

Linux용 IBM Communications Server



관리 안내서

버전 6.4

Linux용 IBM Communications Server



관리 안내서

버전 6.4

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 209 페이지의 부록 D 『주의사항』의 정보를 읽으십시오.

제 6 판(2009년 5월)

이 개정판은 새 개정판 또는 기술 관련 소식지에 명시되기 전까지 Linux용 IBM Communications Server 버전 6.4와 모든 후속 릴리스 및 수정판에 적용됩니다.

책을 주문하려면 IBM 담당자 또는 해당 지역의 IBM 지방 사무소로 문의하십시오. 다음 주소에서는 책을 구비하고 있지 않습니다.

IBM은 고객의 의견을 환영합니다. 아래 주소로 의견을 보내주십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

다음 방법 중 하나를 통해 의견을 보낼 수 있습니다.

• 팩스:

- 02-3781-5200

• 인터넷 메일:

- ibmkspoe@kr.ibm.com

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

목차

표	vii	계획 워크시트	43
그림	ix	타스크 시트	43
이 책의 사용 방법	xi	로컬 시스템에서 Linux용 Communications Server 사용 가능화 및 사용 불가능화	43
이 책의 사용자	xi	Linux용 Communications Server 프로그램의 경 로 지정.	44
이 책의 사용 방법	xii	Linux용 Communications Server 사용 가능화	44
이 책의 구성.	xii	Linux용 Communications Server 사용 불가능화	46
활자체 규약.	xiii	Motif 관리 프로그램 사용	47
그래픽 규약.	xiv	Motif 관리 프로그램 가동	48
이 릴리스의 새로운 사항.	xiv	자원 창.	48
새로운 기능.	xiv	자원 대화상자	55
사라진 기능	xv	상태 대화상자	57
추가 정보.	xv	도움말 창	57
제 1 장 SNA 용어 및 개념	1	명령행 관리 프로그램 사용	58
시스템 네트워크 아키텍처(SNA).	1	제 3 장 기본 구성 타스크	61
기본 SNA 개념	2	클라이언트/서버 기능 구성	61
네트워크 유형	2	노드 구성	62
SNA 노드	2	노드 구성 매개변수.	63
연결성	6	추가 구성	63
트랜잭션 프로그램	6	로깅 구성	64
응용프로그램 프로그래밍 인터페이스(API).	6	제 4 장 연결 구성요소 정의.	67
네트워크 액세스 가능 장치	7	DLC, 포트 및 연결 네트워크 정의	68
세션.	10	DLC, 연결 네트워크 및 포트 구성 매개변수	69
대화.	12	추가 구성	73
모드.	13	링크 스테이션 정의.	73
전송 경로 선택	14	링크 스테이션 구성 매개변수	75
서비스 클래스(COS)	14	추가 구성	81
기본 APPN 개념	14	DLUR PU 정의.	82
APPN 노드 유형	15	DLUR PU 구성 매개변수	82
APPN 제어점(CP)	18	다운스트림 노드에 대한 패스-스루 DLUR 매개변 수	84
자원 찾기	19	추가 구성	84
세션 경로지정	23	제 5 장 종속 LU 구성	85
Branch Extender	30	LU 유형 0-3 정의.	85
APPN 네트워크에서 부속 영역 네트워크에 액세스	31	LU 유형 0-3 구성 매개변수	86
제 2 장 Linux용 Communications Server 관리	33	추가 구성	87
Linux용 Communications Server 관리의 개요	33	LU 풀 정의	88
관리 책임	33	LU 풀 구성 매개변수	89
관리 도구	34		
관리 권한	41		
Linux용 Communications Server 구성 계획	42		

제 6 장 APPC 통신 구성	91
로컬 LU 정의	92
로컬 LU 구성 매개변수	93
추가 구성	93
원격 노드 정의	94
원격 노드 구성 매개변수	95
추가 구성	95
상대방 LU 정의	95
상대방 LU 구성 매개변수	96
상대방 LU의 링크 스테이션 경로지정 정의	98
추가 구성	99
TP 정의	99
서버에서의 TP 기동 매개변수	102
TP 정의 매개변수	106
모드 및 서비스 클래스 정의	107
모드 구성 매개변수	108
추가 구성	111
CPI-C 부가 정보 정의	112
CPI-C 구성 매개변수	112
추가 구성	114
APPC 보안 구성	114
세션 보안 구성	114
대화 보안 구성	115
보안 액세스 리스트 구성	116
제 7 장 사용자 응용프로그램 구성	117
제 8 장 패스-스루 서비스 구성	119
TN 서버 구성	119
TN 서버 액세스 레코드 구성	120
TN 서버 연결 레코드 구성	124
TN 방향 재지정자 구성	124
TN 방향 재지정자 액세스 레코드 구성	124
SNA 게이트웨이 구성	129
다운스트림 LU 구성 매개변수	130
추가 구성	131
DLUR 구성	131
제 9 장 NetView에서 Linux용 Communications Server 관리	133
호스트 NetView 프로그램 사용	133
NetView 화면 표시장치	134
명령 입력 영역 크기 변경	134
RCF 명령 구문 개요	135
대문자 및 이탤 문자	135
SPCF 사용	136
SPCF에서 사용되는 관리 명령의 제한사항	136

SPCF 명령 예제	137
UCF 사용	138
UCF 명령 구문	138
사용할 수 있는 명령	139
UCF 명령 예제	139
Linux 시스템 명령의 출력	140
명령 취소	140
UCF 보안	141

제 10 장 Linux용 Communications Server 클라이언트/서버 시스템 관리

클라이언트/서버 시스템 관리	143
클라이언트/서버 구성 변경	144
클라이언트를 다른 도메인으로 이동	145
IP 네트워킹 요구사항	146
IPv4 및 IPv6 주소지정	146
클라이언트/서버 구성의 호스트 이름	146
IP 포트 번호 설정	147
LAN 액세스 시간종료	148
원격 API 클라이언트에 대한 HTTPS 액세스	149
Windows에서 원격 API 클라이언트 관리	150
Windows의 원격 API 클라이언트 사용 가능화	151
Windows의 원격 API 클라이언트 상태 보기	151
Windows의 원격 API 클라이언트 사용 불가능화	152
Windows의 원격 API 클라이언트 구성	152
AIX 또는 Linux의 원격 API 클라이언트 관리	169
AIX 또는 Linux의 원격 API 클라이언트 사용 가능화 및 사용 불가능화	169
클라이언트 네트워크 데이터 파일(sna_clnt.net)	170
클라이언트 TP 정의	174
부록 A. 구성 계획 워크시트	175
노드 워크시트	175
APPN 네트워크 노드	175
APPN 끝 노드	176
APPN 브랜치 네트워크 노드	176
LEN 노드	177
연결성 워크시트	178
SDLC	178
토큰링	180
이더넷	182
QLLC(X.25)	183
다중 경로 채널	185
Enterprise Extender(HPR/IP)	185
패스-스루 서비스 워크시트	187
로컬 노드의 DLUR	187
다운스트림 노드에 대한 패스-스루 DLUR	187

SNA 게이트웨이	188	상표	211
TN 서버	188	참고 서적	213
TN 방향 재지정자	189	Linux용 Communications Server 버전 6.4 서적	213
사용자 응용프로그램 지원 워크시트	190	시스템 네트워크 아키텍처(SNA) 서적	215
APPC	191	호스트 구성 서적	215
CPI-C	193	z/OS Communications Server 서적	216
5250	194	TCP/IP 서적	216
3270	194	X.25 서적	216
LUA	195	APPC 서적	216
부록 B. 명령행에서 기동 가능 TP 구성	197	프로그래밍 서적	216
기동 가능 TP 정의의 파일 형식	199	기타 IBM 네트워킹 서적	217
부록 C. DDDLU용 TN3270 LU 모델 구성	207	색인	219
부록 D. 주의사항	209		

표

1. 활자체 규약 xiii
2. 표준 모드 및 COS 이름. 108
3. RCF 명령에서 이탈 문자 사용. 136

그림

1. SNA 부속 영역 네트워크	4	10. APPN 네트워크에서 노드 EN1으로부터 모든 노드로의 직접 링크에 필요한 정의	28
2. 복수 및 병렬 세션	12	11. 가상 노드를 사용한 직접 링크에 필요한 정의	29
3. 트랜잭션 프로그램 및 논리 장치 사이의 통신	13	12. Branch Extender	30
4. 샘플 APPN 네트워크의 일부.	16	13. Linux용 Communications Server 도메인 창	50
5. LEN 노드 디렉토리.	20	14. 노드 창.	51
6. 끝 노드 디렉토리	21	15. Linux용 Communications Server 도구 모음	54
7. 네트워크 노드 디렉토리	22	16. 샘플 대화상자.	56
8. 네트워크 노드의 네트워크 토폴로지 데이터베이 스	24	17. 샘플 상태 대화상자.	57
9. 공유 액세스 전송 기능(SATF)을 사용하는 APPN 네트워크	27	18. 샘플 도움말 창	58
		19. NetView 화면 예제	134

이 책의 사용 방법

이 책에서는 Linux를 실행 중인 컴퓨터가 시스템 네트워크 아키텍처(SNA) 네트워크에 있는 다른 노드와 정보를 교환할 수 있게 하는 IBM® 소프트웨어 제품인 Linux용 IBM Communications Server를 사용, 구성 및 관리하는 방법에 대해 소개합니다.

