

System z의 Linux용 IBM Communications
Server



빠른 시작

버전 6.2.2

System z의 Linux용 IBM Communications
Server



빠른 시작

버전 6.2.2

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 119 페이지의 『주의사항』을 읽어보십시오.

제 3 판(2006년 7월)

이 개정판은 새 개정판 또는 기술 관련 소식지에 명시되기 전까지 Linux용 IBM Communications Server 버전 6.2.2와 모든 후속 릴리스 및 수정판에 적용됩니다.

책에 대한 주문은 한국 IBM 담당자 또는 해당 지역의 IBM 지방 사무소로 문의하십시오. 다음 주소에서는 책을 구비하고 있지 않습니다.

IBM은 고객의 의견을 환영합니다. IBM 한글 지원에 관한 설문 양식이 이 책 맨 뒤에 첨부되어 있습니다. 양식이 없으면 아래 주소로 의견을 보내주십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

전자 방법으로 의견을 보내려면 다음 방법 중 하나를 사용하십시오.

- IBM 링크: RALVM17의 CIBMORCF
- IBM 우편: IBMMAIL의 USIB2HPD
- 인터넷: USIB2HPD@vnet.ibm.com
- 팩스: 1-800-227-5088

IBM에 정보를 보내면, IBM은 귀하의 권리를 침해하지 않는 범위 내에서 IBM이 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

목차

표	vii
그림	ix
Linux용 IBM Communications Server 소개	xi
이 책의 사용 방법	xi
로드 맵	xi
활자체 규약	xii
이 책에서 사용된 약어	xiii
새로운 사항	xiv
새로운 기능	xiv
사라진 기능	xv
제 1 장 Linux용 IBM Communications Server에 관한 정보	1
Linux용 IBM Communications Server 기능 및 패키징	1
Linux용 IBM Communications Server	1
고급 네트워킹 기능	8
기능 및 이점	14
융통적인 빌딩 블록	14
클라이언트/서버 운영	14
쉬운 구성	15
관리에 대한 추가 사용자 인터페이스 선택사항	15
더 좋은 성능	16
보안 옵션	16
네트워크 관리 융통성	17
신뢰성, 사용 가능성 및 서비스 가능성	17
네트워크 통합, 성장 및 변경	18
제 2 장 System z의 Linux용 Communications Server 및 네트워크에 대한 계획 수립	19
네트워크 계획 단계	19
네트워크에 대한 기능적 요구사항 식별	19
Linux용 Communications Server 구성 방법 결정	20
설치 및 운영을 위한 자원 요구사항 식별	21
명명 규칙	24
제 3 장 Linux 서버에 Linux용 Communications Server 설치	25
Linux용 Communications Server 라이선스가 있는 프로그램의 패키지 방법	25
Linux용 Communications Server 설치 준비	26
전제조건 소프트웨어 설치	26
제품 설치 세부사항 표시	26
언어 환경 변수 변경	26
이전 레벨의 Linux용 Communications Server에서 이주	27
Linux용 Communications Server 라이선스가 있는 프로그램 설치	29

Linux용 Communications Server 설치	29
Linux용 Communications Server 온라인 문서	31
Host Access Class 라이브러리(HACL)	32
WebSphere Application Server 구성	32
WebSphere Application Server의 보안 인증서 설정	32
WebSphere Application Server 구성	32
서버 구성 파일 설치	33
설치 후 절차	34
클라이언트/서버 운영	34
설치 후 정리	34
PDF 서적 보기	34
현재 릴리스 정보 검토	35
TN 서버 또는 TN 방향 재지정자에서 사용하기 위해 SSL 구성	35
Linux용 Communications Server 구성 파일 백업	36
Linux용 Communications Server 구성 파일의 백업 사본 복원	36
구성 파일 재초기화	37
Linux용 Communications Server 설치 제거	38
제 4 장 Linux에 IBM 원격 API 클라이언트 설치	41
하드웨어 및 소프트웨어 요구사항	41
하드웨어 요구사항	41
Linux 운영 체제 버전	42
Java	42
GSKIT	42
제품 설치 세부사항 표시	42
언어 환경 변수 설정	43
Linux에 원격 API 클라이언트 설치	43
GSKIT를 사용하여 HTTPS 보안 인증서 설정	45
Linux에서 원격 API 클라이언트 설치 제거	46
제 5 장 System z용 Linux에 IBM 원격 API 클라이언트 설치	49
하드웨어 및 소프트웨어 요구사항	49
하드웨어 요구사항	49
Linux 운영 체제 버전	49
Java	49
GSKIT	50
제품 설치 세부사항 표시	50
언어 환경 변수 설정	50
System z용 Linux에 원격 API 클라이언트 설치	51
GSKIT를 사용하여 HTTPS 보안 인증서 설정	52
System z용 Linux에서 원격 API 클라이언트 설치 제거	54
제 6 장 AIX 시스템에 IBM 원격 API 클라이언트 설치	55
하드웨어 및 소프트웨어 요구사항	55
하드웨어 요구사항	55
운영 체제 버전	55
Java	55

GSKIT.	55
언어 환경 변수 변경	56
AIX에 원격 API 클라이언트 설치.	56
AIX 워크스테이션에 파일을 복사하여 원격 API 클라이언트 설치	56
CD에서 원격 API 클라이언트 설치	57
GSKIT를 사용하여 HTTPS 보안 인증서 설정.	58
AIX에서 원격 API 클라이언트 설치 제거	59
제 7 장 Windows에 대한 원격 API 클라이언트 설치 계획 세우기	61
하드웨어 및 소프트웨어 요구사항	61
설치 프로그램에 액세스	62
설치 프로그램을 사용하여 Windows에 원격 API 클라이언트 설치	63
원격 API 클라이언트 구성을 위한 고급 옵션	66
명령행에서 원격 API 클라이언트 소프트웨어 설치	67
GSKIT 소프트웨어 설치 및 보안 인증서 설정.	69
GSKIT를 사용하여 HTTPS 보안 인증서 설정.	70
설치 후 원격 API 클라이언트 소프트웨어 사용자 조정	72
원격 API 클라이언트 소프트웨어 재설치.	72
원격 API 클라이언트 소프트웨어 설치 제거	73
GSKIT 소프트웨어 설치 제거	73
도움말	74
제 8 장 Linux용 Communications Server 구성 및 사용	75
Linux용 Communications Server 구성 계획	76
계획 워크시트	77
타스크 시트	77
Motif 관리 프로그램의 사용.	77
Linux용 Communications Server 프로그램의 경로 지정	77
Linux용 Communications Server 사용 기능화	78
Motif 관리 프로그램으로 Linux용 Communications Server 관리	78
클라이언트/서버 기능 구성	83
노드 구성	84
연결성 구성	85
중속 트래픽에 대한 SDLC 링크 구성	86
중속 및 독립 트래픽을 지원하도록 이더넷 링크 구성	87
Enterprise Extender 링크 구성.	88
유형 0-3 LU 구성.	89
유형 0-3 LU 정의.	90
LU 풀 정의	90
APPC 통신 구성	92
단일 APPN 네트워크 구성	93
중속 APPC 구성	98
CPI 통신을 위한 구성.	98
LUA 구성.	99
LU 풀 정의	99
SNA 게이트웨이 구성	101
암시적 다운스트림 LU 지원	102

다운스트림 LU 정의	103
DLUR 구성	103
로컬 노드 상의 DLUR 지원 구성	105
다운스트림 노드에 대한 패스-스루 DLUR 지원 구성	106
TN 서버 구성	107
3270 LU 정의	108
LU 풀 정의	109
TN3270 서버 구성	110
TN 방향 재지정자 구성	111
TN 방향 재지정자 구성	111
Linux용 Communications Server 사용 불가능화	112
Linux용 Communications Server 자동 시작	113
Linux용 Communications Server 사용 가능화	113
SNA 노드 초기화	114
포트 및 링크 스테이션의 활성화	114
다시 시동 시 Linux용 Communications Server 시작	114
제 9 장 Linux용 Communications Server 및 SNA에 대한 정보 자원	117
SNA 라이브러리	117
네트워크에서 이용할 수 있는 정보	117
추천 문서	118
부록. 주의사항	119
상표	121
참고 서적	123
Linux용 Communications Server 버전 6.2.2 서적	123
시스템 네트워크 아키텍처(SNA) 서적	125
호스트 구성 서적	125
z/OS Communications Server 서적	125
TCP/IP 서적	126
X.25 서적	126
APPC 서적	126
프로그래밍 서적	126
기타 IBM 네트워킹 서적	126
색인	129

표

1. 시작하기 로드 맵 xii
2. 활자체 규약 xii

그림

1. System z의 Linux용 Communications Server 호스트 통신.	2
2. 여러 다운스트림 Linux 컴퓨터를 호스트 컴퓨터에 링크하는 SNA 게이트웨이.	9
3. Branch Extender	10
4. TN 서버	12
5. 노드 창.	80
6. Linux용 Communications Server 도구 모음	83
7. SNA 게이트웨이	101
8. DLUR을 제공하는 Linux용 Communications Server	105

Linux용 IBM Communications Server 소개

이 책에서는 Linux[®]를 실행 중인 컴퓨터가 시스템 네트워크 아키텍처(SNA) 네트워크에 있는 다른 노드와 정보를 교환할 수 있게 하는 IBM[®] 소프트웨어 제품인, System z의 Linux용 IBM Communications Server(Linux용 Communications Server)를 소개합니다.

작동하는 하드웨어에 따라 다음과 같은 두 가지 Linux용 IBM 설치 계열이 있습니다.

Linux용 Communications Server

Linux용 Communications Server, 프로그램 제품 번호 5724-i33은 다음에서 작동합니다.

- Linux를 실행하는 32비트 Intel[®] 워크스테이션(i686)
- Linux를 실행하는 64비트 AMD64/Intel EM64T 워크스테이션(x86_64)
- Linux를 실행하는 IBM pSeries[®] 컴퓨터(ppc64)

System z의 Linux용 Communications Server

System z의 Linux용 Communications Server, 프로그램 제품 번호 5724-i34는 System z용 Linux를 실행하는 System z 메인프레임(s390 또는 s390x)에서 작동합니다.

Linux용 Communications Server 빠른 시작 책에는 두 개의 사본이 있는데, 각각 다른 설치 방법을 제공합니다. 올바르게 Linux용 Communications Server를 설치하기 위해서는 해당되는 사본을 구비하고 있어야 합니다. 이 책은 System z의 Linux용 Communications Server에 적용됩니다.

Linux용 Communications Server는 매우 다양한 네트워킹 요구 및 솔루션을 위한 빌딩 블록을 제공합니다. 이는 SNA 네트워크에 있는 노드와 정보를 교환하거나 TCP/IP를 통해 텔넷 프로그램에 대한 호스트 액세스를 제공하는 데 사용될 수 있습니다.

이 책의 사용 방법

이 절에서는 이 책에 들어 있는 정보의 구성 및 표시 방법을 설명합니다.

로드 맵

이 책은 네트워크 계획과 관련된 관리 및 기술 담당자 그리고 Linux 운영 체제용 Communications Server에 관심있는 사용자들을 위한 것입니다.

Linux용 Communications Server를 시작하는 데 필요한 정보를 찾으려면 xii 페이지의 표 1을 사용하십시오.

표 1. 시작하기 로드 맵

용도	참조할 페이지
Linux용 Communications Server에 대해 읽기	1 페이지의 제 1 장 『Linux용 IBM Communications Server에 관한 정보』
사용자 네트워크에서 Linux용 Communications Server의 사용 방법 계획	19 페이지의 제 2 장 『System z의 Linux용 Communications Server 및 네트워크에 대한 계획 수립』
Linux 서버에 Linux용 Communications Server 설치	25 페이지의 제 3 장 『Linux 서버에 Linux용 Communications Server 설치』
Linux에 원격 API 클라이언트(32비트 Intel, 64비트 Intel/AMD 또는 pSeries) 설치	41 페이지의 제 4 장 『Linux에 IBM 원격 API 클라이언트 설치』
System z용 Linux에 원격 API 클라이언트 설치	49 페이지의 제 5 장 『System z용 Linux에 IBM 원격 API 클라이언트 설치』
AIX®에 원격 API 클라이언트 설치	55 페이지의 제 6 장 『AIX 시스템에 IBM 원격 API 클라이언트 설치』
Windows®에 원격 API 클라이언트 설치	61 페이지의 제 7 장 『Windows에 대한 원격 API 클라이언트 설치 계획 세우기』
Linux용 Communications Server 구성	75 페이지의 제 8 장 『Linux용 Communications Server 구성 및 사용』
온라인 정보를 포함하여 Linux 문서 및 기타 서적에 관한 정보 찾기	117 페이지의 제 9 장 『Linux용 Communications Server 및 SNA에 대한 정보 자원』
주의사항 및 등록상표 정보를 읽으려는 경우	119 페이지의 『주의사항』

활자체 규약

표 2에는 이 문서에서 사용된 인쇄 스타일이 나와 있습니다.

표 2. 활자체 규약

특수 요소	활자체 샘플
강조어	삭제 전 파일 백업
문서 제목	<i>Linux용 Communications Server 관리 안내서</i>
파일 또는 경로 이름	/usr/spool/uucp/myfile.bkp
프로그램 또는 응용프로그램	snaadmin
사용자 입력	Op1
컴퓨터 출력	닫기

이 책에서 사용된 약어

이 책은 다음과 같은 약어를 사용합니다.

AIX	Advanced Interactive Executive
API	응용프로그램 프로그래밍 인터페이스
APPC	고급 프로그램간 통신
APPN	Advanced Peer-to-Peer Networking [®] (고급 시스템간 대등 통신)
BrNN	분기 네트워크 노드
COS	서비스 클래스(COS)
CPI-C	Common Programming Interface for Communications(통신용 공통 프로그래밍 인터페이스)
CSV	공통 서비스 Verb
DDDLU	중속 LU의 동적 정의
DES	Data Encryption Standard(데이터 암호화 표준)
DLC	데이터 링크 제어
DLUR	중속 LU 리퀘스터
DLUS	중속 LU 서버
FTP	File Transfer Protocol(파일 전송 프로토콜)
HPR	High-Performance Routing(고성능 경로지정)
IETF	Internet Engineering Task Force
ISO	국제 표준화 기구
ISR	중계 세션 경로지정(ISR)
LAN	근거리 통신망
LDAP	경량 디렉토리 액세스 프로토콜
LEN	Low-entry 네트워킹
LLC2	논리적 링크 제어 2
LU	논리 장치
LUA	Conventional LU Application Programming Interface(Conventional LU 응용프로그램 프로그래밍 인터페이스)
MDS-NMVT	복수 도메인 지원—네트워크 관리 백터 전송
MPC	다중 경로 채널
MS	관리 서비스
NMVT	네트워크 관리 백터 전송
NOF	Node Operator Facility(노드 연산자 기능)
OSI	개방 시스템 상호 접속
PU	물리 장치
RFC	Request For Comments(설명 요청)
RLE	Run-Length Encoding(실행 길이 인코딩)
SAA [®]	시스템 응용프로그램 아키텍처 [®]
SAP	서비스 액세스점
SNA	Systems Network Architecture(시스템 네트워크 아키텍처)
SSL	보안 소켓 계층
TCP/IP	전송 제어 프로토콜/인터넷 프로토콜
TN	Telnet(텔넷)
TP	트랜잭션 프로그램
VT	가상 터미널
WAN	광역 네트워크

새로운 사항

Linux용 Communications Server 버전 6.2.2는 Linux용 Communications Server 버전 6.2 및 Linux용 Communications Server 버전 6.2.1을 대체합니다.

계속 지원되는 이 제품의 릴리스는 다음과 같습니다.

- Linux용 Communications Server 버전 6.2
- Linux용 Communications Server 버전 6.2.1

이 제품의 다음 릴리스는 더 이상 지원되지 않습니다.

- PRPQ 5799-RXL로 사용 가능했던 Linux용 Communications Server 버전 6.0.1(V6.0.1)

새로운 기능

이번 릴리스에서는 다음 기능이 Linux용 Communications Server에 추가되었습니다.

- 이제 클라이언트/서버에 광범위한 서버 및 클라이언트 컴퓨터가 포함됩니다.
 - 서버는 System z용 Linux에서 실행하는 System z 메인프레임(s390 또는 s390x)
 - 클라이언트는 i686, x86_64, ppc64, s390이나 s390x 시스템, AIX 워크스테이션 또는 32비트 버전의 Microsoft® Windows(Microsoft Windows 2000, 2003 또는 XP) 또는 x64 버전의 Microsoft Windows(Microsoft Windows Server 2003 x64 Edition 또는 Microsoft Windows XP Professional x64 Edition)를 실행하는 PC가 될 수 있습니다. 이러한 클라이언트는 TCP/IP를 사용하거나 WebSphere® 서버를 통한 HTTPS를 사용하여 Linux용 Communications Server(또는 CS/AIX 서버)와 통신할 수 있습니다.
 - Linux, System z용 Linux 및 AIX 클라이언트는 System z용 Linux 또는 Linux에서 사용 가능한 모든 API 라이브러리를 지원합니다. (이전 CS/AIX v4.2 제품에 포함되었던 API는 이 버전에서는 지원되지 않습니다.)
 - 32비트 또는 x64 버전의 Microsoft Windows에서 Windows 클라이언트는 APPC, CPI-C, CSV, LUA(LUA RUI와 SLI 모두 포함) 및 NOF API를 지원합니다.
 - 모든 클라이언트 유형에서 NOF API에 대한 클라이언트 지원은 조회 기능만 제공합니다. 자원을 정의, 삭제, 시작 또는 중지하기 위한 호출 기능은 제공되지 않습니다.
 - Windows 클라이언트는 WTS(Windows Terminal Server) 환경에서 실행 가능한데, 여기서 둘 이상의 사용자는 동일한 클라이언트를 공유할 수는 있지만 해당 응용프로그램들은 서로를 간섭하지 않고 독립적으로 실행됩니다.
- SLES10 Linux 분배는 서버와 클라이언트 모두에 대해 더 이상 지원되지 않습니다.
- Linux용 Communications Server는 이제 Linux 서버를 사용하여 SDLC 및 X.25(QLLC) 연결을 지원합니다. 지원을 위해 어댑터 벤더의 장치 드라이버 및 DLC

코드가 필요합니다. IBM은 이러한 어댑터나 장치 드라이버를 제공하지 않습니다. 적합한 WAN 어댑터 벤더에 대한 자세한 내용은 Linux용 Communications Server 지원 웹 페이지를 참조하십시오.

- 이제 TN 서버 및 TN 방향 재지정자에 TN 패스-스루 서비스가 클라이언트 연결을 수신 대기하는 특정 로컬 주소를 지정할 수 있는 기능이 포함되어 있습니다. 이 기능은 선택적입니다. TN3270 클라이언트 연결을 임의의 로컬 주소에서 지원하거나 특정 주소로 제한하도록 서비스를 구성할 수 있습니다.

사라진 기능

클라이언트 및 서버 코드는 RedHat Advanced Server 2.1에서 더 이상 지원되지 않습니다.

제 1 장 Linux용 IBM Communications Server에 관한 정보

이 장에서는 Linux용 Communications Server가 패키징된 방법을 설명하고 이의 기능, 특징 및 이점에 대해 설명합니다.

Linux용 IBM Communications Server 기능 및 패키징

Linux용 Communications Server는 Linux 운영 체제에서 실행되는 통신 소프트웨어입니다. 『Linux용 IBM Communications Server』 및 8 페이지의 『고급 네트워킹 기능』에 설명된 특징들로 구성되어 있습니다.

System z의 Linux용 Communications Server는 서버를 통합하고 네트워크를 단순화하기 위한 솔루션입니다.

- 가상 서버 간의 프로세서 및 메모리 공유 등의 장점을 제공하는 System z 기술을 사용하여 여러 서버를 단일 하드웨어 플랫폼으로 통합할 수 있습니다.
- 이중 SNA와 IP 네트워크를 IP 네트워크로 단순화하여, SNA 트래픽을 IP 네트워크를 통해 System z의 Linux용 Communications Server로 전송할 수 있습니다. 그런 다음 이 Communications Server에서 SNA로 변환하여 보안 및 신뢰성이 뛰어난 z/OS[®] CS와의 인터페이스를 통해 전송할 수 있습니다. 이 네트워크 단순화의 또 다른 장점은 SNA 기술의 필요성이 한 위치로만 집중된다는 것입니다.

Linux용 IBM Communications Server

Linux용 IBM Communications Server는 SNA 및 TCP/IP 네트워크에서 응용프로그램을 연결합니다. 이 Communications Server는 Linux를 실행하는 System z VM 또는 LPAR를 SNA 자원 및 프로토콜을 제공하여 SNA 노드로 변환합니다. 이렇게 하면 SNA 네트워크의 다른 컴퓨터와 통신할 수 있습니다. 또한 Linux용 IBM Communications Server를 해당 TCP/IP 네트워크 내 또는 TCP/IP와 SNA 네트워크 간의 경계에서 사용할 수 있도록 TCP/IP 기능을 제공합니다.

Linux용 Communications Server는 SNA 호스트 컴퓨터와 통신하는 경우 여러 가지 방식으로 작동할 수 있습니다. 2 페이지의 그림 1에서는 Linux용 Communications Server 전개 방법에 대한 두 가지 예제를 보여줍니다.

- 첫 번째 예제에서는 Linux용 Communications Server가 별도의 z800 시스템에 설치되어 기본 z/OS 시스템의 로드를 덜어줍니다. 두 시스템을 연결하는 데 IP를 사용하는 엔터프라이즈 익스텐더 링크나 LLC2 링크를 사용합니다.
- 두 번째 예제에서는 Linux용 Communications Server가 기본 z/OS 시스템에 있는 하나 이상의 VM 또는 LPAR에 설치되어 있습니다. Linux용 Communications Server와 z/OS Communications Server가 동일한 메인프레임에 설치되어 있다고 해

도 이들은 두 개의 분리된 SNA 노드이므로, 이 두 노드 사이에는 여전히 HyperSockets IP를 사용하는 Enterprise Extender 링크나 LLC2 링크가 필요합니다.

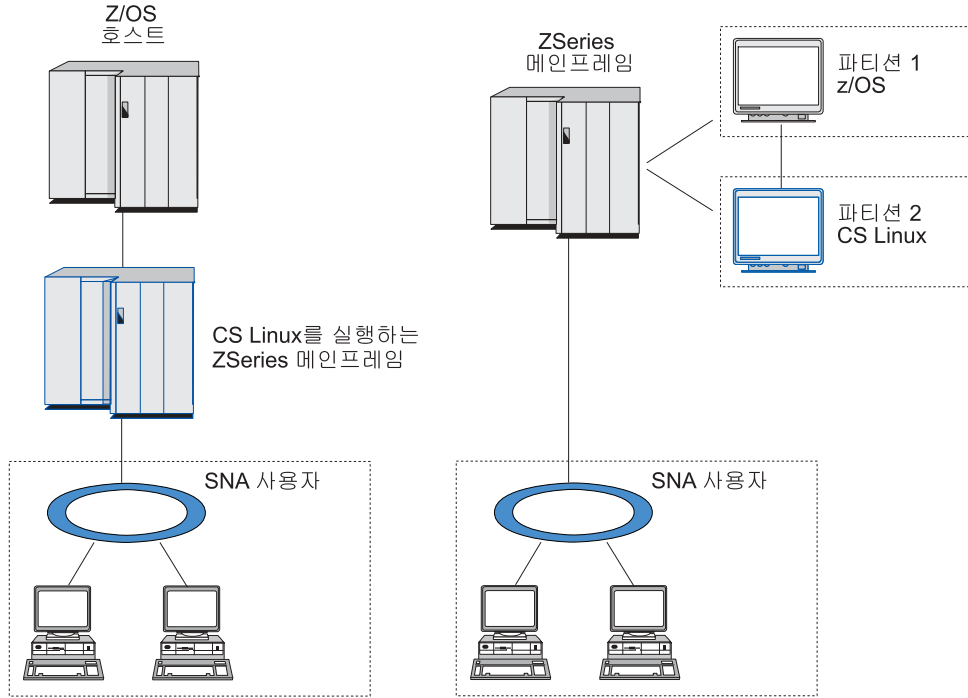


그림 1. System z의 Linux용 Communications Server 호스트 통신

그림에 나타난 두 배열은 개념적으로 동일하며, 두 배열 모두 동일한 Linux용 Communications Server 구성(Linux용 Communications Server와 SNA 호스트 사이의 통신 링크 포함)을 요구합니다. 정확하게 얘기하면, 이 책에 있는 다이어그램은 Linux용 Communications Server와 SNA 호스트가 별도의 컴퓨터에 있는 첫 번째 배열을 보여줍니다.

Linux용 Communications Server는 다음과 같은 서비스를 제공합니다.

네트워크 지원

Linux용 Communications Server는 부속 영역 및 피어간 네트워크를 지원합니다.

SNA 부속 영역 네트워크

이 네트워크(호스트 중재 네트워크라고도 함)는 컴퓨터간 통신을 제어하고 네트워크를 관리하며 처리 서비스 및 고용량 데이터 기억영역을 제공하는 하나 이상의 호스트 컴퓨터와 함께 계층적으로 구성됩니다. 네트워크 내의 다른 모든 노드는 호스트의 제어를 받습니다.

Linux 컴퓨터는 호스트 종속 노드로 구성되어 부속 영역 네트워크에 참여할 수 있습니다.

피어간 네트워크

분산 처리 환경의 경우 Linux용 Communications Server는 APPN 네트워크를 지원합니다. 이러한 피어간 네트워크에서 Linux 컴퓨터는 처리 기능을 보존하고 서로 대등한 자격으로 직접 통신합니다.

APPN 네트워크는 다음 유형의 피어 노드로 이루어집니다.

- APPN 네트워크 노드(트래픽 제어, 동적 경로 계산 및 선택 서비스, 네트워크 관리 서비스 제공)
- APPN 끝 노드(APPN 네트워크 노드 서비스를 사용하여 피어 노드와 통신)
- LEN 노드(인접 노드 또는 인접해 나타나도록 구성된 노드와 통신)

주: 호스트 컴퓨터는 독립적인 LU 6.2를 사용하여 APPN 네트워크에 있는 피어 노드로 기능하여 네트워크 내의 Linux 컴퓨터 및 다른 호스트와 통신할 수 있습니다.

APPN 네트워크에서 부속 영역 기능 제공

종속 LU 리퀘스터(DLUR) 기능은 호스트와 호스트 종속 노드 사이의 트래픽이 APPN 네트워크에서 전달될 수 있도록 합니다.

데이터 링크 제어 옵션

링크 레벨에서 Linux용 Communications Server는 해당 네트워크의 크기, 속도, 보안 및 비용 고려사항을 충족시킬 수 있는 다양한 연결 옵션을 제공합니다. (지원되는 링크 유형의 자세한 목록은 22 페이지의 『설치 요구사항』을 참조하십시오.) 다음과 같이 여러가지 네트워크 유형에 대한 데이터 링크를 지원합니다.

근거리 통신망

LAN 연결성의 경우, 토큰링, 표준 이더넷 및 802.3 이더넷 프로토콜을 사용하여 통신하는 데 적합한 링크를 설치할 수 있습니다.

광역 네트워크(WAN)

Linux용 Communications Server는 SDLC 및 X.25(QLLC) 연결을 지원합니다. 이는 각 플랫폼의 OEM 어댑터 지원에 따라 달라집니다.

로컬 접속

Linux용 Communications Server는 로컬 접속을 위해 MPC(Multipath Channel) 연결을 지원합니다(System z의 Linux용 Communications Server 전용).

IP 통합

회사의 백본(backbone) 네트워크가 IP를 기반으로 하고 있는 경우 Linux용 Communications Server의 Enterprise Extender(HPR/IP) 기

능을 사용하여 이를 SNA와 통합할 수 있으므로 SNA 응용프로그램이 IP 네트워크를 통해 통신할 수 있습니다.

LU 지원

논리적 장치(LU)는 SNA 네트워크 내의 각 노드에 상주하는 응용프로그램 고유의 네트워크 자원입니다. 각 LU는 응용프로그램이 다른 노드의 상대 응용프로그램과 네트워크를 통해 통신하기 위해 링크에 액세스할 때 사용하는 인터페이스 역할을 합니다.

Linux용 Communications Server는 다양한 클래스의 응용프로그램에 대해 다양한 유형의 LU를 지원합니다.

- 부속 영역 네트워크에서 Linux용 Communications Server는 종속 LU를 지원하며 다음 유형 중 하나가 될 수 있습니다.
 - LU 0
 - LU 1
 - LU 2
 - LU 3
 - LU 6.2

LU 0은 원시 프로그램간 통신을 지원하며, 일반적으로 소매상이나 은행 업무에서 POS 트랜잭션에 사용됩니다. LU 2는 Linux 컴퓨터가 IBM 3270 제품군 단말기를 에뮬레이트할 수 있도록 단말기 에뮬레이션 응용프로그램을 지원합니다. 기타 LU 유형은 응용프로그램이 분산 처리에 참여할 수 있게 하거나 여러가지 프린터나 대화식 표시 단말기와 통신할 수 있도록 합니다.

Linux용 Communications Server는 종속 LU의 동적 정의(DDDLU)를 사용하는 호스트 시스템을 지원합니다. DDDLU란 SNA 시스템에서 호스트로 통신 링크가 설정될 때 SNA 시스템의 종속 LU가 호스트 구성에 추가될 수 있도록 하는 호스트 기능입니다. DDDLU를 사용할 경우, LU는 호스트에 정적으로 구성될 필요가 없습니다. (사용자는 여전히 Linux용 Communications Server 노드에 종속 LU를 정의해야 합니다.) 이는 호스트에서 요구되는 초기 구성을 줄여주고, 나중에 쉽게 확장할 수 있게 합니다.

Linux용 Communications Server는 요구되는 구성의 차이 없이 DDDLU 가능 및 DDDLU 불가능 호스트 모두와 통신할 수 있습니다. Linux용 Communications Server 노드에서 호스트로 통신 링크가 설정되면 DDDLU 가능 호스트는 노드에게 DDDLU를 지원한다고 알려줍니다. 그러면 노드는 이 링크를 사용하는 종속 LU를 정의하는 데 필요한 정보를 전송합니다. 호스트가 DDDLU 가능 호스트가 아닐 경우 Linux용 Communications Server는 이 정보를 보내지 않습니다. LU가 이미 호스트에 정적으로 정의되어 있다고 가정합니다.

- 독립 LU 6.2는 APPN 네트워크에서 독립적인 트래픽을 지원합니다. 독립 LU 6.2는 분산 처리뿐 아니라 자율적인 통신 및 네트워크 관리를 지원합니다.

또한 Linux용 Communications Server의 DLUR 기능을 사용하여 종속 LU의 트래픽을 APPN 네트워크를 통해 전달합니다.

- 1차 RUI 지원을 통해 Linux용 Communications Server 응용프로그램은 마치 메인프레임처럼 다운스트림 LAN/WAN에 접속된 종속 LU 장치를 관리할 수 있습니다. 이 기능은 연결에 있어 약간의 제한을 받지만, 완전한 메인프레임 응용프로그램이 없어도 응용프로그램이 종속 LU 장치 간에 데이터를 전달할 수 있도록 합니다.

세션 지원

세션은 상대방 LU 간의 임시적인 논리적 채널입니다. 통상적으로 각 LU와 연관된 상대방 응용프로그램은 이 세션을 통해 통신합니다. Linux용 Communications Server는 수천 개의 세션을 지원할 수 있습니다. Linux용 Communications Server는 U 모양의 세션(1차 및 2차 LU가 같은 Linux 컴퓨터에 상주하는 『로컬/원격 투명성』)도 지원할 수 있습니다. 이 세션은 링크 연결을 필요로 하지 않고 한 컴퓨터에 있는 한 쌍의 소스 및 목표 트랜잭션 프로그램을 개발하고 검사할 수 있도록 합니다.

두 상대방 LU간의 한 세션에서 전달되는 데이터를 압축하여 필요한 대역폭 (bandwidth)을 줄일 수 있습니다.

- LU 유형 6.2의 경우, Linux용 Communications Server를 사용하면 세션에 사용되는 모드 구성에 압축 사용을 지정할 수 있습니다. 다른 여러 압축 알고리즘 사용을 지정할 수 있으며 이 알고리즘 각각은 서로 다른 레벨의 압축을 제공합니다(RLE, LZ9 또는 LZ10). 또한 세션에서 다른 방향으로 전달되는 데이터에 대해 서로 다른 압축 레벨을 지정하거나 한 방향으로만 압축을 지정할 수도 있습니다.
- LU 유형 0-3의 경우, Linux용 Communications Server를 사용하면 세션에 사용되는 PU 또는 링크 스테이션 구성에 압축 사용을 지정할 수 있습니다. RLE 압축은 인바운드 방향에 사용하며 LZ9는 아웃바운드 방향에 사용합니다.

API 지원

Linux용 Communications Server에는 특정 유형의 LU, 분산 처리, 네트워크 관리, Linux용 Communications Server 자체 관리를 위한 응용프로그램 개발에 필요한 응용프로그램 프로그래밍 인터페이스(API)가 포함되어 있습니다. Linux용 Communications Server는 다른 운영 체제에서 실행되는 Communications Server 제품군의 구성원이 제공하는 API와 호환되는 다양한 API를 제공합니다.

API는 트랜잭션 프로그램(TP)이 자신의 지원 LU와 통신할 수 있도록 하는 인터페이스입니다. TP는 SEND_DATA와 같은 조치를 요청하기 위해 해당 LU를 전달하는 데 필요한 명령(Verb)을 선택하는 명령(기능, 호출 및 서브루틴이라고도 함)의 라이브러리로 이루어집니다. 그 다음에 LU는 이 명령을 처리하여 적합한 프로토콜에 따라 데이터 스트림을 빌드하고, 목적지 주소를 나타내는 헤더를 첨부하여, 링크를 통해 데이터를 특정 LU로 보냅니다.

통신용 공통 프로그래밍 인터페이스(CPI-C)는 그 이식성 때문에 가장 강력한 API 중 하나입니다. 종속 및 독립 LU 6.2를 지원하기 위해 소개된 CPI-C는 다른 플랫폼 및 운영 체제를 통합할 수 있도록 시스템 응용프로그램 아키텍처(SAA) 규칙을 준수합니다. CPI-C는 모든 시스템에 공통적인 구문 규칙을 사용하며 표준 API입니다.

Linux용 Communications Server에는 표준 C 언어 CPI-C API뿐만 아니라 Java™ 응용프로그램에서 사용하는 CPI-C API도 포함되어 있습니다. 자세한 내용은 *Communications Server for Linux CPI-C Programmer's Guide*를 참조하십시오. Linux용 Communications Server 책에서 CPI-C에 관한 모든 참조서는 특별히 명시하지 않는 한 Java CPI-C를 포함합니다.

다른 Linux용 Communications Server API에는 다음이 포함됩니다.

- LU 6.2를 사용하는 응용프로그램 프로그램들 간의 시스템간 대등 통신을 위한 APPC API. API는 블로킹이 해제되는 옵션을 가집니다. TP가 블로킹 해제 명령을 사용하면, API는 요청된 조치가 완료되기 전에 제어를 TP로 리턴할 수 있습니다. 나중에 이 TP는 조치가 언제 완료되었는지 통보됩니다.
- 호스트 응용프로그램과 통신을 위한 LUA API.
- 문자 변환 및 응용프로그램 추적 제어와 같은 유틸리티 기능을 위한 CSV(공통 서비스 명령) API.

또한, Linux용 Communications Server는 다음과 같은 관리 프로그래밍 인터페이스를 포함합니다.

- 네트워크 메시지 기능을 위한 MS(관리 서비스) API.
- Linux용 Communications Server 자원을 구성 및 관리하는 응용프로그램을 위한 노드 연산자 기능(NOF) API

API에 대한 자세한 정보는 API의 프로그래밍 안내서를 참조하십시오.

클라이언트/서버 지원

Linux용 Communications Server를 실행하는 컴퓨터는 클라이언트/서버 프로토콜을 사용하여 통신하도록 구성할 수 있습니다. 클라이언트/서버 프로토콜을 한 네트워크에서 사용하는 경우 네트워크 통신을 위해 클라이언트/서버 프로토콜을 사용하는 모든 컴퓨터가 『도메인이 됩니다.』

클라이언트/서버 구성에서 Linux용 Communications Server를 실행하는 컴퓨터는 다음과 같은 역할을 수행할 수 있습니다.

- 서버에 SNA 노드 및 연관된 연결 구성요소가 포함됩니다. 서버는 Linux용 Communications Server 도메인에 있는 로컬 시스템 또는 다른 컴퓨터의 응용프로그램에 SNA 연결을 제공합니다. 서버는 Linux 시스템이어야 합니다.
- 원격 API 클라이언트는 SNA 노드 구성요소를 포함하지 않지만 서버를 통해 이 구성요소에 액세스할 수 있습니다. 클라이언트가 하나 이상의 서버에 동시에 액세스할 수 있으며 필요에 따라 응용프로그램을 동시에 실행할 수도 있습니다. 클라이언트는 Linux 또는 Windows를 실행할 수 있습니다. Linux 컴퓨터는 서버 또는 클라이언트일 수 있지만 두 역할을 모두 수행할 수는 없습니다. 즉, 한 컴퓨터에 서버와 클라이언트를 함께 설치할 수 없습니다.

서버 및 클라이언트는 Linux용 Communications Server 도메인에서 TCP/IP를 사용하여 통신합니다. 또는, HTTPS를 사용하여 WebSphere 서버를 통해 통신할 수도 있습니다. 이 경우 보안 인증서를 사용하여 클라이언트 연결을 인증합니다. 클라이언트가 공용 네트워크를 통해 연결되는 경우 사용자들은 일반적으로 HTTPS를 사용하길 원합니다.

한 도메인에 여러 개의 Linux용 Communications Server가 있는 경우 한 서버가 Linux용 Communications Server 도메인 구성 파일의 마스터 사본을 보관합니다. 이 서버를 마스터 서버라고 합니다. 도메인의 다른 서버를 백업 서버로 정의하거나 피어 서버로 둘 수 있습니다. 도메인 구성 파일이 백업 서버가 시작될 때 또는 마스터 사본의 변경 시에 백업 서버로 복사되어 모든 백업 서버가 최신 정보 사본을 보유하게 됩니다. 피어 서버가 필요에 따라 마스터 서버의 도메인 구성 정보를 확보하지만 백업 서버의 역할은 수행할 수 없습니다. 마스터 서버의 실패 시 도메인에 정의된 서버 목록의 첫 번째 백업 서버가 마스터 역할을 수행합니다. 이 서버의 도메인 구성 파일이 마스터 사본으로 사용되며 필요에 따라 다른 서버로 복사됩니다. 마스터 서버가 다시 시작될 경우 이 서버가 현재 마스터 역할을 수행하는 백업 서버로부터 도메인 구성 사본을 받아 마스터 역할을 수행합니다.

분산 응용프로그램에 대한 지원

클라이언트/서버 Linux용 Communications Server 시스템의 경우, 원격 API 클라이언트에서 실행 중인 응용프로그램은 서버의 연결 자원과 협력하여 단일 작업을 실행합니다. 다른(Linux용 Communications Server 이외의) 컴퓨터에서 실행 중인 응용프로그램도 Linux용 Communications Server 컴퓨터의 응용프로그램과 협력하여 분산 처리를 수행할 수 있습니다.

Linux용 Communications Server는 APPC(LU 6.2라고도 함)를 사용하여 분산 응용프로그램을 지원합니다.

고급 네트워킹 기능

Linux용 Communications Server 기반 제품에는 고급 네트워킹 기능을 추가하는 기능 세트가 포함되어 있습니다. 이들 기능은 다음과 같습니다.

- SNA 게이트웨이는 LAN을 부속 영역 SNA 네트워크에 연결합니다.
- 1차 LU 지원은 호스트 메인프레임 응용프로그램과 동일한 방식으로 다운스트림 종속 LU 장치 제어에 필요한 지원을 제공합니다.
- Branch Extender는 자원을 여러 위치로(예를 들어, 큰 조직의 각 브랜치로) 분리시켜 거대한 APPN 네트워크를 단순화시킵니다. 이 방법은 효율적인 자원 위치를 계속 허용하면서, 저장되어야 하는 토폴로지 정보의 양을 감소시킵니다.
- APPC Application Suite는 APPN 네트워크에서 사용하기 위해 선택된 응용프로그램을 제공합니다.
- Enterprise Extender(EE 또는 HPR/IP로 알려짐)는 SNA 트래픽이 본래부터 IP 네트워크를 통해 전송될 수 있도록 합니다.
- TN 서버는 TN3270 및 TN3270E 클라이언트와 함께 TN3270 클라이언트라고 불리우며, 이들 클라이언트에게 SNA를 통한 호스트 액세스를 제공합니다.
- TN 방향 재지정자는 TN3270, TN3270E, TN5250 및 VT 클라이언트와 함께 텔넷 클라이언트로 불리우며, 이들 클라이언트에게 패스-스루 TCP/IP 호스트 액세스를 제공합니다.

SNA 게이트웨이

게이트웨이는 유사한 네트워크나 컴퓨터 시스템을 연결하는 사용자-트랜스패런트 장치이며, 연결하는 두 환경을 모두 지원합니다. 일반 사용자는 서로 같은 네트워크에 상주하고 있는 것으로 인식합니다.

SNA 게이트웨이를 통해 Linux용 Communications Server 컴퓨터는 9 페이지의 그림 2에 설명된 것처럼, SNA 네트워크에 있는 여러 다운스트림 컴퓨터를 하나 이상의 호스트 물리 장치(PU)에 링크하는 게이트웨이 역할을 수행할 수 있습니다. 호스트 연결을 단순화시키고 과도한 링크를 제거하기 위해 SNA 게이트웨이는 PU 집중기 역할을 수행합니다. 즉, 여러 개의 컴퓨터를 단일 PU(SNA 게이트웨이 노드에 상주하는 것으로 나타남)로 취급하고 하나의 실제 접속으로 호스트와 통신합니다.

