL 핸드셰이크가 단일한 보안 세션을 완료하는 것이 필요하기 때문입니다. 그러나, 값이 너무 길게 설정된 경우, 보안 세션의 보안에 지장을 줄 수 있습니다.

SSL 터널링

이는 정방향 프록시 구성에만 적용합니다.

Caching Proxy가 정방향 프록시로 구성될 때 SSL 터널링을 사용하여 클라이언트와 콘텐츠 서버 간의 보안 연결을 지원합니다. SSL 터널링에서 암호화된 데이터는 프록시 서버를 통하여 변경되지 않고 전달됩니다. 프록시 서버가 데이터를 암호 해독하지 않기 때문에 요청이나 문서 헤더를 읽는 데 프록시 서버가 필요한 기능은 SSL 터널링에서 지원되지 않습니다. 또한 터널을 통과한 요청은 캐시되지 않습니다.

128 페이지의 그림 2는 SSL 터널링을 사용하여 연결이 어떻게 이루어지는 지를 보여줍니다.

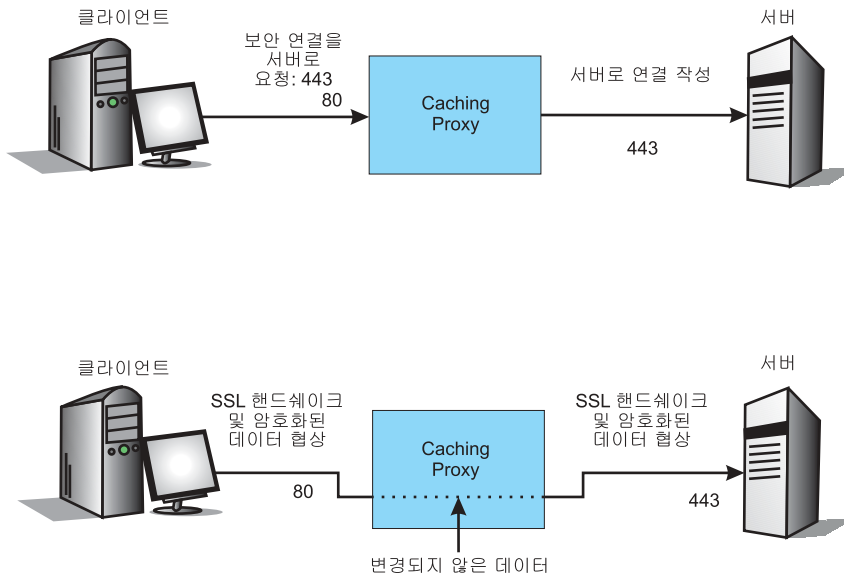


그림 2. SSL 터널링

SSL 터널링 프로세스는 다음과 같습니다.

1. 클라이언트가 터널링 요청을 합니다. `CONNECT server-host-name:port HTTP/1.1`(또는 `HTTP/1.0`) 포트 번호는 선택적이며 주로 443입니다. 정방향 프록시가 브라우저에 구성된 경우, 클라이언트의 브라우저는 모든 HTTPS 요청에 대해 자동으로 `CONNECT` 요청을 먼저 프록시 서버에 전송합니다.
2. 프록시는 포트 80의 연결을 승인하고 요청을 수신하여 클라이언트가 요청한 포트의 대상 서버로 연결합니다.
3. 프록시는 연결이 작성되었음을 클라이언트에 응답합니다.
4. 프록시는 SSL 핸드셰이크 메시지를 클라이언트에서 대상 서버로, 대상 서버에서 클라이언트의 양방향으로 릴레이합니다.
5. 일단 보안 핸드셰이크가 완료 후 프록시가 암호화된 데이터를 전송하고 수신하여 클라이언트나 대상 서버에서 암호가 해독되도록 합니다.
6. 클라이언트나 대상 서버가 각 포트 단기를 요청하면 프록시 서버는 두 연결(포트 443 및 80)을 모두 닫고 표준 활동을 재개합니다.

SSL 터널링 구성

정방향 프록시 설정에서는 SSL 터널링만 사용 가능합니다. SSL 터널링을 사용 가능하게 하려면, 구성 및 관리 양식에서 **프록시 구성** -> **프록시 설정**을 선택하십시오. **SSL 터널링** 선택란을 선택하십시오.

`CONNECT` 메소드(기본적으로 사용 불가능함)가 또한 SSL 터널링 접속을 위해 사용 가능해야 합니다. 구성 양식에서 이 메소드를 사용 가능하게 하려면 **서버 구성** -> **요청 처리**를 선택한 후, **HTTP 메소드** 양식을 사용하십시오.

세 개의 옵션(OutgoingPorts, OutgoingIPs, IncomingIPs)이 고급 SSL 터널링 보안을 위해 Enable CONNECT 지시문에 대해 제공됩니다. 최소한 OutgoingPorts에 대한 값을 지정하는 것이 필요합니다. 그렇지 않은 경우, CONNECT 메소드를 사용 가능으로 할 수 없습니다.

- OutgoingPorts (원격 서버 포트로 SSL 터널링 액세스 제한). 형식은 다음과 같습니다.

```
Enable CONNECT OutgoingPorts [all | [port1|port1-port2|port1-*],...]
```

클라이언트가 SSL 터널링에 대해 원격 서버의 포트 443에만 연결하도록 허용하려면, 다음 지시문을 설정하십시오. (포트 443은 정상적으로 원격 서버에서 HTTPS 요청을 위한 것입니다.)

```
Enable CONNECT OutgoingPorts 443
SSLTunneling on
```

클라이언트가 SSL 터널링에 대해 원격 서버의 모든 서버에 연결하도록 허용하려면, 다음 지시문을 설정하십시오.

```
Enable CONNECT OutgoingPorts all
SSLTunneling on
```

클라이언트가 SSL 터널링에 대해 원격 서버의 포트 80, 8080-8088 및 9000과 위의 포트에 연결하도록 허용하려면, 다음 지시문을 설정하십시오.

```
Enable CONNECT OutgoingPorts 80,8080-8088,9000-*
SSLTunneling on
```

포트 및 포트 범위는 목록에서 공백 없이 쉼표로 구분됩니다.

중요: 정방향 프록시 구성에 대해, OutgoingPorts 옵션에 443 또는 모두를 지정하여, 정상적인 SSL 터널링을 사용 가능하게 하십시오.

- OutgoingIPs (원격 서버의 IP 주소로 SSL 터널링 액세스 제한). 형식은 다음과 같습니다.

```
Enable CONNECT OutgoingIPs [[!]IP_pattern,...]
```

예를 들어, 클라이언트가 IP/호스트 이름 *.ibm.com과 일치하고, 192.168.*.*과 일치하지 않는 원격 서버의 모든 포트에 연결하도록 허용하려면, 다음 지시문을 설정하십시오.

```
Enable CONNECT OutgoingPorts all OutgoingIPs *.ibm.com,!192.168.*.*
SSLTunneling on
```

주: IP_patterns는 목록에서 공백 없이 쉼표로 구분됩니다.

- IncomingIPs(클라이언트의 IP 주소로 SSL 터널링 액세스 제한). 형식은 다음과 같습니다.

```
Enable CONNECT IncomingIPs [[!]IP_Pattern,...]
```

예를 들어, IP 주소 192.168.*.*에서 들어오는 클라이언트가 SSL 터널링에 대해 원격 서버의 모든 포트에 연결하도록 하려면, 다음 지시문을 설정하십시오.

```
Enable CONNECT OutgoingPorts all IncomingIPs 192.168.*.*
SSLTunneling on
```

주:

1. 192.168.*.*을 내부 LAN IP 마스크로 가정한 경우, 위의 옵션은 내부 사용자만 연결 메소드 및 SSL 터널링 기능을 사용하도록 허용합니다.
2. IP_patterns는 목록에서 공백 없이 쉼표로 구분됩니다.

프록시 구성 파일을 편집하여 SSL 터널링 및 CONNECT 지시문을 사용 가능하게 하는 것에 대한 자세한 정보는 다음 지시문에 대한 185 페이지의 부록 B 『구성 파일 지시문』의 참조 섹션을 참조하십시오.

- 226 페이지의 『Enable — HTTP 메소드 사용 가능』
- 306 페이지의 『SSLTunneling — SSL 터널링 사용 가능』

보안 원격 관리 구성

Caching Proxy의 원격 관리는 SSL과 암호 인증이 제공하는 보안 기능을 사용하여 보호될 수 있습니다. 이렇게 되면 권한없는 사용자가 프록시 서버에 액세스할 가능성이 크게 줄어듭니다.

서버를 원격으로 관리할 때 SSL을 적용하려면 `https://` 요청이 아니라 `http://` 요청을 사용하여 구성 및 관리 양식을 여십시오. 예제는 다음과 같습니다.

```
https://your.server.name/yourFrontPage.html
```

키 및 인증 관리

위에서 설명한 대로 SSL을 구성하기 전에 키 데이터베이스를 설정하여 인증을 획득하거나 작성해야 합니다. 인증은 서버 ID를 인증하는 데 사용됩니다. IBM Key Management 유틸리티(iKeyman이라고도 함)를 사용하여 인증 파일을 설정하십시오. 유틸리티는 Application Server와 함께 GSKit 소프트웨어에 있습니다. GSKit에도 인증 파일을 열기 위한 Java 기반의 그래픽 인터페이스가 있습니다.

다음은 SSL 키 및 인증을 설정하기 위한 기본 단계입니다.

1. GSKit가 설치되었는지 확인하십시오. 대부분의 플랫폼에서 Caching Proxy 컴포넌트와 함께 자동으로 설치됩니다. 패키지의 이름은 `gsk7ikm`(i386용 Linux 시스템에서는 `gsk7ikm_gcc295`)입니다. GSKit은 일반적으로 `ibm/gsk7/` 디렉토리(AIX 시스템에서는 `ibm/gskit/`)에 설치됩니다. Windows 플랫폼의 경우에는 이를 시작 메뉴에서 액세스할 수도 있습니다.

주: Windows에서 InstallShield를 사용할 때 GSKit는 설치하지 않는 경우, 매체 디렉토리 경로에 공백이 포함되어 있지 않음을 확인하십시오.

2. 키 관리자를 사용하여, 보안 네트워크 통신에 대한 키를 작성하고 인증 기관에서 인증을 받으십시오. 인증 기관에서 인증을 받으려고 대기하는 동안 자체 인증을 작성할 수 있습니다.
3. 키 데이터베이스를 작성하고 키 데이터베이스 암호를 지정하십시오.

주: 키 및 keystash 파일은 Caching Proxy가 설치 해제될 때마다 설치 해제됩니다. 인증 기관에 새로운 인증을 요청해야 하는 상황을 피하려면, 프록시 소프트웨어를 설치 해제하기 전에 이 두 개 파일 백업 사본을 다른 디렉토리에 저장하십시오.

Linux를 제외한 모든 운영 체제에서는 인증이 만기된 경우, Caching Proxy가 올바르게 시작되지 않고, 키 데이터베이스가 만기되었음을 표시하는 오류 메시지가 나타납니다. Linux에서는 프록시가 시작되지만, 프로세스가 빠르게 사라지고 오류 메시지가 생성되지 않습니다.

Red Hat Enterprise Linux 3.0 시스템에서 이 문제점을 방지하려면, GCC 패키지가 다음 레벨 이상에 있음을 확인하십시오.

- libstdc++-3.2.3-52
- libgcc-3.2.3-52

인증 기관

공용 키는 서버의 신뢰할 수 있는 루트 CA로 지정된 CA(인증 기관)에서 디지털로 서명한 인증과 연관되어야 합니다. 인증 요청을 CA(인증 기관) 제공업체에 제출하여 서명된 인증을 구입할 수 있습니다. Caching Proxy는 다음과 같은 외부 CA를 지원합니다.

- VeriSign
- Thawte

기본적으로 다음 기관이 신뢰할 수 있는 인증 기관으로 지정되었습니다.

- Verisign 클래스 1 개별 가입자 인증 기관 - 사람이 검증되지 않음
- Verisign 클래스 2 개별 가입자 인증 기관 - 사람이 검증되지 않음
- Verisign 클래스 3 개별 가입자 인증 기관 - 사람이 검증되지 않음
- VeriSign 클래스 3 국제 서버 인증 기관
- VeriSign 클래스 2 OnSite 개별 인증 기관
- VeriSign 클래스 1 공용 기본 인증 기관
- VeriSign 클래스 2 공용 기본 인증 기관
- VeriSign 클래스 3 공용 기본 인증 기관

- VeriSign 클래스 1 공용 기본 인증 기관 - G2
- VeriSign 클래스 2 공용 기본 인증 기관 - G2
- RSA 보안 서버 인증 기관(VeriSign)
- Thawte 개인 기본 인증 기관
- Thawte 개인 프리메일 인증 기관
- Thawte 개인 프리미엄 인증 기관
- Thawte 프리미엄 서버 인증 기관
- Thawte 서버 인증 기관

IBM Key Manager 유틸리티 사용

이 절에서는 IBM Key Manager 유틸리티(iKeyman)의 사용에 대한 빠른 참조를 제공합니다. 키 관리자를 사용하여, SSL 키 데이터베이스 파일, 공용-개인용 키 쌍 및 인증 요청을 작성하십시오. 인증 기관 서명된 인증을 받은 후, 키 관리자를 사용하여 원래의 인증 요청을 작성한 키 데이터베이스에 인증을 배치하십시오.

IBM Key Manager 및 GSKit에 대한 보다 자세한 문서는 GSKit 소프트웨어에 패키징되어 있습니다.

키 관리자를 실행하도록 시스템 설정

IKeyman GUI를 시작하기 전에 다음을 수행하십시오.

1. IBM 또는 IBM 호환 32비트 Java 2 Technology, 버전 1.4.2를 설치하십시오.
2. JAVA_HOME을 Java 디렉토리 위치로 설정하십시오. 예제는 다음과 같습니다.
 - Windows: set JAVA_HOME=C:\Program Files\IBM\Java142
 - Linux 및 UNIX: export JAVA_HOME=/usr/opt/IBMJava2-142
3. ibmjssc.jar 및 gskikm.jar(있을 경우) 및 ibmjcaprovider.jar 파일을 JAVA_HOME/jre/lib/ext 디렉토리에서 제거하십시오.

주:

- a. Solaris의 경우, JAVA_HOME/jre/lib/ext 디렉토리를 JAVA_HOME/lib/ext/ 디렉토리로 대체하십시오.
 - b. 다른 제품(예를 들어 WebSphere Application Server)과 연관된 JDK에서 jars를 이동하거나 삭제하지 마십시오. 이렇게 하면 올바르게 조작되거나 운영되지 않을 수 있습니다. JDK가 사용 중인지 여부가 확실하지 않은 경우, IBM 키 관리 유틸리티에 대한 별도의 JDK를 설치하십시오.
4. 다음 모든 jar 파일은 현재 *GSKit_Installation_path/classes/jre/lib/ext/*에 있습니다.
 - 지정된 jar 파일을 JAVA_HOME/jre/lib/에 복사

```
ibmjcefw.jar
ibmpkcs11.jar
```

- 지정된 jar 파일을 JAVA_HOME/jre/lib/ext에 복사

```
ibmjceprovider.jar
ibmpkcs.jar
```

- 지정된 jar 파일을 JAVA_HOME/jre/lib/security에 복사

```
local_policy.jar
US_export_policy.jar
```

5. IBM JCE, IBM CMS 및/또는 IBMJCEFIPS 서비스 프로바이더를 등록하십시오.

Sun 프로바이더 이후에 IBM CMS 및 IBM JCE 프로바이더 모두를 추가하도록 JAVA_HOME/jre/lib/security/java.security 파일을 갱신하십시오. 예제는 다음과 같습니다.

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.ibm.spi.IBMCMSProvider
security.provider.3=com.ibm.crypto.provider.IBMJCE
```

예제 java.security 파일은 *GSKit_Installation_path/classes/gsk_java.security*에 있습니다.

- FIPS 조작을 사용 가능하게 하려면 Sun 프로바이더 이후에 IBMJCEFIPS도 추가하도록 update the JAVA_HOME/jre/lib/security/java.security 파일을 갱신하십시오. IBMJCEFIPS 프로바이더가 IBMJCE보다 높은 우선순위로 등록되었는지 확인하십시오. 예제는 다음과 같습니다.

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.ibm.spi.IBMCMSProvider
security.provider.3=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.4=com.ibm.crypto.provider.IBMJCE
```

6. (선택적) JSSE 사용자이며 JSSE를 사용하여 기밀 하드웨어를 액세스하는 경우, JAVA_HOME/jre/lib 디렉토리의 ibmpkcs11.jar를 설치하고

*GSKit_Installation_path/classes/native/native-support.zip*의 지시사항에 따라 기밀 하드웨어 공유 라이브러리를 설정하십시오.

주: ibmpkcs11.jar를 2002년 8월 5일 이후에 릴리스된 JSSE 패키지에서 찾을 수도 있습니다. IBMPKCS11 서비스 프로바이더를 등록하기 위해, JAVA_HOME/jre/lib/security/java.security 파일을 갱신하는 예는 다음과 같습니다.

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.ibm.crypto.provider.IBMJCE
security.provider.3=com.ibm.crypto.pkcs11.provider.IBMPKCS11
```

키 관리자 시작

키 관리자 그래픽 사용자 인터페이스를 다음과 같이 시작하십시오.

- Linux 및 UNIX 플랫폼에서는 명령 프롬프트에서 gsk7ikm을 입력하십시오.

- Windows 플랫폼에서는 시작 -> 프로그램 -> IBM WebSphere -> Edge Components -> Caching Proxy -> 키 관리 유틸리티 시작을 누르십시오.

이 세션 중에 새 키 데이터베이스를 작성하면, 이 파일이 키 관리자를 시작한 디렉토리에 저장됩니다.

새 키 데이터베이스, 암호, 숨김 파일 작성

키 데이터베이스는 서버가 하나 이상의 키 쌍 및 인증을 저장하기 위해 사용되는 파일입니다. 모든 키 쌍과 인증에 하나의 키 데이터베이스를 사용하거나 여러 데이터베이스를 작성할 수 있습니다. 키 관리 유틸리티는 새 키 데이터베이스를 작성하고 암호 및 숨김 파일을 지정하는 데 사용됩니다.

키 데이터베이스 및 숨김 파일을 작성하려면 다음을 수행하십시오.

1. 키 관리 유틸리티를 시작하십시오.
2. 기본 메뉴에서 키 데이터베이스 파일 -> 새로 작성을 선택하십시오.
3. 새로 작성 대화상자에서 **CMS** 키 데이터베이스 파일 유형을 선택했는지 확인하십시오. 키 데이터베이스 이름 및 파일 위치를 입력하거나 **key.kdb** 기본값을 승인하십시오. 확인을 누르십시오.
4. 암호 프롬프트 대화상자에서 이 데이터베이스에 대한 암호를 입력하고 확인하십시오. 확인을 누르십시오.
5. 암호 파일을 숨기기 위한 확인란을 선택하십시오. 프롬프트되면 암호를 입력하고 확인하여 검증하십시오. 다음 메시지가 표시됩니다. DB-Type: CMS key database file *keyfile_database_name*

주: 암호 파일을 스테시하지 않으면 서버가 시작해도 포트 443에서 인식되지 않습니다.

새 키 데이터베이스를 작성할 때 지정한 암호는 개인용 키를 보호합니다. 개인용 키는 문서에 서명하거나 공용 키로 암호화된 메시지를 해독할 수 있는 유일한 키입니다.

암호를 지정할 때는 다음 가이드 라인을 사용하십시오.

- 암호는 U.S. 영어 문자 세트로 구성되어야 합니다.
- 암호는 문자 길이가 최소 6개이며, 최소한 두 개의 비연속 숫자를 포함해야 합니다. 본인의 이름이나 직계 가족의 이름, 이니셜, 생일 등 사용자에게 대해서 공개적으로 얻을 수 있는 정보로 암호를 구성하지 않도록 하십시오.
- 암호를 숨기십시오.

키 데이터베이스 암호는 자주 변경하는 것이 좋습니다. 그러나 암호의 만기 날짜를 지정한 경우에는, 암호를 변경한 시기의 기록을 보존하십시오. 암호가 변경하기 전에 만기 되면 오류 로그에 메시지가 작성되고 서버가 시작되지만, 보안 네트워크 연결은 작성할 수 없습니다.

다음 단계를 수행하여 키 데이터베이스 암호를 변경하십시오.

1. 기본 메뉴에서 키 데이터베이스 파일 -> 열기를 누르십시오.
2. 열기 대화 상자에 키 데이터베이스 이름을 입력하거나 **key.kdb** 기본값을 승인하십시오. 확인을 누르십시오.
3. 암호 프롬프트 대화 상자에서 성립된 암호를 입력하고 확인을 누르십시오.
4. 기본 메뉴에서 키 데이터베이스 파일 -> 암호 변경을 누르십시오.
5. 암호 변경 대화 상자에 새 암호를 입력하고 확인하십시오. 확인을 누르십시오.

프록시 및 LDAP 서버 간 SSL 연결에 대해, pac_keyring.pwd 파일에 키 데이터베이스 암호를 입력하십시오. (pac_keyring.pwd 파일은 IKeyMan에서 생성된 숨김 파일입니다.)

새 키 쌍 및 인증 요청 작성하기

키 데이터베이스는 키 쌍 및 인증 요청을 저장합니다. 공용-개인용 키 쌍 및 인증 요청을 작성하려면 다음 단계를 수행하십시오.

1. 키 데이터베이스를 작성하지 않았으면, 134 페이지의 『새 키 데이터베이스, 암호, 숨김 파일 작성』에 있는 지침을 따르십시오.
2. 키 관리 유틸리티의 기본 메뉴에서 키 데이터베이스 -> 파일 -> 열기를 누르십시오.
3. 열기 대화 상자에 키 데이터베이스 이름을 입력하십시오(기본값을 사용하는 경우, **key.kdb**를 누르십시오). 확인을 누르십시오.
4. 암호 프롬프트 대화 상자에서 암호를 입력하고 확인을 누르십시오.
5. 기본 메뉴에서 작성 -> 새 인증 요청을 누르십시오.
6. 새 키 및 인증 요청 대화 상자에서 다음 사항을 지정하십시오.
 - 키 레이블: 데이터베이스에 키와 인증을 식별하는 데 사용되는 이름(레이블)을 입력하십시오. my self-signed certificate 또는 www.companyA.com 등이 이름의 예입니다.
 - 키 크기: 키의 크기(예: 1024). (128비트 암호화의 장점을 이용하려면 1024의 키 크기를 권장함).
 - 조직 이름: 키와 연관된 조직의 이름. 예제: Company A.
 - 조직 단위(선택적)
 - 소재(선택적)
 - 시/도(선택적)
 - 우편번호(선택적)
 - 국가: 국가 코드. 최소한 두 문자를 지정해야 합니다(예: US).

- 인증 요청 파일: 요청 파일의 이름 선택적으로, 기본 이름을 사용할 수 있습니다.
7. 확인을 누르십시오. 확인 메시지가 나타납니다.
파일
`keyfile_database_name`에 새 인증 요청이
작성되었습니다.
 8. 확인을 누르십시오. 입력한 레이블 이름은 개인 인증 요청 표제 아래에 표시됩니다.
 9. 정보 대화 상자에서 확인을 누르십시오. 파일을 인증 기관으로 전송하도록 다시 알려줍니다.
 10. 자체 인증(자세한 내용은 다음 섹션 "자체 인증 작성" 참조)을 작성하지 않았으면, 인증 요청을 인증 기관으로 전송하십시오.
 - 키 관리자가 계속 실행되도록 하십시오.
 - 웹 브라우저를 시작하고, 인증을 획득하려는 인증 기관의 URL을 입력하십시오.
 - 인증 요청을 전송하려면 인증 기관에서 제공한 명령을 따르십시오.

인증 요청을 수행하려면 2주에서 3주 정도 걸릴 수 있습니다. 인증 기관이 인증 요청을 처리할 때까지 대기하는 중, 직접 인증 기관으로서 iKeyman을 사용하여, 자체 서버 인증을 작성하여 클라이언트와 Caching Proxy 서버 간에 SSL 세션이 사용 가능하도록 할 수 있습니다.

자체 인증 작성

인증이 발행되기를 기다리는 동안 클라이언트와 프록시 서버 간에 SSL 세션을 사용 가능하도록 하기 위해, 키 관리 유틸리티를 사용하여 자가 서명된 서버 인증을 작성할 수 있습니다. 목적을 검사하는 데 자체 인증을 사용할 수도 있습니다.

이 프로시저를 수행하여 자체 인증을 작성하십시오.

1. 키 데이터베이스를 작성하지 않았으면, 134 페이지의 『새 키 데이터베이스, 암호, 숨김 파일 작성』에 있는 지침을 따르십시오.
2. 키 관리 유틸리티의 기본 메뉴에서 키 데이터베이스 -> 파일 -> 열기를 누르십시오.
3. 열기 대화 상자에 키 데이터베이스 이름(기본값 **key.kdb**를 승인)을 입력하십시오. 확인을 누르십시오.
4. 암호 프롬프트 대화 상자에서 암호를 입력하고 확인을 누르십시오.
5. 키 데이터베이스 내용 프레임에서 개인 인증을 선택하고 자체 인증 신규 작성 단추를 누르십시오.
6. 자체 인증 신규 작성 창에서 다음 사항을 지정하십시오.
 - 키 레이블: 데이터베이스에서 키와 인증을 식별하는 데 사용되는 이름(레이블) 예제: 자체 인증

- 키 크기: 키의 크기, 예제: 512.
 - 일반 이름: 서버의 전체 호스트 이름, 예제 `www.myserver.com`
 - 조직 이름: 키와 연관된 조직의 이름 예제: `Company A`
 - 조직 단위(선택적)
 - 소재(선택적)
 - 시/도(선택적)
 - 우편번호(선택적)
 - 국가: 국가 코드. 최소한 두 문자를 지정해야 합니다(예제: `US`).
 - 유효 기간: 인증이 유효한 기간
7. 확인을 누르십시오.
 8. 키 파일과 숨김 파일을 구성 설정에 추가하여, 서버에 키 데이터베이스를 등록하십시오(134 페이지의 『새 키 데이터베이스, 암호, 숨김 파일 작성』 참조).

키 내보내기

이 프로시저를 사용하여 다른 키 데이터베이스에 키를 내보내십시오.

1. 키 관리 유틸리티를 시작하십시오.
2. 기본 메뉴에서 키 데이터베이스 파일 -> 열기를 누르십시오.
3. 열기 대화 상자에 키 데이터베이스 이름(기본값 **key.kdb**를 승인)을 입력하십시오. 확인을 누르십시오.
4. 암호 프롬프트 대화 상자에서 암호를 입력하고 확인을 누르십시오.
5. 키 데이터베이스 내용 프레임에서 개인 인증을 선택한 다음 레이블의 내보내기가져오기 단추를 누르십시오.
6. 내보내기가져오기 키 창에서 다음을 수행하십시오.
 - 내보내기 키를 선택하십시오.
 - 목표 데이터베이스 유형을 선택하십시오(예: **PKCS12**).
 - 파일 이름을 입력하거나 찾아보기를 눌러 선택하십시오.
 - 정정 위치를 입력하십시오.
7. 확인을 누르십시오.
8. 암호 프롬프트 대화 상자에서, 올바른 암호를 입력하고 한번 더 입력하여 확인한 다음에 확인을 눌러 선택된 키를 다른 키 데이터베이스로 내보내십시오.

키 가져오기

키를 다른 키 데이터베이스에서 가져오려면 다음을 수행하십시오.

1. 키 관리 유틸리티를 시작하십시오.
2. 기본 메뉴에서 키 데이터베이스 파일 -> 열기를 선택하십시오.

3. 열기 대화 상자에 키 데이터베이스 이름(기본값 **key.kdb**를 승인)을 입력하십시오.
확인을 누르십시오.
4. 암호 프롬프트 대화 상자에서 올바른 암호를 입력하고 확인을 누르십시오.
5. 키 데이터베이스 내용 프레임에서 개인 인증을 선택한 다음 레이블의 내보내기/가져오기 단추를 누르십시오.
6. 내보내기/가져오기 키 창에서 다음을 수행하십시오.
 - 가져오기 키를 선택하십시오.
 - 키 데이터베이스 유형을 선택하십시오(예: **PKCS12**).
 - 파일 이름을 입력하거나 찾아보기를 눌러 선택하십시오.
 - 정정 위치를 선택하십시오.
7. 확인을 누르십시오.
8. 암호 프롬프트 대화 상자에서 올바른 암호를 입력하고 확인을 누르십시오.
9. 키 레이블 목록의 선택란에서 올바른 이름을 선택하고 확인을 누르십시오.

인증 기관 나열

키 데이터베이스에 CA(인증 기관) 목록을 표시하려면, 다음을 수행하십시오.

1. 키 관리 유틸리티를 시작하십시오.
2. 기본 메뉴에서 키 데이터베이스 파일 -> 열기를 누르십시오.
3. 열기 대화 상자에 키 데이터베이스 이름(기본값 **key.kdb**를 승인)을 입력하십시오.
확인을 누르십시오.
4. 암호 프롬프트 대화 상자에서 올바른 암호를 입력하고 확인을 누르십시오.
5. 키 데이터베이스 내용 프레임에서 서명자 인증을 선택하십시오.
6. 서명자 인증, 개인 인증 또는 인증 요청을 눌러서, 키 정보 창에 인증 기관 목록을 표시하십시오.

인증 기관 인증 받기

이 프로시저를 통해, 기본적으로 신뢰할 수 있는 인증 기관으로 지정된 인증 기관으로부터 전자 우편으로 인증을 받으십시오(131 페이지의 『인증 기관』 목록 참조). 인증 기관 서명된 인증을 발행하는 인증 기관이 키 데이터베이스에 있는 인증 기관이 아니면, 우선 인증 기관의 인증을 저장하고 인증 기관을 신뢰할 수 있는 인증 기관으로 지정해야 합니다. 이렇게 하면 데이터베이스로 인증 기관 서명된 인증을 받을 수 있습니다. 신뢰할 수 있는 인증 기관이 아닌 인증 기관에서 인증 기관 서명된 인증을 받을 수 없습니다(139 페이지의 『인증 기관 인증 저장』 참조).

인증 기관 서명된 인증을 키 데이터베이스로 수신하려면 다음을 수행하십시오.

1. 키 관리 유틸리티를 시작하십시오.
2. 기본 메뉴에서 키 데이터베이스 파일 -> 열기를 선택하십시오.

- 열기 대화 상자에 키 데이터베이스 이름(기본값 **key.kdb**를 승인)을 입력하십시오. 확인을 누르십시오.
- 암호 프롬프트 대화 상자에서 암호를 입력하고 확인을 누르십시오.
- DB 유형 목록의 파일 이름이 올바른지 확인하십시오.
- 키 데이터베이스 창에서 개인 인증을 선택한 후, 수신을 누르십시오.
- 파일의 인증 수신 대화 상자에서, 인증 파일 이름 텍스트 필드에 base 64-encoded 유효 파일의 이름을 입력하십시오. 확인을 누르십시오.
- 키 관리자 유틸리티를 닫으려면, 기본 메뉴에서 키 데이터베이스 파일 -> 종료를 누르십시오.

인증 기관 인증 저장

신뢰할 수 있는 인증 기관이 서명한 인증만 보안 연결이 이루어지도록 허용됩니다. 신뢰할 수 있는 인증 기관 목록에 인증 기관을 추가하려면, 신뢰할 수 있는 인증으로 획득 및 저장해야 합니다. 데이터베이스에 인증을 받기 전에, 이 프로시저를 수행하여 새 인증 기관에서 인증을 저장하십시오.

- 키 관리 유틸리티를 시작하십시오.
- 기본 메뉴에서 키 데이터베이스 파일 -> 열기를 누르십시오.
- 열기 대화 상자에 키 데이터베이스 이름(기본값 **key.kdb**를 승인)을 입력하십시오. 확인을 누르십시오.
- 암호 프롬프트 대화 상자에서 암호를 입력하고 확인을 누르십시오.
- 키 데이터베이스 내용 프레임에서 서명자 인증을 선택한 다음 레이블의 추가 단추를 누르십시오.
- 파일의 인증 기관 인증 추가 대화 상자에서 base 64-encoded ASCII 데이터 인증 파일 이름을 선택하거나 찾아보기 옵션을 사용하십시오. 확인을 누르십시오.
- 레이블 대화 상자에서 레이블 이름을 입력하고 확인을 누르십시오.
- 선택란을 사용하여 인증서를 신뢰(기본값)로 지정하십시오.

주: "보기/편집" 단추를 사용하여 인증서를 작성한 이후에 선택란을 보십시오. 선택란은 패널에 나열되지만 인증서 추가 중에는 표시되지 않습니다.

키 데이터베이스에 기본 키 표시

기본 키 항목을 다음과 같이 표시하십시오.

- 키 관리 유틸리티를 시작하십시오.
- 기본 메뉴에서 키 데이터베이스 파일 -> 열기를 누르십시오.
- 열기 대화 상자에 키 데이터베이스 이름(기본값 **key.kdb**를 승인)을 입력하십시오. 확인을 누르십시오.
- 암호 프롬프트 대화 상자에서 암호를 입력하고 확인을 누르십시오.

5. 키 데이터베이스 내용 프레임에서 개인 인증을 선택한 다음 인증 기관 인증 레이블 이름을 선택하십시오.
6. 키 정보 창에서 보기/편집을 눌러 인증 기본 키 정보를 표시하십시오.

지원된 암호 스펙

다음 테이블에는 SSL 버전 2 및 3에 사용되는 암호화 알고리즘 및 해시가 나열됩니다.

키 쌍 세대: RSA 512–1024 개인용 키 크기

SSL 버전 2

US 버전	내보내기 버전
RC4 US	RC4 내보내기
RC2 US	RC2 내보내기
DES 56비트	적용할 수 없음
Triple DES US	적용할 수 없음
RC4 내보내기	적용할 수 없음
RC2 내보내기	적용할 수 없음

SSL 버전 3

US 버전	내보내기 버전
Triple DES SHA US	DES SHA 내보내기
DES SHA 내보내기	RC2 MD5 내보내기
RC2 MD5 내보내기	RC4 MD5 내보내기
RC4 SHA US	널 SHA
RC4 MD5 US	널 MD5
RC4 MD5 내보내기	널 널
RC4 SHA 56 비트	적용할 수 없음
DES CBC SHA	적용할 수 없음
널 SHA	적용할 수 없음
널 MD5	적용할 수 없음
널 널	적용할 수 없음

또한 프록시 구성 파일을 직접 편집하여 SSL 스펙을 구성할 수 있습니다. 자세한 내용은 185 페이지의 부록 B 『구성 파일 지시문』의 참조 섹션에서 다음 지시문을 참조하십시오.

- 313 페이지의 『V2CipherSpecs — SSL 버전 2에 지원되는 암호 스펙 나열』
- 313 페이지의 『V3CipherSpecs — SSL 버전 3에 지원되는 암호 스펙 나열』

- 234 페이지의 『FIPSEnable — SSLV3 및 TLS에 대해 FIPS(Federal Information Processing Standard) 승인 암호를 사용 가능하게 함』

Caching Proxy에 대한 128비트 암호화

Caching Proxy의 128비트 암호화 버전만 전달됩니다. 56비트 버전은 더이상 사용할 수 없습니다. 이전 버전을 갱신하는 경우, Caching Proxy를 현재 설치된 128비트 또는 56비트 버전에 직접 설치할 수 있습니다. 이전에 56비트(내보내기) 브라우저를 사용했다면, 프록시에서 128비트 암호화를 사용하기 위해 128비트 브라우저로 업그레이드해야 합니다.

Caching Proxy를 56비트 버전에서 128비트 버전으로 업그레이드한 후, 인증을 암호화하는 데 사용된 키 크기가 1024로 설정되어 있으면 구성을 변경할 필요가 없습니다. 그러나 키 크기가 512로 설정되어 있으면, 프록시의 128비트 암호화를 사용하기 위해 키 크기가 1024인 새 인증을 작성해야 합니다. IBM Key Manager 유틸리티(iKeyman)를 사용하여 새 키를 작성하십시오.

1. 키 관리자를 시작하십시오.
 - Linux 및 UNIX 플랫폼에서는 명령 프롬프트에서 gsk7ikm을 입력하십시오.
 - Windows 시스템에서는 시작 -> 프로그램 -> IBM WebSphere -> Edge Components -> 키 관리 유틸리티 시작을 누르십시오.
2. 기본 메뉴에서 키 데이터베이스 파일 -> 열기를 누르십시오.
3. 열기 대화 상자에 키 데이터베이스 이름을 입력한 다음(또는 기본값을 사용하는 경우, **key.kdb**를 누르십시오), 확인을 누르십시오.
4. 암호 프롬프트 대화 상자를 열려면 암호를 입력하고 확인을 누르십시오.
5. 기본 메뉴에서 작성 -> 새 인증 요청을 누르십시오.
6. 새 키 및 인증 요청 창에서 다음 사항을 지정하십시오.
 - 키 레이블: 데이터베이스에 키와 인증을 식별하는 데 사용되는 이름을 입력하십시오.
 - 키 크기: **1024**를 선택하십시오.
 - 조직 이름: 키와 연관된 조직의 이름을 입력하십시오.
 - 국가: 국가 코드를 입력하십시오. 최소한 두 문자를 지정해야 합니다(예: US).
 - 인증 요청 파일 이름: 요청 파일의 이름을 입력하거나 선택적으로 기본 이름을 사용하십시오.
7. 확인을 누르십시오.

IBM Key Manager 유틸리티에 대한 자세한 설명은 130 페이지의 『키 및 인증 관리』의 내용을 참조하십시오.

제품의 이 버전은 SUSE Linux의 암호화를 지원하지 않습니다.

제 27 장 암호 하드웨어 지원 사용 가능

이는 역방향 프록시 구성에만 적용합니다.

이 프로시저를 수행하여 SSL 핸드셰이크 루틴을 암호 하드웨어 카드에 오프로드되도록 하십시오.

1. 제조 업체의 지침에 따라 암호 하드웨어 카드를 설치하십시오.
2. Caching Proxy에 SSL을 사용 가능하게 하십시오. 자세한 정보는 125 페이지의 제 26 장 『SSL(Secure Sockets Layer)』을 참조하십시오.
3. 수동으로 ibmproxy.conf 구성 파일의 SSLCryptoCard 지시문을 편집하십시오. 이 지시문에 대한 항목이 구성 및 관리 양식에 표시되지 않습니다. 자세한 정보는 SSLCryptoCard 지시문 참조 304 페이지의 『SSLCryptoCard — 설치된 암호 카드 지정』을 참조하십시오.

AIX에서, IBM 4960 PCI 암호화 액셀러레이터 카드를 지원하려면, 271 페이지의 『PKCS11DefaultCert, PKCS11DriverPath, PKCS11TokenPassword — IBM 4960 PCI 암호화 액셀러레이터 카드(AIX 전용) 지원』을 참조하십시오.

제 28 장 Tivoli Access Manager 플러그인 사용

Caching Proxy 플러그인은 Caching Proxy가 인증 및 권한 부여를 위해 Access Manager를 사용할 수 있도록 하는 Tivoli Access Manager(이전에는 Tivoli Policy Director라 함)와 함께 제공됩니다. 이 플러그인을 사용하면 웹 액세스 제어를 위해 Access Manager를 사용하는 엔터프라이즈가 프록시 서버에 대한 별도 권한 부여 설정을 설정하여 작업을 중복하지 않고도 에지 기술을 추가할 수 있습니다.

Tivoli Access Manager에 대한 추가 정보는 <http://www.ibm.com/software/tivoli/products/>의 제품 웹 사이트를 참조하십시오. 소프트웨어 및 하드웨어 요구사항에 대한 정보 및 Access Manager 플러그인의 설치에 대한 정보는 Tivoli Access Manager와 함께 제공되는 문서를 참조하십시오.

주: Tivoli Access Manager 플러그인은 Red Hat Linux에서는 지원되지 않을 수도 있습니다. Linux 플랫폼에서의 현재 지원 정보는 Tivoli사에 문의하십시오.

구성

Caching Proxy에 대한 설정 스크립트는 Access Manager 플러그인과 함께 제공됩니다.

구성 스크립트를 사용하기 전에 수행할 단계

스크립트를 실행하기 전에 다음 사항을 수행하십시오.

- 필요한 모든 소프트웨어를 설치하십시오.
- 프록시 서버는 포트 80을 사용하도록 설정되어야 합니다(이것이 기본값입니다).
- LDAP 및 Access Manager 컴포넌트를 구성하고 Access Manager 플러그인을 구성하는 중에 이들이 실행 중인지 확인하십시오.
- 사용 가능한 Access Manager 관리자 ID 및 LDAP 관리자 이름이 있는지 확인하십시오. 이 값은 프록시 서버를 설정하는 데 필요합니다.

구성 스크립트 사용

설정 스크립트는 **wslconfig.sh**로 명명되고 /opt/pdweb-lite/bin/ 디렉토리에 제공됩니다. 프롬프트가 나타나면 Access Manager 관리자 ID 및 LDAP 관리자 이름을 입력하십시오.

구성 스크립트가 자동으로 다음 단계를 수행합니다.

- Caching Proxy 사용자 ID를 root로, 그룹 ID를 other로 설정합니다.

- noLog 지시문을 *로 설정하며, 이로 인해 Caching Proxy 액세스 로그에 항목이 기록되지 않습니다.
- 다음의 정보가 있는 ServerInit 지시문을 작성합니다.

```
ServerInit /opt/pdweb-lite/lib/wesauth.so:WTESeal_Init /opt/pdweb-lite/etc/ibmwesas.conf
```

- 다음의 정보가 있는 PreExit 지시문을 작성합니다.

```
PreExit /opt/pdweb-lite/lib/wesauth.so:WTESeal_PreExit
```

- 다음의 정보가 있는 Authorization 지시문을 작성합니다.

```
Authorization * /opt/pdweb-lite/lib/wesauth.so:WTESeal_Authorize
```

- 다음의 정보가 있는 ServerTerm 지시문을 작성합니다.

```
ServerTerm /opt/pdweb-lite/lib/wesauth.so:WTESeal_Term
```

다음과 같이 모든 요청을 Access Manager 인증 프로세스로 전달하는 보호문 및 보호 설정을 작성합니다.

```
Protection PROXY-PROT {
    ServerId WebSEAL-Lite
    Mask      All@(*)
    AuthType Basic
}
Protect * PROXY-PROT
```

Caching Proxy 및 Access Manager 플러그인 시작

프록시 서버 및 Access Manager 플러그인을 구성한 이후에는 **wslstartwte** 명령을 **ibmproxy start** 대신에 사용하여 프록시 서버를 시작하십시오. **wslstartwte** 명령은 초기화를 위해 Access Manager 플러그인이 필요로 하는 환경 변수를 자동으로 로드합니다. 프록시 서버 시작 중에 **wslstartwte**를 사용하지 않는 경우에는 Access Manager 플러그인에 대한 오류 메시지가 표시됩니다. 플러그인을 사용할 경우, 해당 정지 명령 **ibmproxy stop**은 여전히 유효합니다.

제 29 장 PAC-LDAP 권한 부여 모듈 사용

개요

PAC-LDAP 권한 부여 모듈은 권한 부여 및 인증 루틴을 수행할 때 Caching Proxy가 LDAP(Lightweight Directory Access Protocol) 서버에 액세스할 수 있도록 합니다. 모듈은 두 개의 컴포넌트 세트 즉, Caching Proxy API에 LDAP 기능을 추가하는 한 쌍의 공유 라이브러리 및 PAC(Policy Authentication Control) 디먼으로 이루어집니다. `ibmproxy.conf` 파일의 `ServerInit` 지시문은 공유 라이브러리에 Caching Proxy 시작 시 하나 이상의 PAC 디먼을 초기화하도록 지시합니다. 공유 라이브러리는 `paccp.conf` 파일을 읽고 PAC 디먼의 수와 특성을 판별합니다. 초기설정 중 디먼은 `pac.conf` 파일에서 구성 지시문을 참조하고 `pacpolicy.conf`에서 정책 정보를 참조합니다. 그리고 `ibmproxy.conf` 파일의 `Authentication` 지시문은 인증이 필요할 때마다 프록시 서버에게 공유 라이브러리를 호출하도록 지시하거나 `Authorization` 지시문이 표준 HTTP 요청 처리 중 Caching Proxy 작업 흐름을 권한없이 사용합니다.

인증

인증 프로세스는 자격사항의 제공된 세트(사용자 이름 및 암호)가 올바른지 판별합니다. 이 프로세스에는 사용자가 레지스트리에 있는지에 대한 검증 및 제공된 암호가 레지스트리에 저장된 암호와 일치하는지에 대한 검증이 포함됩니다. 이는 인증 단계 중에 PAC-LDAP 모듈을 사용하여 수행되는 조치입니다.

PAC-LDAP 권한 부여 모듈이 인증에 대해 사용 가능하면 이 모듈이 사용자 ID, 암호 및 그룹을 검색할 기본 저장 장소가 됩니다. HTTP 요청이 Caching Proxy 작업 흐름을 통해 전달되면, 각 `Protect` 지시문이 요청된 URL을 요청 템플릿과 비교합니다. 일치사항이 있으면 `Protect` 지시문이 서버 ID, 사용할 인증 유형, 요청 중인 클라이언트에 적용할 마스크 규칙 및 암호와 그룹 파일의 위치를 포함하는 보호 스키마를 호출합니다. 암호 파일이 정의되지 않은 경우, 사용자 ID 및 암호는 PAC-LDAP 권한 부여 모듈을 통해 검색됩니다. 유형 0, 1, 2, 3 정책은 인증 설계를 정의합니다. 인증이 전달되면 요청이 제공되고 인증이 전달되지 않으면 Caching Proxy가 클라이언트에게 401 오류를 리턴합니다.

권한 부여

권한 부여 프로세스는 사용자가 보호 설정된 자원을 액세스하기 위한 필수 권한이 있는지 여부를 판별합니다. PAC-LDAP 모듈이 사용되는 경우, 권한 부여 프로세스에는 HTTP 요청에 대한 `pacpolicy.conf` 파일에 존재하는 권한 부여 규칙의 적용이 포함됩니다.

PAC-LDAP 권한 부여 모듈이 권한 부여에 대해 사용 가능하면, `pacpolicy.conf` 파일 내에서 권한 부여 규칙이 HTTP 요청에 적용됩니다. HTTP 요청이 Caching Proxy 작업 흐름을 통해 전달되면, 각 Protect 지시문이 요청된 URL을 요청 템플릿과 비교합니다. 일치사항이 있으면, Protect 지시문이 보호 스키마를 호출합니다. 이 경우, 보호 스키마는 PAC-LDAP 권한 부여 모듈에서 권한없이 사용한 권한 부여 루틴입니다. Authorization 지시문은 요청된 URL을 요청 템플릿과 비교하고 일치사항이 있으면, PAC-LDAP 권한 부여 모듈이 호출됩니다. `pacpolicy.conf`에 정의된 유형 4 정책은 다양한 URL 요청에 필요한 인증을 세분합니다.

LDAP(Lightweight Directory Access Protocol)

LDAP는 시스템 자원을 최소로 사용하여 X.500 디렉토리에 대화식 액세스를 제공합니다. IANA는 TCP 포트 389 및 UDP 포트 389를 LDAP에 지정했습니다. 자세한 정보는 LDAP를 정의하는 RFC 1777을 참조하십시오.

지원되는 LDAP 클라이언트의 예는 IBM Tivoli LDAP 클라이언트 및 IBM SecureWay LDAP 클라이언트입니다.

설치

PAC-LDAP 권한 부여 모듈의 모든 컴포넌트는 WebSphere Application Server, 버전 6.1의 Caching Proxy 시스템을 설치할 때 자동으로 설치됩니다. Linux 및 UNIX 시스템의 경우, Caching Proxy 라이브러리(`./lib/`) 디렉토리, PAC-LDAP 권한 부여 모듈 라이브러리(`./lib/plugins/pac/`) 디렉토리, 2진(`./bin/`) 디렉토리 및 구성(`./etc/`) 디렉토리는 `/opt/ibm/edge/cp/` 디렉토리 내에 작성됩니다. 그런 다음, `/usr/lib/`, `/usr/sbin/` 및 `/etc` 디렉토리에서 제품 고유의 디렉토리로 기호 링크가 작성됩니다.

디렉토리 구조

Linux 및 UNIX 디렉토리	Windows 디렉토리	내용
<code>/opt/ibm/edge/cp/</code>	<code>\Program Files\IBM\edge\cp\</code>	Caching Proxy 기본 디렉토리(<code>cp_root</code>)
<code>cp_root/sbin/</code>	<code>\Program Files\IBM\edge\cp\Bin\</code>	Caching Proxy 2진 및 스크립트
<code>/usr/sbin/</code>		<code>cp_root/sbin/</code> 에 대한 기호 링크
<code>cp_root/etc/</code>	<code>\Program Files\IBM\edge\cp\etc\</code>	Caching Proxy 구성 파일
<code>/etc/</code>		<code>cp_root/etc/</code> 에 대한 기호 링크
<code>cp_root/lib/</code>	<code>\Program Files\IBM\edge\cp\lib\plugins\</code>	Caching Proxy 라이브러리
<code>cp_root/lib/ plugins/pac/</code>	<code>\Program Files\IBM\edge\cp\lib\plugins\pac\</code>	PAC-LDAP 권한 부여 모듈 라이브러리

Linux 및 UNIX 디렉토리	Windows 디렉토리	내용
/usr/lib/		cp_root/lib/ 및 cp_root/lib/plugins/pac/에 대한 기호 링크
cp_root/server_root/pac/data/	\Program Files\IBM\edge\cp\server_root\pac\data\	PAC-LDAP 권한 부여 모듈 데이터 저장 영역
cp_root/server_root/ pac/creds/	\Program Files\IBM\edge\cp\server_root\pac\creds\	PAC-LDAP 권한 부여 모듈 자격사항

LDAP 플러그인 파일

Linux 및 UNIX 파일 이름	Windows 파일 이름	설명
libpacwte.so	pacwte.dll	공유된 라이브러리
libpacman.so	pacman.dll	공유된 라이브러리
pacd_restart.sh	pacd_restart.bat	PAC 디먼 재시작 스크립트
paccp.conf, pac.conf, pacpolicy.conf	paccp.conf, pac.conf, pacpolicy.conf	구성 및 정책 파일

보안 PACD-LDAP 서버 연결의 추가 요구사항 및 제한사항

LDAP 패키지에서 필요한 GSKit

PACD 디먼 및 LDAP 서버 간의 SSL(Secure Socket Layer) 접속을 사용 가능하게 하려면, LDAP 클라이언트 패키지에 필요한 GSKit 패키지를 설치해야 합니다. GSKit 7은 Caching Proxy 시스템에서 기본적으로 필요하므로 제공됩니다. 그러나, 시스템의 LDAP 클라이언트에서 필요한 버전이 아닐 수도 있습니다. 다른 프로세스의 동일 시스템에서 다른 GSKit 버전을 사용할 수 있습니다.

GSKit key 파일을 \$pacd_creds_dir/pac_keyring.kdb에 위치시키고, 암호를 \$pacd_creds_dir/pac_keyring.pwd에 위치시키십시오.

주: LDAP 서버에 대한 GSKit 요구조건 정보는 다음 웹 사이트의 ITDS(IBM Tivoli Directory Server) 문서를 참조하십시오.

<http://www.ibm.com/software/tivoli/products/directory-server/>

Linux 시스템에는 LD_PRELOAD 환경 변수를 설정해야 합니다.

Linux 시스템의 경우에는 PACD 디먼과 LDAP 서버 간에 SSL 연결을 사용 가능하게 하기 위해, 다음과 같이 LD_PRELOAD 환경 변수를 구성해야 합니다. 다음 변수 값을 설정하십시오.

LD_PRELOAD=/usr/lib/libstdc++-libc6.1-1.so.2

이 절에서 이전에 참조한 GSKit 요구조건은 Linux 시스템에도 적용됩니다.

Linux 시스템에서 IBM Tivoli Directory Server (ITDS) 6.0 LDAP 클라이언트를 사용하는 경우, PACD 프로세스의 시작 실패

Red Hat Enterprise Linux 4.0 시스템에서, Caching Proxy가 인증에 대해 ITDS 6.0 LDAP 플러그인을 사용하도록 구성된 경우 PACD 프로세스가 시작되지 않습니다. 다음은 오류 메시지 결과입니다.