작동하는 하드웨어에 따라 다음과 같은 두 가지 Linux용 IBM Communications Server 설치가 있습니다.

Linux용 Communications Server

Linux용 Communications Server(프로그램 제품 번호 5724-i33)는 다음과 같은 시스템에서 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-i34)는 System z용 Linux를 실행하는 System z 메인프레임(s390 또는 s390x)에서 작동합니다.

이 책에서는 Linux용 Communications Server와 System z의 Linux용 Communications Server의 차이점을 명시할 경우를 제외하고, 두 가지 모두에 대해 Linux용 Communications Server를 사용합니다. 또한, 『Linux용 Communications Server 컴퓨터』는 Linux용 Communications Server를 실행하는 모든 종류의 컴퓨터입니다.

이 책은 버전 6.4의 Linux용 Communications Server에 적용됩니다.

이 책의 사용자

이 책은 Linux용 Communications Server를 사용하는 시스템 관리자와 응용프로그램 프로그래머를 위한 책입니다.

시스템 관리자

시스템 관리자는 Linux용 Communications Server를 설치하고 네트워크 연결에 맞게 시스템을 구성하며 시스템을 유지보수합니다. Linux용 Communications Server가 작동되는 하드웨어와 Linux 운영 체제에 익숙해야 합니다. 또한 시스템이 연결된 네트워크에 대한 지식이 있어야 하고 일반적인 SNA 개념도 알고 있어야 합니다.

응용프로그램 프로그래머

응용프로그램 프로그래머는 Linux용 Communications Server 프로그래밍 인터페이스를 사용하여 SNA 네트워크를 통해 데이터를 송수신하는 응용프로그램 및 트랜잭션을 설계하고 코드화합니다. SNA, 트랜잭션이나 응용프로그램이 통신하는 원격 프로그램, Linux 운영 체제 프로그래밍 및 운영 환경에 전반적으로 익숙해야 합니다.

응용프로그램 작성에 대한 자세한 정보는 각 API의 매뉴얼에 제공됩니다. Linux용 Communications Server 서적에 대한 추가 정보는 참고 서적을 참조하십시오.

이 책의 사용 방법

이 책은 Linux용 Communications Server를 사용, 구성 및 관리하는 방법에 대해 설명합니다.

이 책의 구성

이 책은 다음과 같이 구성됩니다.

- 1 페이지의 제 1 장 『SNA 용어 및 개념』: SNA 및 APPN(Advanced Peer-to-Peer Networking[®]) 개념에 대한 개요를 제공합니다.
- 33 페이지의 제 2 장 『Linux용 Communications Server 관리』: Linux용 Communications Server 관리 도구에 대해 설명하고 Linux용 Communications Server 구성 준비 방법, 서버에서 Linux용 Communications Server 소프트웨어를 사용 가능 및 사용 불가능하게 하는 방법, Motif 및 명령행 관리 프로그램 사용 방법에 대해 설명합니다.
- 61 페이지의 제 3 장 『기본 구성 TASK』: Linux용 Communications Server에 대한 메시지 로깅 구성, SNA 노드 구성 및 클라이언트/서버 조작 구성을 비롯하여 Linux용 Communications Server에 대한 기본 구성 TASK 수행 방법을 설명합니다.
- 67 페이지의 제 4 장 『연결 구성요소 정의』: Linux용 Communications Server 노드에 대한 연결 구성 방법을 설명합니다.
- 85 페이지의 제 5 장 『종속 LU 구성』: LU 유형 0-3 및 LU 풀에 대한 종속 LU(논리 장치) 구성 방법을 설명합니다.
- 91 페이지의 제 6 장 『APPC 통신 구성』: 고급 프로그램간 통신(APPC) 구성 방법을 설명합니다.
- 117 페이지의 제 7 장 『사용자 응용프로그램 구성』: 사용자 응용프로그램 구성 방법을 설명합니다.
- 119 페이지의 제 8 장 『패스-스루 서비스 구성』: 직접 연결되지 않은 로컬 시스템과 호스트 시스템 사이의 통신을 지원하는 패스-스루 서비스 구성 방법을 설명합니다.

- 133 페이지의 제 9 장 『NetView에서 Linux용 Communications Server 관리』: Linux용 Communications Server 원격 명령 기능(RCF)을 사용하여 Linux용 Communications Server를 관리하고 NetView를 실행하는 호스트를 통해 Linux용 Communications Server 노드에서 명령을 실행하는 방법을 설명합니다.
- 143 페이지의 제 10 장 『Linux용 Communications Server 클라이언트/서버 시스템 관리』: IBM 원격 API 클라이언트 구성 및 관리 방법을 설명합니다.
- 175 페이지의 부록 A 『구성 계획 워크시트』: Linux용 Communications Server의 구성 워크시트를 제공합니다.
- 197 페이지의 부록 B 『명령행에서 기동 가능 TP 구성』: 사용자나 TP 설치 프로그램 작성자가 기동 가능 TP를 정의할 수 있는 명령행 유틸리티에 대한 정보를 제공합니다.
- 207 페이지의 부록 C 『DDDLU용 TN3270 LU 모델 구성』에서는 클라이언트가 DDDLU를 사용하는 경우 호스트에서 TN3270 클라이언트 장치 유형과 LU 모델 간의 매핑을 변경할 수 있게 해주는 **tn3270dev.dat** 파일에 대해 설명합니다.

활자체 규약

표 1에는 이 문서에서 사용된 인쇄 스타일이 나와 있습니다.

표 1. 활자체 규약

특수 요소	활자체 샘플
강조어	삭제 전 파일 백업
문서 제목	<i>Linux용 IBM Communications Server 관리 안내서</i>
파일 또는 경로 이름	/usr/spool/uucp/myfile.bkp
프로그램 또는 응용프로그램	snaadmin
매개변수 또는 Motif 필드	<i>op 코드; LU 이름</i>
사용자가 입력할 수 있는 리터럴 값이나 선택사항 (디폴트 값 포함)	255; 노드 시동 시
Motif 버튼	상태
Motif 메뉴	서비스
Motif 메뉴 항목	노드 매개변수 구성
사용자 입력	0p1
컴퓨터 출력	닫기
명령 또는 Linux 유틸리티	define_node; cd
특정 유형의 모든 명령에 대한 일반 참조 사항	query_* (자원의 세부사항을 조회하는 모든 관리 명령을 나타냅니다.)
옵션 또는 플래그	-i
제공하는 값을 표시하는 변수	<i>filename; LU_name; user_ID</i>
리턴값	0; -1
3270 키	ENTER
키보드 키	Ctrl+D; Enter
16진 값	0x20
환경 변수	PATH
함수, 호출 또는 시작점	ioctl1
프로그래밍 Verb	GET_LU_STATUS

그래픽 규약

AIX, LINUX

이 기호는 AIX 또는 Linux 운영 체제에만 적용되는 텍스트 섹션의 시작을 표시하는 데 사용됩니다. 이것은 Linux 서버 및 AIX, Linux, pSeries용 Linux 또는 System z용 Linux에서 실행되는 IBM 원격 API 클라이언트에 적용됩니다.

WINDOWS

이 기호는 Windows의 IBM 원격 API 클라이언트에 적용되는 텍스트 섹션의 시작을 표시하는 데 사용됩니다.



이 기호는 운영 체제 특정 텍스트 섹션의 끝을 표시합니다. 이 기호 뒤에 오는 정보는 운영 체제에 관계없이 적용됩니다.

이 릴리스의 새로운 사항

Linux용 Communications Server 버전 6.4는 Linux용 Communications Server 버전 6.2.3, Linux용 Communications Server 버전 6.2.2, Linux용 Communications Server 버전 6.2.1 및 Linux용 Communications Server 버전 6.2를 대체합니다.

계속 지원되는 이 제품의 릴리스는 다음과 같습니다.

- Linux용 Communications Server 버전 6.2.3
- Linux용 Communications Server 버전 6.2.2
- Linux용 Communications Server 버전 6.2.1

이 제품의 다음 릴리스는 더 이상 지원되지 않습니다.

- Linux용 Communications Server 버전 6.2
- PRPQ 5799-RQA 또는 5799-RXL로 사용 가능했던 Linux용 Communications Server 버전 6.0.1(V6.0.1)

Linux용 Communications Server 버전 6.4는 IBM 원격 API 클라이언트 버전 6.4.0 또는 6.3.1.0과 함께 작동합니다.