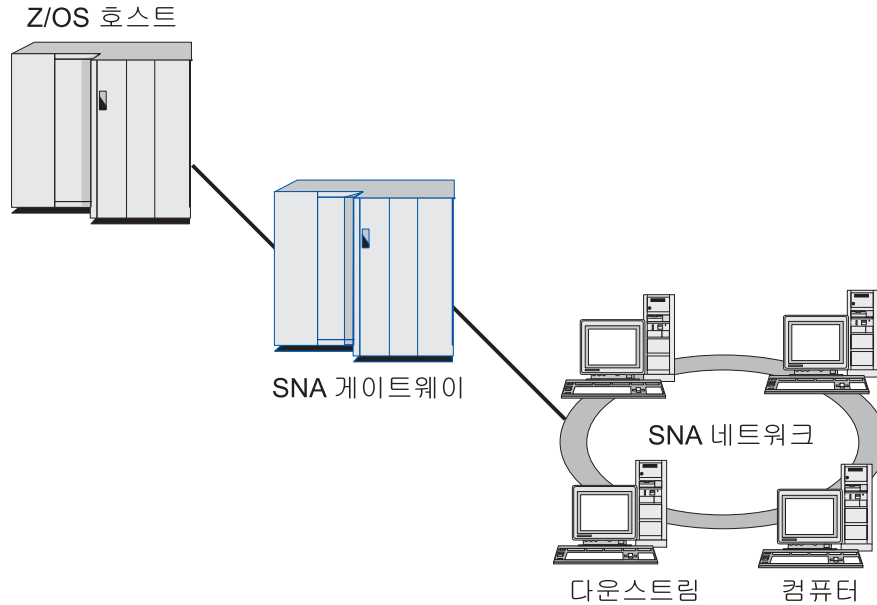


그림 2. 여러 다운스트림 Linux 컴퓨터를 호스트 컴퓨터에 링크하는 SNA 게이트웨이

1차 LU 지원

1차 LU 지원을 통해 Linux 응용프로그램은 메인프레임 호스트 응용프로그램처럼 다운스트림 종속 LU 장치를 제어할 수 있습니다.

LUA 응용프로그램은 일반적으로 2차 LU로 호스트 메인프레임에 연결되므로, 호스트 응용프로그램은 세션 정의를 제어하며 세션 시작을 위해 BIND를 보내야 하는 책임이 있습니다. 또한 Linux용 Communications Server는 1차 RUI 인터페이스를 사용하여 LAN을 통해, 다운스트림 종속 SNA 장치에 대해 1차 LU 역할을 수행할 수도 있습니다. 이 인터페이스를 사용하여 응용프로그램은 호스트 메인프레임이 없어도 다운스트림 종속 LU 세션에 연결할 수 있습니다.

1차 LU 응용프로그램을 사용하려면 노드는 #PRIRUI#라는 호스트 LU 이름을 사용하여 다운스트림 LU(또는 다운스트림 PU 템플릿)와 함께 구성되어야 합니다. 이를 통해 1차 RUI를 사용하는 응용프로그램이 이 PU 및 할당된 LU 자원을 제어한다는 것을 서버에게 알려 줍니다. PU는 LAN 포트에서만 사용될 수 있습니다. 1차 RUI를 사용하도록 응용프로그램을 프로그래밍하는 데 대한 자세한 내용은 *Communications Server for Linux LUA Programmer's Guide*를 참조하십시오.

Branch Extender

APPN 네트워크의 네트워크 노드는 토폴로지 정보(네트워크 내의 다른 노드들의 위치 및 이들 간의 통신 링크에 관한 정보)를 유지하고, 토폴로지가 변경될 때 이 정보를 네트워크에 전송할 필요가 있습니다. 네트워크의 크기가 커짐에 따라, 저장된 정보의 양과 토폴로지와 관련된 네트워크 트래픽량이 더욱 커져서 관리하기 어려워질 수 있습니다.

네트워크를 서브네트워크로 나누어 이러한 문제를 피할 수 있으며, 따라서 각 노드는 자체 서브네트워크에 있는 노드에 관한 토폴로지 정보만 유지하게 할 수 있습니다. 그러나, 이는 다른 서브네트워크에 자원을 배치하려 할 때 네트워크 트래픽이 증가되는 결과가 됩니다.

그림 3에 설명된 APPN의 Branch Extender 기능은 이 문제에 대한 해결책을 제공합니다.

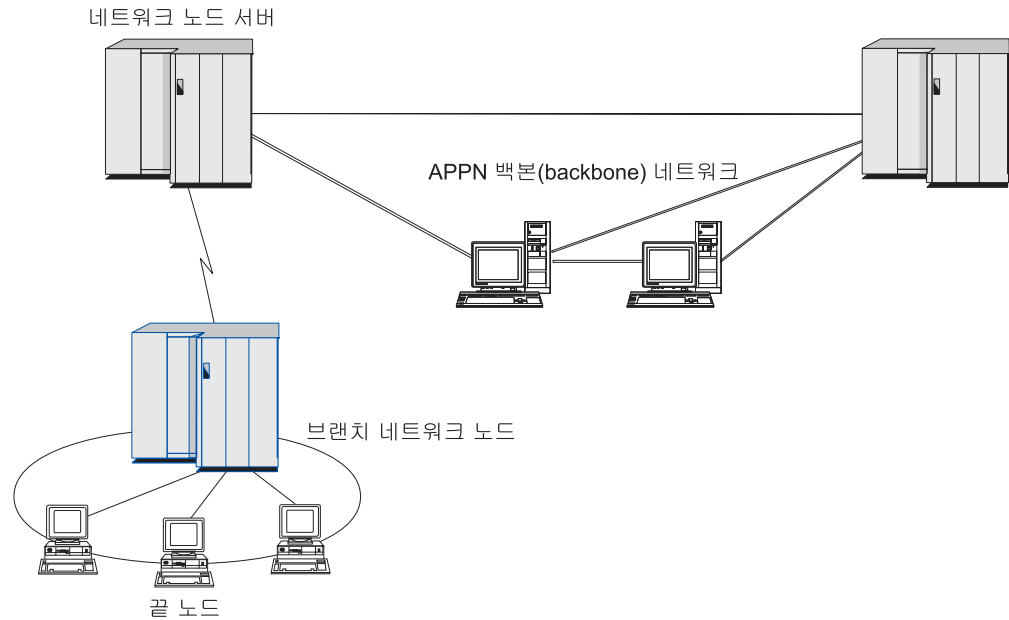


그림 3. Branch Extender

Branch Extender는 그 이름이 나타내듯이 큰 조직의 각 브랜치와 같이 별개의 지역으로 나눌 수 있는 네트워크용으로 설계되었습니다. 기본 백본(backbone) APPN 네트워크(예를 들어, 조직의 본부에 있는 네트워크)에서 브랜치들을 분리시켜 작동합니다.

각 브랜치는 브랜치 네트워크 노드(BrNN)라는 새로운 유형의 노드를 갖는데, 이 노드는 기본 APPN 백본(backbone) 네트워크에 있는 네트워크 노드에 연결됩니다. BrNN은 APPN 네트워크 노드와 APPN 끝 노드의 기능을 결합합니다.

- 백본(backbone) 네트워크에 대해 BrNN은 끝 노드로 나타나며, 백본(backbone) 네트워크에 있는 해당 네트워크 노드 서버(NNS)에 연결됩니다.
 - 백본(backbone) 네트워크에 있는 노드는 브랜치 내에 있는 노드를 알아 보지 못하며, 저장되어야 하는 토폴로지 정보의 양을 줄여줍니다.
 - BrNN은 끝 노드로 나타나기 때문에 백본(backbone) 네트워크에서 토폴로지 정보를 수신하지 않습니다. (토폴로지 정보는 네트워크 노드 사이에서만 전송됩니다.)

- BrNN은 자원들이 마치 BrNN 자체에 위치된 것처럼, 해당 NNS가 있는 브랜치에 모든 자원을 등록합니다. 이는 백본(backbone) 네트워크에 있는 노드가 브랜치에 있는 각 노드를 인식할 필요없이 브랜치에 자원을 위치시킬 수 있음을 나타냅니다.
- 브랜치 네트워크에 대해, BrNN은 네트워크 노드로 나타나며, 브랜치에 있는 끝 노드에 대한 NNS 역할을 합니다. 브랜치에 있는 각 노드는 나머지 네트워크가 표준 NNS의 경우와 같은 방법으로 해당 NNS를 통해 연결되는 것으로 파악합니다.

APPC Application Suite

APPC Application Suite는 APPN 네트워크의 분산 처리 기능을 구현하는 응용프로그램 세트로, 구성 검증 및 문제점 진단에 도움이 될 수 있습니다. APPC Application Suite는 네트워크에서 자주 수행되는 파일 전송과 같은 조작에 대한 지원을 제공하는 데 사용될 수 있습니다.

APPC Application Suite에는 다음과 같은 응용프로그램들이 포함되어 있습니다.

- **ACOPY**(APPC COPY)
- **AFTP**(APPC File Transfer Protocol)
- **ANAME**(APPC Name Server)
- **APING**(APPC Ping)
- **AREXEC**(APPC Remote EXECution)
- **ATELL**(APPC TELL)

이 응용프로그램은 Linux 또는 Windows 클라이언트나 서버에서 액세스할 수 있습니다.

Enterprise Extender

Enterprise Extender(HPR/IP로 알려짐)는 SNA 응용프로그램을 IP 네트워크와 통합시키기 위한 메커니즘을 제공합니다.

SNA 응용프로그램은 SNA 프로토콜을 사용하여 SNA 네트워크를 통해 다른 SNA 응용프로그램과 통신하도록 설계되었습니다. Enterprise Extender를 사용하여 TCP/IP 네트워크에서 설치될 때 SNA 응용프로그램은 계속 통신할 수 있으며, Enterprise Extender 기능은 IP 네트워크를 통해 SNA 프로토콜을 전송하기 위한 메커니즘을 제공합니다. 특히, APPN 고성능 경로지정(HPR) 기능을 제공하며, 응용프로그램에 APPN과 IP 연결성의 이점을 제공합니다.

Linux용 Communications Server의 Enterprise Extender는 통신 링크로 간단히 구현됩니다. IP를 통해 두 개의 SNA 응용프로그램을 연결하려면 다른 링크 유형(예: SDLC 또는 이더넷)과 동일한 방법으로 Enterprise Extender 링크를 정의합니다.

TN 서버

TCP/IP를 통해(SNA 네트워크를 통하지 않고) 통신하는 3270 에뮬레이션 프로그램은 『TN3270 프로그램』(텔넷 3270 에뮬레이션 프로그램)으로 일컬어집니다.

TN3270 프로그램은 TN3270E(Telnet 3270 표준 확장)를 위한 지원도 포함할 수 있습니다. TN3270E는 (단말기와 프린터를 포함하여) 텔넷을 사용하는 3270 장치 에뮬레이션을 지원합니다. 텔넷 클라이언트가(LU 이름 또는 LU 풀 이름을 지정하여) 특정 장치를 선택할 수 있게 하며, ATTN과 SYSREQ 키 및 SNA 응답 처리를 포함하는 여러 가지 기능에 대한 확장 지원을 제공합니다.

주: 이 안내서에서는 TN3270, TN3287 및 TN3270E 프로토콜에 동등하게 적용되는 정보에 대해 TN3270이라는 용어를 사용합니다.

Linux용 Communications Server TN 서버를 사용하면 다른 컴퓨터의 TN3270 사용자가 3270 호스트 컴퓨터에 액세스할 수 있습니다. TN 서버를 사용하면 TN3270 사용자가 직접 링크 없이도 Linux용 Communications Server 또는 다른 TN3270 사용자와 호스트 연결을 공유할 수 있습니다. 또한 TN 서버는 TN3270 사용자가 TCP/IP를 실행하고 있지 않는 호스트에 액세스할 수 있게 해 줍니다.

Linux용 Communications Server TN 서버 기능이 그림 4에 설명되어 있습니다.

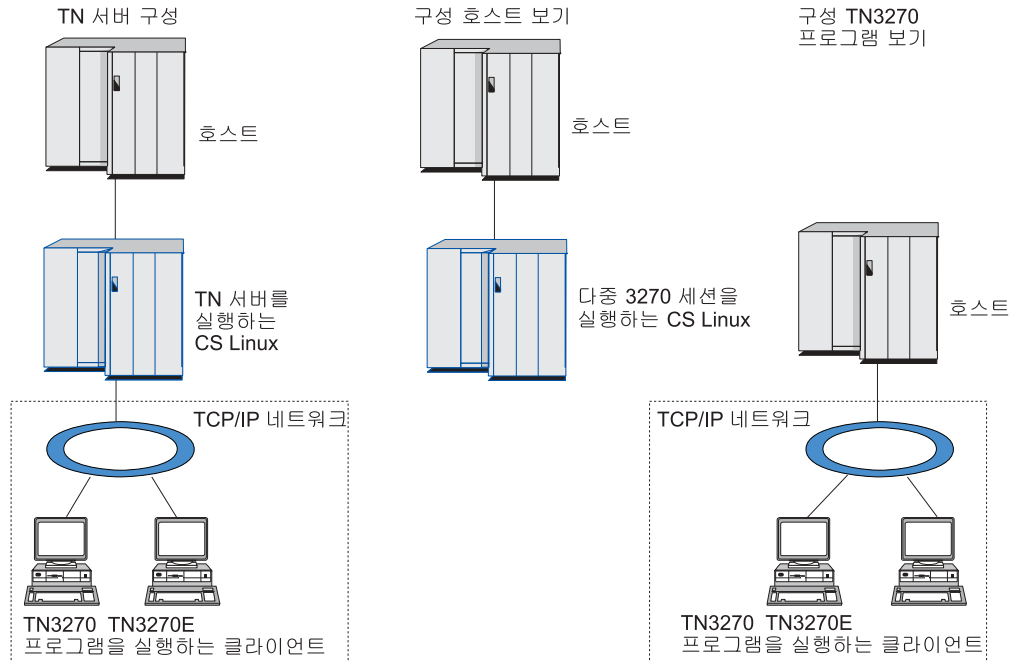


그림 4. TN 서버

Linux용 Communications Server TN 서버 기능은 TN3270 사용자와 Linux용 Communications Server 3270 LU 사이의 연결을 제공합니다. TN3270 사용자로부터

의 모든 데이터의 경로가 LU로 지정됩니다. 이는 호스트와 TN3270 사용자 둘 다를 위한 구성이 직접 연결된 것과 같다는 것을 의미합니다. 따라서 데이터의 경로가 TN 서버를 통해 지정되고 있음을 알 필요가 없습니다.

Linux용 Communications Server TN 서버는 IETF RFC 1123, 1576, 1646, 1647 및 2355에 정의된 프로토콜을 정확하게 구현하는 모든 TN3270 클라이언트 에뮬레이션 프로그램을 지원합니다.

보안 기능: Linux용 Communications Server TN 서버는 보안 소켓 계층(SSL) 소프트웨어를 사용하여 데이터 암호화, 서버 인증, 클라이언트 인증 및 익스프레스 로그온을 지원합니다.

- 데이터 암호화는 TN 서버와 TN3270 에뮬레이터 간에 흐르는 데이터가 암호화된 형식으로 되는 것을 의미합니다.
- 서버 인증은 TN3270 클라이언트가 자신이 연결한 TN 서버가 예상하는 서버인지 확인할 수 있도록 합니다.
- 클라이언트 인증은 TN 서버가 연결 중인 TN3270 클라이언트가 예상하는 클라이언트인지 확인할 수 있도록 합니다. 또한 TN 서버는 클라이언트의 인증이 취소되지 않았는지 확인하기 위해 외부 디렉토리에 있는 취소 목록을 확인할 수 있게 되었습니다.
- 익스프레스 로그온은 클라이언트 인증과 함께 작동하므로 호스트 연결 시 TN3270 클라이언트가 요구하는 사용자 ID 및 암호를 제공하지 않아도 됩니다. 대신 클라이언트의 보안 인증서는 필요한 사용자 ID 및 암호 정보를 검색하는 데 사용됩니다.

TN 방향 재지정자

Linux용 Communications Server TN 방향 재지정자 기능은 TCP/IP를 통해 3270, 5250 또는 VT 세션에 대해 패스-스루 서비스를 제공합니다. 텔넷 사용자는 TCP/IP 연결을 통해 Linux용 Communications Server와 통신하며, Linux용 Communications Server는 다른 TCP/IP 연결을 통해 호스트와 통신합니다.

Linux용 Communications Server TN 방향 재지정자는 3270용 TN 서버에서와 같은 방법으로 보안 소켓 계층(SSL) 소프트웨어를 사용하여 데이터 암호화, 서버 인증 및 클라이언트 인증을 지원합니다. 이 방법은 필요할 때 보안 소켓 계층(SSL) 보안 검사를 사용할 수 있도록 하며 완전한 사용자-호스트 연결에서는 사용하지 않습니다. 예를 들어 다음을 지정하십시오.

- 클라이언트가 확인이 필요하지 않은 TCP/IP LAN을 통해 Linux용 Communications Server에 연결되어 있지만 SSL을 요구하는 원격 호스트에도 연결되어 있을 경우, Linux용 Communications Server와 호스트 사이의 TCP/IP 연결을 통해 SSL을 사용할 수 있습니다. 이는 모든 클라이언트에 대해 보안이 한 번 검사되고, 각 클라이언트는 보안 정보를 제공할 필요가 없음을 의미합니다.

- Linux용 Communications Server가 호스트와 같은 사이트에 설치되어 있지만 클라이언트가 외부 사이트에서 연결되는 경우, SSL 소프트웨어를 호스트에 설치하지 않아도 Linux용 Communications Server와 클라이언트 간 연결을 통해 SSL을 사용할 수 있습니다.

기능 및 이점

Linux용 Communications Server에는 구성 단순화부터 문제점 진단 개선에 이르기까지 네트워크 성능을 향상시킬 수 있는 다양한 기능 및 이점이 있습니다.

융통적인 빌딩 블록

Linux용 Communications Server는 대부분의 환경 및 노드 기능을 지원합니다. 모든 유형의 네트워크, 부속 영역 또는 APPN에서 Linux 컴퓨터가 다음 중 하나 또는 여러 기능을 수행할 수 있도록 합니다.

- 호스트 종속 노드
- 피어 노드(APPN 피어 노드의 설명에 대해서는 1 페이지의 『Linux용 IBM Communications Server』에 있는 대등 네트워크의 설명을 참조하십시오.)
- 분산 응용프로그램의 상대방(소스 또는 목적지)
- SNA 네트워크를 상호연결하는 게이트웨이 노드

네트워크 관리 API를 통해 Linux 컴퓨터는 분산 네트워크 관리를 지원하는 관리 서비스(MS) 시작점 기능을 수행하도록 구성될 수도 있습니다. 링크 레벨에서 Linux 컴퓨터는 지원 링크 유형(1 페이지의 『Linux용 IBM Communications Server』 및 22 페이지의 『설치 요구사항』에서 설명)을 사용하여 다양한 LAN 및 WAN에 연결될 수 있습니다.

클라이언트/서버 운영

클라이언트/서버 구성에서 다음과 같은 이점을 제공합니다.

- 서버의 SNA 자원을 집중시키면 클라이언트의 로드를 줄이고 클라이언트 성능을 향상시키며 SNA 서비스를 클라이언트로 제공하는 데 필요한 기억영역을 최소화시킬 수 있습니다.
- 다른 시스템에서 여러 사용자가 단일 데이터 링크를 공유할 수 있으면 각 시스템이 실제 SNA 네트워크 연결을 수행할 필요가 없습니다.
- 여러 개의 서버가 중복 연결을 제공할 수 있습니다(예를 들어 여러 개의 서버가 동일한 호스트로 액세스 제공). SNA 자원으로 연결되는 경로가 여러 개 있으면 서로 다른 여러 서버의 로드가 균형을 유지하게 되고 특정 서버 또는 링크 실패 시에 바로 백업이 제공됩니다.
- 여러 개의 서버에 LU 풀을 사용하면 관리자가 서버 및 사용자를 쉽게 구성 및 추가할 수 있습니다.

- 호스트 연결을 위한 링크 및 PU 수를 적게하면 호스트 VTAM® 정의의 크기가 줄어듭니다.
- 관리 유틸리티를 사용하여 노드 자원(도메인에 있는 컴퓨터의 경우)과 공유 자원 모두를 구성 및 관리할 수 있습니다. Linux용 Communications Server 관리 도구에서 제공하는 클라이언트/서버 지원 기능을 사용하면 도메인에 있는 전체 컴퓨터의 모든 도메인 자원을 명확하게 관리할 수 있습니다.
- SNA 응용프로그램은 인증 및 보안을 위해 그리고 방화벽을 통과하기 위해 HTTPS 및 TCP/IP를 사용하여 인터넷 프로토콜을 통해 연결될 수 있습니다.

쉬운 구성

Linux용 Communications Server는 구성 시간과 네트워크 복잡성을 줄이는 구성 옵션과 기능을 사용하여 설계됩니다. 예를 들어 다음을 지정하십시오.

Motif 관리 프로그램

Linux용 Communications Server 구성을 정의 및 수정하는 가장 쉬운 방법은 Motif 관리 프로그램(**xsnaadmin**)을 사용하는 것입니다. 이 프로그램은 사용자가 Linux용 Communications Server 자원을 보거나 관리할 수 있는 그래픽 사용자 인터페이스를 제공합니다. 이 프로그램은 또한 일반적으로 설치에 따라 값이 변하는 필드만 표시하고, 다른 필드에 대해서는 디폴트 값을 사용하여 구성을 단순화시킵니다.

Motif 관리 프로그램에는 SNA 및 Linux용 Communications Server에 대한 개요 정보, Linux용 Communications Server 대화상자에 대한 참조 정보, 그리고 특정 타스크 수행을 위한 지침을 제공하는 도움말 화면이 포함되어 있습니다.

APPN 네트워크에서 동적 구성

노드나 네트워크의 구성은 APPN 네트워크의 동적 구성을 통해 보다 쉽게 수행됩니다. 예를 들어, APPN 끝 노드와 응용프로그램은 LU 6.2 세션을 지원하기 위해 구성 데이터를 동적으로 등록하여 세션 구성을 선택적으로 만듭니다. 그리고, 노드 제어점을 디폴트 로컬 LU로 작동하게 하여, LU 6.2 구성을 모두 피할 수 있습니다.

APPN은 또한 구성된 링크 스테이션이 없는 상태로 동적 링크 스테이션 구성을 지원합니다.

관리에 대한 추가 사용자 인터페이스 선택사항

Motif 관리 프로그램은 Linux용 Communications Server 구성 및 관리를 위해 권장되는 인터페이스입니다. 그러나 Linux용 Communications Server에 대한 인터페이스는 선택이 가능하므로, 사용자의 장치, 요구 및 선호도에 맞는 인터페이스를 사용하여 작업할 수 있습니다.

명령행 관리 프로그램

명령행 관리 프로그램(**snaadmin**)은 개별 Linux용 Communications Server 자원을 관리하기 위한 명령을 실행하는 데 사용할 수 있습니다. Linux 명령 프롬프트나 셸 스크립트에서 직접 **snaadmin**을 사용할 수 있습니다.

NOF API

Linux용 Communications Server NOF API는 명령행 관리 프로그램과 동일한 관리 기능을 제공하며, 명령 스크립트가 아니라 프로그램 내에서 사용하기에 적합한 인터페이스를 제공합니다. NOF API를 사용하여 Linux용 Communications Server를 관리하기 위한 사용자 전용 응용프로그램을 작성할 수 있습니다.

더 좋은 성능

Linux용 Communications Server는 SNA 네트워크의 우수한 성능을 더욱 강화하였으며 서비스 운영 클래스를 사용합니다. Linux용 Communications Server는 또한 다음과 같이 네트워크 크기에 따라 트래픽 흐름을 조정하는 여러 가지 트래픽 관리 방법과 LU 0-3 세션 데이터에 대한 SNA 데이터 압축을 통해 네트워크 속도를 최적화합니다.

- APPN 네트워크에서 Linux용 Communications Server는 고성능 경로지정(HPR) 및 중계 세션 경로지정(ISR)을 지원하고 연결 네트워크 옵션을 제공합니다. ISR이 소형 네트워크에 대해서는 효율적으로 작동하는 반면, 보다 큰 네트워크의 성능은 떨어뜨립니다.
- LAN 연결성 옵션(토큰링 또는 이더넷)을 사용하거나 Enterprise Entender를 사용하는 대형 네트워크의 경우 연결 네트워크 옵션을 사용하여 통신 효율성을 향상시킬 수 있습니다. 연결 네트워크 옵션은 노드 간의 통신 경로를 직접적으로 작성합니다. 이는 중간 네트워크 노드를 우회하는 방법으로 전송될 수 있게 합니다.
- 다른 트래픽 제어 메커니즘, 적응 세션 레벨 속도는 LU가 상대 LU에게 메시지 단위를 보내는 전송률을 규제하여 자동으로 정체를 조정합니다.

보안 옵션

네트워크가 더욱 복잡해지고 개방형 구조로 이동해감에 따라, 보안이 중요한 문제로 나타나고 있습니다. Linux용 Communications Server가 실행 중인 SNA 네트워크에서 특정 유형의 링크를 구현하고 구성을 통해 다양한 보안 레벨을 정의하여 자원을 보호할 수 있습니다. 예를 들어 다음을 지정하십시오.

- 클라이언트/서버 시스템에서 WebSphere 서버를 설정하여 원격 API 클라이언트에서 서버로 HTTPS 액세스를 제공할 수 있습니다. 이는 보안 인증서를 사용하여 클라이언트 연결이 인증됨을 의미합니다. (이 기능에는 표준 Linux용 Communications Server 제품과 일부 추가 소프트웨어가 필요합니다. 자세한 내용은 22 페이지의 『설치 요구사항』을 참조하십시오.)

- LU 6.2 사용자는 세션, 자원 및 변환의 세 개 레벨의 보안을 정의할 수 있습니다. 각기 올바른 LU가 세션에 예약되어 있는지 확인하고 특정 LU에 연관된 모든 응용 프로그램에 대한 액세스를 제한하며 특정 응용프로그램에 대한 액세스를 제한합니다. 데이터 암호화 루틴을 통해 추가 보안이 가능합니다.
- Linux용 Communications Server TN 서버 및 TN 방향 재지정자는 보안 소켓 계층(SSL) 소프트웨어를 사용하여 Linux용 Communications Server와 TN3270 또는 텔넷 클라이언트 간에 데이터 암호화, 서버 인증 및 클라이언트 인증을 제공할 수 있습니다. (이 기능에는 표준 Linux용 Communications Server 제품과 일부 추가 소프트웨어가 필요합니다. 자세한 내용은 22 페이지의 『설치 요구사항』을 참조하십시오.)

네트워크 관리 융통성

Linux용 Communications Server는 복수 도메인 지원-네트워크 관리 벡터 전송 (MDS-NMVT)의 네트워크 관리 구조를 지원하는데, 이 관리 구조는 집중, 분산 또는 계층적 관리 구조로 작동될 수 있습니다. 이는 높은 융통성을 제공하는 포컬(focal)포인트/엔트리 포인트 아키텍처를 기반으로 하고 있습니다.

포컬(focal) 포인트는 엔트리 포인트(네트워크내의 다른 모든 노드에 상주하는 관리 응용프로그램)에서 수집한 데이터에 따라 네트워크를 관리하는 제어 노드입니다.

- 중앙 관리의 경우, 단일 포컬(focal) 포인트는 전체 네트워크에 대한 제어 포인트로 작동합니다.
- 분산 관리의 경우, 여러 포컬 포인트가 네트워크 관리를 공유합니다.
- 계층적 관리의 경우, 포컬 포인트는 기능에 따라 중첩되어 있습니다.

따라서 MDS-NMVT는 부속 영역, 표준 APPN 및 매우 큰 APPN 네트워크를 관리하는 데 적합합니다.

신뢰성, 사용 가능성 및 서비스 가능성

Linux용 Communications Server는 안정적인 시스템 운영에 도움이 되는 다양한 표시 기능 및 문제점 진단 도구를 제공합니다.

- Motif 관리 프로그램은 다음을 포함하는 확장 구성 및 관리 도구를 제공합니다.
 - 구성 정보의 즉시 갱신
 - 링크, 세션 및 노드 자원에 대한 상태 정보
- 조회 및 상태 명령은 다음에 대한 정보를 제공합니다.
 - LU-LU 세션
 - APPN 중계 세션
 - 활성 링크
 - 링크 정보를 저장하는 APPN 토폴로지

- 문제점 진단 도구는 다른 구성 및 작동 단계에서 사용자에게 도움을 줄 수 있습니다. 다음과 같은 기능을 포함합니다.
 - 서비스 정보를 쉽게 수집할 수 있게 하는 진단 정보 수집 도구(snagetpd)
 - 구성 루틴 및 시스템 운영에 대한 문제점을 해결하는 데 도움을 주는 상태 및 오류 메시지
 - 네트워크 오류, 장애 및 감사 정보를 수집하기 위한 로그
 - 자세한 문제점 정보를 수집하고 형식화하기 위한 추적 설비

다른 유틸리티들은 응용프로그램 간의 링크 연결성 및 통신을 검사하는 데 도움을 줍니다.

Linux용 Communications Server에는 네트워크 경고 및 문제점 데이터를 송수신하는 도구 개발에 사용되는 관리 서비스 API도 포함되어 있습니다.

이러한 관리 및 문제점 진단 도구 모두가 Linux용 Communications Server 클라이언트/서버 모델에 완전히 통합되어 있으므로, Linux용 Communications Server 도메인 전체를 관리하거나 네트워크의 단일 지점으로부터 진단 정보를 수집할 수 있습니다.

네트워크 통합, 성장 및 변경

네트워크 통합, 추가 및 변경을 지원할 수 있도록 Linux용 Communications Server API를 사용하여 해당 비즈니스 요구에 따라 특정 LU, 플랫폼 또는 운영 체제에 적합한 응용프로그램을 개발할 수 있습니다. CPI-C는 다른 플랫폼 및 운영체제에서 일관성을 지키므로 특히 중요한 API입니다. 모든 시스템에서 실행될 수 있는 응용프로그램을 개발할 때 사용됩니다.

Enterprise Extender 또한 SNA와 TCP/IP 네트워크를 통합하기 위한 메커니즘을 제공합니다.

제 2 장 System z의 Linux용 Communications Server 및 네트워크에 대한 계획 수립

이 장에서는 Linux용 Communications Server를 실행하는 네트워크에 대한 계획 단계를 개괄적으로 설명합니다. Linux 컴퓨터에 대해 구성할 수 있는 기능을 요약하고 이 기능을 지원하는 데 필요한 자원을 예측하기 위한 지침을 제시합니다.

네트워크 계획 단계

이 절에서는 여러 Linux용 Communications Server 및 Linux 유틸리티를 사용하여 네트워크를 계획, 구성 및 관리하기 위한 일반적인 몇몇 지침을 제시합니다.

네트워크 계획에는 균형화 기능, 성능, 자원 및 비용이 포함됩니다. 네트워크에 대해 단 하나의 최상의 계획은 없어도 몇가지 일반적인 지침과 기술을 사용하여 사용자 요구에 맞도록 계획을 수립할 수 있습니다. 네트워크 계획을 세우려면 다음 작업을 수행하십시오.

- (파일 전송 또는 3270 에뮬레이션과 같은) 네트워크가 제공해야 하는 기능과 성능 요구사항을 결정하십시오.
- 사용자에게 필요한 기능을 제공할 수 있도록 Linux용 Communications Server를 구성하는 방법을 결정하십시오.
- 성능 및 용량 요구사항과 Linux용 Communications Server 기능을 지원하도록 Linux용 Communications Server를 설치하는 데 필요한 자원을 예측하고 관련 비용을 파악하십시오.

네트워크에 대한 기능적 요구사항 식별

네트워크가 제공해야 하는 기능을 결정하려면 다음과 같은 사항을 고려해야 합니다.

- 네트워크는 APPN이 되어야 하는가?
- Linux용 Communications Server가 클라이언트/서버 시스템으로 실행됩니까? 그렇다면 모든 컴퓨터가 단일 Linux용 Communications Server 도메인에서 운영됩니까? 아니면 두 개 이상의 각기 다른 도메인을 정의해야 합니까?
- 연결 자원의 로드 균형을 유지하기 위해 Linux용 Communications Server 도메인에서 두 개 이상의 서버가 필요합니까? 그렇다면 어떤 서버가 마스터 구성 서버가 되어야 합니까? 하나 이상의 백업 구성 서버를 제공해야 합니까?
- HTTPS를 사용하여 Linux용 Communications Server에 연결된 원격 API 클라이언트를 지원해야 합니까?
- 사용자 응용프로그램이 서버 또는 Linux 클라이언트 컴퓨터에서 실행됩니까?

- 서버가 Windows 클라이언트에서 실행 중인 Windows 응용프로그램(예: API 트랜잭션 프로그램)에 대해 연결 자원을 제공합니까?
- 서버가 세션에 대해 끝점이 되어야 합니까? 아니면 다음 게이트 유형 중 하나가 되어야 합니까?
 - APPN
 - LU 0
 - LU 2
 - TN 서버 또는 TN 방향 재지정자
- 네트워크가 어떤 유형의 물리적 링크를 사용할 것인가?

이러한 질문에 대한 답변은 네트워크가 요구하는 Linux용 Communications Server 기능을 결정하는 데 도움을 줍니다.

Linux용 Communications Server 구성 방법 결정

Linux용 Communications Server의 기능 수행 방법을 결정하려면 먼저 네트워크를 통한 작업 흐름 방법을 결정해야 합니다. 다음 사항을 고려해야 합니다.

- 네트워크를 통해 어떤 자원(예: 응용프로그램)이 사용 가능해야 하는가?
- 몇 명의 사용자가 원격 자원에 액세스해야 하는가?
- 각 자원은 얼마나 자주 액세스되는가?
- 사용자는 어떻게 네트워크에 액세스할 수 있는가?
- 사용자의 요청은 어떻게 네트워크를 통해 경로지정되는가?

예를 들어 다음과 같은 다수의 기능을 지원하도록 Linux용 Communications Server를 구성할 수 있습니다.

- 중계 세션 경로지정(ISR)을 위한 APPN 네트워크 노드
- APPN 끝 노드(인접 노드와 자율적으로 통신하지만 인접하지 않는 피어 노드와 통신하기 위해 APPN 네트워크 노드 서비스를 사용함)
- LEN(Low-entry network) 노드(인접 노드 또는 인접해 나타나도록 구성된 노드와 직접 통신함)
- LU 0, LU 1, LU 2, LU 3 및 LU 6.2(종속 및 독립)의 사용
- 호스트와 SNA 게이트웨이 연결

사용자 요구에 따라 지정 노드에 대해 하나 이상의 기능을 구성할 수 있습니다. 예를 들어 Linux용 Communications Server를 경로지정 서비스와 ISR을 제공하기 위한 APPN 네트워크 노드로 구성하고, SNA 게이트웨이에 대해 같은 노드를 사용하여 LU 0 및 LU 2와 같은 종속 LU 세션을 경로지정할 수 있습니다. 이와 마찬가지로 호스트의 MQSeries®와 연결되는 독립 LU 6.2 연결뿐 아니라, 공유 데이터베이스를 지원하고 TN 서버를 실행하도록 Linux용 Communications Server를 구성할 수 있습니다.

설치 및 운영을 위한 자원 요구사항 식별

Linux용 Communications Server의 기능에 대한 지원을 예측하려면 다음 사항을 고려해야 합니다.

- 담당자의 어떤 기술이 필요한가?
- 어떤 전송 매체를 사용해야 하는가?
- 어떤 설치 요구사항을 구성에 선택해야 하는가?
- 운영에 필요한 메모리와 페이지 공간은 얼마인가?

이 질문에 대한 답변은 Linux용 Communications Server가 19 페이지의 『네트워크에 대한 기능적 요구사항 식별』에 설명된 여러 기능 중 일부를 지원하도록 구성되었을 때 사용하는 자원 유형을 식별하는 데 도움을 줍니다. 이 질문에 답변은 Linux용 Communications Server 기능, Linux 자원 및 네트워크 자원 간의 관계를 이해하는 데에도 도움이 됩니다.

노드에 자원을 할당하는 방법은 네트워크의 수행 방법을 결정합니다.

요원 요구사항

Linux용 Communications Server를 설치, 운영 및 조정할 경우 다음과 같은 인원이 필요합니다.

- 네트워크 관리자. 네트워크를 계획하고 새로운 장치를 추가하며, 전체적인 네트워크 성능을 유지보수하고 확장합니다.
- 시스템 관리자. Linux용 Communications Server와 이 서버가 운영되는 하드웨어를 설치 및 유지보수하고, 네트워크 연결을 위해 시스템을 구성합니다.
- 프로그래머. 트랜잭션 프로그램 또는 네트워크 관리 루틴과 같은 사용자 조정된 응용프로그램을 개발합니다.

네트워크 및 시스템 관리자는 Linux용 Communications Server가 운영되는 하드웨어와 Linux 운영 체제에 대해 잘 알고 있어야 합니다. 또한 다양한 시스템이 연결되는 네트워크에 대해서도 잘 알아야 하며 일반적인 SNA 개념도 이해하고 있어야 합니다. 다음에 대해서도 잘 알고 있어야 합니다.

- Motif 인터페이스
- rpm, Linux용 설치 도구
- 클라이언트/서버 기능, TN 서버 또는 Enterprise Extender를 사용하려고 계획한 경우, TCP/IP
- Linux용 Communications Server 시스템에 Windows 원격 API 클라이언트가 포함된 경우 Windows 2000, Windows XP 또는 Windows 서버 2003 운영 체제
- Linux용 Communications Server 시스템에 HTTPS를 사용하여 서버에 연결하는 원격 API 클라이언트가 포함된 경우 WebSphere Application Server

SNA용으로 사용자 정의된 응용프로그램을 개발하는 프로그래머는 C 언어 사용 경험이 있고(또는 Java CPI-C를 사용하는 경우 Java 사용 경험 필요) Linux용 Communications Server에서 제공하는 API에 익숙해야 합니다.

전송 매체

Linux용 Communications Server는 다른 통신 프로토콜과 기본 전송 매체(예: SDLC, 토큰링)를 공유해야 하는 경우가 있습니다. 따라서, 물리적 계층 대역폭(bandwidth) 요구사항은 이 전송 매체를 공유하는 모든 프로토콜 및 응용프로그램에 적합해야 합니다.

주: Linux용 Communications Server는 토큰링 및 이더넷 어댑터를 다른 프로토콜(예 : TCP/IP)과 공유할 수 있습니다. 사용할 각 프로토콜에 대해 고유한 서비스 액세스 점(SAP) 주소를 지정해야 합니다.

설치 요구사항

Linux용 Communications Server에 지정하는 기능(19 페이지의 『네트워크에 대한 기능적 요구사항 식별』에서 설명)에 따라 설치 요구사항이 결정됩니다. 이 절에서는 Linux용 Communications Server 설치에 필요한 컴퓨터 자원을 개괄적으로 설명합니다. 자세한 내용은 각 제품과 함께 제공되는 문서를 참조하십시오.

하드웨어

Linux용 Communications Server는 아래 Linux 분배 방법 중 하나에 의해 지원되는 컴퓨터를 요구합니다.

uname -m 명령을 사용하여 목표 컴퓨터의 CPU 클래스를 확인하십시오. 다음 테이블은 각 서버 유형에 적합한 하드웨어 및 **uname -m**에 대한 해당 하드웨어의 응답을 표시합니다.

서버 유형	하드웨어	uname 응답
System z	31비트 S/390® 또는 zSeries®	s390
System z	64비트 zSeries 또는 System z9™	s390x

Linux 운영 체제

Linux용 Communications Server는 다음과 같은 Linux 계열을 지원합니다. 특정 버전 번호와 각 계열에서 지원하는 커널 빌드에 대한 최신 정보는 Linux용 Communications Server 설치 이미지에 있는 **README** 파일을 참조하십시오.

- S/390용 RedHat Enterprise Linux 3(RHEL3-s390)
- zSeries용 RedHat Enterprise Linux 3(RHEL3-s390x)
- S/390용 RedHat Enterprise Linux 4(RHEL4-s390)
- zSeries용 RedHat Enterprise Linux 4(RHEL4-s390x)
- IBM 메인프레임용 SUSE Linux Enterprise Server 8(SLES8-s390*)
- IBM 메인프레임용 SUSE Linux Enterprise Server 9(SLES9-s390*)

- IBM 메인프레임용 SUSE Linux Enterprise Server 10(SLES10-s390x)

하드웨어 링크

링크 하드웨어는 클라이언트가 아닌 서버에만 필요합니다.

Linux용 Communications Server는 Enterprise Extender를 사용하는 TCP/IP 인터페이스나, System z용 Linux MPC(MultiPath Channel) 장치 드라이버를 사용하는 가상 MPC 인터페이스(System z의 Linux용 Communications Server 전용), Linux **ics** 장치 드라이버를 사용하는 802.2 이더넷 또는 토큰링 OSA 연결(OSE **chpid** 모드에서 OSA2 필요), 또는 layer2/VSwitch를 지원하는 Linux **qeth** 장치 드라이버를 사용하는 802.2 이더넷 OSA 연결에서 사용할 수 있습니다.

추가 소프트웨어: Linux

Linux용 Communications Server는 다음과 같은 추가 소프트웨어를 필요로 합니다. 특정 버전 요구사항의 자세한 내용(Linux 계열에 따름)과 이 소프트웨어 패키지의 설치 지시사항에 대해서는 Linux용 Communications Server 설치 이미지의 **README** 파일을 참조하십시오.