```
"error while loading shared libraries:
/usr/lib/libldapiconv.so: R_PPC_REL24 relocation at 0x0fb58ad0
for symbol 'strpbrk' out of range"
```

현재 제한에서는 ITDS 6.0이 RHEL 4.0 시스템을 지원하지 않습니다.

AIX 시스템에서 IBM Tivoli Directory Server(ITDS) LDAP 클라이언트를 사용하는 경우, PAC-LDAP 모듈은 로드할 수 없음

ITDS LDAP 클라이언트를 사용하는 경우, AIX 시스템에서는 분석되지 않는 링크로 인해 PACD 프로세스가 시작되지 않습니다. PACD 프로세스를 시작할 때, 다음 오류가 발생할 수 있습니다.

```
exec(): 0509-036 Cannot load program /usr/sbin/pacd
because of the following errors:
0509-022 Cannot load module /usr/lib/libpacman.a.
0509-150 Dependent module libldap.a could not be loaded.
0509-022 Cannot load module libldap.a.
```

LDAP 클라이언트의 ITDS 버전 5에 대한 이 문제점을 해결하려면, 다음 기호를 작성하십시오.

```
ln -s /usr/lib/libibmldap.a /usr/lib/libldap.a
```

LDAP 클라이언트의 ITDS 버전 6에 대한 이 문제점을 해결하려면, 다음 기호를 작성하십시오.

```
ln -s /opt/IBM/ldap/V6.0/lib/libibmldap.a /usr/lib/libldap.a
```

ibmproxy.conf 파일을 편집하여 PAC-LDAP 권한 부여 모듈을 사용 가능하게 하기

PAC-LDAP 권한 부여 모듈을 초기화하려면 ServerInit, Authorization 또는 Authentication의 세 지시문과 ServerTerm을 ibmproxy.conf 파일의 API 지시문 섹션에 추가해야 합니다. 이 지시문을 작성하려면, ibmproxy.conf 파일을 수동으로 편집하거나 프록시 서버가 이미 실행 중인 경우에는 인터넷 브라우저로 구성 및 관리 양식에 연결한 다음 API 요청 처리 양식을 여십시오(서버 구성 -> 요청 처리 -> API 요청 처리 클릭). 이 섹션에 제공된 예제에서 명확성을 위해 행 구분이 포함되었는지 여부에 관계 없이 각 지시문은 프록시 구성 파일에 한 행으로 표시되어야 합니다.

표준 지시문(설명 양식)이 `ibmproxy.conf` 파일의 API 섹션에 제공됩니다. API 지시문은 중요도 순으로 나열되어 있습니다. API 지시문을 추가하여 새로운 기능 및 플러그인 모듈을 사용할 수 있을 때, 구성 파일의 표준 섹션에 표시된 대로 지시문을 나열하십시오. 또는 필요한 경우, 원하는 각 기능이나 플러그인에 대한 지원을 포함할 API 지시문의 설명을 지우거나 편집하십시오.

`ServerInit` 지시문에는 다음의 세 가지 인수 즉, (1)공유 라이브러리의 전체 경로, (2)함수 호출, (3)`paccp.conf`의 전체 경로가 있습니다. 첫 번째 및 두 번째 인수는 콜론(:)으로 구분됩니다. 두 번째 및 세 번째 인수는 공백으로 구분됩니다. 첫 번째 및 세 번째 인수는 시스템에 고유하며 플러그인 컴포넌트가 설치된 위치에 따라 다릅니다. 두 번째 인수는 공유 라이브러리에 하드 코드되며, 표시된 대로 정확히 입력해야 합니다. API 요청 처리 양식을 사용하여 `ServerInit` 지시문을 작성할 때, 두 번째와 세 번째 인수를 모두 함수 이름 필드에 입력해야 합니다. 세 번째 인수는 **IP** 템플릿 컬럼에 표시됩니다.

`Authorization` 지시문에는 다음의 세 가지 인수 즉, (1)요청 템플릿, (2)공유 라이브러리의 전체 경로, (3)함수 이름이 있습니다. HTTP 요청은 요청 템플릿과 비교되어 응용프로그램 함수가 호출되는지 여부를 판별합니다. 요청 템플릿에는 프로토콜, 도메인 및 호스트가 포함될 수 있고, 앞에 슬래시(/)가 붙을 수 있으며, 별표(*)를 와일드 카드로 사용할 수 있습니다. 예를 들어, `/front_page.html`, `http://www.ics.raleigh.ibm.com`, `/pub*`, `/*` 및 `*`는 모두 유효합니다. 함수 이름은 프로그램 내에서 응용프로그램 함수에 제공된 이름입니다. 이것은 하드 코드되며 표시된 대로 정확히 입력해야 합니다. 처음의 두 인수는 공백으로 구분됩니다. 마지막 두 인수는 콜론(:)으로 구분됩니다.

`Authentication` 지시문에는 다음의 두 가지 인수 즉, (1)공유 라이브러리의 전체 경로, (2)함수 이름이 있습니다. 두 인수는 콜론(:)으로 구분됩니다. 첫 번째 인수는 시스템에 고유하며 공유 라이브러리가 설치되는 위치에 따라 다릅니다. `Caching Proxy`를 역방향 프록시로 사용하는 경우, 첫 인수에 대한 URL 템플릿은 문서 루트(/)에서 시작해야 합니다. 두 번째 인수는 공유 라이브러리에 하드 코드되며, 표시된 대로 정확히 입력해야 합니다.

`ServerTerm` 지시문에는 다음의 두 가지 인수 즉, (1)공유 라이브러리의 전체 경로, (2)함수 이름이 있습니다. 두 인수는 콜론(:)으로 구분됩니다. 첫 번째 인수는 시스템에 고유하며 공유 라이브러리가 설치되는 위치에 따라 다릅니다. 두 번째 인수는 공유 라이브러리에 하드 코드되며, 표시된 대로 정확히 입력해야 합니다. 이 지시문은 프록시 서버가 종료될 때 PAC 디먼을 종료합니다. 디먼의 소유자가 프록시 서버의 소유자와 다른 경우, 프록시 서버가 디먼을 정지시키지 못할 수 있습니다. 이 경우, 관리자가 수동으로 디먼을 정지시켜야 합니다.

`ServerInit path_of_shared_library:pacwte_auth_init path_of_conf_policy_file`

Linux 및 UNIX 예제:

```
ServerInit /usr/lib/libpacwte.so:pacwte_auth_init /etc/pac.conf
```

Windows 예제:

```
ServerInit C:\Progra ~1\IBM\edge\cp\lib\plugins\  
pac\pacwte.dll:pacwte_auth_init C:\Progra ~1\IBM\edge\cp  
Authorization request-template path_of_shared_library:pacwte_auth_policy
```

Linux 및 UNIX 예제:

```
Authorization http://* /usr/lib/libpacwte.so:pacwte_auth_policy
```

Windows 예제:

```
Authorization http://* C:\Program Files\IBM\edge\cp\lib\plugins\  
pac\pacwte.dll:pacwte_auth_policy  
Authentication BASIC path_of_shared_library:pacwte_auth_policy
```

Linux 및 UNIX 예제:

```
Authentication BASIC /usr/lib/plugins/pac/libpacwte.so:pacwte_auth_policy
```

Windows 예제:

```
Authentication BASIC C:\Program Files\IBM\edge\cp\lib\plugins\  
pac\pacwte.dll:pacwte_auth_policy  
ServerTerm path_of_shared_library:pacwte_shutdown
```

Linux 및 UNIX 예제:

```
ServerTerm /usr/lib/libpacwte.so:pacwte_shutdown
```

Windows 예제:

```
ServerTerm BASIC C:\Program Files\IBM\edge\cp\lib\plugins\  
pac\bin\pacwte.dll:pacwte_shutdown
```

PAC-LDAP 권한 부여 모듈 구성 파일 편집

PAC-LDAP 권한 부여 모듈 구성 파일 및 정책 파일은 텍스트 편집기를 사용하여 직접 편집해야 합니다. 지시문 이름은 첫 번째 인수와 콜론(:)으로 구분됩니다. 여러 인수는 쉼표(,)로 구분됩니다. 구성 및 정책 파일에 있는 설명이 편집하는 데 도움이 됩니다. 키 정책 지시문이 아래 표시됩니다.

paccp.conf

Caching Proxy의 초기설정 중 공유 라이브러리가 `paccp.conf`를 읽으며, 이 파일에는 시작할 각 PAC 디먼의 정의(`[PAC_MAN_SERVER]` 스탠자)가 포함됩니다. 각 PAC 디먼에는 자체 `[PAC_MAN_SERVER]` 스탠자가 있어야 합니다.

```
[PAC_MAN_SERVER]  
hostname:                # name of PAC daemon  
port:                    # port pacd is listening on
```

```
[PACWTE_PLUGIN]
hostname_check:[true|false] # enables DNS lookup. Must have
                             # DNS lookup turned on for ibmproxy to work.
```

pac.conf

pac.conf 파일은 PAC 디먼에서 연결을 시도할 LDAP 서버를 지정합니다.

```
[PAC_MAN_SERVER]
hostname:                # name of PAC daemon
port:                    # port pacd is listening on
conn_type:ssl            # comment out if you do not use SSL
authentication_sequence:[primary|secondary|none]
authorization_sequence:[primary|secondary|none]

[LDAP_SERVER]
hostname:                # LDAP Server hostname
port:389                 # Port LDAP is listening on
ssl_port:636             # SSL port used by the LDAP server
admin_dn:                # User with permission to access the LDAP server
                        # specify admin_dn=NULL to enable anonymous binding
search_base:             # Portion of LDAP tree to search for policy info
                        # If not required, specify search_base=NULL
search_key:              # ID field to search

[CACHE]
cred_cache_enabled [TRUE|FALSE] # turn credentials cache on
cred_cache_min_size:100        # minimum number of credentials to cache in pacd
cred_cache_max_size:64000      # maximum number of credentials to cache in pacd
cred_cache_expiration:86400    # when a credential expires
policy_cache_enabled:[TRUE|FALSE] # turns policy cache on/off
policy_cache_min_size:100      # min. number of policy related items to cache
policy_cache_max_size:64000    # max. number of policy related items to cache
policy_cache_expiration:86400  # when a policy related item expires
```

pacpolicy.conf

모든 LDAP 정책은 구성 및 정책 파일에 다음의 템플리트를 사용합니다. 각 정책은 대괄호 안의 대문자 키워드 POLICY로 시작해야 합니다.

```
[POLICY]
default_policy:[grant|deny] # describes the default policy for users
                             # that are not described in the POLICY section
pac_client_hostname:        # the instances of Caching Proxy that are allowed
                             # to use a policy list
id:                          # the id for the LDAP entry or ip/hostname
                             # (wildcard supported, such as *.ibm.com)
grant:[true|false]          # true means to grant access, false means
                             # to deny access
type:[0|1|2|3|4]            # 0 LDAP entry that is a group,
                             # 1 LDAP entry that is not a group,
                             # 2 IP address
                             # 3 hostname
                             # 4 URL
propagate:[true|false]      # true means that the access rights (grant
                             # or deny) will be propagated to all
                             # descendants or members
stop_entry:[entry|NULL]     # Propagation of the access right stops
                             # at this entry. If the id is a group,
                             # stop_entry must be set to NULL.
```

```

# stop_entry may be applied to an IP
# address or hostname. Each stop_entry
# must be on its own line
exception_entry:[entry|NULL] # Assignment of the access right skips
# these entries, but continues through their
# subtrees. This may be a list of entries.
# exception_entry may be applied to a group,
# IP address, or hostname. Each
# exception_entry must be on its own line.

Exception_type:
Exception:

```

와일드 카드(*)는 id 및 stop_entry 지시문에서 IP 주소의 마지막 위치나 호스트 이름의 첫 번째 위치에만 지원됩니다. exception_entry에는 와일드 카드가 지원되지 않습니다. 와일드 카드는 모든 필드의 모든 LDAP 항목에 지원되지 않습니다.

여러 정책이 지원되며 정책이 충돌하는 경우에는 거짓값이 항상 우선순위가 높습니다. 즉, 정책에서 한 번의 거부가 액세스를 차단합니다. 구성 및 정책 파일에서 정책이 나열된 순서는 관계가 없으며 이것이 우선순위를 설정하지 않습니다.

일련의 정책 예제에 대해서는 구성 파일 디렉토리의 pacpolicy.conf 파일을 참조하십시오.

주: 중첩된 그룹은 상위 그룹의 정책을 계승하지 않습니다. 그룹에서 계승해야 할 유일한 정책은 해당 그룹이 명시적 구성원인 정책입니다.

pac_ldap.cred 작성

/cp_root/server_root/pac/creds에 일반 텍스트 파일(pac_ldap.cred)을 작성하십시오. 이 파일에는 pac.conf 파일에 있는 admin_dn 지시문의 사용자 이름에 해당하는 암호가 있습니다.

주: 익명 바인딩을 사용 가능하게 하려면, pac.conf의 admin_dn 지시문을 admin_dn:NULL로 변경하고 더미 문자열을 pac_ldap.cred 파일에 입력하십시오.

PAC 디먼이 처음 파일을 읽을 때 이 암호를 암호화합니다.

Linux 및 UNIX 플랫폼에서 pac_ldap.cred 파일을 작성하려면 다음 명령을 실행하십시오.

```

cd cp_root/server_root/pac/creds
echo "password" > pac_ldap.cred
chown nobody pac_ldap.cred
chgrp nobody pac_ldap.cred
(on SUSE Linux, use chgrp nogroup pac_ldap.cred.)

```

Windows 플랫폼에서 파일을 작성하려면 텍스트 파일에서 암호를 입력하고 이 파일을 server_root\pac\creds\ 디렉토리에 저장하십시오.

pacd 시작 및 정지

pacd가 처리되면서 LDAP 권한 부여 디먼이 실행됩니다. 제공되는 스크립트를 사용하여 Caching Proxy를 인터럽트하지 않고도 LDAP 권한 부여 디먼을 재시작할 수 있습니다. 다음과 같이 pacd 스크립트를 실행하십시오.

- Linux 및 UNIX 플랫폼의 경우:

```
/usr/sbin/pacd_restart.sh pacd_user_id
```

- Windows 플랫폼의 경우:

```
C:\Program Files\IBM\edge\cp\Bin\pacd_restart.bat CP_install_root
```

주: **stopsrc -ibmproxy** 명령(AIX 시스템의 경우) 또는 **ibmproxy -stop** 명령(HP-UX, Linux 및 Solaris 시스템의 경우)을 사용하여 Caching Proxy 서버를 시스템 종료한 이후에 pacd 프로세스가 계속해서 실행될 수 있습니다. pacd 프로세스는 다음과 같이 **kill** 명령을 사용하여 안전하게 종료할 수 있습니다.

```
kill -15 pacd_process_ID
```

HP-UX의 경우: PAC-LDAP 플러그인 및 pacd는 런타임에 모든 자체 관련 공유 라이브러리를 로드하지 않을 수 있습니다. 이들 라이브러리를 사용하기 전에 시스템 변수가 다음과 같이 설정되어 있는지 확인하십시오.

```
SHLIB_PATH=/usr/lib:/usr/IBMDap/lib  
PATH=/usr/IBMDap/bin:$PATH  
PATH=/usr/IBMDap/bin
```

/usr/IBMDap/는 HP-UX에서 LDAP 클라이언트에 대한 기본 설치 경로입니다. LDAP 클라이언트가 다른 위치에 설치되어 있으면 PATH 및 SHLIB_PATH를 이에 맞게 조정하십시오. 이 변수를 설정하지 않으면 다음 오류가 발생할 수 있습니다.

- PAC-LDAP 플러그인을 사용 가능하게 한 이후에 다음 메시지가 오류 로그에 나타날 수 있습니다.

```
"Serverinit 오류: 서버가 함수를 DLL 모듈  
/opt/ibm/edge/cp/lib/plugins/pac/libpacwte.sl에서 로드하지 않음"
```

- /usr/sbin/pacd를 시작하려고 시도하는 경우, 다음 연결 오류가 표시됩니다.

```
"/usr/lib/dld.sl: 공유 라이브러리에 대한 경로를 찾을 수 없음: libibmdap.sl  
/usr/lib/dld.sl: 해당 파일 또는 디렉토리가 없음  
중단"
```

Linux의 경우: SUSE Linux Enterprise Server 9에 대해, ldd pacd는 libldap.so를 찾을 수 없음을 보고할 수 있습니다. 이 문제점을 해결하려면, 다음 기호를 작성하십시오.

```
ln -s /usr/lib/libldap.so.19 /usr/lib/libldap.so
```

AIX의 경우: IBM Tivoli Directory Server 5.2에서 pacd를 시작하는 경우, 다음 오류가 나타나면서 PAC-LDAP 모듈을 로드할 수 없을 수 있습니다.

```
exec(): 0509-036 Cannot load program /usr/sbin/pacd because of the following errors:  
0509-022 Cannot load module /usr/lib/libpacman.a.  
0509-150 Dependent module libldap.a could not be loaded.  
0509-022 Cannot load module libldap.a.
```

이 문제점을 해결하려면, 다음 기호를 작성하십시오.

```
ln -s /usr/lib/libibmldap.a /usr/lib/libldap.a
```

주: Caching Proxy를 구성하여 LDAP 인증을 사용하면, 다음 오류가 표시됩니다.

```
Could not extract a value for: Uid, return code:3
```

이 오류는 LDAP 인증이 올바르게 기능하는 경우에도 표시되므로, 무시할 수 있습니다.

제 6 부 Caching Proxy 모니터링

이 파트에서는 로그 및 서버 활동 모니터를 사용하여 Caching Proxy를 모니터링하는 데 필요한 명령을 제공합니다.

이 파트에는 다음과 같은 장이 들어 있습니다.

159 페이지의 제 30 장 『로그 구성』

167 페이지의 제 31 장 『서버 활동 모니터 사용』

제 30 장 로그 구성

로그를 사용자 정의하려면 구성 및 관리 양식을 사용하거나 프록시 구성 파일의 지시문을 편집할 수 있습니다. 다음 옵션을 설정할 수 있습니다.

- 로그 파일을 저장할 경로 및 파일 이름
- 액세스 로그 파일에 있는 정보를 포함하거나 제외하는 필터
- 로그를 보존하거나 삭제하기 위한 유지보수 옵션

로그 정보

Caching Proxy는 이벤트 로그 및 오류 로그 외 세 가지 유형의 액세스 로그를 작성할 수 있습니다.

- 액세스 로그는 다음과 같습니다.
 - 액세스 로그—Caching Proxy 자체에 대한 로컬 관리 요청을 추적합니다.
 - 캐시 액세스 로그—캐시에 있는 오브젝트에 대한 요청을 추적합니다.
 - 프록시 액세스 로그—기점 서버로부터 프록시된 요청을 추적합니다.
- 이벤트 로그—캐시 정보 메시지를 추적합니다.
- 오류 로그—Caching Proxy와 관련된 오류 메시지를 추적합니다.

Caching Proxy는 매일 자정에 새 로그 파일을 작성합니다. 프록시가 자정에 실행되고 있지 않으면, 그날 서버가 처음 시작할 때 새 로그가 작성됩니다. 각 로그 파일에 대한 디렉토리 및 파일 이름 접두부를 지정할 수 있습니다. 또한 작성된 각 로그 파일에는 `.Mmmddyyyy`(예: `.Apr142000`) 양식의 날짜 접미부가 있습니다.

로그는 공간을 많이 차지하므로, 오류 방지를 위해 운영 체제 및 캐시와 별도의 저장영역 장치에 로그 파일을 저장하는 것도 좋은 방법입니다. 추가적으로 163 페이지의 『로그 유지보수 및 보존』에 지정된 것과 같이 로그 유지보수 루틴을 구성해야 합니다.

로그 파일 이름 및 기본 옵션

프록시 서버 로그의 기본 구성을 지정하려면, 구성 및 관리 양식에서 서버 구성 -> 로그 -> 로그 파일을 선택하십시오. 사용하려는 각 로그 파일의 경로 및 파일 이름을 지정하십시오. 각 로그의 현재 파일 이름은 해당 텍스트 상자에 표시됩니다. 경로를 지정하지 않으면 기본 경로가 표시됩니다.

프록시 로그에 로그되는 정보는 시스템 로그에 자동으로 작성되지 않지만, 자체 로그 대신 또는 추가로 시스템 로그에 작성하도록 Caching Proxy를 구성할 수 있습니다. 로그 파일 양식에서 시스템 로그에 정보 로그 선택란을 선택하십시오. 시스템 로그는 이 옵션을 선택하기 전에 작성해야 합니다.

프록시 서버 로그 정보가 시스템 로그에만 작성되도록 지정하려면 프록시 구성 파일을 편집해야 합니다. 252 페이지의 『LogToSyslog — 액세스 정보를 시스템 로그에 전송할지 여부 지정(Linux 및 UNIX 전용)』에 대한 참조 섹션을 참조하십시오.

관련 구성 파일 지시문

프록시 구성 파일을 사용하여 로그를 설정하려면 다음 지시문에 대한 185 페이지의 부록 B 『구성 파일 지시문』 참조 섹션을 참조하십시오.

- 188 페이지의 『AccessLog — 액세스 로그 파일에 대한 경로 이름 지정』
- 201 페이지의 『CacheAccessLog — 캐시 액세스 로그 파일에 대한 경로 지정』
- 227 페이지의 『ErrorLog — 서버 오류를 로그할 파일 지정』
- 229 페이지의 『EventLog — 이벤트 로그 파일에 대한 경로 지정』
- 252 페이지의 『LogToSyslog — 액세스 정보를 시스템 로그에 전송할지 여부 지정 (Linux 및 UNIX 전용)』
- 257 페이지의 『MaxLogFileSize — 각 로그 파일의 최대 크기 지정』
- 284 페이지의 『ProxyAccessLog — 프록시 액세스 로그 파일에 대한 경로 이름 지정』

액세스 로그 필터

액세스 로그는 호스트 시스템, 프록시, 캐시의 활동을 기록합니다. 프록시가 수신하는 각 액세스 요청에 대한 적절한 액세스 로그의 항목에 다음 정보가 포함됩니다.

- 요청된 콘텐츠
- 요청된 시기
- 요청자
- 요청 방법
- 서버가 요청에 대한 응답으로 전송한 파일 유형
- 요청 성공 여부를 표시하는 리턴 코드
- 전송된 데이터 크기

액세스 오류는 서버의 오류 로그에 기록됩니다.

로그된 콘텐츠를 제어하려는 이유

로그된 콘텐츠를 제한하는 몇 가지 이유가 있습니다.

- 로그 크기를 줄일 수 있습니다.

사용 중인 서버의 로그 파일은 서버의 디스크 공간을 모두 채울만큼 커질 수 있습니다. 기본적으로 모든 액세스 요청은 로그됩니다. 그러면, HTML 페이지뿐만 아니라 이 페이지에 포함된 각 이미지에 대해서도 로그 입력 항목이 생성됩니다. 중요한 액세스 요청만 포함한다면 로그의 항목 수를 크게 줄일 수 있습니다. 예를 들어, GIF 이미지 요청을 제외한 HTML 페이지 액세스 요청에 대한 로그 항목을 포함하도록 액세스 로그를 구성할 수 있습니다.

- 목표 정보를 수집할 수 있습니다.

예를 들어, 회사 외부에서 서버에 액세스하고 있는 사용자가 누구인지 알고 싶으면, 회사 내의 IP 주소에서 발생한 액세스 요청을 필터할 수 있습니다. 특정 웹 사이트 방문자 수를 알고 싶은 경우에는 해당 URL의 액세스 요청만을 표시하는 액세스 로그를 작성할 수 있습니다.

액세스 로그에서 제외된 정보는 액세스 보고서에 기록되지 않으며, 이 정보는 나중에 사용할 수 없습니다. 따라서 추적해야 할 액세스 정보량을 정확히 알지 못하는 경우에는, 서버 모니터링에 대한 경험이 충분해질 때까지 제외 필터를 신중히 적용하십시오.

액세스 로그 필터 구성

액세스 로그 입력 항목은 다음 속성에 따라 필터할 수 있습니다.

- URL(파일 또는 디렉토리)
- IP 주소 또는 호스트 이름
- 사용자 에이전트
- 방법
- MIME 유형
- 리턴 코드

필터를 지정하려면, 구성 및 관리 양식에서 서버 구성 -> 로그 -> 액세스 로그 제외를 선택하십시오. 원하는 제외만을 지정하십시오. 모든 카테고리를 사용할 필요는 없습니다.

- 다음 디렉토리 또는 파일에 대한 요청을 액세스 로그에 로그하지 마십시오 섹션에 로그 항목을 제외하려는 URL 경로를 나열하십시오.
- 다음 사용자-에이전트 요청을 로그하지 마십시오 섹션에, 로그 항목을 제외하려는 프록시 에이전트를 나열하십시오.
- 다음 호스트 이름 또는 IP 주소 요청을 로그하지 마십시오 섹션에, 로그 항목을 제외하려는 호스트 이름 또는 IP 주소를 나열하십시오.

- 다음 메소드가 들어 있는 요청을 로그하지 마십시오 섹션에서, 로그 항목을 제외하려는 메소드 상자를 확인하십시오.
- 다음 MIME 유형의 파일에 대한 요청을 로그하지 마십시오 섹션에서, 로그 항목을 제외하려는 MIME 유형의 상자를 확인하십시오.

주: 이 지시문은 프록시 액세스 로그에만 영향을 미칩니다. MIME 유형별로 캐시된 오브젝트를 나열하는 로그는 필터할 수 없습니다. 이를 수행하려면 AccessLogExcludeURL을 사용하십시오.

- 다음 리턴 코드가 들어 있는 요청을 로그하지 마십시오 섹션에서, 로그 항목을 제외하려는 요청 리턴 코드의 상자를 확인하십시오.

제출을 누르십시오.

관련 구성 파일 지시문

프록시 구성 파일을 사용하여 액세스 로그 필터를 설정하려면, 다음 지시문에 대한 185 페이지의 부록 B 『구성 파일 지시문』 참조 섹션을 참조하십시오.

- 189 페이지의 『AccessLogExcludeMethod — 지정된 메소드에서 요청한 파일 또는 디렉토리에 대한 로그 항목 억제』
- 189 페이지의 『AccessLogExcludeMimeType — 특정 MIME 유형에 대한 프록시 액세스 로그 입력 억제』
- 190 페이지의 『AccessLogExcludeReturnCode — 고유한 리턴 코드에 대한 로그 입력 항목 억제』
- 190 페이지의 『AccessLogExcludeURL — 고유한 파일 또는 디렉토리에 대한 로그 입력 항목 억제』
- 191 페이지의 『AccessLogExcludeUserAgent — 고유한 브라우저에서 로그 입력 항목 억제』
- 263 페이지의 『NoLog — 템플릿과 일치하는 고유한 호스트나 도메인에 대한 로그 입력 항목 압축』

기본 로그 설정

- 기본 경로

모든 로그는 Caching Proxy 기본 구성에서 사용 가능합니다. 모든 로그는 설치 디렉토리인 logs/ 하위 디렉토리에 저장됩니다. 기본 경로는 다음과 같습니다.

- 로컬(관리) 액세스 로그의 경우
 - Linux 및 UNIX: /opt/ibm/edge/cp/server_root/logs/local
 - Windows: drive:\Program Files\IBM\edge\cp\logs\local
- 캐시 액세스 로그의 경우

- Linux 및 UNIX: /opt/ibm/edge/cp/server_root/logs/cache
- Windows: *drive:\Program Files\IBM\edge\cp\logs\cache*
- 프록시 액세스 로그의 경우
 - Linux 및 UNIX: /opt/ibm/edge/cp/server_root/logs/proxy
 - Windows: *drive:\Program Files\IBM\edge\cp\logs\proxy*
- 오류 로그의 경우
 - Linux 및 UNIX: /opt/ibm/edge/cp/server_root/logs/error
 - Windows: *drive:\Program Files\IBM\edge\cp\logs\error*
- 이벤트 로그의 경우
 - Linux 및 UNIX: /opt/ibm/edge/cp/server_root/logs/event
 - Windows: *drive:\Program Files\IBM\edge\cp\logs\event*

각 로그 파일 이름은 기본 이름과 *.Mmmddyyyy* 양식으로 된 날짜 접미부의 결합입니다(예: proxy.Feb292000).

• 기본 형식

로그는 기본적으로 일반 파일 형식으로 저장됩니다. 결합 로그 형식을 사용할 수도 있으며, 다음 행을 프록시 구성 파일(ibmproxy.conf)에 추가하여 설정할 수 있습니다.

```
LogFileFormat combined
```

결합 로그 형식은 일반 형식과 유사하지만, 참조자, 사용자 에이전트 및 쿠키 정보를 표시하는 필드가 추가로 존재합니다. 로컬 시간 형식이 기본 시간 형식입니다.

• 기본 콘텐츠

기본적으로 모든 액세스 요청은 적절한 액세스 로그에 기록되며 액세스 정보는 시스템 로그에 기록되지 않습니다. 오류 로그 정보는 오류 로그에만 기록되고 이벤트 로그 정보는 이벤트 로그에만 기록됩니다.

• 기본 유지보수

기본 구성에서 로그는 보존되거나 삭제되지 않습니다.

로그 유지보수 및 보존

Caching Proxy는 플러그인을 사용하여 로그를 관리합니다. 260 페이지의 『Midnight — 로그 보존에 사용되는 API 플러그인 지정』 구성 파일 지시문에 대한 자세한 정보는 185 페이지의 부록 B 『구성 파일 지시문』 참조 페이지를 참조하십시오.

매일 기록된 로그를 보존하거나 제거하는 방법을 지정하십시오. 기본 옵션은 다음과 같습니다.

- 지정된 유효 기간이 지난 로그 압축 및 제거
- 특정 유효 기간이 지나거나 해당 로그 카테고리가 특정 크기에 이른 후에 로그 제거
- 밤마다 자정에 프로그램을 실행하여 로그 유지 및 보존

기본적으로 현재 및 이전 로그는 유지보수 에이전트에 의해 삭제되지 않습니다. 모든 현재 로그 및 이전 캐시 액세스 로그는 유지보수 에이전트로 압축할 수 없습니다.

로그 유지보수를 구성하려면, 구성 및 관리 양식에서 서버 구성 -> 로그 -> 로그 보존을 선택하십시오. 이 양식에서 드롭다운 상자를 사용하여 유지보수 메소드를 지정하십시오.

- 제거를 선택했으면, 삭제할 로그를 판별하는 데 사용할 유효 기간이나 파일 크기 또는 이 두 가지 모두를 설정하십시오. 유효 기간 및 크기로 파일을 제거할 경우, 최대 유효 기간이 지난 파일이 최대 크기를 넘어선 파일보다 먼저 제거됩니다. 파일을 크기로 별로 제거하면 오래된 로그가 먼저 삭제됩니다.
- 압축을 선택하면 압축할 로그의 유효 기간 및 로그 파일(모든 매개변수를 포함하여)을 압축하는 데 사용할 명령을 설정하십시오. 또한 로그의 최대 유효 기간도 설정하십시오. 유지보수 에이전트는 로그를 압축한 다음, 최대 유효 기간이 지난 압축 로그를 삭제합니다.

관련 구성 파일 지시문

프록시 구성 파일을 사용하여 로그 보존을 설정하려면, 다음 지시문에 대한 185 페이지의 부록 B 『구성 파일 지시문』 참조 섹션을 참조하십시오.

- 213 페이지의 『CompressAge — 로그 압축 시기 지정』
- 214 페이지의 『CompressDeleteAge — 로그 삭제 시기 지정』
- 213 페이지의 『CompressCommand — 압축 명령 및 매개변수 지정』
- 251 페이지의 『LogArchive — 로그 보존 작동 지정』
- 260 페이지의 『Midnight — 로그 보존에 사용되는 API 플러그인 지정』
- 289 페이지의 『PurgeAge — 로그의 유효 기간 한계 지정』
- 289 페이지의 『PurgeSize — 로그 보존 크기의 한계 지정』.

로그 파일 시나리오

다음의 예제는 사용자의 필요에 따라 로그를 사용자 정의할 수 있는 방법을 보여줍니다. 방금 Caching Proxy를 구입하여 설치했습니다. 다음과 같은 요구사항의 액세스 및 오류 정보를 로그하도록 서버를 설정하려 합니다.

- 로그에는 로컬 시간 소인이 제공되며, 공통 로그 파일 형식이어야 합니다.
- 액세스 로그는 30일이 지나거나 로그 전체 크기가 25MB에 이르면 폐기됩니다.
- 다음의 요청 유형은 액세스 로그에 로그되지 않습니다.

- GIF 이미지에 대한 요청
- 130.128.*.*에 일치하는 IP 주소가 있는 호스트의 요청
- 경로 재지정 요청(이러한 요청에서 300과 399 사이의 리턴 코드가 발생함)

이 기준에 따라 로그를 보존하도록 Caching Proxy를 설정하려면, 구성 및 관리 양식에서 서버 선택 -> 로깅을 선택하십시오.

1. 선택적으로 로그 파일 양식을 선택하여 액세스 로그 파일의 경로를 지정하십시오 (디폴트 경로가 제공됩니다).
2. 로그 보존 양식을 사용하여, 파일을 보존할 방법을 지정하십시오.
 - 보존 방법으로 제거를 지정하십시오.
 - 제거 사용 시에서 다음과 같이 필드에 기입하십시오.
 - 30일이 지난 로그 삭제
 - 25MB보다 큰 로그 삭제
3. 액세스 로그 제외 양식을 사용하여 다음과 같이 로그 입력 항목을 필터하십시오.
 - 다음 호스트 이름 또는 IP 주소의 요청을 로그하지 마십시오 목록에서 제외 호스트 필드에 130.128.*.*를 추가하십시오.
 - 다음 MIME 유형의 파일에 대한 요청을 로그하지 마십시오에서 image/gif 선택란을 선택하십시오.
 - 다음 리턴 코드가 들어 있는 요청을 로그하지 마십시오에서 (3xx) 경로 재지정 선택란을 선택하십시오.

위의 지시대로 따르면 프록시 구성 파일에 다음 행이 생성됩니다.

```
LogArchive purge
PurgeAge 30
PurgeSize 25
AccessLogExcludeURL *.gif
NoLog 130.128.*.*
AccessLogExcludeReturnCode 300
```


제 31 장 서버 활동 모니터 사용

Caching Proxy의 서버 활동 모니터는 서버 및 네트워크 성능 통계, 서버 및 네트워크 상태, 액세스 로그 입력 항목을 표시합니다. 모니터는 원격으로 사용할 수 있으며, 실행 프록시 서버에 함께 위치해 있지 않아도 됩니다. 서버 활동 모니터는 기본적으로 사용 가능하며 구성이 필요없습니다.

서버 활동 모니터를 여는 경로는 두 가지가 있습니다.

- 연결된 웹 브라우저에서, 서버의 전체 이름이 지시되어 있으면 이 이름을 사용하여 다음 URL을 입력하십시오.

`http://your.server.name/Usage/Initial`

- 구성 및 관리 양식에서 서버 활동 모니터를 선택하십시오.

구성 클라이언트의 다른 양식과 달리, 이 카테고리의 양식은 서버에 대한 구성을 설정하지 않지만 서버 사용에 관한 데이터를 표시합니다. 이 양식은 단일 콘솔 창에 표시될 수 있는 것보다 훨씬 많은 정보를 제공합니다.

다음 섹션에서는 서버 활동 모니터가 제공하는 정보의 유형을 표시하며 성능 조정을 위한 정보 사용 방법을 제안합니다.

사용 가능한 몇 가지 서버 활동 모니터 페이지는 다음과 같습니다.

- 활동 통계
- 네트워크 통계
- 액세스 통계
- 프록시 액세스 통계
- 캐시 통계
- 캐시 새로 고침 요약

각 페이지에는 새로 고침 단추가 있으며 이 단추를 사용하여 정보를 갱신할 수 있습니다.

활동 통계

표 4는 활동 통계 페이지의 예제를 표시합니다.

표 4. 활동 통계

활동 통계	
연결	활성화 1, 최대 431
응답 시간	알 수 없음
처리량	연결 0 건/초

표 4. 활동 통계 (계속)

활동 통계	
오늘 처리된 요청	0
전체 처리된 요청	114
요청 오류	3

이러한 서버 활동 통계는 처리한 요청 수, 응답 시간, 처리량, 오늘 처리된 요청, 전체 처리된 요청 및 오류에 대해 서버 통신량을 모니터링하는 데 사용할 수 있습니다. 다음의 구성 변경사항은 활동 페이지의 통계에 영향을 줄 수 있습니다.

- **활성 스레드 수**—서버 요청에 사용할 스레드의 수를 지정합니다. 남은 메모리에 따라 사용 가능한 스레드의 수를 늘리거나 줄일 수 있습니다. 활성화 스레드의 수를 변경하려면, 구성 및 관리 양식에서 서버 구성 -> 서버 관리 -> 성능을 선택하거나 구성 파일의 MaxActiveThreads 지시문을 편집하십시오(256 페이지의 『MaxActiveThreads — 최대 활성 스레드 수 지정』 참조).
- **지속적인 연결**—프록시가 네트워크 연결에 영향을 줄 수 있는 클라이언트에게 지속적인 연결을 허용할지 여부. 구성 및 관리 양식에서 이 설정을 변경하려면 프록시 구성 -> 프록시 성능을 선택하여 지속적인 연결을 사용 가능 또는 사용 불가능하게 하거나, 서버 구성 -> 서버 관리를 선택하여 연결 유지 방법을 정의하십시오. 프록시 구성 파일을 편집하여 지속적인 연결을 구성하려면, 다음 지시문에 대한 참조 섹션을 참조하십시오.
 - 258 페이지의 『MaxPersistRequest — 지속적인 연결에서 수신할 요청의 최대수 지정』
 - 269 페이지의 『PersistTimeout — 클라이언트가 다른 요청을 전송하기 위한 대기 시간 지정』
 - 286 페이지의 『ProxyPersistence — 지속적인 연결 허용』

네트워크 통계

표 5는 네트워크 통계 페이지의 예제를 표시합니다.

표 5. 네트워크 통계

네트워크 통계	
전송 데이터:	1KB/초
수신 데이터:	1KB/초
줄어드는 대역폭:	3KB(0KB/초)
오늘 줄어드는 대역폭:	0KB(0KB/초)

네트워크 통계 양식은 전송 및 수신된 바이트의 데이터 속도를 포함하여 프록시가 실행 중인 네트워크에 대한 정보를 제공합니다.

액세스 통계

액세스 통계 페이지는 액세스 로그의 최근 입력 항목 20개를 표시합니다. 이 페이지는 프록시 액세스 로그(검은색 유형) 및 캐시 액세스 로그(파란색 유형)에 있는 최근 입력 항목을 표시합니다. 로그된 것을 사용자 정의하여, 표시된 것을 사용자 정의할 수 있습니다. 액세스 로그 통계에 대한 자세한 정보는 160 페이지의 『액세스 로그 필터』를 참조하십시오.

프록시 액세스 통계

프록시 액세스 통계 양식은 요청된 URL 및 캐시에서 제공되었는지 여부와 같은 프록시 활동에 대한 정보를 제공합니다. URL 다음은 클라이언트 및 파일 크기(바이트)에 주어진 리턴 코드입니다. 다음 설정을 사용하면 프록시 액세스 통계를 개선할 수 있습니다.

- 자동 캐시 새로 고침을 사용하여, 요청된 문서가 캐시에 있게될 확률을 높이십시오. 자세한 내용은 99 페이지의 제 20 장 『자동 새로 고침 및 사전 로드에 대한 캐시 에이전트 구성』을 참조하십시오.
- 캐시된 파일의 최소 보유 시간을 늘리십시오. 자세한 내용은 96 페이지의 『캐시 최신 정보 구성』을 참조하십시오.
- 로컬 도메인에서 제공된 파일을 캐시하지 마십시오. 이 설정이 캐시가 제공하는 요청의 수를 줄이는 경향이 있다 해도, 캐시에서 제공하는 것 만큼(때로는 더 빠름) 빨리 로컬 인트라넷에서 파일을 제공하면 성능이 손실되지 않습니다. 자세한 내용은 89 페이지의 제 18 장 『캐시되는 내용 제어』를 참조하십시오.

캐시 통계

캐시가 사용 가능하면, 캐시 통계 페이지는 최신의 캐시 액세스 정보를 표시합니다. 이것은 다음과 같은 캐시 및 색인에 대한 정보를 제공합니다.

- 캐시가 현재 조작 가능한지 서버를 시작할 때 다시 색인되는지 여부
- 가비지 콜렉션이 실행 중인지 여부
- 캐시 적중률

여러 캐시 구성 옵션은 캐시 통계 결과를 변경합니다(77 페이지의 제 4 부 『프록시 서버 캐시 구성』 참조).

캐시 새로 고침 요약

캐시 에이전트가 캐시에서 파일을 사전 로드하도록 구성되면, 캐시 새로 고침 요약 페이지에는 최근의 캐시 에이전트 실행에 대한 정보가 표시됩니다. 캐시 에이전트는 정보를 표시하기 위해서 적어도 한 번은 실행되었어야 합니다. 캐시 새로 고침 에이전트가 작동하는 방법을 변경하려면, 다음 사항을 고려하십시오.

- 인트라넷 대부분의 통신이 로컬 웹 사이트로 가지 않으면, 로컬 도메인에 대한 캐시를 사용 불가능하게 하십시오. 자세한 내용은 89 페이지의 제 18 장 『캐시되는 내용 제어』를 참조하십시오.

- 많은 클라이언트 요청이 캐시 액세스 로그에 표시되지 않는 페이지를 요청하면, 수동으로 로드할 URL을 구성할 수 있습니다. 명령은 104 페이지의 『관련 프록시 구성 파일 지시문』을 참조하십시오.
- 사전 로드할 즐겨찾기 URL 수를 조정하십시오. 명령은 104 페이지의 『관련 프록시 구성 파일 지시문』을 참조하십시오.
- 캐시 에이전트가 실행할 수 있는 최대 시간을 지정하십시오. 명령은 104 페이지의 『관련 프록시 구성 파일 지시문』을 참조하십시오.

부록 A. Caching Proxy 명령 사용

이 부록에서는 프록시 서버 명령에 대한 참조를 제공합니다.

cgiparse 명령

목적

cgiparse 명령을 사용하여 CGI 스크립트에 대한 QUERY_STRING 환경 변수를 구문 분석하십시오. QUERY_STRING 환경 변수가 설정되어 있지 않으면, 명령이 표준 입력에서 CONTENT_LENGTH 문자를 읽습니다. 리턴된 출력은 모두 표준 출력에 기록됩니다.

형식

```
cgiparse -Flag [Modifier]
```

매개변수

플래그 즉, 한 문자(-k -f -v -r -i -s -p -c -q -P)로 된 동의어와 기능은 다음과 같습니다.

-keywords | -k

키워드에 대한 QUERY_STRING 구문을 분석합니다. 키워드는 해독되어 한 행에 하나씩 표준 출력에 기록됩니다.

-form | -f

양식 요청으로, QUERY_STRING 구문을 분석합니다. 셸에 의해 평가될 때, 필드 이름 앞의 접두부 FORM_과 함께 셸 변수를 설정하는 문자열을 리턴합니다. 필드 값은 변수의 내용입니다.

-value field-name | -v field-name

양식 요청으로, QUERY_STRING 구문을 분석합니다. *field-name* 값만 돌려보냅니다.

-read | -r

표준 입력에서 CONTENT_LENGTH 문자를 읽고 표준 출력에 기록합니다.

-init | -i

QUERY_STRING이 설정되어 있지 않으면, 표준 입력 값을 읽고 QUERY_STRING을 설정하는 SET 명령문을 이 값으로 돌려보냅니다. 이것은 GET 및 POST 방법과 함께 사용될 수 있습니다. 일반적으로 다음과 같이 사용됩니다.