새로운 기능

이번 릴리스에서는 다음 기능이 Linux용 Communications Server에 추가되었습니다.

- TN3270 장치 유형과 이 장치가 DDDLU를 사용하여 Linux용 Communications Server TN 서버를 통해 연결하는 경우, Linux용 Communications Server가 호스트에서 요청해야 하는 LU 모델 사이의 매핑을 지정할 수 있습니다. 그러면 각 유형의 장치에 대해 가장 알맞은 LU 모델을 사용할 수 있습니다.
- 이제 HPR(High Performance Routing)에는 다음과 같은 추가 구성 옵션이 포함되었습니다.
 - 가상 및 원격 시스템에서 응답 시간 변화를 처리하는 경우, HPR 흐름 제어를 개선하는 ARB 진행형 모드 조작(ARB-P라고도 함)
 - 네트워크 트래픽에서 일시적인 지연으로 인해 불필요한 경로 전환이 발생하지 않도록 도와주는 구성 가능한 경로 전환 지연 타이머
- APPC 인터페이스에는 다른 Verb가 여전히 미해결인 경우에도 대화를 할당해제할 수 있는 새 Verb CANCEL_CONVERSATION이 있습니다. 이 Verb는 CPI-C Cancel_Conversation(cmcanc) 호출과 유사한 방식으로 작동하며 AIX, Linux 및 Windows에서 사용 가능합니다.
- Windows용 IBM Remote API Client에서 클라이언트를 설치할 때 사용 언어를 지정하지 않아도 지원되는 모든 언어로 유틸리티 프로그램(클라이언트 구성 유틸리티 **sxclconf**, 클라이언트 모니터 **sxclappl** 및 호출 가능한 TP 구성을 위한 명령행 유틸리티 **tpinst32**)을 실행할 수 있습니다.
- Windows용 IBM Remote API Client에서 클라이언트 구성 유틸리티 **sxclconf**를 사용하여 예외 로깅, 감사 로깅 및 추적(모든 API에서 API 추적 및 클라이언트와 서버 사이의 메시지에 대한 클라이언트/서버 추적)을 사용 가능 또는 사용 불가능으로 지정할 수 있습니다.
- Java CPI-C 응용프로그램이 AIX 및 Linux는 물론 Windows용 IBM Remote API Client에서도 지원됩니다.

사라진 기능

이 릴리스에서 사라진 기능은 없습니다.

추가 정보

기타 추가 정보는 SNA 및 워크스테이션과 연관된 추가 정보가 들어 있는 서적뿐 아니라 Linux용 Communications Server 라이브러리에 있는 다른 서적을 참조하십시오.

제 1 장 SNA 용어 및 개념

이 장에서는 Linux용 Communications Server를 이해하고 사용하는 데 중요한 시스템 네트워크 아키텍처(SNA)의 용어 및 개념을 정의합니다. Linux용 Communications Server, 해당 기능 및 설명된 여러 SNA 개념의 구현 방법에 대해서는 *Linux용 IBM Communications Server* 빠른 시작을 참조하십시오. 이미 SNA와 Linux용 Communications Server에 익숙한 경우에는 33 페이지의 제 2 장 『Linux용 Communications Server 관리』부터 시작하면 됩니다.

이 장은 다음 부분들로 구성됩니다.

- 『시스템 네트워크 아키텍처(SNA)』: SNA 정의를 제공합니다.
- 2 페이지의 『기본 SNA 개념』: SNA 네트워크에 적용되는 용어 및 개념을 설명합니다.
- 14 페이지의 『기본 APPN 개념』: 고급 피어간 통신(APPN)을 지원하는 SNA 네트워크에만 적용되는 용어 및 개념을 설명합니다.
- 31 페이지의 『APPN 네트워크에서 부속 영역 네트워크에 액세스』: SNA와 APPN을 결합하는 네트워크에 적용되는 용어 및 개념을 설명합니다.

주: 이 장은 SNA 개념에 대한 완전한 참조서는 아닙니다. SNA에 대한 자세한 정보는 참고 서적에 나열된 SNA 서적에서 볼 수 있습니다.

시스템 네트워크 아키텍처(SNA)

시스템 네트워크 아키텍처(SNA)는 다양한 하드웨어 및 소프트웨어 데이터 통신 제품 사이의 통신을 위한 공통 규약을 지정하는 IBM 데이터 통신 아키텍처입니다. 이 아키텍처는 두 종류의 정의에 해당되는 형식(네트워크 구성요소에 의해 교환되는 메시지 레이아웃 정의)과 프로토콜(메시지에 대한 응답으로 네트워크 구성요소가 취하는 조치 정의)로 구성됩니다.

SNA 네트워크는 서로 링크되고 SNA를 사용하여 통신하는 컴퓨터 콜렉션입니다.

원래 SNA는 호스트 컴퓨터와의 통신이 가능하도록 하기 위해 설계된 것입니다. 각각의 네트워크 또는 서브네트워크는 호스트에 의해 제어되고, 다른 컴퓨터들은 호스트와 직접 통신하지만 서로 통신할 수는 없었습니다. 이와 같이 호스트가 제어하는 기존 스타일의 네트워크를 종종 부속 영역 SNA라고 합니다. 이후에 SNA는 호스트 없이 네트워크에서 컴퓨터 사이의 직접 피어간 통신을 지원하도록 확장되었습니다. 이 새로운 피어 레벨 네트워킹이 APPN입니다.

많은 SNA 네트워크가 부속 영역 및 피어간 네트워킹 둘 다의 요소를 가지고 있습니다. 네트워크는 부속 영역 SNA에서 APPN으로 이주하므로, APPN 가능 호스트는 이전 시스템을 제어하는 역할을 하는 반면 새로운 시스템에 대해 피어 역할도 할 수 있습니다. 마찬가지로, 단일 컴퓨터가 피어 컴퓨터(APPN 네트워크에 있는)와 이전 호스트 둘 다에 액세스할 수 있으며 호스트와의 통신은 호스트에 의해 제어됩니다. 그러나 다른 컴퓨터와의 통신은 피어간 통신으로서 호스트는 관여하지 않습니다.

기본 SNA 개념

SNA는 메인프레임에서 단말기에 이르는 장치에 사용되는 표준, 프로토콜 및 기능을 정의하여 SNA 네트워크에서 서로 통신할 수 있도록 합니다.

SNA 기능은 별도의 계층으로 구성되는 계층 구조로 나뉘지며, 각 계층은 특정 기능 세트를 수행합니다. 이렇게 네트워크 기능을 계층으로 나누면 네트워크 장치가 네트워크에 있는 각 장치에 대한 자세한 정보가 없어도 정보 및 처리 자원을 공유할 수 있습니다. 워크스테이션에 있는 사용자는 네트워크에 있는 물리 장치나 장치 사이의 연결에 대해 전혀 알지 못해도 다른 사용자와 통신할 수 있습니다.

네트워크 유형

SNA는 다음 유형의 네트워크를 지원합니다.

- 부속 영역 네트워크는 부속 영역 노드와 주변 노드로 구성되는 계층 구조 네트워크입니다. 호스트 및 통신 제어기와 같은 부속 영역 노드는 일반적인 네트워크 경로지정을 처리합니다. 단말기와 같은 주변 노드는 일반적인 네트워크 경로지정을 인식하지 않고 네트워크에 접속합니다.
- 피어 네트워크는 모두 일반 네트워크 경로지정에 관여하는 피어 노드로 구성되는 공동 구성 네트워크입니다.
- 혼합 네트워크는 호스트 제어 통신과 피어 통신 둘 다를 지원하는 네트워크입니다.

주: Linux용 Communications Server를 실행하는 Linux 시스템은 부속 영역 네트워크의 주변 노드로, 피어 네트워크의 피어 노드로, 또는 동시에 두 노드 모두로 작동할 수 있습니다.

SNA 노드

SNA 네트워크에서 노드는 Linux 시스템이나, SNA 프로토콜을 구현하고 네트워크에서 다른 노드에 대한 최소 하나의 통신 경로를 가지고 있는 소프트웨어 구성요소와 연관되는 기타 장치입니다. 각 노드는 네트워크 통신 경로에서 해당되는 끝을 관리하고 SNA 프로토콜을 사용하여 각 경로의 다른 끝에 있는 노드와 통신합니다.

부속 영역 네트워크와 피어 네트워크는 노드 사이의 관계를 다르게 정의하므로, 노드 유형마다 다른 용어를 사용합니다(네트워크에서 노드가 수행하는 역할을 설명하기 위해).

부속 영역 네트워크에서의 노드 유형

SNA 부속 영역 네트워크는 다음과 같은 노드 유형을 지원합니다.