- Linux Streams(LiS).
- OpenMotif(클라이언트가 아닌 서버에만 필요). Linux용 Communications Server를 구성 및 관리하기 위해 권장되는 방법인, Motif 관리 프로그램을 사용하려면 이 소프트웨어가 필요합니다.
- Java(Java CPI-C를 사용할 경우 필요). JRE(Java Runtime Environment)가 필요합니다. Java CPI-C 응용프로그램에 사용하도록 새 Java 클래스를 컴파일해야 하는 경우 Java SDK도 필요합니다.
- System z용 Linux MultiPath Channel 장치 드라이버(System z의 Linux용 Communications Server 전용. 서버에서만 필요하고 클라이언트에서는 필요하지 않습니다. MPC 연결을 사용하여 VM/VTAM 시스템에 연결 중인 경우에 이 구성요소가 필요합니다).

WebSphere Application Server(HTTPS 액세스용)

원격 API 클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결된 클라이언트/서버 시스템을 실행하는 경우 이들 클라이언트에서 서버에 HTTPS를 사용하여 액세스할 수 있도록 WebSphere Application Server를 실행해야 합니다.

Linux용 Communications Server는 WebSphere가 지원하는 모든 운영 체제를 실행하는 컴퓨터에 설치될 수 있는 WebSphere Application Server 버전 5와 함께 운영됩니다. 필요한 경우 Linux용 Communications Server가 설치된 Linux 컴퓨터에 설치될 수 있습니다. 설치에 대한 자세한 내용은 WebSphere Application Server 문서를 참조하십시오. 32 페이지의 『WebSphere Application

Server 구성』에서 설명한 것처럼, Linux용 Communications Server와 함께 WebSphere를 사용하려면 이 컴퓨터에 Linux용 Communications Server 플러그인도 추가로 설치해야 합니다.

메모리 및 기억영역

다양한 구성 및 서비스를 지원하려면 Linux용 Communications Server가 실행되는 워크스테이션에 Linux 분배에 필요한 최소 메모리 + 32MB의 메모리와 120MB의 디스크 공간이 필요합니다. 또한 설치 중에 250MB의 임시 기억영역이 필요합니다.

소프트카피(PDF) 양식으로 Linux용 Communications Server 관련 문서를 설치할 경우에는 추가 하드 디스크 기억영역이 필요합니다. 소프트카피 책을 모두 설치하려면 75MB의 디스크 공간이 필요합니다.

주: 기타 라이선스가 있는 프로그램, 사용자 응용프로그램 및 데이터를 위한 메모리 및 하드 디스크 요구사항은 본 요구사항에 포함되지 않습니다. 모든 시스템, 메모리 및 하드 디스크 요구사항은 해당 IBM 담당자나 허가된 판매업자와 함께 상세히 검토하십시오.

명명 규칙

네트워크 ID를 사용하여 물리적 네트워크를 논리적으로 세그먼트화할 수 있습니다. 또한, 다른 네트워크에 연결하려 할 때, 네트워크 이름이 충돌하지 않도록 반드시 네트워크 ID를 등록하는 것이 좋습니다.

네트워크 및 LU 이름을 다음과 같이 정의할 수 있습니다.

네트워크 이름

APPN 네트워크의 단편화를 제공하기 위해 다른 네트워크 이름(네트워크 ID)을 정의할 수 있습니다. 단편화는 네트워크 토폴로지 데이터베이스의 크기 및 각 네트워크를 통한 브로드캐스트 LOCATE 요청의 주기를 제한합니다.

네트워크 ID의 고유성을 보장하기 위해 네트워크 관리자는 네트워크 ID를 세계적인 IBM 레지스트리에 등록할 수 있습니다. IBM 레지스트리는 각 네트워크 ID가 등록된 ID 중에서 고유하다는 것을 보장합니다. 레지스트리 표준은 국제 표준 협회(ISO)에서 확립된 Open Systems Interconnection(OSI) 표준과 일치하며 OSI 국가 코드를 포함합니다. 등록에 관한 자세한 정보는 *User's Guide for SNA Network Registry*를 참조하십시오.

LU 이름

시스템 정의 및 네트워크 검색을 최소화시키기 위해 LU 이름에 대한 와일드카드를 사용할 수 있습니다.

제 3 장 Linux 서버에 Linux용 Communications Server 설치

이 장은 Linux용 Communications Server를 Linux 서버에 설치하는 데 필요한 단계에 대한 일반 정보를 제공합니다. 설치 프로세스의 자세한 단계별 지시사항은 해당 Linux 계열과 관련된 상세한 최신 정보를 제공하는 Linux용 Communications Server 설치 이미지의 **README** 파일을 참조하십시오. **README** 파일에는 다음 사항이 포함되어 있습니다.

- 필요한 추가 소프트웨어 패키지의 정확한 패키지 이름 및/또는 버전 번호
- 설치 및 설정 명령의 세부사항.

Linux용 Communications Server 라이선스가 있는 프로그램의 패키지 방법

Linux용 Communications Server 라이선스가 있는 프로그램은 다음 항목이 포함된 3개의 CD 이미지로 제공됩니다.

CD #1: 빠른 시작

이 CD에는 Linux용 Communications Server PDF 문서의 전체 세트가 포함되어 있습니다(/DOCS 디렉토리).

CD #2: 서버

이 CD에는 서버 설치에 필요한 다음과 같은 파일들이 들어 있습니다.

- Linux용 Communications Server 문서 발행 이후의 제품 변경에 대한 정보가 포함된 **README** 파일
- 설치 스크립트
- 서버 및 PDF 매뉴얼에 대한 설치 이미지

CD #3: 클라이언트

이 CD에는 다음과 같은 유형의 클라이언트 각각을 설치하는 데 필요한 **README** 파일, 설치 스크립트 및 설치 이미지가 들어 있습니다.

- 32비트 Intel(i686)용 Linux 클라이언트
- 64비트 AMD64/Intel EM64T(x86_64)용 Linux 클라이언트
- pSeries(ppc64)용 Linux 클라이언트
- System z9 또는 System z(s390/s390x)용 Linux 클라이언트
- AIX 클라이언트
- 32비트 Windows 클라이언트
- x64 Windows 클라이언트

Linux용 Communications Server 라이선스가 있는 프로그램의 패키지 방법

CD 이미지에 포함된 파일의 모든 세부사항에 대해서는 **README** 파일을 참조하십시오.

주: 기억영역 요구사항은 22 페이지의 『설치 요구사항』을 참고하십시오.

Linux용 Communications Server 설치 준비

전제조건 소프트웨어 설치

Linux용 Communications Server를 설치하기 전에 22 페이지의 『설치 요구사항』에 나열된 전제조건 소프트웨어를 설치하십시오.

- LiS 스트림
- OpenMotif
- Java
- System z용 Linux MultiPath Channel 장치 드라이버(System z의 Linux용 Communications Server 전용. VM/VTAM 시스템에 연결하기 위해 MPC 연결을 사용하려는 경우에만 필요)

필수 소프트웨어 패키지의 세부사항과 단계별 설치 지시사항은 Linux용 Communications Server 설치 이미지의 **README** 파일을 참조하십시오.

제품 설치 세부사항 표시

Linux용 Communications Server 및 이미 설치된 관련 소프트웨어 패키지에 대한 정보를 표시할 수 있습니다. 설치된 모든 패키지를 나열하려면 다음 명령을 사용하십시오.

```
rpm -q -a
```

특정 패키지의 세부사항을 보려면 다음 명령을 사용하십시오.

```
rpm -q -i packagename
```

여기서 *packagename*은 설치한 패키지의 기본 이름입니다(예: **ibm-commserver**).

언어 환경 변수 변경

Linux용 Communications Server를 사용할 경우 사용할 언어를 표시하도록 LANG 변수를 올바르게 설정하십시오.

다음 명령을 사용하여 LANG 변수를 변경하십시오.

```
export LANG=language
```

여기서 *language*를 사용할 언어의 식별자로 대체하십시오. 다음 중 하나가 될 수 있습니다.

식별자	언어
en_US	영어(미국)
ja_JP	일본어(PC)
de_DE	독일어
es_ES	스페인어
fr_FR	프랑스어
ko_KR	한국어
pt_BR	포르투갈어
zh_CN	중국어(단순화된 EUC)
zh_TW	대만어

이전 레벨의 Linux용 Communications Server에서 이주

이전 버전의 Linux용 Communications Server가 설치된 상태에서 버전 6.2.2로 이주하는 경우 다음 단계를 수행하십시오.

사용자 정의된 구성 파일의 저장

아래에 나열된 파일이 존재하는 경우 이 파일을 임시 디렉토리에 저장하십시오. 이 파일 모두가 모든 설치 과정마다 존재하는 것은 아닙니다.

```

/etc/opt/ibm/sna/sna_node.cfg
/etc/opt/ibm/sna/sna_domn.cfg
/etc/opt/sna/sna.net
/etc/opt/ibm/sna/sna_tps
/etc/opt/ibm/sna/ibmcs.kdb
/etc/opt/ibm/sna/ibmcs.sth
/etc/opt/ibm/sna/ibmcs.rdb
/etc/opt/ibm/sna/ibmcs.crl

```

또한 시스템 시작 시에 Linux용 Communications Server가 자동으로 시작되지 않도록 78 페이지의 『Linux용 Communications Server 사용 가능화』에 설명된 것처럼 `/etc/rc.d/init.d/snastart` 시작 파일을 사용자 정의하여 `sna start` 명령을 제거한 경우 해당 파일에 대한 변경사항을 적어 두십시오.

이전 릴리스 설치 제거

다음 명령을 사용하여 Linux용 Communications Server를 중지하고 설치 제거하십시오. 설치된 이전 버전 및 설치 방법에 따라, 나열된 RPM 패키지 중 일부는 해당 시스템에 존재하지 않을 수도 있습니다.

```

sna stop
rpm -e ibm-conmsserver-ptf
rpm -e ibm-commserver-docs
rpm -e ibm-commserver-ecl
rpm -e ibm-commserver
rpm -e gsk6bas

```

LiS 설치 제거

다음 명령을 사용하여 현재 레벨의 LiS 오픈 소스 패키지를 설치 제거하십시오.

```
PATH=$PATH:/sbin
unset LD_PRELOAD
rmmod streams
cd /usr/src/LiS
make uninstall
make very-clean
cd /usr/src
rm -rf LiS*
```

PATH 및 기타 환경 변수 변경

이전 버전의 Linux용 Communications Server에 대한 다음 환경 변수를 수정한 경우, Linux용 Communications Server 버전 6.2.2 프로그램 제품의 경우 경로가 달라질 수 있으므로 변경사항을 제거해야 합니다.

```
PATH
LD_LIBRARY_PATH
LD_RUN_PATH
LD_PRELOAD
CLASSPATH
```

env 명령을 사용하여 **sna**에 대한 참조로 사용된 모든 환경 변수를 확인하는 것이 좋습니다.

```
env | grep sna
```

기타 패키지

Java 설치를 확인하고 필요에 따라 최신 레벨로 갱신하는 것이 좋습니다.

최종 정리

다음 명령을 사용하면 이전 레벨의 제품에 남아 있는 항목이 제거됩니다.

```
rm -rf /etc/opt/ibm/sna /var/opt/ibm/sna /opt/ibm/sna
```

새 레벨의 Linux용 Communications Server 버전 6.2.2 프로그램 제품 설치

본 매뉴얼 및 **README** 파일의 지시사항을 준수하여 제품을 설치하십시오.

저장된 구성의 복원

본 프로세스의 첫 단계에서 구성 파일을 저장한 경우 지금이 파일을 복원하는 시점입니다. 먼저 다음 명령을 사용하여 Linux용 Communications Server 소프트웨어를 중지하십시오.

```
/opt/ibm/sna/bin/sna stop
```

본 프로세스의 첫 단계에서 **ibmcs.*** 파일 중 하나라도 저장한 경우 지금 **/etc/opt/ibm/sna** 디렉토리에서 모든 **ibmcs.*** 파일을 제거하십시오. 예를 들어 **ibmcs.kdb** 및 **ibmcs.sth**를 저장한 경우, **ibmcs.crl** 및 **ibmcs.rdb**를 대체할 파일을 저장하지 않았다고 해도 이들 파일을 제거해야 합니다. 저장된 파일과 새 파일을 같이 실행하지 않아야 합니다.

저장한 파일을 **/etc/opt/ibm/sna** 디렉토리로 복원하십시오.

또한 **/etc/rc.d/init.d/snastart** 시작 파일의 변경사항을 저장한 경우, 시스템 시작 시에 Linux용 Communications Server가 자동으로 시작되지 않도록 78 페이지의 『Linux용 Communications Server 사용 가능화』에 설명된 것처럼 파일의 새 사본을 변경하십시오.

이제 다음 명령을 사용하여 Linux용 Communications Server 소프트웨어를 다시 시작하십시오.

```
/opt/ibm/sna/bin/sna start
```

Linux용 Communications Server 라이선스가 있는 프로그램 설치

Linux용 Communications Server 설치

Linux용 Communications Server의 이전 레벨이 이미 설치되어 있는 경우, 이 새로운 레벨을 설치하기 전에 27 페이지의 『이전 레벨의 Linux용 Communications Server에서 이주』에 있는 단계를 따라 이전 레벨을 제거하십시오.

Linux용 Communications Server를 설치하려면 다음 단계를 수행하십시오.

1. **ibm-commserver-6.2.2.0-s390.tgz** 파일을 CD-ROM에서 Linux System z 시스템으로 복사하거나 FTP로 전송하십시오. 파일을 복사하거나 FTP에서 다운로드하는 데 2진 모드를 사용하십시오.
2. 루트로 Linux System z 시스템에 로그인하십시오.
3. tar 파일의 압축을 풀어 빈 임시 디렉토리에 위치시키십시오.

```
mkdir /tmp/ibmcs
```

```
cd /tmp/ibmcs
```

```
zcat ibm-commserver-6.2.2.0-s390.tgz | tar -xvf -
```

4. 다음과 같이 **installibmcs** 셸 스크립트를 실행하십시오.

```
./installibmcs
```

Linux용 Communications Server 라이선스가 있는 프로그램 설치

이 셸 스크립트는 특정 전제조건을 검사하며 조건에 충족되지 못할 경우에는 경고 메시지를 발행합니다. 다음 사항도 표시합니다. 프롬프트에 응답하면, 셸 스크립트가 **rpm** 패키지를 설치합니다.

- Linux용 Communications Server 라이선스 조항을 읽고 승인했는지 확인
- Linux용 Communications Server 도메인에서 마스터 서버가 될 서버의 이름. 독립형 노드로 Linux용 Communications Server를 실행할 경우 이 매개변수를 지정하지 마십시오. 이 경우 노드는 클라이언트/서버 기능을 지원하지 않습니다.

아래에 설명된 대로 **installibmcs** 명령에 추가 매개변수를 지정하면 이 프롬프트를 대체할 수 있습니다.

셸 스크립트에 오류가 발생해 Linux용 Communications Server를 설치할 수 없으면 오류 메시지가 표준 출력(일반적으로 화면)에 기록됩니다. 이와 같은 오류를 해결하는 데 도움을 받으려면 Linux용 Communications Server 설치 이미지의 **README** 파일을 참조하십시오.

메모리가 제한된 시스템에서는 Linux용 Communications Server를 설치한 후 다시 시동해야 SNA 노드를 시작할 수 있습니다. 대형 시스템의 경우에는 이것이 필요하지 않을 수도 있습니다. Linux용 Communications Server 노드가 시작되지 않으면, **/var/log/messages** 파일에 다음과 같은 항목이 있는지 확인하십시오.

kernel: SNA Trace Driver can only getX blocks of memory — please reboot

다시 시동해도 이 메시지가 계속 표시되면, 메모리가 더 필요합니다.

5. Linux용 Communications Server 2진 디렉토리를 PATH에 추가하십시오. 이를 자동 수행하도록 프로파일을 변경할 수도 있습니다.

```
export PATH="$PATH:/opt/ibm/sna/bin:/opt/ibm/sna/bin/X11"
```

```
export LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib
```

```
export LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib
```

Java CPI-C 응용프로그램의 경우 다음 환경 변수도 설정해야 합니다.

```
export CLASSPATH=$CLASSPATH:/opt/ibm/sna/java/cpic.jar
```

일부 응용프로그램의 경우 LD_PRELOAD 환경 변수를 설정할 수도 있지만 해당 프로파일에서 이 변수를 전체 바꾸기로 수행하지 않아야 합니다.

```
export LD_PRELOAD=/usr/lib/libpLiS.so
```

6. Linux용 Communications Server를 시작하십시오. 설치 후에 시스템이 다시 시동될 때는 자동으로 시작된다는 것을 알아두십시오.


```
cd /
```

```
sna start
```

7. Linux용 Communications Server Motif 관리 프로그램을 실행하십시오. Linux용 Communications Server 운영에 익숙해질 때까지 이 프로그램을 사용할 것을 강력하게 권장합니다.

Linux System z 시스템은 XWindows 클라이언트 기능만을 포함하므로 원격 XWindows 서버를 사용해야 합니다. XWindows 서버에서 다음 명령을 사용하십시오.

```
xhost +XXXX
```

XXXX는 Linux System z 시스템의 TCP/IP 이름 또는 주소입니다.

xsnaadmin 클라이언트에게 서버가 있는 위치와 다시 시작하라고 알립니다.

```
export DISPLAY=YYYY:Z
```

```
xsnaadmin &
```

YYYY는 XWindows 서버의 TCP/IP 이름이나 주소이고, Z는 가상 디스플레이 번호(일반적으로 0)입니다.

자동 설치가 필요한 경우 **installibmcs** 명령에 매개변수를 추가하여 Linux용 Communications Server 라이선스 조항 승인을 확인하고 마스터 서버 이름을 지정할 수 있습니다. 이 경우 셸 스크립트가 추가 정보에 대한 프롬프트를 표시하지 않고 실행됩니다. 다음 명령을 사용하십시오.

```
./installibmcs license_accepted [ master_name ]
```

여기서 *master_name*은 마스터 서버의 이름입니다. 독립형 노드로 Linux용 Communications Server를 실행할 경우 이 매개변수를 지정하지 마십시오. 이 경우 노드는 클라이언트/서버 기능을 지원하지 않습니다.

Linux용 Communications Server 온라인 문서

29 페이지의 『Linux용 Communications Server 설치』에 있는 단계를 따라서 **tgz** 파일의 압축을 푼 후에 **installibmcsdocs** 셸 스크립트를 실행하십시오.

```
./installibmcsdocs
```

Host Access Class 라이브러리(HACL)

HACL 파일은 Linux용 Communications Server를 설치하면 자동으로 설치됩니다. 라이브러리 코드는 rpm의 `ibm-commserver-ecl` 패키지에 있습니다. README를 비롯하여 이들 파일은 `/opt/ibm/sna/ecl`에 있거나, 다음 명령을 실행하여 찾을 수 있습니다.

```
rpm -ql ibm-commserver-ecl
```

WebSphere Application Server 구성

원격 API 클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결된 클라이언트/서버 시스템을 실행하는 경우 22 페이지의 『설치 요구사항』에 설명된 것처럼 이들 클라이언트에서 서버에 HTTPS를 사용하여 액세스할 수 있도록 WebSphere Application Server를 실행하는 컴퓨터가 필요합니다.

이 절에서는 Linux용 Communications Server와 함께 사용하도록 WebSphere를 설정하는 방법을 설명합니다.

- 클라이언트에 제시할 WebSphere 서버의 보안 인증서 설정
- Linux용 Communications Server와 함께 작동하도록 WebSphere Application Server 구성
- WebSphere 서버에 서버 구성 파일 설치

WebSphere Application Server에 액세스하려면 각각의 원격 API 클라이언트에 클라이언트 보안 인증서 및 클라이언트 네트워크 데이터 파일도 설정해야 합니다. 자세한 내용은 해당 클라이언트 유형 설치와 관련된 장을 참조하십시오.

WebSphere Application Server의 보안 인증서 설정

서버에서 보안 인증서를 설정하는 데 대한 지시사항은 WebSphere Application Server 문서를 참조하십시오. 이 인증서는 서버가 HTTPS를 사용하여 연결을 시도할 때 인증 프로세스 과정에서 원격 API 클라이언트에 제시할 인증서입니다.

클라이언트 인증을 강제하도록 WebSphere를 구성하는 것이 좋습니다. 자세한 내용은 WebSphere Application Server 문서를 참조하십시오. 이는 WebSphere가 인증 프로세스 과정에서 원격 API 클라이언트의 보안 인증서를 요청하여, 클라이언트 인증서의 신뢰성을 검증할 수 있는 경우에만 원격 API 클라이언트에서 들어오는 연결을 승인할 것임을 의미합니다.

WebSphere Application Server 구성

Linux용 Communications Server와 함께 작동하도록 WebSphere Application Server를 구성하려면 다음 단계를 수행하십시오. 자세한 내용은 WebSphere Application Server 문서를 참조하십시오.

1. 2개의 파일 **snahttpsrv.ear** 및 **snahttpsrv.cfg**를 원격 API 클라이언트 설치 CD의 **ibm-commserver-https** 디렉토리에서 WebSphere 관리 콘솔이 실행되는 컴퓨터의 디렉토리 또는 이 컴퓨터에서 액세스 가능한 네트워크 디렉토리로 복사하거나 FTP로 전송하십시오.

관리 콘솔이 Windows에서 실행 중인 경우 CD에서 직접 파일에 액세스할 수 있으므로 파일을 복사할 필요가 없습니다. 단지 원격 API 클라이언트 설치 CD를 Windows 컴퓨터 CD 드라이브에 넣으면 됩니다.

2. WebSphere 관리 콘솔을 시작하십시오.
3. WebSphere 문서에 따라 SSL 보안 연결을 통해서만 액세스 가능한 가상 호스트를 작성하십시오. 이 가상 호스트는 SNA HTTPS 연결을 관리하는 Java 플러그인을 위해 사용됩니다.
4. 메뉴 표시줄에서 응용프로그램, 새 응용프로그램 설치를 선택하십시오.
5. **snahttpsrv.ear** 파일의 위치를 지정하십시오. 다음 버튼을 선택하십시오.
6. 처음 두 화면에서 가상 호스트 이름을 지정하라는 프롬프트가 표시되면 HTTPS용으로 설정한 가상 호스트의 이름을 입력하십시오. 특정 WebSphere 구성을 사용할 필요가 없는 경우, 다른 모든 매개변수에 대해서는 기본 옵션을 승인할 수 있습니다. 다음 버튼이 완료 버튼으로 바뀔 때까지 다음 버튼을 선택하고, 완료 버튼이 나오면 완료 버튼을 선택하십시오. 그러면 성공적으로 설치된 응용프로그램이 화면에 나타납니다.
7. 마스터 구성에 저장을 누른 다음 저장 버튼을 누르십시오.
8. 메뉴 표시줄에서 응용프로그램, 엔터프라이즈 응용프로그램을 선택하십시오.
9. 응용프로그램 목록에서 **SnaHttpTransport**를 찾아 바로 옆의 선택란을 누른 다음 시작 버튼을 눌러 응용프로그램을 시작하십시오. (이렇게 하면 WebSphere Application Server를 시작할 때 응용프로그램이 자동으로 자동으로 시작됩니다.)
10. 메뉴 표시줄에서 환경, 웹 서버 플러그인 갱신을 선택하고 확인 버튼을 누르십시오. 이렇게 하면 WebSphere 구성이 갱신됩니다.

서버 구성 파일 설치

WebSphere Application Server가 Linux용 Communications Server와 함께 작동하려면 HTTPS를 사용하여 액세스할 수 있는 Linux용 Communications Server 목록이 필요합니다. 다음 단계에 따라 이 목록을 작성 및 설치하십시오.

1. WebSphere 관리 콘솔 메뉴 표시줄에서 환경, WebSphere 변수 관리를 선택하십시오.
2. 이 목록에서 **USER_INSTALL_ROOT** 변수를 찾아 해당 값을 기록해 두십시오. 이 값은 WebSphere 서버의 디렉토리 경로입니다. 환경 변수 목록은 두 페이지가 넘을 수 있으므로 다음 버튼을 사용하여 목록을 스크롤해야 할 수도 있습니다.

3. **snahttpsrv.cfg** 파일을 32 페이지의 『WebSphere Application Server 구성』에 저장한 위치(또는 설치 CD)에서 **USER_INSTALL_ROOT** 변수가 지정하는 디렉토리로 복사한 다음 HTTPS를 사용하여 원격 API 클라이언트가 액세스할 수 있는 Linux용 Communications Server 목록을 포함하도록 텍스트 편집기를 사용하여 이 파일을 편집하십시오. 각 서버는 파일에서 별도의 행에 다음 형식으로 지정되어야 합니다.

```
server=servername.domainname.com
```

설치 후 절차

이 절에서는 Linux용 Communications Server 설치 후에 요구되는 유지보수 TASK 수행 방법에 대해 설명합니다.

클라이언트/서버 운영

Linux용 Communications Server는 설치된 후 처음에는 독립형 서버(모든 구성요소가 한 Linux 시스템에 존재)로 작동됩니다. 이 Communications Server를 클라이언트/서버 도메인의 서버로 실행하기 위한 지시사항에 대해서는 *Communications Server for Linux Administration Guide*의 Managing Communications Server for Linux Client-Server Systems 장을 참조하십시오.

설치 후 정리

설치가 완료된 후에는 설치 프로세스 동안 작성했던 **tgz** 파일 및 임시 디렉토리를 지울 수 있습니다.

PDF 서적 보기

이 제품의 설치 매체에 들어 있는 매뉴얼은 PDF(Portable Document Format) 형식입니다. 소프트웨어 형식이 있으면 관련 정보로의 하이퍼텍스트 링크를 사용하여 정보를 보다 쉽게 검색하고 찾아보거나 인쇄할 수 있습니다. 또한 PDF 표시기는 많은 다른 플랫폼에서 사용 가능하기 때문에 사용자 사이트에서 라이브러리를 더 쉽게 공유할 수 있습니다.

제품 설치 시 PDF 매뉴얼을 설치하도록 선택하면 이 매뉴얼은 **/opt/ibm/sna/docs** 디렉토리에 설치됩니다. 매뉴얼은 Linux용 Communications Server 설치 매체의 **/DOCS** 디렉토리와 **tgz** 파일에도 포함되어 있습니다.

PDF 표시기를 사용하여 PDF 매뉴얼을 읽을 수 있습니다(예: Windows의 Adobe Acrobat 또는 Intel Linux의 **xpdf**).

현재 릴리스 정보 검토

`/opt/ibm/sna` 디렉토리에 포함된, 제품에 대한 **README** 파일의 최신 갱신 버전에는 Linux 라이브러리 발행 이후의 모든 제품 변경에 대한 정보가 포함됩니다. 이 파일은 Linux용 Communications Server 설치 매체의 루트 디렉토리 및 **tgz** 파일에도 들어 있습니다. 제품 갱신 버전을 받을 때마다 **README** 파일을 검토하십시오.

TN 서버 또는 TN 방향 재지정자에서 사용하기 위해 SSL 구성

SSL 기능이 있는 TN 서버 또는 TN 방향 재지정자를 사용하려는 경우, Linux용 Communications Server를 설치한 후 SSL 소프트웨어를 구성해야 합니다.

SSL 소프트웨어에는 다음과 같은 두 가지 구성요소가 필요합니다.

- 데이터 암호화 및 암호 해제를 위한 키 쌍
- 서버 인증을 위한 인증서

인증서와 키 쌍은 키링(keyring) 데이터베이스에서 단일 레코드를 구성하며, 이는 TN 서버 또는 TN 방향 재지정자가 실행되는 Linux용 Communications Server에 저장됩니다. Linux용 Communications Server는 데이터베이스를 사용하여 SSL을 구현합니다.

키 링 데이터베이스를 관리하려면, OpenMotif를 설치해야 합니다. OpenMotif 설치에 대한 지시사항은 Linux용 Communications Server 설치 이미지의 **README** 파일을 참조하십시오.

키링 데이터베이스를 관리하려면 Linux 명령 프롬프트에 다음 명령을 입력하십시오.

snakeyman

snakeyman 명령은 Java 프로그램을 실행합니다. 자세한 내용은 이 프로그램과 함께 제공된 도움말을 참조하십시오.

데이터베이스 내의 각 레코드는 레이블이라고 하는 고유 이름에 의해 식별됩니다. 다른 TN 서버 또는 TN 방향 재지정자에서 두 개 이상의 레코드를 사용하고 있다면, 데이터베이스를 설정할 때 레이블의 설명을 작성해야 합니다. 이들 레이블은 각 세션에서 사용될 레코드를 식별할 때 사용됩니다. 다른 레코드의 레이블을 명시적으로 지정하지 않고 세션이 이 레코드를 사용하도록, 레코드 중 하나를 디폴트로 나타낼 수 있습니다.

snakeyman을 사용하여 서버 인증서를 갱신한 다음 **snakeyman** 프로그램을 종료하고 Linux용 Communications Server 노드를 중지했다가 다시 시작해야 갱신된 인증서를 사용할 수 있습니다. 다음 명령을 사용하여 노드를 중지했다가 다시 시작하십시오.

snaadmin term_node

snaadmin init_node

Linux용 Communications Server 구성 파일 백업

Linux용 Communications Server는 노드, 도메인 및 TP 구성 파일에 영향을 미치는 변경을 수행할 때마다 자동으로 이들 파일을 백업합니다(Linux용 Communications Server 관리 도구 사용). 예를 들어 노드 구성 파일(`sna_node.cfg`)에 영향을 주는 변경을 수행하면 Linux용 Communications Server는 `sna_node.bk n` 이라는 백업 파일을 작성하는데, 여기서 n 은 1 또는 2입니다.

- 파일을 처음으로 변경하면 기존 구성은 `sna_node.bk1`에 저장됩니다.
- 파일을 두 번째 변경할 경우 기존 구성은 `sna_node.bk2`에 저장되고, `sna_node.bk1`은 변경되지 않습니다.
- 파일을 세 번째 변경하고 또 계속 변경하면, `sna_node.bk1`은 제거되고 `sna_node.bk2`가 `sna_node.bk1`로 이름이 바뀌며, 기존 구성은 `sna_node.bk2`에 저장됩니다.

이 프로세스는 노드 구성 파일에 대해 언제든지 최대 두 개의 백업 파일이 존재하는 것을 의미합니다. 다른 백업 파일에 대한 파일 이름 확장자를 생성할 때도 같은 프로세스가 사용됩니다.

자동 백업외에, 다음 상황에서의 데이터 유실에 대비하여 구성 파일을 백업해야 합니다.

- Linux 운영 체제의 새 레벨을 설치하기 전
- Linux용 Communications Server의 새 릴리스를 설치하기 전
- 새 구성을 작성한 후

다음 명령을 사용하여 구성 파일을 백업할 수 있습니다.

```
cd /etc/opt/ibm/sna
tar cvf Devicename sna_node.cfg sna.net sna_tps sna_domn.cfg
ibmcs.*
```

Linux용 Communications Server 구성 파일의 백업 사본 복원

『Linux용 Communications Server 구성 파일 백업』에 설명된 대로 백업된 Linux용 Communications Server 구성 파일을 복원하려면 다음을 수행하십시오.

1. Linux용 Communications Server가 활동 중이 아닌지 확인하십시오. 이를 판별하려면, 다음 명령을 입력하십시오.

```
snaadmin status_node
```

Linux용 Communications Server가 활동 중인 경우 명령을 실행하면 로컬 노드의 상태에 관한 정보가 표시됩니다. 그렇지 않은 경우에는 Linux용 Communications Server가 활동 중이 아님을 나타내는 메시지가 표시됩니다.

Linux용 Communications Server가 활동 중이면 다음 명령을 입력하여 비활성화 시키십시오.

```
sna stop
```

2. 다음 명령을 입력하십시오.

```
cd /etc/opt/ibm/sna  
tar xvf Devicename
```

이 명령에서 *Devicename*은 파일 백업 시에 사용하는 장치의 경로 및 파일 이름입니다.

이 명령은 모든 기존 구성 파일을 **/etc/opt/ibm/sna** 디렉토리에 있는 것과 같은 이름으로 겹쳐씁니다.

구성 파일 재초기화

Linux용 Communications Server 구성 파일이 실수로 수정되어 이 파일의 정보를 더 이상 사용할 수 없는 경우 새로 설치된 것처럼 Linux용 Communications Server를 재구성할 수 있도록 파일을 재초기화해야 합니다. 구성 정보를 회복시킬 수 없다고 확신하는 때에만 이를 수행하여야 합니다.

주: 유효한 백업 구성 파일이 있으면 해당 파일을 **/etc/opt/ibm/sna** 디렉토리에 복사하고 **sna start** 명령을 사용하여 노드를 초기화할 때 이 파일을 사용할 수 있습니다.

다음 구성 파일들을 재초기화할 수 있습니다.

- 노드 구성 파일 **sna_node.cfg**
- 도메인 구성 파일 **sna_domn.cfg**
- TP 구성 파일 **sna_tps**
- SSL 키링 데이터베이스 파일 및 암호 스택시(stash) 파일

다음 단계를 수행하여 구성 파일들을 재초기화하십시오.

1. 활동 중이면 관리 프로그램을 종료하고 다음 명령을 실행하여 Linux용 Communications Server를 사용 불가능 상태로 만드십시오.

```
sna stop
```

2. 재초기화하고 있는 파일들을 서로 다른 위치에 복사하여 기존 구성 파일들을 백업하십시오.
3. 재초기화하고 있는 파일들을 삭제하십시오.

4. 도메인 구성 파일을 삭제했다면 Linux용 Communications Server에서 제공한 빈 도메인 구성 파일로부터 복사를 수행하는 다음 명령을 실행하여 파일을 다시 작성하십시오.

```
cp -p /opt/ibm/sna/samples/empty.cfg /etc/opt/ibm/sna/sna_domn.cfg
```

이 명령에 의해 Linux용 Communications Server를 시작하는 데 필요한 새 도메인 구성 파일이 작성됩니다.

5. SSL 키링 데이터베이스 파일을 삭제했다면 Linux용 Communications Server에서 제공한 샘플 파일로부터 복사를 수행하는 다음 명령을 실행하여 파일을 다시 작성하십시오.

```
cp -p /opt/ibm/sna/samples/ibmcs.* /etc/opt/ibm/sna
```

6. 다음 명령을 실행하여 Linux용 Communications Server를 다시 시작하십시오.

```
sna start
```

7. Motif 관리 프로그램을 시작하십시오.

```
xsnaadmin &
```

sna_node.cfg 파일이 존재하지 않으면 관리 프로그램은 노드를 구성할지 묻는 메시지를 표시합니다. 75 페이지의 제 8 장 『Linux용 Communications Server 구성 및 사용』 또는 *Communications Server for Linux Administration Guide*에 있는 대로 노드 및 기타 자원을 구성하여 계속할 수 있습니다.

유효한 **sna_node.cfg** 파일을 사용한다면 노드를 초기화하는데 새 구성 파일이 사용됩니다.

Linux용 Communications Server 설치 제거

Linux용 Communications Server 제품은 언제든지 설치 제거할 수 있습니다. 다음 프로시저를 사용하십시오.

1. 루트 권한으로 로그인하십시오.
2. Linux용 Communications Server가 활동 중이 아닌지 확인하십시오. 이를 판별하려면, 다음 명령을 입력하십시오.

```
snaadmin status_node
```

Linux용 Communications Server가 활동 중인 경우 명령을 실행하면 로컬 노드의 상태에 관한 정보가 표시됩니다. 그렇지 않은 경우에는 Linux용 Communications Server가 활동 중이 아님을 나타내는 메시지가 표시됩니다.

Linux용 Communications Server가 활동 중이면 다음 명령을 입력하여 비활성화 시키십시오.

```
sna stop
```

3. 다음 명령을 사용하여 Linux용 Communications Server 패키지 및 연관된 소프트웨어 패키지를 제거하십시오.

```
rpm -e ibm-commsserver-ptf
```

```
rpm -e ibm-commsserver-docs
```

```
rpm -e ibm-commsserver-ecl
```

```
rpm -e ibm-commsserver
```

```
/sbin/shutdown -r now
```

제 4 장 Linux에 IBM 원격 API 클라이언트 설치

이 장에서는 Linux에 IBM 원격 API 클라이언트를 설치하는 방법을 설명합니다. 이 소프트웨어를 사용하면 SNA 스택이 완전히 설치되지 않은 상태에서도 Linux 워크스테이션이 SNA 응용프로그램을 실행할 수 있습니다. Linux에 설치된 원격 API 클라이언트는 TCP/IP 네트워크를 사용하여 하나 이상의 Linux용 Communications Server(또는 CS/AIX 서버. 하지만 동시에 양쪽에 연결될 수는 없음)에 연결될 수 있습니다. CS Linux 서버는 CS/AIX 서버와 같은 도메인에서 작동될 수 없습니다.

이 장의 내용은 32비트 Intel(i686), 64비트 AMD64/Intel EM64T(x86_64) 및 pSeries(ppc64) 컴퓨터에서 실행되는 IBM 원격 API 클라이언트에 적용됩니다. IBM 원격 API 클라이언트를 System z 컴퓨터(s390/s390x)에 설치하는 경우에는 49 페이지의 제 5 장 『System z용 Linux에 IBM 원격 API 클라이언트 설치』를 참조하십시오.

IBM 원격 API 클라이언트 README 파일을 비롯한 설치 프로그램 및 연관된 파일은 설치 CD에 있는데, 다음과 같이 클라이언트 유형에 따라 디렉토리는 약간 다릅니다.

클라이언트 유형	CD의 디렉토리
32비트 Intel(i686)	<code>/ibm-commserver-clients/linux</code>
64비트 AMD64/Intel EM64T(x86_64)	<code>/ibm-commserver-clients/linux-x86_64</code>
pSeries(ppc64)	<code>/ibm-commserver-clients/linux—ppc64</code>

소프트웨어를 설치하기 전에 IBM 원격 API 클라이언트 README 파일을 읽는 것이 좋습니다.

하드웨어 및 소프트웨어 요구사항

하드웨어 요구사항

IBM 원격 API 클라이언트는 아래 Linux 분배 방법 중 하나에 의해 지원되는 컴퓨터를 요구합니다.

`uname -m` 명령을 사용하여 목표 컴퓨터의 CPU 클래스를 확인하십시오. 다음 테이블은 각 클라이언트 유형에 적합한 하드웨어 및 `uname -m`에 대한 해당 하드웨어의 응답을 표시합니다.

클라이언트 유형	하드웨어	uname 응답
32비트 Intel	Pentium® II 이상의 32비트 Intel 시스템 또는 Opteron 기반 시스템	i686
64비트 AMD64/Intel EM64T	x86_64(AMD64 또는 Intel EM64T) 시스템	x86_64
pSeries	pSeries POWER5™ 또는 OpenPower™ 시스템	ppc64

Linux 운영 체제 버전

IBM 원격 API 클라이언트의 현재 버전은 다음 Linux 운영 체제 버전과 함께 테스트되었습니다. 이 소프트웨어는 다른 Linux 분배에서도 만족스럽게 작동할 수 있습니다.

- RedHat Enterprise Linux 3(RHEL3): i686만 해당
- RedHat Enterprise Linux 4(RHEL4)
- SUSE Linux Enterprise Server 8(SLES8): i686만 해당
- SUSE Linux Enterprise Server 9(SLES9)
- SUSE Linux Enterprise Server 10(SLES10)

필요한 선택적 패키지에 대한 세부사항은 설치 CD의 **README** 파일을 참조하십시오.

Java

Java CPI-C API를 사용하는 경우 Java 소프트웨어가 필요합니다. 세부사항은 설치 CD의 **README** 파일을 참조하십시오.

GSKIT

클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결하는 경우 WebSphere 서버를 통해 서버에 HTTPS 프로토콜로 액세스하려면 GSKIT 소프트웨어가 필요합니다. GSKIT 소프트웨어는 설치 CD에 포함되어 있지만 이를 설치하려면 선택적인 일부 Linux 운영 체제 패키지가 필요합니다. 필요한 선택적 패키지에 대한 세부사항은 설치 CD의 **README** 파일을 참조하십시오.

클라이언트 설치 프로세스를 실행할 때 모든 전제조건 패키지가 설치되어 있는 경우(이 장의 뒷부분에서 설명) GSKIT 소프트웨어도 이 프로세스의 일부로 자동으로 설치됩니다. 그렇지 않은 경우에는 나중에 설치할 수 있습니다.

제품 설치 세부사항 표시

원격 API 클라이언트 및 이미 설치된 관련 소프트웨어 패키지에 대한 정보를 표시할 수 있습니다. 설치된 모든 패키지를 나열하려면 다음 명령을 사용하십시오.

```
rpm -q -a
```

특정 패키지의 세부사항을 보려면 다음 명령을 사용하십시오.

```
rpm -q -i packagename
```

여기서 *packagename*은 설치한 패키지의 기본 이름입니다(예: **ibm-commserver-client**).

언어 환경 변수 설정

다음 명령을 사용하여 사용할 언어를 나타내는 LANG 변수를 변경하십시오.

```
export LANG=language
```

여기서 *language*를 사용할 언어의 식별자로 대체하십시오. 다음 중 하나가 될 수 있습니다.

식별자	언어
en_US	영어(미국)
ja_JP	일본어(PC)
de_DE	독일어
es_ES	스페인어
fr_FR	프랑스어
ko_KR	한국어
pt_BR	포르투갈어
zh_CN	중국어(단순화된 EUC)
zh_TW	대만어

Linux에 원격 API 클라이언트 설치

전제조건 소프트웨어의 설치를 마쳤으면 IBM 원격 API 클라이언트를 설치할 준비가 된 것입니다.

IBM 원격 API 클라이언트의 이전 레벨이 이미 설치되어 있는 경우, 이 새로운 레벨을 설치하기 전에 46 페이지의 『Linux에서 원격 API 클라이언트 설치 제거』에 있는 단계를 따라 이전 레벨을 제거하십시오. 새로운 설치에서 사용할 수 있도록 구성 정보는 그대로 남아 있습니다.