```
eval 'cgiparse -init'
```

이것은 GET 또는 POST 방법에 상관없이 QUERY_STRING 환경 변수를 설정합니다.

cgiparse는 GET 방법이 사용될 경우 스크립트에서 여러 번 호출될 수 있지만, POST 메소드가 사용되면 한 번만 호출될 수 있습니다. 표준 입력을 읽은 후 다음 **cgiparse**가 POST 메소드를 사용하여 표준 입력 공백을 찾아 무기한으로 대기합니다.

-sep separator | -s separator

여러 값을 분리하는 데 사용되는 문자열을 지정합니다. **-value** 플래그를 사용하는 경우, 기본 분리자는 새 행입니다. **-form** 플래그를 사용하는 경우, 기본 분리자는 콤마(,)입니다.

-prefix prefix | -p prefix

-POST 및 **-form**과 함께 사용되어, 환경 변수 이름을 작성할 때 사용할 접두부를 지정합니다. 기본값은 "FORM_"입니다.

-count | -c

-keywords, **-form**, **-value**와 함께 사용되어, 이들 플래그와 관련된 항목의 계수를 돌려보냅니다.

-keywords | -k

키워드의 수를 돌려보냅니다.

-form | -f

고유한 필드의 수를 돌려보냅니다(여러 값이 하나의 값으로 계산됩니다).

-value field-name | -v field-name

*field-name*에 대한 값의 수를 리턴합니다(*field-name*으로 명명된 필드가 없는 경우, 출력은 0입니다).

-number

-keywords, **-form**, **-value**와 함께 사용되어, 이들 플래그와 관련되어 지정된 어커런스를 돌려보냅니다.

-keywords

*n*번째 키워드를 돌려보냅니다(예를 들어, **-2** -키워드는 두 번째 키워드를 출력합니다).

-form

*n*번째 필드의 모든 값을 돌려보냅니다(예를 들어, **-2 -form**은 두 번째 필드의 모든 값을 출력합니다).

-value field-name

field-name 필드의 여러 값 중 *n*번째 값을 돌려보냅니다(예를 들어, **-2 -value** **-whatsit**은 **whatsit** 필드의 두 번째 값을 출력합니다).

-quiet | -q

모든 오류 메시지를 억제합니다(0이 아닌 종료 상태가 여전히 오류를 표시합니다).

-POST | -P

표준 입력에서의 정보(또는, 파일 이름이 예정된 경우에는 stdin 파일)는 직접 해독되어 셸 변수에 구문 분석되며, QUERY_STRING은 사용되지 않습니다. **-POST**는 **-init** 및 **-form** 옵션을 연속적으로 사용하는 것과 동일합니다.

예

다음 예제는 실제로 QUERY_STRING이 이미 서버에 의해 설정된 사실을 무시합니다. 다음 예에서 \$는 Bourne 셸 프롬프트입니다.

- 키워드 탐색

```
$ QUERY_STRING="is+2%2B2+really+four%3F"
$ export QUERY_STRING
$ cgifparse -keywords
is
2+2
4
four?
$
```

- 모든 양식 필드 구문 분석

```
$ export QUERY_STRING="name1=Value1&name2=Value2%3f+That%27s+right%21";
$ cgifparse -form
FORM_name1='Value1'; FORM_name2='Value2? That'\s right!'
$ eval `cgifparse -form`
$ set | grep FORM
FORM_name1="Value1"
FORM_name2="Value2? That's right!"
$
```

- 하나의 필드값만 추출

```
$ QUERY_STRING="name1=value1&name2=Second+value%3F+That'\s%27s
$ cgifparse -value name1
value1
$ cgifparse -value name2
Second value? That's right!
$
```

결과

- | | |
|---|---|
| 0 | 성공 |
| 1 | 잘못된 명령행 |
| 2 | 환경 변수가 올바르게 설정되지 않음 |
| 3 | 요청 정보를 획득하는 데 실패함 (예를 들어, 요청한 필드가 없거나, 양식 필드 값 요청시 키워드를 포함하는 QUERY_STRING이 없습니다.) |

주: 이러한 오류 코드 중 하나를 수신할 때, 추가 정보 메시지도 수신할 수 있습니다. 메시지는 발행된 명령에 따라 다릅니다.

cgiutils 명령

목적

구문 분석되지 않은 헤더 프로그램의 **cgiutils** 명령을 사용하여 전체 HTTP 1.0 응답을 생성하십시오.

주: 사용자 고유의 리턴값을 돌려보내기 위해 구문 분석하지 않은 헤더(nph) 프로그램을 공급하려면 프로그램 이름이 **nph**-로 시작해야 합니다. 그러면, 서버 헤더가 리턴값을 표준 서버 리턴값으로 덮어쓰는 것을 막을 수 있습니다.

형식

```
cgiutils -Flag [Modifier]
```

*Modifier*에 공백이 있으면, 인용 부호("")로 묶으십시오.

매개변수

-version

버전 정보를 돌려보냅니다.

-nodate

Date: 헤더를 돌려보내지 않습니다.

-noel

헤더 다음에 공백 행을 돌려보내지 않습니다. 이것은 처음 헤더 행 다음에 다른 MIME 헤더를 원하는 경우 유용합니다.

-status nnn

한 세트의 HTTP 헤더 대신에 상태 코드가 *nnn*인 전체 HTTP 응답을 돌려보냅니다. **Expires:** 헤더만을 원하는 경우에는 이 플래그를 사용하지 마십시오.

-reason explanation

HTTP 응답에 대한 이유 행을 지정합니다. 이 플래그는 **-status nnn** 플래그와 함께 사용할 경우에만 가능합니다.

-ct [type/subtype]

MIME Content-Type 헤더를 지정합니다. 이 예제는 text/html의 MIME 내용 유형을 지정합니다.

```
cgiutils -ct text/html
```

*type/subtype*을 생략한 경우, MIME 내용 유형이 기본 text/plain으로 설정됩니다. 이 예제는 MIME 내용 유형을 text/plain으로 설정합니다.

```
cgiutils -ct
```

-ce *encoding*

MIME Content-Encoding 헤더를 지정합니다. 예제는 다음과 같습니다.

```
cgiutils -ce x-compress
```

-cl *language-code*

MIME Content-Language 헤더를 지정합니다. 예제는 다음과 같습니다.

```
cgiutils -cl en_UK
```

-length *nnn*

MIME Content-Length 헤더를 지정합니다.

-expires *Time-Spec*

MIME **Expires:** 헤더를 지정합니다. 이 플래그는 날짜, 시, 분, 초의 결합으로 문서가 남아 있을 시간(문서의 만기 날짜)을 지정합니다. 이것은 문서가 유효하다고 간주되는 시간의 길이입니다. 예제는 다음과 같습니다.

```
cgiutils -expires 2 days 12 hours
```

cgiutils 명령은 지정한 시간을 그리니치 표준시에 추가하여 만기 날짜를 판별합니다. 만기 날짜는 HTTP 형식으로 **만기:** 헤더에 올라갑니다.

-expires now

Date: 헤더와 일치하는 **Expires:** 헤더를 생성합니다.

-uri *URI*

되돌아온 문서에 URI(Universal Resource Identifier)를 지정합니다. URI는 URL과 같은 것으로 간주할 수 있습니다.

-estra *xxx: yyy*

다른 방법으로는 **cgiutils** 명령에 지정할 수 없는 추가 헤더를 지정합니다.

예

- 이 예는 **Expires:** 헤더에 대한 만기 날짜를 계산합니다.

```
cgiutils -expires "1 year 3 months 2 weeks 4 days 12 hours 30 mins"
```

- 다음 예제는 상태 코드와 이유를 지정하고, **Date:** 헤더와 일치하는 **Expires:** 헤더를 지정합니다.

```
cgiutils -status 200 -reason "Virtual doc follows" -expires now
```

다음 경우와 유사한 헤더를 생성할 수 있습니다.

```
HTTP/1.0 200 Virtual doc follows
MIME-Version: 1.0
Server: IBM-ICS
Date: Tue, 05 Jan 1996 03:43:46 GMT
Expires: Tue, 05 Jan 1996 03:43:46 GM
```

cgiutils 명령은 CGI 환경에서 사용 가능하기 때문에 **Server:** 헤더를 자동으로 생성합니다. **Date:** 필드 또한 **-nodate** 플래그가 지정되지 않으면 자동으로 생성됩니다.

MIME 헤더 섹션의 종료를 표시하기 위해 출력 다음에 공백 행이 있습니다. 본인의 몇 가지 헤더와 함께 이것을 따르려면, 다음 예에 표시된 대로 **-noel** (NO-Empty-Line) 플래그를 사용하십시오.

- 헤더 행 다음에 공백 행을 넣지 않으려면, **-noel** 플래그를 사용하십시오.

```
cgiutils -noel -expires "2 days" -nodate
HTTP/1.0 200 Virtual doc follows
MIME-Version: 1.0
Server: IBM-ICS
Expires: Tue, 07 Jan 1996 03:43:46 GMT
```

htadm 명령

목적

htadm 명령을 사용하여 서버 암호 파일을 제어하십시오. 서버에서 파일에 대한 액세스를 제어하기 위해 암호 파일을 사용합니다. 암호 파일에 사용자 이름을 추가하고, 암호 파일에서 사용자 이름을 삭제하고, 사용자 암호를 확인하고, 빈 암호 파일을 작성할 수 있습니다. 또한 이전 사용자 암호를 삭제한 다음 새 암호를 작성하여 사용자의 암호를 변경할 수 있습니다.

주: **htadm** 명령을 사용하여 사용자를 추가하거나 암호를 변경하거나 암호를 확인하려면, 명령행에 암호를 입력해야 합니다. 이 명령은 명령행에서 암호를 최대한 빠르게 없애기 때문에, 시스템(예를 들어, **ps** 명령과 함께)에 나열된 처리를 살펴봄으로써 사용자 암호를 발견하는 것은 거의 불가능합니다.

형식

`htadm -Flag [Modifier]`

매개변수

-adduser *password-file* *user-name* [*password* [*real-name*]]

암호 파일에 사용자 및 암호를 추가합니다. 이 명령을 *password-file*과 함께만 입력하면, 다른 매개변수에 대한 메시지가 나타납니다.

password-file

사용자를 추가하려는 암호 파일의 경로 및 이름.

user-name

추가하려는 사용자의 이름.

사용자 이름으로는 영문자와 숫자만 사용하십시오. 특수 문자는 사용하지 마십시오.

암호 파일에 같은 이름의 사용자가 있으면 명령이 실패합니다.

password

사용자 이름에 대해 정의하려는 암호.

암호는 최대 32자입니다. 암호로는 영문자와 숫자만 사용하십시오. 특수 문자는 사용하지 마십시오.

주:

1. 일부 브라우저는 8자 이상의 암호는 읽고 전송할 수 없습니다. 이러한 한계 때문에, 8자 이상의 암호를 정의한 경우에는 서버가 전체 암호나 처음 8자를 유효한 것으로 인식합니다.

2. 관리자 사용자 이름 및 암호는 운영 체제가 대소문자를 구분하지 않더라도 대소문자를 구분합니다. 구성 및 관리 양식에 액세스할 때, `htadm` 명령을 사용하여 정확한 사용자 이름 및 암호를 입력하십시오.

real-name

추가하고 있는 사용자 이름을 식별하는 데 사용하려는 설명이나 이름. 입력 내용은 모두 암호 파일에 기록됩니다.

-deluser *password-file* [*user-name*]

암호 파일에서 사용자를 삭제합니다. 이 명령을 *password-file*과 함께만 입력하면, *user-name* 매개변수에 대한 메시지가 나타납니다.

password-file

사용자를 삭제하려는 암호 파일의 경로 및 이름.

user-name

삭제하려는 사용자의 이름. 지정한 사용자 이름이 암호 파일에 없으면 명령이 실패합니다.

-passwd *password-file* [*user-name* [*password*]]

암호 파일에 이미 정의된 사용자 이름에 대한 암호를 변경합니다. 이 명령을 *password-file*과 함께만 입력하면, 다른 매개변수에 대한 메시지가 나타납니다.

password-file

암호를 변경하려는 사용자 이름이 있는 암호 파일의 경로 및 이름.

user-name

암호를 변경하려는 사용자 이름. 지정한 사용자 이름이 암호 파일에 없으면 명령이 실패합니다.

password

사용자 이름에 정의하려는 새 암호.

암호는 최대 32자입니다. 암호로는 영문자와 숫자만 사용하십시오. 특수 문자는 사용하지 마십시오.

주:

1. 일부 브라우저는 8자 이상의 암호는 읽고 전송할 수 없습니다. 이러한 한계 때문에, 8자 이상의 암호를 정의한 경우에는 서버가 전체 암호나 처음 8자를 유효한 것으로 인식합니다.
2. 관리자 사용자 이름 및 암호는 운영 체제가 대소문자를 구분하지 않더라도 대소문자를 구분합니다. 구성 및 관리 양식에 액세스할 때, `htadm` 명령을 사용하여 정확한 사용자 이름 및 암호를 입력하십시오.

-check password-file [user-name [password]]

암호 파일에 이미 정의된 사용자 이름에 대한 암호를 확인하고 암호가 올바른지 여부를 알려줍니다. 이 명령을 *password-file*과 함께만 입력하면, 다른 매개변수에 대한 메시지가 나타납니다.

password-file

암호를 확인하려는 사용자 이름이 있는 암호 파일의 경로 및 이름.

user-name

암호를 확인하려는 사용자 이름. 지정한 사용자 이름이 암호 파일에 없으면 명령이 실패합니다.

password

확인하려는 암호. 입력한 암호가 사용자 이름에 정의된 암호면, 이 명령은 표준 출력에 **Correct**를 기록하고 0 리턴 코드로 완료됩니다. 입력한 암호가 사용자 이름에 정의된 암호가 아니면, 이 명령은 표준 출력에 **Incorrect**를 기록합니다.

-create password-file

빈 암호 파일을 작성하십시오.

password-file

작성하려는 암호 파일의 경로 및 이름.

예

- 암호 파일에 사용자를 추가하려면 다음을 입력하십시오.

- Linux 및 UNIX 시스템:

```
htadm -adduser /opt/ibm/edge/cp/server_root/protect/heroes.pwd  
clark superman "Clark Kent"
```

- Windows 시스템:

```
htadm -adduser "C:\Program Files\IBM\edge\cp\server_root\protect\  
heroes.pwd" clark superman "Clark Kent"
```

주: htadm 명령은 한 행에 있어야 합니다. 여기에는 읽기 가능한 행이 하나 이상 표시됩니다. **clark**과 **superman** 사이에 적어도 하나의 공간을 두고 한 행에 실재 명령을 입력하십시오.

- 암호 파일에서 사용자를 삭제하려면 다음을 입력하십시오.

- Linux 및 UNIX 시스템:

```
htadm -deluser /opt/ibm/edge/cp/server_root/protect/  
heroes.pwd clark superman "Clark Kent"
```

- Windows 시스템:

```
htadm -deluser "C:\Program Files\IBM\edge\cp\server_root\protect\  
heroes.pwd" clark superman "Clark Kent"
```

htcformat 명령

목적

htcformat 명령을 사용하여 프록시 캐시를 보유하기 위해서 미처리된 장치나 파일을 준비하십시오. 이 형식 명령은 장치를 프록시 캐시와 함께 사용하는 것으로 지정하기 전에 사용되어야 합니다.

장치 경로가 미처리된 장치를 지정해야 합니다. 미처리된 장치 액세스 방법에 대한 세부사항은 파일 시스템의 문서를 참조하십시오. 예는 77 페이지의 제 4 부 『프록시 서버 캐시 구성』에 있습니다.

주: Linux 2.2 커널은 미처리된 장치로의 캐시를 지원하지 않습니다. Linux 플랫폼에서는 파일 및 메모리만 캐시 저장영역으로 사용될 수 있습니다.

Caching Proxy 캐시의 최소 크기는 2049 블록으로 된 16392KB입니다.

형식

```
htcformat device [-blocksize <block size>] [-blocks number of blocks]
htcformat -file filepath [-blocksize block size] -blocks number of blocks
```

매개변수

-blocksize

이것은 캐시 장치의 중간에 위치하는 블록의 크기를 설정합니다. 블록 크기는 바이트 단위로 되어 있습니다. 기본값은 8192이며, 모든 상황에서 사용됩니다.

-blocks

장치나 파일에 작성할 블록의 수. 파일을 포맷할 때, 파일 크기를 지정하기 위해서 이 인수가 필요합니다. 또한 이 인수는 캐시 저장영역에 사용될 특정 장치나 파티션의 용량을 제한하는 데 사용될 수 있습니다. 블록 인수가 지정되지 않은 경우, 파티션에 적합한 많은 인수가 작성됩니다.

-file

저장 장치 대신 파일을 포맷합니다.

사용법

캐시 시스템은 추가적으로, 색인 및 가비지 콜렉션을 위한 컨테이너로 캐시 파일이나 장치를 분리합니다. 컨테이너의 크기는 특정 블록 수로 설정되며, 컨테이너 크기는 구성할 수 없습니다. 가비지 콜렉션이 실행되기 위해서는, 최소한 두 개의 컨테이너가 필요합니다. 최소 캐시 크기는 16392KB입니다.

htcformat 명령은 두 개 이하의 컨테이너로 된 캐시 장치를 허용하는 포맷 요청을 거부합니다.

예

다음 예는 Solaris의 c0t0d0s0 디스크 파티션을 포맷합니다.

```
htcformat /dev/rdisk/c0t0d0s0
```

다음 예제는 AIX에서 lv02라고 하는 디스크 파티션을 포맷합니다.

```
htcformat /dev/r1v02
```

다음 예제는 Windows에서 d:라고 하는 디스크 파티션을 포맷합니다.

```
htcformat \\.\d:
```

다음 예는 filecache로 명명된 파일을 약 1GB로 포맷합니다.

```
htcformat -file /opt/ibm/edge/cp/filecache -blocks 131072
```

ibmproxy 명령

목적

ibmproxy 명령을 사용하여 서버를 시작하십시오.

서버 구성 파일의 지시문을 사용하여 모든 플래그(-r 제외)를 설정할 수 있습니다.

보통 그 디렉토리를 처음 대하는 사람이 읽게 되는 지침이나 주의사항을 받고 있는 README로 명명된 파일을 작성합니다. 기본적으로 **ibmproxy** 명령은 하이퍼텍스트 버전의 디렉토리에 모든 README 파일을 삽입합니다. 또한 README 파일 명령은 DirReadme 구성 지시문을 사용하여 설정될 수 있습니다.

형식

ibmproxy [-Flag [-Flag [-Flag...]]]

매개변수

-nobg

백그라운드 프로세스가 아니라 포그라운드 프로세스로 서버를 실행하십시오. 기본값은 백그라운드 프로세스로 실행하도록 되어 있습니다.

-nosnmp

SNMP 지원을 사용하지 않습니다.

-p *port-number*

이 포트 수에 대해 인식합니다. 기본 포트 수는 80입니다. 이 플래그는 구성 파일에 지정된 포트 지시문을 덮어씁니다. 기본값이나 구성 파일에 지정된 값을 사용하면, 이 플래그를 생략하십시오.

-r *configuration-file*

구성 파일로 사용할 파일을 지정하십시오. 기본 구성 파일 이외의 구성 파일로 서버를 시작하려는 경우, 이 플래그를 사용해야 합니다. 그러면, 여러 구성 파일을 사용할 수 있습니다.

-restart

현재 실행하고 있는 서버를 재시작합니다. **ibmproxy** 명령은 PidFile에서 실행하고 있는 서버의 프로세스 수를 획득하여, HUP(HangUP) 신호로 전송합니다. 그리고 구성 파일을 재로드하고 로그 파일을 다시 엽니다. 손상을 방지하기 위해 동일한 PidFil, 로그 파일, 프록시 캐시를 사용하여 동시에 두 개의 서버 인스턴스를 실행하지 마십시오.

http 디먼이 PidFile에 액세스하기 위해서 서버가 현재 사용하고 있는 구성 파일을 읽어야 하기 때문에, 재시작할 때 같은 구성 파일을 지정해야 합니다. 서버를 시작할 때, **-r** 플래그 및 고유한 구성 파일을 사용했으면, 이 플래그 및 같은 파일을 **-restart**로 지정해야 합니다.

-snmp

SNMP 지원을 사용합니다.

-unload

Linux에서는 연관된 방화벽 규칙을 제거합니다.

신호 처리 옵션도 Linux 및 UNIX 플랫폼에만 있습니다. Linux 및 UNIX 플랫폼에서 다음 옵션을 사용할 수 있습니다.

SIGTERM

ibmproxy 명령이 완료되면 정지한 후 종료합니다. 즉시 종료하기 위해 **SIGKILL**이나 **CANCEL**을 사용할 수 있습니다.

SIGHUP

실행하면 **ibmproxy** 명령이 재시작되고, 구성 파일을 재로드하며 처리를 계속합니다.

예

- 기본값인 **/etc/ibmproxy.conf** 대신에 **/usr/etc/ibmproxy.conf** 구성 파일을 사용하여 포트 8080에서 서버를 시작하려면 다음을 입력하십시오.

```
ibmproxy -p 8080 -r /usr/etc/ibmproxy.conf
```

- AIX에서, 시스템 자원 제어를 사용하여 기본 구성 파일의 서버를 시작하려면 다음을 입력하십시오.

```
startsrc -s ibmproxy
```

기본 구성 파일이 없으면, **ibmproxy** 명령이 **/Public** 디렉토리 구조를 내보냅니다. 이 구조에는 다른 디렉토리 구조에 대한 소프트 연결을 포함할 수 있습니다.

부록 B. 구성 파일 지시문

이 부록에서는 ibmproxy.conf 구성 파일에 포함된 지시문을 설명합니다.

- **Linux 및 UNIX 시스템의 경우.** 이 지시문은 /etc/ 디렉토리의 ibmproxy.conf 구성 파일에 위치합니다.
- **Windows 시스템의 경우.** 이 지시문은 일반적으로 C:\Program Files\IBM\edge\cp\에 위치합니다.

ibmproxy.conf 파일을 편집하여 서버를 구성할 경우, 이 정보를 참조하십시오. 구성 및 관리 양식을 사용하는 경우에는 이 장을 참조하지 않아도 됩니다.

지시문은 영문자순으로 나열됩니다.

재시작 시 변경되지 않는 지시문

일부 지시문이 서버를 재시작할 때 새로 고쳐지지 않았습니다. 다음과 같이 지시문이 서버 실행 중에 변경되면, 서버를 직접 정지시킨 다음 다시 시작해야 합니다 (15 페이지의 제 5 장『Caching Proxy 시작 및 정지』 참조).

표 6. 재시작 시 새로 고쳐지지 않는 지시문

지시문 그룹	지시문
CGI	DisinheritEnv, InheritEnv
캐시	캐시
로그	AccessLog, CacheAccessLog, ErrorLog, ProxyAccessLog, ServerRoot
네트워크 액세스	BindSpecific, Hostname, ListenBacklog, Port
성능	MaxActiveThreads
RTSP	모든 RTSP 지시문
SSL	모든 SSL 지시문
Linux 및 UNIX 프로세스 제어	GroupId, UserId
기타	TransparentProxy

지시문 개요

이 부록에서는 각 지시문에 대한 다음과 같은 정보를 제공합니다.

- 지시문의 이름과 간단한 설명이 있는 표제
- 사용 명령어
- 일반 구문을 따르는 지시문의 형식:

DirectiveName value

- 해당하는 경우, 구성 파일에서 지시문에 대한 가능한 설정 예제

주: Windows 고유 경로가 있는 지시문 예제에는 설치 시 선택된 서버의 루트 디렉토리인 *server_root*가 포함되기도 합니다.

- 기본값 또는 지시문 값

이 값은 기본 구성 파일에서 코드화된 원래 값입니다. 기본값과 다르게 하려는 구성 파일 부분만 변경하십시오. 처음에 코드화된 기본 값이 없는 지시문은 파일에서 주석 마커(#) 뒤에 표시됩니다. 지시문의 값을 지정하려면 주석 마커를 제거하고 구성 파일의 해당 행에 값을 추가하십시오.

허용 가능 값

다음 목록은 구성 파일에서 허용되는 값에 대해 설명합니다.

- 일부 지시문의 참조 정보에는 *value* 부분에 요청, 경로 이름 또는 호스트 이름에 대한 템플릿이 있습니다. 달리 명시한 경우가 아니면, 템플릿에 별표(*)를 사용할 수 있습니다. 템플릿을 일치시키려면, 별표를 문자열이나 단일 문자로 바꿀 수 있습니다.
- 구성 지시문에서 긍정 문자열로 승인되는 입력값은 다음과 같습니다.
 - Yes
 - On
 - OK
 - Enable
- 구성 지시문에서 부정 문자열로 승인되는 입력값은 다음과 같습니다.
 - No
 - Off
 - None
 - Disable
- 구성 지시문에서 기간을 지정하는 입력값은 다음을 결합한 값입니다.
 - *hh*—시간
 - *hh:mm*—시간 및 분
 - *hh:mm:ss*—시간, 분 및 초
 - *n years*—365일의 년 수
 - *n months*—30일의 개월 수
 - *n weeks*—7일의 주 수
 - *n days*—24시간의 일 수
 - *n hours*—60분의 시간 수
 - *n minutes*—60초의 분 수
 - *n seconds*—초 수

– *n* fortnights—14일간의 수

모든 입력 항목은 초로 변환되어 추가됩니다.

- 공백 문자는 구성 파일에 지정된 파일 이름에 허용되지 않습니다. 공백은 분리자로 처리됩니다.

구성 파일 레코드 구문

구성 파일을 편집할 때 다음 요구사항을 기억하십시오.

- 각 지시문은 새로운 행에서 시작해야 합니다.
- 값은 하나 이상의 공백으로 분리됩니다. 공백 문자와 탭 문자 사이의 구분이 없습니다.
- 설명의 시작은 # 기호로 표시됩니다. # 기호와 행의 끝 사이에 있는 모든 문자는 무시됩니다.
- 지시문에 숫자 기호 또는 공백이 지정되어야 하는 경우, 이스케이프 문자로 백슬래시(\)를 숫자 기호 또는 공백 앞에 사용하십시오. 이스케이프 문자는 다음에 오는 문자를 명령이 아닌 문자로 해석해야 함을 나타냅니다. 예를 들어, 행에 \#이 있으면 서버는 이를 설명의 시작이 아니라, # 기호로 해석하여 문자 읽기를 계속 실행합니다. 행에 \가 있으면, 서버는 이를 값 분리자가 아닌 공백으로 해석하여 값을 작성하기 위해 문자 읽기를 계속 실행합니다.

Caching Proxy 지시문

Caching Proxy 지시문은 다음과 같습니다.

AcceptAnything — 모든 파일 제공

파일의 MIME 유형이 클라이언트가 전송한 ACCEPT: 헤더와 일치하지 않더라도 이 지시문을 사용하여 클라이언트에 파일을 제공하십시오. 이 지시문이 OFF로 설정되어 있으면, 클라이언트가 승인할 수 있는 유형과 MIME 유형이 서로 달라집니다. 대신 오류 페이지가 표시됩니다.

형식

AcceptAnything {on | off}

예제

AcceptAnything off

기본값

AcceptAnything on

AccessLog — 액세스 로그 파일에 대한 경로 이름 지정

이 지시문을 사용하여 액세스 통계를 로그하려는 디렉토리 및 파일의 이름을 지정하십시오. 기본적으로 클라이언트가 로컬 서버에 저장된 데이터에 대한 요청을 서버로 전송할 때마다 서버는 이 로그에 입력 항목을 기록합니다. 일반적으로 이들 입력 항목에는 Caching Proxy 시스템을 기점 서버로 사용할 때 구성 클라이언트 또는 액세스로부터 받은 요청만 포함됩니다. 이 로그에는 프록시나 캐시 액세스 정보가 없습니다.

NoLog 지시문을 사용하여 요청을 로그하지 않을 클라이언트를 지정하십시오. NoLog 지시문에 대한 설명은 263 페이지의 『NoLog — 템플릿과 일치하는 고유한 호스트나 도메인에 대한 로그 입력 항목 압축』을 참조하십시오.

서버가 실행 중이면 매일 자정에 새 로그 파일을 시작합니다. 서버가 실행하지 않고 있으면, 해당 날짜에 로그 파일을 처음 시작할 때 새 로그 파일을 시작합니다. 파일을 작성할 때, 서버는 지정한 파일 이름을 사용하고 날짜 접미부를 추가합니다. 날짜 접미부는 *Mmmddyyyy* 형식으로 되어 있으며, 여기서 *Mmm*은 월의 처음 세 글자이고, *dd*는 해당 월의 일이며, *yyyy*는 년도입니다.

주: 사용자 ID, 그룹 ID, 또는 로그 디렉토리 경로에 대한 서버 기본값을 변경한 경우, 새 디렉토리를 작성하고 디렉토리의 권한 및 소유권을 갱신하십시오. 서버에서 사용자 정의된 로그 디렉토리에 정보를 기록하도록 하려면, 해당 디렉토리에 대한 권한을 755로 설정하고 사용자 정의된 서버 사용자 ID를 소유자로 설정하십시오. 예를 들어, 서버의 사용자 ID를 기본값에서 *jdoe*로 변경하고, 기본 로그 디렉토리를 *server_root/account*로 변경한 경우, *server_root/account* 디렉토리는 755의 권한을 가져야 하며 *jdoe*가 소유해야 합니다.

이전 로그 파일은 하드 드라이브의 상당한 공간을 사용하므로, 제거하는 것이 좋습니다.

형식

AccessLog */directory_path/logfile_name*

예제

AccessLog */logs/accesslog*

기본값

- **Linux 및 UNIX 시스템:** AccessLog */opt/ibm/edge/cp/server_root/logs/local*
- **Windows 시스템:** AccessLog *drive:\Program Files\IBM\edge\cp\logs\local*

AccessLogExcludeMethod — 지정된 메소드에서 요청한 파일 또는 디렉토리에 대한 로그 항목 억제

이 지시문을 사용하여, 파일이나 디렉토리에 액세스하기 위한 특정 메소드가 작성한 요청에 대한 로그를 방지하십시오. 예를 들어, 파일이나 디렉토리에 대한 DELETE 요청을 기록하지 않을 수 있습니다.

구성 파일에 이 지시문에 대한 여러 개의 어커런스를 가질 수 있습니다. 또한 메소드를 방법을 하나 이상의 공백으로 분리한다면, 동일한 지시문에 방법을 여러 개 넣을 수 있습니다.

형식

```
AccessLogExcludeMethod method [...]
```

예제

```
AccessLogExcludeMethod GET
AccessLogExcludeMethod PUT
AccessLogExcludeMethod POST
AccessLogExcludeMethod DELETE
AccessLogExcludeMethod GET PUT
```

기본값

없음. 서버에는 모든 메소드 유형으로 요청한 파일 및 디렉토리가 액세스 로그에 있습니다.

AccessLogExcludeMimeType — 특정 MIME 유형에 대한 프록시 액세스 로그 입력 억제

이 지시문을 사용하여 지정된 MIME 유형의 디렉토리 또는 파일에 액세스하기 위한 프록시 액세스 로그 요청을 기록하지 않도록 지정하십시오. (MIME 유형의 예는 텍스트/html, 이미지/gif, 이미지/jpeg입니다). 예를 들어, GIF 이미지에 대한 액세스 요청을 로그하지 않을 수 있습니다.

구성 파일에 이 지시문에 대한 여러 개의 어커런스를 가질 수 있습니다. 또한 어커런스를 하나 이상의 영역으로 분리한 경우에는 동일한 지시문에 여러 MIME 유형을 넣을 수 있습니다.

주: 이 지시문은 프록시 액세스 로그에만 영향을 미칩니다. MIME 유형별로 캐시된 오브젝트를 나열하는 로그는 필터할 수 없습니다. 이를 수행하려면 AccessLogExcludeURL을 사용하십시오.

형식

```
AccessLogExcludeMimeType MIME_type [...]
```

예제

```
AccessLogExcludeMimeType image/gif
AccessLogExcludeMimeType text/html
AccessLogExcludeMimeType image/gif text/html
```

기본값

없음. 액세스 로그에는 MIME 유형의 파일 및 디렉토리에 대한 모든 요청이 있습니다.

AccessLogExcludeReturnCode — 고유한 리턴 코드에 대한 로그 입력 항목 억제

이 지시문을 사용하여, 지정 범위 내의 오류 코드 수에 있는 액세스 요청을 로그하지 않도록 지정하십시오. 이 오류 코드 수는 프록시 서버 상태 코드입니다. 개별적인 코드를 지정할 수 없습니다. 300을 지정하면, 경로 재지정 코드(301, 302, 303, 304)와 함께 액세스 요청을 제외시킨다는 의미입니다.

구성 파일에 이 지시문에 대한 여러 개의 어커런스를 가질 수 있습니다. 또한 어커런스를 하나 이상의 영역으로 분리한 경우에는 동일한 지시문에 여러 리턴 코드를 넣을 수 있습니다.

형식

```
AccessLogExcludeReturnCode range
```

예제

```
AccessLogExcludeReturnCode 300
```

기본값

없음. 액세스 로그에는 코드와는 상관없이 서버에 대한 모든 요청이 있습니다.

AccessLogExcludeURL — 고유한 파일 또는 디렉토리에 대한 로그 입력 항목 억제

이 지시문을 사용하여, 지정된 URL 템플릿과 일치하는 특정 파일이나 디렉토리에 대한 액세스 요청을 로그하지 않도록 지정하십시오. 예를 들어, GIF 이미지에 대한 액세스 요청을 로그하지 않거나 서버의 특정 파일이나 디렉토리에 대한 액세스 요청을 로그하지 않을 수 있습니다.

구성 파일에 이 지시문에 대한 여러 개의 어커런스를 가질 수 있습니다. 또한 어커런스를 하나 이상의 영역으로 분리한 경우에는 동일한 지시문에 여러 입력 항목을 넣을 수 있습니다.

형식

```
AccessLogExcludeURL file_or_type [...]
```


예제

```
AccessLogExcludeURL *.gif
AccessLogExcludeURL /Freebies/*
AccessLogExcludeURL *.gif /Freebies/*
```

기본값

없음. 서버에서 모든 파일 및 디렉토리에 대한 액세스 요청을 로그합니다.

AccessLogExcludeUserAgent — 고유한 브라우저에서 로그 입력 항목 억제

이 지시문을 사용하여 특정 사용자 에이전트(예: Internet Explorer 5.0)에서 작성된 액세스 요청을 로그하지 않도록 지정하십시오.

구성 파일에 이 지시문에 대한 여러 개의 어커런스를 가질 수 있습니다. 또한 어커런스를 하나 이상의 영역으로 분리한 경우에는 동일한 지시문에 여러 입력 항목을 넣을 수 있습니다.

형식

```
AccessLogExcludeUserAgent user_agent [...]
```

예제

```
AccessLogExcludeUserAgent *Mozilla/2.0
AccessLogExcludeUserAgent *MSIE 5*
```

기본값

기본적으로 ibmproxy.conf 파일에는 AccessLogExcludeUserAgent 지시문에 대한 다음 정의가 포함됩니다.

```
AccessLogExcludeUserAgent IBM_Network_Dispatcher_HTTP_Advisor
AccessLogExcludeUserAgent IBM_Network_Dispatcher_WTE_Advisor
```

위에 나열된 사용자 에이전트는 Caching Proxy 서버 앞에 일반적으로 위치하는 특정 Load Balancer 어드바이저에 대해 정의된 사용자 에이전트입니다. 로그에 대한 쓰기 횟수를 최소화하여 성능을 높이기 위해 이들 사용자 에이전트는 로그되지 않습니다. 기본적으로 서버는 모든 기타 사용자 에이전트에 의해 작성된 액세스 요청을 로그합니다.

AddBlankIcon — 디렉토리 목록의 표제 정렬에 사용된 아이콘의 URL 지정

이 지시문을 사용하여, 서버가 FTP 요청에 대한 프록시로 작동할 때 리턴한 디렉토리 목록의 표제를 정렬하는 데 사용할 아이콘을 지정하십시오. 사용자가 파일들을 구분할 수 있도록 아이콘이 연관된 파일들 옆에 표시됩니다.

아이콘은 공백 아이콘이거나 디렉토리 목록의 표제에 나타날 다른 아이콘일 수 있습니다. 정렬이 올바르게 되려면, 사용하는 아이콘의 크기가 디렉토리 목록에 사용 중인 다른 아이콘의 크기와 동일해야 합니다.

형식

`AddBlankIcon icon_URL alternative_text`

icon_URL

아이콘에 대한 URL의 최종 부분을 지정합니다. 서버는 이 값을 `/icons/` 디렉토리에 추가하여 정식 URL 요청을 완료합니다. 로컬 파일의 요청인 경우, 서버는 맵핑 지시문을 통해서 요청을 변환합니다. 아이콘을 검색하려면, 맵핑 지시문이 요청이 전달할 수 있도록 해야 합니다.

서버를 프록시로 사용하고 있는 경우, 완전한 요청은 서버에 위치 지정하는 정식 URL이어야 합니다.

alternative_text

요청하는 브라우저가 그래픽을 표시하지 않는 경우, 아이콘에 사용할 대체 텍스트를 지정합니다.

예제

```
AddBlankIcon logo.gif logo
```

기본값

- **Linux 및 UNIX:** `AddBlankIcon blank.m.png`
- **Windows:** `AddBlankIcon blank.gif`

아이콘이 공백이므로 기본값이 대체 텍스트를 지정하지 않습니다.

AddDirIcon — 디렉토리 목록의 디렉토리에 대한 아이콘 URL 지정

이 지시문을 사용하여, 디렉토리 목록에 디렉토리를 표시하기 위한 아이콘을 지정하십시오.

형식

`AddDirIcon icon_URL alternative_text`

icon_URL

아이콘에 대한 URL의 최종 부분을 지정합니다. 서버는 이 값을 `/icons/` 디렉토리에 추가하여 정식 URL 요청을 완료합니다. 로컬 파일의 요청인 경우, 서버는 맵핑 지시문을 통해서 요청을 변환합니다. 아이콘을 검색하려면, 맵핑 지시문이 요청이 전달할 수 있도록 해야 합니다.

서버를 프록시로 사용하고 있는 경우, 완전한 요청은 서버에 위치 지정하는 정식 URL이어야 합니다. URL을 로컬 파일에 맵핑하고, 맵핑 지시문은 URL이 전달될 수 있도록 하십시오.

alternative_text

요청하는 브라우저가 그래픽을 표시하지 않는 경우, 아이콘에 사용할 대체 텍스트를 지정합니다.

예제

```
AddDirIcon direct.gif DIR
```

기본값

- **Linux 및 UNIX:** AddDirIcon dir.m.pm.gif DIR
- **Windows:** AddDirIcon dir.gif DIR

AddEncoding — 특정 접미부가 있는 파일의 MIME 콘텐츠 인코딩 지정

이 지시문을 사용하여 특정 접미부가 있는 파일을 MIME 인코딩 유형으로 바인드하십시오. 이 지시문은 거의 사용되지 않습니다.

형식

```
AddEncoding .extension encoding
```

.extension

파일 접미부 패턴을 지정합니다.

encoding

해당 접미부 패턴과 일치하는 파일에 바인드하려는 MIME 인코딩 유형을 지정합니다.

예제

```
AddEncoding .qp quoted_printable
```

기본값

```
AddEncoding .Z x-compress
```

AddIcon — MIME 콘텐츠 유형이나 인코딩 유형에 아이콘 바인드

이 지시문을 사용하여 파일을 고유한 MIME 콘텐츠 유형이나 인코딩 유형과 함께 표시하기 위한 아이콘을 지정하십시오. 서버에서 FTP 디렉토리 목록을 비롯한 디렉토리 목록에 아이콘을 사용합니다.

형식

```
AddIcon icon_URL alternative_text MIME_type_template
```

icon_URL

아이콘에 대한 URL의 최종 부분을 지정합니다. 서버는 이 값을 /icons/ 디렉토리에 추가하여 정식 URL 요청을 완료합니다. 로컬 파일의 요청인 경우, 서버는 맵핑 지시문을 통해서 요청을 변환합니다. 아이콘을 검색하려면, 맵핑 지시문이 요청이 전달할 수 있도록 해야 합니다.

서버를 프록시로 사용하고 있는 경우, 완전한 요청은 서버에 위치 지정하는 정식 URL이어야 합니다. URL을 로컬 파일에 맵핑하고, 맵핑 지시문은 URL이 전달될 수 있도록 하십시오.

alternative_text

요청하는 브라우저가 그래픽을 표시하지 않는 경우, 아이콘에 사용할 대체 텍스트를 지정합니다.

type_template

MIME 콘텐츠 유형이나 인코딩 유형 템플릿을 지정합니다. 콘텐츠 유형 템플리트에는 항상 슬래시(/)가 있습니다. 인코딩 유형 템플리트에는 슬래시가 없습니다.

예제

```
AddIcon    video_file.m.pm.gif    MOV    video/*
```

기본값

ibmproxy.conf 구성 파일의 AddIcon 지시문에 여러 개의 기본값이 설정됩니다.

AddParentIcon — 디렉토리 목록의 상위 디렉토리를 표시하는 아이콘에 URL 지정

이 지시문을 사용하여 디렉토리 목록에 상위 디렉토리를 표시하기 위한 아이콘을 지정 하십시오.

형식

```
AddParentIcon    icon_URL    alternative_text
```

icon-URL

아이콘에 대한 URL의 최종 부분을 지정합니다. 서버는 이 값을 /icons/ 디렉토리에 추가하여 정식 URL 요청을 완료합니다. 로컬 파일의 요청인 경우, 서버는 맵핑 지시문을 통해서 요청을 변환합니다. 아이콘을 검색하려면, 맵핑 지시문이 요청이 전달할 수 있도록 해야 합니다.

서버를 프록시로 사용하고 있는 경우, 완전한 요청은 서버에 위치 지정하는 정식 URL이어야 합니다. URL을 로컬 파일에 맵핑하고, 맵핑 지시문은 URL이 전달될 수 있도록 하십시오.

alternative_text

요청하는 브라우저가 그래픽을 표시하지 않는 경우, 아이콘에 사용할 대체 텍스트를 지정합니다.

예제

```
AddParentIcon parent.gif UP
```

기본값

```
AddParentIcon dir-up.gif UP
```

AddType — 특정 접미부가 있는 파일의 데이터 유형 지정

이 지시문을 사용하여 특정 접미부가 있는 파일을 MIME 유형 및 하위유형으로 바인드하십시오. 구성 파일에 이 지시문에 대한 여러 개의 어커런스를 가질 수 있습니다. 서버는 일반적으로 사용되는 대부분의 접미부에 대한 기본값을 제공합니다.

형식

```
AddType .extension type/subtype encoding [quality[ character_set]]
```

.extension

파일 접미부 패턴. 다음의 두 가지 특수 접미부 패턴에서만 와일드 카드 문자(*)를 사용할 수 있습니다.

- *.*** 점 문자(.)가 있고 다른 규칙과 일치하지 않는 모든 파일 이름에 연결합니다.
- *** 점 문자(.)가 없고 다른 규칙과 일치하지 않는 모든 파일 이름에 연결합니다.

type/subtype

해당 접미부 패턴과 일치하는 파일에 바인드하려는 MIME 유형 및 하위유형.

encoding

데이터가 변환된 MIME 콘텐츠 인코딩. 또한 FTP 프록시 서버가 인코딩을 사용하여 파일이 2진 모드에서 검색될 것인지를 판별할 수 있습니다. 대부분의 경우, 적절한 인코딩은 7bit, 8bit 또는 binary이며, 다음과 같이 판별됩니다.

7bit 데이터는 모두 1000자 이하의 축약형 8859-1 ASCII 데이터 행으로 표시됩니다. 원본 코드나 일반 텍스트 파일은 주로 이 카테고리에 해당합니다. 밑줄 친 문자나 강조된 문자가 들어 있는 파일은 예외입니다.

8bit 데이터는 축약형 행으로 표시되지만, 이 데이터에는 고 비트 세트(예: 밑줄 친 문자 또는 강조된 문자)가 있는 문자가 들어 있습니다. 유럽 사이트의 포스트스크립트 파일이나 텍스트 파일이 주로 이 목록에 해당합니다.

binary

이 인코딩은 모든 데이터 유형에 사용할 수 있습니다. 데이터에는 비 ASCII 문자뿐만 아니라 1000자 이상의 긴 행도 들어 있습니다. image/*, audio/* 및 video/* 유형의 거의 모든 파일이 이 카테고리에 해당하며, application/* 유형의 2진 데이터 파일도 여기에 해당합니다.

다른 인코딩 값은 2진과 동일하게 처리되며, 콘텐츠 인코딩 MIME 헤더로서 MIME 헤더에 전달됩니다. 스펙 7bit 및 8bit는 MIME 헤더로 전송되지 않았습니다.

quality

콘텐츠 유형의 상대값(0.0에서 1.0의 크기)의 선택적 지시자를 지정합니다. 파일의 여러 표현이 요청과 일치하는 경우, 품질값이 사용됩니다. 서버는 최고 품질값과 연관된 파일을 선택합니다. 예를 들어, 파일 Internet.ps를 요청하면, 서버는 다음과 같은 AddType 지시문 세트를 갖습니다. 서버는 품질 번호가 높기 때문에 application/postscript 행을 사용합니다.