- 부속 영역 노드는 접속된 모든 노드에 대한 통신 및 네트워크 자원을 제어합니다. SNA는 해당 기능 및 다른 노드에 대한 제어 정도에 따라 부속 영역 노드를 분류합니다.
 - 유형 5 노드는 네트워크 자원을 제어하고, 트랜잭션 프로그램과 네트워크 운영자를 지원하고, 일반 사용자 서비스를 제공하는 SNA 기능을 제공합니다. 이러한 기능은 종종 호스트 프로세서가 제공하므로, 유형 5 노드를 호스트 노드라고도 합니다. 유형 5 부속 영역 노드가 제어하는 장치 및 자원은 해당 노드의 도메인을 구성합니다.
 - 유형 4 노드는 네트워크 일부에서 데이터 흐름을 제어하고 경로를 지정하는 SNA 기능을 제공합니다. 이러한 기능은 종종 통신 제어가 제공하므로, 유형 4 노드를 통신 제어기 노드라고도 합니다.
- 주변 노드는 부속 영역 네트워크에서 종속 역할을 수행합니다. 예를 들어, 주변 노드는 3270 에뮬레이션이나 종속 LU 6.2 통신을 지원할 수 있습니다. 주변 노드는 분산 프로세서, 클러스터 제어기 또는 워크스테이션과 같은 장치로, 유형 2.0 및 유형 2.1 노드으로도 분류됩니다.
 - 유형 2.0 노드는 항상 유형 4 또는 5 노드가 제어합니다. 이 노드는 유형 4 또는 5 노드의 관여 없이 다른 노드와의 통신을 설정할 수 없습니다. 유형 2.0 노드는 종속 노드라고 합니다.
 - 유형 2.1 노드는 종속 노드로 작동할 수 있지만, 직접 다른 유형 2.1 노드와 통신할 수도 있습니다.

주: Linux용 Communications Server를 실행하는 Linux 컴퓨터는 유형 2.1 또는 유형 2.0 노드로 작동할 수 있습니다.

주변 노드가 접속된 유형 4 또는 5 부속 영역 노드는 경계 노드로 작동합니다. 이 노드는 부속 영역 노드가 사용하는 네트워크 주소와 주변 노드가 사용하는 로컬 주소 사이에서 변환하여 경계 기능을 수행합니다.

간단한 부속 영역 네트워크에는 다음 구성요소가 포함됩니다.

호스트 호스트는 원래 IBM System/370™과 호환 가능한 메인프레임 컴퓨터입니다. 호스트는 일반적으로 유형 5 노드입니다. 하지만 System z의 Linux용 Communications Server는 호스트 컴퓨터에서 유형 2.1 또는 2.0 노드로 실행됩니다.

통신 제어기

통신 제어기는 호스트에 접속된 별도의 프로세서로, 프론트 엔드 프로세서(FEP)라고도 하며 다른 컴퓨터와의 호스트 통신을 관리합니다.

통신 링크

통신 링크는 호스트 사이트를 일반 사용자 사이트와 연결합니다. 사용자는 보통 호스트와는 별도의 사이트에 있으므로, 두 사이트를 통신 링크로 연결해야 합니다.

터미널 제어기

원격에서 통신 링크의 끝은 터미널 제어기(클러스터 제어기라고도 함)입니다. 이 제어기는 링크 사용을 제어하고 데이터를 단말기로 전송합니다. 가장 잘 알려진 IBM 터미널 제어기는 3174 및 3274입니다.

단말기 사용자는 호스트 응용프로그램을 실행하거나 단말기에서 호스트로 작업을 제출합니다. 가장 잘 알려진 IBM 터미널은 3270입니다. 단말기는 터미널 제어기를 통해 연결하거나 직접 통신 제어기에 연결할 수 있습니다.

프린터 IBM 3287과 같은 프린터를 터미널 제어기에 접속할 수도 있습니다. 이 프린터는 호스트에서 출력을 수신합니다.

그림 1에 표시된 것처럼, 부속 영역 네트워크 다이어그램은 트리를 거꾸로 한 것처럼 보입니다.

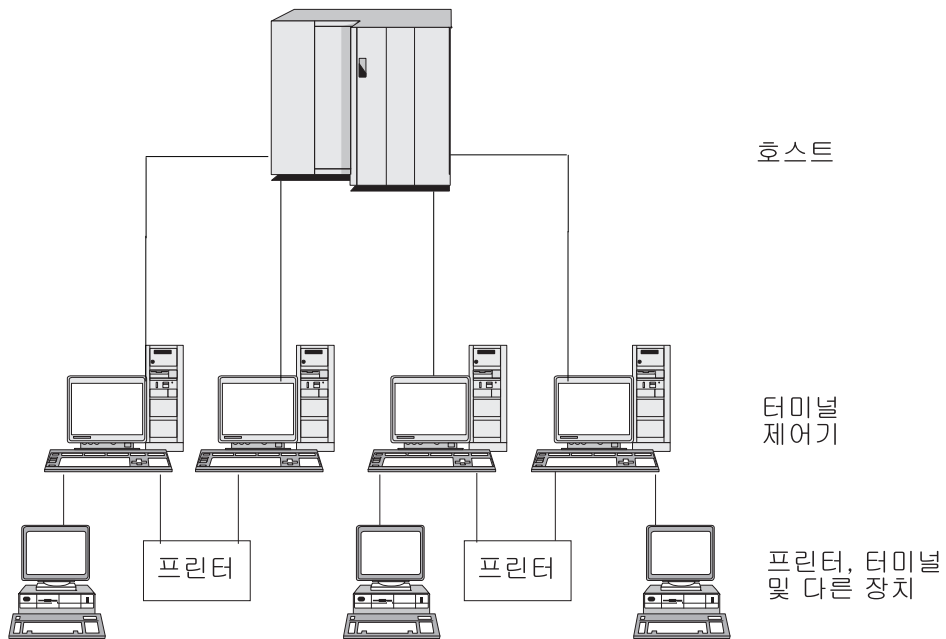


그림 1. SNA 부속 영역 네트워크

트리 루트(다이어그램의 맨 위)는 네트워크를 제어하는 컴퓨터입니다. 브랜치는 호스트에서 네트워크의 다른 컴퓨터로의 통신 링크이고(터미널 제어기), 단말 노드(다이어그램의 맨 아래)는 이러한 컴퓨터에 접속되어 있으면서 사용자가 액세스하는 단말기 또는 프린터입니다.

여기에 설명된 일반적인 부속 영역 SNA 설정으로 사용자는 단일 호스트 시스템 자원을 사용할 수 있습니다. 단말기는 터미널 제어기에 데이터를 입력하고 터미널 제어기로 부터 데이터를 표시하는 기능만 제공합니다. 터미널 제어기가 단말기와 호스트 사이의 SNA 통신을 처리합니다.

터미널 제어기 및 해당 터미널은 Linux용 Communications Server 등의 제품을 사용하여 SNA 노드로 대체될 수 있습니다. 호스트 관점에서, 노드는 터미널 제어기로 표시됩니다. 그러나 사용자에게 여러 호스트 시스템에 액세스할 수 있고 화면 표시를 사용자 조정할 수 있는 것과 같은 추가 기능을 제공합니다. 또한 Linux용 Communications Server는 Linux 컴퓨터에서 실행되어 SNA에 관련되지 않은 다른 타스크에도 사용할 수 있습니다(호스트와의 통신에만 사용되는 터미널 제어기와는 달리).

피어 네트워크에서의 노드 유형

피어 네트워크는 노드를 계층적으로 분류하지 않고 부속 영역 네트워크에서처럼 유지됩니다. 다른 노드와의 교환은 호스트나 다른 중앙 프로세서가 제어하지 않습니다. 대신 모든 노드는 다른 모든 노드와의 통신을 설정할 수 있습니다.

피어 네트워크는 유형 2.1 노드로 구성됩니다. 피어 네트워크의 노드는 다음 역할을 수행할 수 있습니다.