1. 루트 권한으로 로그인하십시오.
2. CD를 마운트하여 현재 디렉토리로 만드십시오.

```
mount /dev/cdrom
```

```
cd /media/cdrom
```

RHEL3(32비트 Intel 클라이언트만 해당)의 경우 **/media/cdrom**을 **/mnt/cdrom**으로 바꾸십시오. 그렇게 하지 않으면 DVD 드라이브를 사용하는 경우 디렉토리 이름 **/media/cdrom**이 다를 수 있습니다. **df** 명령을 사용하여 Linux에서 CD가 마운트된 위치를 확인하십시오.

3. CD의 해당 서브디렉토리로 변경한 다음 셸 스크립트를 실행하여 클라이언트를 설치하십시오. 아래 예제는 32비트 Intel(i686) 클라이언트의 **/linux** 서브디렉토리를 보여줍니다. 필요한 경우 이를 **/linux-x86_64** 또는 **/linux-ppc64**로 대체하십시오.

```
cd ibm-commsserver-clients/linux  
./installibmcscli
```

이 셸 스크립트는 특정 전제조건을 검사하며 조건이 충족되지 않는 경우 경고 메시지를 발행합니다. 라이선스 계약을 읽고 동의하라는 메시지가 표시되면 스크립트가 RPM을 설치합니다. 해당 전제조건이 이미 설치된 경우 스크립트는 GSKIT 소프트웨어도 설치합니다.

4. IBM 원격 API 클라이언트 2진 디렉토리를 PATH에 추가하십시오. 이를 자동 수행하도록 프로파일을 변경할 수도 있습니다.

```
export PATH="$PATH:/opt/ibm/sna/bin"
```

```
export LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib
```

```
export LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib
```

Java CPI-C 응용프로그램의 경우 다음 환경 변수도 설정해야 합니다.

```
export CLASSPATH=$CLASSPATH:/opt/ibm/sna/java/cpic.jar
```

일부 응용프로그램의 경우 LD_PRELOAD 환경 변수를 설정할 수도 있지만 해당 프로파일에서 이 변수를 전체 바꾸기로 수행하지 않아야 합니다.

```
export LD_PRELOAD=/usr/lib/libpLiS.so
```

5. IBM 원격 API 클라이언트를 시작하십시오. 설치 후에는 시스템이 다시 시동될 때 자동으로 시작됩니다. 이 작업을 수행할 때는 아직도 CD 디렉토리에 있지 않은지 확인하십시오.

```
cd /  
sna start
```

주: IBM 원격 API 클라이언트가 HTTPS를 사용하여 서버에 연결하기 전에 클라이언트에서 GSKIT 키 관리자 프로그램을 사용하여 보안 인증서 구성을 설정해야 합니다. 자세한 내용은 45 페이지의 『GSKIT를 사용하여 HTTPS 보안 인증서 설정』을 참조하십시오.

또한 클라이언트 네트워크 데이터 파일을 갱신하여 클라이언트가 연결할 수 있도록 Linux용 Communications Server를 지정하고 HTTPS 지원을 제공하는 WebSphere 서버 이름을 지정해야 합니다. 자세한 정보는 *Communications Server for Linux Administration Guide*의 원격 API 클라이언트 관리 절을 참조하십시오.

GSKIT를 사용하여 HTTPS 보안 인증서 설정

클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결하는 경우 GSKIT 키 관리자 소프트웨어가 설치되어 있어야 합니다. 설치 CD의 **README** 파일에 설명된 것처럼 필요한 Linux 운영 체제 전체 조건이 설치된 경우 이 소프트웨어의 설치는 일반적으로 클라이언트 설치 과정의 일부로 이루어집니다. GSKIT가 클라이언트 설치 과정의 일부로 설치되지는 않았지만 지금 막 전체 조건을 설치한 경우 다음 단계에 따라 GSKIT 소프트웨어를 설치할 수 있습니다.

1. 루트 권한으로 로그인하십시오.
2. CD를 마운트하여 현재 디렉토리로 만드십시오.

```
mount /dev/cdrom
cd /media/cdrom
```

RHEL3(32비트 Intel 클라이언트만 해당)의 경우 **/media/cdrom**을 **/mnt/cdrom**으로 바꾸십시오. 그렇게 하지 않으면 DVD 드라이브를 사용하는 경우 디렉토리 이름 **/media/cdrom**이 다를 수 있습니다. **df** 명령을 사용하여 Linux에서 CD가 마운트된 위치를 확인하십시오.

3. CD의 해당 서브디렉토리로 변경한 다음 셸 스크립트를 실행하여 GSKIT 소프트웨어를 설치하십시오. 아래 예제는 32비트 Intel(i686) 클라이언트의 **/linux** 서브디렉토리를 보여줍니다. 필요한 경우 이를 **/linux-x86_64** 또는 **/linux-ppc64**로 대체하십시오.

```
cd ibm-commsserver-clients/linux
./installgskit
```

IBM 원격 API 클라이언트가 HTTPS를 사용하여 서버에 연결하기 전에 클라이언트에서 GSKIT 키 관리자 프로그램을 사용하여 보안 인증서 구성을 설정해야 합니다. 다음 단계를 수행하십시오.

1. 다음 명령을 사용하여 GSKIT 키 관리자를 실행하십시오.

```
/opt/ibm/sna/bin/snakeyman
```

키 관리자 사용자 인터페이스 내에서 CMS 형식으로 된 키 데이터베이스 파일 **/etc/opt/ibm/sna/ibmcs.kdb**를 여십시오.

2. 키 데이터베이스의 초기 암호는 **ibmcs**입니다. 보안 인증서를 설정하기 전에 이 암호를 반드시 변경하여 구성 보안을 유지해야 합니다. 암호 변경 대화상자에서 ‘암호를 파일로 보관’ 선택란에 표시하여 클라이언트가 키 데이터베이스를 열 수 있도록 새 암호를 저장해야 합니다.

GSKIT를 사용하여 HTTPS 보안 인증서 설정

3. Web Server의 보안 인증서에 서명하기 위해 사용된 인증 기관(CA) 인증서 사본을 얻어 키 데이터베이스에 설치하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 서명자 인증서를 선택하고 추가를 누르십시오.
4. WebSphere 서버가 클라이언트 보안 인증서를 요구하도록 구성된 경우 클라이언트는 Web Server의 보안 인증서 데이터베이스에 자체 인증서가 있는 CA가 발행한 인증서를 갖고 있어야 합니다. 새 인증서를 요청하려면 다음과 같이 하십시오.
 - a. 키 관리자 사용자 인터페이스에서 작성, 새 인증서 요청을 선택하고 요구되는 세부사항을 채우십시오.
 - b. 인증서를 저장하고 이를 파일로 발췌하여 CA에 보내십시오.
 - c. 인증서가 발행되면 Web Server의 데이터베이스에 저장하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 개인용 인증서를 선택하고 수신을 누르십시오.

자체 내부 테스트를 위한 임시 조치로, CA에서 인증서를 가져오지 않고 스스로 서명한 클라이언트 인증서를 작성할 수 있습니다. 그러나, 이 방법은 필요한 레벨의 보안을 보장하지 못하므로 실제 시스템에서는 이 방법을 사용해서는 안 됩니다. 스스로 서명한 인증서를 작성하려면 다음과 같이 하십시오.

- a. 키 관리자 사용자 인터페이스에서 작성, 자체 서명한 인증서 새로 작성을 선택하고 요구되는 세부사항을 채우십시오.
 - b. 인증서를 저장하고 파일로 발췌하십시오.
 - c. 인증서 파일을 Web Server의 데이터베이스에 저장하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 개인용 인증서를 선택하고 수신을 누르십시오.
5. 구성 인증서를 완성했으면 GSKIT 키 관리자를 종료하십시오.

Linux에서 원격 API 클라이언트 설치 제거

다음 명령을 사용하여 Linux에서 원격 API 클라이언트를 설치 제거할 수 있습니다.

```
/opt/ibm/sna/bin/sna stop  
rpm -e ibm-commserver-ptf  
rpm -e ibm-commserver-docs  
rpm -e ibm-commserver-ecl  
rpm -e ibm-commserver-cli  
rpm -e ibm-commserver  
rpm -e gsk7bas  
/sbin/shutdown -r now
```

이 명령에 나열된 모든 패키지가 모든 시스템에 설치되어 있는 것은 아닙니다.

Linux에서 IBM 원격 API 클라이언트를 설치 제거해도 나중에 설치할 때 사용할 수 있도록, 사용자 정의된 구성 정보가 그대로 남아 있습니다.

제 5 장 System z용 Linux에 IBM 원격 API 클라이언트 설치

이 장에서는 Linux에 IBM 원격 API 클라이언트를 설치하는 방법을 설명합니다. 이 소프트웨어를 사용하면 SNA 스택이 완전히 설치되지 않은 상태에서도 System z 메인프레임이 SNA 응용프로그램을 실행할 수 있습니다. System z용 Linux에 설치된 원격 API 클라이언트는 TCP/IP 네트워크를 사용하여 하나 이상의 Linux용 Communications Server(또는 CS/AIX 서버)에 연결될 수 있습니다.

소프트웨어를 설치하기 전에 IBM 원격 API 클라이언트 README 파일을 읽는 것이 좋습니다. 이 파일은 설치 CD의 `/ibm-commserver-clients/linux-systemz` 디렉토리에 있습니다.

하드웨어 및 소프트웨어 요구사항

하드웨어 요구사항

IBM 원격 API 클라이언트는 『Linux 운영 체제 버전』에 나열된 Linux 분배 방법 중 하나에 의해 지원되는 31비트 또는 64비트 System z 시스템을 요구합니다.

`uname -m` 명령을 사용하여 CPU 클래스를 확인하십시오. 31비트 환경을 나타내는 경우 s390을, 64비트 환경을 나타내는 경우 s390x를 보고해야 합니다.

Linux 운영 체제 버전

IBM 원격 API 클라이언트의 현재 버전은 다음 Linux 운영 체제 버전과 함께 테스트되었습니다. 이 소프트웨어는 다른 Linux 분배에서도 만족스럽게 작동할 수 있습니다.

- S/390용 RedHat Enterprise Linux 3(RHEL3-s390)
- zSeries용 RedHat Enterprise Linux 3(RHEL3-s390x)
- S/390용 RedHat Enterprise Linux 4(RHEL4-s390)
- zSeries용 RedHat Enterprise Linux 4(RHEL4-s390x)
- IBM 메인프레임용 SUSE Linux Enterprise Server 8(SLES8-s390*)
- IBM 메인프레임용 SUSE Linux Enterprise Server 9(SLES9-s390*)
- IBM 메인프레임용 SUSE Linux Enterprise Server 10(SLES10-s390x)

필요한 선택적 패키지에 대한 세부사항은 설치 CD의 **README** 파일을 참조하십시오.

Java

Java CPI-C API를 사용하는 경우 Java 소프트웨어가 필요합니다. 세부사항은 설치 CD의 **README** 파일을 참조하십시오.

GSKIT

클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결하는 경우 WebSphere 서버를 통해 서버에 HTTPS 프로토콜로 액세스하려면 GSKIT 소프트웨어가 필요합니다. GSKIT 소프트웨어는 설치 CD에 포함되어 있지만 이를 설치하려면 선택적인 일부 Linux 운영 체제 패키지가 필요합니다. 필요한 선택적 패키지에 대한 세부사항은 설치 CD의 **/ibm-commserver-clients/linux-systemz** 디렉토리에 있는 **README** 파일을 참조하십시오.

클라이언트 설치 프로세스를 실행할 때 모든 전제조건 패키지가 설치되어 있는 경우(이 장의 뒷부분에서 설명) GSKIT 소프트웨어도 이 프로세스의 일부로 자동으로 설치됩니다. 그렇지 않은 경우에는 나중에 설치할 수 있습니다.

제품 설치 세부사항 표시

원격 API 클라이언트 및 이미 설치된 관련 소프트웨어 패키지에 대한 정보를 표시할 수 있습니다. 설치된 모든 패키지를 나열하려면 다음 명령을 사용하십시오.

```
rpm -q -a
```

특정 패키지의 세부사항을 보려면 다음 명령을 사용하십시오.

```
rpm -q -i packagename
```

여기서 *packagename*은 설치한 패키지의 기본 이름입니다(예: **ibm-commserver-client**).

언어 환경 변수 설정

다음 명령을 사용하여 사용할 언어를 나타내는 LANG 변수를 변경하십시오.

```
export LANG=language
```

여기서 *language*를 사용할 언어의 식별자로 대체하십시오. 다음 중 하나가 될 수 있습니다.

식별자	언어
en_US	영어(미국)
ja_JP	일본어(PC)
de_DE	독일어
es_ES	스페인어
fr_FR	프랑스어
ko_KR	한국어
pt_BR	포르투갈어
zh_CN	중국어(단순화된 EUC)
zh_TW	대만어

System z용 Linux에 원격 API 클라이언트 설치

전제조건 소프트웨어의 설치를 마쳤으면 IBM 원격 API 클라이언트를 설치할 준비가 된 것입니다.

IBM 원격 API 클라이언트의 이전 레벨이 이미 설치되어 있는 경우, 이 새로운 레벨을 설치하기 전에 54 페이지의 『System z용 Linux에서 원격 API 클라이언트 설치 제거』에 있는 단계를 따라 이전 레벨을 제거하십시오. 새로운 설치에서 사용할 수 있도록 구성 정보는 그대로 남아 있습니다.

1. CD-ROM의 `/ibm-commsserver-clients/linux-systemz` 디렉토리에서 Linux System z 시스템으로 `ibm-commsserver-client-6.3.0.1-s390.tgz` 파일을 복사하거나 FTP로 전송하십시오. 파일을 복사하거나 FTP에서 다운로드하는 데 2진 모드를 사용하십시오.
2. 루트로 Linux System z 시스템에 로그인하십시오.
3. tar 파일의 압축을 풀어 빈 임시 디렉토리에 위치시키십시오.

```
mkdir /tmp/ibmcs
```

```
cd /tmp/ibmcs
```

```
tar -xzf ibm-commsserver-client-6.3.0.1-s390.tgz
```

4. `installibmcscli` 셸 스크립트를 실행하십시오.

```
./installibmcscli
```

이 셸 스크립트는 특정 전제조건을 검사하며 조건에 충족되지 못할 경우에는 경고 메시지를 발행합니다. 사용자가 Linux용 Communications Server 라이선스 조항을 읽고 승인했는지 확인할 수 있도록 프롬프트를 표시하기도 합니다. 아래에 설명된 대로 `installibmcscli` 명령에 추가 매개변수를 지정하면 이 프롬프트를 대체할 수 있습니다. 프롬프트에 응답하면 셸 스크립트가 `rpm` 패키지를 설치합니다. 해당 전제조건이 이미 설치된 경우 스크립트는 GSKIT 소프트웨어도 설치합니다.

5. IBM 원격 API 클라이언트 2진 디렉토리를 PATH에 추가하십시오. 이를 자동 수행하도록 프로파일을 변경할 수도 있습니다.

```
export PATH="$PATH:/opt/ibm/sna/bin"
```

```
export LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib
```

```
export LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib
```

64비트 응용프로그램을 실행하려는 경우에는 다음 명령을 사용하십시오.

```
export LD_LIBRARY_PATH=/usr/lib64:/opt/ibm/sna/lib64
```

```
export LD_RUN_PATH=/usr/lib64:/opt/ibm/sna/lib64
```

Java CPI-C 응용프로그램의 경우 다음 환경 변수도 설정해야 합니다.

```
export CLASSPATH=$CLASSPATH:/opt/ibm/sna/java/cpic.jar
```

일부 응용프로그램의 경우 LD_PRELOAD 환경 변수를 설정할 수도 있지만 해당 프로파일에서 이 변수를 전체 바꾸기로 수행하지 않아야 합니다.

```
export LD_PRELOAD=/usr/lib/libpLiS.so
```

6. IBM 원격 API 클라이언트를 시작하십시오. 설치 후에는 시스템이 다시 시동될 때 자동으로 시작됩니다. 이 작업을 수행할 때는 아직도 CD 디렉토리에 있지 않은지 확인하십시오.

```
cd /
```

```
sna start
```

7. 설치가 완료된 후에는 설치 프로세스 동안 작성했던 **tgz** 파일 및 임시 디렉토리를 지울 수 있습니다.

주: IBM 원격 API 클라이언트가 HTTPS를 사용하여 서버에 연결하기 전에 클라이언트에서 GSKIT 키 관리자 프로그램을 사용하여 보안 인증서 구성을 설정해야 합니다. 자세한 내용은 『GSKIT를 사용하여 HTTPS 보안 인증서 설정』을 참조하십시오.

또한 클라이언트 네트워크 데이터 파일을 갱신하여 클라이언트가 연결할 수 있도록 Linux용 Communications Server를 지정하고 HTTPS 지원을 제공하는 WebSphere 서버 이름을 지정해야 합니다. 자세한 정보는 *Communications Server for Linux Administration Guide*의 원격 API 클라이언트 관리 절을 참조하십시오.

GSKIT를 사용하여 HTTPS 보안 인증서 설정

클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결하는 경우 GSKIT 키 관리자 소프트웨어가 설치되어 있어야 합니다. 설치 CD의 **README** 파일에 설명된 것처럼 필요한 Linux 운영 체제 전체 조건이 설치된 경우 이 소프트웨어의 설치는 일반적으로 클라이언트 설치 과정의 일부로 이루어집니다. GSKIT가 클라이언트 설치 과정의 일부로 설치되지 않는다면 지금 막 전체 조건을 설치한 경우 다음 단계에 따라 GSKIT 소프트웨어를 설치할 수 있습니다.

1. CD-ROM의 **/ibm-commserver-clients/linux-systemz** 디렉토리에서 Linux System z 시스템으로 **ibm-commserver-client-6.3.0.1-s390.tgz** 파일을 복사하거나 FTP로 전송하십시오. 파일을 복사하거나 FTP에서 다운로드하는 데 2진 모드를 사용하십시오.
2. 루트로 Linux System z 시스템에 로그인하십시오.

- tar 파일의 압축을 풀어 빈 임시 디렉토리에 위치시키십시오.

```
mkdir /tmp/ibmcs
```

```
cd /tmp/ibmcs
```

```
tar -xzf ibm-commsserver-client-6.3.0.1-s390.tgz
```

- 다음과 같이 **installgskit** 셸 스크립트를 실행하십시오.

```
./installgskit
```

- 설치가 완료된 후에는 설치 프로세스 동안 작성했던 **tgz** 파일 및 임시 디렉토리를 지울 수 있습니다.

IBM 원격 API 클라이언트가 HTTPS를 사용하여 서버에 연결하기 전에 클라이언트에서 GSKIT 키 관리자 프로그램을 사용하여 보안 인증서 구성을 설정해야 합니다. 다음 단계를 수행하십시오.

- 다음 명령을 사용하여 GSKIT 키 관리자를 실행하십시오.

```
/opt/ibm/sna/bin/snakeyman
```

키 관리자 사용자 인터페이스 내에서 CMS 형식으로 된 키 데이터베이스 파일 **/etc/opt/ibm/sna/ibmcs.kdb**를 여십시오.

- 키 데이터베이스의 초기 암호는 **ibmcs**입니다. 보안 인증서를 설정하기 전에 이 암호를 반드시 변경하여 구성 보안을 유지해야 합니다. 암호 변경 대화상자에서 ‘암호를 파일로 보관’ 선택란에 표시하여 클라이언트가 키 데이터베이스를 열 수 있도록 새 암호를 저장해야 합니다.
- Web Server의 보안 인증서에 서명하기 위해 사용된 인증 기관(CA) 인증서 사본을 얻어 키 데이터베이스에 설치하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 서명자 인증서를 선택하고 추가를 누르십시오.
- WebSphere 서버가 클라이언트 보안 인증서를 요구하도록 구성된 경우 클라이언트는 Web Server의 보안 인증서 데이터베이스에 자체 인증서가 있는 CA가 발행한 인증서를 갖고 있어야 합니다. 새 인증서를 요청하려면 다음과 같이 하십시오.
 - 키 관리자 사용자 인터페이스에서 작성, 새 인증서 요청을 선택하고 요구되는 세부사항을 채우십시오.
 - 인증서를 저장하고 이를 파일로 발췌하여 CA에 보내십시오.
 - 인증서가 발행되면 Web Server의 데이터베이스에 저장하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 개인용 인증서를 선택하고 수신을 누르십시오.

자체 내부 테스트를 위한 임시 조치로, CA에서 인증서를 가져오지 않고 스스로 서명한 클라이언트 인증서를 작성할 수 있습니다. 그러나, 이 방법은 필요한 레벨의

GSKIT를 사용하여 HTTPS 보안 인증서 설정

보안을 보장하지 못하므로 실제 시스템에서는 이 방법을 사용해서는 안 됩니다. 스스로 서명한 인증서를 작성하려면 다음과 같이 하십시오.

- a. 키 관리자 사용자 인터페이스에서 작성, 자체 서명한 인증서 새로 작성을 선택하고 요구되는 세부사항을 채우십시오.
 - b. 인증서를 저장하고 파일로 발취하십시오.
 - c. 인증서 파일을 Web Server의 데이터베이스에 저장하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 개인용 인증서를 선택하고 수신을 누르십시오.
5. 구성 인증서를 완성했으면 GSKIT 키 관리자를 종료하십시오.

System z용 Linux에서 원격 API 클라이언트 설치 제거

다음 명령을 사용하여 System z용 Linux에서 원격 API 클라이언트를 설치 제거할 수 있습니다.

```
/opt/ibm/sna/bin/sna stop  
rpm -e ibm-commserver-ptf  
rpm -e ibm-commserver-docs  
rpm -e ibm-commserver-ecl  
rpm -e ibm-commserver-cli  
rpm -e ibm-commserver  
rpm -e gsk7bas  
/sbin/shutdown -r now
```

이 명령에 나열된 모든 패키지가 모든 시스템에 설치되어 있는 것은 아닙니다.

System z용 Linux에서 IBM 원격 API 클라이언트를 설치 제거해도 나중에 설치할 때 사용할 수 있도록, 사용자 정의된 구성 정보가 그대로 남아 있습니다.

제 6 장 AIX 시스템에 IBM 원격 API 클라이언트 설치

이 장에서는 AIX에 IBM 원격 API 클라이언트를 설치하는 방법을 설명합니다. 이 소프트웨어를 사용하면 SNA 스택이 완전히 설치되지 않은 상태에서도 AIX 워크스테이션이 SNA 응용프로그램을 실행할 수 있습니다. AIX에 설치된 원격 API 클라이언트는 TCP/IP 네트워크를 사용하여 하나 이상의 Linux용 Communications Server(또는 CS/AIX 서버)에 연결될 수 있습니다.

소프트웨어를 설치하기 전에 IBM 원격 API 클라이언트 README 파일을 읽는 것이 좋습니다. 이 파일은 설치 CD의 `/ibm-commserver-clients/aix` 디렉토리에 있습니다.

하드웨어 및 소프트웨어 요구사항

하드웨어 요구사항

IBM 원격 API 클라이언트는 『운영 체제 버전』에 나열된 AIX 운영 체제 중 하나에 의해 지원되는 pSeries 시스템을 요구합니다.

운영 체제 버전

IBM 원격 API 클라이언트의 현재 버전은 다음 운영 체제 버전과 함께 테스트되었습니다.

- AIX v5.2-ML5
- AIX v5.3-ML1

Java

Java CPI-C API를 사용하는 경우 Java 소프트웨어가 필요합니다. <http://www.ibm.com/developerworks/java/jdk>에서 제공하는 Java 1.4.2 SDK는 모든 요구 사항을 충족시킵니다. 64비트 Java도 제공될 수 있지만, Java CPI-C API는 32비트 JVM을 요구합니다.

`installp` 명령을 사용하여 Java SDK 패키지를 설치하십시오.

GSKIT

클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결하는 경우 WebSphere 서버를 통해 서버에 HTTPS 프로토콜로 액세스하려면 GSKIT 소프트웨어가 필요합니다. 세부사항은 설치 CD의 `/ibm-commserver-clients/aix`에 있는 README 파일을 참조하십시오. GSKIT 소프트웨어는 이 장의 뒷부분에서 설명할 기본 클라이언트 설치 프로세스의 일부로 설치됩니다.

언어 환경 변수 변경

원격 API 클라이언트를 사용할 경우 LANG 변수가 C로 설정되지 않았는지 확인하십시오.

다음 프로시저에 따라 사용 중인 LANG 변수를 표시하거나 LANG 변수를 변경하십시오.

1. 기본 SMIT 메뉴에서 시스템 환경을 선택하십시오.
2. 다음 SMIT 메뉴에서 언어 환경 관리를 선택하십시오.
3. 다음 SMIT 메뉴에서 기본 언어 환경 변경/표시를 선택하십시오.
4. 다음 SMIT 메뉴에서 기본 문화, 언어 또는 키보드 변경/표시를 선택하십시오.
5. 사용할 언어를 선택하십시오. 예를 들어 미국 영어 메시지를 사용하는 경우 en_US를 선택하십시오.

AIX에 원격 API 클라이언트 설치

전제조건 소프트웨어의 설치를 마쳤으면 IBM 원격 API 클라이언트를 설치할 준비가 된 것입니다.

IBM 원격 API 클라이언트의 이전 레벨이 이미 설치되어 있는 경우, 이 새로운 레벨을 설치하기 전에 59 페이지의 『AIX에서 원격 API 클라이언트 설치 제거』에 있는 단계를 따라 이전 레벨을 제거하십시오. 새로운 설치에서 사용할 수 있도록 구성 정보는 그대로 남아 있습니다.

AIX 워크스테이션에 파일을 복사하여 원격 API 클라이언트 설치

원격 API 클라이언트를 설치하려면 다음 단계를 수행하십시오.

1. CD-ROM의 **/ibm-commserver-clients/aix** 디렉토리에서 AIX 워크스테이션으로 **sna.client.6.3.0.1.bff** 파일을 복사하거나 FTP로 전송하십시오. 파일을 복사하거나 FTP에서 다운로드하는 데 2진 모드를 사용하십시오.

클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결하는 경우 CD의 동일한 디렉토리에서 2개의 파일 **gskta.*.I** 및 **gksa.*.I**도 복사하거나 FTP로 전송해야 합니다. 이들 파일에는 클라이언트에서 HTTPS로 액세스하는 데 필요한 GSKIT 소프트웨어가 포함되어 있습니다.

2. 루트로 AIX 워크스테이션에 로그인하십시오.
3. **smit** 또는 **installp**를 사용하여 AIX 클라이언트를 설치하십시오. 이 작업 수행에 대한 지시사항은 설치 CD의 **/ibm-commserver-clients/aix**에 있는 **README** 파일을 참조하십시오.
4. 클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결하는 경우 **README** 파일의 지시사항에 따라 GSKIT 파일을 설치하십시오.

5. 설치 프로세스가 완료되면 작업 디렉토리에서 **sna.client.6.3.0.1.bff** 파일 및 GSKIT 파일을 삭제할 수 있습니다.
6. IBM 원격 API 클라이언트를 시작하십시오. 설치 후에는 시스템이 다시 시동될 때 자동으로 시작됩니다.

```
cd /
sna start
```

주: IBM 원격 API 클라이언트가 HTTPS를 사용하여 서버에 연결하기 전에 클라이언트에서 GSKIT 키 관리자 프로그램을 사용하여 보안 인증서 구성을 설정해야 합니다. 자세한 내용은 58 페이지의 『GSKIT를 사용하여 HTTPS 보안 인증서 설정』을 참조하십시오.

또한 클라이언트 네트워크 데이터 파일을 갱신하여 클라이언트가 연결할 수 있도록 Linux용 Communications Server를 지정하고 HTTPS 지원을 제공하는 WebSphere 서버 이름을 지정해야 합니다. 자세한 정보는 *Communications Server for Linux Administration Guide*의 원격 API 클라이언트 관리 절을 참조하십시오.

CD에서 원격 API 클라이언트 설치

원격 API 클라이언트를 설치하려면 다음 단계를 수행하십시오.

1. 루트로 AIX 워크스테이션에 로그인하십시오.
2. 다음 명령을 사용하여 AIX 워크스테이션에 CD를 마운트하십시오.

```
mount -o ro /dev/cd0 /mnt
```

3. **smit** 또는 **installp**를 사용하여 AIX 클라이언트를 설치하십시오. 이 작업 수행에 대한 지시사항은 설치 CD의 **/ibm-commserver-clients/aix**에 있는 **README** 파일을 참조하십시오.
4. 클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결하는 경우 **README** 파일의 지시사항에 따라 GSKIT 파일을 설치하십시오.
5. 설치 프로세스가 완료되면 다음 명령을 사용하여 CD를 마운트 해제하십시오.

```
umount /mnt
```

6. IBM 원격 API 클라이언트를 시작하십시오. 설치 후에는 시스템이 다시 시동될 때 자동으로 시작됩니다. 이 작업을 수행할 때는 아직도 CD 디렉토리에 있지 않은지 확인하십시오.

```
cd /
sna start
```

주: IBM 원격 API 클라이언트가 HTTPS를 사용하여 서버에 연결하기 전에 클라이언트에서 GSKIT 키 관리자 프로그램을 사용하여 보안 인증서 구성을 설정해야 합니다. 자세한 내용은 『GSKIT를 사용하여 HTTPS 보안 인증서 설정』을 참조하십시오.

또한 클라이언트 네트워크 데이터 파일을 갱신하여 클라이언트가 연결할 수 있도록 Linux용 Communications Server를 지정하고 HTTPS 지원을 제공하는 WebSphere 서버 이름을 지정해야 합니다. 자세한 정보는 *Communications Server for Linux Administration Guide*의 원격 API 클라이언트 관리 절을 참조하십시오.

GSKIT를 사용하여 HTTPS 보안 인증서 설정

IBM 원격 API 클라이언트가 HTTPS를 사용하여 서버에 연결하기 전에 클라이언트에서 GSKIT 키 관리자 프로그램을 사용하여 보안 인증서 구성을 설정해야 합니다. 다음 단계를 수행하십시오.

1. 다음 명령을 사용하여 GSKIT 키 관리자를 실행하십시오.

```
/usr/bin/snakeyman
```

키 관리자 사용자 인터페이스 내에서 CMS 형식으로 된 키 데이터베이스 파일 **/etc/sna/ibmcs.kdb**를 여십시오.

2. 키 데이터베이스의 초기 암호는 **ibmcs**입니다. 보안 인증서를 설정하기 전에 이 암호를 반드시 변경하여 구성 보안을 유지해야 합니다. 암호 변경 대화상자에서 ‘암호를 파일로 보관’ 선택란에 표시하여 클라이언트가 키 데이터베이스를 열 수 있도록 새 암호를 저장해야 합니다.
3. Web Server의 보안 인증서에 서명하기 위해 사용된 인증 기관(CA) 인증서 사본을 얻어 키 데이터베이스에 설치하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 서명자 인증서를 선택하고 추가를 누르십시오.
4. WebSphere 서버가 클라이언트 보안 인증서를 요구하도록 구성된 경우 클라이언트는 Web Server의 보안 인증서 데이터베이스에 자체 인증서가 있는 CA가 발행한 인증서를 갖고 있어야 합니다. 새 인증서를 요청하려면 다음과 같이 하십시오.
 - a. 키 관리자 사용자 인터페이스에서 작성, 새 인증서 요청을 선택하고 요구되는 세부사항을 채우십시오.
 - b. 인증서를 저장하고 이를 파일로 발췌하여 CA에 보내십시오.
 - c. 인증서가 발행되면 Web Server의 데이터베이스에 저장하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 개인용 인증서를 선택하고 수신을 누르십시오.

자체 내부 테스트를 위한 임시 조치로, CA에서 인증서를 가져오지 않고 스스로 서명한 클라이언트 인증서를 작성할 수 있습니다. 그러나, 이 방법은 필요한 레벨의

보안을 보장하지 못하므로 실제 시스템에서는 이 방법을 사용해서는 안 됩니다. 스스로 서명한 인증서를 작성하려면 다음과 같이 하십시오.

- a. 키 관리자 사용자 인터페이스에서 작성, 자체 서명한 인증서 새로 작성을 선택하고 요구되는 세부사항을 채우십시오.
 - b. 인증서를 저장하고 파일로 발췌하십시오.
 - c. 인증서 파일을 Web Server의 데이터베이스에 저장하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 개인용 인증서를 선택하고 수신을 누르십시오.
5. 구성 인증서를 완성했으면 GSKIT 키 관리자를 종료하십시오.

AIX에서 원격 API 클라이언트 설치 제거

다음 명령을 사용하여 원격 API 클라이언트를 설치 제거할 수 있습니다.

1. 이 소프트웨어가 실행 중인 경우 다음 명령을 사용하여 클라이언트 소프트웨어를 중지하십시오.

sna stop

2. 루트 권한으로 로그인하십시오.
3. 다음 명령 중 하나를 사용하여 원격 API 클라이언트 패키지 및 연관된 소프트웨어 패키지를 제거하십시오.

installp를 사용하여 패키지를 제거하려면 다음과 같이 하십시오.

installp -u sna.client

smit를 사용하여 패키지를 제거하려면 다음과 같이 하십시오.

smit remove

제 7 장 Windows에 대한 원격 API 클라이언트 설치 계획 세우기

이 장에서는 Windows에 IBM 원격 API 클라이언트를 설치하는 방법을 설명합니다. 이 소프트웨어를 사용하면 PC에 SNA 스택이 완전히 설치되지 않은 상태에서도 해당 PC가 SNA 응용프로그램을 실행할 수 있습니다. Windows에 설치된 원격 API 클라이언트는 TCP/IP 네트워크를 사용하여 하나 이상의 Linux용 Communications Server(또는 CS/AIX 서버)에 연결될 수 있습니다.

사용하는 하드웨어 및 Windows 버전에 따라 Windows에서 두 가지 계열의 IBM 원격 API 클라이언트를 사용할 수 있습니다. 이 장에서 설명하는 정보는 차이점이 명시되지 않은 한 두 계열 모두에 적용됩니다.

- 32비트 클라이언트는 Microsoft Windows 2000, 2003 또는 XP를 실행하는 Intel 기반 컴퓨터에서 실행됩니다.
- x64 클라이언트는 Microsoft Windows Server 2003 x64 Edition 또는 Microsoft Windows XP Professional x64 Edition을 실행하는 AMD64 또는 Intel EM64T 컴퓨터에서 실행됩니다.

Windows의 IBM 원격 API 클라이언트가 제공하는 인터페이스는 Windows 및 Microsoft Host Integration Server용 IBM Communications Server에서 제공하는 인터페이스와 광범위하게 호환됩니다.

Windows의 IBM 원격 API 클라이언트 소프트웨어 개발 키트(SDK)은 선택적 패키지로, 이 패키지를 사용하면 원격 API 클라이언트에서 APPC, CPI-C, LUA 및 CSV API를 사용하여 응용프로그램을 개발할 수 있습니다. 이 API에 대한 자세한 정보는 해당 프로그래머용 참조 안내서를 참조하십시오. 원격 API 클라이언트를 사용하여 기존 응용프로그램을 실행만(새 응용프로그램 개발이 아님) 하는 경우에는 이 패키지를 설치할 필요가 없습니다.

하드웨어 및 소프트웨어 요구사항

Windows에서 Remote API Client 및 설치 프로그램을 실행하려면 컴퓨터가 다음 요구사항을 충족시켜야 합니다.

- 다음 운영 체제 중 하나에서 실행되어야 합니다.
 - 32비트 Windows 클라이언트:
 - Windows 2000
 - Windows XP
 - Windows 2003

- x64 Windows 클라이언트:
 - Microsoft Windows XP Professional x64 Edition
 - Microsoft Windows Server 2003 x64 Edition
- 다음 메커니즘 중 하나를 사용하여 하나 이상의 Linux용 Communications Server에 액세스할 수 있어야 합니다.
 - TCP/IP 네트워크를 통해 서버에 액세스
 - Linux용 Communications Server에 대한 HTTPS 액세스를 제공하는 WebSphere 서버에 액세스(이 경우 클라이언트 설치 프로세스의 일부로 GSKIT 소프트웨어가 설치되어야 합니다.)

설치 프로그램에 액세스

원격 API 클라이언트 및 SDK 소프트웨어, GSKIT 소프트웨어 및 설치 프로그램은 설치 CD에 Windows 형식으로 들어 있으므로, CD에서 Windows 컴퓨터로 설치할 수 있습니다. 원격 API 클라이언트 소프트웨어를 각각의 Windows 클라이언트 PC에 설치해야 합니다. SDK는 Windows Remote API를 사용하여 클라이언트에서 새 응용 프로그램을 개발하는 경우에만 필요하며 기존 응용 프로그램을 실행만 하는 경우에는 필요하지 않습니다. 클라이언트가 HTTPS를 사용하여 Linux용 Communications Server에 연결하는 경우에만 GSKIT 소프트웨어가 필요합니다.

Windows의 원격 API 클라이언트 설치 이미지는 설치 CD에 포함된 스스로 압축이 풀리는 ZIP 실행 파일입니다.

- 32비트 클라이언트의 경우 CD의 **/ibm-commserver-clients/windows** 디렉토리에 있는 **i_w32cli.exe**입니다.
- 64비트 클라이언트의 경우 CD의 **/ibm-commserver-clients/win-x64** 디렉토리에 있는 **i_w64cli.exe**입니다.

이 파일을 네트워크에서 다른 Windows PC로 복사할 수 있기 때문에 Linux용 Communications Server 제공 CD에 직접 액세스하지 않아도 설치가 가능합니다. 이 실행 파일을 실행하면 설치 이미지의 압축이 풀리면서 설치 프로그램이 자동으로 실행됩니다. 임시 디렉토리로 설치 이미지의 압축만 풀려는 경우 예를 들어, 명령행에서 설치 프로그램을 실행하려는 경우에는 스스로 압축이 풀리는 ZIP 실행 파일을 압축 풀기 프로그램에 로드하여 이를 수행할 수 있습니다.

특정 컴퓨터에서 설치 프로그램을 처음 실행하는 경우 프로그램이 선택된 소스에서 실행됩니다. 프로그램이 자동으로 설치 프로세스 전체를 처리하고 기본 구성을 설정하며 아이콘을 설치 및 작성합니다. 설치가 완료된 후, 소프트웨어의 재설치가 필요할 경우 이 설치 프로그램을 사용할 수 있습니다(파일 관리자에서 선택하거나 아이콘을 선택).

임시 디렉토리에 원격 API 클라이언트 설치 이미지의 압축을 푼 다음 다음 두 방법 중 하나를 사용하여 소프트웨어를 설치할 수 있습니다.

- 『설치 프로그램을 사용하여 Windows에 원격 API 클라이언트 설치』에 설명된 대로 Windows를 통해 설치 프로그램을 실행하십시오. SDK를 설치하려는 경우 이 방법을 사용해야 합니다.
- 67 페이지의 『명령행에서 원격 API 클라이언트 소프트웨어 설치』에 설명된 대로 명령행에서 **setup** 명령을 입력하십시오. 이 방법을 사용하면 SDK를 설치할 수 있습니다.

주: IBM 원격 API 클라이언트가 HTTPS를 사용하여 서버에 연결하기 전에 클라이언트에 GSKIT 소프트웨어를 설치하고 GSKIT 키 관리자 프로그램을 사용하여 보안 인증서 구성을 설정해야 합니다. 자세한 내용은 69 페이지의 『GSKIT 소프트웨어 설치 및 보안 인증서 설정』 및 70 페이지의 『GSKIT를 사용하여 HTTPS 보안 인증서 설정』을 참조하십시오.

또한 클라이언트 네트워크 데이터 파일을 갱신하여 클라이언트가 연결할 수 있도록 Linux용 Communications Server를 지정하고 HTTPS 지원을 제공하는 WebSphere 서버 이름을 지정해야 합니다. 자세한 정보는 *Communications Server for Linux Administration Guide*의 원격 API 클라이언트 관리 절을 참조하십시오.

설치 프로그램을 사용하여 Windows에 원격 API 클라이언트 설치

설치 프로그램을 스스로 압축이 풀리는 ZIP 실행 파일 **i_w32cli.exe**(32-bit client) 또는 **i_w64cli.exe**(x64 클라이언트) 실행의 일부로 자동으로 실행하거나, 명령행에서 수동으로 실행하십시오. 프로그램에서 설치 언어 선택 화면이 먼저 나타납니다.

1. 원격 API 클라이언트 설치 및 구성에 사용할 언어를 선택하고 확인을 누르십시오.

설치 프로그램을 소개하는 시작 화면이 표시됩니다.

2. 다음을 선택하여 설치를 계속 진행하십시오.

프로그램이 소프트웨어 라이선스 계약을 표시하는데, 이 계약을 읽고 이해해야 합니다.

3. 라이선스 조항을 승인할 경우 승인을 선택하고 계속 진행하십시오.

프로그램에서 파일이 설치될 대상 디렉토리를 지정하라는 프롬프트를 표시합니다.

4. 대상 디렉토리를 입력하십시오.

원하는 설치 유형을 선택하라는 프롬프트가 표시됩니다.

표준 SDK를 설치할 필요가 없는 경우 이 옵션을 선택하십시오. SDK는

설치 프로그램을 사용하여 Windows에 원격 API 클라이언트 설치

Windows Remote API를 사용하여 클라이언트에서 새 응용프로그램을 개발하는 경우에만 필요하며 기존 응용프로그램을 실행만 하는 경우에는 필요하지 않습니다.

개발자 SDK를 설치할 필요가 있는 경우에는 이 옵션을 선택하십시오. 즉, 클라이언트에서 Windows Remote API를 사용하여 새 응용프로그램을 개발하려는 경우를 말합니다.

주: SDK를 설치하려면 개발자를 선택해야 합니다.

5. 설치 유형을 선택하십시오.

이제 Windows의 원격 API 클라이언트용 아이콘이 표시될 프로그램 폴더 이름을 입력하라는 프롬프트가 나타납니다.