```
AddType .ps application/postscript 8bit 1.0
AddType *.* application/binary binary 0.3
```

character_set

텍스트 파일과 연관시키려는 문자 세트의 선택적 지시자. 문자 세트를 지정한 파일의 경우, 서버는 파일을 표시할 때 사용할 문자 세트의 콘텐츠를 클라이언트 브라우저에 알립니다. *character_set* 필드의 값을 설정할 경우, *quality* 필드의 값도 포함시켜야 합니다.

예제

```
AddType .bin application/octet-stream binary 0.8
```

기본값

구성 파일(ibmproxy.conf)에는 AddType 지시문에 대한 여러 개의 기본 설정이 있습니다.

AddUnknownIcon — 디렉토리 목록의 알 수 없는 파일 유형에 대한 아이콘 URL 지정

이 지시문을 사용하여, 디렉토리 목록에 알 수 없는 파일 유형이 있는 파일을 표시하기 위한 아이콘을 지정하십시오.

형식

```
AddUnknownIcon icon_URL alternative_text
```

icon_URL

아이콘에 대한 URL의 최종 부분을 지정합니다. 서버는 이 값을 /icons/에 추가하여 정식 URL 요청을 완료합니다. 로컬 파일의 요청인 경우, 서버는 맵핑 지시문을 통해서 요청을 변환합니다. 아이콘을 검색하려면, 맵핑 지시문이 요청이 전달될 수 있도록 해야 합니다.

서버를 프록시로 사용하고 있는 경우, 완전한 요청은 서버에 위치 지정하는 정식 URL이어야 합니다. URL을 로컬 파일에 맵핑하고, 맵핑 지시문은 URL이 전달될 수 있도록 하십시오.

alternative_text

요청하는 브라우저가 그래픽을 표시하지 않는 경우, 아이콘에 사용할 대체 텍스트를 지정합니다.

예제

```
AddUnknownIcon saywhat.gif unknown
```

기본값

- **Linux 및 UNIX:** AddUnknownIcon unknown.gif ???
- **Windows:** AddUnknownIcon unknown.gif ???

AdminPort — 관리 페이지나 양식을 요청하기 위한 포트 지정

이 지시문을 사용하여, 관리자가 서버 상태 페이지나 구성 양식에 액세스하는 데 사용할 수 있는 포트를 지정하십시오. 이 포트에 대한 요청은 포트 지시문을 사용하여 정의된 표준 포트에 대한 다른 모든 수신 요청과 함께 대기열에 넣을 수 없습니다. 그러나 AdminPort의 요청은, 예를 들어 Pass, Exec, Protect와 동일한 일반 액세스 제어 및 요청 매핑 규칙을 거칩니다.

주: 관리 포트는 포트 지시문을 사용하여 정의된 표준 포트와 동일해서는 안됩니다.

형식

```
AdminPort port_number
```

예제

```
AdminPort 2001
```

기본값

```
AdminPort 8008
```

AggressiveCaching — 캐시할 수 없는 파일에 대한 캐시 지정

이 지시문을 사용하여, 캐시할 수 없는 파일로 표시되고 기점 서버가 리턴한 파일을 캐시할지 여부를 지정하십시오. 이 지시문에 따라 캐시된 캐시할 수 없는 파일은 must revalidate로 표시됩니다. 파일을 요청할 때마다, 프록시 서버에서 기점 서버로 If-Modified-Since 요청을 전송하여 캐시에서 응답을 제공하기 전에 응답의 유효성을 재확인합니다. 현재, 이 지시문의 영향을 받는 캐시할 수 없는 파일만 기점 서버의 응답이며, 이 응답에는 cache-control: no-cache 헤더가 들어 있습니다. 이 지시문은 여러 번 지정할 수 있습니다.

형식

```
AggressiveCaching url_pattern
```

예제

AggressiveCaching http://www.hosta.com/*
AggressiveCaching http://www.hostb.com/*

역호환의 경우, 이 지시문에 대한 이전 구문(AggressiveCaching {on | off})이 이제 다음과 같이 처리됩니다.

AggressiveCaching on은 AggressiveCaching * .로 처리됩니다.

AggressiveCaching off는 무시됩니다.

주: AggressiveCaching off와 AggressiveCaching url_pattern이 모두 지정된 경우, AggressiveCaching off가 무시되고 경고 메시지가 표시됩니다.

기본값

없음

AlwaysWelcome — 환영 파일의 요청된 디렉토리 탐색 여부 지정

디렉토리 이름은 있지만 파일 이름은 없는 요청에 대해서, AlwaysWelcome 지시문은 서버가 리턴할 환영 파일을 디렉토리에서 찾을지 여부를 제어합니다. 기본적으로 AlwaysWelcome은 on 값으로 설정됩니다. 이렇게 하면, 서버에서는 Welcome 지시문에서 지정된 이름과 일치하는 파일을 요청된 디렉토리에서 찾습니다. 일치하는 파일이 있으면, 요청자에게 되돌아 갑니다. 서버가 디렉토리의 파일과 Welcome 지시문의 파일 이름간에 일치하는 파일을 하나 이상 발견하면, Welcome 지시문의 순서가 돌려보낼 파일을 판별합니다. 서버에서는 구성 파일의 맨 위와 가장 가까운 Welcome 지시문을 사용합니다.

형식

AlwaysWelcome on | off

기본값

AlwaysWelcome on

관련 지시문

- 315 페이지의 『Welcome — 환영 파일의 이름 지정』

appendCRLFtoPost — CRLF를 POST 요청에 추가

이 지시문을 사용하여, Caching Proxy가 캐리지 리턴 및 줄 바꿈 문자를 POST 요청의 본문 끝에 추가해야 하는 대상 URL을 지정하십시오. 이 지시문은 여러 번 지정할 수 있습니다.

주: POST 요청을 처리하는 알려진 문제점을 가지고 있는 URL에 대해서만 이 지시문을 지정하십시오.

형식

`appendCRLFtoPost url_pattern`

예제

`appendCRLFtoPost http://www.hosta.com/`

기본값

없음

ArrayName — 원격 캐시 배열 이름 지정

이 지시문을 사용하여 서버가 공유할 원격 캐시 배열을 지정하십시오.

주: 배열을 설정할 때, Hostname 지시문을 해당 배열의 모든 구성원에 동일하게 구성하십시오.

형식

`ArrayName array_name`

기본값

없음

Authentication — 인증 단계 사용자 정의

이 지시문을 사용하여 서버 요청 프로세스의 인증 단계 중에 서버가 호출할 사용자 정의된 응용프로그램 기능을 지정하십시오. 이 코드는 인증 설계에 따라 실행됩니다. BASIC 인증만 지원됩니다.

주: 인증은 권한 부여 프로세스의 일부이므로, 권한 부여가 필요할 때에만 발생합니다.

형식

`Authentication type /path/file:function_name`

type

응용프로그램 기능이 호출되는지를 추가적으로 판별할 인증 설계를 지정합니다. 별표(*) 및 BASIC은 모두 승인된 값입니다.

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능에 부여한 이름을 지정합니다.

예제

`Authentication BASIC /ics/api/bin/icsextpgm.so:basic_authentication`

기본값

없음

Authorization — 권한 부여 단계 사용자 정의

이 지시문을 사용하여 서버 요청 프로세스의 인증 단계 동안 서버가 호출하는 사용자 정의된 응용프로그램 기능을 지정하십시오. 이 코드는 요청한 오브젝트를 클라이언트에 제공할 수 있는지를 확인합니다.

형식

Authorization request_template /path/file:function_name

request_template

응용프로그램 기능이 호출되는지 여부를 추가적으로 판별하는 요청에 대한 템플릿을 지정합니다. 스펙은 프로토콜, 도메인, 호스트를 포함할 수 있고, 앞에 슬래시(/)가 붙을 수 있으며, 별표(*)를 와일드 카드로 사용할 수 있습니다. 예를 들어, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /* 및 *는 모두 유효합니다. Caching Proxy를 역방향 프록시로 사용하는 경우, 요청 템플릿은 문서 루트(/)에서 시작해야 합니다.

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능에 부여한 이름을 지정합니다.

예제

```
Authorization /index.html /api/bin/icsextpgm.so:auth_url
```

기본값

없음

AutoCacheRefresh — 캐시 새로 고침을 사용할지 여부 지정

이 지시문을 사용하여, 캐시 새로 고침을 사용 또는 사용하지 마십시오. 새로 고침을 사용하는 경우 캐시 콘텐츠가 자동으로 새로 고쳐집니다. 새로 고침을 사용하지 않는 경우에는 캐시 에이전트가 호출되지 않고 모든 설정이 무시됩니다. 예를 들어, Linux 및 UNIX 시스템에서 **cron** 작업을 사용하는 것과 같이 다른 방법으로 캐시 에이전트를 시작하는 경우, 이 지시문을 off로 설정하십시오.

형식

AutoCacheRefresh {on | off}

기본값

AutoCacheRefresh On

BindSpecific — 서버가 하나 또는 모든 IP 주소로 바인드 여부 지정

다중 홈 시스템에 이 지시문을 사용하여 서버가 단일 네트워크 주소를 인식하는지를 지정하십시오. 값을 On으로 설정한 경우, 서버는 모든 로컬 IP 주소에 바인드하지 않고 Hostname 지시문에 지정된 IP 주소에 바인드합니다.

이 지시문이 지정되지 않을 경우, 서버를 기본 호스트 이름으로 바인드합니다.

이 지시문을 변경하면 서버를 직접 정지시킨 다음 다시 시작해야 합니다. 서버를 재시작하기만 할 경우에는 서버가 변경사항을 인식하지 않습니다 (15 페이지의 제 5 장 『Caching Proxy 시작 및 정지』 참조).

형식

BindSpecific {on | off} [OutgoingSrcIp *ip_addr* | *host_name*]

[OutgoingSrcIp *ip_addr* | *host_name*]

전송 연결을 하려는 경우, OutgoingSrcIp 옵션은 Caching Proxy를 통해 특정 소스 IP 주소를 사용하도록 합니다. 이는 DMZ에서 Caching Proxy 설정에 유용하고, 특수 방화벽 규칙이 필요로 할 때 유용합니다.

기본값

BindSpecific Off

BlockSize — 캐시 내 블록 크기 지정

이 지시문은 캐시 장치의 매체에 있는 블록의 크기(바이트)를 지정합니다. 기본적으로 값은 8192입니다. 유일하게 지원되는 크기이므로 값을 변경하지 마십시오. 자세한 정보는 181 페이지의 『htcformat 명령』의 참조 섹션을 참조하십시오.

형식

BlockSize *size*

기본값

기본적으로 구성 파일에 있는 BlockSize의 설정값은 없습니다 (기본값이 8192입니다).

CacheAccessLog — 캐시 액세스 로그 파일에 대한 경로 지정

이 지시문을 사용하여 서버가 프록시 캐시로 액세스 로그를 저장하기를 원하는 경로 및 파일의 이름을 지정하십시오. 이 지시문은 서버가 프록시로 실행하고 있는 경우에만 유효합니다. 211 페이지의 『CacheRefreshTime — 캐시 에이전트를 시작할 시기 지정』에서 자세한 내용을 참조하십시오.

프록시 캐시에 대한 요청을 로그할 수 있게 하려면, Caching 지시문을 On으로 설정하고 CacheMemory 및 CacheAccessLog 지시문에 대한 값을 설정해야 합니다. 선택적으로, CacheDev 지시문을 사용하여 하나 이상의 캐시 장치를 정의할 수 있습니다.

CacheAccessLog는 ServerRoot에 대한 절대 경로이거나 또는 상대 경로일 수 있습니다 (각각 한 개의 예제만 표시합니다).

형식

CacheAccessLog *path/file*

예제

```
CacheAccessLog /absolute/path/logfile
CacheAccessLog /logs/logfile
```

기본값

- **Linux 및 UNIX 시스템:** CacheAccessLog /opt/ibm/edge/cp/server_root/logs/cache
- **Windows 시스템:** CacheAccessLog drive:\Program Files\IBM\edge\cp\logs\cache

CacheAlgorithm — 캐시 알고리즘 지정

이 지시문을 사용하여, 가비지 콜렉션 도중 서버가 사용할 캐시 알고리즘을 지정하십시오.

형식

CacheAlgorithm {bandwidth | responsetime | blend}

bandwidth

네트워크 대역폭 줄임을 최대화하기 위해 시도합니다.

responsetime

사용자 응답 시간을 최소화하기 위해 시도합니다.

blend

bandwidth와 responsetime을 균형있게 조합하기 위해 사용합니다.

기본값

CacheAlgorithm bandwidth

CacheByIncomingUrl — 캐시 파일 이름 생성을 위한 기초 지정

이 지시문을 사용하여 생성된 캐시 파일 이름을 수신 요청 URL에 기초할 것인지를 지정합니다.

이 지시문이 on으로 설정되면, 캐시 파일 이름은 수신 URL을 기초로 생성됩니다. 이 지시문이 off로 설정되면, 수신 URL은 먼저 모든 적용 가능한 이름 변환 플러그인, MAP 규칙 및 PROXY 규칙을 통해 전달되며 생성된 캐시 파일 이름은 결과 URL을 기반으로 합니다.

주: URL 기반 캐시 필터에 대해 역방향 프록시 시나리오에서 캐시 필터를 정의하는 경우, /(슬래시) 문서 루트로 시작하는 형식을 사용하십시오(예: /test/index.html). 이 형식에는 프로토콜을 포함하지 않습니다(예: http://).

형식

CacheByIncomingUrl {on | off}

기본값

CacheByIncomingURL off

CacheClean — 캐시 파일 보존 기간 지정

이 지시문을 사용하여 서버가 캐시된 파일을 보존하는 기간을 지정하십시오. 가비지 콜렉션이 실행될 경우, 서버는 파일의 만기 날짜와 관계없이 해당 시간을 초과한 캐시된 파일을 삭제합니다. 언제든지 파일은 지정된 시간보다 오래 캐시에 보존되어야 하며, 서버는 파일을 제공하기 전에 유효한지 파일의 유효성을 다시 검증합니다.

형식

CacheClean *time_specification*

예제

CacheClean 2 weeks

기본값

CacheClean 1 month

CacheDefaultExpiry — 파일에 대한 기본 만기 시간 지정

이 지시문을 사용하여 만기 또는 최종 변경 헤더를 제공하지 않은 파일에 대해 서버가 기본 만기 시간을 설정합니다. URL 템플리트와 URL이 이 템플리트와 일치하는 파일에 대해 만기 시간을 지정하십시오. 이 지시문에 대한 여러 개의 어커런스가 구성 파일에 포함될 수 있습니다. 각 템플리트에 대한 별개의 지시문을 포함시키십시오. URL 템플리트에는 프로토콜이 있어야 합니다. 개월, 주, 일, 시간을 조합한 시간값을 지정하십시오.

형식

CacheDefaultExpiry *URL_template expiration_time*

기본값

CacheDefaultExpiry ftp:* 1 day
CacheDefaultExpiry gopher:* 2 days
CacheDefaultExpiry http:* 0 days

주: HTTP 프로토콜의 기본 만기는 0일입니다. 대부분의 스크립트 프로그램에서 만기 날짜를 제공하지 않으므로 이 값을 유지하는 것이 좋습니다. 이는 출력 즉시 만기됩니다. 0이 아닌 값은 클라이언트에게 시효가 지난 콘텐츠를 제공할 수 있습니다.

CacheDev — 캐시 저장영역 장치 지정

이 지시문을 사용하여 캐시 저장영역을 지정하십시오. 파일이나 공 디스크 파티션을 지정할 수 있습니다. AIX 플랫폼에는 미처리된 논리 볼륨을 지정할 수 있습니다. 메모리 캐시를 사용하지 않을 때, 공 디스크 캐시의 성능이 최고가 됩니다

캐시 장치를 지정하려면 먼저 캐시 장치를 준비해야 합니다. 캐시 장치를 준비하려면 **htcformat** 명령을 사용하여 포맷하십시오. 자세한 내용은 181 페이지의 『htcformat 명령』을 참조하십시오.

여러 캐시 장치를 지정할 수 있습니다. 각 장치는 동일한 CacheMemory 및 BlockSize 값과 연관됩니다. 그러나 메모리가 약 8MB인 프록시 서버 시스템에서 각 캐시 장치에 메모리 오버헤드가 발생합니다. 용량이 적은 장치가 여러 개 있는 것보다 갯수가 적더라도 용량이 큰 장치가 더 효율적입니다. 효율성을 최대로 높이려면, 전체 디스크를 다른 파티션 없이 하나의 큰 파티션으로 사용하십시오. 캐시 저장영역에 대한 자세한 내용은 115 페이지의 『디스크 캐시 성능 최대화』에 나와 있습니다.

형식

CacheDev {raw_disk_partition | file}

예제

AIX: CacheDev /dev/r1v02

HP-UX: CacheDev /dev/rdisk/clt15d0

Linux: CacheDev /opt/IBMWTE/filecache1

Solaris: CacheDev /dev/rdisk/clt3d0s0

Windows: CacheDev \\.\E:

기본값

없음

CacheExpiryCheck — 서버가 만기된 파일을 리턴할지 여부 지정

이 지시문을 사용하여 서버가 만기된 캐시 파일을 리턴할 것인지를 지정합니다. 서버가 만기된 파일을 리턴할 수 있도록 하려면 이 값을 off로 설정하십시오. 클라이언트가 만기된 파일을 요청할 때 프록시가 기점 서버에서 보다 최신 버전을 체크하도록 하려면,

기본값인 On을 사용하십시오. 일반적으로 관리자는 서버가 만기된 파일을 리턴하도록 설정하지 않습니다. 그러나 서버를 시연 중이거나 리턴되는 콘텐츠가 그다지 중요하지 않다고 판단하는 경우에는 예외입니다.

형식

CacheExpiryCheck {on | off}

기본값

CacheExpiryCheck On

CacheFileSizeLimit — 캐시될 파일의 최대 크기 지정

이 지시문을 사용하여 캐시될 파일의 최대 크기를 지정합니다. 이 크기보다 큰 파일은 캐시되지 않습니다. 값은 바이트(B), 킬로바이트(K), 메가바이트(M), 기가바이트(G)로 지정할 수 있습니다. 스펙이 숫자와 조치 단위(B, K, M, G) 사이에 공간을 포함하지 여부는 상관없습니다.

형식

CacheFileSizeLimit *maximum* {B | K | M | G}

기본값

CacheFileSizeLimit 4000 K

CacheLastModifiedFactor — 만기 날짜 판별을 위한 값 지정

이 지시문을 사용하여 고유한 URL이나 템플리트와 일치하는 모든 URL의 만기 날짜를 계산하기 위해 사용할 값을 지정합니다.

HTTP 서버에서는 파일에 최종 변경 시간을 제공하는 경우는 많지만, 만기 날짜는 제공하지 않습니다. 마찬가지로 FTP 파일에 최종 변경 시간 소인은 있지만, 만기 날짜는 없습니다. Caching Proxy는 최종 변경 시간에 기초하여, 파일의 만기 날짜를 계산합니다. 최종 변경 시간을 사용하여, 파일이 변경된 이후의 기간을 판별하고 이 기간을 CacheLastModifiedFactor 지시문에 있는 값으로 곱합니다. 이 계산의 결과는 파일의 수명 또는 파일이 유효하지 않게 될 때까지의 시간입니다.

또한 off 또는 -1을 지정하여 지시문을 사용하지 않음으로써 만기 날짜를 계산하지 않을 수 있습니다. 프록시 서버는 구성 파일에 나타나는 순서대로

CacheLastModifiedFactor 지시문을 읽습니다. 프록시 서버는 캐시된 파일에 적용할 수 있는 첫 번째 지시문을 사용합니다.

형식

CacheLastModifiedFactor *url factor*

url

프로토콜을 포함하여 캐시 중인 파일의 전체 URL을 지정합니다. 마스크를 적용하기 위해서 별표(*)가 있는 URL 템플리트를 와일드 카드로 사용할 수 있습니다.

factor

계산에 사용할 요소를 지정합니다. 값을 off 또는 -1로 지정할 수도 있습니다.

예제

```
CacheLastModifiedFactor *://hosta/* off
CacheLastModifiedFactor ftp://hostb/* 0.30
CacheLastModifiedFactor ftp://* 0.25
CacheLastModifiedFactor http://* 0.10
CacheLastModifiedFactor * 0.50
```

기본값

```
CacheLastModifiedFactor http://* 0.10
CacheLastModifiedFactor http://*.htm* 0.20
CacheLastModifiedFactor http://*.gif 1.00
CacheLastModifiedFactor http://*.jpg 1.00
CacheLastModifiedFactor http://*.jpeg 1.00
CacheLastModifiedFactor http://*.png 1.00
CacheLastModifiedFactor http://*.tar 1.00
CacheLastModifiedFactor http://*.zip 1.00
CacheLastModifiedFactor http:* 0.15
CacheLastModifiedFactor ftp:* 0.50
CacheLastModifiedFactor * 0.10
```

기본값이 0.14이면, 일주일 전에 수정된 파일이 하루 안에 만기됩니다.

CacheLocalDomain — 로컬 도메인을 캐시할지 여부 지정

이 지시문을 사용하여 해당 도메인에 있는 호스트의 URL을 프록시로 캐시할지 여부를 지정합니다. 인트라넷의 로컬 사이트는 일반적으로 URL을 빠르게 로드하기에 충분한 내부 대역폭을 가지고 있기 때문에 캐시할 필요가 없습니다. 로컬 사이트를 캐시하지 않으면, 검색하는 데 시간이 더 오래 걸리는 URL에 대한 캐시 영역을 절약하게 됩니다.

형식

```
CacheLocalDomain {on | off}
```

기본값

```
CacheLocalDomain on
```

CacheMatchLanguage — 리턴된 캐시 콘텐츠의 언어 환경 설정을 지정

백엔드 서버에 동일한 URL의 고객에 언어 변환을 리턴하는 성능이 있는 경우, 이 지시문을 사용하여 동일한 URL에 대해 다른 언어의 캐시를 지원하십시오. 지시문은 Caching Proxy가 캐시 응답 언어로 된 요청에 언어 환경 설정을 확인하도록 허용합니다.

Caching Proxy가 캐시 콘텐츠를 로드하기 전에, CacheMatchLanguage가 사용 가능한 경우, 요청에 대한 Accept-Language 헤더에서의 언어 환경 설정을 캐시 콘텐츠의 언어와 비교합니다. 또한 Caching Proxy는 간격에 대한 환경 설정을 비교합니다. 간격 환경 설정이 지정된 한계 미만인 경우, 캐시 사본을 리턴합니다. 그렇지 않은 경우, 프록시는 요청을 백엔드 서버에 전달하여 요청된 언어로 새로운 사본을 얻습니다.

형식

```
CacheMatchLanguage {on | off} lang-prefer-distance-limit special-id-for-all-lang  
lang-prefer-distance-limit
```

0.001– 0.9999의 범위 내에서 값을 지정합니다.

special-id-for-all-lang

Content-Language 헤더에서 서버로부터 리턴된 언어 문자열을 지정하여 응답이 모든 언어 환경 설정에 대해 사용할 수 있음을 알려십시오.

예제

다음은 지시문, 캐시 오브젝트 및 요청에 대한 구성 예입니다.

```
CacheMatchLanguage On 0.2
```

캐시 오브젝트가 중국어(zh_cn)이고, 요청은 다음과 같은 경우:

```
GET / HTTP/1.1  
...  
Accept-Language: en_US;q=1.0, zh_cn;q=0.7, ja;q=0.3  
....
```

이 요청에 대해, 고객은 영어(코드 및 품질은 en_US/1.0), 중국어(코드 및 품질은 zh_cn/0.7) 및 일본어(코드 및 품질은 ja/0.3) 순으로 페이지를 요청합니다. 캐시 오브젝트는 중국어로 되어 있습니다. 따라서 최상의 품질과 현재 설정된 언어의 품질 차이는 1.0에서 0.7을 뺀 0.3입니다. CacheMatchLanguage 지시문이 한계를 0.2로 지정하고, 0.3은 한계보다 큰 값이므로, 프록시는 캐시 오브젝트를 리턴하는 대신, 해당 URL의 새로운 사본을 서버에 요청합니다.

응답을 리턴하고 다음 요청이 들어오지 않는 경우, 서버가 Content-Language 헤더에서 언어를 지정하거나 special-id-for-all-lang을 지정하지 않는 경우, 프록시는 언어 환경 설정을 일치시키지 않고 캐시 사본을 리턴합니다.

기본값

```
CacheMatchLanguage off
```

CacheMaxExpiry — 캐시 파일의 최대 수명 지정

이 지시문을 사용하여 파일이 캐시에 남아 있을 수 있는 최대 기간을 정의합니다. 캐시된 파일의 수명은 파일이 갱신 기점을 확인하지 않고도 캐시에서 제공될 수 있는 기간입니다. 어떤 경우에는 캐시 파일의 추정 수명이 파일을 보존하기를 원하는 기간보다 길

수 있습니다. 기점에서 지정하거나 Caching Proxy가 계산하든지간에 파일의 수명은 CacheMaxExpiry 지시문에 지정된 한계보다 길 수는 없습니다.

이 지시문에 대한 어커런스가 여러 개 구성 파일에 있을 수 있습니다. 각 템플릿에 대한 별개의 지시문을 포함시키십시오.

형식

CacheMaxExpiry *URL lifetime*

URL

프로토콜을 포함하여 캐시 중인 파일의 완전한 URL을 지정합니다. 마스크를 적용하기 위해서 별표(*)가 있는 URL 템플릿을 와일드 카드로 사용할 수 있습니다.

lifetime

URL 템플릿과 일치하는 캐시 파일의 최대 수명을 지정합니다. 시간은 개월, 주, 일, 시간, 분, 초의 조합으로 지정할 수 있습니다.

예제

CacheMaxExpiry ftp:* 1 month

CacheMaxExpiry http://www.santaclaus.np/* 2 days 12 hours

기본값

CacheMaxExpiry 1 month

CacheMemory — 캐시 RAM 지정

이 지시문을 사용하여 캐시와 연관된 메모리 용량을 지정합니다. 디스크 캐시의 최적 성능을 위해, 캐시 색인을 포함하여 캐시 하부 구조 지원에 대해 64MB의 최소 캐시 메모리 값을 권장합니다. 캐시 크기가 증가되면, 캐시 색인이 증가되고 색인을 저장하기 위한 추가 캐시 메모리가 필요하게 됩니다. 64MB의 캐시 메모리 값은 캐시 하부 구조 지원을 제공하고 최대 6.4GB의 디스크 캐시에 대한 캐시 색인을 저장하기에 충분합니다. 보다 큰 디스크 캐시의 경우, 캐시 메모리는 캐시 크기의 1%여야 합니다.

메모리 캐시를 사용할 경우, 캐시와 캐시 색인에 필요한 메모리 용량을 모두 포함하도록 이 지시문을 설정하십시오.

이 지시문의 최대 권장 값은 1600MB입니다. 이 한계는 32비트 응용프로그램으로서의 Caching Proxy가 최대 2GB 메모리를 사용할 수 있다는 사실에 의해 결정됩니다. 캐시에 필요한 메모리와 루틴 처리에 사용되는 메모리를 더한 용량이 2GB에 근접하거나 초과하는 경우에는, Caching Proxy가 정상적으로 작동하지 않습니다.

용량은 바이트(B), 킬로바이트(K), 메가바이트(M), 기가바이트(G)의 단위로 지정할 수 있습니다.

형식

CacheMemory *amount* {B | K | M | G}

기본값

CacheMemory 64 M

CacheMinHold — 파일을 사용 가능하게 보존하는 기간 지정

이 지시문을 사용하여 만기를 덮어쓸 파일에 대한 URL을 지정하십시오. 일부 사이트는 파일의 수명이 끝나기 전에 파일이 만기되도록 설정하여, 서버가 파일을 자주 요청해야 합니다. CacheMinHold 지시문에 의해 만기된 파일은 다시 요청되기 전에 지정된 기간 동안 캐시에 남아 있게 됩니다. 이 지시문은 여러 번 지정할 수 있습니다.

주: 만기 날짜를 덮어쓰면 캐시에 있는 파일이 유효하지 않게 될 수 있습니다.

예제

CacheMinHold http://www.cachebusters.com/* 1 hour

기본값

없음

CacheNoConnect — 독립형 캐시 모드 지정

이 지시문을 사용하여 프록시 서버가 원격 서버에서 파일을 검색할 것인지를 지정합니다. 기본값(Off)은 서버가 원격 서버에서 파일을 검색하도록 합니다. 값 On은 서버가 독립형 캐시 모드에서 실행하도록 설정합니다. 그러면, 서버가 캐시에 저장된 파일만 돌려보낼 수 있습니다. 일반적으로, 서버가 이 모드에서 실행 중일 때 CacheExpiryCheck 지시문을 Off로 설정할 수도 있습니다.

서버를 시연 목적으로 사용하는 경우, 독립 실행 캐시 모드에서 실행하는 것이 유용할 수 있습니다. 시연 목적으로 사용하려는 파일이 모두 캐시에 저장되어 있다면 네트워크를 연결할 필요가 없습니다.

형식

CacheNoConnect {on | off}

기본값

CacheNoConnect Off

CacheOnly — 템플리트와 일치하는 URL이 있는 파일만 캐시

이 지시문을 사용하여 지정된 템플리트와 일치하는 URL이 있는 파일만 캐시되도록 지정합니다. 이 지시문에 대한 여러 개의 어커런스를 구성 파일에서 사용할 수 있습니다. 각 템플리트에 대한 별개의 지시문을 포함시키십시오. URL 템플리트에는 프로토콜이

있어야 합니다. 이 지시문에 설정된 값이 없는 경우, NoCaching 지시문과 일치하지 않는 임의의 URL을 캐시할 수 있습니다. 또한 CacheOnly 또는 NoCaching 지시문이 구성 파일에 포함되지 않은 경우, 임의의 URL을 캐시할 수 있습니다.

형식

CacheOnly *url_pattern*

예제

CacheOnly http://realstuff/*

기본값

없음

CacheQueries — 물음표(?)를 포함하는 URL에 캐시 응답 지정

이 지시문을 사용하여 조회 요청에 대한 응답이 캐시될 URL을 지정합니다. 값 PUBLIC *url_pattern*이 사용될 경우, 기점 서버에 cache-control: public 헤더가 포함되고 응답을 달리 캐시할 수 있으면, URL에 물음표가 있는 GET 요청에 대한 응답이 캐시됩니다. 값 ALWAYS *url_pattern*이 지정되는 경우, 응답을 다른 방법으로 캐시할 수 있으면 URL에 물음표가 있는 GET 요청에 대한 응답이 캐시됩니다.

이 지시문은 여러 번 지정할 수 있습니다.

CacheQueries {ALWAYS | PUBLIC} *url_pattern*

예제

CacheQueries ALWAYS http://www.hosta.com/*

CacheQueries PUBLIC http://www.hostb.com/*

주: 역호환의 경우, CacheQueries의 이전 구문{ALWAYS | PUBLIC | NEVER}은 다음과 같이 처리됩니다.

- CacheQueries ALWAYS 및 CacheQueries PUBLIC는 CacheQueries ALWAYS * 및 CacheQueries PUBLIC *으로 처리됩니다.
- CacheQueries NEVER는 무시됩니다.
- CacheQueries NEVER 및 CacheQueries *url_pattern*이 모두 지정되면, CacheQueries NEVER는 무시되고 경고 메시지가 발행됩니다.

기본값

없음

CacheRefreshInterval — 캐시된 오브젝트의 유효성을 재확인하기 위한 시간 간격 지정

이 지시문을 사용하여, 캐시된 파일을 변경할지 여부를 판별하기 위해 기점 서버를 확인할 시기를 지정하십시오.

CacheClean이 이 지시문과 유사하게 표시되어도 둘 사이에는 차이가 있습니다. CacheRefreshInterval은 프록시가 파일을 사용하기 전에 파일의 유효성을 재확인하도록 지정하는 반면, CacheClean 지시문은 지정된 기간이 지나면 파일을 캐시에서 제거합니다.

형식

- 다음 형식은 URL 패턴과 일치하는 파일의 새로 고침 간격을 지정합니다.

```
CacheRefreshInterval URL_pattern time_period
```

- 다음 형식은 URL 패턴과 일치하지 않는 파일의 새로 고침 간격을 지정합니다. 새로 고침 간격만 지정합니다.

```
CacheRefreshInterval time_period
```

예제

```
CacheRefreshInterval *.gif 8 hours
CacheRefreshInterval 1 week
```

기본값

```
CacheRefreshInterval 2 weeks
```

CacheRefreshTime — 캐시 에이전트를 시작할 시기 지정

이 지시문을 사용하여 캐시 에이전트를 시작할 시기를 지정합니다. 특정 시간에 캐시 에이전트를 시작할 수 있습니다.

형식

```
CacheRefreshTime HH:MM
```

기본값

```
CacheRefreshTime 03:00
```

CacheTimeMargin — 파일 캐시를 위한 최소 수명 지정

CacheTimeMargin 지시문은 순서대로 캐시되기 위해 필요한 파일의 최소 수명을 지정합니다.

Caching Proxy는 각 파일에 대한 만기 시간을 추정합니다. 파일이 만기되기 전에 파일에 대한 다른 요청이 수신되지 않는 경우, Caching Proxy는 파일을 캐시하기에 파일의 수명이 너무 짧은 것으로 간주합니다. 기본적으로 Caching Proxy는 수명이 10분

미만인 파일은 캐시하지 않습니다. 캐시 용량이 최대치에 근접하지 않은 경우, 이 지시문을 초기값으로 남겨두십시오. 캐시가 용량에 가깝게 근접한 경우, 최소 수명의 값을 증가시켜 보십시오.

형식

`CacheTimeMargin minimum_lifetime`

기본값

`CacheTimeMargin 10 minutes`

주: 이 지시문을 4시간 이상으로 설정하면 캐시의 효율성이 크게 감소됩니다.

CacheUnused — 사용되지 않는 캐시 파일 보존 기간 지정

이 지시문을 사용하여 서버가 지정된 템플리트와 일치하는 URL이 있는 파일을 사용하지 않는 캐시 파일을 보존할 최대 시간을 설정합니다. 템플리트와 일치하는 URL이 있는 사용하지 않는 캐시 파일이 만기 날짜와 상관없이 지정된 기간에 캐시되고 나면, 서버는 이 파일을 삭제합니다. 이 지시문에 대한 여러 개의 어커런스를 구성 파일에 포함할 수 있습니다. 각 템플리트에 대한 별개의 지시문을 포함시키십시오. URL 템플리트에는 프로토콜이 있어야 합니다. 개월, 주, 일, 시간을 조합한 시간값을 지정하십시오.

형식

`CacheUnused url_template time_length`

예제

```
CacheUnused ftp:* 3 weeks
CacheUnused gopher:* 3 days 12 hours
CacheUnused * 4 weeks
```

기본값

```
CacheUnused ftp:* 3 days
CacheUnused gopher:* 12 hours
CacheUnused http:* 2 days
```

Caching — 프록시 캐시 사용 가능

이 지시문을 사용하여 파일의 캐시를 사용할 수 있도록 합니다. 캐시가 on으로 설정되면, 프록시 서버는 로컬 캐시의 다른 서버에서 검색한 파일을 저장합니다. 프록시 서버는 이제 다른 서버에서 검색하지 않고도 동일한 파일에 대한 후속 요청에 응답합니다.

형식

`Caching {on | off}`

기본값

`Caching On`

주: Caching 지시문을 변경하면 서버를 직접 정지시킨 다음 재시작해야 합니다 (15 페이지의 제 5 장『Caching Proxy 시작 및 정지』참조).

CompressAge — 로그 압축 시기 지정

이 지시문을 사용하여 로그를 압축한 후 유효 기간을 지정합니다. 로그가 CompressAge 값 세트보다 이전 것이면 압축됩니다. CompressAge가 0이면 로그는 더 이상 압축되지 않습니다. 당일이나 그 전날에 대한 로그는 압축되지 않습니다.

형식

CompressAge *number_of_days*

기본값

CompressAge 1

관련 지시문

- 214 페이지의『CompressDeleteAge — 로그 삭제 시기 지정』
- 『CompressCommand — 압축 명령 및 매개변수 지정』
- 251 페이지의『LogArchive — 로그 보존 작동 지정』
- 260 페이지의『Midnight — 로그 보존에 사용되는 API 플러그인 지정』
- 289 페이지의『PurgeAge — 로그의 유효 기간 한계 지정』
- 289 페이지의『PurgeSize — 로그 보존 크기의 한계 지정』

CompressCommand — 압축 명령 및 매개변수 지정

이 지시문을 사용하여 로그를 압축하는 데 사용될 압축 유틸리티를 식별하고 이 유틸리티로 매개변수를 전달할 명령을 작성하십시오. 보존된 로그에 대한 경로를 포함시키십시오.

압축 유틸리티는 해당 시스템의 경로에 나열된 디렉토리에 설치되어야 합니다.

형식

CompressCommand *command*

command

단일 행에 입력된 사용하려는 명령 및 매개변수를 포함합니다. 일반적으로 매개변수는 %%LOGFILES%% 및 %%DATE%%를 포함합니다.

%%LOGFILES%%

특정 %%DATE%%에 사용 가능한 로그 파일의 목록을 지정합니다.

%%DATE%%

로그 파일에 데이터 스탬프를 지정합니다.

예제

- **Linux 및 UNIX:**

```
CompressCommand tar -cf /logarchs/log%%DATE%%.tar %%LOGFILES%% ;  
gzip /logarchs/log%%DATE%%.tar  
CompressCommand tar -cf /logarchs/log%%DATE%%.tar %%LOGFILES%% ;  
compress /logarchs/log%%DATE%%.tar  
CompressCommand zip -q /logarchs/log%%DATE%%.zip %%LOGFILES%%
```

주: 명령 및 모든 매개변수는 단일 행에 입력되어야 합니다. 앞의 예제에서 처음 두 개의 명령 예제는 읽기 가능하도록 분리됩니다.

- **Windows:**

```
CompressCommand pkzip -q d:\logarchs\log%%DATE%%.tar %%LOGFILES%%
```

기본값

없음

관련 지시문

- 213 페이지의 『CompressAge — 로그 압축 시기 지정』
- 『CompressDeleteAge — 로그 삭제 시기 지정』
- 251 페이지의 『LogArchive — 로그 보존 작동 지정』
- 260 페이지의 『Midnight — 로그 보존에 사용되는 API 플러그인 지정』
- 289 페이지의 『PurgeAge — 로그의 유효 기간 한계 지정』
- 289 페이지의 『PurgeSize — 로그 보존 크기의 한계 지정』

CompressDeleteAge — 로그 삭제 시기 지정

이 지시문을 사용하여 압축한 후 로그를 삭제할 시기를 지정합니다. 로그가 CompressDeleteAge 값에 대해 설정한 일 수보다 이전의 것이면 삭제됩니다. CompressDeleteAge가 0으로 설정되거나 값이 CompressAge 지시문에 설정된 값보다 작은 경우, 로그가 삭제되지 않습니다.

주: 압축 플러그인은 당일 또는 전일의 로그는 삭제하지 않습니다.

형식

```
CompressDeleteAge number_of_days
```

기본값

```
CompressDeleteAge 7
```

관련 지시문

- 213 페이지의 『CompressAge — 로그 압축 시기 지정』
- 213 페이지의 『CompressCommand — 압축 명령 및 매개변수 지정』

- 251 페이지의 『LogArchive — 로그 보존 작동 지정』
- 260 페이지의 『Midnight — 로그 보존에 사용되는 API 플러그인 지정』
- 289 페이지의 『PurgeAge — 로그의 유효 기간 한계 지정』
- 289 페이지의 『PurgeSize — 로그 보존 크기의 한계 지정』

CompressionFilterAddContentType — 압축하려는 HTTP 응답의 콘텐츠 유형을 지정

이 지시문을 사용하여 압축하려는 HTTP 응답의 콘텐츠 유형을 지정합니다.

HTTP 응답을 압축하면, 네트워크 로드를 줄이고, 프록시 서버 성능을 향상시킵니다. 압축 필터 기능을 사용 가능하게 하고, 브라우저가 HTTP 압축을 지원하며, HTTP 응답이 현재 압축되지 않은 경우, Caching Proxy는 HTTP 응답을 압축하고, 압축된 콘텐츠를 브라우저에 리턴합니다.

예제

다음 2개의 지시문을 ibmproxy.conf 파일에 추가하여 압축 필터 기능을 사용할 수 있습니다.

- **HP-UX 시스템의 경우:**

```
CompressionFilterEnable /opt/ibm/edge/cp/lib/mod_z.sl
CompressionFilterAddContentType type-1[,type-n]
```

- **기타 UNIX® 시스템 및 Linux 시스템의 경우:**

```
CompressionFilterEnable /opt/ibm/edge/cp/lib/mod_z.so
CompressionFilterAddContentType type-1[,type-n]
```

- **Windows 시스템의 경우:**

```
CompressionFilterEnable C:\Progra~1\IBM\edge\cp\Bin\mod_z.dll
CompressionFilterAddContentType type-1[,type-n]
```

CompressionFilterEnable 지시문에서 참조된 mod_z 라이브러리는 zlib1.1.4의 동적 버전입니다.

변수 유형 -n은 콘텐츠 유형 헤더에 대한 유효값입니다. 예를 들어, text/html 또는 image/bmp입니다.

주: HTTP 응답의 특정 유형에 대한 콘텐츠(예: JPEG 이미지 또는 비디오 스트림)는 이미 응용프로그램으로 압축되어 있습니다. 그러므로, 이 기능을 사용하여 압축하지 말아야 합니다.

기본값

없음

CompressionFilterEnable — 압축 필터를 사용하여 HTTP 응답을 압축

이 지시문을 사용하여 백엔드 서버 또는 프록시 서버 캐시로부터의 HTTP 응답을 압축하는 압축 필터를 사용 가능하게 합니다.

이 지시문 사용 방법에 대한 예는 215 페이지의 『CompressionFilterAddContentType — 압축하려는 HTTP 응답의 콘텐츠 유형을 지정』을 참조하십시오.

기본값

없음

ConfigFile — 추가 구성 파일의 이름 지정

이 지시문을 사용하여 추가 구성 파일의 이름 및 위치를 지정합니다. 지정된 구성 파일에 있는 지시문은 현재 구성 파일 다음에 처리됩니다.

주: 추가 구성 파일은 캐시 에이전트가 이 파일을 읽을 수 있도록 nobody 사용자의 Read로 설정된 권한을 가지는지 확인하십시오.

예제

- **Linux 및 UNIX:** ConfigFile /etc/rca.conf
- **Windows:** ConfigFile c:\WINNT\rca.conf

기본값

없음

ConnThreads — 연결 관리에 사용되는 연결 스레드의 수를 지정

이 지시문을 사용하여 연결 관리에 사용되는 연결 스레드의 수를 지정합니다.

형식

ConnThreads *number*

기본값

ConnThreads 5

관련 지시문

- 256 페이지의 『MaxActiveThreads — 최대 활성 스레드 수 지정』

ContinueCaching — 캐시에 필요한 파일 수 지정

이 지시문을 사용하여 클라이언트 연결이 종료되더라도 Caching Proxy에 전송되어야 할 요청 파일의 수를 지정하여 캐시 파일 작성을 완료합니다. 이 변수에 대한 유효값은 0 - 100 범위의 정수입니다.

예를 들어, ContinueCaching 75가 지정된 경우, Caching Proxy가 클라이언트 연결이 종료된 것을 발견하기 전에 파일의 75% 이상이 이미 전송되었으면, Caching Proxy는 콘텐츠 서버에서 파일을 계속 전송하여 캐시 파일을 생성합니다.

형식

ContinueCaching *percentage*

기본값

ContinueCaching 75

DefinePicsRule — 콘텐츠 필터링 규칙 공급

이 지시문을 사용하여 등급 서비스 정보를 포함한 콘텐츠의 URL을 필터하기 위해 필수 정보를 프록시에 공급합니다. 이 지시문을 여러 번 지정할 수 있습니다.

형식

DefinePicsRule "*filter_name*" {

기본값

DefinePicsRule "RSAC Example" {

DefProt — 템플리트와 일치하는 요청에 대한 기본 보호 설정 지정

이 지시문을 사용하여 기본 보호 설정을 템플리트와 일치하는 요청과 연관시킵니다.

주: 보호를 올바르게 작동시키려면, DefProt 및 Protect 지시문을 구성 파일에 있는 Pass 또는 Exec 지시문 앞에 위치시켜야 합니다.

형식

DefProt *request_template* *setup_name* [FOR *server_IP_address* | *host_name*]

request_template

기본 보호 설정과 연관시키려는 요청 템플리트를 지정합니다. 서버는 템플리트에 대한 수신 클라이언트 요청을 비교하여, 일치되는 것이 있으면 보호 설정과 연관시킵니다.

요청이 후속 보호 지시문의 템플리트와 일치하지 않으면, 보호는 템플리트와 일치하는 요청에 대해 활성화되지 않습니다. DefProt과 함께 Protect 지시문을 사용하는 방법에 대한 설명은 274 페이지의 『Protect — 템플리트와 일치하는 요청에 대한 보호 설정 활성화』를 참조하십시오.

setup

*request_template*와 일치하는 요청과 연관시킬 구성 파일에 정의된 이름 지정 보호를 설정합니다. 보호 설정은 보호 부 지시문으로 정의됩니다. 이 매개변수는 다음 세 가지 형식 중 하나를 취할 수 있습니다.

- 보호 부 지시문을 포함하는 별개의 파일을 지정하는 전체 경로 및 파일 이름
- 보호 지시문에 이전에 정의된 이름과 일치하는 보호 설정 레이블 이름. 보호 지시문은 보호 부 지시문을 포함합니다.
- 실제 보호 부 지시문. 부 지시문은 중괄호({})로 묶어야 합니다. 왼쪽 중괄호 문자는 DefProt 지시문과 동일한 행의 마지막 문자여야 합니다. 각 부 지시문은 부 지시문 행에 있어야 합니다. 오른쪽 중괄호 문자는 마지막 부 지시문 행 다음의 부 지시문 행에 있어야 합니다. 중괄호 사이에 설명 행이 들어갈 수 없습니다. 보호 부 지시문에 대한 설명은 다음을 참조하십시오.
 - 280 페이지의 『AuthType — 인증 유형 지정』
 - 280 페이지의 『DeleteMask — 파일 삭제를 허용한 사용자 이름, 그룹 및 주소 지정』
 - 281 페이지의 『GetMask — 파일 가져오기를 허용한 사용자 이름, 그룹 및 주소 지정』
 - 281 페이지의 『GroupFile — 연관된 그룹 파일의 위치 지정』
 - 281 페이지의 『Mask — HTTP 요청을 허용한 사용자 이름, 그룹 및 주소 지정』
 - 281 페이지의 『PasswdFile — 연관된 암호 파일의 위치 지정』
 - 282 페이지의 『PostMask — 파일 게시를 허용한 사용자 이름, 그룹 및 주소 지정』
 - 282 페이지의 『PutMask — 파일 넣기를 허용한 사용자 이름, 그룹 및 주소 지정』
 - 282 페이지의 『ServerID — 암호 파일과 연관시킬 이름 지정』

[FOR *Server_IP_address* | *host_name*]

여러 IP 주소나 가상 호스트를 사용하고 있다면, 이 매개변수를 사용하여 IP 주소나 호스트 이름을 지정하십시오. 서버는 이 IP 주소나 호스트의 서버에 도달하는 요청이나 이 호스트에 대해서만 지시문을 사용합니다. IP 주소는 요청하는 클라이언트의 주소가 아니라 서버 네트워크 연결의 주소입니다.

IP 주소(예: FOR 240.146.167.72) 또는 호스트 이름(예: FOR hostA.bcd.com)을 지정할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수가 없으면, 서버는 요청이 들어오는 IP 주소나 URL의 호스트 이름과 상관없이 모든 요청에 대해 지시문을 사용합니다.

주:

1. *setup* 매개변수가 경로 및 파일 이름의 양식에 지정되거나 보호 설정 레이블 일 때에만 이 매개변수를 사용할 수 있습니다. 이 매개변수는 중괄호 안의 실제 보호 부 지시문 양식으로 지정된 *setup* 매개변수와 함께 사용할 수 없습니다.

2. 이 매개변수를 사용하려면, FOR나 (공백 없는) 기타 문자 스트링을 *setup* 매개변수와 *IP_address* 또는 *host_name* 사이에 넣어야 합니다.

서버 IP 주소로 와일드 카드 문자를 지정할 수 없습니다.

주: 이 지시문은 한 행에 입력되어야 합니다.

예제

- 다음 예제에서 보호 부 지시문이 들어 있는 분리된 파일을 식별합니다.

```
DefProt /secret/* /server/protect/setup1.acc
```

- 다음 예제에서 레이블 이름을 사용하여 보호 부 지시문을 지시합니다. 레이블 이름은 보호 지시문의 레이블 이름과 일치해야 합니다. 보호 지시문은 DefProt 지시문 앞에 와야 합니다.

```
DefProt /secret/* SECRET-PROT
```

- 다음 예제는 DefProt 지시문의 일부로 보호부 지시문을 포함합니다.

```
DefProt {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/etc/WWW/restrict.password
    GroupFile /docs/etc/WWW/restrict.group
    GetMask authors
    PutMask authors
}
```

- 다음 예제에서는 선택적 IP 주소 매개변수를 사용합니다. 서버가 /secret/로 시작하는 요청을 수신하면, 다른 기본 보호 설정과 요청이 들어오는 네트워크 연결의 IP 주소에 기초하여 요청을 연관시킵니다. 0.67.106.79로 들어오는 요청의 경우, 서버는 이 요청을 CustomerA-PROT 레이블이 있는 보호 지시문에 정의된 기본 보호와 연관시킵니다. 0.83.100.45로 들어오는 요청의 경우, 서버는 이 요청을 CustomerB-PROT 레이블이 있는 보호 지시문에 정의된 기본 보호와 연관시킵니다.

```
DefProt /secret/* CustomerA-PROT 0.67.106.79
DefProt /secret/* CustomerB-PROT 0.83.100.45
```

- 다음 예제에서는 선택적인 호스트 이름 매개변수를 사용합니다. 서버가 /secret/로 시작하는 요청을 수신하면, 다른 기본 보호 설정을 URL의 호스트 이름에 기초하여 요청과 연관시킵니다. hostA로 들어오는 요청의 경우, 서버는 이 요청을 CustomerA-PROT 레이블이 있는 보호 지시문에 정의된 기본 보호와 연관시킵니다. hostB로 들어오는 요청의 경우, 서버는 이 요청을 CustomerB-PROT 레이블이 있는 보호 지시문에 정의된 기본 보호와 연관시킵니다.