- APPN 네트워크 노드(NN)는 네트워크 자원의 위치를 식별하고, 자원 사이의 세션 전송 경로를 판별하며, 세션을 전송하고 네트워크 노드에 직접 접속된 하위 네트워크 (LEN) 노드와 끝 노드(EN) 역할을 수행합니다. APPN 네트워크 노드의 도메인은 도메인 자체와 네트워크 서비스를 제공하는 끝 노드로 구성됩니다.
- APPN 끝 노드는 끝 노드에 해당 자원을 구성하지 않아도 원격 자원에 액세스할 수 있습니다. 끝 노드는 자체에 있는 인접 노드와 통신할 수 있지만 인접 노드에 액세스하기 위한 네트워크 노드 서버 서비스가 필요합니다. APPN 끝 노드의 도메인은 자체만 포함합니다.
- APPN 브랜치 네트워크 노드를 사용하면 APPN 네트워크가 브랜치로 구분되어 해당 토폴로지를 단순화하고 네트워크 관리 오버헤드를 줄일 수 있습니다. 이 노드는 기본 APPN 네트워크에서 구분된 브랜치에 있는 끝 노드에 네트워크 노드 기능을 제공하지만 기본 네트워크 자체에서는 끝 노드의 역할을 수행합니다. 자세한 정보는 30 페이지의 『Branch Extender』를 참조하십시오.
- 하위 네트워크 노드(LEN 노드)는 APPN 기능을 지원하지 않는 유형 2.1 노드입니다. 이 노드는 APPN 네트워크에 있는 인접 노드와는 통신할 수 있지만 APPN 네트워크에는 관여하지 않습니다. LEN 노드에서는 원격 LU가 있는 가능한 모든 세션을 사전에 정의해야 합니다(모든 원격 LU가 특정 링크를 사용하여 액세스할 수 있는 인접 네트워크 노드에 상주함을 표시하는 단일 디폴트 항목을 통하거나 특수하게 정의함). LEN 노드 도메인은 자체만 포함합니다.

피어 지향 노드 유형에 대한 자세한 정보는 15 페이지의 『APPN 노드 유형』을 참조하십시오.

연결성

통신할 두 노드는 각각 노드 사이의 데이터 흐름을 지원하는 하드웨어 및 소프트웨어 조합을 가지고 있어야 합니다. 하드웨어 구성요소는 각 노드에 있는 어댑터와 두 개의 어댑터를 연결하는 전송 매체로 구성됩니다. 소프트웨어 구성요소는 하드웨어 제어와 하드웨어에서 교환되는 데이터의 제어를 제공합니다.

네트워크에 연결된 각 노드는 특정 인접 노드로의 데이터 흐름을 제어하는 노드의 하드웨어 및 소프트웨어인 하나 이상의 링크 스테이션을 가지고 있습니다. 두 인접 노드 사이의 통신을 설정하려면 링크 스테이션 중 하나가 먼저 노드 사이의 링크를 활성화해야 합니다.

트랜잭션 프로그램

SNA 네트워크에서 정보를 교환하는 프로그램을 트랜잭션 프로그램(TP)이라고 합니다.

다음은 SNA TP를 포함할 수 있는 응용프로그램의 예제입니다.

- 애플리케이션 프로그램
- 파일 전송
- 데이터베이스 트랜잭션 프로세싱
- 네트워크 관리
- 중앙 데이터 서비스

TP는 다른 노드에 있는 상대방 LU와의 세션을 설정 및 유지보수하는 논리 장치(LU)를 통해 네트워크에 액세스합니다. 논리 장치에 대한 자세한 정보는 7 페이지의 『논리 장치』를 참조하십시오.

주: Linux용 Communications Server에는 지원되는 대부분의 API에 대한 샘플 TP가 포함됩니다. 샘플 TP에 대한 자세한 정보는 API용 프로그래머 안내서를 참조하십시오. SNA TP를 다른 제품의 일부로 구매하거나 사용자 고유의 TP를 작성할 수 있습니다(『응용프로그램 프로그래밍 인터페이스(API)』 참조).

응용프로그램 프로그래밍 인터페이스(API)

SNA TP는 응용프로그램 프로그래밍 인터페이스(API)를 사용하여 작성됩니다. API는 데이터 교환 및 제어 기능 수행을 위한 서브루틴과 같이 SNA TP가 SNA 기능에 액세스할 수 있도록 하는 특정 서브루틴을 제공합니다. 이 서브루틴을 사용하여 SNA TP는 원격 노드의 다른 SNA TP와 통신할 수 있습니다.

Linux용 Communications Server에는 다음과 같은 API가 포함됩니다(모든 플랫폼에서).

- APPC—LU 유형 6.2만
- CPI-C(통신용 공통 프로그래밍 인터페이스)—LU 유형 6.2만
- CSV(공통 서비스 Verb) API
- LUA API

또한 Linux용 Communications Server는 다음과 같은 관리 프로그래밍 인터페이스를 포함합니다.

- MS(관리 서비스) API (AIX 또는 Linux 시스템 전용)
- NOF(노드 연산자 기능) API

네트워크 액세스 가능 장치

TP 및 SNA 네트워크 사이의 통신은 다른 네트워크 자원이 (고유 로컬 주소를 통해) 액세스할 수 있는 고유 네트워크 자원인 네트워크 액세스 가능 장치 또는 NAU(이전의 『네트워크 주소 지정 가능 장치』)를 통해 발생합니다.

SNA는 다음 유형의 NAU를 제공합니다.

- 물리 장치(PU)(『물리 장치(PU)』 참조)
- 논리 장치(『논리 장치』 참조)
- 제어점(CP)(9 페이지의 『제어점(CP)』 참조)

주: TP는 구성요소가 아닌 네트워크 사용자로 간주되므로 NAU로 분류되지 않습니다.

물리 장치(PU)

각 SNA 노드에는 물리 장치(PU)가 있습니다. PU는 자원(예: 링크 자원)을 관리하고 호스트와의 통신을 지원합니다.

주: 유형 2.1 노드(APPN 노드가 될 수 있음)에서는 제어점(CP)이 PU 서비스와 기타 서비스를 제공합니다(9 페이지의 『제어점(CP)』 참조). 두 개의 유형 2.1 노드(예: Linux용 Communications Server 노드)는 통신 설정을 위한 호스트 서비스가 없어도 직접 통신할 수 있습니다.

논리 장치

각 SNA 노드에는 하나 이상의 논리 장치(LU)가 있습니다. LU는 TP 및 일반 사용자가 사용하는 기능 세트를 제공하여 네트워크에 액세스할 수 있도록 합니다. LU는 로컬 TP 및 장치와 직접 통신합니다.

SNA는 각각 응용프로그램의 특정 클래스에 맞게 최적화된 몇 개의 LU 유형을 정의합니다. 유형이 다른 LU는 서로 통신할 수 없지만, 유형이 같은 LU는 다른 종류의 시스템에 상주하고 있어도 통신할 수 있습니다.

예를 들어, 두 TP가 동일한 LU 유형을 사용하면 Linux 시스템에서 실행하는 TP는 다른 Linux 시스템의 TP와 쉽게 통신할 수 있는 것처럼 AS/400 컴퓨터의 TP와 통신할 수 있습니다.

Linux용 Communications Server는 다음과 같은 LU 유형을 지원합니다.

LU 6.2(APPC, 5250, APPC Application Suite 및 CPI-C용)

LU 6.2는 분산 데이터 처리 환경에서 프로그램간 통신을 지원합니다. LU 6.2 데이터 스트림은 구조화 필드 데이터 스트림인 SNA 일반 데이터 스트림(GDS) 이거나 사용자 정의 데이터 스트림입니다. LU 6.2는 두 개의 유형 5 노드, 유형 5 노드와 유형 2.0 또는 2.1 노드, 또는 두 개의 유형 2.1 노드 사이의 통신에 사용됩니다. (유형 2.1 노드는 APPN 노드 역할을 수행할 수 있습니다.) 이 LU 유형은 다른 LU 유형보다 더 많은 기능과 더 큰 융통성을 제공합니다. 기존 하드웨어 또는 소프트웨어의 제한이 없을 경우 LU 6.2는 새 응용프로그램을 개발할 때 논리적인 선택이 됩니다.

주: LU 6.2만 독립 LU 기능을 제공할 수 있습니다.

LU 3(3270 인쇄용)

LU 3는 SNA 3270 데이터 스트림을 사용하는 프린터와 응용프로그램을 지원합니다.

예를 들어, LU 3는 고객 정보 제어 시스템(CICS)에서 실행 중이고 데이터를 IBM 3174 설정 제어기에 접속된 IBM 3262 프린터로 송신하는 응용프로그램을 지원할 수 있습니다.

LU 2(3270 표시장치용)

LU 2는 SNA 3270 데이터 스트림을 사용하여 대화식 환경에서 통신하는 응용프로그램 및 표시장치 워크스테이션을 지원합니다. 유형 2 LU는 또한 파일 전송을 위해 SNA 3270 데이터 스트림도 사용합니다.

예를 들어, LU 2 프로토콜은 워크스테이션 IBM 3270 제품군 터미널 기능을 수행할 수 있도록 하는 3270 에뮬레이션 프로그램을 지원할 수 있습니다. 또한 LU 2는 보통 3270 표시장치에 출력을 제공하는 호스트 응용프로그램과 통신하기 위해 다른 프로그램에서 사용됩니다. 이러한 TP를 사용하면 워크스테이션이 호스트와의 통합 처리 양식을 획득할 수 있습니다.