6. 폴더 이름을 입력하십시오.

7. 이름이 이 설치 프로그램에 사용되는 파일과 같지만 원격 API 클라이언트 파일(예 : 일부 다른 SNA 소프트웨어의 파일)이 아닌 **.DLL** 파일이 시스템 디렉토리에 이미 들어 있는 경우, 다음 중 하나를 수행하라는 프롬프트가 표시됩니다.

- 원격 API 클라이언트 **.DLL** 파일을 복사하면서 기존 **.DLL** 파일을 겹쳐쓰십시오.
- 기존 **.DLL** 파일을 설치 디렉토리 내의 **OTHERSNA**라는 이름의 서브디렉토리로 복사한 다음 원격 API 클라이언트 **.DLL** 파일을 설치합니다. 이 옵션을 사용하면 원격 API 클라이언트 파일을 나중에 설치 제거할 경우 원격 API 클라이언트 설치 이전의 원래 설정을 복원할 수 있습니다(73 페이지의 『원격 API 클라이언트 소프트웨어 설치 제거』 참조).
- 클라이언트 소프트웨어 설치를 취소하십시오.

원격 API 클라이언트 **.DLL** 파일이 이미 존재하는 경우 설치 프로그램에 이를 나타내는 메시지가 표시됩니다. 기존 파일의 버전 번호가 설치 프로그램 **.DLL** 파일의 버전 번호보다 낮은 경우에만 새 **.DLL** 파일이 기존 **.DLL** 파일을 겹쳐씁니다.

8. 이 시점에서 설치 프로그램이 지정된 소스로부터 파일을 복사한 후 해당 위치에 설치합니다. 이 프로세스 동안 정보 표시줄에 완료된 설치 부분이 표시됩니다. **.DLL** 파일이 시스템 또는 동등한 디렉토리로 복사되고 기타 파일은 2단계에서 지정한 대상 디렉토리로 복사됩니다. 각 파일 전송 조작 동안 지정된 디렉토리에 작성되어 있는 **setup.log** 파일로 레코드가 작성됩니다. 작성될 파일 중 일부가 『읽기 전용』이거나 다른 이유로 파일을 복사할 수 없는 경우 새 파일이 제거되며 **setup.log** 파일을 살펴보는 메시지가 발행됩니다.

9. 설치 프로그램이 실행되는 소스에 필요한 파일이 모두 없는 경우 디렉토리 이름에 대해 프롬프트가 표시됩니다. 필요한 파일이 있는 디렉토리 이름을 입력하십시오.

설치 프로그램을 사용하여 Windows에 원격 API 클라이언트 설치

지정된 정보만으로 원격 API 클라이언트 파일의 사본을 찾을 수 없는 경우 프로그램에 이 화면이 다시 표시됩니다.

10. 필요한 파일이 복사되면 설치 프로그램이 구성 창을 표시합니다.

기본 구성 값은 도메인 구성 파일에서 가져옵니다. 자세한 내용은 *Communications Server for Linux Administration Guide*를 참조하십시오. 이 디폴트 값을 사용하지 않을 경우 아래에 표시된 대로 구성할 수 있습니다.

도메인 Linux용 Communications Server 클라이언트/서버 도메인 이름을 지정하십시오.

다음 설정은 선택적입니다.

서버 이름

이 클라이언트가 연결될 수 있는 최대 9개 서버로 이루어진 목록이 화면에 표시됩니다. 이 목록에 표시되는 서버 순서는 클라이언트가 이 서버를 선택하는 순서입니다. 클라이언트를 목록의 첫 번째 서버에 연결할 수 없는 경우 다음 서버에 연결을 시도합니다.

- 새 서버를 목록에 추가하려면 추가 버튼을 사용하십시오.
- 목록에서 서버를 제거하려면 서버를 선택한 후 제거 버튼을 사용하십시오.
- 서버를 목록 위 또는 아래로 이동하려면 서버를 선택한 후 목록 측면에 있는 버튼을 사용하십시오.

클라이언트가 서버와 동일한 사설 네트워크에 있고 TCP/IP를 사용하여 액세스되는 경우 각 서버는 단지 서버 이름으로 식별됩니다.

클라이언트가 HTTPS를 사용하여 서버에 액세스하는 경우 HTTPS 지원을 제공하는 WebSphere 서버 이름 및 Linux용 Communications Server의 이름을 다음 형식으로 지정하여 각 서버를 식별해야 합니다.

webservername : servername1

이 경우 WebSphere가 HTTPS 연결을 위해 기본 포트 443을 사용하도록 설정되어 있다고 가정합니다. 네트워크 관리자가 다른 포트 번호를 사용하도록 WebSphere를 구성한 경우 다음 형식으로 포트 번호를 포함하십시오.

webservername : portnumber : servername1

HTTPS 연결을 지원하도록 WebSphere를 구성하는 데 대한 자세한 내용은 32 페이지의 『WebSphere Application Server 구성』을 참조하십시오.

UDP 브로드캐스트

이 클라이언트가 UDP 브로드캐스트를 사용하여 서버에 연결할지 여부를

지정하십시오. 이 옵션을 선택할 경우 클라이언트가 네트워크에서 UDP 브로드캐스트를 전송하여 특정 서버에 직접 연결하지 않고 서버 연결 위치를 지정합니다.

디폴트 설정으로 UDP 브로드캐스트를 사용합니다. 이 설정을 변경하려면 상자를 누르십시오.

고급 설치 프로그램에서 제공한 디폴트 위치에 값을 추가하려면 창의 맨 아래에 있는 고급 버튼을 누르십시오. 설치 프로그램이 Windows 클라이언트 구성을 위한 고급 설정이 포함된 고급 옵션 창을 표시합니다. 대부분의 사용자가 이 매개변수에 대해 디폴트 설정을 사용할 수 있기 때문에 해당 대화상자에서 설정을 변경할 필요가 없습니다.

이 매개변수에 대한 자세한 내용은 『원격 API 클라이언트 구성을 위한 고급 옵션』을 참조하십시오.

구성 매개변수 또는 설정에 대한 자세한 내용은 도움말을 누르십시오.

11. 구성 창을 완료했다면 확인을 누르십시오. 이 화면이 올바르게 완료되지 않으면 설치 프로그램이 메시지를 표시합니다.
12. 설치가 완료되면 완료 창이 표시됩니다. 설치 프로그램을 종료한 후 다음 수행 조치 중 하나 또는 모두를 선택할 수 있습니다.

README 파일 보기

README 파일을 보십시오.

클라이언트 시작

Linux용 Communications Server 클라이언트 실행을 시작하십시오.

완료를 선택하여 설치 프로그램을 종료하십시오.

원격 API 클라이언트 구성을 위한 고급 옵션

고급 옵션 창을 사용하여 원격 API 클라이언트의 일부 고급 매개변수를 구성할 수 있습니다. 대부분의 사용자가 이 매개변수를 수정할 필요는 없지만 필요에 따라 디폴트 설정을 조정할 수는 있습니다.

LAN 액세스 시간종료

서버에 대한 클라이언트 연결이 닫히기 전까지 대기 상태로 남아 있을 수 있는 시간을 초 단위로 지정하십시오. 이 체크 박스에 표시가 되지 않으면 LAN 액세스 시간종료가 지정되지 않은 것입니다(따라서 무한정 시간종료가 사용됨). 이 상자를 선택하면 인접 필드에 시간종료 값을 초 단위로 입력할 수 있습니다. 최소값은 60(60초)입니다. 상자를 공백으로 두거나 60보다 작은 값을 지정하면 원격 API 클라이언트가 최소값(60)을 사용합니다.

최대 브로드캐스트 시도 수

클라이언트가 브로드캐스트를 사용하여 서버에 연결을 시도하는 최대 횟수를 지정하십시오. 고급 옵션 창이 열리면 디폴트 값(5)이 표시됩니다. 이 상자의 값은 주 구성 창에서 UDP 브로드캐스트 체크 박스를 선택한 경우에만 사용됩니다.

재연결 시간종료

서버 종료 이후 서버에 재연결을 시도하기 전 클라이언트가 대기하는 시간을 초 단위로 지정하십시오. 고급 옵션 창이 열리면 디폴트 값(200)이 표시됩니다.

이 매개변수에 대한 자세한 내용은 [도움말](#)을 누르십시오.

고급 옵션 창을 완료했으면 **확인**을 누르십시오. 화면을 올바르게 완료하면 설치 프로그램이 구성 창으로 리턴됩니다. 원격 API 클라이언트를 새로 설치하는 경우 66 페이지의 11단계로 리턴하십시오. 그렇지 않으면 구성 대화상자의 **확인** 버튼을 눌러 구성을 완료하십시오.

명령행에서 원격 API 클라이언트 소프트웨어 설치

주: SDK를 설치하려는 경우 63 페이지의 『설치 프로그램을 사용하여 Windows에 원격 API 클라이언트 설치』에 설명된 대로 설치 프로그램을 사용해야 합니다. 명령행에서 SDK를 설치할 수는 없습니다.

임시 디렉토리에 원격 API 클라이언트 설치 이미지의 압축을 푼 다음 Windows를 통해 설치 프로그램을 사용하지 않고 명령행에서 원격 API 클라이언트를 설치할 수 있습니다. 명령행에서 하나 이상의 옵션으로 **setup** 명령을 입력하십시오. 이 옵션은 대문자 또는 소문자로 입력이 가능하며 /(슬래시) 또는 -(하이픈)이 앞에 올 수 있습니다. 폴더와 같은 매개변수가 공백이 포함된 문자열인 경우 문자열에 큰따옴표를 붙여야 합니다.

setup 명령을 입력하면 설치 프로그램이 명령행에 포함되지 않은 정보에 대해 프롬프트를 표시하며 여러 설치 단계에서 확인 메시지를 표시합니다. 설치 프로그램에서 프롬프트가 표시되지 않도록 하려면 소프트웨어 라이선스 계약 조항을 승인할 수 있도록 **-accept-s** 옵션을 사용하여 자동 모드에서 프로그램을 실행시켜십시오.

다음은 **setup** 명령 옵션입니다.

- .-?** 명령행 옵션 목록을 표시합니다. 이 옵션 목록은 **-h** 옵션과 같습니다.
- h** 명령행 옵션 목록을 표시합니다. 이 옵션 목록은 **.-?** 옵션과 같습니다.

-accept -s

소프트웨어 라이선스 계약 조항을 승인할 수 있도록 자동 모드에서 설치를 실행시키십시오. 이 계약은 Windows 설치 이미지의 **license** 서브디렉토리에서 찾을 수 있습니다.

-s 옵션은 명령행에서 마지막에 있어야 하며 지정할 도메인 이름(**-i** 옵션) 및 기타 매개변수를 지정했는지 확인해야 합니다. 자동 모드에서 설치를 실행할 경우 매개변수에 대해 프롬프트를 표시하지 않거나 확인 메시지를 표시하지 않습니다. **-s** 뒤의 명령행 인수는 무시됩니다.

-f2 자동 모드 설치 중에 작성된 설치 로그 파일의 전체 경로 이름을 지정하십시오 (**-s** 옵션 사용).

이 옵션을 지정하지 않으면 설치 프로그램이 실행된 디렉토리에 **setup.log** 파일이 작성됩니다. CD 드라이브에서 자동 모드로 설치할 경우 파일이 해당 컴퓨터에 작성되도록 이 옵션을 지정해야 합니다(CD 드라이브에 작성될 수 없기 때문).

-k폴더 프로그램 폴더를 지정하십시오.

-p디렉토리

설치 디렉토리를 지정하십시오.

-i도메인

이 클라이언트의 도메인 이름을 지정하십시오. 이 매개변수는 필수이며 디폴트가 없습니다.

-w디렉토리

소스가 디스크 또는 CD에 있는 경우 Linux용 Communications Server 클라이언트 소프트웨어 파일이 포함된 소스 디렉토리를 지정하십시오. 그렇지 않으면 **-v** 옵션을 사용하십시오.

-v서버 클라이언트 소프트웨어 파일이 다운로드되는 서버를 지정하십시오. 서버 이름 또는 TCP/IP 주소를 지정할 수 있습니다. 디스크 또는 CD에서 소스 파일을 복사할 경우 **-v** 옵션 대신 **-w** 옵션을 사용하십시오.

-l서버 이 클라이언트가 액세스할 수 있는 서버 목록에 포함될 서버를 지정하십시오. 클라이언트가 서버와 동일한 사설 네트워크에 있고 TCP/IP를 사용하여 액세스되는 경우 각 서버는 단지 서버 이름으로 식별됩니다.

클라이언트가 HTTPS를 사용하여 서버에 액세스하는 경우 HTTPS 지원을 제공하는 WebSphere 서버 이름 및 Linux용 Communications Server의 이름을 다음 형식으로 지정하여 각 서버를 식별해야 합니다.

webservername : servername1

이 경우 WebSphere가 HTTPS 연결을 위해 기본 포트 443을 사용하도록 설정되어 있다고 가정합니다. 네트워크 관리자가 다른 포트 번호를 사용하도록 WebSphere를 구성한 경우 다음 형식으로 포트 번호를 포함하십시오.

webservername : portnumber : servername1

HTTPS 연결을 지원하도록 WebSphere를 구성하는 데 대한 자세한 내용은 32 페이지의 『WebSphere Application Server 구성』을 참조하십시오.

- o 기존 .DLL 파일을 겹쳐쓰십시오. 원격 API 클라이언트 .DLL 파일이 이미 존재하는 경우 설치 프로그램은 이들 파일이 설치 프로그램 .DLL 파일의 버전 번호보다 큰 경우에도 이들 파일을 겹쳐씁니다.
- y 기존 .DLL 파일을 저장하십시오. 원격 API 클라이언트 .DLL 파일이 필요한 디렉토리에 이미 존재하는 경우 설치 프로그램은 기존 .DLL 파일을 설치 디렉토리의 서브디렉토리로 복사한 다음 원격 API 클라이언트 .DLL 파일을 설치합니다. 서브디렉토리의 사본은 원격 API 클라이언트 클라이언트 소프트웨어를 설치 제거하는 경우 설치 제거 프로세스가 완료되도록 하기 위한 것입니다.
- n 기존 원격 API 클라이언트 .DLL 파일을 찾은 경우 설치를 취소하십시오.

-a 시간종료

LAN 액세스 시간종료를 초 단위로 지정하십시오. 이 시간은 서버에 대한 클라이언트 연결이 닫히기 전까지 대기 상태로 남아 있을 수 있는 시간입니다. 값 0은 시간종료가 없음을 표시합니다.

-b 최대 브로드캐스트

UDP 브로드캐스트의 최대 시도 횟수를 지정하십시오. UDP 브로드캐스트는 특정 서버가 아닌 도메인의 임의 서버에 클라이언트가 연결을 시도하는 것입니다. 값 0은 브로드캐스트 시도가 없음을 표시합니다.

-j 재연결 시간종료

서버 종료 이후 서버에 재연결을 시도하기 전 클라이언트가 대기하는 시간을 초 단위로 지정하십시오.

GSKIT 소프트웨어 설치 및 보안 인증서 설정

주: Windows의 x64 원격 API 클라이언트의 초기 릴리스에서는 서버에 대한 HTTPS 연결을 지원하지 않았습니다. 이 기능은 자세한 설치 및 구성 지시사항과 함께 PTF에서 제공될 것입니다. 아래 지시사항은 32비트 클라이언트에만 적용됩니다.

IBM 원격 API 클라이언트가 HTTPS를 사용하여 서버에 연결하는 경우 클라이언트에 GSKIT 소프트웨어를 설치하여 보안 인증서를 관리해야 합니다. 다음 단계를 수행하십시오.

1. Windows 컴퓨터에 설치 파일을 보관할 임시 디렉토리를 작성하십시오.

2. 32비트 클라이언트의 경우:

- **gsk7bas.exe** 파일을 설치 CD의 **/ibm-commserver-clients/windows** 디렉토리에서 임시 디렉토리로 복사하십시오.
- 명령 창에서 임시 디렉토리로 변경하십시오.
- 다음 명령을 실행하여 설치 파일의 압축을 푸십시오.

```
gsk7bas tempdir /D
```

*tempdir*은 임시 디렉토리의 전체 경로입니다.

- 다음 명령을 실행하여 설치 프로그램을 시작하십시오.

```
setup sxclient
```

3. 화면 지시사항에 따라 GSKIT 소프트웨어를 설치하십시오.

4. 설치가 끝나면 첫 단계에서 작성한 임시 디렉토리 및 포함된 항목을 제거하십시오.

이제 다음 절에 설명된 것처럼 보안 인증서를 설정할 수 있습니다.

32비트 클라이언트에서 자동 설치를 수행해야 하는 경우(여러 클라이언트 컴퓨터에 GSKIT 소프트웨어를 설치하려고 하는데 일괄처리 파일을 사용하여 이 프로세스를 자동화하려는 경우) 다음 명령을 실행하여 설치 프로그램을 자동 모드로 실행하십시오.

```
setup gskit -s -f1 "tempdir\SETUP.ISS"
```

*tempdir*은 첫 단계에서 작성한 임시 디렉토리의 전체 경로입니다.

이 명령은 설치 프로그램의 사용자 인터페이스를 표시하지 않고 소프트웨어를 설치하며 사용자 입력을 요구하지 않습니다. 설치에 대한 디폴트는 **SETUP.ISS** 파일에서 가져옵니다. 이 파일은 설치 프로세스 시작 과정에서 임시 디렉토리로 압축을 푼 파일 중 하나입니다. GSKIT 소프트웨어를 설치할 디렉토리를 변경해야 하는 경우 임시 디렉토리에 파일의 압축을 푼 다음 텍스트 편집기를 사용하여 **SETUP.ISS**를 편집하여 이 작업을 수행할 수 있습니다. 그러면, 갱신된 파일을 포함하여 이 디렉토리의 파일을, 매번 파일의 압축을 풀고 설치 디렉토리를 변경하지 않아도, 다른 클라이언트 컴퓨터에 설치할 때도 사용할 수 있습니다.

GSKIT를 사용하여 HTTPS 보안 인증서 설정

IBM 원격 API 클라이언트가 HTTPS를 사용하여 서버에 연결하기 전에 클라이언트에서 GSKIT 키 관리자 프로그램을 사용하여 보안 인증서 구성을 설정해야 합니다. 다음 단계를 수행하십시오.

1. GSKIT 키 관리자 프로그램인 `installdir\snakeyman.exe`를 실행하십시오. `installdir`은 클라이언트 소프트웨어를 설치한 디렉토리를 나타냅니다. 이 디렉토리는 클라이언트 설치 과정에서 다른 위치를 지정하지 않는 한 `C:\IBMCS\w32cli`(32비트 클라이언트)입니다.

키 관리자 사용자 인터페이스 내에서 CMS 형식으로 된 키 데이터베이스 파일 `installdir\ibmcs.kdb`를 여십시오.

2. 키 데이터베이스의 초기 암호는 `ibmcs`입니다. 보안 인증서를 설정하기 전에 이 암호를 반드시 변경하여 구성 보안을 유지해야 합니다. 암호 변경 대화상자에서 ‘암호를 파일로 보관’ 선택란에 표시하여 클라이언트가 키 데이터베이스를 열 수 있도록 새 암호를 저장해야 합니다.
3. Web Server의 보안 인증서에 서명하기 위해 사용된 인증 기관(CA) 인증서 사본을 얻어 키 데이터베이스에 설치하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 서명자 인증서를 선택하고 추가를 누르십시오.
4. WebSphere 서버가 클라이언트 보안 인증서를 요구하도록 구성된 경우 클라이언트는 Web Server의 보안 인증서 데이터베이스에 자체 인증서가 있는 CA가 발행한 인증서를 갖고 있어야 합니다. 새 인증서를 요청하려면 다음과 같이 하십시오.
 - a. 키 관리자 사용자 인터페이스에서 작성, 새 인증서 요청을 선택하고 요구되는 세부사항을 채우십시오.
 - b. 인증서를 저장하고 이를 파일로 발췌하여 CA에 보내십시오.
 - c. 인증서가 발행되면 Web Server의 데이터베이스에 저장하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 개인용 인증서를 선택하고 수신을 누르십시오.

자체 내부 테스트를 위한 임시 조치로, CA에서 인증서를 가져오지 않고 스스로 서명한 클라이언트 인증서를 작성할 수 있습니다. 그러나, 이 방법은 필요한 레벨의 보안을 보장하지 못하므로 실제 시스템에서는 이 방법을 사용해서는 안 됩니다. 스스로 서명한 인증서를 작성하려면 다음과 같이 하십시오.

- a. 키 관리자 사용자 인터페이스에서 작성, 자체 서명한 인증서 새로 작성을 선택하고 요구되는 세부사항을 채우십시오.
 - b. 인증서를 저장하고 파일로 발췌하십시오.
 - c. 인증서 파일을 Web Server의 데이터베이스에 저장하십시오. 이를 수행하려면 키 관리자 사용자 인터페이스에서 개인용 인증서를 선택하고 수신을 누르십시오.
5. 구성 인증서를 완성했으면 GSKIT 키 관리자를 종료하십시오.

설치 후 원격 API 클라이언트 소프트웨어 사용자 조정

Linux용 Communications Server 프로그램 그룹에 있는 구성 유틸리티 프로그램을 실행하여 초기 설치 후 언제든지 사용자 정의된 설정을 변경할 수 있습니다. 프로그램이 초기 설치 프로세스에 표시되었던 동일한 구성 창을 표시합니다. 63 페이지의 『설치 프로그램을 사용하여 Windows에 원격 API 클라이언트 설치』의 프로시저를 준수하여 필드의 정보를 변경할 수 있습니다.

초기 설치 중에 SDK 파일을 설치하지 않아 지금 추가하려는 경우 설치 프로그램을 다시 실행하면 이를 수행할 수 있습니다. 설치 유형으로 사용자 정의를 선택하고 SDK 패키지를 선택하십시오.

원격 API 클라이언트 소프트웨어 재설치

예를 들어 소프트웨어 업그레이드가 필요한 경우 언제든지 원격 API 클라이언트 소프트웨어를 재설치할 수 있습니다.

이를 수행하려면 이전과 같이 63 페이지의 『설치 프로그램을 사용하여 Windows에 원격 API 클라이언트 설치』 또는 67 페이지의 『명령행에서 원격 API 클라이언트 소프트웨어 설치』의 지시사항을 사용하여 설치 프로그램을 실행하십시오. 설치 프로그램이 초기 설치 중에 클라이언트 소프트웨어 파일이 복사되었던 위치를 표시합니다. 확인을 눌러 동일한 이 위치에서 파일의 새 사본을 얻으십시오. 확인을 누르면 설치 프로그램이 파일을 복사하고 옵션 화면으로 리턴합니다.

주: 자동 모드에서 원격 API 클라이언트 소프트웨어를 재설치할 경우(67 페이지의 『명령행에서 원격 API 클라이언트 소프트웨어 설치』에서 설명) 컴퓨터를 다시 시작하여 설치를 완료해야 할 수 있습니다. 이는 일부 프로그램 파일이 설치 프로세스 과정에서 사용 중이어서(예: 원격 API 클라이언트가 실행 중인 경우) 새 파일로 대체될 수 없기 때문입니다. 이 경우 새 파일이 임시 디렉토리로 복사되며 컴퓨터가 다시 시작될 때 제 위치로 자동 이동됩니다.

컴퓨터를 다시 시작해야 하는지 확인하려면 설치 프로세스가 완료되었을 때 텍스트 편집기(예: 메모장)를 사용하여 설치 로그 파일을 보십시오. 설치 로그 파일은 **setup.log**이며 **-f2** 명령행 옵션을 사용하여 다른 경로 및 파일 이름을 지정하지 않으면 설치 프로그램이 실행되는 디렉토리에 작성됩니다.

파일 끝의 응답 결과라는 표제 아래에서 응답 코드라는 텍스트 뒤에 두 개의 값(0 또는 -12) 중 하나가 와야 합니다. 값이 0인 경우 컴퓨터를 다시 시작할 필요가 없으며 값이 -12인 경우에는 컴퓨터를 다시 시작한 다음 Windows 클라이언트의 사용을 시도하십시오.

원격 API 클라이언트 소프트웨어 설치 제거

Windows 제어판에서 프로그램 추가/제거 옵션을 사용하여 언제든지 원격 API 클라이언트 소프트웨어를 설치 제거할 수 있습니다. 일단 설치 제거 프로세스가 확인되면 Windows가 다음을 수행합니다.

- 설치된 파일을 모두 삭제합니다.
- 초기 설치 중에 .DLL 파일이 서브디렉토리에 저장된 경우 파일을 원래 위치로 복원합니다.
- 서브디렉토리가 비어 있는 한 저장된 .DLL 파일이 있던 서브디렉토리를 삭제합니다.
- 비어 있는 경우 프로그램 폴더 및 작성된 디렉토리를 제거합니다.
- 설치 제거가 완료되면 모든 파일 전송 및 삭제 내용이 포함된 **setup.log** 파일을 삭제합니다.
- 설치 제거가 완료되었거나 설치의 일부가 실패하여 사용자가 **setup.log** 파일을 확인해야 한다고 메시지를 표시합니다.

종료 버튼을 사용하면 Windows로 리턴됩니다.

32비트 Windows 원격 API 클라이언트와 함께 사용하기 위해 GSKIT를 설치했지만 이 소프트웨어를 사용하는 다른 응용프로그램이 없는 경우, 『GSKIT 소프트웨어 설치 제거』에 설명된 것처럼 이 소프트웨어를 설치 제거하고자 할 수 있습니다.

GSKIT 소프트웨어 설치 제거

32비트 Windows 원격 API 클라이언트와 함께 사용하기 위해 GSKIT를 설치했지만 이 소프트웨어를 사용하는 다른 응용프로그램이 없는 경우, Windows 원격 API 클라이언트를 설치 제거한 다음 이 소프트웨어를 설치 제거하고자 할 수 있습니다. 다음 단계를 수행하십시오.

1. 명령 창에서 Windows가 설치된 디렉토리로 변경하십시오. 일반적으로 이 디렉토리는 **C:\Windows**입니다.
2. 다음 명령을 실행하십시오.

```
gskbui sxclient
```

GSKIT를 사용하는 다른 응용프로그램이 없는 경우 Windows에서 파일을 제거합니다. GSKIT를 사용하는 응용프로그램이 존재하는 경우 다른 응용프로그램이 설치 제거될 때까지 파일은 설치된 채로 남아 있습니다.

도움말

F1 키를 눌러 언제든지 도움말에 액세스할 수 있습니다. 구성 및 고급 옵션 창에도 각각 도움말 버튼이 있습니다.

제 8 장 Linux용 Communications Server 구성 및 사용

Linux용 Communications Server 구성을 정의 및 수정하는 가장 쉬운 방법은 Motif 관리 프로그램(`xsnadmin`)을 사용하는 것입니다. 이 프로그램은 사용자가 로컬 노드 상에서 SNA 자원을 열람하고 관리할 수 있는 그래픽 사용자 인터페이스를 제공합니다. 또한 명령행 관리와 같은 다른 관리 도구를 사용할 수 있지만 Motif 프로그램을 권장합니다.

Motif 관리 프로그램에는 SNA 및 Linux용 Communications Server에 대한 개요 정보, Linux용 Communications Server 대화상자에 대한 참조 정보, 그리고 특정 타스크 수행을 위한 지침을 제공하는 도움말 화면이 포함되어 있습니다. 각각의 타스크(예: 노드 구성)나 통신 유형(예: TN3270이나 APPC)에 대해, 프로그램이 필요한 자원의 구성 설정으로 안내합니다.

Motif 관리 프로그램을 사용하면 표준 Linux용 Communications Server 구성에 필요한 모든 필수 매개변수를 설정할 수 있습니다. 고급 매개변수들에 있어서는, Motif 관리 프로그램이 디폴트 값을 제공합니다. 사용자는 단지 필수 구성 정보만을 제공하면 되는데, 이는 SNA 통신을 신속하고 쉽게 설정할 수 있게 합니다.

또한 Motif 관리 프로그램을 사용하여 실행 중인 Linux용 Communications Server 시스템을 관리할 수 있습니다. 관리 프로그램을 사용하면 Linux용 Communications Server가 활동 중인 동안에 구성을 변경하고 적용할 수 있으며, 노드 자원에 대한 상태 정보에 쉽게 액세스할 수 있습니다.

Motif 관리 프로그램은 Linux용 Communications Server 자원에 대한 상태 정보를 자동으로 표시합니다. 이 정보 대부분이 노드 창에 나타납니다(78 페이지의 『Motif 관리 프로그램으로 Linux용 Communications Server 관리』 참고). 또한 노드 창의 시작 및 종료 버튼을 사용하여 노드와 링크 스테이션과 같은 특정 자원을 제어할 수 있습니다. 다른 자원을 항상 자동으로 시작 및 종료되며, 그래서 그것을 수동으로 제어할 필요가 없습니다.

주:

1. Linux용 Communications Server 자원을 정의하거나 수정하려면 로그인 그룹 `sna`의 구성원이어야 합니다.
2. Motif 관리 프로그램을 사용하려면 X 터미널이 필요합니다.
3. Motif 관리 프로그램의 창에 나타나는 버튼과 아이콘을 포함하여 사용자 인터페이스에 대한 자세한 내용은 이 프로그램의 도움말 화면 또는 *Communications Server for Linux Administration Guide*를 참조하십시오.

4. Motif 관리 프로그램에 있는 창과 대화 상자는 특정 대화 상자에서 선택한 사항에 따라 이 안내서에 나와 있는 내용과 다를 수 있습니다.

명령행 관리 및 NOF 응용프로그램을 비롯한 기타 Linux용 Communications Server 관리 도구에 관한 자세한 내용은 *Communications Server for Linux Administration Guide*, *Communications Server for Linux Administration Command Reference* 또는 *Communications Server for Linux NOF Programmer's Guide*를 참조하십시오.

Linux용 Communications Server 구성 계획

구성 변경을 취하기 전에 철저히 계획하는 것이 중요합니다. 사용자가 취하는 변경이 로컬 노드 내의 사용자들뿐만 아니라 전체 네트워크에 걸쳐 있는 사용자들에게 방해를 야기할 수 있습니다.

네트워크의 토폴로지(topology)에 취하고 있는 어떤 변경이든지 도표를 그리는 것이 유용하다는 걸 알게될 것입니다. 다른 노드로의 연결을 추가하거나 제거하고 있다면, 사용자의 노드와 다른 노드들을 보여주는 그림을 그리십시오. Motif 관리 프로그램을 사용하여 모든 기존 연결에 대한 구성 정보를 수집하고 그 정보를 도표에 추가할 수 있습니다.

도표에 새 자원들을 추가하면, 그것들이 기존 것과 중복되는지 혹은 이름이 충돌하지 않는지 살펴보기가 용이합니다. 마찬가지로, 도표는 어느 자원을 제거할 필요가 있는지 결정하는 것을 도와주고 필수 자원을 삭제하는 것을 피할 수 있게 도와줍니다.

두 개 이상의 노드로 클라이언트/서버 Linux용 Communications Server 시스템을 구성하는 경우 모든 Linux용 Communications Server 노드 및 연결 자원이 해당 다이어그램에 들어 있는지 확인하십시오. 그러면 이 장에 설명된 대로 독립형 노드를 구성하는 방법과 같이 각 노드를 차례로 구성할 수 있습니다.

일단 취할 필요가 있는 변경을 판별하면, 필요한 구성 정보를 수집하십시오. 특정 Linux용 Communications Server 기능을 위한 구성 정보를 수집하는 데 대한 안내를 받으려면, Motif 관리 프로그램의 온라인 도움말에서 제공하는 타스크 시트 또는 *Communications Server for Linux Administration Guide*에서 제공하는 계획 워크시트를 사용할 수 있습니다.

이 장에서는 Linux용 Communications Server에서 가장 자주 사용되는 기능을 구성하기 위한 지시사항을 제공합니다. 각각의 구성 타스크에 대해, 이 안내서는 자원 구성에 앞서 모을 필요가 있는 정보의 주석을 달기도 합니다.

주: 이 안내서는 Linux용 Communications Server 대화상자에 입력해야 하는 구성 정보에 대해 자세히 설명하지는 않습니다. 특정 대화상자의 필드에 대한 자세한 정보는 Motif 관리 프로그램의 해당 대화상자의 온라인 도움말을 참조하십시오.

계획 워크시트

Linux용 Communications Server의 자원 구성을 시작하기 전에 새 자원에 대한 모든 구성 정보를 수집하십시오. 지원해야 하는 특정 기능이나 응용프로그램에 대한 모든 정보를 기록하려면 *Communications Server for Linux Administration Guide*에 제공된 계획 워크시트를 사용하십시오.

아마도 네트워크 관리자, 호스트 관리자, 어플리케이션 프로그래머, 그리고 최종 사용자와 같은 여러 소스로부터 구성 정보를 모을 필요가 있을 것입니다.

다른 노드에 연결하려 하고 있다면, 그 노드에 있는 관리자가 주요 연락처입니다. 노드를 위한 관리자는 그 노드 상의 모든 자원의 이름, 주소 및 특성을 사용자에게 알려줄 수 있습니다. 종종, 대응 구성 매개변수들이 로컬 노드와 원격 노드에서 입력되는 지 확인할 필요가 있을 것입니다.

타스크 시트

Motif 관리 프로그램에 있는 온라인 도움말 화면은 특정 구성 타스크를 위한 지시사항을 제공하는 타스크 시트를 포함하고 있습니다. 타스크 시트는 구성 정보를 입력하기 위해 사용할 대화상자에 대한 모든 도움말 화면으로의 포인터를 담고 있습니다. 이를 사용하여 도움말을 열람하고 정확히 어떤 데이터를 수집해야 하는지 볼 수 있습니다.

타스크 시트는 또한 구성 정보를 입력하기 위해 입력해야 하는 각각의 개별 창과 대화상자를 위한 좀 더 상세한 도움말을 참조합니다. 그러한 도움말 화면들은 기입해야 하거나 선택해야 하는 각 필드를 설명합니다.

Motif 관리 프로그램의 사용

Motif 관리 프로그램을 사용하기 전에 시스템이 실행 프로그램을 찾을 수 있도록 **.login**이나 **.profile** 파일에 경로 정보를 추가할 필요가 있을 수 있습니다(『Linux용 Communications Server 프로그램의 경로 지정』참고). 또한 관리 프로그램을 사용하려면 먼저 Linux용 Communications Server 소프트웨어를 사용 가능하게 설정해야 합니다(78 페이지의 『Linux용 Communications Server 사용 가능화』 참조).

Motif 관리 프로그램 호출에 대한 정보와 이 프로그램 사용의 개요에 대해서는 78 페이지의 『Motif 관리 프로그램으로 Linux용 Communications Server 관리』를 참고하십시오.

Linux용 Communications Server 프로그램의 경로 지정

Linux용 Communications Server 프로그램을 실행하려면 Linux용 Communications Server 실행 가능 프로그램이 포함된 디렉토리의 경로를 지정해야 합니다. 처음으로 프로그램을 실행하기 전에 PATH 환경 변수에 디렉토리를 추가하거나 프로그램 실행시마다 디렉토리를 포함하여 경로를 지정할 수 있습니다.

Motif 관리 프로그램 사용

Motif 관리 프로그램은 `/opt/ibm/sna/bin/X11` 디렉토리에 저장되며 기타 프로그램은 `/opt/ibm/sna/bin` 디렉토리에 저장됩니다. 이러한 디렉토리를 `.login`이나 `.profile` 과 일의 `PATH` 환경 변수 정의에 추가하면 Linux용 Communications Server는 자동으로 프로그램을 찾습니다. 대안으로, 다음 보기에서 처럼 프로그램을 수행할 때 디렉토리명을 지정할 수 있습니다.

```
/opt/ibm/sna/bin/sna start
```

```
/opt/ibm/sna/bin/X11/xsnaadmin
```

이 메뉴얼에서 보여준 샘플 명령행은 `PATH` 환경 변수에 디렉토리를 추가했다고 가정하며, 디렉토리명을 포함하지 않습니다.

Linux용 Communications Server 사용 가능화

로컬 노드를 구성하거나 관리하기 전에 로컬 시스템에서 Linux용 Communications Server를 사용 가능하게 해야 합니다. X/Motif 어플리케이션에 있어서는, `DISPLAY` 환경 변수를 알맞은 X 서버를 지시하도록 설정할 필요가 있습니다.

Linux용 Communications Server를 사용 가능하게 하려면 Linux 명령 프롬프트에서 다음 명령을 입력하십시오.

```
sna start
```

Linux용 Communications Server를 설치할 때 설치 유틸리티는 `sna start` 명령이 포함되도록 자동으로 시작 파일 `/etc/rc.d/init.d/snastart`를 갱신합니다. 이렇게 함으로써 시스템이 시작될 때 Linux용 Communications Server가 자동으로 시작됩니다. Linux용 Communications Server가 자동으로 시작되지 않도록 하려면 이 행을 제거하거나 주석으로 표시한 다음 이 절의 지시사항에 따라 수동으로 Linux용 Communications Server 소프트웨어를 사용 가능하게 하십시오.

Linux용 Communications Server는 초기화 중임을 나타내거나 초기화의 성공 여부를 나타내는 메시지를 표준 오류(보통 단말기의 화면)에 기록합니다.

Motif 관리 프로그램으로 Linux용 Communications Server 관리

Linux용 Communications Server에 대해 Motif 관리 프로그램을 사용하려면 먼저 『Linux용 Communications Server 사용 가능화』에 설명된 것처럼 Linux용 Communications Server가 초기화되었는지 확인해야 합니다. (또한 적절한 X 서버를 나타내도록 `DISPLAY` 환경 변수를 설정할 필요가 있을 것입니다.)

백그라운드에서 Motif 관리 프로그램을 시작하려면 다음 명령을 발행하십시오.

```
xsnaadmin &
```


Linux용 Communications Server는 도메인 창을 표시합니다. 이 창에서 정의된 모든 노드를 표시하고 노드를 시작 및 중지할 수 있습니다. 모드를 두 번 누르면 80 페이지의 그림 5에 표시된 대로 해당 노드의 노드 창이 나타납니다.

노드 창은 노드 및 그 자원에 대한 정보를 보여줍니다. 아직 노드를 구성하지 않았으면 관리 프로그램이 84 페이지의 『노드 구성』에 설명된 것처럼 노드를 구성하라는 메시지를 표시합니다.

주: 이 안내서에서는 창이라는 용어를 사용하여 Linux용 Communications Server 자원에 대한 정보를 표시하는 Motif 창을 설명합니다. 창에는 하나 혹은 그 이상의 섹션 또는 연결 창이 있을 수 있습니다. 대화상자는 정보를 입력할 수 있는 Motif 창입니다.

노드 창은 사용자가 필요로 하는 대부분의 정보를 보여주며, 그 밖의 모든 것을 쉽게 액세스할 수 있게 합니다. 로컬 노드 상의 모든 주요 자원들을 보여줍니다.

두 개 이상의 노드로 클라이언트/서버 Linux용 Communications Server 시스템을 구성하는 경우 이 창의 지시사항에 따라 각 노드를 차례로 구성하십시오(다음 노드를 선택하려면 도메인 창으로 리턴).

노드 창의 창 메뉴로부터 다른 창으로 갈 수 있습니다. 이들 창에는 다음이 포함됩니다.

- LU 풀 사용
- CPI-C 목적지 이름 창

노드 창의 서비스 메뉴는 자원을 추가하는 빠른 방법과 구성 및 관리 task에 대한 도움말을 제공합니다. 진단 메뉴를 사용하면 로깅 대화 상자와 추적 대화 상자로 이동할 수 있습니다.

노드 창

샘플 노드 창은 80 페이지의 그림 5에 나타납니다. 제목 표시줄에는 Linux 시스템의 이름이 표시됩니다.

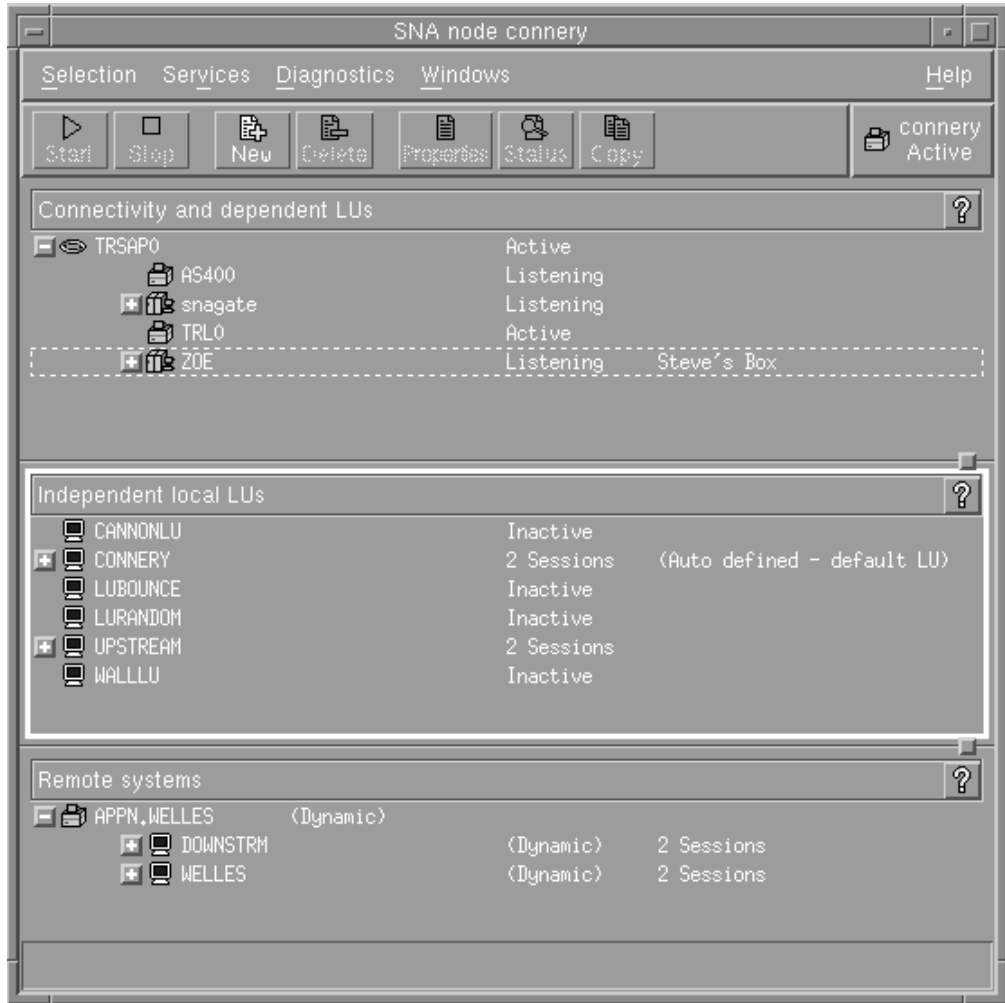


그림 5. 노드 창

노드 창에서 Linux용 Communications Server 노드의 모든 자원 및 구성요소를 구성 및 관리할 수 있습니다.