```
DefProt /secret/* CustomerA-PROT hostA.bcd.com
DefProt /secret/* CustomerB-PROT hostB.bcd.com
```

기본값

없음

DelayPeriod — 요청 간 일시정지 지정

이 지시문을 사용하여 캐시 에이전트가 대상 서버로 요청을 전송하는 사이에 대기할지를 지정합니다. 요청 사이에 연기를 지정하면, 프록시 시스템과 네트워크 연결에 대한 로드뿐만 아니라 대상 서버의 로드도 줄일 수 있습니다. 연기를 지정하지 않으면 캐시 에이전트가 최대 속도로 실행됩니다. 인터넷 연결이 느린 경우, 네트워크를 최대로 사용하려면 연기 기간을 지정하지 마십시오.

주: 인터넷 연결 속도가 128kbps보다 빠른 경우, DelayPeriod를 On으로 설정하여 새로 고친 사이트에 지나치게 많은 요청이 너무 빨리 전송되지 않게 하십시오.

형식

DelayPeriod {on | off}

기본값

DelayPeriod On

DelveAcrossHosts — 도메인을 통한 캐시 지정

이 지시문을 사용하여 캐시 에이전트가 호스트 사이의 하이퍼텍스트 연결을 따르는지의 여부를 지정합니다. 캐시된 URL에 다른 서버로의 연결이 있는 경우, 서버가 연결을 무시하거나 연결을 따를 수 있습니다. DelveInto 지시문이 never로 설정된 경우, 이 지시문은 적용되지 않습니다.

형식

DelveAcrossHosts {on | off}

기본값

DelveAcrossHosts Off

DelveDepth — 캐시하는 동안 연결을 따르는 범위 지정

이 지시문을 사용하여 캐시에 로드할 페이지를 탐색할 때 따를 연결 레벨의 수를 지정합니다. DelveInto 지시문이 never로 설정된 경우, 이 지시문은 적용되지 않습니다.

형식

DelveDepth *number_of_levels*

기본값

DelveDepth 2

DelveInto — 캐시 에이전트가 연결을 따르는지를 지정

이 지시문을 사용하여 캐시 에이전트가 캐시된 URL에서 연결 페이지를 로드하는지 여부를 지정합니다.

형식

DelveInto {always | never | admin | topn}

always

캐시 에이전트가 이전에 캐시된 모든 URL에서의 연결을 따릅니다.

never

캐시 에이전트가 URL의 모든 연결을 무시합니다.

admin

캐시 에이전트가 LoadURL 지시문에서 지정된 URL의 연결만 따릅니다.

topn

캐시 에이전트가 캐시에서 검색된 빈도가 많은 연결만 따릅니다.

기본값

DelveInto always

DirBackgroundImage — 디렉토리 목록에 백그라운드 이미지 지정

이 지시문을 사용하여 프록시 서버가 생성한 디렉토리 목록에 백그라운드 이미지를 적용합니다. 프록시 서버가 FTP 사이트 찾아보기에 사용될 때 디렉토리 목록이 생성됩니다.

백그라운드 이미지의 절대 경로를 지정합니다. 이미지가 다른 서버에 있을 경우, 백그라운드 이미지는 전체 URL로 지정되어야 합니다. 백그라운드 이미지가 지정되지 않으면, 일반 백색 백그라운드가 사용됩니다.

형식

DirBackgroundImage */path/file*

예제

DirBackgroundImage /images/corplogo.png

DirBackgroundimage http://www.somehost.com/graphics/embossed.gif

기본값

없음

DirShowBytes — 디렉토리 목록에 작은 파일에 대한 바이트 수 표시

이 지시문을 사용하여, 디렉토리 목록이 1KB보다 작은 파일에 대한 정확한 바이트 수를 나타낼지 여부를 지정하십시오. Off 값은 디렉토리 목록이 1KB 또는 그 이하의 모든 파일에 대해 1KB 크기를 표시한다는 의미입니다.

형식

DirShowBytes {on | off}

기본값

DirShowBytes Off

DirShowCase — 디렉토리 목록에 파일을 분류할 때 대소문자 사용

이 지시문을 사용하여 디렉토리 목록이 파일 이름을 분류할 때 대문자와 소문자를 구분해야 하는지 여부를 지정합니다.

On 값은 파일 목록에서 소문자 앞에 대문자가 온다는 의미입니다.

형식

DirShowCase {on | off}

기본값

DirShowCase On

DirShowDate — 디렉토리 목록에 최종 변경 날짜 표시

이 지시문을 사용하여 디렉토리 목록이 각 파일이 최종 수정된 날짜를 포함하는지 여부를 지정합니다.

형식

DirShowDate {on | off}

기본값

DirShowDate On

DirShowDescription — 디렉토리 목록에 파일에 대한 설명 표시

이 지시문을 사용하여 디렉토리 목록에 HTML 파일에 대한 설명이 있는지 여부를 지정합니다. 설명은 파일의 HTML <title> 태그에서 볼 수 있습니다.

MIME 유형을 판별할 수 있으면, FTP 디렉토리 목록에 대한 설명은 파일의 MIME 유형을 표시합니다.

형식

DirShowDescription {on | off}

기본값

DirShowDescription On

DirShowHidden — 디렉토리 목록에 숨겨진 파일 표시

이 지시문을 사용하여 디렉토리 목록에 디렉토리에 숨겨진 파일이 있는지 여부를 지정합니다. 서버는 마침표(.)로 시작하는 이름을 가진 파일을 숨겨진 파일로 간주합니다.

형식

`DirShowHidden {on | off}`

기본값

`DirShowHidden On`

DirShowIcons — 디렉토리 목록에 아이콘 표시

이 지시문을 사용하여 서버가 디렉토리 목록에 아이콘을 포함하는지 여부를 지정합니다. 아이콘은 목록에 있는 파일의 콘텐츠 유형에 대한 그래픽 표현을 제공하는 데 사용할 수 있습니다. 아이콘은 `AddBlankIcon`, `AddDirIcon`, `AddIcon`, `AddParentIcon` 및 `AddUnknownIcon` 지시문으로 정의됩니다.

형식

`DirShowIcons {on | off}`

기본값

`DirShowIcons On`

DirShowMaxDescrLength — 디렉토리 목록에서 설명의 최대 길이 지정

이 지시문을 사용하여 디렉토리 목록의 설명 필드에 표시할 문자의 최대수를 설정합니다.

형식

`DirShowMaxDescrLength number_of_characters`

기본값

`DirShowMaxDescrLength 25`

DirShowMaxLength — 디렉토리 목록에서 파일 이름의 최대 길이 지정

이 지시문을 사용하여 디렉토리 목록의 파일 이름에 사용되는 문자의 최대수를 설정합니다.

형식

`DirShowMaxDescrLength number_of_characters`

기본값

`DirShowMaxLength 25`

DirShowMinLength — 디렉토리 목록에 파일 이름의 최소 길이 지정

이 지시문을 사용하여 디렉토리 목록의 파일 이름에 항상 예약될 문자의 최소 수를 설정합니다. 디렉토리의 파일 이름은 이 숫자를 초과할 수 있습니다. 그러나 파일 이름은 DirShowMaxLength 지시문에 지정된 숫자보다 길면 안됩니다.

형식

DirShowMinLength *number_of_characters*

기본값

DirShowMinLength 15

DirShowSize — 디렉토리 목록에 파일 크기 표시

이 지시문을 사용하여 디렉토리 목록이 각 파일의 크기를 포함해야 하는지 여부를 지정합니다.

형식

DirShowSize {on | off}

기본값

DirShowSize On

Disable — HTTP 메소드 사용 불가능

이 지시문을 사용하여 서버에서 승인하지 않는 HTTP 메소드를 지정합니다. 서버가 거부(reject)할 각 메소드에 별개의 Disable 지시문을 입력하십시오.

기본 구성 파일에서 GET, HEAD, OPTIONS, POST 및 TRACE 메소드는 사용 가능하며 기타 지원되는 모든 HTTP 메소드는 사용 불가능합니다. 현재 사용 가능한 메소드를 사용 불가능하게 하려면, Enable 지시문에서 이 메소드를 삭제하고 Disable 지시문으로 추가합니다.

형식

Disable *method*

주: 구성 및 관리 양식은 POST 메소드를 사용하여 서버 구성을 갱신합니다. POST 메소드를 사용 불가능하게 하면 구성 및 관리 양식을 사용할 수 없습니다.

기본값

Disable PUT
Disable DELETE
Disable CONNECT

DisInheritEnv — CGI 프로그램이 계승하지 않는 환경 변수 지정

이 지시문을 사용하여(CGI 처리에 고유한 CGI 환경 변수가 아닌) CGI 프로그램에서 계승하지 않을 환경 변수를 지정합니다.

기본적으로 모든 환경 변수는 CGI 프로그램이 계승합니다. 이 지시문을 사용하여, 개별 환경 변수가 계승되지 못하도록 제외할 수 있습니다.

형식

DisInheritEnv *environment_variable*

예제

```
DisInheritEnv PATH
DisInheritEnv LANG
```

이 예제에서는 PATH 및 LANG을 제외한 모든 환경 변수가 CGI 프로그램에 의해 계승됩니다.

기본값

없음

DNS-Lookup — 서버가 클라이언트 호스트 이름을 조회할지 여부 지정

이 지시문을 사용하여 서버가 요청 중인 클라이언트의 호스트 이름을 조회할지 여부를 지정하십시오.

형식

DNS-Lookup {on | off}

사용하는 값은 서버의 작동 방법에 대한 다음 사항에 영향을 줍니다.

- 서버의 성능. 기본값 Off를 사용하면, 호스트 이름 조회를 수행하는 자원을 사용하지 않기 때문에 서버의 성능이 개선되고 응답 시간이 줄어듭니다.
- 로그 파일에 기록할 때 서버가 클라이언트에 대해 기록하는 정보.
 - Off—IP 주소로 클라이언트를 식별합니다.
 - On—호스트 이름으로 클라이언트를 식별합니다.
- 보호 설정, 서버 그룹 파일 및 ACL(액세스 제어 목록) 파일의 주소 템플릿에 호스트 이름을 사용할 수 있는지 여부.
 - Off—주소 템플릿에서 호스트 이름을 사용할 수 없으며, IP 주소를 사용해야 합니다.
 - On—주소 템플릿에서 호스트 이름을 사용할 수 없으며, IP 주소도 사용할 수 없습니다.

주: 보호 규칙에서 도메인 이름을 사용하려면 DNS-Lookup 지시문을 On으로 설정해야 합니다.

기본값

DNS-Lookup Off

Enable — HTTP 메소드 사용 가능

이 지시문을 사용하여 서버에서 승인한 HTTP 메소드를 지정합니다.

HTTP 메소드를 필요한 만큼 사용 가능하게 할 수 있습니다. 서버에서 승인할 각 메소드에 대해서 별도의 Enable 지시문을 입력하십시오.

형식

Enable *method*

특정 URL에 대한 Service 지시문이 없으면, Enable 지시문을 사용하여 HTTP 메소드에 대해 사용자 정의된 프로그래밍을 수행할 수 있습니다. 이 지시문에 지정한 프로그램은 해당 메소드에 대한 표준 처리를 덮어씁니다.

Enable *method* /path/fileDLL:function_name

Enable CONNECT 메소드에 대한 형식 및 사용 가능한 옵션에 대한 정보는 128 페이지의 『SSL 터널링 구성』을 참조하십시오.

기본값

Enable GET
Enable HEAD
Enable POST
Enable TRACE
Enable OPTIONS

EnableTcpNodelay — TCP NODELAY 소켓 옵션을 사용 가능하게 함

이 지시문을 사용하여 TCP NODELAY 소켓 옵션을 사용 가능하게 합니다.

EnableTcpNodelay 지시문은 SSL 핸드셰이크 또는 짧은 HTTP 응답과 같은 소량의 IP 패킷을 Caching Proxy 및 클라이언트를 통해 전송하는 경우, 성능을 향상시킵니다. 기본적으로, TCP NODELAY 옵션은 모든 소켓에 대해 사용 가능합니다.

형식

EnableTcpNodelay {All | HTTP | HTTPS | None}

기본값

EnableTcpNodelay All

Error — 오류 단계 사용자 정의

이 지시문을 사용하여 오류 단계에서 서버가 호출할 사용자 정의된 응용프로그램 기능을 지정합니다. 이 코드는 오류가 발생할 때 사용자 정의된 오류 루틴을 제공하기 위해 실행됩니다.

형식

Error *request_template* */path/file:function_name*

request_template

응용프로그램 기능이 호출되는지 여부를 추가적으로 판별하는 요청에 대한 템플리트를 지정합니다. 스펙은 프로토콜, 도메인, 호스트를 포함할 수 있고, 앞에 슬래시(/)가 붙을 수 있으며, 별표(*)를 와일드 카드로 사용할 수 있습니다. 예를 들어, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /* 및 *는 모두 유효합니다.

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

예제

```
Error    /index.html /ics/api/bin/icsext05.so:error_rtns
```

기본값

없음

ErrorLog — 서버 오류를 로그할 파일 지정

이 지시문을 사용하여, 서버에서 내부 오류를 로그하려는 경로 및 파일의 이름을 지정하십시오.

주: 사용자 ID, 그룹 ID, 또는 로그 디렉토리 경로에 대한 서버 기본값을 변경한 경우, 새 디렉토리를 작성하고 디렉토리의 권한 및 소유권을 갱신하십시오. 서버에서 사용자 정의된 로그 디렉토리에 정보를 기록하도록 하려면, 해당 디렉토리에 대한 권한을 755로 설정하고 사용자 정의된 서버 사용자 ID를 소유자로 설정하십시오. 예를 들어, 서버의 사용자 ID를 기본값에서 jdoe로 변경하고, 기본 로그 디렉토리를 server_root/account로 변경한 경우, server_root/account 디렉토리는 755의 권한을 가져야 하며 jdoe가 소유해야 합니다.

서버가 실행 중이면 매일 자정에 새 로그 파일을 시작합니다. 그렇지 않으면, 서버는 해당 날짜에 로그 파일을 처음 시작할 때 새 로그 파일을 시작합니다. 파일을 작성할 때, 서버는 지정한 파일 이름을 사용하고 날짜 접미부를 추가합니다. 날짜 접미부는

Mmmddyyyy 형식입니다. 여기서 *Mmm*은 월의 처음 세 글자를 나타내고, *dd*는 해당 월의 일을 나타내며, *yyyy*는 연도를 나타냅니다.

형식

ErrorLog */path/logs_directory/file_name*

기본값

- **Linux** 및 **UNIX** 시스템: ErrorLog */opt/ibm/edge/cp/server_root/logs/error*
- **Windows** 시스템: ErrorLog *drive:\Program Files\IBM\edge\cp\logs\error*

ErrorPage — 특정 오류 조건에 대해 사용자 정의된 메시지 지정

이 지시문을 사용하여 서버에 특정 오류 조건이 발생할 때 요청하는 클라이언트에게 전송하는 파일의 이름을 지정합니다. 구성 파일 *ibmproxy.conf*는 오류 키워드를 오류 메시지 파일과 연관시키는 ErrorPage 지시문을 제공합니다.

오류 메시지를 사용자 정의하기 위해, ErrorPage 지시문을 수정하여 오류 키워드를 다른 파일과 연관시키거나 제공된 오류 메시지 파일을 변경할 수 있습니다. 예를 들어, 문제의 원인에 대한 자세한 정보를 포함하거나 문제를 고칠 수 있는 해결책을 제시하도록 메시지를 변경할 수 있습니다. 내부 네트워크의 경우, 사용자가 호출할 문의처를 제공할 수 있습니다.

ErrorPage 지시문은 구성 파일 내의 어느 곳에나 배치할 수 있습니다. 오류가 발생할 때 파일은 구성 파일에 정의된 맵핑 규칙에 따라 처리됩니다. 따라서 전송하려는 파일은 Fail, Map, NameTrans, Pass, Redirect 및 Service 지시문에 의해 정의된 대로 맵핑 규칙을 통해 도달할 수 있는 위치에 있어야 합니다. 최소한 서버가 오류 메시지 파일을 전달하도록 허용하는 Pass 지시문이 필요합니다.

형식

ErrorPage *keyword /path/filename.html*

keyword

오류 조건과 연관된 키워드 중 하나를 지정합니다. *ibmproxy.conf* 파일의 ErrorPage 지시문에 키워드가 나열됩니다. 키워드는 변경할 수 없습니다.

/path/filename.html

웹에서 클라이언트가 표시하도록 오류 파일의 완전한 웹 이름을 지정합니다. 기본 오류 메시지 파일은 */HTML/errorpages/*에 있습니다.

예제

ErrorPage scriptstart */HTML/errorpages/scriptstart.htmls*

이 예제에서 scriptstart 조건이 발생할 때, 서버는 */HTML/errorpages/* 디렉토리에 있는 *scriptstart.htmls* 파일을 클라이언트로 전송합니다.

다음의 HTML 텍스트는 파일에 포함될 수 있는 내용의 예제입니다.

```
<HTML>
<HEAD>
<TITLE>Message for SCRIPTSTART condition</TITLE>
</HEAD>
<BODY>
The CGI program could not be started.
<P>
<A HREF="mailto:admin@websvr.com">Notify the administrator</A>
of this problem.
</BODY>
</HTML>
```

서버 구성 파일에서 위 경로와 일치하는 지시문이 `PASS /* /wwwhome/*`면, 이 메시지 파일에 대한 전체 경로는 `/wwwhome/HTML/errorpages/scriptstart.htmls`입니다.

서버가 리턴한 오류 메시지 사용자 정의

각 오류 조건은 키워드로 식별됩니다. 사용자 정의하려는 오류 메시지를 결정하려면, 우선 `/HTML/errorpages`를 찾을 수 있는 Caching Proxy와 함께 제공된 오류 메시지 파일을 검토하십시오. 오류 페이지에는 오류 번호, 기본 메시지, 원인 설명 및 적절한 복구 조치 등이 있습니다.

그런 다음, 다음 중 하나를 수행하여 오류 메시지를 변경하십시오.

- 기존의 HTML이나 HTMLS 파일을 수정하거나(우선 백업 사본 작성) 원하는 텍스트와 함께 새 HTML이나 HTMLS 파일을 작성하십시오. HTML 편집기나 ASCII 편집기를 사용할 수 있습니다. 정보 포함을 사용하려면 HTMLS 파일을 사용해야 합니다.
- 다른 이름을 가진(또는 다른 경로로 된) 오류 파일을 작성했으면, 그 파일을 지시하기 위해 해당 키워드에 대한 ErrorPage 지시문을 수정하십시오.

오류 조건, 원인 및 기본 메시지

모든 오류 키워드 및 기본 오류 메시지 파일은 ErrorPage 지시문 섹션의 `ibmproxy.conf`에 나열됩니다. 오류 메시지 파일에는 오류 메시지 번호, 키워드, 기본 메시지, 설명, 사용자 응답(조치)이 있습니다.

기본값

다수의 기본값이 `ibmproxy.conf` 파일에 포함됩니다.

오류 조건에 대한 ErrorPage 지시문을 수정하지 않으면, 이 조건에 대한 서버의 기본 오류 페이지가 전송됩니다.

EventLog — 이벤트 로그 파일에 대한 경로 지정

이 지시문을 사용하여, 이벤트 로그 경로 및 파일 이름을 지정하십시오. 이벤트 로그는 캐시 자체에 대한 정보 메시지를 캡처합니다.

주: 사용자 ID, 그룹 ID 또는 로그 디렉토리 경로에 대한 서버 기본값을 변경한 경우, 새 디렉토리를 작성하고 디렉토리의 권한 및 소유권을 갱신하십시오. 서버에서 사용자 정의된 로그 디렉토리에 정보를 기록하도록 하려면, 해당 디렉토리에 대한 권한을 755로 설정하고 사용자 정의된 서버 사용자 ID를 소유자로 설정하십시오. 예를 들어, 서버의 사용자 ID를 기본값에서 jdoe로 변경하고, 기본 로그 디렉토리를 server_root/account로 변경한 경우, server_root/account 디렉토리는 755의 권한을 가져야 하며 jdoe가 소유해야 합니다.

서버가 실행 중이면 매일 자정에 새 로그 파일을 시작합니다. 그렇지 않으면, 서버는 해당 날짜에 로그 파일을 처음 시작할 때 새 로그 파일을 시작합니다. 파일을 작성할 때, 서버는 지정한 파일 이름을 사용하고 날짜 접미부를 추가합니다. 날짜 접미부는 *Mmmddyyyy* 형식입니다. 여기서 *Mmm*은 월의 처음 세 글자를 나타내고, *dd*는 해당 월의 일을 나타내며, *yyyy*는 연도를 나타냅니다.

형식

EventLog /path/logs_directory/file_name

기본값

- **Linux 및 UNIX 시스템:** EventLog /opt/ibm/edge/cp/server_root/logs/event
- **Windows 시스템:** EventLog drive:\Program Files\IBM\edge\cp\logs\event

Exec — 요청을 일치시키기 위한 CGI 프로그램 실행

이 지시문을 사용하여 CGI 프로그램을 실행함으로써 승인하고 응답하려는 요청에 대한 템플릿을 지정합니다. 일단 요청이 Exec 지시문의 템플릿과 일치하면, 요청은 후속 지시문의 요청 템플릿과 비교되지 않습니다.

형식

Exec request_template program_path [Server_IP_address | host_name]

request_template

서버에서 CGI 프로그램을 실행함으로써 승인하고 응답하는 요청에 대한 템플릿을 지정합니다.

*request-template*와 *program-path* 모두에서 별표(*)를 와일드 카드로 사용해야 합니다. *request_template* 와일드 카드와 일치하는 요청 부분은 CGI 프로그램이 있는 파일의 이름으로 시작해야 합니다.

요청은 또한 PATH_INFO 환경 변수의 CGI 프로그램으로 전달된 추가 데이터를 포함할 수 있습니다. 추가적인 데이터는 요청에 대한 CGI 프로그램 파일 이름 다음에 오는 첫 번째 슬래시(/) 뒤에 나옵니다. 데이터는 CGI 스펙에 따라 전달됩니다.

program_path

서버에서 요청에 대해 실행하는 CGI 프로그램을 포함하는 파일 경로를 지정합니다. *program_path*는 와일드 카드도 포함해야 합니다. 와일드 카드는 CGI 프로그램이 있는 파일의 이름으로 바뀝니다.

Exec 지시문은 되풀이되면 모든 부 지시문에 적용됩니다. cgi-bin 및 admin-bin에 있는 각 지시문에 대해 분리된 Exec 지시문이 필요없습니다.

[*Server_IP_address* | *host_name*]

여러 IP 주소나 가상 호스트를 사용하고 있다면, 이 매개변수를 사용하여 IP 주소나 호스트 이름을 지정하십시오. 서버는 이 IP 주소나 호스트의 서버에 도달하는 요청이나 이 호스트에 대해서만 지시문을 사용합니다. IP 주소는 요청하는 클라이언트의 주소가 아니라 서버 네트워크 연결의 주소입니다.

IP 주소(예:240.146.167.72) 또는 호스트 이름(예: hostA.bcd.com)을 지정할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수가 없으면, 서버는 요청이 들어오는 IP 주소나 URL의 호스트 이름과 상관없이 모든 요청에 대해 지시문을 사용합니다.

와일드 카드는 서버 IP 주소를 지정하는 데 사용할 수 없습니다.

예제

다음 예제에서 서버가 /idd/depts/plan/c92 요청을 받으면, 프로그램에 입력하여 전달된 c92가 있는 /depts/bin/plan.exe CGI 프로그램을 실행합니다.

다음 예제에서는 선택적 IP 주소 매개변수를 사용합니다. 서버가 /cgi-bin/으로 요청을 받으면, 요청이 들어오는 네트워크 연결의 IP 주소에 기반한 다른 디렉토리에서 요청을 제공합니다. 130.146.167.72로 들어오는 요청에 대해서 서버는 /CGI-BIN/customerA 디렉토리를 사용합니다. 0.83.100.45 주소와의 연결로 들어오는 요청에 대해서는 /CGI-BIN/customerB 디렉토리를 사용합니다.

```
Exec    /cgi-bin/*      /CGI-BIN/customerA/*  130.129.167.72
Exec    /cgi-bin/*      /CGI-BIN/customerB/*  0.83.100.45
```

다음 예제에서는 선택적 호스트 이름 매개변수를 사용합니다. 서버가 /cgi-bin으로 시작하는 요청을 수신하면, URL로된 호스트 이름에 기반한 다른 디렉토리에서 요청을 제공합니다. hostA.bcd.com에 들어오는 요청에 대해서 서버는 /CGI-BIN/customerA 디렉토리를 사용합니다. hostB.bcd.com로 들어오는 요청에 대해서 서버는 /CGI-BIN/customerB 디렉토리를 사용합니다.

```
Exec    /cgi-bin/*      /CGI-BIN/customerA/*  hostA.bcd.com
Exec    /cgi-bin/*      /CGI-BIN/customerB/*  hostB.bcd.com
```

기본값

- Linux 및 UNIX 시스템

```
Exec /cgi-bin/*      /opt/ibm/edge/cp/server_root/cgi-bin/*
Exec /admin-bin/*    /opt/ibm/edge/cp/server_root/admin-bin/*
```

- **Windows 시스템**

```
Exec server_root/cgi-bin/*
Exec server_root/admin-bin/*
Exec server_root/DOCS/admin-bin/*
```

ExportCacheImageTo — 디스크로 캐시 메모리 내보내기

이 지시문을 사용하여 캐시 콘텐츠를 덤프 파일로 내보낼 수 있습니다. 이는 메모리 캐시가 재시작 중에 유실되거나 다중 프록시에 대해 동일 캐시를 전개하는 경우에 유용합니다.

형식

ExportCacheImageTo *export_file_name*

기본값

없음

ExternalCacheManager — IBM WebSphere Application Server의 동적 캐시에 대한 Caching Proxy 구성

이는 역방향 프록시 구성에만 적용합니다.

이 지시문을 사용하여 동적 자원을 캐시할 수 있는 IBM WebSphere Application Server(Caching Proxy 어댑터 모듈로 구성됨)를 인식하도록 Caching Proxy를 구성하십시오. Caching Proxy는 Application Server의 동적 캐시에도 저장되는 JSP 결과 사본을 저장합니다. Caching Proxy는 ExternalCacheManager 항목과 일치하는 그룹 ID가 있는 IBM WebSphere Application Server의 콘텐츠만 캐시합니다.

또한 이 기능을 사용 가능하게 하려면 Service 지시문을 Caching Proxy 구성 파일에 추가해야 합니다. 추가 구성 단계는 Application Server에도 필요합니다. 완료 정보는 111 페이지의 제 22 장 『동적 생성 콘텐츠 캐시』를 참조하십시오.

형식

ExternalCacheManager *External_Cache_Manager_ID* *Maximum_Expiry_Time*

External_Cache_Manager_ID

프록시를 제공 중인 IBM WebSphere Application Server에 지정된 ID. ID는 Application Server의 dynacache.xml 파일에 있는 externalCacheGroup: group id 속성에 설정된 ID와 일치해야 합니다.

Maximum_Expiry_Time

외부 캐시 관리자 대신 캐시된 자원에 설정된 기본 만기 시간. 외부 캐시 관리자가

지정된 시간 내에 캐시된 자원을 무효화하지 않으면 자원이 지정된 시간에 만기됩니다. 분 또는 초로 시간을 지정할 수 있습니다.

예제

다음 항목은 `www.xyz.com` 도메인에 있고 자원이 20초 이전에 만기된 외부 캐시 관리자(IBM WebSphere Application Server)를 정의합니다.

```
ExternalCacheManager IBM-CP-XYZ-1 20초
```

기본값

없음

Fail — 일치하는 요청 거부

이 지시문을 사용하여 서버가 처리하지 않는 요청에 대한 템플릿을 지정합니다. 일단 요청이 Fail 지시문의 템플릿과 일치하면, 이 요청은 후속 지시문의 요청 템플릿과 비교되지 않습니다.

형식

```
Fail request_template [Server_IP_address | host_name]
```

request_template

서버가 거부(reject)하는 요청에 대한 템플릿을 지정합니다. 요청이 템플릿과 일치하면, 서버는 요청자에게 오류 메시지를 전송합니다.

이 템플릿에서 별표를 와일드 카드로 사용할 수 있습니다. 슬래시(/) 바로 뒤의 틸데(tilde) 문자는 정확히 일치해야 합니다. 와일드 카드는 틸데(tilde) 문자(~)를 일치시키는 데 사용될 수 없습니다.

```
[Server_IP_address | host_name]
```

여러 IP 주소나 가상 호스트를 사용하고 있다면, 이 매개변수를 사용하여 IP 주소나 호스트 이름을 지정하십시오. 서버는 이 IP 주소나 호스트의 서버에 도달하는 요청이나 이 호스트에 대해서만 지시문을 사용합니다. IP 주소는 요청하는 클라이언트의 주소가 아니라 서버 네트워크 연결의 주소입니다.

IP 주소(예: 240.146.167.72) 또는 호스트 이름(예: hostA.bcd.com)을 지정할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수가 없으면, 서버는 요청이 들어오는 IP 주소나 URL의 호스트 이름과 상관없이 모든 요청에 대해 지시문을 사용합니다.

서버 IP 주소로 와일드 카드 문자를 지정할 수 없습니다.

예제

다음 예제에서 서버가 `/usr/local/private/`로 시작하는 요청을 거부합니다.

```
Fail /usr/local/private/*
```

다음 예제에서는 선택적 IP 주소 매개변수를 사용합니다. 요청이 IP 주소 240.146.167.72와 연결된 네트워크로 들어오면, 서버는 /customerB/로 시작하는 요청을 거부합니다. 요청이 IP 주소 0.83.100.45와 연결된 네트워크로 들어오면, 서버는 /customerA/로 시작하는 요청을 거부합니다.

```
Fail    /customerB/*    240.146.167.72
Fail    /customerA/*    0.83.100.45
```

다음 예제에서는 선택적인 호스트 이름 매개변수를 사용합니다. 요청이 hostA.bcd.com으로 들어오면, 서버는 /customerB/로 시작하는 요청을 거부합니다. 요청이 hostB.bcd.com으로 들어오면, /customerA/로 시작하는 요청을 거부합니다.

```
Fail    /customerB/*    hostA.bcd.com
Fail    /customerA/*    hostB.bcd.com
```

기본값

없음

FIPSEnable — SSLV3 및 TLS에 대해 FIPS(Federal Information Processing Standard) 승인 암호를 사용 가능하게 함

이 지시문을 사용하여 SSL 연결에서 SSLV3 및 TLS 프로토콜에 대해 FIPS 승인 암호를 사용 가능하게 합니다. 이 지시문이 사용 가능한 경우, SSLV3 (V3CipherSpecs 지시문)에 대해 지원되는 암호 스펙 목록이 무시됩니다. 또한, 허용된 TLS 암호 스펙이 352F0AFF09FE로 설정되고, SSLV3 암호 스펙이 FFFE로 지정됩니다.

형식

```
FIPSEnable {on | off}
```

기본값

```
FIPSEnable off
```

flexibleSocks — 융통성있는 SOCKS 구현 사용 가능

이 지시문을 사용하여 프록시가 연결 유형을 판별하기 위해 SOCKS 구성 파일을 사용하도록 지시합니다.

형식

```
flexibleSocks {on | off}
```

기본값

```
flexibleSocks on
```

FTPDDirInfo — 디렉토리에 대한 환영 또는 설명 메시지 생성

이 지시문을 사용하여, FTP 서버가 디렉토리에 대한 환영 또는 설명 메시지를 생성할 수 있게 하십시오. 이 메시지는 FTP 목록의 일부로 선택적으로 표시될 수 있습니다. FTPDirInfo 지시문을 사용하여 메시지가 표시될 위치를 제어할 수 있습니다.

형식

FTPDDirInfo {top | bottom | off}

top

디렉토리의 파일 목록 이전에 페이지의 맨 위에 환영 메시지를 표시합니다.

bottom

디렉토리의 파일 목록 다음에 페이지의 맨 아래에 환영 메시지 표시합니다.

off

환영 페이지를 표시하지 않습니다.

기본값

FTPDDirInfo top

ftp_proxy — FTP 요청에 대한 다른 프록시 서버 지정

프록시 서버가 프록시 체인의 일부인 경우, 이 지시문을 사용하여 이 서버가 FTP 요청에 대해 접속할 다른 프록시의 이름을 지정합니다. 마지막에 슬래시 문자(/)를 포함하여 전체 URL을 지정해야 합니다. 선택적 도메인 이름 또는 템플릿 사용에 관한 정보는 263 페이지의 『no_proxy — 도메인에 직접 연결하기 위한 템플릿 지정』을 참조하십시오.

이는 정방향 프록시 구성에만 적용합니다.

형식

ftp_proxy full_URL [domain_name_or_template]

예제

ftp_proxy http://outer.proxy.server/

기본값

없음

FTPUriPath — FTP URL이 해석되는 방법 지정

이 지시문을 사용하여 FTP URL의 경로 정보가 로그인한 사용자의 작업 디렉토리와 관련하여 해석될지 또는 루트 디렉토리로 해석될지 여부를 지정합니다.

형식

FTPUrlPath {relative | absolute}

FTPUrlPath 지시문이 absolute로 설정되면, 로그인한 사용자의 FTP 작업 디렉토리가 FTP URL 경로에 포함되어야 합니다. FTPUrlPath Relative가 지정되면, 로그인한 사용자의 FTP 작업 디렉토리가 FTP URL 경로에서 생략되어야 합니다. 예를 들어, 로그인된 사용자용 작업 디렉토리 /export/home/user1에 포함된 test1.html 파일에 액세스하려면, FTPUrlPath 지시문의 설정에 따라 다음과 같은 URL 경로가 필요합니다.

- FTPUrlPath absolute로 설정되면, 필수 URL 경로는 ftp://ftphost/export/home/user1/test1.html입니다.
- FTPUrlPath relative로 설정되면, 필수 URL 경로는 ftp://ftphost/test1.html입니다.

기본값

없음

Gc — 가비지 콜렉션 지정

이 지시문을 사용하여 가비지 콜렉션이 사용되는지 여부를 지정합니다. 캐시가 사용 가능하면, 서버는 가비지 콜렉션 프로세스를 사용하여 더 이상 캐시되지 말아야 할 파일을 삭제합니다. 파일은 만기 날짜 및 기타 프록시 서버 지시문 값에 따라 삭제됩니다. 일반적으로 캐시를 사용할 수 있으면, 가비지 콜렉션이 사용됩니다. 가비지 콜렉션이 사용되지 않을 경우, 프록시 캐시는 비효율적으로 사용됩니다.

형식

Gc {on | off}

기본값

Gc On

GCAdvisor — 가비지 콜렉션 프로세스 사용자 정의

이 지시문을 사용하여, 서버가 가비지 콜렉션에 사용하려는 사용자 정의된 응용프로그램을 지정하십시오.

형식

GCAdvisor /path/file:function_name

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

예제

GCAvior /api/bin/customadvise.so:gcadv

GcHighWater — 가비지 콜렉션 시작 시간 지정

이 지시문을 사용하여, 가비지 콜렉션을 트리거하기 위해 채워져야 할 총 캐시 용량에 대한 백분율을 지정하십시오. 이 백분율을 최고 수준 표시라고 합니다. 최고 수준은 전체 캐시 용량에 대한 백분율로 지정됩니다. 가비지 콜렉션은 최저 수준 표시에 도달할 때까지 계속됩니다. 최저 수준 설정에 대한 정보는 『GcLowWater — 가비지 콜렉션 종료 시기 지정』을 참조하십시오. 최고 수준 백분율을 50 - 80으로 설정할 수 있습니다.

형식

GcHighWater *percentage*

기본값

GcHighWater 90

GcLowWater — 가비지 콜렉션 종료 시기 지정

이 지시문을 사용하여, 가비지 콜렉션 종료를 트리거하는 총 캐시 용량에 대한 백분율을 지정하십시오. 이 백분율을 최저 수준 표시라고 합니다. 최저 수준은 전체 캐시 용량에 대한 백분율로 지정됩니다. 최고 수준으로 설정한 값보다 낮게 설정해야 합니다. 최고 수준 설정에 대한 정보는 『GcHighWater — 가비지 콜렉션 시작 시간 지정』을 참조하십시오.

형식

GcLowWater *percentage*

기본값

GcLowWater 60

gopher_proxy — Gopher 요청에 대한 다른 프록시 서버 지정

프록시 서버가 프록시 체인의 일부인 경우, 이 지시문을 사용하여 이 서버가 Gopher 요청에 대해 접속할 다른 프록시의 이름을 지정합니다. 마지막에 슬래시(/)를 포함하여 전체 URL을 지정해야 합니다. 선택적 도메인 이름 또는 템플릿 사용에 관한 정보는 263 페이지의 『no_proxy — 도메인에 직접 연결하기 위한 템플릿 지정』을 참조하십시오.

이는 정방향 프록시 구성에만 적용합니다.

형식

gopher_proxy *full_URL*[*domain_name_or_template*]

예제

`gopher_proxy http://outer.proxy.server/`

기본값

없음

GroupId — 그룹 ID 지정

이 지시문을 사용하여 파일에 액세스하기 전에 서버가 변경할 대상 그룹의 이름이나 번호를 지정합니다.

이 지시문을 변경한 경우, 서버를 수동으로 정지한 다음 재시작해야 변경사항을 적용할 수 있습니다. 서버를 재시작하기만 하는 경우에는 변경사항이 적용되지 않습니다(15 페이지의 제 5 장 『Caching Proxy 시작 및 정지』 참조).

주: 사용자 ID, 그룹 ID, 또는 로그 디렉토리 경로에 대한 서버 기본값을 변경한 경우, 새 디렉토리를 작성하고 디렉토리의 권한 및 소유권을 갱신하십시오. 서버에서 사용자 정의된 로그 디렉토리에 정보를 기록하도록 하려면, 해당 디렉토리에 대한 권한을 755로 설정하고 사용자 정의된 서버 사용자 ID를 소유자로 설정하십시오. 예를 들어, 서버의 사용자 ID를 기본값에서 `jdoe`로 변경하고, 기본 로그 디렉토리를 `server_root/account`로 변경한 경우, `server_root/account` 디렉토리는 755의 권한을 가져야 하며 `jdoe`가 소유해야 합니다.

형식

`GroupId { group_name | group_number }`

기본값

AIX: `GroupId nobody`

HP-UX: `GroupId other`

Linux:

Red Hat: `GroupId nobody`

SUSE: `GroupId nogroup`

Solaris: `GroupId nobody`

HeaderServerName — HTTP 헤더에 리턴되는 프록시 서버 이름 지정

이 지시문을 사용하여 HTTP 헤더에 리턴되는 프록시 서버의 이름을 지정합니다.

형식

`HeaderServerName name`

기본값

없음

Hostname — 서버에 대한 정식 도메인 이름 또는 IP 주소 지정

이 지시문을 사용하여, 파일 요청에서 클라이언트로 리턴된 도메인 이름이나 IP 주소를 지정하십시오. 도메인 이름을 지정하면, 도메인 이름 서버가 이름을 IP 주소로 해석할 수 있어야 합니다. IP 주소를 지정하면, 도메인 이름 서버가 필요없고 액세스되지 않습니다.

주: 배열을 설정할 때, Hostname 지시문은 해당 배열의 모든 구성원이 식별할 수 있도록 구성해야 합니다.

형식

Hostname {*name* | *IP address*}

기본값

기본적으로 이 지시문은 초기 구성 파일에 지정되지 않습니다. 이 지시문을 구성 파일에 지정하지 않으면, 사용자 도메인 이름 서버에 정의된 호스트 이름을 기본값으로 합니다.

http_proxy — HTTP 요청에 대한 다른 프록시 서버 지정

프록시 서버가 프록시 체인의 일부인 경우, 이 지시문을 사용하여 이 서버가 HTTP 요청에 대해 접속할 다른 프록시의 이름을 지정합니다. 마지막에 슬래시(/)를 포함하여 전체 URL을 지정해야 합니다. 선택적 도메인 이름 또는 템플릿 사용에 관한 정보는 263 페이지의 『no_proxy — 도메인에 직접 연결하기 위한 템플릿 지정』을 참조하십시오.

형식

http_proxy *full_URL*[*domain_name_or_template*]

예제

http://outer.proxy.server/

기본값

없음

HTTPSCheckRoot — HTTPS 요청 필터

이 지시문을 사용하여, Caching Proxy가 비보안 홈페이지에서 URL을 검색하여 레이블 검색을 시도할지를 지정하십시오. 레이블을 발견하면 보안 요청을 적용합니다. 예를

들어, `https://www.ibm.com/`을 요청하면, Caching Proxy가 `http://www.ibm.com/`을 검색하여 레이블을 탐색하고 발견된 레이블을 사용하여 `https://www.ibm.com/`을 필터하는 데 사용합니다.

HTTPSCheckRoot를 off로 설정한 경우, Caching Proxy가 비보안 홈 페이지에서 레이블을 검색하지 않습니다.

형식

HTTPSCheckRoot {on | off}

기본값

HTTPSCheckRoot on

ICP_Address — ICP 조회의 IP 주소 지정

부 지시문을 사용하여 ICP 조회를 전송하고 수신하는 데 사용되는 IP 주소를 지정하십시오. <MODULEBEGIN> ICP 및 <MODULEEND> 지시문 내에 포함되어야 합니다.

형식

ICP_Address *IP_address*

기본값

기본적으로 이 지시문은 초기 구성 파일에 지정되지 않습니다. 이 지시문을 구성 파일에 지정하지 않으면, 모든 인터페이스에서 ICP 조회를 승인하고 전송하도록 기본값이 설정됩니다.

ICP_MaxThreads — ICP 조회를 위한 최대 스레드 지정

부 지시문을 사용하여 ICP 조회를 인식할 스레드 수를 지정하십시오. <MODULEBEGIN> ICP 및 <MODULEEND> 지시문 내에 포함되어야 합니다.

주: Redhat Linux 6.2 이하에서, 프로세스당 작성할 수 있는 스레드 수가 작기 때문에 이 숫자가 작아야 합니다. ICP 사용에 대해 많은 수의 스레드를 지정하면 요청 서비스에 사용할 수 있는 스레드의 수가 제한될 수 있습니다.

형식

ICP_MaxThreads *number_of_threads*

기본값

ICP_MaxThreads 5

Occupier — ICP 클러스터의 구성원 지정

프록시 서버가 ICP 클러스터의 일부인 경우, 부 지시문을 사용하여 ICP 피어를 지정하십시오. <MODULEBEGIN> ICP 및 <MODULEEND> 지시문 내에 포함되어야 합니다.

새 피어가 ICP 클러스터에 추가되면, ICP 피어 정보를 모든 기존 피어의 구성 파일에 추가해야 합니다. 각 피어에 하나의 행을 사용하십시오. 또한 현재 호스트가 피어 목록에 포함되어야 합니다. ICP가 초기화되면 현재 호스트의 항목을 무시합니다. 그러면 현재 호스트를 제거하기 위한 편집을 하지 않고 다른 피어 시스템에 복사할 수 있는 단일 구성 파일을 가질 수 있습니다.

형식

```
ICP_Peer hostname http_port icp_port
```

hostname

피어의 이름

http_port

피어의 프록시 포트

icp_port

피어의 ICP 서버 포트

예제

다음 행은 프록시 포트가 80이고 ICP 포트가 3128인 호스트(abc.xcompany.com)를 피어로서 추가합니다.

```
ICP_Peer abc.xcompany.com 80 3128
```

기본값

없음

ICP_Port — ICP 조회의 포트 번호 지정

부 지시문을 사용하여 ICP 서버가 ICP 조회를 인식할 포트 번호를 지정하십시오. <MODULEBEGIN> ICP 및 <MODULEEND> 지시문 내에 포함되어야 합니다.

형식

```
ICP_Port port_number
```

기본값

```
ICP_Port 3128
```

ICP_Timeout — ICP 조회의 최대 대기 시간 지정

부지시문을 사용하여 Caching Proxy가 ICP 조회에 대한 응답을 기다릴 최대 시간을 지정하십시오. 시간은 밀리초 단위로 지정됩니다. <MODULEBEGIN> ICP 및 <MODULEEND> 지시문 내에 포함되어야 합니다.

형식

ICP_Timeout *timeout_in_milliseconds*

기본값

ICP_Timeout 2000

IgnoreURL — 새로 고치지 않을 URL 지정

이 지시문을 사용하여 캐시 에이전트가 로드하지 않을 URL을 지정합니다. 이 지시문은 캐시 에이전트가 캐시 URL에 연결된 페이지를 로드할 때 유용합니다. 다른 URL이나 URL 마스크를 지정하여 여러 개의 IgnoreURL 지시문 어커런스를 사용할 수 있습니다. 이 지시문의 값은 마스크를 적용하기 위해 별표(*)를 와일드 카드로 포함할 수 있습니다.

형식

IgnoreURL *URL*

예제

IgnoreURL *http://www.yahoo.com/*
IgnoreURL *http://*.ibm.com/**

기본값

IgnoreURL **/cgi-bin/**

imbeds — 정보 포함 처리가 사용될지 여부 지정

이 지시문을 사용하여 파일 시스템, CGI 프로그램 또는 둘 다에서 제공된 파일에 대해 정보 포함 처리를 수행할 것인지 여부를 지정합니다. 정보 포함 처리는 콘텐츠 유형이 ext/x-ssi-html인 파일에서 처리됩니다. 선택적으로, 정보 포함 처리는 콘텐츠 유형이 text/html인 파일에 대해서도 수행될 수 있도록 지정할 수 있습니다. 자세한 내용은 195 페이지의 『AddType — 특정 접미부가 있는 파일의 데이터 유형 지정』을 참조하십시오.

정보 포함 처리를 사용하여, 리턴될 파일에 동적으로 정보를 삽입할 수 있습니다. 이러한 정보에는 날짜, 파일의 크기, 파일의 최종 변경 날짜, CGI 또는 정보 포함 환경 변수, 또는 텍스트 문서 등이 포함될 수 있습니다. 정보 포함 처리는 로컬로 생성된 파일에서만 수행됩니다. Caching Proxy는 프록시된 오브젝트나 캐시된 오브젝트에서 정보 포함 처리를 수행하지 않습니다.

정보 포함 처리에 의해서, 서버는 파일이 제공될 때마다 특수한 명령에 대한 문서를 탐색합니다. 이것은 서버의 성능에 영향을 주게 되어 클라이언트에 대한 응답 시간이 늦어집니다.

형식

`imbeds {on | off | files | cgi | noexec} {SSIOnly | html}`

on 정보 포함 처리가 파일 시스템 및 CGI 프로그램의 파일에 대해 수행됩니다.

off

정보 포함 처리가 모든 파일에 대해서 수행되지 않습니다.

files

정보 포함 처리가 파일 시스템의 파일에 대해서만 수행됩니다.

cgi

정보 포함 처리가 CGI 프로그램이 리턴한 파일에 대해서만 수행됩니다.

noexec

SSIOnly

정보 포함 처리가 콘텐츠 유형이 `text/x-ssi-html`인 파일에 대해서만 수행됩니다.

html

정보 포함 처리가 콘텐츠 유형이 `text/html` 및 `text/x-ssi-html`인 파일에 대해서만 수행됩니다.

서버는 검색한 각 파일의 콘텐츠 유형과 처리한 각 CGI 프로그램의 출력을 확인합니다.

정보 포함 처리는 일반적으로 콘텐츠 유형이 `text/x-ssi/html`인 파일에 대해서만 수행됩니다. 그러나 콘텐츠 유형이 `text/html`인 파일도 정보 포함에 대해 처리되도록 지정할 수 있습니다.

주: 서버는 `html`, `.html`, `.htm`을 `html`로 처리합니다. 다른 것은 `SSIOnly`로 처리됩니다.

각 접미부에는 올바른 콘텐츠 유형으로 정의된 `AddType` 지시문이 있어야 합니다. `.htm` 또는 `.html` 이외의 접미부를 사용하는 경우에는, `AddType` 지시문이 `text/x-ssi/html` 콘텐츠 유형으로 정의되도록 하십시오.

기본값

`imbeds on SSIOnly`

ImportCacheImageFrom — 파일에서 캐시 메모리 가져오기

이 지시문을 사용하여 덤프 파일에서 캐시 콘텐츠를 가져올 수 있습니다. 이는 메모리 캐시가 재시작 중에 유실되거나 다중 프록시에 대해 동일 캐시를 전개하는 경우에 유용합니다.

형식

ImportCacheImageFrom *import_file_name*

기본값

없음

InheritEnv — CGI 프로그램이 계승할 환경 변수 지정

이 지시문을 사용하여 CGI 프로그램이 계승하기를 원하는 CGI 처리에 고유한 CGI 환경 변수 이외의 환경 변수를 지정합니다.

InheritEnv 지시문을 포함하지 않으면, 모든 환경 변수가 CGI 프로그램에 의해 계승됩니다. InheritEnv 지시문을 포함하면, InheritEnv 지시문에 지정된 환경 변수만 CGI 고유한 환경 변수와 함께 계승됩니다. 이 지시문을 사용하여 계승될 변수값을 선택적으로 초기화할 수 있습니다.

형식

InheritEnv *environment_variable*

예제

```
InheritEnv PATH
InheritEnv LANG=ENUS
```

이 예제에서는 PATH 및 LANG 환경 변수만 CGI 프로그램에 의해 계승되며, LANG 환경 변수는 ENUS 값으로 초기화됩니다.

기본값

없음. 기본적으로 모든 환경 변수는 CGI 프로그램이 계승합니다.

InputTimeout — 입력 시간 종료 지정

이 지시문을 사용하여, 클라이언트가 서버에 연결한 후 요청을 전송하는 데 허용된 시간을 설정하십시오. 클라이언트는 먼저 서버에 연결할 다음, 요청을 전송합니다. 클라이언트가 이 지시문에 지정된 시간 내에 요청을 전송하지 않으면, 서버가 연결을 닫습니다. 시간, 분, 초의 결합으로 시간값을 지정하십시오.

형식

InputTimeout *time*

예제

InputTimeout 3 mins 30 secs

기본값

InputTimeout 2 minutes

JunctionReplaceUrlPrefix — JunctionRewrite 플러그인과 사용되는 경우, 접두부 삽입 대신 URL 바꾸기

이 지시문은 프록시가 html 페이지에서 특정 URL 링크를 지정하도록 허용하면서, JunctionRewrite 플러그인의 기본 조치를 대체합니다. JunctionRewrite 지시문과 함께 사용됩니다.

이는 역방향 프록시 구성에만 적용합니다.

JunctionReplaceUrlPrefix 지시문은 JunctionRewrite 플러그인을 지시하여 URL의 시작 부분에 접두부를 삽입하는 대신 URL을 *url_pattern_1* to *url_pattern_2*에서 바꿉니다.

형식

JunctionReplaceUrlPrefix *url_pattern_1 url_pattern_2*

예제

JunctionReplaceUrlPrefix /server1.internaldomain.com/* /server1/*

예를 들어, URL이 /server1.internaldomain.com/notes.nsf 및 접두부가 /server1이라 가정하십시오. 접두부를 삽입하여 URL을 /server1/server1.internaldomain.com/notes.nsf로 재작성하는 대신, JunctionRewrite 플러그인은 URL을 /server1/notes.nsf로 변경합니다.

기본값

없음

JunctionRewrite — URL 재작성 사용 가능

이 지시문은 서버의 상대 URL이 결합 사용 시 해당하는 기점 서버로 반드시 맵핑되도록 기점 서버의 응답을 재작성하는 결합 재작성 루틴을 Caching Proxy 내에서 사용 가능하게 합니다.

이는 역방향 프록시 구성에만 적용합니다.

UseCookie 옵션 없이 **JunctionRewrite on**을 설정하는 경우에는 결합 재작성 플러그인도 사용 가능해야 합니다. 결합은 프록시 맵핑 규칙에 의해 정의됩니다.

JunctionRewrite에 대한 추가 정보는 49 페이지의 『JunctionRewrite에 대한 대안으로서의 UseCookie』 및 50 페이지의 『JunctionRewrite 기능성의 확장을 위한 샘플 transmogrifier 플러그인』의 내용을 참조하십시오.

형식

```
JunctionRewrite {on | on UseCookie | off}
```

기본값

```
JunctionRewrite off
```

JunctionRewriteSetCookiePath — JunctionRewrite 플러그인과 사용되는 경우, Set-Cookie 헤더에 경로 옵션을 재작성

지시문은 쿠키 이름이 일치되는 경우, 프록시가 Set-Cookie 헤더의 경로 옵션을 재작성하도록 허용합니다. 응답에 결합이 필요하고, 결합 접두부가 정의되어 있는 경우, 접두부는 각 경로 앞에 삽입됩니다. JunctionRewrite 플러그인과 함께 사용하거나 RewriteSetCookieDomain 지시문과 함께 사용할 수 있습니다.

이는 역방향 프록시 구성에만 적용합니다.

형식