LU 1(SCS 인쇄 및 RJE용)

LU 1은 응용프로그램 및 대화식, 일괄처리 데이터 전송 또는 분산 데이터 처리 환경에서 통신하는 단일 또는 다중 장치 데이터 처리 워크스테이션을 지원합니다. LU 유형 1에 사용되는 데이터 스트림은 SNA 문자열 또는 문서 내용 아키텍처(DCA)를 준수합니다.

예를 들어, LU 유형 1은 정보 관리 시스템/가상 기억영역(IMS/VS)에서 실행 중이고 IBM 8100 정보 시스템과 통신하는 응용프로그램을 지원할 수 있습니다. 이로서 운영자는 응용프로그램이 유지보수하는 데이터베이스를 정정할 수 있습니다.

LU 1을 사용하는 응용프로그램은 종종 원격 작업 입력(RJE) 응용프로그램으로 설명됩니다.

LU 0(LUA-용)

LU 0(이전 LU 정의)은 기본 프로그램간 통신을 지원합니다. IMS/VS(정보 관리 시스템/가상 기억영역)와 같은 특정 호스트 데이터베이스 시스템 및 IBM 4680 저장 시스템 운영 체제와 같은 소매상 및 금융 업계용 일부 POS(Point of Sale)는 LU 0을 사용합니다. 이들 제품의 최근 릴리스는 새 응용프로그램에 자주 사용되는 프로토콜인 LU 6.2 통신도 지원합니다.

주: SNA 논리 장치에서 사용하는 데이터 스트림에 대한 정보는 시스템 네트워크 아키텍처 기술 참조서를 참조하십시오.

제어점(CP)

제어점(CP)은 해당 도메인 내에서 네트워크 자원을 관리하여 자원 활성화, 비활성화 및 상태 모니터링을 제어하는 NAU입니다. CP는 링크와 같은 물리 자원과 네트워크 주소와 같은 논리 정보 둘 다를 관리합니다.

SNA는 다음 유형의 네트워크 제어점을 정의합니다.

시스템 서비스 제어점

유형 5 노드에서는 CP를 시스템 서비스 제어점(SSCP)이라고 합니다. SSCP는 부속 영역 네트워크에서 네트워크 자원을 관리 및 제어합니다. 예를 들어, SSCP는 네트워크 자원 디렉토리를 사용하여 해당 제어 하에 있는 특정 LU를 찾고 해당 도메인에서 두 LU 사이의 통신을 설정할 수 있습니다. SSCP는 또한 다른 SSCP와 공동으로 서로 다른 부속 영역 도메인에 있는 LU 사이의 연결성을 설정할 수 있습니다.

SSCP는 또한 호스트 시스템에서 네트워크의 자원을 검사하고 제어할 수 있는 네트워크 운영자와의 인터페이스를 제공합니다.

물리 장치 제어점

부속 영역 네트워크에서 유형 4 노드와 유형 2.0 노드의 경우, 제어점을 물리 장치 제어점(PUCP)이라고 합니다.

제어점(CP)

유형 2.1 노드에서 제어점은 로컬 링크 스테이션 활성화, 로컬 운영자와의 상호작용 및 로컬 자원 관리와 같은 PU 및 LU 기능 둘 다를 제공합니다. 이 제어점은 또한 상대방 LU 위치 및 로컬 LU의 전송 경로 선택과 같은 네트워크 서비스도 제공합니다.

부속 영역 네트워크에서 Linux용 Communications Server 노드의 CP는 유형 2.0 PU로 작동합니다. 호스트의 SSCP와는 통신하지만 부속 영역 네트워크의 다른 CP와는 통신하지 않습니다.

APPN 네트워크에 참여할 때, CP는 인접 노드의 CP와 네트워크 제어 정보를 교환합니다. CP는 또한 유형 6.2의 독립 LU로 작동할 수도 있습니다. CP는 로컬 노드에서 TP의 디폴트 LU로 작동합니다. APPN 제어점(CP)에 대한 자세한 정보는 18 페이지의 『APPN 제어점(CP)』을 참조하십시오.

세션

NAU는 세션이라고 하는 일시적인 논리 통신 채널을 거쳐 다른 노드의 NAU와 통신합니다. 두 개의 TP가 통신할 수 있으려면, 먼저 해당 LU가 세션을 작성해야 합니다. 로컬 노드에서 세션을 관리하는 LU를 로컬 LU라고 하고, 원격 노드에서 세션을 관리하는 LU를 상대방 LU라고 합니다.

세션 유형

Linux용 Communications Server는 기본적으로 다음 유형의 세션과 관련됩니다.

LU-LU 세션

두 개의 TP가 통신하도록 하려면 TP를 지원하는 LU가 먼저 LU-LU 세션을 작성해야 합니다. 일반적으로, 세션은 하나의 SNA 노드에 있는 TP가 다른 노드의 TP와 통신하려고 하는데 두 노드에 있는 LU 사이의 기존 세션을 사용할 수 없을 경우에 작성됩니다.

SSCP-LU 세션

종속 LU(11 페이지의 『독립 및 종속 LU』 참조)가 부속 영역 네트워크에서 LU와의 세션을 가지려면 먼저 유형 5 노드에서 SSCP와의 활성화된 SSCP-LU 세션을 가지고 있어야 합니다. SSCP-LU 세션이 활성화되면 종속 LU는 LU-LU 세션을 요청할 수 있습니다.

SSCP-PU 세션

SSCP-LU 세션을 작성하려면 먼저 LU를 제어하는 PU가 유형 5 노드에서 SSCP와의 활성화된 SSCP-PU 세션을 가지고 있어야 합니다. SSCP-PU 세션은 PU와 SSCP 사이에 제어 데이터 및 네트워크 관리 데이터를 전달하는 데 사용됩니다.

CP-CP 세션

APPN 네트워크에서는 인접 노드가 CP-CP 세션을 작성합니다. 이 세션은 APPN 네트워크에서 자원을 검색하고 토폴로지 정보를 유지보수하는 데 사용됩니다(18 페이지의 『APPN 제어점(CP)』 참조).

세션의 논리 장치 속성

논리 장치는 LU-LU 세션 중 상호작용하는 방법을 판별하는 속성을 가지고 있습니다. 이 속성은 SNA 아키텍처에 의해 판별됩니다. LU는 1차 또는 2차, 그리고 종속 또는 독립 LU가 될 수 있습니다.

1차 및 2차 LU: 세션을 작성하려면 하나의 LU가 다른 LU로 BIND 요청을 송신하여 세션 활성화를 요청합니다.

- 1차 LU는 주어진 LU-LU 세션에 대한 BIND 요청을 송신하는 LU입니다.
- 2차 LU는 BIND 요청을 수신하는 LU입니다.

피어 네트워크는 고정된 노드 계층을 사용하지 않으므로 사전에 판별되는 1차 또는 2차 LU를 가지고 있지 않습니다.

주: 피어 네트워크에서는 복수 세션에 참여하는 독립 LU(『복수 및 병렬 세션』 참조)가 한 세션에서는 1차 LU로, 다른 세션에서는 2차 LU로 작동할 수 있습니다.

독립 및 종속 LU: 유형 0, 1, 2, 3 LU는 모두 종속 LU입니다. 유형 6.2 LU는 독립 또는 종속 LU로 구성될 수 있습니다.

- 종속 LU(SSCP 종속 LU라고도 함)에는 다른 LU와의 세션을 작성하기 위한 SSCP 서비스가 필요합니다. 종속 LU-LU 세션을 작성하려면 먼저 SSCP-LU 세션이 작성되어 있어야 합니다.

종속 LU는 SNA 호스트에서 LU가 있는 세션에만 있을 수 있습니다. 이러한 제한으로 인해 종속 LU는 보통 부속 영역 네트워크(호스트 중재 네트워크라고도 함)를 사용합니다. 그러나 종속 LU 리퀘스터(DLUR) 기능은 종속 LU의 세션 트래픽이 APPN 네트워크로 전달될 수 있도록 합니다. DLUR에 대한 자세한 정보는 31 페이지의 『APPN 네트워크에서 부속 영역 네트워크에 액세스』를 참조하십시오.

주변 노드의 종속 LU는 항상 2차 LU입니다.

- 독립 LU는 SNA 호스트의 지원이 없어도 다른 독립 LU와의 세션을 작성할 수 있습니다. LU 6.2는 독립 LU가 될 수 있는 유일한 LU 유형입니다.

독립 LU는 세션을 작성할 때 1차 또는 2차 LU로 작동할 수 있습니다.

복수 및 병렬 세션

독립 LU는 동시에 여러 개의 원격 LU가 있는 세션(복수 세션)에 관여할 수 있습니다.

독립 LU는 또한 병렬 세션이나, 동일한 원격 LU가 있는 복수의 동시 세션에도 관여할 수 있습니다.

종속 LU(종속 LU 6.2 포함)는 복수 세션을 가질 수 없습니다.