- 포트
- 링크 스테이션
- 유형 0-3의 LU 및 유형 6.2의 종속 LU
- DLUR 내부 PU
- 독립 로컬 LU
- 원격 노드
- 상대방 LU

노드 창으로부터 이러한 자원 모두를 추가, 삭제, 수정 및 관리할 수 있습니다. 창에서 자원들의 배열은 자원들간의 관계를 보여주며 사용자로 하여금 어느 자원들이 표시되는지를 제어할 수 있게 합니다.

포트, 로컬 LU, 그리고 원격 노드들이 항상 표시됩니다. 노드 창은 상대 포트 밑에 각각의 링크 스테이션과, 상위 링크 스테이션 밑에 각각의 종속 LU를 보여줍니다. 또한 로컬 LU들 밑에 그리고 원격 노드 밑에 상대방 LU들을 보여줍니다.

노드 창은 노드를 위한 서로 다른 유형의 자원들을 위한 별개의 섹션들을 포함합니다.

- 노드 창의 맨 위 오른쪽 구석에 있는 노드 상자는 노드가 활동 중 또는 활동 중 이 아님 상태인지를 지시합니다.
- 노드 창의 상단 분할창(연결 분할창)에는 포트, 각 포트 상의 링크 스테이션이나 PU 그리고 특정 링크 스테이션이나 PU 상의 종속 LU를 포함하여 노드에 대한 연결 자원이 나열되어 있습니다. 각각의 자원에 대해, 이 창이 현재 상태 정보를 보여줍니다.
- 중간 분할창(종속 로컬 LU 부분)은 로컬 노드에 정의된 독립 LU들을 나타냅니다. 이 창에는 특정 LU를 사용하는 세션에 관한 정보와 여기에 액세스하는데 사용되는 링크 스테이션으로 상대방 LU의 위치를 정의해주는 레코드들이 나타납니다.
- 하단 분할창(원격 시스템 부분)은 원격 노드 및 상대방 LU들에 관한 정보를 보여줍니다. 각 원격 노드나 상대방 LU를 위한 세션 정보 역시 보여줍니다.

분할창을 눌러 이러한 분할창 중의 어느 것이든 선택할 수 있습니다. 자원을 위한 라인을 눌러 분할창 내의 특정 자원을 선택할 수도 있습니다. 항목을 위한 구성을 보거나 수정하려면, 항목을 두 번 누르십시오. (이 창의 버튼 및 메뉴를 사용하여 특정 자원을 위한 구성 정보를 액세스할 수 있습니다.)

나열된 각각의 항목에 대해, 그 항목에 속하는 자원이 그 항목을 위한 정보에 내포됩니다. 예를 들어, 링크 스테이션은 그들이 속하는 포트 아래에 그룹으로 묶입니다. 항목 옆의 확장 버튼(+)을 눌러 현재 표시되어 있지 않은 해당 항목의 자원을 표시할 수 있으며 축소 버튼(-)을 누르면 항목의 자원을 숨길 수 있습니다.

노드 창으로부터 다음의 관리 작업을 수행할 수 있습니다.

자원 시작 또는 종료

자원을 선택하고 시작 또는 종료 버튼을 누르십시오. (또는 선택 메뉴에서 항목 시작이나 항목 종료를 선택할 수 있습니다.)

항목을 위한 자원 추가

항목을 선택하고 추가 버튼을 누르십시오. (또는 선택 메뉴에서 새 자원을 선택하십시오.) 예를 들어, 포트에 대한 링크 스테이션을 추가하려면 포트를 선택하고 새 포트 버튼을 누르십시오.

자원 삭제

자원을 선택하고 삭제 버튼을 누르십시오. (또는 선택 메뉴에서 삭제를 선택하십시오.)

임의의 자원에 대한 구성 보기 또는 수정

자원을 선택하고 등록정보 버튼을 누르십시오. (또는 선택 메뉴에서 등록정보를 선택하십시오.)

임의의 자원에 대한 상태 정보 확보

자원을 선택하고 상태 버튼을 누르십시오. (또는 선택 메뉴에서 상태를 선택하십시오.)

임의의 자원에 대한 구성 복사

자원을 선택하고 복사 버튼을 누르십시오. (또는 선택 메뉴에서 복사를 선택하십시오.)

또한 서비스 메뉴에서 노드에 대한 특정 구성 타스크를 선택하고 진단 메뉴에서 로깅 (도메인의 경우)과 추적(노드의 경우)을 제어하고 창 메뉴의 항목 중 하나를 선택하여 도메인 자원을 보거나 수정할 수 있습니다.

자원 항목

창에서 자원들의 배열은 자원들간의 관계를 보여줍니다.

항목에 관련된 하위 항목이 하나 이상 있으면, 확장 기호(+) 또는 축소 기호(-)가 그 옆에 나타납니다.

- 확대 기호는 관련된 하위 항목이 숨겨져 있음을 나타냅니다. 확장 기호를 누르거나, 숫자 키패드에 있는 + 키를 눌러 항목을 표시할 수 있습니다.
- 축소 기호는 하위 항목이 표시되어 있음을 나타냅니다. 축소 기호를 누르거나 숫자 키패드에 있는 - 키를 눌러 항목을 감출 수 있습니다.
- 항목 옆에 기호가 없으면, 항목에 연관된 하위 자원이 없는 것입니다.

예를 들어, 링크 스테이션이 특정 포트와 연관됩니다. 노드 창의 연결 분할창에서, 그 포트와 연관된 모든 다른 링크 스테이션들과 함께 상위 포트 아래 링크 스테이션이 표시됩니다. 포트는 항상 표시되나, 연관된 링크 스테이션의 목록이 표시되는지 은닉되는지를 선택할 수 있습니다. 유사하게, 연관된 LU들의 목록을 가진 링크 스테이션들이 확대되어 LU들을 표시할 수 있으며, 또는 축소되어 그들을 숨길 수도 있습니다.

부모(Parent) 자원은 항상 자(Child) 자원에 앞서 구성되어야 하며, 부모(Parent) 자원을 삭제하면 모든 자(Child) 자원들도 삭제됩니다.

도구 모음 버튼

자원 창들은 공통의 기능들을 쉽게 수행할 수 있게 하는 도구 모음 버튼들을 포함합니다. Linux용 Communications Server의 도구 모음이 83 페이지의 그림 6에 나타나 있습니다.



그림 6. Linux용 Communications Server 도구 모음

각 자원 창의 도구 모음에 모든 버튼들이 나타나는 것은 아닙니다. 버튼의 조작이 현재 선택된 항목에 유효하지 않으면(또는 항목이 선택되어야 하는 조작이 있을 때 선택된 항목이 없는 경우) 버튼의 외곽이 회색으로 표시되고, 기능을 선택할 수 없습니다. (버튼을 눌러도 영향을 미치지 않습니다.) 다음 버튼들이 자원 창 위에 나타날 수 있습니다.

시작 선택된 항목을 시작합니다.

종료 선택된 항목을 종료합니다.

추가 새 자원 항목을 추가합니다.

삭제 선택된 자원을 삭제합니다.

특성 선택된 항목에 대한 대화상자를 열어 항목의 구성을 보거나 수정합니다.

상태 선택된 항목의 현재 상태를 표시합니다.

복사 선택된 항목을 복사합니다. 이 버튼을 누르면 포함되어 있는 필드가 선택된 항목의 구성을 복제해주는 대화 상자가 열립니다. 대화상자의 필드를 완료하여(새 항목의 이름을 기입) 새 자원을 추가합니다.

포트 및 링크 스테이션과 같은 많은 자원들이 활동 중인 동안 수정될 수 있습니다. 그러나 자원을 선택하고 특성 버튼을 눌러 대화 상자를 열어 활동 중인 자원의 매개변수를 볼 수 있습니다. 작업을 완료하면 닫기 버튼을 누르십시오.

클라이언트/서버 기능 구성

이 절은 클라이언트/서버 환경에서 실행하기 위해 Linux용 Communications Server를 설치한 경우(동일한 네트워크에 Linux용 Communications Server 노드가 여러 개 있는 상태)에만 관련됩니다.

클라이언트/서버 환경에서 서버를 구성 서버로 표시할 수 있습니다. Linux용 Communications Server는 이 구성 서버 목록을 관리합니다. 첫 번째로 나열된 서버가 마스터 서버이며 나열된 다른 모든 서버는 백업 서버입니다. 서버가 순서대로 나열되어 있기 때문에 마스터 서버를 사용할 수 없는 경우 두 번째 서버(첫 번째 백업 서버)가 마스터 서버 역할을 수행하고 마스터 또는 첫 번째 백업 서버를 사용할 수 없는 경우 세 번째 서버(두 번째 백업 서버)가 마스터 서버 역할을 수행하는 등 순서대로 해당 역할을 수행합니다.

도메인의 노드가 활동 중인 경우 도메인에서 첫 번째로 사용 가능한 구성 서버(접속이 가능하고 Linux용 Communications Server 소프트웨어가 실행 중인 첫 번째 서버)가

마스터 서버가 됩니다. 현재 마스터를 사용할 수 없는 경우(네트워크 장애로 접속이 불가능하거나 실행 중인 SNA 소프트웨어가 중지된 경우) 목록에서 사용 가능한 다음 구성 서버가 새 마스터가 됩니다.

Linux용 Communications Server는 마스터 없이 실행될 수 있습니다. 이것은 구성 서버 목록에 접속 가능한 서버가 없는 경우에 해당됩니다. 이 경우 접속 가능한 서버에서만 노드 자원을 보고 구성할 수 있습니다.

주: 마스터 서버로 역할을 수행하는 노드를 직접 표시할 수 없습니다. 마스터 서버는 노드가 구성 서버 목록에 추가되는 순서에 따라 선택됩니다. 서버를 목록의 맨 위로 이동시킬 경우 목록에서 다른 모든 노드를 제거한 다음 다시 추가하십시오.

Motif 관리 프로그램의 도메인 창에서 선택 메뉴로부터 구성 서버 작성을 선택하면 구성 서버를 추가할 수 있습니다. 서버가 목록 끝에 추가되며 다른 모든 구성 서버를 사용할 수 없는 경우에만 마스터 서버가 됩니다. 서버를 제거하려면 선택 메뉴에서 구성 서버 제거를 선택하십시오.

주: Linux용 Communications Server 소프트웨어가 실행 중인 서버가 나열된 유일한 서버인 경우 마스터 서버 역할을 대신할 수 있는 다른 서버가 없으므로 이 서버를 삭제할 수 없습니다. 클라이언트/서버 구성에서는 사용 가능한 마스터 서버가 적어도 하나는 있어야 합니다.

클라이언트/서버 Linux용 Communications Server 시스템의 구성 및 관리에 대한 자세한 내용은 *Communications Server for Linux Administration Guide*를 참조하십시오. 이 매뉴얼은 클라이언트 및 서버를 다른 Linux용 Communications Server 도메인으로 이동하는 방법과 클라이언트 운영 세부사항을 구성하는 방법을 비롯하여 고급 클라이언트/서버 구성에 대한 정보도 제공합니다.

노드 구성

시스템에 Linux용 Communications Server를 구성하는 첫 번째 단계는 로컬 노드를 구성하는 것입니다. 노드 구성은 SNA 네트워크에서 통신하기 위해 노드가 필요로 하는 기본적인 정보를 제공합니다. 노드를 위한 연결성 및 자원을 정의할 수 있기 전에 노드를 구성해야 합니다.

노드가 이미 구성되어 있으면, 이 섹션에서 설명된 절차를 사용하여 노드 구성을 수정할 수 있습니다. 그러나 구성 변경을 취하기 전에 노드를 종료해야 합니다.

Linux용 Communications Server 노드를 APPN 네트워크 노드나 APPN 끝 노드로 구성할 수 있습니다. Linux용 Communications Server가 호스트와의 통신을 위해서만 SNA를 사용하는 경우 Linux용 Communications Server 노드를 끝 노드 또는 브랜치 네트워크 노드로 구성하려고 할 것입니다.

로컬 노드가 APPN 네트워크의 일부일 때, 노드가 다른 노드를 위한 APPN 경로지정 서비스를 제공한다면 로컬 노드를 APPN 네트워크 노드로서 구성하십시오. 다른 노드들이 경로지정 서비스를 제공한다면, 로컬 노드를 APPN 끝 노드로서 구성하십시오.

노드 구성을 시작하기 전에, 다음의 정보를 모으십시오.

- APPN 지원 유형(네트워크 노드, 브랜치 네트워크 노드, 끝 노드).
- 제어점 이름(그리고 서로 다른 경우에는 그 별명). 네트워크 계획자와 논의하여 이 이름을 판별하십시오.
- 디폴트 노드 ID(개별 통신 링크를 구성할 때 이 디폴트를 대체할 수 있습니다).

노드를 구성하려면, 노드 창으로부터 다음의 단계들을 수행하십시오.

1. 서비스 메뉴에서 노드 매개변수 구성을 선택하거나 노드 창의 맨 위 오른쪽 구석에 있는 노드 상자를 두 번 누르십시오. Linux용 Communications Server는 노드 매개변수 대화상자를 표시합니다.
2. APPN 지원 레벨, 제어점(CP) 이름과 필요할 경우 디폴트 노드 ID를 지정하십시오.
3. 확인 버튼을 눌러 노드를 정의하십시오. 노드를 정의할 때 Linux용 Communications Server는 제어점(CP)과 같은 이름으로 디폴트 LU를 자동으로 정의합니다.

입력한 값을 저장하지 않고 종료하려면 취소 버튼을 누르십시오.

연결성 구성

Linux용 Communications Server 노드가 다른 노드와 통신하려면 최소한 하나의 인접 노드로 연결을 구성해야 합니다. 연결하는 링크는 종속 트래픽, 독립 트래픽 또는 둘 다를 수행하도록 구성될 수 있습니다.

컴퓨터에 하나 혹은 그 이상의 링크 프로토콜을 위한 어댑터 카드를 설치할 수 있습니다. 연결성을 구성하기 위해 입력할 필요가 있는 정보의 많은 것들이 사용 중인 링크 프로토콜에 달려 있습니다. Linux용 Communications Server가 지원하는 링크 프로토콜의 목록은 22 페이지의 『설치 요구사항』을 참조하십시오.

링크를 구성하려면, 포트와(대부분의 경우에) 링크 스테이션을 정의할 필요가 있습니다. Motif 관리 프로그램을 사용할 때에는, DLC(데이터 링크 제어)가 포트를 위한 구성의 일부로서 자동 구성됩니다. 게다가, 포트를 연결 네트워크의 일부로서 정의하는 옵션을 가집니다.

구성할 필요가 있는 링크는 무엇을 달성하고자 하는지와 사용자의 네트워크가 APPN 네트워크인지 여부에 달려 있습니다. 필요한 정보는 링크 프로토콜과, 그리고 링크가 종속 트래픽인지, 독립 트래픽인지, 또는 둘 다인지의 여부에 따라 다릅니다.

보기로서, 이 섹션에서는 다음 유형의 링크들을 구성하는 방법을 설명합니다.

- SDLC 회선을 사용하여 호스트 시스템과의 종속 트래픽을 지원하는 링크
- 이더넷 링크 프로토콜을 사용하는 APPN 네트워크로 종속 및 독립 트래픽을 지원하는 링크. 이 보기는 또한 이더넷 포트상에 연결 네트워크를 정의합니다.
- APPN 네트워크에 대한 Enterprise Extender 링크(Enterprise 링크는 독립 트래픽만 지원합니다).

기타 링크 프로토콜에 대해서는 *Communications Server for Linux Administration Guide* 또는 Motif 관리 프로그램의 온라인 도움말을 참조하십시오.

종속 트래픽에 대한 SDLC 링크 구성

동기 데이터 링크 제어(SDLC) 포트에 대해 다음 정보가 필요합니다.

- SNA 포트 이름(일반적으로 디폴트를 사용할 수 있습니다). SDLC 장치 번호도 제공해야 합니다.
- 노드가 시작될 때 포트가 자동으로 활성화되어야 하는지의 여부.
- 회선 유형(교환식 전송, 교환식 입력 또는 전용 회선)
- 링크 역할(1차, 2차, 조정 가능, 1차 분산 분기 회선 또는 2차 다중 PU)
- 폴 주소(1차가 아닌 포트의 교환식 입력 회선인 경우만 해당). 다른 유형의 포트에 대해서는 링크 스테이션에 폴 주소를 구성해야 합니다.

SDLC 링크 스테이션의 경우 다음 추가 정보가 필요합니다.

- 활성화 방식(관리자에 의해, 노드 시동 시 또는 요구가 있을 때)
- 지원되는 트래픽 유형(이 예제에서는 종속형만 해당)
- 원격 노드 역할(이 예제에서는 호스트)

SDLC 링크를 구성하려면 노드 창에서 다음 단계를 수행하십시오.

1. 포트 구성:

- a. 창의 연결 분할 창을 선택하십시오.
- b. 서비스 메뉴의 연결성 하위 메뉴에서 새 포트를 선택하십시오. (또는 버튼 표시 줄에서 새로 작성 버튼을 누르십시오.)
- c. 그 결과 나오는 대화상자에서, 옵션 메뉴에서 프로토콜 유형을 선택하고 나서 포트를 정의하기로 선택하십시오.

확인 버튼을 누르면 Linux용 Communications Server에 SDLC 포트 대화상자가 표시됩니다.

- d. 대화상자의 필드들에 적당한 값을 입력하십시오.
- e. 확인 버튼을 눌러 포트를 정의하십시오.

노드 창의 연결 분할창에 포트가 나타납니다.

2. 포트에 링크 스테이션 정의:

- a. 링크 스테이션이 노드 창의 연결 분할창에 추가되고 있는 포트를 선택했는지 확인하십시오.
- b. 서비스 메뉴의 연결성 하위 메뉴에서 새 링크 스테이션을 선택하십시오. (또는 버튼 표시줄에서 새로 작성 버튼을 누르십시오.)
- c. 확인 버튼을 누르십시오.

Linux용 Communications Server는 SDLC 링크 스테이션 대화상자를 표시합니다.

- d. 대화상자의 필드들에 적당한 값을 입력하십시오.
- e. 확인 버튼을 눌러 링크 스테이션을 정의하십시오.

링크 스테이션은 노드 창의 연결성 분할창에서 그것이 속하는 포트 아래에 나타납니다.

종속 및 독립 트래픽을 지원하도록 이더넷 링크 구성

이 보기는 종속과 독립 트래픽 모두를 APPN 네트워크로 지원하는 이더넷 링크를 구성하는 방법을 보여줍니다. 또한, 이더넷 포트상에 연결 네트워크를 정의합니다.

이더넷 포트에 있어서는, 다음의 정보를 필요로 합니다.

- SNA 포트 이름(일반적으로 디폴트를 사용할 수 있습니다). 복수의 이더넷 네트워크 어댑터 카드가 있으면, 또한 이더넷 카드 번호도 공급할 필요가 있습니다. 또한 로컬 서비스 액세스점(SAP) 번호를 지정해야 합니다(Intel 및 OSA2 어댑터의 경우 일반적으로 04). OSA-Express 어댑터의 경우, 로컬 SAP 번호는 이 Linux 이미지에서 ethX 인터페이스에 대응하는 입출력 장치 주소의 OSA/SF에 정의된 번호와 일치해야 합니다.
- 노드가 시작될 때 포트가 자동으로 활성화되어야 하는지의 여부.
- 연결 네트워크 이름(동일한 연결 네트워크에 있는 모든 포트에서 동일해야 합니다).

이더넷 링크 스테이션의 경우 다음의 추가 정보가 필요합니다.

- 활성화 방식(관리자에 의해, 노드 시동 시 또는 요구가 있을 때)
- 지원되는 트래픽 유형(이 보기에서는, 종속형과 독립형 둘 다)
- 원격 노드 제어점 이름(LEN 노드를 위해서만 필요함)
- 원격 노드 유형(네트워크 노드, 끝 노드 또는 검색)
- 원격 노드 역할(이 예제에서는 다운스트림 SNA 게이트웨이 또는 패스-스루 DLUR)
- 선택적 링크 스테이션을 구성하려면, 원격 스테이션을 위한 MAC(매체 액세스 제어(MAC)) 주소와 SAP 번호(보통 04)가 필요합니다. 주소 정보를 제공하지 않고 활성화 필드에 대해 관리자에 의해 지정하는 경우, 링크 스테이션은 비선택적 연결대기 링크 스테이션입니다.

이더넷 링크를 구성하려면, 노드 창으로부터 다음의 단계를 수행하십시오.

1. 포트 구성:

- a. 창의 연결 분할 창을 선택하십시오.
- b. 서비스 메뉴의 연결성 하위 메뉴에서 새 포트를 선택하십시오. (또는 버튼 표시 줄에서 새로 작성 버튼을 누르십시오.)
- c. 그 결과 나오는 대화상자에서, 옵션 메뉴에서 프로토콜 유형을 선택하고 나서 포트를 정의하기로 선택하십시오.

확인 버튼을 누르면 Linux용 Communications Server에 이더넷 SAP 대화상자가 표시됩니다.

- d. 대화상자의 필드들에 적당한 값을 입력하십시오.
- e. 확인 버튼을 눌러 포트를 정의하십시오.

노드 창의 연결 분할창에 포트가 나타납니다.

2. 포트에 링크 스테이션 정의:

- a. 링크 스테이션이 노드 창의 연결 분할창에 추가되고 있는 포트를 선택했는지 확인하십시오.
- b. 서비스 메뉴의 연결성 하위 메뉴에서 새 링크 스테이션을 선택하십시오. (또는 버튼 표시줄에서 새로 작성 버튼을 누르십시오.)
- c. 확인 버튼을 누르십시오.

Linux용 Communications Server는 이더넷 링크 스테이션 대화상자를 표시합니다.

- d. 대화상자의 필드들에 적당한 값을 입력하십시오.
- e. 확인 버튼을 눌러 링크 스테이션을 정의하십시오.

링크 스테이션은 노드 창의 연결성 분할창에서 그것이 속하는 포트 아래에 나타납니다.

Enterprise Extender 링크 구성

이 예는 Enterprise Extender 링크를 APPN 네트워크에 구성하는 방법을 보여줍니다. Enterprise Extender 링크는 독립 LU 트래픽만 지원합니다.

Enterprise Extender 포트에는 다음 정보가 필요합니다.

- SNA 포트 이름(일반적으로 디폴트를 사용할 수 있습니다). IP를 실행 중인 네트워크 어댑터 카드가 여러 개 있는 경우 사용할 IP 인터페이스 이름(예: eth0)을 제공해야 합니다.
- 노드가 시작될 때 포트가 자동으로 활성화되어야 하는지의 여부.

Enterprise Extender 링크 스테이션에는 다음과 같은 추가 정보가 필요합니다.

- 활성화 방식(관리자에 의해, 노드 시동 시 또는 요구가 있을때)
- 원격 노드 유형(네트워크 노드, 끝 노드 또는 검색)
- 선택적 링크 스테이션을 구성하려면, 원격 스테이션에 대한 IP 호스트 이름 또는 IP 주소가 필요합니다. 이 정보를 제공하지 않고 활성화 필드에 관리자에 의해를 지정할 경우, 링크 스테이션은 비선택적 연결대기 링크 스테이션입니다.

Enterprise Extender 링크를 구성하려면, 노드 창에서 다음 단계를 수행하십시오.

1. 포트 구성:

- a. 창의 연결 분할 창을 선택하십시오.
- b. 서비스 메뉴의 연결성 하위 메뉴에서 새 포트를 선택하십시오. (또는 버튼 표시 줄에서 새로 작성 버튼을 누르십시오.)
- c. 그 결과 나오는 대화상자에서, 옵션 메뉴에서 프로토콜 유형을 선택하고 나서 포트를 정의하기로 선택하십시오.

확인 버튼을 누르면 Linux용 Communications Server에 IP 포트 대화상자가 표시됩니다.

- d. 대화상자의 필드들에 적당한 값을 입력하십시오.
- e. 확인 버튼을 눌러 포트를 정의하십시오.

노드 창의 연결 분할창에 포트가 나타납니다.

2. 포트에 링크 스테이션 정의:

- a. 링크 스테이션이 노드 창의 연결 분할창에 추가되고 있는 포트를 선택했는지 확인하십시오.
- b. 서비스 메뉴의 연결성 하위 메뉴에서 새 링크 스테이션을 선택하십시오. (또는 버튼 표시줄에서 새로 작성 버튼을 누르십시오.)
- c. 확인 버튼을 누르십시오.

Linux용 Communications Server는 IP 링크 스테이션 대화상자를 표시합니다.

- d. 대화상자의 필드들에 적당한 값을 입력하십시오.
- e. 확인 버튼을 눌러 링크 스테이션을 정의하십시오.

링크 스테이션은 노드 창의 연결성 분할창에서 그것이 속하는 포트 아래에 나타납니다.

유형 0-3 LU 구성

유형 0-3 LU를 사용하는 사용자 응용프로그램을 지원하려면 종속 LU를 구성해야 합니다. 이 작업을 하기 전에 다음 구성을 수행해야 합니다.

- 84 페이지의 『노드 구성』의 설명대로 노드를 구성하십시오.

유형 0-3 LU 구성

- 85 페이지의 『연결성 구성』에 설명된 것처럼 종속 LU 트래픽을 지원하도록 링크를 구성하십시오.

SNA 게이트웨이를 사용하는 다른 노드로 연결되는 업스트림 링크가 있거나 DLUR 을 사용하는 경우 호스트에 대한 직접 링크를 구성할 필요가 없습니다. 자세한 내용은 101 페이지의 『SNA 게이트웨이 구성』 및 103 페이지의 『DLUR 구성』을 참조하십시오.

호스트 시스템과의 통신을 지원하려면 유형 0-3의 종속 LU를 구성해야 합니다. 이 절의 정보를 사용하여 LUA, DLUR 또는 PU 집중을 지원하도록 LU를 정의할 수 있습니다. 또한 단일 조각으로 동일한 유형의 여러 LU를 구성하도록 LU 범위를 정의할 수도 있습니다.

뿐만 아니라, LU를 정의할 때 한 LU를 풀에 지정하거나 이미 정의된 LU를 풀에 지정하여 필요한 경우 사용될 LU 풀을 정의할 수 있습니다.

유형 0-3 LU 정의

3270 LU를 구성하기 전에, 다음의 정보를 모으십시오.

- LU 이름(이 LU 이름은 로컬 식별자이며 호스트 구성과 일치시키지 않아야 합니다.)
- LU 번호(또는 LU 범주에 대한 번호)
- LU 유형(3270 표시장치 모델이나 3270 프린터)
- 풀 이름(풀에 LU를 추가하고 있다면)

이전에 정의된 링크 스테이션에 대해 LU 유형 0-3을 구성하려면, 노드 창에서 다음 단계를 수행하십시오.

1. 창의 연결 분할창에서 호스트로의 링크 스테이션을 선택하십시오.
2. 추가 버튼을 누르십시오.
3. 그 결과 나타나는 대화 상자에서 LU 유형(새 **3270** 표시장치 LU 또는 새 **3270** 프린터 LU)을 선택하십시오.

이 항목을 선택하고 확인을 누르면 Linux용 Communications Server가 LU 유형 0-3 대화상자를 표시합니다.

4. 대화상자의 필드들에 적당한 값을 입력하십시오.
5. 확인을 눌러 LU를 정의하십시오.

호스트로의 링크 스테이션 아래에, 노드 창의 연결 분할창에 LU가 나타납니다.

LU 풀 정의

LU 유형 0-3의 경우, 설정하는 호스트 세션에서 보다 나은 융통성을 제공하고 사용자 구성을 단순화하기 위해 LU 풀을 정의할 수 있습니다. 예를 들어 하나의 LU 풀에 여

러 개의 LU를 정의하고 나서 이 LU 풀을 사용하여 여러 사용자를 구성할 수 있습니다. 이렇게 하면 더욱 간단하게 사용자 세션을 구성하고 세션에서 풀에 있는 임의의 LU를 사용할 수 있습니다.

주: 사용자 세션을 특정 LU 또는 LU 풀로 지정할 수 있습니다.

- 사용자 세션을 풀에 있는 특정 LU로 지정할 경우 세션이 사용 가능하다면 이 LU를 사용하며, 그렇지 않다면 특정 LU 대신 LU 풀로 지정했다라도 풀에서 아무 제약이 없는 임의의 LU를 사용합니다.
- LU가 이미 사용 중이면 사용자 세션을 설정할 수 없도록 사용자가 지정된 LU만을 사용하게 할 경우 LU가 풀에 없는지 확인하십시오.

LU 풀 창을 사용하여 로컬 Linux용 Communications Server 노드에 대한 LU 풀을 볼 수 있습니다. 이 창은 로컬 시스템상에 구성된 LU 풀을 열거하며, LU 풀이 추가할 LU를 선택할 수 있게 합니다.

다음 LU 유형을 풀에 추가할 수 있습니다. (서로 다른 유형의 LU를 동일한 풀에서 같이 사용하지 마십시오.)

- 3270 표시장치 LU
- 무제한 LU

풀에 LU를 추가할 수 있기 전에, 로컬 노드상에 LU가 정의되어야 합니다.

LU 풀을 구성하려면, 노드 창으로부터 다음의 단계를 수행하십시오.

1. 창 메뉴에서 **LU** 풀을 선택하십시오.

Linux용 Communications Server는 LU 풀 창을 표시합니다.

2. 추가 버튼을 누르십시오.

Linux용 Communications Server는 LU 풀 구성 대화상자를 표시합니다.

오른쪽의 상자는 풀에 아직 할당되지 않은 LU를 나열합니다. 이러한 LU의 어느 것이든 새 풀에 포함될 수 있습니다.

3. 풀에 추가하려는 LU를 선택하고 추가 버튼을 눌러 선택한 LU를 왼쪽 상자로 옮기십시오.

왼쪽 상자에서 LU를 제거하려면 선택한 후에 제거 버튼을 누르십시오.

4. 확인을 눌러 LU 풀을 정의하십시오.

왼쪽 상자에 있는 모든 LU가 LU 풀에 추가됩니다.

LU 풀 창에 풀이 나타납니다.

APPC 통신 구성

APPC 응용프로그램 및 CPI-C 응용프로그램의 경우 사용자가 먼저 APPC를 구성해 주어야 합니다. APPC 응용프로그램은 노드의 LU 유형 6.2를 사용하여, 지정된 모드를 사용해서 호스트나 대등 컴퓨터 상의 다른 APPC나 CPI-C 응용프로그램과 통신합니다.

APPC 통신을 구성할 수 있기 전에, 다음의 구성을 수행해야 합니다.

1. 84 페이지의 『노드 구성』의 설명대로 노드를 구성하십시오.
2. 85 페이지의 『연결성 구성』의 설명대로 연결성을 구성하십시오.

나머지 구성 단계는 구성의 종속 트래픽을 지원하는지, 독립 트래픽들을 지원하는지, 혹은 둘 다를 지원하는지에 따라 다릅니다.

독립 APPC

독립 APPC는 독립 LU를 사용합니다. 각각의 LU-LU 세션은 로컬 LU와 상대방 LU를 수반합니다.

로컬 LU에 있어서는, 노드 제어점(CP)과 연관된 사전 정의된 디폴트 LU를 사용하거나, 새로운 로컬 LU들을 구성할 수 있습니다.

Linux용 Communications Server 노드가 APPN 네트워크에서 끝 노드이거나 네트워크 노드인 경우에는 상대방 LU를 구성할 필요가 전혀 없는데, 이는 APPN이 동적으로 상대방 LU를 찾을 수 있기 때문입니다. 그렇지만, 사용자의 네트워크가 APPN 네트워크가 아니거나 노드가 LEN 노드이면 상대방 LU를 구성해야 합니다. 이 경우 상대방 LU가 위치하는 원격 노드를 구성한 다음 이 원격 노드 상에 상대방 LU를 정의해야 합니다.

종속 APPC

원격 노드가 독립 LU 6.2를 지원하지 않는 호스트이면 종속 트래픽에 대해 구성하십시오. 종속 APPC에 있어서는, 로컬 LU를 구성해야 합니다.

응용프로그램이 CPI-C를 사용하는 경우 APPC를 구성한 후에 추가 CPI-C 구성을 수행할 필요가 있을 지도 모릅니다(98 페이지의 『CPI 통신을 위한 구성』 참고). CPI-C 응용프로그램은 노드의 LU 유형 6.2와 모드 자원을 사용하여, 호스트나 대등 컴퓨터 상의 다른 APPC나 CPI-C 응용프로그램과 통신합니다. APPC 응용프로그램을 위한 것과 똑같은 CPI-C 응용프로그램 자원을 정의합니다. 또한, Linux용 Communications Server 컴퓨터의 TP가 소스 TP(대화를 시작하는 TP)라고도 하는 호출 TP인 경우에는 98 페이지의 『CPI 통신을 위한 구성』에 설명된 것처럼 하나 이상의 관련 부가 정보 항목을 정의해야 합니다. 이러한 항목들 각각은 그것을 액세스하는 데 사용되는 상대방 TP, LU 및 모드 자원과, 그리고 필요한 보안 정보를 제공합니다.

이 절에서는 93 페이지의 『단일 APPN 네트워크 구성』에서 설명한 대로, 네트워크 노드, 끝 노드 및 LEN 노드를 구성하는 단일 APPN 네트워크(독립 LU 6.2를 사용)를

구성하는 방법에 대해 설명합니다. 이 시나리오는 두 노드 간의 CP-CP 세션을 위한 상태 정보를 확보하는 방법에 대해서도 보여줍니다.)

이 절에서는 98 페이지의 『종속 APPC 구성』에서 설명한 대로 종속 APPC 통신을 구성하는 방법에 대해서도 설명합니다.

이 시나리오들 둘 다 APPC 세션들이 표준 모드와 서비스 클래스(COS)를 사용한다고 가정합니다.

모드, 보안 및 호출 가능한 (목표) TP와 같은 추가 APPC 정보를 구성하는 데 대한 자세한 내용은 *Communications Server for Linux Administration Guide*를 참조하십시오.

단일 APPN 네트워크 구성

구성할 수 있는 가장 간단한 APPN 네트워크는 단지 두 개의 노드, 즉 APPN 네트워크 노드와 APPN 끝 노드만을 포함하는 것입니다. 네트워크 노드는 끝 노드를 위한 세션 경로지정을 처리합니다.

네트워크 노드 구성

이 시나리오는 제어점(CP) LU와 표준 모드를 사용하고 있으며 LAN 링크 유형(토큰 링, 이더넷)을 사용하고 있다고 가정합니다. 이 경우, 다음의 구성 작업을 수행함으로써 간단히 네트워크 노드를 구성할 수 있습니다.

1. 84 페이지의 『노드 구성』의 설명대로 노드를 구성하십시오. APPN 지원 필드의 경우 값 네트워크 노드를 선택하십시오. 제어점 이름을 기록해 두십시오.
2. 85 페이지의 『연결성 구성』의 설명대로 연결성을 구성하십시오. 독립 트래픽을 지원하도록 링크를 구성하십시오.

인접 끝 노드로부터 이 네트워크 노드에 접속하려면, 네트워크 노드 상의 포트의 MAC 주소와 SAP 번호를 알 필요가 있습니다. 다음 프로시저에 따라 Linux용 Communications Server 노드에서 MAC 주소를 확보할 수 있습니다.

1. 노드 창에서 포트를 선택하십시오.
2. 시작 버튼을 눌러 포트를 시작하십시오.
3. 상태 버튼을 눌러 포트에 대한 상태 정보를 보십시오. 포트 상태 대화 상자에는 MAC 주소와 SAP 번호가 나타납니다.
4. 끝 노드를 위한 링크 스테이션 구성 대화상자에 값을 입력할 수 있도록 MAC 주소와 SAP 번호를 기록해 두십시오.

끝 노드 구성

이 시나리오는 제어점(CP) LU와 표준 모드를 사용하고 있으며 LAN 링크 유형(토큰 링, 이더넷)을 사용하고 있다고 가정합니다. 이 경우, 다음의 구성 작업을 수행함으로써 간단히 네트워크 노드를 구성할 수 있습니다.

1. 84 페이지의 『노드 구성』의 설명대로 노드를 구성하십시오. APPN 지원 필드의 경우 끝 노드를 선택하십시오.
2. 85 페이지의 『연결성 구성』의 설명대로 연결성을 구성하십시오. 독립 트래픽을 지원하도록 링크를 구성하고 링크 스테이션에 대해 다음 정보를 제공하십시오.
 - 원격 노드 필드에 네트워크 노드 이름을 입력하십시오(93 페이지의 『네트워크 노드 구성』 참조).
 - 링크 스테이션 구성 대화상자의 접속 정보 연결 창에서 네트워크 노드 상의 포트에 대한 MAC 주소와 SAP 번호를 입력하십시오.

APPN 네트워크에서, 각각의 원격 노드에 대해 별도의 링크 스테이션을 구성할 필요가 없도록, 네트워크에 있는 임의의 원격 노드와 통신하는 데 인접 네트워크 노드로의 단일 링크 스테이션이 사용될 수 있습니다.

두 노드 간의 연결성 검증

이 시나리오는 사용자가 93 페이지의 『네트워크 노드 구성』의 설명대로 네트워크 노드를 구성하고 93 페이지의 『끝 노드 구성』의 설명대로 끝 노드를 구성했다고 가정합니다. 끝 노드로부터 다음의 절차를 수행할 수 있습니다.

1. 노드 창에서, 인접 네트워크 노드에 연결하는 링크 스테이션을 선택하십시오.
2. 시작 버튼을 눌러 링크 스테이션을 시작하십시오.

링크 스테이션이 시작할 때, 두 노드간의 CP-CP 세션이 자동으로 형성될 수 있습니다. 그러한 세션들이 노드 창의 독립 로컬 LU 연결 창에 표시됩니다.

3. 세션에 대한 상태 정보를 보려면 노드 창에서 세션을 선택하고 상태 버튼을 누르십시오.

독립 APPC LU 구성

많은 경우에, 응용프로그램들은 노드를 구성할 때 자동으로 정의되는 로컬 노드의 제어 점을 사용할 수 있습니다. 이것은 디폴트 LU입니다. 응용프로그램이 특정 LU를 지정하지 않으면 디폴트 LU를 사용할 수 있습니다. 응용프로그램이 디폴트 LU를 사용하면, 로컬 LU를 정의할 필요가 없습니다. APPC 응용프로그램에 대한 문서를 살펴보거나 응용프로그램 프로그래머에게 문의하십시오.

독립 LU 6.2를 구성하기 위해서는, 다음과 같은 정보가 필요합니다.

- 로컬 LU 이름.
- 로컬 LU 별명(별명이 이 LU가 지원하는 TP에서 사용되는 경우).

독립 로컬 LU를 구성하려면, 노드 창으로부터 다음의 단계들을 수행하십시오.

1. 창의 독립 로컬 LU 분할창을 선택하십시오.
2. 서비스 메뉴의 APPC 하위 메뉴에서 새 독립 로컬 LU를 선택하십시오. (또는 새로 작성 버튼을 누르십시오.)

Linux용 Communications Server는 로컬 LU 대화상자를 표시합니다.

3. 대화상자의 필드들에 적당한 값을 입력하십시오.
4. 확인 버튼을 눌러 로컬 LU를 정의하십시오. 노드 창의 독립 로컬 LU 분할창에 독립 LU가 나타납니다.

LEN 노드를 위한 상대방 LU 구성

다음과 같은 상황에서 원격 노드(그리고 노드 상의 상대방 LU)를 정의해야 합니다.

- 로컬 노드가 LEN 노드이면, 모든 원격 노드들과 APPC를 사용하여 통신하는 원격 노드 상의 어떠한 상대방 LU들이든 정의해야 합니다. LEN 노드는 동적으로 상대방 LU의 위치를 찾을 수 있습니다. 원격 노드 정의가 그렇게 하도록 합니다.
- 로컬 노드가 APPN 네트워크의 일부가 아니라면(예를 들어, 네트워크 노드 서버없이 두 개의 끝 노드가 직접 연결되었다면), LU들의 위치를 자동으로 찾을 수 있습니다. 이 경우, 각각의 상대방 LU를 구성해야 합니다.
- 원격 노드가 LEN 노드이고 로컬 노드가 LEN 노드의 네트워크 노드 서버로서 행동하는 네트워크 노드이면, 네트워크 노드 서버 상의 원격 노드로서 LEN 노드(그리고 그 상대방 LU들)를 정의해야 합니다. 이 정의는 APPN 네트워크의 나머지에 있는 노드들이 LEN 노드 상의 LU들의 위치를 찾을 수 있습니다.
- 원격 노드가 서로 다른 APPN 네트워크에 있다면, 동적으로 위치가 찾아질 수 없기 때문에 원격 노드를 정의해야 합니다.

로컬과 원격 노드가 모두 동일한 APPN 네트워크의 일부라면, 상대방 LU들을 정의하지 마십시오.