```
JunctionRewriteSetCookiePath cookie-name1 cookie-name2...
```

cookie-name

Set-Cookie 헤더의 쿠키 이름.

기본값

없음

JunctionSkipUrlPrefix — JunctionRewrite 플러그인과 사용되는 경우, 이미 접두부를 포함하는 URL 재작성을 건너뛰기

이 지시문은 URL 패턴이 이미 일치하는 경우, URL을 재작성하지 않고 JunctionRewrite 플러그인의 기본 조치를 대체합니다. 이는 html 페이지의 일부 URL 링크를 정정하는 방법을 제공하는 JunctionRewrite 플러그인과 작동합니다. 보통, 이 지시문은 이미 접두부를 포함하는 URL을 건너뛰는 데 사용됩니다.

이는 역방향 프록시 구성에만 적용합니다.

형식

```
JunctionSkipUrlPrefix url_pattern
```

예제

```
JunctionSkipUrlPrefix /server1/*
```


예를 들어, URL은 /server1/notes.nsf이고, 결합 접두부는 /server1/이라 가정하십시오. URL을 /server1/server1/notes.nsf로 재작성하는 대신, JunctionRewrite 플러그인은 URL 재작성을 건너뛰어, URL은 변경되지 않고 /server1/notes.nsf입니다.

기본값

없음

KeepExpired — 자원이 프록시에서 갱신되는 경우, 자원의 만기된 사본을 리턴하도록 지정

이 지시문을 사용하여 캐시 오브젝트를 다시 검증하는 동안, 백엔드 서버에 요청이 집중되는 것을 방지할 수 있습니다.

캐시 오브젝트를 백엔드 서버에서 콘텐츠와 재확인하는 경우, 동일한 자원에 대한 요청이 백엔드 서버로 프록시됩니다. 동일한 요청이 집중되는 경우, 백엔드 서버의 성능이 저하될 것입니다. 이 지시문을 사용하여 이러한 상황이 발생하는 것을 방지할 수 있습니다. 지시문을 사용 가능하게 되면, 만기되거나 오래된 자원의 사본은 자원이 프록시에서 갱신된 경우, 리턴됩니다.

형식

KeepExpired {on | off}

기본값

KeepExpired off

KeyRing — 키 링 데이터베이스에 대한 파일 경로 지정

이 지시문을 사용하여 서버가 SSL 요청에 사용하는 키 링 데이터베이스에 파일 경로를 지정합니다. 키 링 파일은 iKeyman 키 관리자 유틸리티를 통하여 생성됩니다.

주: SSL 지시문은 SUSE Linux에서 지원되지 않습니다.

형식

KeyRing *filename*

예제

Windows: KeyRing c:\Program Files\IBM\edge\cp\\key.kdb

Linux 및 UNIX: KeyRing /etc/key.kdb

기본값

없음

KeyRingStash — 키 링 데이터베이스 암호 파일에 대한 파일 경로 지정

이 지시문을 사용하여 키 링 데이터베이스 암호 파일에 대한 파일 경로를 지정합니다. 암호 파일은 키 링 데이터베이스 파일을 작성할 때 iKeyman 키 관리자 유틸리티를 통하여 생성됩니다.

주: SSL 지시문은 SUSE Linux에서 지원되지 않습니다.

형식

KeyRingStash *file_path*

예제

Windows: KeyRingStash c:\Program Files\IBM\edge\cp\key.sth

Linux 및 UNIX: KeyRingStash /etc/key.sth

기본값

없음

LimitRequestBody — PUT 또는 POST 요청의 최대 본문 크기 지정

이 지시문을 사용하여 PUT 또는 POST 요청의 최대 본문 크기를 제어할 수 있습니다. LimitRequest 지시문은 공격으로부터 프록시를 보호하기 위해 사용됩니다.

값은 킬로바이트(K), 메가바이트(M) 또는 기가바이트(G)로 지정할 수 있습니다.

형식

LimitRequestBody *max_body_size* {K | M | G}

기본값

LimitRequestBody 10 M

LimitRequestFields — 클라이언트 요청에서 헤더의 최대 수 지정

이 지시문을 사용하여 클라이언트 요청에서 전송될 수 있는 헤더의 최대 수를 지정할 수 있습니다. LimitRequest 지시문은 공격으로부터 프록시를 보호하기 위해 사용됩니다.

형식

LimitRequestFields *number_headers*

기본값

LimitRequestFields 32

LimitRequestFieldSize — 최대 헤더 길이 및 요청 행 지정

이 지시문을 사용하여 요청에서 각 헤더의 최대 길이 및 요청 행의 최대 길이를 지정할 수 있습니다. LimitRequest 지시문은 공격으로부터 프록시를 보호하기 위해 사용됩니다.

값은 바이트(B) 또는 킬로바이트(K)로 지정할 수 있습니다.

형식

LimitRequestFieldSize *max_hdr_length* {B | K}

기본값

LimitRequestFieldSize 4096 B

ListenBacklog — 서버가 수행할 수 있는 인식 백로그 클라이언트 연결 수 지정

이 지시문을 사용하여 서버가 연결 거부된 메시지를 클라이언트로 보내기 전에 수행하는 인식 백로그 클라이언트 연결 수를 지정합니다. 연결 수는 서버가 몇 초 안에 처리할 수 있는 요청의 수에 따라 다릅니다. 클라이언트가 시간 종료되어 연결을 중단하기 전에 서버에서 처리할 수 있는 수보다 큰 값을 설정하지 마십시오.

주: ListenBacklog 값이 TCP/IP가 지원하는 SOMAXCONN 값보다 크면, 대신 SOMAXCONN 값이 사용됩니다.

형식

ListenBacklog *number_of_requests*

기본값

ListenBacklog 128

LoadInlineImages — 삽입된 이미지 새로 고침 제어

이 지시문을 사용하여 인라인 이미지가 캐시 에이전트에 의해 검색되는지 여부를 지정합니다. LoadInlineImages가 on으로 설정되면, 캐시 중인 페이지에 삽입된 이미지도 캐시됩니다. off로 설정되면, 삽입된 이미지는 캐시되지 않습니다.

형식

LoadInlineImages {on | off}

기본값

LoadInlineImages on

LoadTopCached — 새로 고칠 즐겨찾기 페이지 수 지정

이 지시문을 사용하여 캐시 에이전트가 전날 밤의 캐시 액세스 로그에 액세스하여 가장 많이 요청되는 URL을 로드하도록 지시합니다.

CacheAccessLog 지시문에 대한 값을 설정하는 경우, Caching 지시문을 On으로 설정해야 하며 LoadTopCached 지시문에 대해 특정 값을 설정해야 합니다.

형식

LoadTopCached *number_of_pages*

기본값

LoadTopCached 100

LoadURL — 새로 고칠 URL 지정

이 지시문을 사용하여 캐시 에이전트에 의해 캐시로 로드될 URL을 지정합니다. 여러 개의 LoadURL 지시문을 구성 파일에 포함할 수 있지만, 와일드 카드는 사용할 수 없습니다.

형식

LoadURL *url*

예제

LoadURL http://www.ibm.com/

기본값

없음

Log — 로그 단계 사용자 정의

이 지시문을 사용하여 로그 단계 동안 서버가 호출하는 사용자 정의된 응용프로그램 기능을 지정합니다. 이 코드는 연결이 종료된 이후 수행되는 로그 및 기타 처리를 제공합니다.

형식

Log *request_template* */path/file:function_name*

request_template

응용프로그램 기능이 호출되는지 여부를 추가적으로 판별하는 요청에 대한 템플릿을 지정합니다. 스펙은 프로토콜, 도메인, 호스트를 포함할 수 있고, 앞에 슬래시(/)가 붙을 수 있으며, 별표(*)를 와일드 카드로 사용할 수 있습니다. 예를 들어, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /* 및 *는 모두 유효합니다.

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다. 열기, 쓰기, 닫기 기능의 이름을 공급해야 합니다.

예제

```
Log      /index.html /api/bin/icsextpgm.so:log_url
```

기본값

없음

LogArchive — 로그 보존 작동 지정

이 지시문을 사용하여 아카이브 루틴의 동작을 지정하십시오. 이 지시문은 글로벌 설정이 있는 모든 로그에 영향을 줍니다. 따라서 로그를 압축하거나 폐기할지, 또는 어떤 작동도 수행되지 않도록 할 것인지를 지정합니다.

Compress를 지정한 경우, CompressAge 및 CompressDeleteAge 지시문을 사용하여 로그를 압축하거나 삭제할 시기를 지정합니다. CompressCommand 지시문을 사용하여 사용할 명령과 해당 매개변수를 지정합니다.

Purge를 지정한 경우, PurgeAge 및 PurgeSize 지시문을 사용하여 로그를 폐기할 시기를 지정합니다.

형식

```
LogArchive {Compress | Purge | none}
```

Compress

아카이브 루틴이 로그를 압축하도록 지정합니다.

Purge

아카이브 루틴이 로그를 지우도록 지정합니다.

none

아카이브 루틴이 아무 작업도 하지 않도록 지정합니다.

기본값

LogArchive Purge

관련 지시문

- 213 페이지의 『CompressAge — 로그 압축 시기 지정』
- 214 페이지의 『CompressDeleteAge — 로그 삭제 시기 지정』
- 213 페이지의 『CompressCommand — 압축 명령 및 매개변수 지정』

- 260 페이지의 『Midnight — 로그 보존에 사용되는 API 플러그인 지정』
- 289 페이지의 『PurgeAge — 로그의 유효 기간 한계 지정』
- 289 페이지의 『PurgeSize — 로그 보존 크기의 한계 지정』

LogFileFormat — 액세스 로그 형식 지정

이 지시문을 사용하여 액세스 로그 파일의 파일 형식을 지정합니다.

형식

```
LogFileFormat {common | combined}
```

기본적으로 로그는 NCSA 공통 로그 형식에 표시됩니다. 대신 combined를 지정하여 NCSA 통합 로그 형식에 로그를 표시합니다. 통합 형식은 URL, 사용자 에이전트, 쿠키(요청에 있는 경우) 참조용 필드를 추가합니다.

기본값

```
LogFileFormat common
```

LogToGUI (Windows only) — 서버 창에 로그 입력 항목 표시

Windows 시스템 전용. 명령행을 통해 프록시를 실행하는 경우에는 이 지시문을 사용하여 액세스 로그로 출력하십시오. 서버 성능을 최적화하기 위해 이 지시문은 기본적으로 off(사용 불가능)로 설정됩니다.

주: 이 지시문은 프록시를 서비스로서 실행할 때 영향이 없습니다.

형식

```
LogToGUI {on | off}
```

기본값

```
LogToGUI off
```

LogToSyslog — 액세스 정보를 시스템 로그에 전송할지 여부 지정(Linux 및 UNIX 전용)

Linux 및 UNIX 시스템 전용. 이 지시문을 사용하여 서버가 액세스 및 오류 로그 파일뿐만 아니라, 액세스 요청 및 오류를 시스템 로그에 로그할 것인지 여부를 지정합니다.

형식

```
LogToSyslog {on | off}
```

오류 로그 정보가 서버에 기록되도록 지정하기 전에 시스템 로그 파일이 서버에 있어야 합니다. 액세스 정보, 오류 정보 또는 둘 다 로그할지 여부를 선택할 수 있습니다.

시스템 로그에 오류 정보만 전송하려면 /etc/syslog.conf 파일에 다음 행을 추가하십시오.

```
user.err syslog_output_file_for_error_information
```

시스템 로그에 액세스 정보만 전송하려면 /etc/syslog.conf 파일에 다음 행을 추가하십시오.

```
user.info syslog_info_file_for_access_information
```

시스템 로그에 오류 및 액세스 정보를 둘다 전송하려면 /etc/syslog.conf 파일에 다음 행을 추가하십시오.

syslog_output_file 및 *syslog_info_file*을 다음과 같은 형식으로 지정합니다.

- **AIX:** /var/adm/name_of_syslog_file
- **HP-UX:** /var/adm/syslog/syslog.log
- **Linux:** /var/adm/messages
- **Solaris:** /var/adm/messages

시스템 로그 파일을 작성한 후 다음 명령으로 재시작할 수 있습니다.

```
kill -HUP 'cat /etc/syslog.pid'
```

기본값

```
LogToSyslog Off
```

Map — 규칙을 일치시키는 요청 경로 문자열을 사용하여 새 요청 문자열에 일치하는 요청 변경

이 지시문을 사용하여 새 요청 문자열에 대해 변경하려는 요청의 템플릿을 지정합니다. 요청을 변경한 후, 서버는 새 요청 문자열을 후속 지시문의 요청 템플릿과 비교합니다.

맵 지시문은 수신 요청 경로 문자열을 사용하여 규칙을 일치시킵니다. 관련 주제: 255 페이지의 『MapQuery — 규칙을 일치시키는 요청 경로 및 조회 문자열을 사용하여 일치하는 요청을 새 요청 문자열로 변경』

형식

```
Map request_template new_request [server_IP_address | host_name]
```

request_template

서버가 변경하는 요청에 대한 템플릿을 지정한 다음 새 요청 문자열을 다른 템플릿과 비교합니다.

이 템플릿에서 별표(*)를 와일드 카드로 사용할 수 있습니다. 슬래시(/) 바로 뒤의 틸데(tilde) 문자는 정확히 일치해야 합니다. 와일드 카드는 틸데(tilde) 문자(~)를 일치시키는 데 사용될 수 없습니다.

new_request

서버가 후속 지시문의 요청 템플릿과 계속 비교할 새 요청 문자열을 지정하십시오. *request_template*에 와일드 카드가 있으면, *new_request*로 지정된 문자열에도 와일드 카드가 포함될 수 있습니다. *request_template* 와일드 카드와 일치하는 요청 부분이 *new_request*의 와일드 카드 대신 삽입됩니다.

[*server_IP_address* | *host_name*]

여러 IP 주소나 가상 호스트를 사용하고 있다면, 이 매개변수를 사용하여 IP 주소나 호스트 이름을 지정하십시오. 서버는 이 IP 주소나 호스트의 서버에 도달하는 요청이나 이 호스트에 대해서만 지시문을 사용합니다. IP 주소는 요청하는 클라이언트의 주소가 아니라 서버 네트워크 연결의 주소입니다.

IP 주소(예: 240.146.167.72) 또는 호스트 이름(예: hostA.raleigh.ibm.com)을 지정할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수가 없으면, 서버는 요청이 들어오는 IP 주소나 URL의 호스트 이름과 상관없이 모든 요청에 대해 지시문을 사용합니다.

서버 IP 주소로 와일드 카드 문자를 지정할 수 없습니다.

예제

- 다음 예제에서 서버는 */stuff/*로 시작하는 모든 요청의 */stuff/* 부분을 */good/stuff/*로 변경합니다. 원래 요청에서 */stuff/* 다음의 모든 부분도 새 요청 문자열에 포함됩니다. 그러므로, */stuff/whatsup/*은 */good/stuff/whatsup/*으로 변경됩니다. 서버는 새 요청 문자열로 후속 지시문의 요청 템플릿을 계속 비교합니다.

```
Map /stuff/* /good/stuff/*
```

- 다음 예제에서는 선택적 IP 주소 매개변수를 사용합니다. 서버가 */stuff/*로 시작하는 요청을 받으면, 서버는 이 요청을 요청이 들어오는 네트워크 연결의 IP 주소에 기반한 다른 요청 문자열로 변경합니다. 240.146.167.72에 들어오는 요청에 대해 서버가 요청의 */stuff/* 부분을 */customerA/good/stuff/*로 변경합니다. 주소가 0.83.100.45인 모든 연결로 들어오는 요청에 대해 서버가 요청의 */stuff/* 부분을 */customerB/good/stuff/*로 변경합니다.

```
Map /stuff/* /customerA/good/stuff/* 240.146.167.72
Map /stuff/* /customerB/good/stuff/* 0.83.100.45
```

- 다음 예제에서는 선택적인 호스트 이름 매개변수를 사용합니다. 서버가 */stuff/*로 시작하는 요청을 받으면, 서버는 이 요청을 URL 호스트 이름에 기반한 다른 요청 문자열로 변경합니다. hostA에 들어오는 요청에 대해 서버가 요청의 */stuff/* 부분

을 /customerA/good/stuff/로 변경합니다. hostB에 들어오는 요청에 대해 서버가 요청의 /stuff/ 부분을 /customerB/good/stuff/로 변경합니다.

Map	/stuff/*	/customerA/good/stuff/*	hostA.bcd.com
Map	/stuff/*	/customerB/good/stuff/*	hostB.bcd.com

기본값

없음

MapQuery — 규칙을 일치시키는 요청 경로 및 조회 문자열을 사용하여 일치하는 요청을 새 요청 문자열로 변경

이 지시문을 사용하여 새 요청 문자열에 대해 변경하려는 요청의 템플릿을 지정합니다. 요청을 변경한 후, 서버는 새 요청 문자열을 후속 지시문의 요청 템플릿과 비교합니다.

지시문의 기능은 맵 규칙과 거의 동일합니다(253 페이지의 『Map — 규칙을 일치시키는 요청 경로 문자열을 사용하여 새 요청 문자열에 일치하는 요청 변경』). 그러나 URL을 조회 문자열로 핸들하려면, MapQuery에서 경로 및 조회 문자열을 사용하여 규칙을 일치시킵니다. 수신 URL이 MapQuery 규칙에 일치되는 경우, 변환된 URL은 규칙의 나머지 부분에 대해 일치되도록 사용됩니다.

MapQuery를 통해 조회 문자열이 있는 URL을 다른 경로 또는 다른 조회 문자열이 있는 다른 URL로 변환할 수 있습니다. 그러나, 다른 모든 맵핑 지시문이 요청 경로만을 사용하므로, 변경된 조회 문자열은 요청 경로가 일치하는 경우 변환된 URL에 추가(패턴을 일치하도록 사용되지 않음)되기만 합니다.

형식

MapQuery *request_template new_request* [*server_IP_address* | *host_name*]

request_template

서버가 변경하는 요청에 대한 템플릿을 지정한 다음 새 요청 문자열을 다른 템플릿과 비교합니다.

이 템플릿에서 별표(*)를 와일드 카드로 사용할 수 있습니다. 슬래시(/) 바로 뒤의 틸데(tilde) 문자는 정확히 일치해야 합니다. 와일드 카드는 틸데(tilde) 문자(~)를 일치시키는 데 사용될 수 없습니다.

new_request

서버가 후속 지시문의 요청 템플릿과 계속 비교할 새 요청 문자열을 지정하십시오. *request_template*에 와일드 카드가 있으면, *new_request*로 지정된 문자열에도 와일드 카드가 포함될 수 있습니다. *request_template* 와일드 카드와 일치하는 요청 부분이 *new_request*의 와일드 카드 대신 삽입됩니다.

[*server_IP_address* | *host_name*]

여러 IP 주소나 가상 호스트를 사용하고 있다면, 이 매개변수를 사용하여 IP 주소

나 호스트 이름을 지정하십시오. 서버는 이 IP 주소나 호스트의 서버에 도달하는 요청이나 이 호스트에 대해서만 지시문을 사용합니다. IP 주소는 요청하는 클라이언트의 주소가 아니라 서버 네트워크 연결의 주소입니다.

IP 주소(예: 240.146.167.72) 또는 호스트 이름(예: hostA.raleigh.ibm.com)을 지정할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수가 없으면, 서버는 요청이 들어오는 IP 주소나 URL의 호스트 이름과 상관없이 모든 요청에 대해 지시문을 사용합니다.

서버 IP 주소로 와일드 카드 문자를 지정할 수 없습니다.

예제

수신 URL을 다음과 같이 가정하십시오.

/getsomthing?type=1

MapQuery 규칙은 다음과 같습니다.

MapQuery /getsomthing?type=* /gettype/*

변환된 URL은 /gettype/1이고, 다음 규칙 맵핑에서 사용됩니다.

Proxy /gettype/* http://server/gettype/*

변환된 URL은 http://server/gettype/1입니다.

기본값

없음

MaxActiveThreads — 최대 활성 스레드 수 지정

이 지시문을 사용하여 한 번에 활성화시킬 스레드의 최대수를 설정합니다. 최대수에 도달하면, 다른 요청이 종료되고 스레드가 사용 가능해질 때까지 서버가 새 요청을 보유합니다. 일반적으로 시스템의 성능이 떨어날수록 이 지시문에 대한 설정값이 커집니다. 시스템이 메모리 스와핑과 같은 오버헤드 task에 너무 많은 시간이 걸리면, 이 값을 줄여보십시오.

형식

MaxActiveThreads *number_of_threads*

기본값

MaxActiveThreads 100

MaxContentLengthBuffer — 동적 데이터에 대한 버퍼 크기 지정

이 지시문을 사용하여, 서버가 생성한 동적 데이터에 대한 버퍼 크기를 설정하십시오. 동적 데이터는 CGI 프로그램, 정보 포함 및 API 프로그램의 출력입니다.

값은 바이트(B), 킬로바이트(K), 메가바이트(M), 기가바이트(G)로 지정할 수 있습니다. 숫자와 값(B, K, M, G) 사이에 공간이 있는지 여부는 상관없습니다.

형식

MaxContentLengthBuffer size

기본값

MaxContentLengthBuffer 100 K

MaxLogFileSize — 각 로그 파일의 최대 크기 지정

이 지시문을 사용하여 각 로그 파일의 최대 크기를 지정할 수 있습니다. 각 로그 파일은 이 지시문에서 정의한 크기를 초과할 수 없습니다. 일단 로그 파일이 최대 정의된 크기에 도달하면, 현재 로그 파일이 닫히고 새 로그 파일이 동일 이름으로 작성되며 이름에는 다음에 증가된 정수 값이 추가됩니다.

주:

1. Caching Proxy는 32비트 응용프로그램이며 32비트 함수로 자체 로그 파일을 엽니다. 이 제한조건으로 인해 2GB를 초과하는 MaxLogFileSize를 지정하지 마십시오. 활성으로 요청을 처리하는 동안 Caching Proxy가 로그 파일에 쓰려고 시도할 때 로그 파일의 크기가 2GB를 초과하면 Caching Proxy가 정지할 수 있습니다.
2. Linux 및 UNIX 플랫폼에서 로그 파일이 있는 디렉토리의 권한이, ibmproxy 디먼이 실행되는 최소한의 그룹에 대해 쓰기 권한이 없는 경우에는 로그 파일이 작성되지 않습니다. 다시 말하면, ibmproxy.conf 파일의 로깅 지시문에 대한 로그 파일 위치는 ibmproxy.conf 파일의 GroupId 지시문에 의해 정의된 최소한의 그룹에 대해 쓰기 권한이 있어야 합니다. 이는 로그 파일의 기본 위치가 변경되었거나 기본 UserId 또는 GroupId 지시문이 ibmproxy.conf 파일에서 변경된 경우에만 문제가 됩니다.

MaxLogFileSize 지시문을 설정하는 데 권장되는 값은 최소 10M이고 200M 미만입니다. 실제 로그 파일 크기는 설정한 크기보다 약간 큼니다. 값을 너무 낮게 설정하면, 프록시 서버에서 로그 파일을 자주 열고 닫게 되므로 프록시 성능에 영향을 미칩니다. 일부 플랫폼에서, 값을 너무 높게 설정하면, 프록시가 I/O 버퍼링에서 더 많은 메모리를 사용하게 합니다. 운영체제에서 I/O 버퍼를 조절할 수는 있지만 로그 파일 크기가 커질수록 프록시에서 사용할 메모리가 부족하거나 메모리 누수가 발생할 수 있습니다.

최대 크기는 바이트(B), 킬로바이트(K), 메가바이트(M) 및 기가바이트(G) 단위 중 하나로 지정이 가능합니다.

형식

MaxLogFileSize maximum {B | K | M | G}

기본값

MaxLogfileSize 128 M

MaxPersistRequest — 지속적인 연결에서 수신할 요청의 최대수 지정

이 지시문을 사용하여 서버가 지속적인 연결에서 수신할 최대 요청 수를 지정합니다. 이 수를 판별할 때 페이지에 사용된 이미지의 수를 고려하십시오. 각 이미지는 분리된 요청을 필요로 합니다.

형식

MaxPersistRequest *number*

기본값

MaxPersistRequest 5

MaxQueueDepth — 대기열에 넣을 URL의 최대수 지정

이 지시문을 사용하여 미해결 페이지 검색 요청의 캐시 에이전트 대기열의 최대 깊이를 지정합니다. 대용량 메모리의 대형 시스템인 경우, 사용 가능한 메모리를 모두 소비하지 않고도 페이지 검색 요청의 대기열을 더 크게 정의할 수 있습니다.

캐시에 대한 URL 대기열은 캐시 에이전트를 각각 시작할 때 판별됩니다. 캐시 에이전트가 다른 URL에 대한 하이퍼텍스트 연결을 따르도록 지시하면, 이들 기타 URL은 캐시 대기열 깊이에 계산되지 않습니다. MaxURLs 지시문에 지정된 값에 도달한 후에는 대기열에 URL이 더 있더라도 캐시 에이전트는 정지합니다.

형식

MaxQueueDepth *maximum_depth*

기본값

MaxQueueDepth 250

MaxRuntime — 캐시 에이전트의 최대 실행 시간 지정

이 지시문을 사용하여 캐시 에이전트 특정 실행 중에 URL을 검색하는 시간의 최대값을 지정합니다. 값이 0이면 캐시 에이전트가 완료될 때까지 실행합니다.

형식

MaxRuntime {0 | *maximum_time*}

예제

MaxRuntime 2 hours 10 minutes

기본값

MaxRuntime 2 hours

MaxSocketPerServer — 서버의 개방형 대기 소켓 최대수 지정

이 지시문을 사용하여, 하나의 기점 서버를 유지할 개방형 대기 소켓의 최대수를 설정하십시오. ServerConnPool 지시문이 on으로 설정된 경우에만 이 지시문을 사용하십시오.

형식

MaxSocketPerServer *num*

예제

MaxSocketPerServer 10

기본값

MaxSocketPerServer 5

MaxURLs — 새로 고칠 URL의 최대수 지정

이 지시문을 사용하여 캐시 에이전트가 특정 실행 중에 검색할 최대 URL 수를 지정합니다. 값이 0이면 한계가 없습니다. 캐시 에이전트의 자동 모드가 사용될 때 LoadURL 및 LoadTopCached 지시문이 MaxURLs 보다 우선순위를 가집니다.

형식

MaxURLs *maximum_number*

기본값

MaxURLs 2000

Member — 배열의 구성원 지정

이 지시문을 사용하여 서버가 원격 캐시 액세스를 사용하여 공유할 배열의 구성원을 지정합니다.

주: 배열을 설정할 때, Hostname 지시문을 해당 배열의 모든 구성원이 식별하도록 구성하십시오.

형식

```
Member name {  
  subdirective  
  subdirective  
  .  
  .  
}
```

다음과 같은 부 지시문이 포함됩니다.

RCAAddr

RCA 통신의 IP 주소나 호스트 이름을 식별하는 필수 부 지시문입니다.

RCAPort

RCA 통신의 포트를 식별하는 필수 부 지시문입니다. 포트 번호는 1024보다 크고 65535 미만이어야 합니다.

CacheSize {*n bytes* | *n Kbytes* | *n Mbytes* | *n Gbytes*}

요청한 부 지시문은 이 구성원의 캐시 크기를 식별합니다. 크기 값은 양수여야 합니다.

[Timeout *n milliseconds* | *n seconds* | *n hours* | *n days* | *n months* | *n years* | forever]

이 구성원을 기다리는 기간을 식별합니다. *n*은 양수여야 합니다. Timeout은 선택적이며, 기본값은 1000 milliseconds입니다. 제한 시간 값은 일반적으로 초나 밀리초 단위로 설정됩니다.

[BindSpecific {on | off}]

개인용 서브넷에서 통신이 발생하도록 허용하며, 보안 조치를 제공합니다. BindSpecific은 선택적이며, 기본값은 On입니다.

[ReuseAddr {on | off}]

배열의 빠른 재결합을 허용합니다. 이를 On으로 설정하면 다른 프로세스가 포트를 차지할 수 있으며, 이는 정의되지 않은 작동을 일으킬 수 있습니다. ReuseAddr은 선택적이며 기본값은 Off입니다.

예제

```
Member bittersweet.chocolate.ibm.com {  
  RCAAddr      127.0.0.1  
  RCAPort      6294  
  CacheSize    25G  
  Timeout      500 milliseconds  
  BindSpecific On  
  ReuseAddr    Off  
}
```

기본값

없음

Midnight — 로그 보존에 사용되는 API 플러그인 지정

이 지시문을 사용하여 로그를 보존하기 위해 밤에 실행되는 응용프로그램 플러그인을 지정할 수 있습니다. 이 지시문은 설치 동안 초기화됩니다. 이 지시문이 구성 파일에 포함되어 있지 않으면, 보존을 수행하지 않습니다.

형식

Midnight */path/file:function_name*

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

기본값

- **Linux 및 UNIX:** Midnight */usr/lib/archive.so:begin*
- **Windows:** Midnight *C:\Program Files\IBM\edge\cp\bin\archive.dll:begin*

NameTrans — 이름 변환 단계 사용자 정의

이 지시문을 사용하여 이름 변환 단계 중에 서버가 호출하는 사용자 정의된 응용프로그램 함수를 지정합니다. 이 코드는 URL을 고유한 오브젝트로 맵핑하면서, 요청의 가상 경로를 서버의 물리적 경로로 변환하기 위한 메커니즘을 공급합니다.

주: 이것은 터미널 맵핑 규칙이 아닙니다. 변형된 URL은 아직 Exec, Fail, Map, Pass, Redirect 중 Service와 같은 터미널 맵핑 규칙 지시문 중 하나와 일치해야 합니다.

형식

NameTrans *request_template* */path/file:function_name*
[*Server_IP_address* | *host_name*]

request_template

응용프로그램 기능이 호출되는지 여부를 추가적으로 판별하는 요청에 대한 템플릿을 지정합니다. 스펙은 프로토콜, 도메인, 호스트를 포함할 수 있고, 앞에 슬래시(/)가 붙을 수 있으며, 별표(*)를 와일드 카드로 사용할 수 있습니다. 예를 들어, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /* 및 *는 모두 유효합니다.

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

[*Server_IP_address* | *host_name*]

여러 개의 IP 주소나 가상 호스트를 사용하고 있으면, 응용프로그램 기능이 고유한 IP 주소로 들어오는 요청이나 특정 호스트에 대해서만 호출될지를 판별합니다.

서버 IP 주소로 와일드 카드 문자를 지정할 수 없습니다.

주: 이 지시문은 여기에는 읽기 쉽도록 두 행으로 표시되어 있지만, 한 행에 입력되어야 합니다.

예제

```
NameTrans /index.html /api/bin/icsextpgm.so:trans_url
```

기본값

없음

NoBG — Caching Proxy 프로세스를 포그라운드에서 실행

Linux 및 UNIX 플랫폼에서 이 지시문을 사용하여 Caching Proxy 서버 프로세스가 포그라운드에서 자동으로 실행되지 않도록 하십시오. 기본으로 off로 설정된 지시문에는 다음과 같은 형식이 있습니다.

NoBG [on | off]

주: **ibmproxy** 명령의 **-nobg** 옵션은 Windows 시스템에서 유효하지 않습니다.

예제

NoBG on

기본값

NoBG off

NoCaching — 템플리트와 일치하는 URL이 있는 파일을 캐시하지 않도록 지정

이 지시문을 사용하여 서버가 지정된 템플리트와 일치하는 URL이 있는 파일은 캐시하지 않도록 지정합니다. 이 지시문에 대한 여러 개의 어커런스를 구성 파일에 포함할 수 있습니다. 각 템플리트에 대한 별개의 지시문을 포함시키십시오. URL 템플리트에는 프로토콜이 있어야 합니다.

CacheOnly 또는 NoCaching 지시문이 설정되어 있지 않으면, 모든 URL이 캐시 대상입니다.

형식

NoCaching *URL_pattern*

예제

NoCaching http://joke/*

기본값

없음

NoLog — 템플리트와 일치하는 고유한 호스트나 도메인에 대한 로그 입력 항목 압축

이 지시문을 사용하여, 지정된 템플리트와 일치하는 고유한 호스트나 도메인의 액세스 요청을 로그하지 않도록 지정하십시오. 예를 들어, 로컬 호스트의 액세스 요청을 로그하지 않을 수 있습니다.

이 지시문에 대한 여러 개의 어커런스를 구성 파일에 포함할 수 있습니다. 또한 템플리트를 하나 이상의 공백으로 분리할 경우, 동일한 지시문에 여러 템플리트를 넣을 수 있습니다. 템플리트에 호스트 이름이나 IP 번호 주소를 사용할 수 있습니다.

주: 호스트 이름 템플리트를 사용하려면, DNS-Lookup 지시문을 On으로 설정해야 합니다. DNS-Lookup 지시문이 Off(기본값)로 설정되면, IP 주소 템플리트만 사용할 수 있습니다.

형식

```
NoLog {host_name | IP_address} [...]
```

예제

```
NoLog 128.0.* *.edu localhost.*
```

기본값

없음

no_proxy — 도메인에 직접 연결하기 위한 템플리트 지정

프록시를 체인하는 데 http_proxy, ftp_proxy 또는 gopher_proxy 지시문을 사용하는 경우, 이 지시문을 사용하여 프록시를 거치지 않고 서버가 직접 연결할 도메인을 지정할 수 있습니다.

도메인 이름이나 도메인 이름 템플리트 문자열로 이 값을 지정합니다. 문자열의 각 입력 항목을 쉼표(,)로 분리하십시오. 문자열에 공백을 사용하지 마십시오.

이 지시문의 템플리트는 다른 지시문과 다르게 입력됩니다. 가장 중요한 점은, 와일드카드 문자(*)를 사용할 수 없는 것입니다. 도메인 이름의 마지막 부분만 포함시켜도 템플리트를 지정할 수 있습니다. 서버는 지정한 템플리트와 일치하는 문자열로 끝나는 도메인에 직접 연결합니다. 이 지시문은 프록시 체인에만 적용되고 SOCKS 구성 파일에 있는 직접 @/= 행과 동등합니다.

형식

```
no_proxy domain_name_or_template[,...]
```

예제

```
no_proxy www.someco.com,.raleigh.ibm.com,.some.host.org:8080
```

이 예제에서 다음 요청에 대해 서버는 프록시를 거치지 않습니다.

- www.someco.com으로 끝나는 도메인에 대한 요청
- blugrass.raleigh.ibm.com 또는 keystone.raleigh.ibm.com과 같이 .raleigh.ibm.com으로 끝나는 도메인에 대한 요청
- myname.some.host.org:8080과 같이 .some.host.org로 끝나는 도메인의 포트 8080에 요청 (기본 포트가 80으로 추정되는 myname.some.host.org와 같이 동일한 도메인의 다른 포트에 대한 요청은 포함되지 않습니다.)

기본값

없음

NoCacheOnRange — 범위 요청에 대해 캐시를 지정하지 않음

기본적으로, 브라우저에서 범위 요청을 수신하는 경우, Caching Proxy에서는 백엔드 서버로부터의 전체 응답이 필요합니다. Caching Proxy는 요청에서 범위 헤더를 제거하고 백엔드 서버로 요청을 전달합니다. 응답이 프록시 서버에서 캐시되면, 동일한 자원에 대한 후속 요청은 범위 요청인지 여부에 상관없이 프록시 서버에서 제공합니다. Caching Proxy의 기본 조치는 성능을 향상시키고 클라이언트에 대한 응답 시간을 더 줄일 수 있습니다. 그러나, 응답이 캐시될 수 없거나, 응답이 매우 큰 경우, 기본 조치는 성능을 저하시킵니다.

범위 요청에 대해 캐시를 지정하지 않는 NoCacheOnRange 지시문을 사용하여 기본 구성을 사용하는 경우, 설명된 문제점을 해결할 수 있습니다.

ibmproxy.conf 파일 전체에서 지시문을 사용 가능하게 하거나 프록시 매핑 규칙에 대한 옵션으로 사용하는 경우, Caching Proxy는 범위 요청 헤더를 백엔드 서버로 전달합니다. 그러나, Caching Proxy는 백엔드 서버에서 206(부분 콘텐츠)을 캐시하지 않습니다.

NoCacheOnRange 지시문을 사용 가능하게 하여, 다음의 경우에 대해 프록시 성능을 향상시킬 수 있습니다.

- 응답을 캐시할 수 없거나 또는 자주 갱신할 수 없는 경우
- 응답 시간이 중요한 응용프로그램의 경우

형식

NoCacheOnRange [on | off]

예제

프록시 맵핑 규칙에서 NoCacheOnRange를 사용 가능하게 할 수 있습니다.

```
Proxy /not-cachable/* http://server.com/no-cachable-resources/* NoCacheOnRange
```

기본값

NoCacheOnRange off

NoProxyHeader — 차단시킬 클라이언트 헤더 지정

이 지시문을 사용하여 차단시킬 클라이언트 URL 헤더를 지정합니다. 필수 헤더를 비롯하여 클라이언트가 전송한 HTTP 헤더는 차단될 수 있습니다. 헤더를 차단할 때 각 별히 주의하십시오. 일반 헤더에는 다음 사항이 포함됩니다.

- **Pragma:**—일반적으로 캐시가 있는 브라우저나 서버가 파일이 요청될 때마다 기점 서버에서 문서를 불러오도록 지시하는 데 사용됩니다.
- **Referer:**—URL을 획득한 파일의 URL

이 헤더와 기타 헤더의 자세한 내용은 HTTP 프로토콜 스펙을 참조하십시오. 이 지시문을 여러 번 지정할 수 있습니다.

형식

NoProxyHeader *header*

예제

NoProxyHeader Referer:

기본값

없음

NumClients — 사용할 캐시 에이전트 스레드의 수 지정

이 지시문을 사용하여 캐시 에이전트가 대기열에서 페이지를 검색하기 위해 사용할 스레드의 수를 지정합니다. 내부 네트워크 및 인터넷 연결 속도에 따라 스레드 수를 정합니다. 허용 가능한 범위는 1 - 100입니다.

주: 여섯 개 이상의 스레드를 사용하면 콘텐츠 서버에 대한 요청이 폭주할 수 있습니다.

형식

NumClients *number*

기본값

NumClients 4

ObjectType — 오브젝트 유형 단계 사용자 정의

이 지시문을 사용하여 오브젝트 유형 단계 동안 서버가 호출하는 사용자 정의된 응용 프로그램 기능을 지정합니다. 이 코드는 파일 시스템에서 요청한 오브젝트를 찾아 MIME 유형으로 지정합니다.

형식

`ObjectType request_template /path/file:function_name`

request_template

응용프로그램 기능이 호출되는지 여부를 추가적으로 판별하는 요청에 대한 템플릿을 지정합니다. 스펙은 프로토콜, 도메인, 호스트를 포함할 수 있고, 앞에 슬래시 (/)가 붙을 수 있으며, 별표(*)를 와일드 카드로 사용할 수 있습니다. 예를 들어, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /* 및 *는 모두 유효합니다.

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

예제

`ObjectType /index.html /api/bin/icsextpgm.so:obj_type`

기본값

없음

OptimizeRuleMapping — 규칙의 수가 증가하는 경우, 수신 요청에 대한 규칙 매핑 프로세스를 최적화

이 지시문은 규칙 수가 증가하는 경우, 수신 요청에 대한 규칙 매핑 프로세스의 속도를 높입니다.

OptimizeRuleMapping 지시문을 사용 가능하게 한 경우, 각 규칙에 대해 수신 URL 요청을 하나씩 매핑하는 대신, 프록시는 접두부 트리에 대해 URI를 매핑합니다. 접두부 트리는 프록시가 매핑 규칙 사이에서 중복되는 문자열 비교를 제거할 수 있도록 합니다. 그 결과로, Caching Proxy는 구성에서 규칙의 수가 300보다 큰 경우, 향상된 성능을 아카이브합니다.

형식

`OptimizeRuleMapping [on | off]`

기본값

`OptimizeRuleMapping off`

OutputTimeout — 출력 시간 종료 지정

이 지시문을 사용하여 서버가 클라이언트에게 출력을 전송하는 데 허용된 최대 시간을 설정합니다. 시간 한계는 로컬 파일의 요청 및 서버가 프록시로 작동하는 요청에 적용됩니다. 시간 한계는 로컬 CGI 프로그램을 시작하는 요청에 적용되지 않습니다.

서버가 이 지시문에 지정된 제한 시간 내에 완료된 응답을 전송하지 않으면, 서버가 연결을 끊습니다. 시간, 분, 초의 결합으로 시간값을 지정하십시오.

형식

OutputTimeout *time*

기본값

OutputTimeout 30 minutes

PacFilePath — PAC 파일이 들어 있는 디렉토리 지정

이 지시문을 사용하여 원격 구성 PAC 파일 양식을 사용하여 생성된 프록시 자동구성 파일이 들어 있는 디렉토리를 지정합니다.

형식

PacFilePath *directory_path*

기본값

- **Windows:** PacFilePath c:\Program Files\IBM\edge\cp\HTML\pacfiles
- **Linux 및 UNIX:** PacFilePath /opt/ibm/edge/cp/server_root/pub/pacfiles

Pass — 요청을 승인하기 위한 템플릿 지정

이 지시문을 사용하여 서버의 파일과 함께 승인하고 응답하려는 요청에 대한 템플릿을 지정합니다. 일단 요청이 Pass 지시문의 템플릿과 일치하면, 이 요청은 후속 지시문의 요청 템플릿과 비교되지 않습니다.

형식

Pass *request_template* [*file_path* [*server_IP_address* | *host_name*]]

request_template

서버가 파일에 대해 승인하고 응답하려는 요청에 대한 템플릿을 지정합니다.

이 템플릿에서 별표(*)를 와일드 카드로 사용할 수 있습니다. 슬래시(/) 바로 뒤의 틸데(tilde) 문자는 정확히 일치해야 합니다. 와일드 카드는 틸데(tilde) 문자(~)를 일치시키는 데 사용될 수 없습니다.

[*file_path*]

서버가 리턴하는 파일의 경로를 지정합니다. *request_template*에 와일드 카드가

있으면, *file_path*에도 와일드 카드를 포함할 수 있습니다. *request_template* 와일드 카드와 일치하는 요청의 부분이 *file_path*의 와일드 카드 대신 삽입됩니다.

이 매개변수는 선택적입니다. 경로를 지정하지 않으면, 요청 자체가 경로로 사용됩니다.

[*server_IP_address* | *host_name*]

여러 IP 주소나 가상 호스트를 사용하고 있다면, 이 매개변수를 사용하여 IP 주소나 호스트 이름을 지정하십시오. 서버는 이 IP 주소나 호스트의 서버에 도달하는 요청이나 이 호스트에 대해서만 지시문을 사용합니다. IP 주소는 요청하는 클라이언트의 주소가 아니라 서버 네트워크 연결의 주소입니다.

IP 주소(예: 240.146.167.72) 또는 호스트 이름(예: hostA.raleigh.ibm.com)을 지정할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수가 없으면, 서버는 요청이 들어오는 IP 주소나 URL의 호스트 이름과 상관없이 모든 요청에 대해 지시문을 사용합니다.

서버 IP 주소로 와일드 카드 문자를 지정할 수 없습니다.

예제

- 다음 예제에서 서버는 운영 체제에 따라 나열된 경로의 파일에 대해 /updates/parts/로 시작하는 요청에 응답합니다. /updates/parts/ 뒤에 오는 어떤 것도 파일을 지정하는 데 사용됩니다.

Linux 및 UNIX 시스템: Pass /updates/parts/* /opt/ibm/edge/cp/server_root/pub/*

Windows 시스템: Pass /updates/parts/* c:\Program Files\IBM\edge\cp\pub*

- 다음 예제에서 서버는 디렉토리 /gooddoc의 파일에 대해 /gooddoc/로 시작하는 요청에 응답합니다. 그러면 서버는 파일 /gooddoc/volume1/issue2/newsletter4.html의 문서에 대한 요청 /gooddoc/volume1/issue2/newsletter4.html에 응답합니다.

Pass /gooddoc/*

- 다음 예제에서는 선택적 IP 주소 매개변수를 사용합니다. 서버가 /parts/로 시작하는 요청을 받으면, 요청이 들어오는 네트워크 연결의 IP 주소에 기초하여 서로 다른 디렉토리의 파일을 리턴합니다. 240.146.167.72에서 들어오는 요청에 대해서, 서버가 /customerA/catalog/의 파일을 돌려보냅니다. 주소가 0.83.100.45인 연결에서 들어오는 요청에 대해서, 서버가 /customerB/catalog/ 파일을 리턴합니다.

Pass /parts/* /customerA/catalog/* 240.146.167.72

Pass /parts/* /customerB/catalog/* 0.83.100.45

- 다음 예제에서는 선택적인 호스트 이름 매개변수를 사용합니다. 서버가 /parts/로 시작하는 요청을 받으면, URL의 호스트 이름에 기초하여 서로 다른 디렉토리의 파

일을 리턴합니다. hostA로 들어오는 요청에 대해서, 서버가 /customerA/catalog/의 파일을 돌려보냅니다. hostB로 들어오는 요청에 대해서, 서버가 /customerB/catalog/의 파일을 돌려보냅니다.

AIX 시스템

```
Pass    /Admin/*    /usr/lpp/internet/server_root/Admin/*
Pass    /Docs/*    /usr/lpp/internet/server_root/Docs/*
Pass    /errorpages/*    /usr/lpp/internet/server_root/pub/errorpages/*
Pass    /*        /usr/lpp/internet/server_root/pub/*
```

Solaris, HP-UX 및 Linux 시스템

```
Pass    /Admin/*    /opt/ibm/edge/cp/server_root/Admin/*
Pass    /Docs/*    /opt/ibm/edge/cp/server_root/Docs/*
Pass    /errorpages/*    /opt/ibm/edge/cp/server_root/pub/errorpages/*
Pass    /*        /opt/ibm/edge/cp/server_root/pub/*
```

기본값

AIX 시스템

```
Pass    /Admin/*    /usr/lpp/internet/server_root/Admin/*
Pass    /Docs/*    /usr/lpp/internet/server_root/Docs/*
Pass    /errorpages/*    /usr/lpp/internet/server_root/pub/errporpages/*
Pass    /*        /usr/lpp/internet/server_root/pub/*
```

HP-UX, Linux 및 Solaris 시스템

```
Pass    /Admin/*    /opt/ibm/edge/cp/server_root/Admin/*
Pass    /Docs/*    /opt/ibm/edge/cp/server_root/Docs/*
Pass    /errorpages/*    /opt/ibm/edge/cp/server_root/pub/errorpages/*
Pass    /*        /opt/ibm/edge/cp/server_root/pub/*
```

Windows 시스템

```
Pass    /icons/*    C:\Program Files\IBM\edge\cp\icons\*
Pass    /Admin/*    C:\Program Files\IBM\edge\cp\Admin\*
Pass    /Docs/*    C:\Program Files\IBM\edge\cp\Docs\*
Pass    /erropages/*    C:\Program Files\IBM\edge\cp\pub\errorpages\*
Pass    /*        C:\Program Files\IBM\edge\cp\pub\*
```

PersistTimeout — 클라이언트가 다른 요청을 전송하기 위한 대기 시간 지정

이 지시문을 사용하여 서버가 지속적인 연결을 취소하기 전에 클라이언트 요청 간에 대기하는 시간을 지정합니다. 시간은 모든 유효한 시간 증가 단위로 지정될 수 있으나, 보통 초 또는 분 단위로 지정됩니다.

서버는 서로 다른 시간 종료 지시문인 InputTimeout을 사용하여 연결이 이루어진 후 클라이언트가 처음 요청을 전송하기 위해 대기하는 기간을 판별합니다. 입력 시간 종료에 대한 자세한 정보는 244 페이지의 『InputTimeout — 입력 시간 종료 지정』을 참조하십시오.

서버는 첫 번째 응답을 전송한 후, PersistTimeout 지시문에 대한 값 설정을 사용하여 지속적인 연결을 취소하기 전에 각 후속 요청을 대기할 기간을 결정합니다.

형식

PersistTimeout *time*

기본값

PersistTimeout 4 seconds

PICSDBLookup — PICS 레이블 검색 단계 사용자 정의

이 지시문을 사용하여 서버가 지정된 URL에 대한 PICS 레이블을 검색하기 위해 호출하는 사용자 정의된 응용프로그램 기능을 지정합니다. 이 기능으로 요청 파일에 대해 동적으로 PICS 레이블을 작성하거나, 대체 파일 또는 데이터베이스에서 PICS 레이블을 탐색할 수 있습니다.

형식

PICSDBLookup */path/file:function_name*

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

예제

PICSDBLookup /api/bin/icsext05.so:get_pics

기본값

없음

PidFile(Linux 및 UNIX 전용) — Caching Proxy의 프로세스 ID를 저장할 파일 지정

Linux 및 UNIX 전용. 이 지시문을 사용하여 Caching Proxy 프로세스 ID가 있는 파일의 위치를 지정합니다. 서버 프로세스가 시작될 때 PID(프로세스 ID)를 파일에 기록합니다. 서버의 여러 인스턴스를 단일 시스템에서 실행 중인 경우, 각 인스턴스에는 자체 PidFile 지시문이 있어야 합니다.

형식

PidFile *path_to_pid_file_info*

예제

PidFile /usr/pidinfo

기본값

- ServerRoot 지시문이 지정된 경우: PidFile *server_root /ibmproxy-pid*
- ServerRoot 지시문이 지정되지 않은 경우: PidFile */tmp/ibmproxy-pid*

PKCS11DefaultCert, PKCS11DriverPath, PKCS11TokenPassword

— IBM 4960 PCI 암호화 액셀러레이터 카드(AIX 전용) 지원

AIX 시스템에서, IBM 4960 PCI 암호화 액셀러레이터 카드를 지원하는 경우, 추가 지시문이 제공됩니다.

세 개의 지시문을 사용하여 프록시가 장치 드라이버를 로드하고 토큰 장치를 열며, 장치에 저장된 인증에 액세스하도록 허용합니다. 장치 드라이버가 로드된 경우, 프록시 서버는 자동으로 장치를 사용하여 SSL 통신 속도를 증가시킵니다.

관련 주제: 304 페이지의 『SSLCryptoCard — 설치된 암호 카드 지정』

형식

PKCS11DefaultCert *default_cert_label*

토큰 장치에 저장된 기본 SSL 인증 레이블을 지정합니다.

PKCS11DriverPath *absolute_path_to_the_card_driver*

암호화 액셀러레이터 카드에 대한 장치 드라이버의 절대 경로를 지정합니다.

PKCS11TokenPassword *password*

암호를 지정하여 토큰 장치를 엽니다.

예제

```
PKCS11DefaultCert MyDefaultCertInTheToken
PKCS11DriverPath /usr/lib/pkcs11/PKCS11_API.so
PKCS11TokenPassword MyPasswordToOpenTheToken
```

기본값

없음

플러그인 모듈 지시문

아래에 나열된 지시문들은 새 기능 및 플러그인의 사용을 위해 Caching Proxy *ibmproxy.conf* 파일에 추가되었습니다. 구성 및 관리 양식은 지시문 대부분을 편집하는 데 사용할 수 없습니다. vi 또는 emacs와 같은 표준 텍스트 편집기는 지시문을 수동으로 편집하는 데 사용해야 합니다. 각각의 새 지시문에 대한 자세한 정보는 이 장에 영문자순으로 표시됩니다.

- 232 페이지의 『ExternalCacheManager — IBM WebSphere Application Server의 동적 캐시에 대한 Caching Proxy 구성』

- 240 페이지의 『ICP_Address — ICP 조회의 IP 주소 지정』
- 241 페이지의 『ICP_Port — ICP 조회의 포트 번호 지정』
- 242 페이지의 『ICP_Timeout — ICP 조회의 최대 대기 시간 지정』
- 241 페이지의 『Occupier — ICP 클러스터의 구성원 지정』
- 240 페이지의 『ICP_MaxThreads — ICP 조회를 위한 최대 스레드 지정』
- 301 페이지의 『SignificantURLTerminator — URL 요청의 종료 코드 지정』
- 303 페이지의 『SSLCertificate — 인증서에 대한 키 레이블 지정』
- 305 페이지의 『SSLOnly — HTTP 요청의 리스너 스레드 사용 불가능』

ibmproxy.conf 파일에서 Caching Proxy 플러그인 모듈을 구성하기 위해 사용된 지시문은 다음 형식으로 입력해야 합니다.