그림 2에 복수 및 병렬 세션을 갖는 LU가 있습니다. LUA 및 LUB는 병렬 세션을 갖습니다. LUA는 복수 세션(LUB가 있는 두 개의 세션 및 LUD가 있는 하나의 세션)도 갖습니다. LUD는 LUA 및 LUC가 있는 복수 세션을 갖습니다.

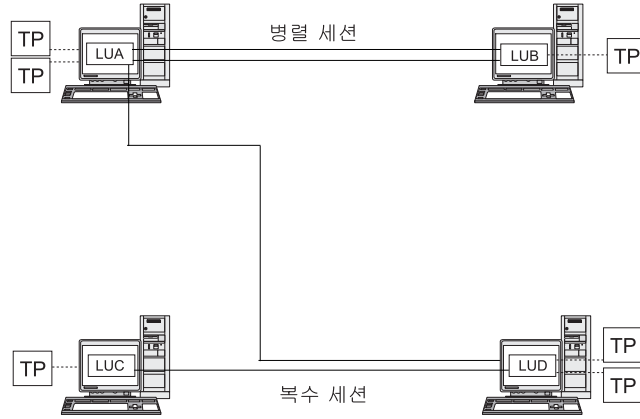


그림 2. 복수 및 병렬 세션

대화

이 절은 LU 6.2에만 적용됩니다.

두 LU 사이에 세션이 작성되면, LU-LU 세션은 트랜잭션을 실행하기 위해 세션을 배타적으로 사용하는 두 TP 사이의 정보 교환을 지원합니다. 이러한 정보 교환을 대화라고 합니다. 한 번에 하나의 대화만 특정 세션을 사용할 수 있지만, 세션을 순차적으로 다시 사용할 수 있습니다. (많은 대화에서 동일 세션을 차례대로 사용할 수 있습니다.)

대화를 시작하기 위해 소스 TP는 원격 TP와의 대화 할당을 요구하는 요청을 해당 LU에 송신합니다. 기동 TP(또는 소스 TP)는 전화 대화에서 호출 당사자처럼 대화를 시작합니다. 기동 가능 TP 또는 목표 TP(원격 TP)는 전화 호출을 수신하는 측과 같이 대화에서 상대방입니다.

13 페이지의 그림 3에 표시된 것처럼, 정보가 TP 및 LU 사이에 교환되어 한 노드가 다른 노드와 통신할 수 있게 됩니다. TP가 직접 통신하는 것처럼 보이지만 각 노드의 LU가 모든 교환에서 중계 역할을 합니다.

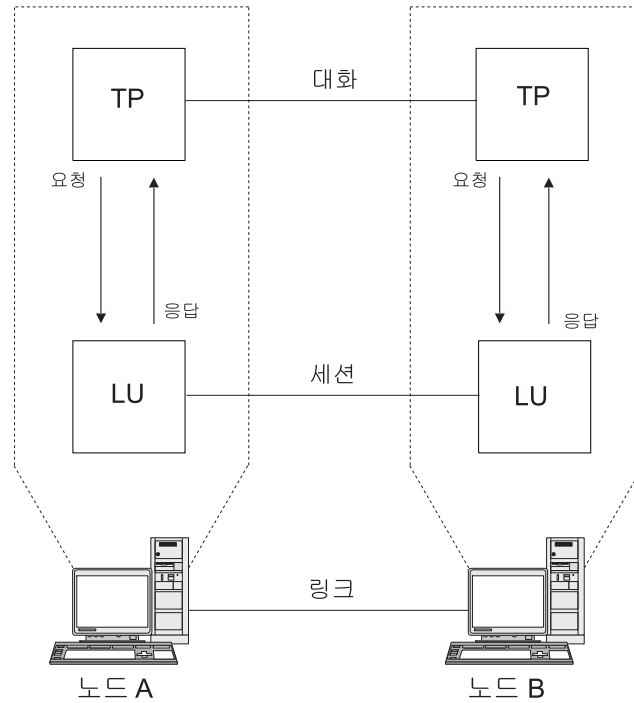


그림 3. 트랜잭션 프로그램 및 논리 장치 사이의 통신

SNA는 두 가지 유형의 대화인 기본 및 직접 대화를 정의합니다. 이 두 유형의 대화는 서로 다른 방법을 사용하여 Linux용 Communications Server와 TP 사이에 처리할 전송 및 수신 데이터 패키지의 길이를 표시합니다.

- 기본 대화에서, 데이터는 SEND 함수에 표시되기 전에 TP에 의해 논리 레코드로 형식화되어야 합니다.

논리 레코드는 2바이트 길이의 필드로 시작하는 2 또는 4바이트 헤더(종종 『LL』로 표시됨)와 최대 32,765바이트의 데이터로 구성됩니다. 논리 레코드는 함께 그룹화하여 블록으로 송신할 수 있습니다. 그러면 여러 논리 레코드를 단일 호출로 SEND 함수로 전송할 수 있습니다.

- 직접 대화에서는 정보가 하나의 형식화되지 않은 데이터 블록에 대한 포인터로 SEND 함수에 전달됩니다. 블록 길이는 다른 매개변수로 전달됩니다. 블록은 여러 개의 논리 레코드로 수신할 수 없으므로 수신 TP는 요구하는 모든 레코드 레벨 형식을 수행해야 합니다.

모드

각 LU-LU 세션에는 세션 특성 세트를 정의하는 연관 모드가 있습니다. 세션 특성에는 페이징 매개변수, 세션 한계(예: 두 LU 사이의 최대 세션 수), 메시지 크기 및 경로 지정 매개변수가 포함됩니다.

각 모드는 고유한 모드 이름으로 식별됩니다. 모드 이름은 해당 모드를 사용하는 모든 SNA 노드에서 같아야 합니다.

전송 경로 선택

LU-LU 세션을 작성하려면 두 LU가 상주하는 노드 사이의 전송 경로를 계산해야 합니다. 전송 경로는 두 노드 사이의 경로를 나타내는 일련의 순서화된 링크 및 노드입니다.

SNA 네트워크는 다음과 같은 전송 경로 선택 방법을 지원합니다.

- 부속 영역 네트워크의 경우 부속 영역 노드 사이의 모든 전송 경로를 사전에 정의해야 합니다.
- APPN을 지원하지 않는 피어 네트워크의 경우, 유형 2.1 노드는 인접 노드와의 세션만 지원할 수 있으며 해당 세션은 중계 노드를 통해 전송할 수 없습니다.
- APPN 네트워크의 경우에는, SNA가 세션 시작 시 세션에서 사용하는 모드에 지정된 서비스 클래스(COS)를 사용하여 동적으로 전송 경로를 계산할 수 있습니다(『서비스 클래스(COS)』 참조).

APPN의 고성능 경로지정(HPR) 기능은 다음과 같은 기능을 제공합니다.

- 고속 전송 프로토콜(RTP)은 세션 전송 경로에서 중계 노드를 통해 네트워크 계층 패킷을 전송하는 데 필요한 주기 및 기억영역 요구사항을 최소화합니다.
- 자동 네트워크 경로지정(ANR)을 사용하면 원래 계산된 전송 경로 일부에 장애가 발생할 경우 APPN 네트워크가 자동으로 세션의 전송 경로를 다시 지정할 수 있습니다.

서비스 클래스(COS)

서비스 클래스(COS)는 로컬 노드가 특정 세션을 작성하기 위해 사용할 수 있는 전송 네트워크(데이터 링크 제어 및 경로 제어) 특성(예: 전송 경로 보안, 전송 우선순위 및 대역폭) 정의입니다. COS 정의는 승인 가능한 보안 레벨, 바이트당 비용, 연결 시간당 비용, 전파 지연 및 유효 용량과 같은 요소에 상대값을 지정합니다.

부속 영역 네트워크에서 COS는 호스트 시스템에 정의된 세션 관련 모드로부터 파생됩니다.

APPN 네트워크 노드는 COS를 사용하여 독립 LU 사이의 세션 전송 경로를 계산합니다. APPN 네트워크에서의 세션 경로지정에 대한 자세한 정보는 23 페이지의 『세션 경로지정』을 참조하십시오.

기본 APPN 개념

고급 피어간 통신(APPN)은 분산 네트워크 제어를 지원하는 네트워크 구조입니다. APPN은 네트워크를 쉽게 구성 및 사용할 수 있게 하고 중앙화된 네트워크 관리를 제공하며 융통성 있는 연결성을 지원합니다.

APPN 네트워크는 유형 2.1 노드로 구성됩니다. 네트워크의 각 노드는 APPN 네트워크에서 최소 하나의 다른 노드에 링크로 연결됩니다. CP-CP 세션은 인접 노드(세 번째 모드를 통하지 않고 직접 링크를 작성할 수 있는 동일 네트워크의 노드들)와의 각 링크를 통해 작성됩니다. APPN 네트워크의 모든 노드는 공통 네트워크 이름을 공유합니다.