원격 노드 정의를 추가할 때, 원격 노드와 동일한 이름을 가진 상대방 LU가 자동으로 추가됩니다. 이는 원격 노드를 위한 제어점 LU입니다. 응용프로그램이 이 상대방 LU를 사용한다면, 비록 상대방 LU를 위한 LU 별명을 추가하기를 원하는 경우에도 다른 상대방 LU를 추가할 필요가 없습니다. 별명을 추가하려면 상대방 LU를 두 번 누르고 상대방 LU 구성 대화 상자에 해당 별명을 입력하십시오.

응용프로그램이 상대방 LU를 참조하기 위해 LU 별명을 사용하는 경우 상대방 LU 별명 정의를 추가해야 합니다.

로컬 노드건 원격 노드건 어느 것이든 LEN 노드이면, LEN 노드가 동적 LU 위치 찾기에 참여할 수 없기 때문에 상대방 LU를 원격 노드의 자식으로서 정의해야 합니다. 어플리케이션이 상대방 LU로서 원격 노드의 제어점 LU를 사용한다면, 원격 노드를 정의할 때 제어점 LU가 자동으로 정의됩니다.

Motif 관리 프로그램을 사용하여 상대방 LU 별명을 추가하거나(『상대방 LU 별명 정의』 참고) 특정 원격 노드 상에 상대방 LU의 정의를 추가하거나(『원격 노드 상의 상대방 LU 정의』 참고) 또는 와일드카드를 사용하여 복수 상대방 LU를 정의할 수 있습니다(97 페이지의 『와일드카드를 사용하여 복수의 상대방 LU 정의』 참고).

원격 노드 정의: 원격 노드를 구성하기 전에, 다음과 같은 정보가 필요합니다.

- 노드의 전체 정식 SNA 네트워크 이름.

원격 노드를 구성하려면, 노드 창으로부터 다음의 단계들을 수행하십시오.

1. 창의 원격 시스템 분할창을 선택하십시오.
2. 서비스 메뉴의 **APPC** 하위 메뉴에서 새 원격 노드를 선택하십시오. (또는 버튼 표시줄에서 새로 작성 버튼을 누른 후, 원격 노드 정의를 선택하십시오.)

Linux용 Communications Server는 원격 노드 구성 대화상자를 표시합니다.

3. 대화상자의 필드들에 적당한 값을 입력하십시오.
4. 확인 버튼을 눌러 원격 노드를 정의하십시오. 노드 창의 원격 시스템 연결 창에 원격 노드가 나타납니다.

원격 시스템을 정의할 때 Linux용 Communications Server는 로컬 노드의 상대방 LU로서 원격 노드의 제어점(CP) LU를 자동으로 정의합니다.

상대방 LU 별명 정의: 상대방 LU 별명을 정의하려면, 다음과 같은 정보가 필요합니다.

- 상대방 LU 전체 정식 이름(SNA 네트워크 이름과 LU 이름)
- 로컬 TP에 의해 사용되는 상대방 LU 별명

상대방 LU 별명을 추가하려면, 노드 창으로부터 다음의 단계를 수행하십시오.

1. 창의 원격 시스템 분할창을 선택하십시오.
2. 서비스 메뉴에서 **APPC**, 새 상대방 LU 및 상대방 LU 별명을 선택하십시오. (또는 버튼 표시줄에서 추가 버튼을 누른 뒤 상대방 LU 별명 정의를 선택하십시오.)

Linux용 Communications Server는 상대방 LU 별명 구성 대화상자를 표시합니다.

3. 대화상자에서 상대방 LU 이름과 별명을 입력하십시오.
4. 확인 버튼을 눌러 상대방 LU 별명을 정의하십시오. 노드 창의 원격 시스템 분할창에 상대방 LU 별명이 나타납니다(네트워크 정의의 일부로서).

원격 노드 상의 상대방 LU 정의: 특정 원격 노드상의 상대방 LU를 정의하려면, 다음과 같은 정보가 필요합니다.

- 전체 정식 상대방 LU 이름.
- 상대방 LU 별명(별명이 TP에 의해 사용된다면).
- 상대방 LU에 대한 디렉토리 정보를 포함하는 전체 정식 노드 이름.

특정 원격 노드를 위한 상대방 LU 정의를 추가하려면, 노드 창으로부터 다음의 단계를 수행하십시오.

1. 원격 노드를 선택하십시오.
2. 서비스 메뉴에서 **APPC**, 새 상대방 LU 및 원격 노드 상의 상대방 LU를 선택하십시오. (또는 버튼 표시줄에서 추가 버튼을 누르고 원격 노드 상의 상대방 LU 정의를 선택하십시오.)

Linux용 Communications Server는 상대방 LU 구성 대화상자를 표시합니다.

3. 대화상자의 필드에 적당한 값을 입력하십시오.
4. 확인 버튼을 눌러 상대방 LU를 정의하십시오. 노드 창의 원격 시스템 분할창에 상대방 LU 별명이, 그것이 속하는 원격 시스템 아래에 나타납니다.

와일드카드를 사용하여 복수의 상대방 LU 정의: 와일드카드를 사용하여 동일한 원격 노드 상에 위치하며 이름이 동일한 문자로 시작하는 상대방 LU 세트에 대한 위치를 구성할 수 있습니다. 와일드카드를 사용한다고 해서 각각의 상대방 LU를 개별적으로 구성할 필요가 없다는 것을 의미하지는 않습니다.

총칭 문자를 사용하여 상대방 LU를 정의할 때에는 다음 정보를 제공해야 합니다.

- 와일드카드 상대방 LU 이름. 와일드카드 상대방 LU 이름은 여러 상대방 LU의 완전한 LU 이름과 일치하는, 두 가지 유형의 A EBCDIC 문자열로 구성되는데, 각각 1-8자로 되어 있습니다.

첫 번째 문자열은 정확히 상대방 LU들을 위한 네트워크 이름에 대응하는 완벽한 SNA 네트워크 이름이거나, 네트워크 이름의 시작에 대응하는 와일드카드 접두어일 수 있습니다. 와일드카드 접두어를 입력하면, 두 번째 문자열을 공백으로 두십시오.

첫 번째 문자열을 위한 완벽한 SNA 네트워크 이름을 제공하면, 두 번째 문자열을 위한 값을 입력할 수도 있습니다. (첫 번째 문자열을 위한 유효한 SNA 네트워크 이름을 제공하지 않는 한 두 번째 문자열을 입력할 수 없습니다.) 두 번째 문자열은 전체 정식 LU 이름의 두 번째 부분과 일치해야 하는 와일드카드 접두어로서 다루어 집니다.

- 상대방 LU가 위치하는 노드 이름.

복수개의 상대방 LU들을 추가하려면, 노드 창으로부터 다음의 단계들을 수행하십시오.

1. 상대방 LU들을 정의하고 있는 해당 원격 노드를 선택하십시오.
2. 서비스 메뉴에서 **APPC**, 새 상대방 LU 및 원격 노드 상의 와일드카드 상대방 LU를 선택하십시오. (또는 버튼 표시줄에서 추가 버튼을 누르고 원격 노드 상의 와일드카드 상대방 LU 정의를 선택하십시오.)

Linux용 Communications Server는 와일드카드 상대방 LU 구성 대화상자를 표시합니다.

3. 대화상자의 필드에 적당한 정보를 입력하십시오.
4. 확인 버튼을 눌러 상대방 LU를 정의하십시오. 노드 창의 원격 시스템 분할창에 상대방 LU들이, 그들이 속하는 원격 노드 아래에 나타납니다.

종속 APPC 구성

종속 LU 6.2를 구성하려면 다음 정보가 필요합니다.

- 로컬 LU 이름.
- 로컬 LU 별명(별명이 이 LU가 지원하는 TP에서 사용되는 경우).
- 호스트로의 연결을 제공하는 링크 스테이션 이름.
- LU 번호.
- LU가 종속 LU 6.2를 위한 디폴트 풀로 할당되어야 하는지 여부.

APPC나 CPI-C 응용프로그램과 사용할 유형 6.2의 종속 LU들을 구성하고 있다면, 이들을 디폴트 풀의 멤버로서 정의하고자 할 것입니다. 특정 로컬 LU를 지정하지 않는 어플리케이션은 디폴트 LU로서 정의된 LU의 풀로부터 비사용 LU로 할당됩니다.

종속 로컬 LU를 구성하려면, 노드 창으로부터 다음의 단계들을 수행하십시오.

1. 창의 연결 분할창에서 호스트로의 링크 스테이션을 선택하십시오.
2. 서비스 메뉴의 **APPC** 하위 메뉴에서 새 종속 로컬 LU를 선택하십시오. (또는 버튼 표시줄에서 새로 작성 버튼을 누르고 새 종속 로컬 LU를 선택하십시오.)

Linux용 Communications Server는 로컬 LU 대화상자를 표시합니다.

3. 대화상자의 필드들에 적당한 값을 입력하십시오.
4. 확인 버튼을 눌러 로컬 LU를 정의하십시오. 종속 LU가 그것이 속하는 링크 스테이션 밑에, 연결 분할창에 나타납니다.

CPI 통신을 위한 구성

CPI-C 기호식 목적지 이름을 사용하는 CPI-C 응용프로그램을 지원하고 있다면, CPI-C 부가 정보를 정의할 필요가 있습니다. 부가 정보는 기호식 목적지명을 대화를 위한 상대방 TP, 상대방 LU, 모드, 그리고 보안에 관한 정보와 연관시킵니다.

CPI-C에 대한 기호식 목적지 이름을 판별하려면, 응용프로그램 개발자에게 문의하십시오. (또는 제3자 응용프로그램에 대해서는, 제품 문서를 살펴보십시오.)

CPI-C 부가 정보를 구성하기 전에, 다음과 같은 정보가 필요합니다.

- TP에 의해 사용되는 기호식 목적지 이름
- 상대방 TP 이름
- 상대방 LU 이름이나 별명
- 모드 이름

CPI-C 부가 정보를 구성하기 위해서는, 노드 창으로부터 다음의 단계를 수행하십시오.

1. 서비스 메뉴의 **APPC** 하위 메뉴에서 **CPI-C**를 선택하십시오.

Linux용 Communications Server는 CPI-C 목적지 이름 창을 표시합니다.

2. 추가 버튼을 누르십시오.

Linux용 Communications Server는 CPI-C 목적지 구성 대화상자를 표시합니다.

3. 대화상자의 필드들에 적당한 값을 입력하십시오.
4. 확인 버튼을 눌러 CPI-C 부가 정보를 정의하십시오.

LUA 구성

LUA API는 LU dbgd 0-3을 사용하여 호스트 컴퓨터와 통신하는 응용프로그램에 사용할 수 있습니다. LUA API에 대한 자세한 내용은 *Communications Server for Linux LUA Programmer's Guide*를 참조하십시오.

LUA를 구성하기 전에, 다음의 구성을 수행하십시오.

1. 84 페이지의 『노드 구성』의 설명대로 노드를 구성하십시오.
2. 85 페이지의 『연결성 구성』의 설명대로 중속 트래픽에 대한 연결을 구성하십시오.(업스트림 SNA 게이트웨이 또는 DLUR을 사용하고 있다면 호스트로의 직접 링크 대신 업스트림 노드로의 링크를 구성하십시오.)

LUA를 구성하려면, 다음과 같은 정보가 필요합니다.

- LU 이름이나 LU 풀 이름.
- 각각에 대한 LU 번호. LU 번호는 호스트상에 구성된 번호와 일치해야 합니다.

LUA를 구성하려면, 다음 절차를 사용하여 LU를 정의하십시오.

1. 노드 창의 연결 분할창에서 호스트로의 링크 스테이션을 선택하십시오.
2. 추가 버튼을 누르십시오.
3. 나타나는 대화 상자에서 **LUA**를 위한 새 **LU**를 선택하십시오.
4. 대화상자의 필드에 적당한 값을 입력하십시오. 무제한 LU 유형을 지정하십시오.
5. 확인 버튼을 누르십시오. 노드 창의 연결 분할창에서 호스트로의 링크 스테이션 아래, LU가 나타납니다.
6. LU 풀을 사용하려는 경우 이들을 90 페이지의 『LU 풀 정의』의 설명대로 정의하십시오.

LU 풀 정의

사용자 구성을 단순화하고 설정되는 호스트 세션에 더욱 향상된 융통성을 제공하기 위해 LU 풀을 정의할 수 있습니다. 예를 들어 여러 개의 LU를 하나의 LU 풀로 정의하

고 나서, 이 LU 풀을 사용하여 여러 개의 LUA 응용프로그램을 구성할 수 있습니다. 이는 응용프로그램 구성을 더욱 용이하게 하고 임의의 응용프로그램이 풀에 있는 임의의 LU를 사용할 수 있게 합니다.

주: 사용자 세션을 특정 LU 또는 LU 풀로 지정할 수 있습니다.

- 사용자 세션을 풀에 있는 특정 LU로 지정할 경우 세션이 사용 가능하다면 이 LU를 사용하며, 그렇지 않다면 특정 LU 대신 LU 풀로 지정했다라도 풀에서 아무 제약이 없는 임의의 LU를 사용합니다.
- LU가 이미 사용 중이면 사용자 세션을 설정할 수 없도록 사용자가 지정된 LU만을 사용하게 할 경우 LU가 풀에 없는지 확인하십시오.

LU 풀 창을 사용하여 로컬 Linux용 Communications Server 노드에 대한 LU 풀을 볼 수 있습니다. 이 창은 로컬 시스템상에 구성된 LU 풀을 열거하며, LU 풀이 추가할 LU를 선택할 수 있게 합니다.

다음의 LU 유형을 3270에 의해 사용될 풀에 추가할 수 있습니다. (서로 다른 유형의 LU를 동일한 풀에 혼합하지 마십시오.)

- 3270 표시장치 LU
- 무제한 LU

풀에 LU를 추가할 수 있기 전에, 로컬 노드상에 LU가 정의되어야 합니다.

LU 풀을 구성하려면, 노드 창으로부터 다음의 단계를 수행하십시오.

1. 창 메뉴에서 **LU** 풀을 선택하십시오.

Linux용 Communications Server는 LU 풀 창을 표시합니다.

2. 추가 버튼을 누르십시오.

Linux용 Communications Server는 LU 풀 구성 대화상자를 표시합니다.

오른쪽의 상자는 풀에 아직 할당되지 않은 LU를 나열합니다. 무제한 LU 유형의 이 LU 중 어느 것이든 LUA의 새 풀에 포함될 수 있습니다.

3. 풀에 추가하려는 LU를 선택하고 추가 버튼을 눌러 선택한 LU를 왼쪽 상자로 옮기십시오.

왼쪽 상자에서 LU를 제거하려면 선택한 후에 제거 버튼을 누르십시오.

4. 확인을 눌러 LU 풀을 정의하십시오.

왼쪽 상자에 있는 모든 LU가 LU 풀에 추가됩니다.

LU 풀 창에 풀이 나타납니다.

SNA 게이트웨이 구성

Linux용 Communications Server는 호스트 컴퓨터에 대한 직접 액세스를 제공할 뿐 아니라 SNA 게이트웨이 기능도 제공할 수 있습니다. 이 기능을 통해 다른 컴퓨터는 각 컴퓨터에서 호스트로 연결되는 별도 연결을 요구하지 않고 Linux용 Communications Server 노드를 통해 호스트 컴퓨터에 액세스할 수 있습니다.

SNA 게이트웨이 기능은 그림 7에 나타냅니다.

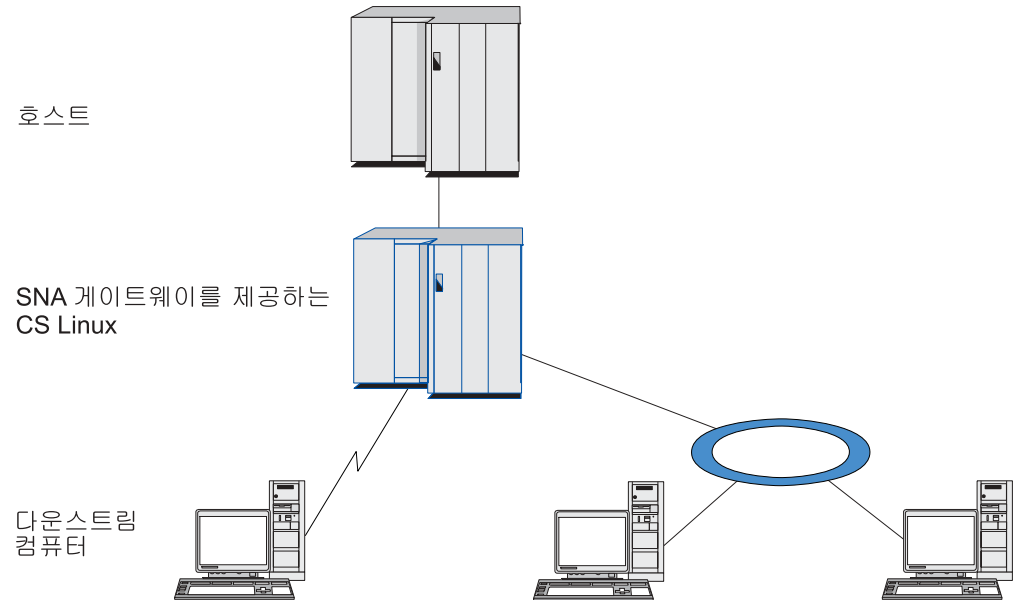


그림 7. SNA 게이트웨이

다운스트림 컴퓨터는 종속 LU들을 지원하기 위해 SNA PU 유형 2.0이나 2.1을 포함해야 합니다. 예를 들면 다운스트림 컴퓨터는 다른 Linux용 Communications Server 컴퓨터 또는 Windows NT[®]용 Communications Server를 실행하는 PC일 수 있습니다.

로컬 Linux용 Communications Server 노드가 SNA 게이트웨이 기능을 사용할 경우 호스트와 다운스트림 컴퓨터 사이에서 전송되는 모든 데이터는 로컬 노드를 통해 경로 지정됩니다. 따라서 다운스트림 컴퓨터는 직접 링크를 요구하지 않고 Linux용 Communications Server나 다른 다운스트림 컴퓨터와 호스트 연결을 공유할 수 있습니다. 예를 들어 로컬 토큰링 네트워크에서 Linux용 Communications Server에 연결된 몇 개의 다운스트림 컴퓨터를 설정할 수 있으므로 이들 모든 컴퓨터는 Linux용 Communications Server에서 호스트로 연결되는 동일한 장거리 전용 회선에 액세스할 수 있습니다.

SNA 게이트웨이를 사용하면 다운스트림 컴퓨터와 컴퓨터 상의 통신 링크를 정의할 필요가 없기 때문에 호스트에서 구성을 단순화할 수 있습니다. 호스트 구성에는 Linux용 Communications Server 컴퓨터 및 해당 호스트 통신 링크만이 포함되어야 합니다. 다

다운스트림 컴퓨터의 LU는 Linux용 Communications Server 컴퓨터 자원의 일부분으로 구성됩니다. 호스트 컴퓨터는 SNA 게이트웨이가 사용되고 있음을 알지 못합니다.

SNA 게이트웨이를 구성하기 전에, 다음의 구성 작업을 수행해야 합니다.

- 84 페이지의 『노드 구성』의 설명대로 로컬 노드를 정의하십시오.
- 85 페이지의 『연결성 구성』의 설명대로 로컬 노드와 호스트 간 종속 트래픽을 위한 포트 및 링크 스테이션을 구성하십시오. 또한, 로컬 노드와 다운스트림 노드간의 종속 트래픽을 위한 포트 및 링크 스테이션을 구성하십시오. 미리 정의하지 않은 다운스트림 LU를 지원할 필요가 있으면 암시적 PU 및 다운스트림 LU를 지원하도록 포트에 템플릿을 정의할 수 있습니다(『암시적 다운스트림 LU 지원』 참고).
- 호스트와의 통신에 사용되는 로컬 노드 상의 LU(업스트림 LU)를 정의하십시오. 업스트림 LU를 무제한(알 수 없음)의 LU 유형을 갖는 LU 유형 0-3으로 정의하십시오.(다운스트림 노드 상의 LU는 어떤 유형이든 무관합니다.)
- LU 풀을 사용하려는 경우 99 페이지의 『LU 풀 정의』에서 설명한 대로 정의하십시오.

암시적 다운스트림 LU 지원

Linux용 Communications Server에 사전 정의되지 않은 다운스트림 LU를 지원하기 위해 암시적 다운스트림 PU 및 LU용 포트에 템플릿을 정의할 수 있습니다(기본 포트 구성은 85 페이지의 『연결성 구성』 참조). 이들 템플릿은 다운스트림 노드 상의 모든 LU를 지원하도록 로컬 노드상에 LU를 구성할 필요없이 다운스트림 LU를 지원 합니다.

SNA 게이트웨이를 위한 다운스트림 LU를 구성하기 전에, 다음과 같은 정보가 필요합니다.

- 다운스트림 LU를 지원하기 위한 LU 번호들의 범주.
- 호스트 LU 이름.

암시적 다운스트림 LU들을 위한 템플릿을 정의하려면, 다음 단계를 수행하십시오.

1. 이미 포트를 구성했다면, 노드 창의 연결 분할창에서 포트 정의를 두 번 누르십시오. Linux용 Communications Server는 포트 구성 대화상자를 표시합니다.

포트를 아직 구성하지 않았다면, 지금 구성하십시오.

- a. 노드 창에서 연결 분할창을 선택하십시오.
- b. 추가 버튼을 누르십시오.
- c. 그 결과 나오는 대화상자에서, 포트 정의를 선택하고 링크 프로토콜 유형을 선택하십시오.

Linux용 Communications Server는 포트 구성 대화상자를 표시합니다.

- d. 85 페이지의 『연결성 구성』의 설명대로 기본 포트 매개변수를 입력하십시오.
2. 대화 상자 하단의 **확장** 버튼을 누르십시오.

Linux용 Communications Server는 포트 매개변수 대화상자를 표시합니다. 아래의 분할 창은 다운스트림 LU 템플릿들에 영향을 미치는 설정들을 보여줍니다.

3. **암시적 PU 액세스를 위한 다운스트림 LU 구성 옵션**을 선택하십시오.
4. **확인**을 누르십시오.

Linux용 Communications Server는 다운스트림 LU 템플릿 구성 대화상자를 표시합니다.

5. 대화상자의 필드들에 적당한 값을 입력하십시오.
6. **확인**을 눌러 암시적 다운스트림 LU 템플릿을 정의하십시오.

다운스트림 LU 정의

SNA 게이트웨이를 위한 다운스트림 LU를 구성하기 전에, 다음과 같은 정보가 필요합니다.

- 각 다운스트림 LU에 대한 LU 이름(이 LU 이름은 로컬 식별자이며 다운스트림 시스템의 구성과 일치시키지 않아야 합니다).
- 각 다운스트림 LU에 대한 LU 번호.
- 다운스트림 노드로의 링크 스테이션.
- 업스트림 LU 이름(호스트 LU용).

SNA 게이트웨이를 위한 다운스트림 LU를 구성하려면, 다음의 단계를 수행하십시오.

1. 노드 창의 연결 분할창에서 다운스트림 노드로의 링크 스테이션을 선택하십시오.
2. **추가** 버튼을 누르십시오.
3. 새 **다운스트림 LU**를 선택하고 **확인**을 누르십시오.

Linux용 Communications Server는 다운스트림 LU 대화상자를 표시합니다.

4. 대화상자의 필드들에 적당한 값을 입력하십시오.
5. **확인**을 눌러 다운스트림 LU를 정의하십시오.

노드 창의 연결 분할창에서 다운스트림 노드로의 링크 스테이션 밑에, LU 정의가 나타납니다.

DLUR 구성

Linux용 Communications Server는 호스트 컴퓨터로의 직접 액세스를 제공할 뿐 아니라 종속 LU 리퀘스터(DLUR) 기능을 제공할 수 있습니다. 이 기능은 종속 LU들을 위한 세션들이 호스트로의 직접 링크를 필요로 하는 대신에, APPN 네트워크에서 복수의 노드로 확장할 수 있게 합니다.

보통, 종속 LU 세션은 호스트 컴퓨터로의 직접 통신 링크를 필요로 합니다. 많은 노드가(호스트 노드를 포함하여) APPN 네트워크에서 함께 연결되면, 그들 중 어떤 것들은 호스트로의 직접 연결을 가질지도 모르나, 다른 노드를 통한 간접 연결만이 그렇습니다. 이러한 간접적으로 연결된 노드에 있는 LU들로부터 호스트로의 종속 LU 세션을 형성하는 것은 가능하지 않습니다.

종속 LU 리퀘스터(DLUR)는 이러한 제한을 극복하기 위해 고안된 APPN 기능입니다.

APPN 노드(예: Linux용 Communications Server 노드)의 DLUR은 호스트에 있는 종속 LU 서버(DLUS)와 함께 작동하여, APPN 네트워크를 통해 DLUR 노드의 종속 LU에서 DLUS 호스트까지 세션을 경로지정합니다. 호스트로의 전송 경로는 복수의 노드로 확장할 수 있으며, APPN의 네트워크 관리, 동적 자원 위치, 그리고 전송 경로 계산 기능을 활용할 수 있습니다. LU들이 위치한 노드 상에 DLUR이 사용 가능해야 하며, 호스트 노드 상에 DLUS가 사용 가능해야 합니다. 그러나 세션 전송 경로에서 임의의 중계 노드에 대해서는 DLUR이 필요하지 않습니다.

Linux용 Communications Server DLUR 노드가 네트워크 노드 또는 브랜치 네트워크 노드이면 Linux용 Communications Server 노드에 연결된 다운스트림 컴퓨터의 종속 LU에 패스-스루 DLUR 기능을 제공할 수도 있습니다. 이러한 LU는 Linux용 Communications Server 노드의 DLUR을 사용하여 노드 내부의 LU에서와 같은 방법으로, 네트워크를 통해 호스트에 액세스할 수 있습니다. 다운스트림 컴퓨터는 DLUR을 실행하지 않으며, DLUR이 사용되고 있다는 것을 인식하지 못해도 됩니다.

105 페이지의 그림 8에는 Linux용 Communications Server를 APPN 네트워크 노드로 구성하여 호스트의 LU(업스트림 노드)와 APPN 네트워크에 있는 노드의 LU(다운스트림 노드) 사이의 세션을 지원하는 패스-스루 DLUR이 구현되어 있습니다.

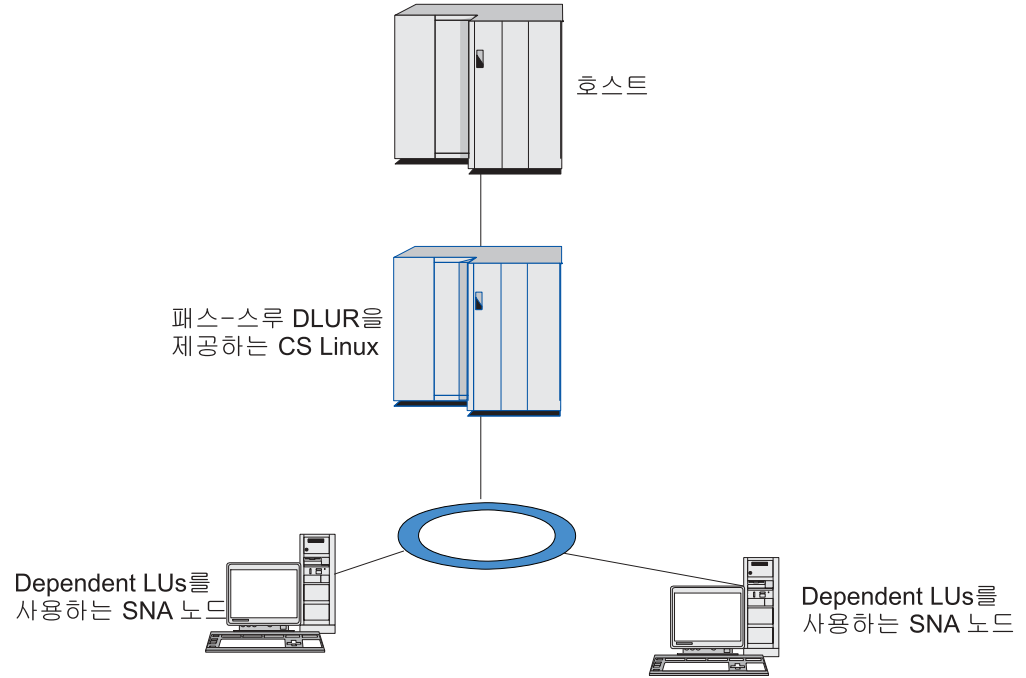


그림 8. DLUR을 제공하는 Linux용 Communications Server

주:

1. LEN 노드 상에서 DLUR을 구성할 수 없습니다.
2. 네트워크 노드 또는 브랜치 네트워크 노드에서만 패스-스루 DLUR을 구성할 수 있습니다.
3. Branch Extender를 사용하면, DLUR을 (브랜치 네트워크 노드를 네트워크 노드 서버로 사용하는) 브랜치의 끝 노드에 구성할 수 없습니다. 단, 패스-스루 DLUR을 브랜치 네트워크 노드에 구성하여 이 노드에서 종속 LU 응용프로그램을 지원할 수 있습니다. (따라서 브랜치에 있는 끝 노드는 DLUR을 실행하지 않지만 브랜치 네트워크 노드에 있는 패스-스루 DLUR을 사용합니다.)

DLUR을 구성하기 위해 수행할 필요가 있는 task들은 종속 LU들이 로컬 노드에 있는 지 다운스트림 노드에 있는지에 따라 다릅니다.

로컬 노드 상의 DLUR 지원 구성

이 task를 위한 다음과 같은 정보가 필요합니다.

- 로컬 노드에서 PU에 대한 PU ID.
- PU 이름(이 LU 이름은 로컬 식별자이며 호스트 구성과 일치시키지 않아야 합니다).
- 호스트의 DLUS 이름(그리고 백업 DLUS가 있는 경우 이 백업 DLUS의 이름)
- 각각의 다운스트림 LU를 위한 LU 이름, LU 번호, 그리고 LU 유형. LU 번호는 호스트상에 구성된 번호와 일치해야 합니다.

로컬 노드 상에 DLUR 지원을 구성하려면, 다음의 구성 task를 수행해야 합니다.

1. 84 페이지의 『노드 구성』의 설명대로 로컬 노드를 정의하십시오. 다운스트림 노드에 패스-스루 DLUR 지원을 제공하고 있는 경우 노드를 APPN 네트워크 노드 또는 브랜치 네트워크 노드로 정의하십시오.
2. APPN 네트워크에 연결성을 구성하십시오. 85 페이지의 『연결성 구성』에서 설명한 대로, APPN 연결성은 로컬 노드와 인접 APPN 네트워크 노드 간의 독립적인 트래픽을 위해 최소한 한개의 포트와 링크 스테이션이 필요합니다.
3. 로컬 노드 상에 DLUR PU를 정의하십시오. (DLUR PU는 호스트로의 연결성을 지원합니다.)

DLUR PU를 구성하려면, 노드 창으로부터 다음의 단계를 수행하십시오.

- a. 서비스 메뉴를 선택한 후, 연결성 하위 메뉴를 선택하고, 새 DLUR PU를 선택하십시오. (또는 버튼 표시줄에서 새로 작성 버튼을 누른 후, DLUR PU를 선택하십시오.)

확인 버튼을 누르면 Linux용 Communications Server에 DLUR PU 구성 대화상자가 표시됩니다.

- b. 대화상자의 필드들에 적당한 값을 입력하십시오.
- c. 확인 버튼을 눌러 DLUR PU를 정의하십시오.

DLUR 항목 아래의 연결 분할창에 DLUR PU가 나타납니다.

4. 로컬 노드에서 LU를 지원하도록 DLUR을 구성하려면 로컬 노드에 LU를 추가해야 합니다. 99 페이지의 『LUA 구성』에 설명된 대로 LUA를 지원하도록 LU를 구성해야 합니다. LU에서 지원하는 사용자 응용프로그램 요구사항에 따라 더 많은 구성을 수행해야 합니다.

다운스트림 노드에 대한 패스-스루 DLUR 지원 구성

이 작업을 위한 다음과 같은 정보가 필요합니다.

- 각 다운스트림 노드 또는 다운스트림 노드의 각 PU에 대한 다운스트림 PU 이름. (이 PU 이름은 로컬 식별자이며 호스트 구성과 일치시키지 않아야 합니다.)
- 호스트 상의 DLUS 이름.

다운스트림 노드에 대해 패스-스루 DLUR 지원을 구성하려면, 다음의 구성 작업을 수행해야 합니다.

1. 로컬 노드를 APPN 네트워크 노드로 정의하십시오(84 페이지의 『노드 구성』 참고).
2. 다운스트림 노드에 연결성을 구성하십시오. 85 페이지의 『연결성 구성』의 설명대로 로컬 노드와 각 다운스트림 노드간 종속 트래픽을 위한 포트 및 링크 스테이션을 구성하십시오 (다운스트림 노드를 위한 패스-스루 DLUR을 지원하도록 DLUR PU를 정의할 필요가 없습니다.)

- 다운스트림 노드는 복수의 PU를 지원할 수 있습니다. 이 경우 각 다운스트림 PU는 다른 링크와 연관되므로, 사용자는 Linux용 Communications Server DLUR 노드와 다운스트림 노드 사이에서 여러 개의 링크를 구성해야 하며 각 링크에 대한 다운스트림 PU 이름을 알아야 합니다.

TN 서버 구성

TCP/IP를 통해(SNA 네트워크를 통해서가 아니라) 통신하는 3270 에뮬레이션 프로그램은 TN3270 프로그램(Telnet 3270 에뮬레이션프로그램)으로 언급됩니다.

TN3270 프로그램은 TN3270E(Telnet 3270 표준 확장)를 위한 지원도 포함할 수 있습니다. TN3270E는 텔넷을 사용한 3270 장치 에뮬레이션(단말기와 프린터를 포함하여)을 지원하는 개방형 프로토콜입니다. 텔넷 클라이언트가 특정 장치를 선택할 수 있게 하며(LU 이름을 지정함으로써), ATTN 및 SYSREQ 키들과 SNA 자원 응답 처리를 포함하여 다양한 기능들을 위한 향상된 지원을 제공합니다.

주: 이 안내서에서는 TN3270, TN3287 및 TN3270E 프로토콜에 동등하게 적용되는 정보에 대해 TN3270이라는 용어를 사용합니다.

Linux용 Communications Server TN 서버를 사용하면 다른 컴퓨터의 TN3270 사용자가 3270 호스트 컴퓨터에 액세스할 수 있습니다. TN 서버를 사용하면 TN3270 사용자가 직접 링크 없이도 Linux용 Communications Server 또는 다른 TN3270 사용자와 호스트 연결을 공유할 수 있습니다. 또한 TN 서버는 TN3270 사용자들이 TCP/IP를 실행하고 있지 않는 호스트에 액세스할 수 있게 해 줍니다.

Linux용 Communications Server TN 서버 기능은 TN3270 사용자와 Linux용 Communications Server 3270 LU 사이의 연결을 제공합니다. TN3270 사용자로부터의 모든 데이터의 경로가 LU로 지정됩니다. 이는 호스트와 TN3270 사용자 둘 다를 위한 구성이 직접 연결된 것과 같다는 것을 의미합니다. 따라서 데이터의 경로가 TN 서버를 통해 지정되고 있음을 알 필요가 없습니다.

Linux용 Communications Server TN 서버는 IETF RFC 1123, 1576, 1646, 1647 및 2355에 정의된 프로토콜을 정확하게 구현하는 모든 TN3270 클라이언트 에뮬레이션 프로그램을 지원합니다.

TN3270 프로그램이 TN 서버와 통신할 때 Linux용 Communications Server는 TN3270 프로그램이 실행 중인 컴퓨터의 TCP/IP 주소로 프로그램을 식별합니다. Linux용 Communications Server는 동일한 컴퓨터에서 서로 다른 사용자가 사용하고 있는 두 개의 서로 다른 TN3270 프로그램을 구분할 수 없습니다. Linux용 Communications Server 매뉴얼에서 TN 서버 사용자라는 용어는 TN3270 프로그램의 개별 사용자가 아니라 해당 프로그램이 실행 중인 컴퓨터를 의미합니다.

TN3270 서버 기능을 사용하여 Linux용 Communications Server에 연결하는 각 TN 서버 사용자는 일반적으로 단일 3270 LU에 액세스하도록 구성되므로 한 번에 하나의 호스트 세션으로 제한됩니다. 그렇지만, 각각의 사용자를 위한 전담 3270 LU를 가지는 대신에, TN 서버 사용자가 3270 LU들의 풀을 액세스하도록 구성할 수도 있습니다. 이는 사용자들이 풀에 있는 사용 가능한 LU와 같은 수의 세션에 액세스할 수 있게 합니다.

TN 서버 액세스를 구성할 수 있기전에, 다음의 구성 작업을 수행해야 합니다.

- 84 페이지의 『노드 구성』의 설명대로 로컬 노드를 정의하십시오.
- 85 페이지의 『연결성 구성』의 설명대로 로컬 노드와 호스트 간 종속 트래픽을 위한 포트 및 링크 스테이션을 구성하십시오.

TN 서버 액세스를 구성하려면 다음 구성 작업을 수행해야 합니다.

- 호스트와의 통신에 사용되는 로컬 노드 상의 3270 LU를 정의하십시오. LU를 추가하려면 『3270 LU 정의』를 참고하십시오.
- LU 풀을 사용하려는 경우 109 페이지의 『LU 풀 정의』에서 설명한 대로 정의하십시오.

3270 LU 정의

3270 LU를 구성하기 전에, 다음의 정보를 모으십시오.

- LU 이름(이 LU 이름은 로컬 식별자이며 호스트 구성과 일치시키지 않아야 합니다.)
- LU 번호(또는 LU 범주에 대한 번호)
- LU 유형(3270 표시장치 모델이나 3270 프린터)
- 풀 이름(풀에 LU를 추가하고 있다면)

이전에 정의된 링크 스테이션에 대해 LU 유형 0-3을 구성하려면, 노드 창에서 다음 단계를 수행하십시오.

1. 창의 연결 분할창에서 호스트로의 링크 스테이션을 선택하십시오.
2. 추가 버튼을 누르십시오.
3. 그 결과 나타나는 대화 상자에서 LU 유형(새 3270 표시장치 LU 또는 새 3270 프린터 LU)을 선택하십시오.

이 항목을 선택하고 확인을 누르면 Linux용 Communications Server가 LU 유형 0-3 대화상자를 표시합니다.

4. 대화상자의 필드들에 적당한 값을 입력하십시오.
5. 확인을 눌러 LU를 정의하십시오.

호스트로의 링크 스테이션 아래에, 노드 창의 연결 분할창에 LU가 나타납니다.

LU 풀 정의

3270에 있어서는, 사용자 구성을 단순화하고 형성하는 호스트 세션에 보다 나은 융통성을 제공하기 위해 LU 풀을 정의할 수 있습니다. 예를 들어 여러 개의 3270 LU를 하나의 LU 풀로 정의하고 나서, 이 LU 풀을 사용하여 여러 개의 TN3270 클라이언트를 구성할 수 있습니다. 이는 3270 세션 구성을 더욱 용이하게 하고 임의의 클라이언트가 풀에 있는 임의의 LU를 사용할 수 있게 합니다.

주: TN3270 클라이언트를 특정 LU 또는 LU 풀로 지정할 수 있습니다.

- 클라이언트를 풀에 있는 특정 LU로 지정할 경우 클라이언트가 사용 가능하다면 이 LU를 사용하며, 그렇지 않다면 특정 LU 대신 LU 풀로 지정했다면 풀에서 아무 제약이 없는 임의의 LU를 사용합니다.
- LU가 이미 사용 중이면 클라이언트 세션을 설정할 수 없도록 클라이언트가 지정된 LU만을 사용하게 할 경우 LU가 풀에 없는지 확인하십시오.

LU 풀 창을 사용하여 로컬 Linux용 Communications Server 노드에 대한 LU 풀을 볼 수 있습니다. 이 창은 로컬 시스템상에 구성된 LU 풀을 열거하며, LU 풀이 추가할 LU를 선택할 수 있게 합니다.

다음의 LU 유형을 3270에 의해 사용될 풀에 추가할 수 있습니다. (서로 다른 유형의 LU를 동일한 풀에 혼합하지 마십시오.)

- 3270 표시장치 LU
- 무제한 LU

풀에 LU를 추가할 수 있기 전에, 로컬 노드상에 LU가 정의되어야 합니다.

LU 풀을 구성하려면, 노드 창으로부터 다음의 단계를 수행하십시오.

1. 창 메뉴에서 **LU** 풀을 선택하십시오.

Linux용 Communications Server는 LU 풀 창을 표시합니다.

2. 추가 버튼을 누르십시오.

Linux용 Communications Server는 LU 풀 구성 대화상자를 표시합니다.

오른쪽의 상자는 풀에 아직 할당되지 않은 LU를 나열합니다. 이러한 LU의 어느 것이든 새 풀에 포함될 수 있습니다.

3. 풀에 추가하려는 LU를 선택하고 추가 버튼을 눌러 선택한 LU를 왼쪽 상자로 옮기십시오.

왼쪽 상자에서 LU를 제거하려면 선택한 후에 제거 버튼을 누르십시오.

4. 확인을 눌러 LU 풀을 정의하십시오.

왼쪽 상자에 있는 모든 LU가 LU 풀에 추가됩니다.

LU 풀 창에 풀이 나타납니다.

TN3270 서버 구성

TN3270 서버를 구성하기 전에 다음과 같은 정보가 필요합니다.