```
<MODULEBEGIN> plugin name
subdirective1
subdirective2
<MODULEEND>
```

각각의 플러그인 프로그램은 ibmproxy.conf 파일을 구문 분석하며 하부 지시문의 자체 고유 블록만 읽습니다. Caching Proxy 구문 분석기는 <MODULEBEGIN>과 <MODULEEND> 사이의 모든 내용을 무시합니다.

Caching Proxy 플러그인 모듈 및 일부 새 기능은 API 지시문이 ibmproxy.conf 파일에 추가되도록 요구합니다. 프록시 서버는 나열된 순서로 플러그인 모듈과 상호작용하므로, 프록시 구성 파일에서 지시문의 순서를 정할 때에는 주의가 필요합니다. 표준 지시문(설명 양식)이 ibmproxy.conf 파일의 API 섹션에 추가되었습니다. API 지시문은 중요도 순으로 나열되어 있습니다. API 지시문을 추가하여 새 기능 및 플러그인 모듈을 사용 가능하게 하는 경우에는 구성 파일의 프로토타입 섹션에 표시된 대로 지시문을 나열하십시오. 아니면 API 지시문의 설명 표시를 제거하고 편집하여 필요한 경우에 각각의 원하는 기능이나 플러그인에 대한 지원을 포함하십시오. 제품과 함께 제공된 모듈 다음에 사용자 생성 플러그인 모듈을 추가하십시오.

Port — 서버가 요청을 인식하는 포트 지정

이 지시문을 사용하여 서버가 요청을 인식하는 포트의 번호를 지정합니다. HTTP에 대한 기본 포트 번호는 80입니다. 1024 미만의 다른 포트 번호는 다른 TCP/IP 응용프로그램에 예약되어 사용해서는 안 됩니다. 프록시 웹 서버에 사용될 일반 포트는 8080 및 8008입니다.

80 이외의 포트가 사용될 때, 클라이언트는 서버에 대한 요청에 대해 고유한 포트 번호를 포함해야 합니다. 포트 번호는 콜론(:) 앞에 오고 URL의 호스트 이름 뒤에 위치합니다. 예를 들어, 브라우저의 URL `http://www.turfco.com:8008/`은 포트 8008에서 인식하고 있는 `www.turfco.com`이라는 호스트에서 기본 환영 페이지를 요청합니다.

ibmproxy 명령의 **-p** 옵션을 사용하여 서버를 시작할 때 이 설정을 덮어쓸 수 있습니다.

형식

Port number

이 지시문을 변경하면, 서버를 직접 정지시킨 다음 재시작해야 변경사항을 적용할 수 있습니다. 서버를 재시작하기만 할 경우에는 서버가 변경사항을 인식하지 않습니다(15 페이지의 제 5 장 『Caching Proxy 시작 및 정지』 참조).

기본값

Port 80

PostAuth — PostAuth 단계 사용자 정의

이 지시문을 사용하여 PostAuth 단계 동안 서버가 호출하는 사용자 정의된 응용프로그램 기능을 지정합니다. 이 코드는 이전 단계나 PostAuth 처리기의 리턴 코드와 상관없이 실행됩니다. 이 코드를 사용하면 요청을 처리하기 위해 할당된 자원을 제거할 수 있습니다.

형식

PostAuth */path/file:function_name*

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

예제

AuthExit */ics/api/bin/icsext05.so:post_exit*

기본값

없음

PostExit — PostExit 단계 사용자 정의

이 지시문을 사용하여 PostExit 단계 동안 서버가 호출하는 사용자 정의된 응용프로그램 기능을 지정합니다. 이 코드는 이전 단계나 PostExit 처리기의 리턴 코드와 상관없이 실행됩니다. 이 코드를 사용하면 요청을 처리하기 위해 할당된 자원을 제거할 수 있습니다.

형식

PostExit */path/file:function_name*

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

예제

```
PostExit      /ics/api/bin/icsext05.so:post_exit
```

기본값

없음

PreExit — PreExit 단계 사용자 정의

이 지시문을 사용하여 PreExit 단계 동안 서버가 호출하는 사용자 정의된 응용프로그램 기능을 지정합니다. 이 코드는 클라이언트 요청을 읽은 후 다른 처리가 아직 발생하기 전에 실행됩니다. 이 단계에서 GoServe 모듈을 호출할 수 있습니다.

형식

```
PreExit /path/file:function_name
```

/path/file

확장자를 포함하는 사용자의 컴파일된 DLL의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

예제

```
PreExit      /ics/api/bin/icsext05.so:pre_exit
```

기본값

없음

Protect — 템플리트와 일치하는 요청에 대한 보호 설정 활성화

이 지시문을 사용하여 템플리트와 일치하는 요청에 대한 보호 설정을 활성화합니다.

주: 보호를 올바르게 작동시키려면, DefProt 및 Protect 지시문을 구성 파일에 있는 모든 Pass, Exec 또는 Proxy 지시문 앞에 위치시켜야 합니다.

보호 설정은 보호 부 지시문으로 정의됩니다. Protect 지시문의 형식은 보호 부 지시문이 들어 있는 레이블이나 파일을 가리킬 것인지, 또는 Protect 지시문의 부분으로 보호 부 지시문 인라인을 포함할 것인지에 따라 달라집니다.

형식

이 매개변수는 다음 양식 중 하나일 수 있습니다.

- Protect 지시문은 보호 부 지시문을 포함하는 별도 파일의 파일 이름 및 전체 경로를 지정할 수 있습니다. 또한 Protection 지시문에서 이전에 정의된 이름과 일치하는 보호 설정 레이블 이름이 지정할 수도 있습니다. Protection 지시문에는 보호 부 지시문이 들어 있습니다. 형식은 다음과 같습니다.

```
Protect request_template [setup_file | label]
    [FOR Server_IP_address | host_name]
```

주: 이 지시문은 여기에는 두 행으로 표시되어 있지만 한 행에 입력되어야 합니다.

- Protect 지시문에 실제 보호 부 지시문 인라인을 지정할 수 있습니다. 부 지시문은 중괄호({})로 묶어야 합니다. 왼쪽 중괄호 문자는 Protect 지시문과 동일한 행의 마지막 문자여야 합니다. 각 부 지시문은 부 지시문 행에 있어야 합니다. 오른쪽 중괄호 문자는 마지막 부 지시문 행 다음의 부 지시문 행에 있어야 합니다. 중괄호 사이에 설명 행이 들어갈 수 없습니다. Protect 지시문의 일부로 보호 부 지시문을 포함하려면 형식은 다음과 같습니다.

```
Protect request_template [FOR Server_IP_address | hhost_name]
    subdirective value
    subdirective value
    .
    .
    .
}
```

다음 매개변수를 사용합니다.

request_template

보호를 활성화시킬 요청에 대한 템플리트를 지정합니다. 서버는 템플리트에 대한 수신 클라이언트 요청을 비교하여 일치되는 것이 있으면 보호를 활성화시킵니다.

[setup_file | label]

보호 부 지시문을 포함하는 레이블이나 파일의 위치를 지정하는 경우, 이 매개변수가 *request_template*과 일치하는 요청을 활성화시키는 보호 설정을 지정합니다.

이 매개변수는 선택적입니다. 이 매개변수가 생략되면, 보호 설정은 일치하는 템플리트를 포함하는 최신의 DefProt 지시문에 의하여 정의됩니다.

[FOR server_IP_address | host_name]

여러 IP 주소나 가상 호스트를 사용하고 있다면, 이 매개변수를 사용하여 IP 주소나 호스트 이름을 지정하십시오. 서버는 이 IP 주소나 호스트의 서버에 도달하는 요청이나 이 호스트에 대해서만 지시문을 사용합니다. IP 주소는 요청하는 클라이언트의 주소가 아니라 서버 네트워크 연결의 주소입니다. IP 주소를 보호하면 IP 주소 및 완전한 호스트 이름이 모두 보호됩니다. 그러나 완전한 호스트 이름이 아닌 이름(예: 호스트 이름 파일의 항목)을 사용하여 해당 네트워크 내에서 서버를 호출할 경우에는 보호되지 않습니다.

예제:

```
Protect http://x.x.x.x PROT-ADMIN
```

웹 브라우저 내에서

- `http://x.x.x.x`는 보호됩니다.
- `http://hostname.example.com`은 보호됩니다.
- `http://hostname`은 보호되지 않습니다.

예제:

`Protect http://hostname.example.com PROT-ADMIN`

웹 브라우저 내에서

- `http://x.x.x.x`는 보호됩니다.
- `http://hostname.example.com`은 보호됩니다.
- `http://hostname`은 보호되지 않습니다.

IP 주소(예: FOR 240.146.167.72) 또는 호스트 이름(예: FOR hostA.bcd.com)을 지정할 수 있습니다.

와일드 카드는 서버 IP 주소를 지정하는 데 사용할 수 없습니다.

이 매개변수는 선택적입니다. 이 매개변수가 없으면, 서버는 요청이 들어오는 IP 주소나 URL의 호스트 이름과 상관없이 모든 요청에 대해 지시문을 사용합니다.

주: `[server_IP_address | host_name]` 매개변수는 `[setup_file | label]` 매개변수 또는 `subdirective value` 매개변수와 함께 사용됩니다.

- `[server_IP_address | host_name]`을 `[setup_file | label]`과 사용하려면, FOR 또는 일부 다른 문자 스트링(공백 없는)을 `[setup_file | label]` 매개변수와 `[server_IP_address | host_name]` 매개변수 사이에 넣어야 합니다.
- `[server_IP_address | host_name]`을 `subdirective value` 매개변수와 사용하려면, `IP_address` 또는 `host_name` 사이에 FOR를 포함하지 마십시오.

subdirective value

Protect 지시문의 일부로 보호 부 지시문을 포함하기 위해 이 매개변수를 사용합니다. 보호 부 지시문에 대한 설명은 다음을 참조하십시오.

- 280 페이지의 『AuthType — 인증 유형 지정』
- 280 페이지의 『DeleteMask — 파일 삭제를 허용한 사용자 이름, 그룹 및 주소 지정』
- 281 페이지의 『GetMask — 파일 가져오기를 허용한 사용자 이름, 그룹 및 주소 지정』
- 281 페이지의 『GroupFile — 연관된 그룹 파일의 위치 지정』

- 281 페이지의 『Mask — HTTP 요청을 허용한 사용자 이름, 그룹 및 주소 지정』
- 281 페이지의 『PasswdFile — 연관된 암호 파일의 위치 지정』
- 282 페이지의 『PostMask — 파일 게시를 허용한 사용자 이름, 그룹 및 주소 지정』
- 282 페이지의 『PutMask — 파일 넣기를 허용한 사용자 이름, 그룹 및 주소 지정』
- 282 페이지의 『ServerID — 암호 파일과 연관시킬 이름 지정』

예제

- 다음 예제에서 서버는 다음과 같이 보호를 활성화시킵니다.
 - /secret/scoop/로 시작하는 요청이 보호를 활성화시킵니다. 보호 설정은 /server/protect/setup1.acc 보호 설정 파일에서 정의됩니다. Protect 지시문이 보호 설정을 지정하지 않으므로, 이전에 일치하는 DefProt 지시문의 보호 설정이 사용됩니다.
 - /secret/business/로 시작하는 요청이 보호를 활성화시킵니다. 보호 설정은 BUS-PROT 레이블이 있는 Protection 지시문에서 정의됩니다.
 - /topsecret/로 시작하는 요청이 보호를 활성화시킵니다. 보호 설정은 Protect 지시문에 직접 포함됩니다.

이 예제에서는 IP 주소를 사용합니다. 서버가 /secret/ 또는 /topsecret/로 시작하는 요청을 수신하면, 요청이 들어오는 네트워크 연결의 IP 주소에 기초하여 요청에 대한 서로 다른 보호 설정을 활성화시킵니다.

- 0.67.106.79에서 들어오는 /secret/ 요청에 대해 서버는 CustomerA-PROT 레이블이 있는 보호 지시문에 정의된 보호 설정을 활성화시킵니다. 0.67.106.79에서 들어오는 /topsecret/ 요청에 대해 서버는 /topsecret/에 대한 Protect 지시문의 인라인에 정의된 보호 설정을 활성화시킵니다.
- 0.83.100.45에서 들어오는 /secret/ 요청에 대해 서버는 CustomerB-PROT 레이블이 있는 보호 지시문에 정의된 보호 설정을 활성화시킵니다. 0.83.100.45에서 들어오는 /topsecret/ 요청에 대해 서버는 /topsecret/에 대한 Protect 지시문의 인라인에 정의된 보호 설정을 활성화시킵니다.

```
Protection BUS-PROT {
    UserID    busybody
    GroupID   webgroup
    AuthType  Basic
    ServerID  restricted
    PasswdFile /docs/WWW/restrict.pwd
    GroupFile  /docs/WWW/restrict.grp
    GetMask   authors
    PutMask   authors
}
DefProt /secret/* /server/protect/setup1.acc
Protect /secret/scoop/*
```



```

Protect /secret/business/* BUS-PROT
Protect /topsecret/* {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/WWW/restrict.pwd
    GroupFile /docs/WWW/restrict.grp
    GetMask topbrass
    PutMask topbrass
}
Pass /secret/scoop/* /WWW/restricted/*
Pass /secret/business/* /WWW/confidential/*
Pass /topsecret/* /WWW/topsecret/*

Protect /secret/* CustomerA-PROT FOR 0.67.106.79
Protect /secret/* CustomerB-PROT FOR 0.83.100.45
Protect /topsecret/* 0.67.106.79 {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/WWW/customer-A.pwd
    GroupFile /docs/WWW/customer-A.grp
    GetMask A-brass
    PutMask A-brass
}
Protect /topsecret/* 0.83.100.45 {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/WWW/customer-B.pwd
    GroupFile /docs/WWW/customer-B.grp
    GetMask B-brass
    PutMask B-brass
}

```

- 다음 예제에서는 가상 호스트를 사용합니다. 서버가 /secret/ 또는 /topsecret로 시작하는 요청을 수신하면, URL의 호스트 이름에 기초하여 요청에 대한 다른 보호 설정을 활성화시킵니다.
 - hostA.bcd.com에 대해 들어오는 /secret/ 요청에 대해 서버는 CustomerA-PROT 레이블이 있는 보호 지시문에 정의된 보호 설정을 활성화시킵니다. hostA.bcd.com에 대해 들어오는 /topsecret/ 요청에 대해 서버는 /topsecret/에 대한 Protect 지시문의 인라인에 정의된 보호 설정을 활성화시킵니다.
 - hostB.bcd.com에 대해 들어오는 /secret/ 요청에 대해 서버는 CustomerB-PROT 레이블이 있는 보호 지시문에 정의된 보호 설정을 활성화시킵니다. hostB.bcd.com에 대해 들어오는 /topsecret/ 요청에 대해 서버는 /topsecret/에 대한 Protect 지시문의 인라인에 정의된 보호 설정을 활성화시킵니다.
 - 프록시된 요청에 대해 서버는 proxy-prot 레이블이 있는 보호 지시문에 정의된 보호 설정을 활성화시킵니다. 예제는 다음과 같습니다.

```

Protect http://host1/* proxy-prot

Protect /secret/* CustomerA-PROT FOR hostA.bcd.com
Protect /secret/* CustomerB-PROT FOR hostB.bcd.com
Protect /topsecret/* hostA.bcd.com {
    AuthType Basic
    ServerID restricted
}

```



```

    PasswdFile /docs/WWW/customer-A.pwd
    GroupFile /docs/WWW/customer-A.grp
    GetMask A-brass
    PutMask A-brass
}
Protect /topsecret/* hostB.bcd.com {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/WWW/customer-B.pwd
    GroupFile /docs/WWW/customer-B.grp
    GetMask B-brass
    PutMask B-brass
}

```

기본값

기본적으로 보호는 `/admin-bin/*` 요청 템플릿이 있는 `Protect` 지시문에 의해 구성 및 관리 양식이 보호됩니다.

Protection — 구성 파일 내에 명명된 보호 설정

이 지시문을 사용하여 구성 파일 안에 보호 설정을 정의합니다. 보호 부 지시문을 사용하여 보호 설정에 이름을 제공하고 보호 유형을 정의할 수 있습니다.

주:

1. 구성 파일에서 `DefProt` 또는 `Protect` 지시문 앞에 `Protection` 지시문을 위치시킵니다.
2. 보호 규칙에서 도메인 이름을 사용하려면 `DNS-Lookup` 지시문을 `on`으로 설정합니다.

형식

```

Protection label_name {
    subdirective value
    subdirective value
    .
    .
    .
}

```

label_name

해당 보호 설정과 연관시킬 이름을 지정합니다. 이 이름은 후속 `DefProt` 및 `Protect` 지시문에서 보호 설정을 지정하는 데 사용할 수 있습니다.

subdirective value

부 지시문은 중괄호(`{ }`)로 묶입니다. 왼쪽 중괄호 문자는 *label_name*과 동일한 행의 마지막 문자여야 합니다. 각 부 지시문은 부 지시문 행에 있어야 합니다. 오른쪽 중괄호 문자는 마지막 부 지시문 행 다음의 부 지시문 행에 있어야 합니다. 중괄호 사이에 설명 행이 들어갈 수 없습니다.

보호 부 지시문에 대한 설명은 『Protection subdirectives — 자원 세트가 보호되는 방법 지정』을 참조하십시오.

예제

```
Protection NAME-ME {
    AuthType Basic
    ServerID restricted
    PasswdFile /WWW/password.pwd
    GroupFile /WWW/group.grp
    GetMask groupname
    PutMask groupname
}
```

기본값

```
Protect /admin-bin/* {
    ServerId      Private_Authorization
    AuthType Basic
    GetMask       All@(*)
    PutMask       All@(*)
    PostMask      All@(*)
    Mask          All@(*)
    PasswdFile    /opt/ibm/edge/cp/server_root/protect/webadmin.passwd
}
```

Protection subdirectives — 자원 세트가 보호되는 방법 지정

다음은 보호 설정에서 사용될 수 있는 보호 부 지시문의 설명입니다. 부 지시문은 영문 자순으로 나열되어 있습니다.

보호 설정은 별도의 파일이나 DefProt, Protect 또는 Protection 지시문의 일부로 구성 파일에 포함할 수 있습니다.

AuthType — 인증 유형 지정

사용자 이름 및 암호에 기초하여 액세스를 제한할 때 이 Protection 부 지시문을 사용하십시오. 클라이언트가 서버로 암호를 전송할 때 사용할 인증 유형을 지정하십시오. 기본 인증(AuthType Basic)이면 암호가 일반 텍스트로 서버에 전송됩니다. 암호는 코드화되지만 암호화되지는 않습니다.

기본값:

AuthType Basic

DeleteMask — 파일 삭제를 허용한 사용자 이름, 그룹 및 주소 지정

이 Protection 부 지시문을 사용하여 보호된 디렉토리에 삭제 요청을 하도록 허가된 사용자 이름, 그룹 및 주소 템플리트를 지정합니다.

예제:

```
DeleteMask authors,(niceguy,goodie)@45.96.3.1,128.0.*.*
```

GetMask — 파일 가져오기를 허용한 사용자 이름, 그룹 및 주소 지정

이 Protection 부 지시문을 사용하여 보호된 디렉토리에 획득 요청을 하도록 허가된 사용자 이름, 그룹 및 주소 템플리트를 지정합니다.

예제:

```
GetMask authors,(niceguy,goodie)@45.96.3.1,128.0.*.*
```

기본값:

```
GetMask All@(*)
```

GroupFile — 연관된 그룹 파일의 위치 지정

이 Protection 부 지시문을 사용하여 보호 설정을 사용할 서버 그룹 파일의 경로 및 파일 이름을 지정합니다. 그러면 서버 그룹 파일 내에 정의된 그룹이 다음과 같이 사용될 수 있습니다.

- 보호 설정의 일부인 마스크 부 지시문(마스크 부 지시문은 DeleteMask, GetMask, Mask, PostMask, PutMask입니다.)
- 보호 설정으로 보호되는 디렉토리의 ACL 파일

예제:

```
GroupFile /docs/etc/WWW/restrict.group
```

Mask — HTTP 요청을 허용한 사용자 이름, 그룹 및 주소 지정

이 부 지시문을 사용하여 허가된 사용자 이름, 그룹, 주소 템플리트를 지정함으로써 다른 마스크 부 지시문에서 다루지 않은 HTTP 요청을 작성하도록 지정합니다.

예제:

```
Mask authors,(niceguy,goodie)@45.96.3.1,128.0.*.*
```

주: Mask 지시문을 사용하는 경우, 마스크에서는 대소문자를 구분합니다. 다음은 사용자 ID에 지정된 Mask 보호의 예제입니다.

```
MASK WEBADM,webadm
```

PasswdFile — 연관된 암호 파일의 위치 지정

사용자 이름 및 암호에 기초하여 액세스를 제한할 때 이 Protection 부 지시문을 사용하십시오. 이 보호 설정을 사용할 암호 파일의 경로 및 파일 이름을 지정합니다.

일부 브라우저는 호스트 내의 보안 카테고리(ServerID)에 의해 사용자 ID와 암호를 캐시하므로, 서버 ID와 암호 파일을 지정할 때 다음의 지침을 따르십시오.

- 동일한 암호 파일을 사용하는 보호 설정은 동일한 서버 ID를 사용합니다.
- 다른 암호 파일을 사용하는 보호 설정은 다른 서버 ID를 사용합니다.

예제:

PasswdFile /docs/etc/WWW/restrict.password

주: 포함된 공백이 암호 파일의 경로나 파일 이름에 들어 있으면, 전체 경로 및 파일 이름은 인용 부호로 묶어야 합니다.

PasswdFile "c:\test this\admin.pwd"

PostMask — 파일 게시를 허용한 사용자 이름, 그룹 및 주소 지정

보안 서버의 경우, 이 Protection 부 지시문을 사용하여 보호된 디렉토리에 게시 요청을 하도록 허가된 사용자, 그룹 및 주소 템플리트를 지정합니다.

예제:

PostMask authors,(niceguy,goodie)@45.96.3.1,128.0.*.*

PutMask — 파일 넣기를 허용한 사용자 이름, 그룹 및 주소 지정

이 Protection 부 지시문을 사용하여 보호된 디렉토리에 PUT 요청을 할 수 있는 권한 있는 사용자, 그룹 및 주소 템플리트를 지정합니다.

예제:

PutMask authors,(niceguy,goodie)@45.96.3.1,128.0.*.*

ServerID — 암호 파일과 연관시킬 이름 지정

사용자 이름 및 암호에 기초하여 액세스를 제한할 때 이 Protection 부 지시문을 사용하십시오. 사용 중인 암호 파일과 연관시키려는 이름을 지정하십시오. 이름은 실제 시스템 이름일 필요가 없습니다.

이름은 요청자에 대한 식별자로 사용됩니다. 다른 보호 설정은 다른 암호 파일을 사용할 수 있으므로, 보호 설정과 연관되는 이름을 가지면 클라이언트가 전송할 암호를 결정하는 데 도움이 될 수 있습니다. 대부분의 클라이언트는 사용자 이름 및 암호에 대한 프롬프트가 나타날 때, 이 이름을 표시합니다.

일부 브라우저는 호스트 내의 보안 카테고리(ServerID)에 의해 사용자 ID와 암호를 캐시하므로, 서버 ID와 암호 파일을 지정할 때 다음의 지침을 따르십시오.

- 동일한 암호 파일을 사용하는 보호 설정은 동일한 서버 ID를 사용합니다.
- 다른 암호 파일을 사용하는 보호 설정은 다른 서버 ID를 사용합니다.

예제:

ServerID restricted

Proxy — 프록시 프로토콜 또는 역방향 프록시 지정

이 지시문을 사용하여 Caching Proxy가 처리할 프로토콜을 지시하고 요청을 서버에 맵핑합니다. 유효한 프로토콜은 http, ftp 및 gopher입니다.

프록시 지시문은 요청을 원격 서버로 전달합니다. 예를 들어, 이러한 전달로 모든 요청을 지정된 URL로 전달할 수 있습니다.

```
Proxy /* http://proxy.server.name/*
```

보안 역방향 프록시 서버의 경우, 다음 지시문을 사용하십시오.

```
Proxy /* https://proxy.server.name/*
```

프록시 서버가 덜 제한적이라도 하려면 구성 파일에서 다음 지시문의 설명 표시를 제거하십시오. 그러나 이 지시문은 프록시가 역방향 프록시로 구성될 때 보안 문제점을 발생시킬 수 있습니다.

```
Proxy http:*
```

```
Proxy ftp:*
```

```
Proxy gopher:*
```

선택적 매개변수:

- UseSession

이는 역방향 프록시 구성에만 적용합니다.

이 옵션을 통해 Caching Proxy가 클라이언트 측 소켓 및 전송 소켓 간 일대일 맵핑을 유지보수하도록 지시합니다. 이 옵션은 프록시가 서버 측 소켓의 활성화를 유지하고 동일한 클라이언트 측 소켓에서 오는 요청에 대한 소켓을 재사용하도록 하는 연결 기반 인증과 같은 일부 응용프로그램에서 유용합니다.

- NoCaching

프록시 규칙이 일치하는 경우, 이 옵션은 프록시가 해당하는 응답을 캐시하지 않도록 지시합니다.

- NoCacheOnRange

프록시 규칙이 일치하고 범위 헤더가 요청에 있는 경우, 이 옵션은 프록시가 해당하는 응답을 캐시하지 않도록 지시합니다. 자세한 정보는 264 페이지의 『NoCacheOnRange — 범위 요청에 대해 캐시를 지정하지 않음』을 참조하십시오.

- NoJunction

이는 역방향 프록시 구성에만 적용합니다.

결합 재작성 플러그인이 사용 가능한 경우 이 옵션을 사용하십시오. 이 옵션은 수신 URL이 일치되는 경우, 프록시가 해당하는 응답을 다시 작성하도록 허용하지 않습니다. 자세한 정보는 47 페이지의 『결합 재작성 사용 가능(선택적)』 및 48 페이지의 『JunctionPrefix 옵션으로 결합 정의(권장하는 메소드)』를 참조하십시오.

- JunctionPrefix

이는 역방향 프록시 구성에만 적용합니다.

결합 재작성 플러그인이 사용 가능한 경우 이 옵션을 사용하십시오. 결합 접두부를 프록시 규칙의 첫 URL 패턴에서 추론하는 대신, 옵션은 결합 재작성 접두부를 명확히 선언합니다. 자세한 정보는 47 페이지의 『결합 재작성 사용 가능(선택적)』 및 48 페이지의 『JunctionPrefix 옵션으로 결합 정의(권장하는 메소드)』를 참조하십시오.

형식

```
Proxy request_template target_server_path [[ip]:port]
[UseSession | NoCaching | NoCacheOnRange | NoJunction | JunctionPrefix:/url_prefix]
```

예제

다음은 Proxy 지시문에 대한 UseSession 옵션의 예제입니다.

```
Proxy /abc/* http://server1/default/abc/* :80 UseSession
```

수신 클라이언트 요청이 포트 80에서 오고, 클라이언트 요청의 URL이 패턴 /abc/*과 일치하는 경우, URL은 http://server1/default/abc/*로 매핑됩니다.

기본값

없음.

ProxyAccessLog — 프록시 액세스 로그 파일에 대한 경로 이름 지정

이 지시문을 사용하여 서버가 프록시 요청에 대해 액세스 통계를 로그하려는 파일의 이름 및 경로를 지정합니다. 기본적으로 서버는 클라이언트 요청에 대해 프록시로 작동할 때마다 이 로그에 입력 항목을 기록합니다. 특정 클라이언트의 요청을 로그하지 않으려면, NoLog 지시문을 사용할 수 있습니다.

서버가 실행 중이면 매일 자정에 새 로그 파일을 시작합니다. 서버가 실행하지 않고 있으면, 해당 날짜에 로그 파일을 처음 시작할 때 새 로그 파일을 시작합니다. 파일을 작성할 때, 서버는 지정한 파일 이름을 사용하여 날짜 접미부나 확장자를 추가합니다. 날짜 접미부나 확장자는 Mmddyyyy 형식으로 되어 있으며, 여기서 Mmm은 월의 처음 세 글자이고, dd는 해당 월의 일이며, yyyy는 년도입니다.

이전 로그 파일은 하드 드라이브의 상당한 공간을 차지하므로, 제거하는 것이 좋습니다.

형식

```
ProxyAccessLog path/file
```

기본값

- **Linux 및 UNIX 시스템:** ProxyAccessLog /opt/ibm/edge/cp/server_root/logs/proxy
- **Windows 시스템:** ProxyAccessLog drive:\Program Files\IBM\edge\cp\logs\proxy

ProxyAdvisor — 프록시 요청의 서비스 사용자 정의

이 지시문을 사용하여 프록시 어드바이저 단계 동안 서버가 호출하기를 원하는 사용자 정의된 응용프로그램을 지정합니다. 이 코드는 요청을 제공합니다.

형식

ProxyAdvisor */path/file:function_name*

/path/file

사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

예제:

ProxyAdvisor /api/bin/customadvise.so:proxyadv

기본값

없음

ProxyForwardLabels — PICS 필터링 지정

ProxyForwardLabels 지시문을 사용하여 프록시 서버 및 클라이언트 또는 프록시 계층의 두 프록시에서 PICS 필터링을 지정합니다.

ProxyForwardLabels가 on으로 설정되면, 프록시 서버는 발견된 모든 PICS 레이블에 대해 PICS 레이블: HTTP 헤더를 생성합니다. 여기에는 기점 서버의 레이블, 레이블 기관, Caching Proxy의 레이블 캐시 및 레이블 공급자 플러그인이 포함됩니다.

ProxyForwardLabels가 Off로 설정되면, PICS-Label: HTTP 헤더가 생성되지 않습니다.

형식

ProxyForwardLabels {on | off}

기본값

ProxyForwardLabels Off

ProxyFrom — 클라이언트를 From: 헤더로 지정

이 지시문을 사용하여 From: 헤더를 생성합니다. 이는 일반적으로 프록시 관리자의 전자 우편 주소를 제공하는 데 사용됩니다.

형식

ProxyFrom *e-mail_address*

예제

ProxyFrom webmaster@proxy.ibm.com 설정은 다음과 같이 헤더 변경이 발생합니다.

원래 헤더

Location: http://www.ibm.com/
Last Modified: Tue 5 Nov 1997 10:05:39
GMT
Pragma: no-cache

변경된 헤더

Location: http://www.ibm.com/
Last Modified: Tue 5 Nov 1997 10:05:39
GMT
From: webmaster@proxy.ibm.com
Pragma: no-cache

기본값

없음

ProxyIgnoreNoCache — 재로드 요청 무시

이 지시문을 사용하여 사용자가 브라우저에서 재로드를 누를 때 서버가 반응하는 방법을 지정하십시오. ProxyIgnoreNoCache 지시문이 on으로 설정되면, 로드가 많이 걸리는 동안 서버는 대상 서버의 페이지를 요청하지 않으며, 사용 가능할 경우에는 파일의 캐시 사본을 공급합니다. 서버는 브라우저에서 전송된 Pragma: no-cache 헤더를 무시합니다.

형식

ProxyIgnoreNoCache {on | off}

기본값

ProxyIgnoreNoCache off

ProxyPersistence — 지속적인 연결 허용

이 지시문을 사용하여 클라이언트와 지속적인 연결을 유지할지 여부를 지정합니다. 지속적인 연결은 사용자에게 대기 시간을 줄여주고 프록시 서버에서 CPU 로드를 줄여주지만, 많은 자원을 필요로 합니다. 지속적인 연결을 위해서는 보다 많은 스레드가 필요하며 이에 따라 프록시 서버 메모리도 늘어나야 합니다.

프록시 중에 HTTP 1.1을 따르지 않는 것이 있으면, 다중 레벨 프록시 서버 설정에 지속적인 연결을 사용해서는 안됩니다.

형식

ProxyPersistence {on | off}

기본값

ProxyPersistence on

ProxySendClientAddress — Client IP Address: 헤더 생성

이 지시문을 사용하여 프록시가 클라이언트의 IP 주소를 대상 서버로 전달할지 여부를 지정합니다.

형식

ProxySendClientAddress {*Client_IP*: | OFF}

예제

지시문 ProxySendClientAddress *Client-IP*:는 다음과 같이 헤더 변경이 발생합니다.

원래 헤더

Location: http://www.ibm.com/
Last Modified: Tue 5 Nov 1997 10:05:39
GMT
Pragma: no-cache

변경된 헤더

Location: http://www.ibm.com
Last Modified: Tue 5 Nov 1997 10:05:39
GMT
Client-IP: 0.67.199.5
Pragma: no-cache

기본값

없음

ProxyUserAgent — 사용자 에이전트 문자열 수정

이 지시문을 사용하여 클라이언트가 전송하는 문자열을 대체하는 사용자 에이전트 문자열을 지정하십시오. 이 지시문을 사용하면 웹 사이트 방문시 익명성을 높일 수 있습니다. 그러나 일부 사이트에는 사용자 에이전트 문자열에 따라 정의된 페이지가 있습니다. ProxyUserAgent 지시문을 사용하면 사용자 정의 페이지가 표시되는 것을 방지합니다.

형식

ProxyUserAgent *product_name/version*

예제

ProxyUserAgent Caching Proxy/6.1 지시문은 다음과 같이 헤더 변경이 발생합니다.

원래 헤더

Location: http://www.ibm.com/
Last Modified: Tue 5 Nov 1997 10:05:39
GMT
User Agent: Mozilla/ 2.02 OS2
Pragma: no-cache

변경된 헤더

Location: http://www.ibm.com
Last Modified: Tue 5 Nov 1997 10:05:39
GMT
User Agent: Caching Proxy/6.1
Pragma: no-cache

기본값

없음

ProxyVia — HTTP 헤더의 형식 지정

이 지시문을 사용하여 HTTP 헤더의 형식을 제어하십시오. 이 지시문에 대해 4가지 가능한 값이 있습니다. ProxyVia가 Full로 설정되면, Caching Proxy가 Via 헤더를 요청이나 응답에 추가합니다. Via 헤더가 이미 스트림에 있으면 Caching Proxy가 호스트 정보를 끝에 추가합니다. Set으로 설정되면 Caching Proxy가 Via 헤더를 호스트 정보로 설정합니다. Via 헤더가 이미 스트림에 있으면 Caching Proxy가 헤더를 제거합니다. Pass로 설정되면 Caching Proxy가 헤더 정보를 그대로 전달합니다. Block으로 설정되면 Caching Proxy는 Via 헤더를 전달하지 않습니다.

형식

ProxyVia {Full | Set | Pass | Block}

예제

ProxyVia Pass

기본값

ProxyVia Full

ProxyWAS — WebSphere Application Server로 요청 전송 지정

이는 역방향 프록시 구성에만 적용합니다.

ProxyWAS 맵핑 지시문은 Proxy 지시문에 동일하게 작용하지만 또한 일치하는 요청이 WebSphere Application Server에 지정됨을 Caching Proxy에 표시합니다. 이 지시문 사용에 대한 예는 282 페이지의 『Proxy — 프록시 프로토콜 또는 역방향 프록시 지정』을 참조하십시오.

형식

ProxyWAS *request_template target_server_path* [[*ip*]:*port*]
[UseSession | NoCaching | NoCacheOnRange | NoJunction | JunctionPrefix:*url_prefix*]

기본값

없음

PureProxy — 전용 프록시 사용 불가능

이 지시문을 사용하여 서버가 프록시 서버로 작동할지 또는 프록시 및 콘텐츠 서버로 작동할지 여부를 지정합니다. Caching Proxy를 프록시로만 사용하는 것이 좋습니다.

형식

PureProxy {on | off}

기본값

PureProxy on

PurgeAge — 로그의 유효 기간 한계 지정

이 지시문을 사용하여 로그가 폐기되기 전에 로그의 유효 기간(단위: 일)을 지정하십시오. PurgeAge가 0이면 로그는 삭제되지 않습니다.

주: 플러그인은 당일 또는 전일에 대한 로그는 삭제하지 않습니다.

형식

PurgeAge *number*

기본값

PurgeAge 7

관련 지시문

- 213 페이지의 『CompressAge — 로그 압축 시기 지정』
- 214 페이지의 『CompressDeleteAge — 로그 삭제 시기 지정』
- 213 페이지의 『CompressCommand — 압축 명령 및 매개변수 지정』
- 260 페이지의 『Midnight — 로그 보존에 사용되는 API 플러그인 지정』
- 251 페이지의 『LogArchive — 로그 보존 작동 지정』
- 『PurgeSize — 로그 보존 크기의 한계 지정』

PurgeSize — 로그 보존 크기의 한계 지정

이 지시문을 사용하여 로그 보존이 폐기되기 전에 늘릴 수 있는 로그 파일의 크기(MB)를 지정합니다. PurgeSize 지시문이 0이면, 크기에는 한계가 없으며 파일이 삭제되지 않습니다.

PurgeSize 설정에 대해서는 로그 유형의 모든 로그를 참조하십시오. 인스턴스에 대해 오류를 로그 중이고(즉, 구성 파일에 작성된 ErrorLog 입력 항목이 있을 경우) PurgeSize가 10MB로 지정된 경우, Caching Proxy는 모든 오류 로그의 크기를 계산하여 모두 추가한 후, 전체 크기가 10보다 작아질 때까지 로그를 삭제합니다.

주: 플러그인은 당일 또는 전일에 대한 로그는 삭제하지 않습니다. 로그 파일을 삭제할 때에는 각 로그 유형의 로그 파일 크기(MB)가 PurgeSize에서 정의된 값과 같거나 미만이 될 때까지 가장 오래된 로그부터 삭제합니다.

형식

PurgeSize *number_of_MB*

기본값

PurgeSize 0

관련 지시문

- 213 페이지의 『CompressAge — 로그 압축 시기 지정』
- 214 페이지의 『CompressDeleteAge — 로그 삭제 시기 지정』
- 213 페이지의 『CompressCommand — 압축 명령 및 매개변수 지정』
- 251 페이지의 『LogArchive — 로그 보존 작동 지정』
- 260 페이지의 『Midnight — 로그 보존에 사용되는 API 플러그인 지정』
- 289 페이지의 『PurgeAge — 로그의 유효 기간 한계 지정』

RCAConfigFile — ConfigFile의 별명 지정

이 지시문을 사용하여 원격 캐시 액세스 구성 파일의 이름 및 위치를 지정합니다.

주: RCA 구성 파일이 ibmproxy.conf 파일로 병합되었습니다. 역호환에 대해서는 RCAConfigFile이 ConfigFile에 대한 별명으로 지원됩니다.

형식

RCAConfigFile */etc/file_name*

예제

RCAConfigFile */etc/user2rca.conf*

기본값

RCAConfigFile */etc/rca.conf*

RCAThreads — 포트당 스레드 수 지정

이 지시문을 사용하여 RCA 포트에서 작동하는 스레드의 수를 지정하십시오.

형식

RCAThreads *number_of_threads*

예제

RCAThreads 50

기본값

MaxActiveThreads x [(ArraySize -1) / (2 x ArraySize -1)]

ReadTimeout — 연결의 시간 한계 지정

이 지시문을 사용하여 연결이 취소되기 전에 네트워크 활동 없이 허용되는 시간 한계를 지정합니다.

형식

ReadTimeout *time*

기본값

ReadTimeout 5 minutes

Redirect — 다른 서버에 전송된 요청에 대한 템플리트 지정

이 지시문을 사용하여 다른 서버로 승인하고 전송하려는 요청에 대한 템플리트를 지정합니다. 일단 요청이 Redirect 지시문의 템플리트와 일치하면, 이 요청은 구성 파일에 있는 다른 지시문의 템플리트와 비교되지 않습니다.

형식

Redirect *request_template* URL [*server_IP_address* | *host_name*]

request_template

서버가 다른 서버로 전송할 요청에 대한 템플리트를 지정합니다.

이 템플리트에서 별표(*)를 와일드 카드로 사용할 수 있습니다. 슬래시(/) 바로 뒤의 틸데(tilde) 문자는 정확히 일치해야 합니다. 와일드 카드는 틸데(tilde) 문자(~)를 일치시키는 데 사용할 수 없습니다.

URL

서버가 다른 서버에 전송하는 URL 요청을 지정합니다. 이 요청에 대한 응답은 응답이 사용자 서버에서 온 것임을 알리지 않고 원래 요청자로 이동합니다.

URL에는 프로토콜 스펙 및 요청을 전송할 서버의 이름이 있어야 합니다. 또한 경로 또는 파일 이름이 있을 수 있습니다. *request_template*가 와일드 카드를 사용하면, URL의 경로나 파일 이름도 와일드 카드를 사용할 수 있습니다. *request_template* 와일드 카드와 일치하는 원래 요청 부분이 URL의 와일드 카드 대신 삽입됩니다.

[*server_IP_address* | *host_name*]

여러 IP 주소나 가상 호스트를 사용하고 있다면, 이 매개변수를 사용하여 IP 주소나 호스트 이름을 지정하십시오. 서버는 이 IP 주소나 호스트의 서버에 도달하는 요청이나 이 호스트에 대해서만 지시문을 사용합니다. IP 주소는 요청하는 클라이언트의 주소가 아니라 서버 네트워크 연결의 주소입니다.

IP 주소(예: 240.146.167.72) 또는 호스트 이름(예: hostA.bcd.com)을 지정할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수가 없으면, 서버는 요청이 들어오는 IP 주소나 URL의 호스트 이름과 상관없이 모든 요청에 대해 지시문을 사용합니다.

서버 IP 주소로 와일드 카드 문자를 지정할 수 없습니다.

예제

- 다음 예제에서는 서버에서 /chief/stuff/로 시작하는 모든 요청을 www.other.org 서버의 wahoo 디렉토리로 전송합니다.

```
Redirect /chief/stuff/* http://www.other.org/wahoo/*
```

- 다음 예제에서는 선택적 IP 주소 매개변수를 사용합니다. 서버가 /stuff/로 시작하는 요청을 받으면, 서버는 이 요청의 경로를 요청이 들어오는 네트워크 연결의 IP 주소를 기초로 다른 서버에 재지정합니다. 240.146.167.72로 들어오는 요청의 경우, 서버에서 요청을 www.chief.org 서버의 wahoo 디렉토리로 전송합니다. 주소가 0.83.100.45인 모든 연결로 들어오는 요청의 경우, 서버에서 요청을 www.dawg.com 서버의 파운드 디렉토리로 전송합니다.

```
Redirect /stuff/* http://www.chief.org/wahoo/* 240.146.167.72
Redirect /stuff/* http://www.dawg.com/pound/* 0.83.100.45
```

- 다음 예제에서는 선택적 IP 주소 매개변수를 사용합니다. 서버에서 /stuff/로 시작하는 요청을 수신하면, URL 호스트 이름에 기반한 다른 서버로 요청의 방향을 전환합니다. hostA로 들어오는 요청의 경우, 서버에서 요청을 www.chief.org 서버의 wahoo 디렉토리로 전송합니다. hostB로 들어오는 요청의 경우, 서버에서 요청을 www.chief.org 서버의 파운드 디렉토리로 전송합니다.

```
Redirect /stuff/* http://www.chief.org/wahoo/* hostA.bcd.com
Redirect /stuff/* http://www.dawg.com/pound/* hostB.bcd.com
```

기본값

없음

RegisterCacheIdTransformer — 쿠키 헤더를 기반으로 하는 자원에 하나 이상 변형을 캐시

이 지시문을 사용하여 Caching Proxy가 쿠키 헤더를 기반으로 하는 자원(URI)에 하나 이상 변형을 캐시하도록 허용합니다.

주: 쿠키가 클라이언트 브라우저에서 사용 불가능한 경우, 클라이언트는 동일한 캐시 오브젝트를 액세스할 수 있습니다.

자세한 내용은 308 페이지의 『SupportVaryHeader — HTTP Vary 헤더를 기반으로 하는 자원에 하나 이상의 변형을 캐시』를 참조하십시오.

형식

```
RegisterCacheIdTransformer Cookie cookie-name
```

*cookie-name*은 클라이언트의 요청에서 쿠키 헤더의 이름입니다.

예제

```
RegisterCacheIdTransformer Cookie Usergroup
```

SupportVaryHeader와 함께 이 지시문을 사용하는 예를 보려면, 308 페이지의 『SupportVaryHeader — HTTP Vary 헤더를 기반으로 하는 자원에 하나 이상의 변형을 캐시』를 참조하십시오.

기본값

없음

ReversePass — 자동으로 재지정된 요청 교차

이는 역방향 프록시 구성에만 적용합니다.

ReversePass 맵핑 지시문은 자동 재지정의 결과로 재작성된 요청을 발견하기 위해 서버 응답 스트림을 검토합니다. 일반적으로 서버가 3xx 클래스의 HTTP 코드를 리턴하면(예: 301, 영구적으로 이동한 경우 또는 303의 경우에는 다른 HTTP 참조) 요청 클라이언트로 하여금 정확한 URL 및 IP 주소로 앞으로의 요청을 지정하도록 지시하는 답장 메시지를 서버가 전송합니다. 역방향 프록시 설정의 경우에는 기점 서버로부터의 재지정 메시지가 클라이언트 브라우저로 하여금 연속 요청에 대한 프록시 서버를 생략하도록 할 수 있습니다. 클라이언트가 기점 서버에 직접 접속하지 못하게 하려면, ReversePass 지시문을 사용하여 기점 서버에 고유하게 작성되는 요청을 교차하게 하십시오.

요청 스트림을 처리하는 다른 맵핑 지시문과는 달리 ReversePass는 해당 템플릿을 응답 스트림에 일치시킵니다. 응답 스트림은 프록시 서버가 기점 서버로부터 받아 클라이언트로 전송하는 응답입니다.

형식

ReversePass *rewritten_URL proxy_URL* [*host:port*]

host:port 옵션을 사용하여 프록시가 백엔드 서버 호스트 이름 및 포트를 기반으로 하는 다른 ReversePass 규칙을 적용하도록 허용합니다.

예제

- 다음 예제 명령문을 사용하면 기점 서버로의 직접 요청을 방어할 수 있습니다.

```
ReversePass http://backend.company.com:9080/* http://edge.company.com/*
```

포트 9080은 Application Service at the Edge의 기본 포트입니다. 이 요청 유형은 기점 Application Server가 3xx 코드를 클라이언트에 리턴할 때 생성됩니다.

- 다음 예제 명령문은 에지 Application Server에서 301 코드가 재지정한 요청을 발견합니다.

```
ReversePass http://edge.company.com:9080/* http://edge.company.com/*
```

주: 와일드 카드(*)를 포함하여 *proxy_URL* 패턴의 콘텐츠는 정확하게 백엔드 서버가 전송하는 것과 일치해야 합니다. 그렇지 않은 경우, 지시문은 실패합니다.

기본값

없음

RewriteSetCookieDomain — 재작성해야 할 도메인 패턴 지정

이는 역방향 프록시 구성에만 적용합니다.

이 지시문을 사용하여 재작성해야 할 도메인 패턴을 지정합니다. 이 지시문은 도메인을 *domain_pattern1*에서 *domain_pattern2*로 변환합니다.

형식

`RewriteSetCookieDomain domain_pattern1 domain_pattern2`

예제

`RewriteSetCookieDomain .internal.com .external.com`

기본값

없음

관련 지시문

- 246 페이지의 『JunctionRewriteSetCookiePath — JunctionRewrite 플러그인과 사용되는 경우, Set-Cookie 헤더에 경로 옵션을 재작성』

RTSPEnable — RTSP 경로 재지정 사용 가능

이는 역방향 프록시 구성에만 적용합니다.

이 지시문을 사용하면 RTSP 경로 재지정을 사용 가능하게 하거나 사용 불가능하게 할 수 있습니다. 옵션은 on 또는 off입니다

형식

`RTSPEnable {on | off}`

예제

`RTSPEnable on`

기본값

없음

rtsp_proxy_server - 경로 재지정 서버 지정

이는 역방향 프록시 구성에만 적용합니다.

이 지시문은 경로 재지정된 요청을 수신하는 RTSP 프록시 서버를 지정하는 데 사용됩니다. 다른 유형의 스트림에는 다른 서버를 지정할 수 있습니다. 지시문 형식은 다음과 같습니다.

`rtsp_proxy_server server dns address[:port] default rank [list of mime types]`

예제

rtsp_proxy_server	rproxy.mycompany.com:554	1
rtsp_proxy_server	fw1.mycompany.com:554	2
rtsp_proxy_server	fw1.mycompany.com:555	3
rtsp_proxy_server	fw2.mycompany.com:557	4

기본값

없음

rtsp_proxy_threshold — 캐시로의 경로 재지정 이전에 요청 수 지정

이는 역방향 프록시 구성에만 적용합니다.

이 지시문은 RTSP 요청이 기점 서버가 아닌 프록시 서버로 경로 재지정되기 전에 수신될 요청의 수를 지정합니다. RealNetworks는 첫 번째 요청에 대해 캐시 스트림을 프록시하고, 캐시는 초기에 스트림을 수신하는 대역폭을 두 배로 늘립니다. 2개 이상의 임계치를 지정하면 캐시 도중 요청이 한 번만 수행되는 것을 막을 수 있습니다. 지시문 형식은 다음과 같습니다.

```
rtsp_proxy_threshold number_of_hits
```

예제

```
rtsp_proxy_threshold 5
```

기본값

없음

rtsp_url_list_size — 프록시 메모리에서 URL 수 지정

이는 역방향 프록시 구성에만 적용합니다.

이 지시문은 경로 재지정용 메모리에 보존된 고유한 URL 수를 지정합니다. 프록시는 이 목록을 참조하여 주어진 URL이 이전에 발생했는지 여부를 판별합니다. 목록 크기가 크면 프록시 서버가 이전 요청을 수신한 동일한 프록시 서버로 후속 요청을 전송할 수 있는 능력이 향상되지만, 각 목록의 입력 시 대략 16바이트를 사용합니다.

형식

```
rtsp_url_list_size size_of_list
```

예제

```
rtsp_url_list_size 8192
```

기본값

없음

RuleCaseSense — 대소문자를 구분하지 않는 응용프로그램 URL에서 요청을 맵핑

기본적으로, Caching Proxy가 ibmproxy.conf 파일에서 정의된 규칙에 대한 요청을 맵핑하는 경우, 일치 프로세스에서는 대소문자를 구분합니다. 그러나 일부 응용프로그램 URL은 대소문자를 구분하지 않습니다. 이 요청을 올바르게 핸들하기 위해, RuleCaseSense 지시문이 제공됩니다. 지시문이 off로 지정된 경우, 프록시는 대소문자를 구분하지 않고, 요청을 일치시킵니다.

주: 글로벌 지시문인 경우, 정의된 모든 맵핑 규칙에 적용됩니다.

형식

```
RuleCaseSense {on | off}
```

기본값

```
RuleCaseSense on
```

ScriptTimeout – 스크립트의 시간 종료 설정 지정

이 지시문을 사용하여 서버에 의해 시작된 CGI 프로그램을 종료하는 데 허용되는 시간을 설정합니다. 시간이 만기되면, 서버에서 프로그램을 종료합니다. Linux 및 UNIX 플랫폼에서 프로그램 종료는 KILL 신호로 완료됩니다.

시간(hours), 분(mimutes 또는 mims), 초(seconds 또는 secs)를 사용하여 시간값을 입력하십시오.

형식

```
ScriptTimeout timeout
```

기본값

```
ScriptTimeout 5 minutes
```

SendHTTP10Outbound — 프록시된 요청에 대한 프로토콜 버전 지정

이 지시문을 사용하여 Caching Proxy에서 다운스트림 서버로 전송된 요청이 HTTP 버전 1.0 프로토콜을 사용할지를 지정하십시오. (다운스트림 서버는 프록시 체인에 있는 다른 프록시 서버이거나 요청을 처리할 기점 서버입니다.)

이 지시문이 사용되면 Caching Proxy는 HTTP 1.0을 요청 행의 프로토콜로 식별합니다. HTTP 1.0 고유 기능과 HTTP 1.0 서버가 지원하는 cache-control 헤더와 같은 특정 HTTP 1.1 기능만 다운스트림 서버로 전송됩니다. 하위 서버가 HTTP 1.1 요청을 올바르게 처리하지 않을 경우에 이 지시문을 사용하십시오.

SendHTTP10Outbound 지시문이 지정되지 않을 경우, Caching Proxy가 HTTP 1.1 을 요청 행의 프로토콜로 식별합니다. 지속적인 연결과 같은 HTTP 1.1 기능도 이 요청에서 사용될 수 있습니다.

형식

SendHTTP10Outbound *url_pattern*

예제

이 지시문은 여러 번 지정할 수 있습니다. 예제는 다음과 같습니다.

```
SendHTTP10Outbound http://www.hosta.com/*
SendHTTP10Outbound http://www.hostb.com/*
```

역호환의 경우, SendHTTP10Outbound의 이전 구문은 다음과 같이 처리됩니다.

- SendHTTP10Outbound on은 SendHTTP10Outbound *가 지정된 것처럼 처리됩니다.
- SendHTTP10Outbound off는 무시됩니다.

주: SendHTTP10Outbound off 및 SendHTTP10Outbound *url_pattern*이 모두 지정 되면, SendHTTP10Outbound off는 무시되고 경고 메시지가 발행됩니다.

기본값

없음

SendRevProxyName — HOST 헤더에 Caching Proxy 호스트 이름 지정

이는 역방향 프록시 구성에만 적용합니다.

Caching Proxy는 역방향 프록시로 기능 시 클라이언트에서 HTTP 요청을 수신해서 이를 기점 서버로 전송합니다. 기본적으로 Caching Proxy는 기점 서버로 전송하는 요청의 HOST 헤더에 기점 서버의 호스트 이름을 작성합니다. SendRevProxyName 지시문을 yes로 설정하면 Caching Proxy가 대신 HOST 헤더에 고유한 호스트 이름을 작성합니다. 이 지시문은 하나의 백엔드 서버에서 다른 서버로 요청을 지정하는 경우에도 기점 서버로의 요청이 프록시 서버에서 발생하는 것처럼 언제나 표시되도록 하기 때문에, 백엔드 서버의 특별 구성을 사용 가능하게 하는 데 사용할 수 있습니다.

이 지시문은 다음과 같은 점에서 ReversePass 맵핑 지시문과 다릅니다. ReversePass 지시문은 특정 구문이 포함된 요청을 교차하여 사용자가 지정하는 다른 요청 콘텐츠를 대체합니다. SendRevProxyName 지시문은 오직 기점 서버 호스트 이름의 Caching Proxy 호스트 이름을 대체하도록 설정할 수 있습니다. 이 지시문은 Application Service at the Edge 구성에는 유용하지 않습니다.

형식

SendRevProxyName {yes | no}

ServerConnGCRun — 가비지 콜렉션 스레드 실행 간격 지정

이 지시문은 가비지 콜렉션 스레드가 시간 종료된 서버 연결을 확인하는 간격을 설정합니다(ServerConnTimeout 지시문으로 설정). ServerConnPool 지시문이 on으로 설정된 경우에만 이 지시문을 사용하십시오.

형식

ServerConnGCRun *time_interval*

예제

ServerConnGCRun 2 minutes

기본값

ServerConnGCRun 2 minutes

ServerConnPool — 기점 서버 연결 풀링 지정

이 지시문으로 프록시가 기점 서버에 전송되는 연결을 함께 풀할 수 있습니다. 이 지시문을 on으로 설정하면, 성능이 향상되며 지속적인 연결을 허용하는 기점 서버를 보다 잘 활용할 수 있습니다. 또한 ServerConnTimeout 지시문을 통해 사용하지 않은 연결을 유지하는 기간을 지정할 수 있습니다.

주: 이 지시문은 제어된 환경에서 최상으로 사용 가능합니다. 정방향 프록시나 기점 서버가 HTTP/1.1을 준수하지 않는 경우에 성능이 저하될 수 있습니다.

형식

ServerConnPool {on | off}

기본값

ServerConnPool off

ServerConnTimeout — 최대 비활성 기간 지정

이 지시문을 사용하여, 연결이 취소되기 전에 네트워크 활동없이 허용된 제한 시간을 지정하십시오. ServerConnPool 지시문이 on으로 설정된 경우에만 이 지시문을 사용하십시오.

형식

ServerConnTimeout *time-spec*

예제

ServerConnTimeout 30 seconds

기본값

ServerConnTimeout 10 seconds

ServerInit — 서버 초기설정 단계 사용자 정의

이 지시문을 사용하여 초기설정 루틴 중에 서버가 호출하는 사용자 정의된 응용프로그램 기능을 지정합니다. 이 코드는 클라이언트 요청이 읽히기 전이나 서버가 재시작할 때마다 실행됩니다.

PreExit 또는 Service 단계에서 GoServe 모듈을 사용하고 있으면, 여기서 gosclone 모듈을 호출해야 합니다.

형식

ServerInit */path/file:function_name* [*initialization_string*]

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

initialization_string

선택적, 응용프로그램 기능으로 전달되는 텍스트 문자열을 지정합니다.

예제

```
ServerInit    /ics/api/bin/icsext05.so:svr_init
```

기본값

없음

ServerRoot — 서버 프로그램이 설치될 디렉토리 지정

이 지시문을 사용하여 서버 프로그램이 설치될 디렉토리(서버의 현재 작업 디렉토리)를 지정합니다. 로그 지시문은 상대 경로 이름이 사용될 때, 현재 작업 디렉토리를 기본 루트로 사용합니다.

Windows에서 이 디렉토리는 설치 중에 식별됩니다.

형식

ServerRoot *directory_path*

기본값

- **Linux 및 UNIX 시스템:** ServerRoot /opt/ibm/edge/cp/server_root/
- **Windows 시스템:** C:\Program Files\IBM\edge\cp\bin\

주: 기본값을 변경할 수 있지만, 서버에서 요청을 처리하는 방법에는 영향을 미치지 않습니다.

주: PASS 및 EXEC 규칙은 이 디렉토리와 무관합니다.

ServerTerm — 서버 종료 단계 사용자 정의

이 지시문을 사용하여 서버 종료 단계 동안 서버가 호출하는 사용자 정의된 응용프로그램 기능을 지정합니다. 이 코드는 규칙적인 시스템 종료 발생 시 및 서버가 재시작할 때마다 실행합니다. 이 코드로 PreExit 응용프로그램 기능이 할당된 자원을 릴리스할 수 있습니다.

형식

ServerTerm */path/file:function_name*

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

예제

ServerTerm /ics/api/bin/icsext05.so:shut_down

기본값

없음

Service — 서비스 단계 사용자 정의

이 지시문을 사용하여 서비스 단계 동안 서버가 호출하는 사용자 정의된 응용프로그램 기능을 지정합니다. 이 코드는 클라이언트 요청을 제공합니다. 예를 들어, 파일을 전송하거나 CGI 프로그램을 실행합니다.

이 지시문에 대한 기본값은 없습니다. 요청이 서비스 규칙과 일치하지만(Service 지시문에 지정된 응용프로그램 기능이 실행됨), 기능이 HTTP_NOACTION을 리턴하면 서버에서 오류를 생성하고 요청은 실패합니다.

형식

Service *request_template/path/file:function_name*

[*server_IP_address* | *host_name*]

request_template

응용프로그램 기능이 호출되는지 여부를 추가적으로 판별하는 요청에 대한 템플릿을 지정합니다. 스펙은 프로토콜, 도메인, 호스트를 포함할 수 있고, 앞에 슬래시(/)가 붙을 수 있으며, 별표(*)를 와일드 카드로 사용할 수 있습니다. 예를 들어, /front_page.html, http://www.ics.raleigh.ibm.com, /pub*, /* 및 *는 모두 유효합니다.

/path/file

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능 이름을 지정합니다.

[*Server_IP_address* | *host_name*]

여러 IP 주소나 가상 호스트를 사용하고 있을 경우, 응용프로그램 기능을 고유한 IP 주소나 호스트로 들어오는 요청에 대해서만 호출할지 여부를 판별합니다.

서버 IP 주소로 와일드 카드 문자를 지정할 수 없습니다.

예제

```
Service /index.html /ics/api/bin/icsext05.so:serve_req
Service /cgi-bin/hexcalc* /ics/api/calculator:HEXcalc*
```

주: *query_string*을 비롯한 전체 경로 변환을 하려면, 두 번째 예에 표시된 대로 *request_template* 및 *path/file:function_name*에 모두 별표(*)가 있어야 합니다.

기본값

없음

SignificantURLTerminator — URL 요청의 종료 코드 지정

이 지시문을 사용하여 URL 요청의 종료 코드를 지정하십시오. 요청에서 종료 코드를 사용하면 Caching Proxy가 요청을 처리하고 결과가 이미 캐시되었는지 여부를 평가할 때 종료 코드 앞의 문자만 평가합니다. 종료 코드가 하나 이상 정의되면, Caching Proxy는 수신 URL을 ibmproxy.conf 파일에 정의된 순서대로 종료 코드와 비교합니다.

형식

SignificantURLTerminator *terminating_string*

예제

SignificantURLTerminator &.

이 예제에서 다음 두 개의 요청이 동일하게 처리됩니다.

```
http://www.exampleURL.com/tx.asp?id=0200&.x=004;y=001
http://www.exampleURL.com/tx.asp?id=0200&.x=127;y=034
```

기본값

없음

SMTPServer(Windows 전용) — sendmail 루틴에 대한 SMTP 서버 설정

이 지시문을 사용하여 Windows용 Caching Proxy 내에서 내부 sendmail 루틴이 사용하는 SMTP 서버를 설정하십시오. 이 루틴에 대하여 다음과 같은 두 가지 지시문을 또한 설정해야 합니다. 314 페이지의 『WebMasterEMail — 서버 선택 보고서를 수신

할 전자 우편 주소 설정』 및 315 페이지의 『WebMasterSocksServer(Windows 전용) — sendmail 루틴에 대한 socks 서버 설정』.

형식

SMTPServer *IP address or hostname of SMTP server*

예제

SMTPServer mybox.com

기본값

없음

SNMP — SNMP 지원 사용 가능 및 사용 불가능

이 지시문을 사용하여 SNMP 지원을 사용 가능하게 하거나 사용 불가능하게 합니다.

형식

SNMP {on | off}

기본값

SNMP off

SNMPCommunity — SNMP에 대한 보안 암호 제공

이 지시문을 사용하여, 웹 서버 DPI(분산 프로토콜 인터페이스) 서브에이전트와 SNMP 에이전트 사이에 암호를 정의합니다. SNMP 공동체 이름은 서버의 지정 공동체에 대해 SNMP가 모니터링하는 성능 변수를 표시할 수 있는 권한을 사용자에게 부여합니다. 시스템 관리자는 암호가 입력될 때 서버에서 표시될 수 있는 변수를 정의합니다. SNMP 공동체 이름을 변경하면, /etc/snmpd.conf 파일에 지정된 공동체 이름도 변경해야 합니다.

형식

SNMPCommunity *name*

기본값

SNMPCommunity public

SSLCaching — 보안 요청에 대한 캐시 사용 가능

이 지시문을 사용하여 역방향 프록시가 사용될 때 보안 요청의 콘텐츠를 캐시합니다. 이 지시문으로 클라이언트 연결 및 백엔드 콘텐츠 서버 연결 등 프록시 서버에 연결된 모든 연결사항에 대한 캐시를 구성합니다.

주: SSL 지시문은 SUSE Linux에서 지원되지 않습니다.

형식

SSLCaching {on | off}

기본값

SSLCaching off

SSLCertificate — 인증서에 대한 키 레이블 지정

이 지시문을 사용하여, Caching Proxy가 고유의 SSL 인증을 제공하는 여러 도메인의 단일 역방향 프록시로 작동할 때 클라이언트에 전송할 인증을 판별할 수 있도록 하는 키 레이블을 지정하고, 프록시 서버에 클라이언트측 PKI 인증을 검색하거나 검색하지 않도록 지시하십시오.

SSLCertificate 지시문을 사용하여, Caching Proxy는 인증을 발행하는 인증 기관(CA) 또는 스스로 지정하는 인증을 구별할 수 있습니다. 그러나 CA 발행 인증 (ClientAuthRequired 옵션)을 승인하여 이 지시문을 사용하는 경우, 유효하지 않은 사용자가 프록시 서버로 액세스할 수 있도록 허용할 수 있습니다. SSLCertificate 지시문의 ClientAuthRequired 옵션을 사용하는 경우, 논리식 옵션을 사용하여 SSL 채널을 액세스할 수 있는 사용자를 판별할 수 있습니다.

추가 논리식이 SSLCertificate 지시문에 추가된 경우, Caching Proxy는 클라이언트 인증서에서 값을 추출하고 논리식에서 값을 계산합니다. 표현식이 클라이언트 인증서의 값을 만족하는 경우, Caching Proxy는 클라이언트에 SSL 연결을 허용합니다. 그렇지 않은 경우, 연결이 종료되고 닫힙니다.

형식

SSLCertificate *serverIP/hostname CertificateLabel*
[NoClientAuth | ClientAuthRequired *logic-expression*]

serverIP/hostname

IP 주소(예: 204.146.167.72)를 지정하거나, SSL 요청이 전달될 서버에 호스트 이름(예: hostA.raleigh.ibm.com)을 지정할 수 있습니다.

CertificateLabel

클라이언트 인증이 필요한 경우, 지정된 IP 주소나 호스트 이름에 전달된 SSL 요청에 사용할 인증 이름.

[NoClientAuth | ClientAuthRequired *logic-expression*]

프록시 서버에 클라이언트측 PKI 인증을 검색하거나 검색하지 않도록 한 지침.

ClientAuthRequired 옵션과 함께 사용되는 경우에만 논리식 옵션이 유효합니다. 추가 논리식이 SSLCertificate 지시문에 추가된 경우, Caching Proxy는 클라이언트 인증서에서 값을 추출하고 논리식에서 값을 계산합니다. 표현식이 클라이언트 인증서의 값을 만족하는 경우, Caching Proxy는 클라이언트에 SSL 연결을 허용합니다. 그렇지 않은 경우, 연결이 종료되고 닫힙니다.

- 논리식의 속성 이름은 IST, ICN, IOU, IC, IL, IO, IE, ST, CN, OU, C, L, O, E입니다.
 - 속성 이름은 클라이언트 인증서에서 다음 필드로 맵핑됩니다.
 IssuerStateOrProvince (IST) IssuerCommonName (ICN) IssuerOrgUnit (IOU) IssuerCountry (IC) IssuerLocality (IL) IssuerOrg (IO) IssuerEmail (IE) StateOrProvince (ST) CommonName (CN) OrgUnit (OU) Country (C) Locality (L) Org (O) Email (E).
- 속성 이름에 대한 값은 따옴표로 구분되어야 합니다.
- 유효한 논리 연산자는 다음과 같습니다. && (AND), || (OR), ! (NOT), = (EQUAL).

예제

```
SSLCertificate www.abc.com ABCCert
SSLCertificate 204.146.167.72 intABCCert
SSLCertificate www.xyz.com XYZCert ClientAuthRequired
SSLCertificate www.xyz.com XYZCert ClientAuthRequired
CN="valid.user.common.name.pattern" && (L="accepted.location.pattern" ||
C!="not.valid.country.pattern")
```

기본값

없음

SSLCryptoCard — 설치된 암호 카드 지정

이는 역방향 프록시 구성에만 적용합니다.

이 지시문을 사용하여 암호 카드 설치를 프록시 서버에 알리고 카드를 지정합니다.

AIX에서 IBM 4960 PCI 암호화 액셀러레이터 카드를 지원하려면, 271 페이지의 『PKCS11DefaultCert, PKCS11DriverPath, PKCS11TokenPassword — IBM 4960 PCI 암호화 액셀러레이터 카드(AIX 전용) 지원』을 참조하십시오.

형식

```
SSLCryptoCard {rainbowcs | nciphernfast} {on | off}
```

예제

```
SSLCryptoCard rainbowcs on
```

기본값

없음

SSLEnable — 보안 요청에 대한 포트 443의 인식 지정

이 지시문을 사용하여 Caching Proxy가 보안 요청에 대해 포트 443을 인식하도록 지정합니다.

주: SSL 지시문은 SUSE Linux에서 지원되지 않습니다.

형식

SSLEnable {on | off}

기본값

SSLEnable off

SSLForwardPort — HTTP SSL 업그레이드에 대해 주소 지정할 포트 지정

이 지시문을 사용하여 Caching Proxy가 SSL를 구현하여 HTTPS 요청으로 업그레이드하는 HTTP 요청 주소에 대한 포트를 지정합니다. 기본 HTTP 포트 80 또는 기본 SSL 포트 443 이외의 포트를 지정합니다.

형식

SSLForwardPort *port number*

예제

SSLForwardPort 8888

기본값

없음

SSLOnly — HTTP 요청의 리스너 스레드 사용 불가능

이 지시문을 사용하여 SSL(일반적으로 포트 443)이 사용 가능할 때 표준 HTTP 요청(일반적으로 포트 80 및 8080)에 대한 리스너 스레드를 사용 불가능하게 하십시오.

형식

SSLOnly {on | off}

기본값

SSLOnly off

SSLPort — 기본값 외 HTTPS 리스닝 포트 지정

이 지시문을 사용하여 ibmproxy의 기본 HTTPS 포트 443 외 HTTPS 리스닝 포트를 지정하십시오.

주: ibmproxy는 각 인스턴스에 대해 하나의 HTTPS 포트를 지원하므로, 지시문은 여러 HTTPS 포트를 지정하도록 사용할 수 없습니다. 여러 HTTPS 포트를 지원하려면, 여러 ibmproxy 인스턴스를 다른 ibmproxy.conf 파일로 시작해야 합니다.

형식

SSLPort *port value*

여기에서 *port value*는 0보다 큰 정수값입니다. 또한 *port value*는 운영 체제에서 허용되어야 하고, 다른 응용프로그램에서 사용할 수 없습니다.

예제

SSLPort 8443

기본값

443

SSLTunneling — SSL 터널링 사용 가능

이는 정방향 프록시 구성에만 적용합니다.

이 지시문을 on으로 설정하면, 대상 서버의 모든 포트에 대한 SSL 터널링을 할 수 있습니다. 이 지시문을 off로 설정하면, 프록시 규칙에서 제공된 포트에 대해서만 SSL 터널링을 할 수 있습니다. SSL 터널링에 대한 프록시 규칙이 없고 SSLTunneling 지시문을 off로 설정하면, SSL 터널링이 허용되지 않습니다. SSLTunneling 지시문을 on으로 설정하면, Enable 지시문을 사용하여 "CONNECT" 메소드도 사용 가능하게 해야 합니다.

Caching Proxy를 정방향 프록시로 사용하는 경우, 이 지시문을 사용해야 합니다. 그러나 Caching Proxy를 역방향 프록시로 사용하는 경우, 이 지시문(기본)을 사용 불가능하게 하면, SSL 터널링 취약성 공격에 대해 보호합니다.

자세한 내용은 127 페이지의 『SSL 터널링』을 참조하십시오.

주: Proxy 지시문을 사용하여 대상 호스트의 고유한 포트에 SSL 터널링을 사용 가능하게 하십시오.

형식

SSLTunneling {on | off}

기본값

SSLTunneling off

SSLVersion — SSL 버전 지정

이 지시문을 사용하여 사용할 SSL 버전(V2, V3 또는 모든 버전)을 지정합니다. SSL 버전 3을 지원할 수 없는 서버를 사용하고 있으면, 이 지시문을 V2로 설정합니다.

주: SSL 지시문은 SUSE Linux에서 지원되지 않습니다.

형식

SSLVersion {SSLV2 | SSLV3 | all}

기본값

SSLVersion SSLV3

SSLV2Timeout — SSLV2 세션이 만기되기 전에 대기할 시간 지정

이 지시문을 사용하여 SSL 버전 2가 세션이 만기되기 전에 활동하지 않고 대기할 기간을 초 단위로 지정합니다.

주: SSL 지시문은 SUSE Linux에서 지원되지 않습니다.

형식

SSLV2Timeout *seconds*

여기서 *seconds*는 0 - 100 사이의 값을 표시합니다.

기본값

SSLV2Timeout 100

SSLV3Timeout — SSLV3 세션이 만기되기 전에 대기할 시간 지정

이 지시문을 사용하여 SSL 버전 3이 만기되기 전에 활동하지 않고 대기할 기간을 초 단위로 지정합니다.

주: SSL 지시문은 SUSE Linux에서 지원되지 않습니다.

형식

SSLV3Timeout *seconds*

여기서 *seconds*는 1 - 86400초(하루를 초로 환산) 사이의 값을 표시합니다.

기본값

SSLV3Timeout 100

SuffixCaseSense — 접미부 정의가 대소문자 구분되는지 여부 지정

이 지시문을 사용하여 파일 접미부를 AddClient, AddCharSet, AddType, AddEncoding, AddLanguage 지시문의 접미부 패턴과 비교할 때, 서버에서 대문자와 소문자를 구분하는지 여부를 지정합니다. 기본적으로 서버에서 대소문자를 구분하지 않습니다.

형식

SuffixCaseSense {on | Off}

기본값

SuffixCaseSense Off

SupportVaryHeader — HTTP Vary 헤더를 기반으로 하는 자원에 하나 이상의 변형을 캐시

이 지시문을 사용하여 Caching Proxy가 HTTP Vary 헤더를 기반으로 하는 자원에 하나 이상의 변형을 캐시합니다.

SupportVaryHeader 지시문이 사용 가능한 경우, 프록시는 URI 기반의 캐시 ID 및 클라이언트 요청으로 선택된 헤더 값을 형성합니다.

선택된 헤더의 이름은 서버의 이전 응답에서 전송된 Vary 헤더에서 지정됩니다. 서버가 자원에 대해 선택한 헤더 이름 세트를 변경한 경우, 자원에 대해 이전에 캐시된 모든 오브젝트는 프록시 캐시에서 제거됩니다.

이 지시문을 RegisterCacheIdTransformer 지시문과 함께 사용할 수 있습니다(292 페이지의 『RegisterCacheIdTransformer — 쿠키 헤더를 기반으로 하는 자원에 하나 이상 변형을 캐시』).

이 지시문을 모두 사용하는 경우, 프록시는 서버 및 클라이언트의 요청 헤더에서 Vary 헤더에서 기반한 내부 캐시 ID 변환기를 작성합니다. 이러한 방법으로, 프록시는 요청된 URI가 동일하더라도 다른 요청에 대해 고유한 캐시 ID, 응답 쌍을 생성합니다.

동일한 URI의 캐시 오브젝트는 요청/응답 또는 기타 구성 설정에서의 만기 및 캐시 제어 헤더에 따라 캐시에서 기본 수명이 있습니다. Dynacache 플러그인이 사용되는 경우, 동일한 URI에 연관된 모든 여러 프리젠테이션은 프록시 캐시에서 함께 유효하지 않습니다.

형식

SupportVaryHeader {on | off}

예제

예를 들어, 다음 지시문은 다음과 같이 ibmproxy.conf에서 사용 가능하게 되고 구성됩니다.

```
SupportVaryHeader on
RegisterCacheIdTransformer Cookie UserGroup
```

클라이언트 게스트는 프록시 서버에 다음으로 액세스합니다.

URI [`<code>`] `http://www.dot.com/group.jpg` [`</code>`]

그리고 응답/요청은 다음과 같습니다.

```
GET /group.jpg HTTP/1.1
Host: www.dot.com
Cookie: UserGroup=Guest
Accept-Language: en_US
```

```
HTTP/1.1 200
Server: my-server
Vary: Accept-Language
.....
```

다음으로, 클라이언트 Admin은 프록시 서버에 동일한 URI로 액세스합니다.

```
http://www.dot.com/group.jpg
```

그리고 응답/요청은 다음과 같습니다.

```
GET /group.jpg HTTP/1.1
Host: www.dot.com
Cookie: UserGroup=Admin
Accept-Language: fr_FR
```

```
HTTP/1.1 200
Server: my-server
Vary: Accept-Language
.....
```

그 결과로, 응답이 캐시 가능한 경우, 프록시 서버는 2개의 다른 캐시 ID를 생성합니다.

1. CacheID(URI, "Guest", "en_US")
2. CacheID(URI, "Admin", "fr_FR")

프록시 서버는 캐시의 서버에서 2개의 다른 변형된 응답을 저장합니다. 그 다음으로, 클라이언트는 언어 환경 설정과 사용자 그룹 값을 결합하여 자원(.../group.jpg)을 요청하는 경우, 프록시 서버는 캐시에서 적절히 변형한 자원을 검색하여 제공합니다.

기본값

```
SupportVaryHeader off
```

TLSV1Enable — 전송 계층 보안 프로토콜 사용 가능

이 지시문을 사용하여 SSL 연결에서 TLS 버전 1 프로토콜을 사용할 수 있도록 하십시오. 이 지시문이 켜진 후에 SSL 연결은 맨 처음으로 TLS 프로토콜, SSLv3 프로토콜, SSLv2 프로토콜 순으로 확인합니다.

주: 이 지시문은 Internet Explorer 및 기타 브라우저에서 작동하며, Netscape에서는 작동하지 않습니다. (Netscape는 Caching Proxy에 사용하는 권장 브라우저가 아닙니다.)

형식

```
TLSV1Enable {on | off}
```

예제

```
TLSV1Enable on
```

초기 구성 파일 설정

없음

Transmogrifier — 데이터 조작 단계 사용자 정의

이 지시문을 사용하여 데이터 조작 단계 동안 서버가 호출하는 사용자 정의된 응용프로그램 기능을 지정합니다. 이 코드는 세 개의 응용프로그램 기능을 다음과 같이 제공합니다.

- 데이터를 처리하기 전에 초기설정을 수행하기 위한 *open* 기능
- 데이터를 처리하기 위한 *write* 기능
- 제거 활동을 수행하기 위한 *close* 기능
- 발생한 문제점의 통지를 제공하기 위한 *error* 기능

각 인스턴스는 여러 변형자를 가질 수 있습니다.

형식

```
Transmogrifier /path/file:function_name:function_name:function_name  
/path/file
```

확장자를 포함하는 사용자의 컴파일된 프로그램의 완전한 파일 이름을 지정합니다.

function_name

사용자 프로그램 내의 응용프로그램 기능에 부여한 이름을 지정합니다. 열기, 쓰기, 닫기 기능의 이름을 부여해야 합니다.

예제

```
Transmogrifier /ics/bin/icsext05.so:open_data:write_data:close_data
```

기본값

없음

TransmogrifiedWarning — 클라이언트에게 경고 메시지 전송

이 지시문을 사용하여 클라이언트에게 데이터를 알려주는 메시지를 전송하십시오.

형식

```
transmogrifiedwaning {yes|no}
```

기본값

Yes

TransparentProxy — Linux에서 투명 프록시를 사용 가능하게 함

이는 정방향 프록시 구성에만 적용합니다.

Linux 시스템 전용의 경우, 이 지시문을 사용하여 서버가 투명 프록시 서버로 실행할 수 있는지 여부를 지정합니다.

TransparentProxy 지시문을 on으로 설정하면, BindSpecific 지시문이 무시되면 기본값은 off입니다. 대부분의 HTTP 통신은 포트 80에서 흐르기 때문에, 구성된 포트 중 하나로 할 것을 권장합니다.

형식

```
TransparentProxy {on | off}  
Port 80
```

기본값

```
TransparentProxy off
```

IPCHAIN 방화벽을 사용하는 경우, 지시문을 사용 가능하게 하면, 투명 프록시를 구성할 수 있습니다. IPTABLES 방화벽을 사용하는 경우, IPTABLES 방화벽 규칙을 수동으로 추가해야 합니다.

IPTABLES 방화벽을 사용하는 경우, TransparentProxy 지시문을 사용 가능하게 한 다음 프록시 서버를 시작하기 전에, 다음 명령을 실행하여 IPTABLES에 방화벽 규칙을 추가하십시오.

```
iptables -t nat -A PREROUTING -i your-network-interface -p tcp --dport 80 -j  
REDIRECT --to-port ibmproxy-listening-port
```

방화벽 및 프록시 서버가 동일한 상자에 있다고 가정하면, 이 규칙은 IPTABLES 방화벽이 포트 80으로 지정된 모든 TCP 통신량을 로컬 프록시 리스닝 포트에 경로를 재지정하도록 지시합니다. 이 규칙을 IPTABLES 구성에서 추가할 수 있습니다. 이는 시스템을 다시 시작하는 경우, 규칙을 자동으로 로드하도록 허용합니다.

투명 프록시를 시작한 후 Caching Proxy 서버를 정지시키려면, 루트로 다음 명령을 실행해야 합니다.

```
ibmproxy -unload
```

Linux 시스템에서 이 명령은 재지정 방화벽 규칙을 제거합니다. 서버를 정지시킨 후 이 명령을 발행하지 않으면, 시스템은 대상이 아닌 요청을 승인합니다.

UpdateProxy — 캐시 대상 지정

이 지시문을 사용하여 캐시 에이전트가 갱신할 프록시 서버를 지정합니다. 이러한 식별은 캐시 에이전트가 실행 중인 로컬 프록시 서버 이외의 프록시 서버를 갱신해야 할 때 필요합니다. 선택적으로, 포트를 지정할 수 있습니다.

주: Linux 및 UNIX 플랫폼에서 이 지시문은 캐시 에이전트를 사용하는 데 필요합니다. 프록시에 대해 하나의 시스템만을 사용 중인 경우에는 호스트 이름을 지정하십시오.

캐시 에이전트가 다른 서버의 캐시를 갱신할 수 있더라도, 해당 시스템에서 캐시 액세스 로그를 검색할 수 없습니다. 따라서, UpdateProxy 지시문이 로컬 호스트 이외의 호스트를 지정하면, LoadTopCached 지시문이 무시됩니다.

형식

UpdateProxy *fully_qualified_host_name_of_proxy_server*

예제

UpdateProxy proxy15.ibm.com:1080

기본값

없음

UserId — 기본 사용자 ID 지정

이 지시문을 사용하여 파일에 액세스하기 전에 서버가 변경할 대상 사용자의 이름이나 번호를 지정합니다.

이 지시문을 변경하면, 서버를 직접 정지시킨 다음 재시작해야 변경사항이 적용됩니다. 서버를 재시작하기만 할 경우에는 서버가 변경사항을 인식하지 않습니다(15 페이지의 제 5 장 『Caching Proxy 시작 및 정지』 참조).

주: 사용자 ID, 그룹 ID, 또는 로그 디렉토리 경로에 대한 서버 기본값을 변경한 경우, 새 디렉토리를 작성하고 디렉토리의 권한 및 소유권을 갱신하십시오. 서버에서 사용자 정의된 로그 디렉토리에 정보를 기록하도록 하려면, 해당 디렉토리에 대한 권한을 755로 설정하고 사용자 정의된 서버 사용자 ID를 소유자로 설정하십시오. 예를 들어, 서버의 사용자 ID를 기본값에서 jdoe로 변경하고, 기본 로그 디렉토리를 server_root/account로 변경한 경우, server_root/account 디렉토리는 755의 권한을 가져야 하며 jdoe가 소유해야 합니다.

형식

UserId {*ID_name* | *number*}

기본값

AIX, Linux, Solaris: UserId nobody

HP-UX: UserId www

V2CipherSpecs — SSL 버전 2에 지원되는 암호 스펙 나열

이 지시문은 SSL 버전 2에서 사용할 수 있는 암호 스펙을 나열합니다.

주: SSL 지시문은 SUSE Linux에서 지원되지 않습니다.

형식

V2CipherSpecs *specification*

승인 가능한 값은 다음 사항의 결합입니다. 두 번 사용될 수 없습니다.

- 1 — RC4 US
- 2 — RC4 내보내기
- 3 — RC2 US
- 4 — RC2 내보내기
- 6 — DES 56비트
- 7 — 3중 DES US
- NULL — 기본 암호 스펙을 사용함

예제

- 미국: V2CipherSpecs '137624'
- 내보내기: V2 Cipherspecs '246'

기본값

없음(SSL은 기본적으로 사용 불가능함)

V3CipherSpecs — SSL 버전 3에 지원되는 암호 스펙 나열

이 지시문은 SSL 버전 3에 대하여 사용 가능한 암호 스펙을 나열합니다.

주: SSL 지시문은 SUSE Linux에서 지원되지 않습니다.

FIPSEnable 지시문이 "on"으로 설정된 경우, V3CipherSpecs 지시문은 무시됩니다. 자세한 내용은 234 페이지의 『FIPSEnable — SSLV3 및 TLS에 대해 FIPS(Federal Information Processing Standard) 승인 암호를 사용 가능하게 함』을 참조하십시오.

형식

V3CipherSpecs *specification*

승인 가능한 값은 다음과 같습니다.

- 00 — NULL NULL
- 01 — NULL MD5
- 02 — NULL SHA

- 03 — RC4 MD5 내보내기
- 04 — RC4 MD5 US
- 05 — RC4 SHA US
- 06 — RC2 MD5 내보내기
- 09 — DES SHA 내보내기
- 0A — 3중 DS SHA US
- 62 — 56비트 DES CBC SHA
- 64 — 56비트 RC4 SHA
- NULL — 기본 암호 스펙을 사용함

예제

- 미국: V3CipherSpecs '0A09060564620403020100'
- 내보내기: V3Cipherspecs '0906646203020100'

기본값

없음(SSL은 기본적으로 사용 불가능함)

WebMasterEMail — 서버 선택 보고서를 수신할 전자 우편 주소 설정

이 지시문을 사용하여 SSL 인증서 만기 30일 전 통지와 같은, Caching Proxy 선택 보고서를 수신할 전자 우편 주소를 설정하십시오. Linux 및 UNIX 시스템에서는 sendmail 프로세스를 실행 중이어야 합니다. Windows 시스템의 경우 sendmail 프로세스가 Caching Proxy에 빌드되므로, 외부 메일 서버가 필요하지 않습니다. 그러나 다음과 같은 두 개의 추가 지시문을 설정해야 합니다. 315 페이지의 『WebMasterSocksServer(Windows 전용) — sendmail 루틴에 대한 socks 서버 설정』 및 301 페이지의 『SMTPServer(Windows 전용) — sendmail 루틴에 대한 SMTP 서버 설정』.

주: 이 전자 우편 주소는 anonymous FTP 암호로도 사용됩니다.

형식

WebMasterEMail *webmastermailaddress*

예제

WebMasterEMail webmaster@computer.com

기본값

WebMasterEMail webmaster

WebMasterSocksServer(Windows 전용) — sendmail 루틴에 대한 socks 서버 설정

이 지시문을 사용하여 Windows용 Caching Proxy 내에서 내부 sendmail 루틴이 사용하는 socks 서버를 설정하십시오. 이 루틴에 대하여 다음과 같은 두 가지 지시문을 또한 설정해야 합니다. 314 페이지의 『WebMasterEMail — 서버 선택 보고서를 수신할 전자 우편 주소 설정』 및 301 페이지의 『SMTPServer(Windows 전용) — sendmail 루틴에 대한 SMTP 서버 설정』.

형식

WebMasterSocksServer *IP address or hostname of socks server*

예제

WebMasterSocksServer socks.mybox.com

기본값

없음

Welcome — 환영 파일의 이름 지정

이 지시문을 사용하여 고유한 파일 이름이 없는 요청에 응답하기 위해 서버가 찾는 파일의 이름을 지정합니다. 구성 파일에 이 지시문에 대한 여러 개의 어커런스를 넣음으로써, 환영 파일 목록을 작성할 수 있습니다.

파일 이름이나 디렉토리 이름이 없는 요청의 경우, 서버에서는 Welcome 지시문에 지정된 이름과 일치하는 파일은 항상 파일 루트 디렉토리에서 찾습니다. 일치하는 파일이 있으면, 요청자에게 되돌아 갑니다.

디렉토리 이름은 있지만 파일 이름은 없는 요청에 대해서, AlwaysWelcome 지시문은 서버가 리턴할 환영 파일을 디렉토리에서 찾을지 여부를 제어합니다. 기본적으로 AlwaysWelcome은 On 값으로 설정됩니다. 이렇게 하면, 서버에서는 Welcome 지시문에서 지정된 이름과 일치하는 파일을 요청된 디렉토리에서 찾습니다. 일치하는 파일이 있으면, 요청자에게 되돌아 갑니다.

서버가 디렉토리의 파일과 Welcome 지시문의 파일 이름간에 일치하는 파일을 하나 이상 발견하면, Welcome 지시문의 순서가 돌려보낼 파일을 판별합니다. 서버에서는 구성 파일의 처음과 가장 가까운 Welcome 지시문을 사용합니다.

형식

Welcome *file_name* [*server_IP_address* | *host_name*]

file_name

환영 파일로 정의할 파일의 이름을 지정합니다.

[server_IP_address | host_name]

여러 IP 주소나 가상 호스트를 사용하고 있다면, 이 매개변수를 사용하여 IP 주소나 호스트 이름을 지정하십시오. 서버는 이 IP 주소나 호스트의 서버에 도달하는 요청이나 이 호스트에 대해서만 지시문을 사용합니다. IP 주소는 요청하는 클라이언트의 주소가 아니라 서버 네트워크 연결의 주소입니다.

IP 주소(예:240.146.167.72) 또는 호스트 이름(예: hostA.bcd.com)을 지정할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수가 없으면, 서버는 요청이 들어오는 IP 주소나 URL의 호스트 이름과 상관없이 모든 요청에 대해 지시문을 사용합니다.

서버 IP 주소로 와일드 카드 문자를 지정할 수 없습니다.

예제

- 다음 예제에서는 두 개의 환영 페이지를 정의하고, AlwaysWelcome 지시문이 기본값 On으로 설정되었다고 가정합니다. 파일 이름이 없는 요청의 경우, 서버에서 요청(또는 요청이 파일 이름이나 디렉토리를 지정하지 않는 경우에는 파일 루트 디렉토리)에 지정된 디렉토리의 환영 파일을 리턴하려고 합니다. 서버는 letsgo.html이라는 이름의 파일을 먼저 찾습니다. 디렉토리에 해당 이름의 파일이 없을 경우, 서버는 Welcome.html이라는 이름의 파일을 찾습니다.

```
Welcome letsgo.html
Welcome Welcome.html
```

- 다음 예제에서 서버는 요청이 들어오는 네트워크 연결의 IP 주소에 기초한 다른 환영 파일을 찾습니다. 0.67.106.79로 들어오는 요청의 경우, 서버는 CustomerA.html이라는 이름의 환영 파일을 찾습니다. 0.83.100.45로 들어오는 요청에 대해 CustomerB.html이라는 이름의 환영 파일을 찾습니다. 요청이 다른 IP 주소로 들어오면, 기본 주소를 찾습니다.

```
Welcome CustomerA.html 0.67.106.79
Welcome CustomerB.html 0.83.100.45
```

- 다음 예제에서 서버는 URL의 호스트 이름에 기초한 다른 환영 파일을 찾습니다. hostA로 들어오는 요청에 대해 CustomerA.html이라는 이름의 환영 파일을 찾습니다. hostB로 들어오는 요청에 대해 CustomerB.html이라는 이름의 환영 파일을 찾습니다. 요청이 다른 호스트로 들어오면, 기본 호스트 이름을 찾습니다.

```
Welcome CustomerA.html hostA.bcd.com
Welcome CustomerB.html hostB.bcd.com
```

기본값

이들 기본값은 기본 구성에서 순서대로 사용됩니다.

```
Welcome Welcome.html
Welcome welcome.html
Welcome index.html
Welcome Frntpage.html
```

주의사항

초판(2006년 5월)

이 정보는 미국내에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서는 이 자료에 기술된 제품 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급하는 것이 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 다른 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가와 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 사용권을 부여하는 것은 아닙니다. 사용권 조치는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 사용권 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다.

IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 “현상태대로” 제공합니다. 일부 국가에서는 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용 또는 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 이 변경사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 비 IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램 및 기타 다른 프로그램(이 프로그램 포함)간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 정보를 원하는 프로그램 사용권자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조항 및 조건(예를 들어, 사용권 지불 등)에 따라 사용할 수 있습니다.

이 문서에서 언급된 사용 허가 프로그램 및 이에 대해 사용 가능한 모든 사용권 자료는 IBM이 IBM 국제 프로그램 사용권 협약 또는 이와 동등한 계약에 의해 제공됩니다.

여기에 나와 있는 모든 성능 데이터는 조정된 환경에서 측정되었습니다. 따라서 다른 운영 환경에서 얻어진 결과들과는 차이가 있을 수 있습니다. 몇몇 결과는 개발 단계의 시스템상에서 측정되었으며, 일반적으로 통용되는 시스템에서 똑같은 평가 결과를 보증할 수는 없습니다. 또한 몇몇 측정 결과는 추정치에 의거하여 도출되었습니다. 실제 결과와는 다를 수 있습니다. 이 문서의 사용자들은 이 데이터를 사용자 고유의 환경에 적용할 수 있는지 확인해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 테스트하지 않았으므로, 이들 제품과 관련된 성능의 정확성, 호환성 또는 기타 주장에 대해서는 확신할 수 없습니다. 비IBM 제품 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM의 향후 지침이나 의도에 관한 모든 문구는 목적과 목표일 뿐 공고없이 수정되거나 철회될 수 있습니다.

이 정보에는 일상적인 업무 운영시 사용되는 데이터 및 보고서에 대한 예제가 있습니다. 가능한 잘 설명하기 위해, 예제에는 개인, 회사, 상표, 제품 이름이 포함될 수 있습니다. 여기 언급된 모든 이름은 가명이며 실제 비슷한 기업 이름이나 주소가 있다면 이는 전적으로 우연입니다.

소프트카피로 이 정보를 본다면 사진이나 컬러 그림은 보이지 않을 수 있습니다.

상표

다음 용어는 미국 또는 기타 국가에서 사용되는 IBM Corporation의 상표입니다.

- AIX[®]
- IBM
- Netfinity[®]
- RS/6000[®]
- SecureWay[®]
- Tivoli
- ViaVoice[®]
- WebSphere

Java[™] 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.

Microsoft[®], Windows, Windows NT[®] 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

Intel[™], Intel Inside(로고), MMX[™] 및 Pentium[®]은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.



GA30-2916-00



Spine information:



**WebSphere Application
Server**

Caching Proxy 관리 안내서

버전 6.1

GA30-2916-00