- 서버가 TN3270만을 지원하는지 혹은 TN3270E(TN3270 지원을 포함하는)도 지원하는지 여부.
- TN3270E 클라이언트가 특정 LU를 요청할 수 있는지 여부.
- 각각의 클라이언트를 위한 표시장치 및 프린터 LU 이름(또는 LU 풀 이름). (프린터 LU 이름은 TN3270E를 지원하는 경우에만 필요합니다.)
- 특정 클라이언트만 허용되거나, 특정 클라이언트를 특정 LU로 제한하려는 경우 클라이언트의 TCP/IP 이름이나 주소가 필요합니다.
- TN 서버 노드 상의 TCP/IP 포트 번호.
- SSL 데이터 암호화 여부와 관계없이, 클라이언트 인증과 서버 인증이 요구됩니다. (이 옵션은 이를 지원하는 데 필요한 추가 소프트웨어를 설치한 경우에만 사용할 수 있습니다.)

표시장치 LU와 프린터 LU를 연관시키려면, 그러한 LU의 이름도 필요합니다. TN 서버 연관 레코드는 TN3270E 프로토콜이 두 개를 연결할 수 있도록, 프린터 LU와 표시장치 LU 사이의 연관을 정의합니다. TN3270E 또는 프린터 LU를 지원하고 있지 않다면 연결 레코드를 정의할 필요가 없습니다.

TN 서버 디폴트 레코드는 모든 TN3270 클라이언트 세션에서 사용되는 매개변수를 정의합니다. 각각의 서버에 대해 단일 디폴트 레코드를 정의할 수 있습니다.

TN3270 서버를 구성하려면 노드 창에서 다음 단계를 수행하십시오.

1. TN 서버 액세스 레코드를 정의하십시오.

- a. 서비스 메뉴에서 **TN** 서버를 선택하십시오.

Linux용 Communications Server는 TN 서버 창을 표시하는데, 위쪽 분할창에는 구성된 모든 TN 서버 액세스 레코드가 나열되고 아래쪽 분할창에는 TN 서버 연결 레코드가 나열됩니다.

- b. TN3270 서버 액세스 레코드가 있는 분할창을 선택하고 새로 작성 버튼을 누르십시오.

Linux용 Communications Server는 TN 서버 액세스 대화상자를 표시합니다.

- c. 대화상자의 필드들에 적당한 값을 입력하십시오.

- d. 확인을 눌러 TN 서버 액세스 레코드를 정의하십시오. TN 서버 창에 레코드가 나타납니다.

2. TN 서버 연관 레코드를 정의하십시오.

- a. TN 서버 창에서 연관 레코드를 포함하는 분할창을 선택하고 추가 버튼을 누르십시오.

Linux용 Communications Server는 TN 서버 연결 레코드 대화상자를 표시합니다.

- b. 대화상자의 필드들에 적당한 값을 입력하십시오.
 - c. 확인을 눌러 TN 서버 연관 레코드를 정의하십시오. TN 서버 창에 레코드가 나타납니다.
3. 강제로라도 프린터 응답이 필요한 경우에는 모든 3270 세션에 대한 활성화 지속 방법을 지정하고 TN3270 클라이언트에 대한 권한을 확인하는 데 사용되는 취소 목록이 있는 외부 LDAP 서버에 액세스하는 방법을 지정하거나, TN3270 SLP(서비스 위치 프로토콜)를 사용하고 TN 서버 고급 매개변수 대화 상자를 사용하여 이를 수행하십시오.

TN 서버용 SSL 지원 구성에 대한 자세한 내용은 <http://www.ibm.com/software/network/commsserver/support/>에 있는 IBM Communications Server 지원 웹 페이지를 참조하십시오.

TN 방향 재지정자 구성

Linux용 Communications Server TN 방향 재지정자 기능은 TN3270, TN3270E, TN5250 및 VT 클라이언트(통칭해서 텔넷 클라이언트라고 함)에 대한 패스-스루 TCP/IP 호스트 액세스를 제공합니다. 텔넷 사용자는 TCP/IP 연결을 통해 Linux용 Communications Server와 통신하며, Linux용 Communications Server는 다른 TCP/IP 연결을 통해 호스트와 통신합니다. 이 방법은 필요할 때 보안 소켓 계층(SSL) 보안 검사를 사용할 수 있도록 하며 완전한 사용자-호스트 연결에서는 사용하지 않습니다. 예를 들어 다음을 지정하십시오.

- 클라이언트가 확인이 필요하지 않은 TCP/IP LAN을 통해 Linux용 Communications Server에 연결되어 있지만 SSL을 요구하는 원격 호스트에도 연결되어 있을 경우, Linux용 Communications Server와 호스트 사이의 TCP/IP 연결을 통해 SSL을 사용할 수 있습니다. 이는 모든 클라이언트에 대해 보안이 한 번 검사되고, 각 클라이언트는 보안 정보를 제공할 필요가 없음을 의미합니다.
- Linux용 Communications Server가 호스트와 같은 사이트에 설치되어 있지만 클라이언트가 외부 사이트에서 연결되는 경우, SSL 소프트웨어를 호스트에 설치하지 않아도 Linux용 Communications Server와 클라이언트 간 연결을 통해 SSL을 사용할 수 있습니다.

TN 방향 재지정자 구성

TN 방향 재지정자 액세스를 구성하려면 먼저 84 페이지의 『노드 구성』에서 설명된 대로 로컬 노드를 정의해야 합니다. 다음 정보가 필요합니다.

TN 방향 재지정자 구성

- 특정 클라이언트만 허용된 경우, 클라이언트의 TCP/IP 이름이나 주소가 필요합니다.
- 클라이언트가 TN 방향 재지정자 노드에 연결하기 위해 사용하는 TCP/IP 포트 번호.
- 호스트의 TCP/IP 이름 또는 주소
- TN 방향 재지정자 노드가 호스트에 연결하기 위해 사용하는 TCP/IP 포트 번호.
- SSL 데이터 암호화 여부와 관계없이, 클라이언트와 TN 방향 재지정자 노드 사이에 클라이언트 인증과 서버 인증이 필요합니다. (이 옵션은 이를 지원하는 데 필요한 추가 소프트웨어를 설치한 경우에만 사용할 수 있습니다.)
- SSL 데이터 암호화 여부는 TN 방향 재지정자 노드와 호스트 사이에 필요합니다.

TN 방향 재지정자 디폴트 레코드는 모든 TN 방향 재지정자 클라이언트 세션에서 사용되는 매개변수를 정의합니다. 각 클라이언트 TCP/IP 포트 번호에 대해 단일 디폴트 레코드를 정의할 수 있습니다.

TN 방향 재지정자를 구성하려면, 노드 창에서 다음 단계를 수행하여 TN 방향 재지정자 액세스 레코드를 정의하십시오.

1. 서비스 메뉴에서 **TN 서버**를 선택하십시오.

Linux용 Communications Server는 구성된 모든 TN3270 서버 액세스 레코드, TN3270 서버 연결 레코드 및 TN 방향 재지정자 액세스 레코드가 나열된 TN 서버 창을 표시합니다.

2. TN 방향 재지정자 액세스 레코드가 들어 있는 분할창을 선택하고 새로 작성 버튼을 누르십시오.

Linux용 Communications Server는 TN 방향 재지정자 액세스 대화상자를 표시합니다.

3. 대화상자의 필드들에 적당한 값을 입력하십시오.
4. 확인을 눌러 TN 방향 재지정자 액세스 레코드를 정의하십시오. TN 서버 창의 TN 방향 재지정자 분할창에 레코드가 나타납니다.

주: 노드의 SNA 자원을 사용하지는 않지만 TN 방향 재지정자를 사용할 수 있도록 SNA 노드가 활동 중이어야 합니다.

Linux용 Communications Server 사용 불가능화

Linux용 Communications Server 소프트웨어를 사용 불가능하게 하면 Linux용 Communications Server 노드 및 연관된 연결 구성요소가 자동으로 종료됩니다. Linux용 Communications Server를 사용 불가능하게 하면 다른 프로세스(예: LUA 응용프로그램)는 이 서버에서 Linux용 Communications Server 자원을 사용할 수 없습니다.

일반적으로 사용자가 사용을 완료하면 개별 서비스를 종료해야 하며 Linux용 Communications Server 활동이 없을 때만 시스템을 사용 불가능하게 해야 합니다.

사용자가 활동 중인 동안 Linux용 Communications Server를 사용 불가능하게 해야 하는 경우 사용자에게 Linux용 Communications Server가 중지될 것임을 경고하고 소프트웨어를 사용 불가능하게 하기 전에 사용자가 활동을 완료할 수 있는 충분한 시간을 제공하십시오.

Linux용 Communications Server 소프트웨어를 사용 불가능하게 할 경우 APPC, CSV, LUA, NOF 또는 MS API를 사용하는 응용프로그램에 COMM_SUBSYSTEM_ABENDED 리턴 코드가 통지되며 CPI-C 응용프로그램에는 CM_PRODUCT_SPECIFIC_ERROR 리턴 코드가 통지됩니다.

Linux용 Communications Server를 사용 불가능하게 하려면 Linux 명령 프롬프트에서 다음 명령을 입력하십시오.

```
sna stop
```

Linux용 Communications Server가 사용 불가능하게 되면 **sna stop**은 0의 종료 코드를 리턴합니다. 다른 종료 코드는 오류가 발생했으며 Linux용 Communications Server 소프트웨어가 사용 불가능하게 되지 않았음을 나타냅니다. 종료 코드 값에 대한 자세한 내용은 *Communications Server for Linux Diagnostics Guide*를 참조하십시오.

Linux용 Communications Server 자동 시작

Linux용 Communications Server가 완벽한 작업 상태로 전환되는 과정은 몇 단계로 나뉘어 살펴볼 수 있습니다.

- Linux용 Communications Server 소프트웨어 사용 가능화
- SNA 노드 초기화
- 노드에 구성된 포트 및 링크 스테이션의 활성화

각 해당 단계는 개별적으로 관리할 수 있으며 필요에 따라 시동 시에 모든 단계를 수행할 수 있습니다. 이 절의 나머지 부분에서는 이 단계가 설명되어 있으며 시동 시에 단계를 제어하는 방법도 설명되어 있습니다.

Linux용 Communications Server 사용 가능화

Linux용 Communications Server는 운영을 위해 여러 개의 커널 모듈을 로드해야 합니다. 이 모듈은 Linux용 Communications Server 소프트웨어를 처음으로 사용 가능하게 할 때 로드되며 커널 실행 레벨을 변경하는 경우에만 로드 해제됩니다.

Linux용 Communications Server는 또한 여러 개의 디먼(프로그램)이 실행되도록 요구합니다. 이 디먼이 실행될 때까지 Linux용 Communications Server를 구성 또는 사용할 수 없습니다.

- 디먼을 시작하고 Linux용 Communications Server 소프트웨어를 사용 가능하게 하려면 **sna start** 명령을 실행하십시오.
- 디먼을 중지하고 Linux용 Communications Server 소프트웨어를 사용 불가능하게 하려면 **sna stop** 명령을 실행하십시오.

SNA 노드 초기화

Linux용 Communications Server 소프트웨어가 사용 가능하게 되면 보통 Motif 관리 프로그램 `xsnaadmin`을 사용하여 SNA 노드 및 해당 자원을 구성할 수 있습니다. 하지만 초기화될 때까지 노드를 사용할 수 없습니다. **snaadmin init_node** 명령을 사용하여 명령행 관리 프로그램 또는 Motif 관리 프로그램 `xsnaadmin`에서 노드를 초기화할 수 있습니다.

포트 및 링크 스테이션의 활성화

다른 상황에서도 포트 및 링크 스테이션이 시작될 수 있도록 구성할 수 있습니다.

- 운영자 개입만 해당
- 요구가 있을 때(응용프로그램이 시작되어 링크의 자원을 사용하는 경우)
- 노드 시작 시(**snaadmin init_node** 명령이 실행되거나 Motif 관리 프로그램에서 노드가 시작되는 경우).

운영자가 시작하는 것이 디폴트이지만 명령행 관리 프로그램 또는 Motif 관리 프로그램을 사용하여 특정 포트 또는 링크에 맞게 시작을 변경할 수도 있습니다.

주: 포트를 시작하면 포트가 다른 컴퓨터로부터 호출을 받지만 전송 호출은 허용하지 않습니다. 링크 스테이션을 시작한다는 것은 Linux용 Communications Server가 원격 컴퓨터 접속을 시도한다는 것을 의미합니다.

다시 시동 시 Linux용 Communications Server 시작

다른 Linux 서비스와 마찬가지로, Linux용 Communications Server는 다시 시동할 때 사용 가능하게 됩니다. 즉, 다시 시동된 후 디폴트로 **sna start** 명령이 실행되지만 SNA 노드는 시작되지 않습니다.

이 초기화는 Linux용 Communications Server 시동 시간 초기화 스크립트 (`/etc/rc.d/init.d/snastart`)에서 수행됩니다. 시작 스크립트에 대해 일반적이므로 이것은 여러 시동 레벨에 대해 `/etc/rc?.d/init.d/snastart`로 링크됩니다.

`/etc/rc.d/init.d/snastart`를 편집하여 다시 시동 시 발생하는 결과를 변경할 수 있습니다. 가장 일반적인 변경은 노드의 초기화를 추가하는 것입니다. 이에 대한 **snaadmin**

init_node 명령이 파일에 이미 포함되어 있지만 주석이 표시되어 있기 때문에 간단히 주석만 제거하면 됩니다. 이 명령을 포함시키면 노드 시작 시에 활성화되도록 구성된 포트 또는 링크 스테이션의 활성화가 트리거됩니다.

Linux용 Communications Server를 사용하는 응용프로그램은 노드 초기화가 끝날 때까지 시작해서는 안 됩니다. 필요하다면 두 가지 방법 중 하나를 사용하여 시동 시에 이 응용프로그램을 자동으로 시작하고 시동 시에 실행해야 하는 다른 **snaadmin** 명령을 추가할 수 있습니다.

- **/etc/rc.d/init.d/snastart** 끝에서 **snaadmin init_node** 명령 뒤에 명령을 추가하십시오.
- Linux용 Communications Server 시작 후에 실행되도록 95보다 큰 숫자를 사용하여 **/etc/rc?.d/init.d** 스크립트를 작성하고 이 스크립트에 명령을 추가하십시오.

주: **/etc/rc.d/init.d/snastart** 파일에 대한 변경사항은 Linux용 Communications Server를 이후 버전으로 업그레이드할 때 저장되지 않습니다. 업그레이드 후에 다시 적용할 수 있도록 변경 사본을 항상 보관하십시오.

제 9 장 Linux용 Communications Server 및 SNA에 대한 정보 자원

이 장에서는 SNA 기술과 IBM의 여러 네트워크 제품 및 서비스 정보를 제공하는 SNA 라이브러리의 자원에 대해 설명합니다. 네트워크 포럼에서 사용할 수 있는 정보에 대해서도 설명합니다.

SNA 라이브러리

SNA 라이브러리에는 다음 주제에 대한 소개 및 자세한 정보를 제공하는 영업용 책자, 서적, 사용자 안내서 및 학습서가 들어 있습니다.

- SNA 이론
- SNA 제품
- 제품 구현
- 시스템 및 네트워크의 구성
- SNA 응용프로그램 프로그램과 API
- 전체적인 계획, 성능 및 튜닝
- 문제점 진단
- 네트워크 관리
- 네트워크 보안

모든 IBM 서적은 IBM 담당자, 해당 지역 서비스를 담당하는 IBM 영업소를 통해 주문하거나, IBM에 직접 1-800-879-2755로 전화하여 주문할 수 있습니다.

추가 서적에 관한 정보는 IBM 담당자에게 문의하십시오.

네트워크에서 이용할 수 있는 정보

정보 교환을 장려하기 위해 IBM은 전자 포럼 및 게시판을 후원합니다. 또한 인터넷에 홈 페이지를 게시하고 WWW(World Wide Web)에서 액세스할 수 있는 온라인 문서를 제공합니다.

IBMLink™를 통한 제품 지원

IBMLink 포럼은 IBM이 소유한 네트워크를 통해 제공됩니다. 라이선스가 있는 IBM 제품을 보유한 고객들이 기술적인 문제점 및 해당 시스템과 네트워크에 관련된 기타 문제점들을 해결할 수 있도록 고안되었습니다. IBM 직원이 질문에 답변하며 IBM 고객간의 온라인 토의를 중재합니다.

IBMLink에 관한 자세한 정보는 <http://www.ibm.link.ibm.com>을 참조하십시오.

IBM 홈 페이지에 있는 정보

인터넷에서 다양한 IBM 홈 페이지를 통해 포럼에 액세스할 수 있습니다. 포괄적인 도움을 받을 수 있도록 IBM 주요 홈 페이지를 인터넷과 WWW(World Wide Web)의 정보 센터를 탐색하는 데 사용할 수 있습니다. 주요 홈 페이지는 <http://www.ibm.com>을 사용하여 액세스할 수 있습니다.

<http://www.ibm.com/software/network>에서 Linux용 Communications Server를 비롯하여 IBM 네트워킹 소프트웨어에 관한 정보에 액세스할 수 있습니다. Linux용 Communications Server에 관한 정보는 <http://www.ibm.com/software/network/commsserver>에 있습니다.

Linux용 Communications Server 지원에 대한 자세한 정보는 <http://www.ibm.com/software/network/commsserver/support>를 참조하십시오.

다운로딩에 대한 정보

<http://www.redbooks.ibm.com>에서 Redbook 책자를 다운로드받을 수 있습니다.

IBM 소프트웨어에 대한 정보는 <http://www.ibm.com/software>에서 액세스할 수 있습니다. 이 웹 사이트에서 Linux용 Communications Server 및 모든 IBM 소프트웨어 서버 관련 페이지로 링크할 수 있습니다.

추천 문서

SNA에 대한 이해를 돕기 위해, 다음 책에서는 SNA 이론과 Linux용 Communications Server의 실제 사용에 대해 설명합니다. 이 책은 SNA에 대한 정통 지식을 얻기 위한 시작점 또는 이 주제의 자세한 내용을 필요로 하는 초심자와 전문가 모두에게 도움이 되는 책입니다.

- *Systems Network Architecture: Technical Overview*(GC30–3073)
- *IBM Communications Server for Linux Administration Guide*

자세한 내용은 해당 지역 IBM 담당자에게 문의하십시오.

부록. 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. IBM은 다른 국가에서 이 책에 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 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 라이선스 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 일체의 보증 없이 이 책을『현상태대로』 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 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 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들어, 사용료 지불 등)하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 단계의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 반드시 테스트하지 않았으므로, 이들 제품과 관련된 성능의 정확성, 호환성 또는 기타 주장에 대해서는 확인할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권: 이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용 없이 이들 샘플 프로그램을 어떠한 형

태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건 하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 귀하는 IBM의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용없이 이러한 샘플 응용프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다. ® (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp. 샘플 프로그램에서 파생됩니다. ® Copyright IBM Corp. 2000, 2005, 2006. All rights reserved.

상표

다음 용어는 미국과 기타 국가에서 IBM Corporation의 상표입니다.

Advanced Peer-to-Peer Networking	Power5
AIX	pSeries
Application System/400	S/390
AS/400	SP
CICS	System/370
IBM	System/390
MQSeries	SAA
MVS	Systems Application Architecture
MVS/ESA	VTAM
MVS/XA	WebSphere
NetView	z/OS
OpenPower	z9
OS/2	zSeries

다음은 타사의 상표 또는 등록상표입니다.

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 로고, Intel Inside, Intel Inside 로고, Intel Centrino, Intel Centrino 로고, Celeron, Intel Xeon, Intel SpeedStep, Itanium 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation 또는 자회사의 상표 또는 등록상표입니다.

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 및 서비스표입니다.

참고 서적

다음 IBM 서적은 이 라이브러리에서 언급된 주제에 관한 정보를 제공합니다. 이 서적은 다음과 같은 광범위한 주제 영역으로 구분됩니다.

- Linux용 Communications Server, 버전 6.2.2
- 시스템 네트워크 아키텍처(SNA)
- 호스트 구성
- z/OS Communications Server
- 전송 제어 프로토콜/인터넷 프로토콜(TCP/IP)
- X.25
- 고급 프로그램간 통신(APPC)
- 프로그래밍
- 기타 IBM 네트워킹 주제

Linux용 Communications Server 라이브러리의 서적에 대해서는 간단한 설명이 제공됩니다. 기타 서적에 대해서는 제목, 주문 번호가 제공되며, 일부 경우에는 이 책의 본문에 사용된 축약된 제목이 제공됩니다.

Linux용 Communications Server 버전 6.2.2 서적

Linux용 Communications Server 라이브러리는 다음 서적으로 이루어져 있습니다. 더불어 이 문서의 소프트웨어 버전이 CD-ROM으로 제공됩니다. CD-ROM의 소프트웨어 파일에 액세스하기 위한 정보는 *Linux용 IBM Communications Server Communications Server 빠른 시작* 참조하십시오. 이 소프트웨어 책을 사용자 시스템에 설치하려면 9-15MB의 하드 디스크 공간이 필요합니다(설치하는 자국어 버전에 따라 다름).

- *Linux용 IBM Communications Server 빠른 시작*(GA30-2285-01)

이 책은 지원되는 네트워크 특성, 설치, 구성 및 운영에 관한 정보를 포함하고 있으며 Linux용 Communications Server에 대해 개괄적으로 설명합니다.

- *Linux용 IBM Communications Server 관리 안내서*(SA30-2286-01)

이 책은 SNA 및 Linux용 Communications Server 개요와 Linux용 Communications Server 구성 및 운영에 관한 정보를 제공합니다.

- *Linux용 IBM Communications Server 명령 참조서*(SA30-3080-01)

이 책은 SNA 및 Linux용 Communications Server 명령에 관한 정보를 제공합니다.

- *Linux용 IBM Communications Server CPI-C 프로그래머 안내서(A30-3083-01)*

이 책은 숙련된 “C” 또는 Java 프로그래머에게 Linux용 Communications Server CPI 통신 API를 사용하여 SNA 트랜잭션 프로그램을 작성할 수 있는 정보를 제공합니다.

- *Linux용 IBM Communications Server APPC 프로그래머 안내서(SA30-3082-01)*

이 책에는 고급 프로그램간 통신(APPC)을 사용하여 응용프로그램을 작성하는 데 필요한 정보가 들어 있습니다.

- *Linux용 IBM Communications Server LUA 프로그래머 안내서(SA30-3085-01)*

이 책에는 일반 LU 응용프로그램 프로그래밍 인터페이스(LUA)를 사용하여 응용프로그램을 작성하는 데 필요한 정보가 들어 있습니다.

- *Linux용 IBM Communications Server CSV 프로그래머 안내서(SA30-3084-01)*

이 책에는 공통 서비스 Verb(CSV) API를 사용하여 응용프로그램을 작성하는 데 필요한 정보가 들어 있습니다.

- *Linux용 IBM Communications Server MS 프로그래머 안내서(SA30-3086-01)*

이 책에는 관리 서비스(MS) API를 사용하여 응용프로그램을 작성하는 데 필요한 정보가 들어 있습니다.

- *Linux용 IBM Communications Server NOF 프로그래머 안내서(SA30-3087-01)*

이 책에는 노드 연산자 기능(NOF) API를 사용하여 응용프로그램을 작성하는 데 필요한 정보가 들어 있습니다.

- *Linux용 IBM Communications Server 진단 안내서(GA30-2279-01)*

이 책에서는 SNA 네트워크 문제점 해결에 대한 정보를 제공합니다.

- *Linux용 IBM Communications Server APPC Application Suite 사용자 안내서(SA30-3081-01)*

이 책은 Linux용 Communications Server와 함께 사용되는 APPC 응용프로그램에 관한 정보를 제공합니다.

- *Linux용 IBM Communications Server 용어집(GA30-2281-01)*

이 책은 Linux용 IBM Communications Server 라이브러리 전반에 걸쳐 사용되는 용어 및 정의에 대한 포괄적인 목록을 제공합니다.

시스템 네트워크 아키텍처(SNA) 서적

다음 책에는 SNA 네트워크에 대한 정보가 들어 있습니다.

- 시스템 네트워크 아키텍처(SNA): *Format and Protocol Reference Manual—Architecture Logic for LU Type 6.2*(SC30-3269)
- 시스템 네트워크 아키텍처(SNA): *Formats*(GA27-3136)
- 시스템 네트워크 아키텍처(SNA): *Guide to SNA Publications*(GC30-3438)
- 시스템 네트워크 아키텍처(SNA): *Network Product Formats*(LY43-0081)
- 시스템 네트워크 아키텍처(SNA): *Technical Overview*(GC30-3073)
- 시스템 네트워크 아키텍처(SNA): *APPN Architecture Reference*(SC30-3422)
- 시스템 네트워크 아키텍처(SNA): *Sessions between Logical Units*(GC20-1868)
- 시스템 네트워크 아키텍처(SNA): *LU 6.2 Reference—Peer Protocols*(SC31-6808)
- 시스템 네트워크 아키텍처(SNA): *Transaction Programmer's Reference Manual for LU Type 6.2*(GC30-3084)
- 시스템 네트워크 아키텍처(SNA): *3270 Datastream Programmer's Reference*(GA23-0059)
- *Networking Blueprint Executive Overview*(GC31-7057)
- 시스템 네트워크 아키텍처(SNA): *Management Services Reference*(SC30-3346)

호스트 구성 서적

다음 책에는 호스트 구성에 대한 정보가 들어 있습니다.

- *ES/9000, ES/3090 IOCP User's Guide Volume A04*(GC38-0097)
- *3174 Establishment Controller Installation Guide*(GG24-3061)
- *3270 Information Display System 3174 Establishment Controller: Planning Guide*(GA27-3918)
- *OS/390 Hardware Configuration Definition (HCD) User's Guide*(SC28-1848)

z/OS Communications Server 서적

다음 책에는 z/OS Communications Server에 대한 정보가 들어 있습니다.

- *z/OS V1R7 Communications Server: SNA Network Implementation Guide*(SC31-8777)
- *z/OS V1R7 Communications Server: SNA Diagnostics*(Vol 1: GC31-6850, Vol 2: GC31-6851)
- *z/OS V1R6 Communications Server: Resource Definition Reference*(SC31-8778)

TCP/IP 서적

다음 책에는 TCP/IP 네트워크 프로토콜에 대한 정보가 들어 있습니다.

- *z/OS V1R7 Communications Server: IP Configuration Guide*(SC31-8775)
- *z/OS V1R7 Communications Server: IP Configuration Reference*(SC31-8776)
- *z/VM V5R1 TCP/IP Planning and Customization*(SC24-6125)

X.25 서적

다음 책에는 X.25 네트워크 프로토콜에 대한 정보가 들어 있습니다.

- *Communications Server for OS/2 Version 4 X.25 Programming*(SC31-8150)

APPC 서적

다음 책에는 고급 프로그램간 통신(APPC)에 대한 정보가 들어 있습니다.

- *APPC Application Suite V1 User's Guide*(SC31-6532)
- *APPC Application Suite V1 Administration*(SC31-6533)
- *APPC Application Suite V1 Programming*(SC31-6534)
- *APPC Application Suite V1 Online Product Library*(SK2T-2680)
- *APPC Application Suite Licensed Program Specifications*(GC31-6535)
- *z/OS V1R2.0 Communications Server: APPC Application Suite User's Guide*(SC31-8809)

프로그래밍 서적

다음 책에는 프로그래밍에 대한 정보가 들어 있습니다.

- *Common Programming Interface Communications CPI-C Reference*(SC26-4399)
- *Communications Server for OS/2 Version 4 Application Programming Guide*(SC31-8152)

기타 IBM 네트워킹 서적

다음 책에는 Linux용 Communications Server와 관련된 기타 주제에 대한 정보가 들어 있습니다.

- *SDLC Concepts*(GA27-3093)
- 근거리 통신망 개념 및 제품: *LAN Architecture*(SG24-4753)
- 근거리 통신망 개념 및 제품: *LAN Adapters, Hubs and ATM*(SG24-4754)
- 근거리 통신망 개념 및 제품: *Routers and Gateways*(SG24-4755)

- 근거리 통신망 개념 및 제품: *LAN Operating Systems and Management*(SG24-4756)
- *IBM Network Control Program Resource Definition Guide*(SC30-3349)

색인

[가]

게이트웨이
정의 8
SNA 게이트웨이 8
경고 18
계획 워크시트 77
고급 프로그램간 통신(APPC) 7
관리 프로그램
명령행 16
Motif 15, 17
NOF API 16
광역 네트워크(WAN) 3
구성 20
계획 76
노드 84
백업 36
보기 82, 104
수정 82
암시적 다운스트림 102
연결성 85
원격 노드 96
파일 36, 37
포트 86
APPC 통신 92
CPI-C 부가 정보 98
DLUR 103
LEN 노드에 대한 상대방 LU 95
LU 6.2 94, 98
LU 유형 0-3 89
SNA 게이트웨이를 위한 다운스트림 LU들 101
TN 방향 재지정자 디폴트 112
TN 서버 디폴트 110
TN 서버 연관 레코드 110
구성 보기 82
구성 서버 83
제거 84
추가 84
구성 수정 82
구성 창, Windows의 원격 API 클라이언트
고급 매개변수 66
매개변수 65
구성요소 관리 79
국제 표준 협회(ISO) 24

근거리 통신망(LAN) 3
기능 6
기능적 요구사항 19
기술 요구사항 21
기억영역 요구사항 24

[나]

네트워크
경고 18
계획 19, 20, 21, 24
관리 17
를 통해 사용할 수 있는 정보 117
명명 규칙 24
부속 영역 2
지원 2
피어 3
노드
구성 84
다운스트림 106
로컬 105
업스트림 104
원격 95, 96
초기화 114
노드 창 79
논리 장치(LU) 4, 103

[다]

다시 시동 시 Linux용 Communications Server 시작 114
다운스트림 노드 104
다운스트림 컴퓨터 101
다운스트림 LU
구성 103
하드웨어 예제 101
SNA 게이트웨이 101
대화상자 79
데이터 링크 제어 옵션 3
데이터 스트림 6
도구 모음 버튼 82
도메인 6
도메인 매개변수 65
도움말
Motif 관리 프로그램 15

독립 LU 구성 94
동기 데이터 링크 제어(SDLC) 86
동적 구성 15

[라]

라이센스가 있는 프로그램 29
AIX의 원격 API 클라이언트 56
Linux의 원격 API 클라이언트 43
System z용 Linux의 원격 API 클라이언트 51
로컬 LU, 정의 94
로컬/원격 트랜스패런시 5
릴리스 정보 35
링크 스테이션
포트에서 정의 86
활성화 114

[마]

마스터 서버 7
매체 액세스 제어(MAC) 87
메모리 요구사항 24
명령행 관리 프로그램 16
명령(Verb) 6
명명 규칙 24
문서, 온라인 117
문제점
데이터 18
진단 도구 18
문제점 해결 도구 18
물리적 장치(PU) 8

[바]

백업
구성 파일 36
복원 36
백업 마스터 서버 83
백업 서버 7
별명, 상대방 LU에 대한 별명 정의 96
보안 소켓 계층(SSL)
데이터 암호화 35, 110, 112
서버 인증 35, 110, 112
클라이언트 인증 110, 112
보안 옵션 16
복수 도메인 지원-네트워크 관리 백터 전송(MDS-NMVT) 17
분기 네트워크 노드 9

분산 처리
응용프로그램 지원 7
환경 3

[사]

사전 설치 TASK 26
상대방 응용프로그램 7
상대방 LU
구성 95
별명 96
와일드카드로 정의 97
원격 노드 96
상대 명령 17
서버 7
제거 84
추가 84
Linux 하드웨어 요구사항 22
서버 이름 65
서버, 탭넷 12
서브루틴 6
서비스 액세스점(SAP) 22, 87
설치
기존 패키지의 세부사항 26
설치 후 유지보수 TASK 34
준비 26
AIX의 원격 API 클라이언트 56, 57
Linux의 원격 API 클라이언트 43
System z용 Linux의 원격 API 클라이언트 51
설치 요구사항 22
설치 후 절차 34
세션
경로지정 16
속도 16
지원 5
U 모양의 5
소프트웨어 기능 8
소프트웨어 요구사항 22, 23
AIX의 원격 API 클라이언트 55
Linux의 원격 API 클라이언트 42
System z용 Linux의 원격 API 클라이언트 49
Windows의 원격 API 클라이언트 61
속도, 세션 레벨 16
시스템 응용프로그램 아키텍처(SAA) 6

[아]

- 암시적 다운스트림 LU 구성 102
- 언어 환경 변수 26, 56
 - Linux의 원격 API 클라이언트 43
 - System z용 Linux의 원격 API 클라이언트 50
- 업스트림 노드 104
- 엔트리 포인트 17
- 연결 네트워크 16
- 연결 네트워크, 구성 88
- 연결성
 - 구성 85
 - 옵션 3
- 온라인
 - 도움말 15
 - 문서 117
 - 토의 그룹 117
 - 포럼 118
 - APPC 118
- 와일드카드 97
- 요구사항
 - 메모리 및 기억영역 24
 - 설치 22
 - 소프트웨어 22, 23
 - 요원 및 기술 21
 - 운영 체제 22
 - HTTPS 23
 - WebSphere Application Server 23
- 요원 요구사항 21
- 운영 체제 요구사항 22
- 워크시트, 계획 77
- 원격 노드
 - 구성 95, 96
 - 상대방 LU 구성 96
- 원격 API 클라이언트
 - AIX 소프트웨어 요구사항 55
 - AIX 하드웨어 요구사항 55
 - Linux 소프트웨어 요구사항 42
 - Linux 하드웨어 요구사항 41
 - System z용 Linux 소프트웨어 요구사항 49
 - System z용 Linux 하드웨어 49
- 응용프로그램 프로그래밍 인터페이스(API) 5
- 이더넷
 - 링크 구성 87
 - SAP 대화 상자 88
- 이전 레벨의 Linux용 Communications Server에서 이주 27
- 인터페이스 선택사항 15

[자]

- 자원
 - 관리 79
 - 삭제 81
 - 시작 81
 - 요구사항 21
 - 정보 117
 - 정의 81
 - 종료하기 81
 - 항목 82
- 자원 삭제 81
- 자원 시작하기 81
- 자원 종료하기 81
- 자원 창의 버튼 82
- 자원 추가 81
- 재연결 시간종료 매개변수 67
- 적용 세션 레벨 속도 16
- 전송 매체 22
- 정보 자원 117
- 조회 명령 17
- 종료 명령 113
- 중속 LU 6.2 98
- 중속 LU 리퀘스터(DLUR) 3
- 중속 LU 서버(DLUS) 104
- 중속 LU의 동적 정의(DDDLU) 4
- 중계 세션 경로지정(ISR) 16

[차]

- 창 79
- 최대 브로드캐스트 시도 수 매개변수 66
- 추천 문서 118

[카]

- 클라이언트 7
- 클라이언트 소프트웨어
 - 사용자 조정 72
 - 설치 제거 73
 - 업그레이드 72
 - 재설치 72
- 클라이언트/서버
 - 구성 83
- 클라이언트/서버 설치 34
- 클라이언트/서버 지원 6

[타]

- 타스크 시트 77
- 토의 그룹, 온라인 117
- 통신용 SAA 공통 프로그래밍 인터페이스(CPI-C) 6
- 트랜스패런시, 로컬/원격 5
- 트랜잭션 프로그램(TP) 6

[과]

- 포럼, 온라인 118
- 포컬(focal) 포인트 17
- 포트
 - 활성화 114
- 포트 구성 86, 88, 89
- 포트 및 링크 스테이션의 활성화 114
- 프로시저
 - Windows에 원격 API 클라이언트 설치 63, 67
- 피어
 - 네트워크 3
- 피어 서버 7

[하]

- 하드 디스크 기억영역 24
- 하드웨어
 - 링크 23
- 하드웨어 링크 23
- 하드웨어 요구사항
 - 서버 22
 - AIX의 원격 API 클라이언트 55
 - Linux의 원격 API 클라이언트 41
 - System z용 Linux의 원격 API 클라이언트 49
 - Windows의 원격 API 클라이언트 61
- 하위 네트워크(LAN) 노드 87
- 한 도메인에서 여러 개의 서버 7
- 호스트
 - 부속 영역 네트워크 2
 - APPN 네트워크 내의 3
 - LU 지원 4
- 호출 6

[숫자]

- 1차 RUI 5
- 3270 LU
 - 정의 108
- TN 서버를 위한 13, 107

A

- AIX에 원격 API 클라이언트 설치 56, 57
- AIX에서 원격 API 클라이언트 설치 제거 59
- API
 - 지원 5
 - Linux용 Communications Server 관리용 16
 - Linux용 Communications Server 유형 6
- APPC
 - 구성 92
 - 독립 LU 6.2 94
 - 분산 응용프로그램 지원 7
 - 온라인 118
 - 중속 LU 6.2 98
 - application suite 8, 11
- APPC Application Suite 11
- APPN
 - 경로지정 16
 - 구성 93, 94
 - 끝 노드 3
 - 네트워크의 단편화 24
 - 노드 유형 3
 - 독립 LU 지원 5
 - 동적 구성 15
 - 부속 영역 기능 3
 - 에 대한 응용프로그램 8, 11
 - 에 대한 호스트 지원 3
 - 연결 네트워크 16
 - DLUR 지원 3
 - network node(네트워크 노드) 3

B

- Branch Extender 9

C

- CPI 통신(CPI-C) 6
- CPI-C
 - 구성 98
 - 상호 이용성 18
 - API 6

D

- DDDLU(중속 LU의 동적 정의) 4
- DLC
 - 구성 88, 89

DLC (계속)

포트 구성 85

DLUR

구성 103

다운스트림 노드에 대한 지원 106

로컬 노드 상 105

설명 3

PU 구성 106

DLUS 104

E

Enterprise Extender

개요 11

링크 구성 88

포트 대화 상자 89

G

GSKIT

설치 제거, Windows의 원격 API 클라이언트 73

AIX의 원격 API 클라이언트 55, 58

Linux의 원격 API 클라이언트 42, 45

System z용 Linux의 원격 API 클라이언트 50, 52

Windows의 원격 API 클라이언트 69, 70

H

HACL 32

Host Access Class 라이브러리 32

HPR

ISR에 비해 16

HPR/IP 11

HTTPS

구성 32

요구사항 23

AIX의 원격 API 클라이언트 58

Linux의 원격 API 클라이언트 45

System z용 Linux의 원격 API 클라이언트 52

Windows의 원격 API 클라이언트 69, 70

I

IP 포트 대화 상자 89

ISO(국제 표준 협회) 24

ISR(중계 세션 경로지정) 16

J

Java

AIX의 원격 API 클라이언트 55

Linux의 원격 API 클라이언트 42

System z용 Linux의 원격 API 클라이언트 49

L

LAN 액세스 시간종료 매개변수 66

LAN(근거리 통신망) 3

LEN 노드

상대방 LU 구성 95

설명 3

원격 노드 식별 87

Linux에 원격 API 클라이언트 설치 43

Linux에서 원격 API 클라이언트 설치 제거 46

Linux용 Communications Server 문서 31

Linux용 Communications Server 사용 가능화

로컬 시스템 78

Linux용 Communications Server 사용 불가능화 113

Linux용 Communications Server 설치 29

Linux용 Communications Server 설치 제거 38

Linux용 Communications Server 소프트웨어 사용 가능화 113

Linux용 Communications Server 시작

시스템 시작 시에 자동 78

Linux용 Communications Server 실행 가능 프로그램의 경로 78

Linux용 Communications Server 실행 가능 프로그램의 디렉토리

78

Linux용 Communications Server 온라인 문서 31

Linux용 Communications Server 자동 시작 113

다시 시동 시 시작 114

포트 및 링크 스테이션의 활성화 114

Linux용 Communications Server 소프트웨어 사용 가능화 113

SNA 노드 초기화 114

Linux의 원격 API 클라이언트

기존 패키지의 세부사항 42

LU

구성 94, 98, 108

다운스트림 101, 103

명명 규칙 24

상대방, 정의 95

지원 4

폴 99, 100, 108, 109

LU 세션 데이터의 압축 5

LU 유형 0-3 89, 90

LU 폴

구성 91

LU 풀 (계속)
 보기 91
 정의 90

M

MAC(매체 액세스 제어) 87
MDS-NMVT(복수 도메인 지원-네트워크 관리 벡터 전송) 17
Motif 관리 프로그램
 관리 기능 17
 도움말 15
 사용 75
 설명 15

N

NOF API 16

O

Open Systems Interconnection(OSI) 24
OSI(Open Systems Interconnection) 24

P

PDF, 서적 보기 34
PU 집중 101
PU 집중기 8
PU(물리적 장치) 8

R

rpm 21

S

SAP(서비스 액세스점) 22, 87
SDK 소프트웨어
 Windows의 원격 API 클라이언트 62
SDLC 3
 구성 86
 중속 트래픽용 86
SNA
 라이브러리 117
SNA 게이트웨이
 개요 8
 구성 101
SNA 노드 초기화 114

snaadmin 프로그램 16
snastart 파일 78, 114
start 명령 78
System z용 Linux에 원격 API 클라이언트 설치 51
System z용 Linux에서 원격 API 클라이언트 설치 제거 54
System z용 Linux의 원격 API 클라이언트
 기존 패키지의 세부사항 50

T

TN 방향 재지정자
 개요 13
 구성 111
 디폴트 구성 112
 액세스 레코드 구성 112
TN 서버
 개요 12
 구성 107
 디폴트 구성 110
 복수 세션 지원 108
 사용자 107
 액세스 레코드 구성 110
 연관 레코드 구성 110
TN3270
 서버 12
 프로그램 12
TN3270 사용자 13, 107
TN3270 프로그램 107
TP(트랜잭션 프로그램) 6

U

U 모양의 세션 5
UDP 브로드캐스트 매개변수 65

W

WebSphere Application Server
 구성 32
 요구사항 23
Windows의 원격 API 클라이언트
 명령행에서 설치 67
 설치 62
 설치 프로그램을 사용한 설치 63
GSKIT 설치 제거 73

X

xsnaadmin 프로그램 15, 75



프로그램 번호: 5724-i34

GA30-2280-01