APPN 노드는 Application System/400®(AS/400®), CS Windows를 실행 중인 PC, 가상 터미널 액세스 방법(VTAM®)을 사용하는 시스템, Linux용 Communications Server를 실행 중인 Linux 서버와 같은 다양한 규모의 프로세서를 포함할 수 있습니다.

APPN은 다음과 같은 기능을 제공합니다.

- APPN 네트워크 노드 및 끝 노드와 APPN이 아닌 피어 노드에 대한 지원(『APPN 노드 유형』 참조)
- APPN 제어점(CP) 기능(18 페이지의 『APPN 제어점(CP)』 참조)
- 특정 논리 장치 찾기를 지원하기 위한 디렉토리 서비스(19 페이지의 『자원 찾기』 참조)
- 중계 세션 경로지정(ISR), 자동 네트워크 경로지정(ANR) 또는 연결 네트워크(CN)를 사용한 세션 작성을 지원하기 위한 토폴로지 및 경로지정 서비스(23 페이지의 『세션 경로지정』 및 28 페이지의 『APPN 연결 네트워크』 참조)

주: APPN 노드는 부속 영역 네트워크에도 연결되어, 피어 네트워크의 APPN 노드와 부속 영역 네트워크의 주변 노드 역할을 모두 수행할 수 있습니다.

APPN 노드 유형

다음 유형의 노드가 APPN 네트워크의 일부가 될 수 있습니다.

- 네트워크 노드(16 페이지의 『APPN 네트워크 노드』 참조)
- 끝 노드(17 페이지의 『APPN 끝 노드』 참조)

또한 하위 네트워크(LCN) 노드는 APPN 네트워크에 연결될 수 있지만 APPN 기능은 사용하지 않습니다(17 페이지의 『LCN 노드』 참조).

16 페이지의 그림 4에 샘플 APPN 네트워크가 있습니다.

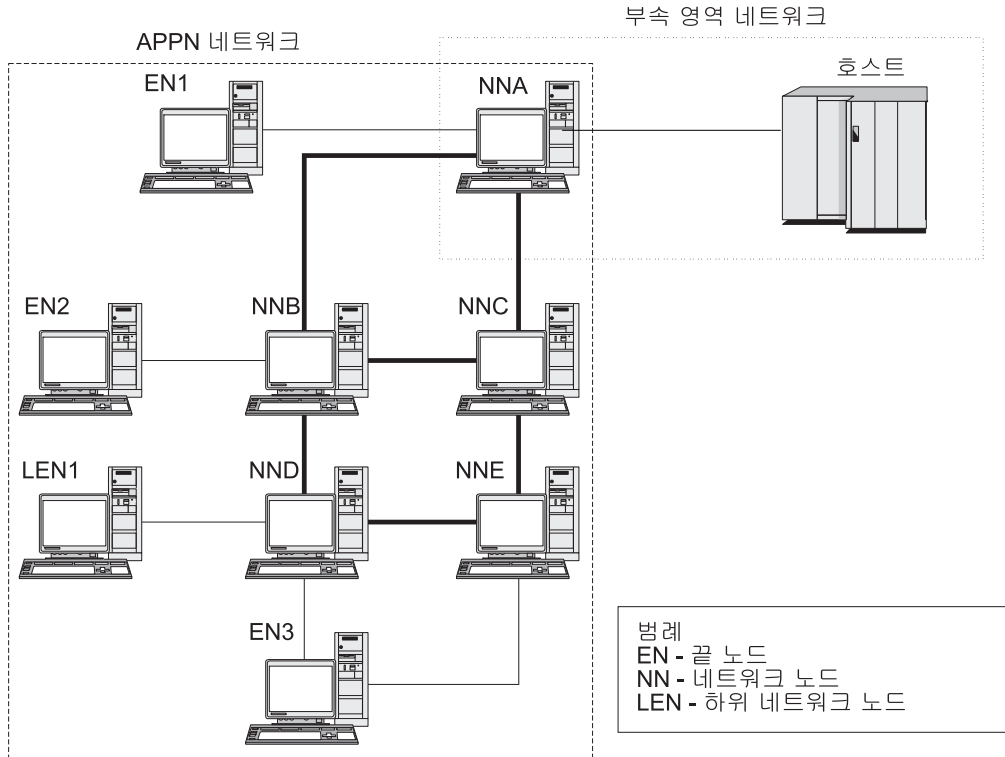


그림 4. 샘플 APPN 네트워크의 일부

이 예제는 다섯 개의 네트워크 노드(NN), 세 개의 끝 노드(EN), 하나의 LEN 노드를 포함하는 APPN 네트워크를 보여줍니다. 네트워크 노드는 APPN 네트워크의 백본(backbone)을 형성합니다. 각각의 끝 노드는 네트워크 노드를 통해 네트워크에 액세스합니다. 노드의 LU 6.2 TP는 네트워크에 있는 다른 LU 6.2 TP와 통신할 수 있습니다.

APPN 네트워크 노드(NNA) 중 하나는 부속 영역 네트워크에도 관여하여 통신 제어를 통해 호스트에 연결합니다. 이 노드는 APPN 네트워크의 노드와 통신할 경우에는 APPN 노드로, 부속 영역 네트워크의 노드와 통신할 경우에는 주변 노드로 작동합니다. 이 네트워크 노드를 통해 APPN 네트워크에 있는 다른 노드의 LU 유형 6.2 LU가 호스트에서 LU 유형 6.2 LU와의 LU-LU 세션을 작성할 수 있습니다.

APPN 네트워크 노드

APPN 네트워크 노드는 해당 도메인에 있는 모든 LU에 대해 분산 디렉토리 및 경로 지정 서비스를 제공하는 유형 2.1 노드입니다. 이 LU는 네트워크 노드 자체나, 네트워크 노드가 서비스를 제공하는 APPN 끝 노드 또는 LEN 노드에 있을 수 있습니다. APPN 네트워크 노드는 해당 도메인에 있는 끝 노드 및 LEN 노드의 네트워크 시작점 역할을 하므로, 네트워크 노드를 해당 노드의 네트워크 노드 서버라고도 합니다.

네트워크 노드는 다음과 같은 서비스를 제공합니다.

- 해당 로컬 LU에 대한 LU-LU 세션 서비스

- 해당 도메인에 있는 모든 LU에 대한 디렉토리 검색 및 전송 경로 선택
- 중계 세션 경로지정(26 페이지의 『중계 경로지정』 참조)
- 서비스를 받는 끝 노드와 MS 포컬 포인트 사이의 관리 서비스(MS) 데이터(예: 경고)에 대한 경로지정

APPN 끝 노드

APPN 끝 노드는 APPN 네트워크에서 끝점 역할을 하는 유형 2.1 노드입니다. 이 노드는 로컬 자원에 대한 디렉토리 정보만 유지합니다. APPN 끝 노드는 로컬 LU와 인접 노드에 있는 LU 사이에 세션을 독립적으로 작성할 수 있습니다. 노드의 LU가 끝 노드에 직접 연결되지 않은 세션의 경우, 끝 노드는 CP-CP 세션을 사용하는 네트워크 노드 서버로부터 경로지정 및 디렉토리 정보를 요청합니다.

APPN 끝 노드는 해당되는 네트워크 노드 서버에 로컬 LU를 등록할 수 있습니다. 이는 네트워크 노드가 서비스를 제공하는 접속된 끝 노드의 모든 LU 이름을 네트워크 노드 서버에 있는 네트워크 운영자가 사전 정의하지 않아도 됨을 의미합니다.

APPN 끝 노드는 여러 네트워크 노드에 접속할 수 있지만(16 페이지의 그림 4의 EN3 참조), 한 번에 하나의 네트워크 노드(해당 네트워크 노드 서버)에 대해서만 CP-CP 세션을 활성화할 수 있습니다. 다른 네트워크 노드는 기본 네트워크 노드 서버가 사용 불가능하게 될 경우 네트워크 노드 서버를 대체하거나 끝 노드의 중계 경로지정을 제공하기 위해서만 사용할 수 있습니다.

APPN 끝 노드는 다른 APPN 끝 노드나 LEN 노드에 직접 링크될 수도 있지만 두 개의 끝 노드 사이에 CP-CP 세션이 작성되지 않습니다.

LEN 노드

하위 네트워크 노드(LEN 노드)는 독립 LU 6.2 프로토콜을 사용하지만 CP-CP 세션을 지원하지 않는 유형 2.1 노드입니다. 이 노드는 APPN 네트워크 노드나 끝 노드에 연결할 수 있지만 APPN 기능은 지원하지 않습니다.

APPN 네트워크 노드는 접속된 LEN 노드에 대한 경로지정 서비스를 제공하여, LEN 노드와 APPN 네트워크의 모든 노드 사이에 링크 스테이션을 정의하지 않아도 LEN 노드가 APPN 네트워크에 관여할 수 있도록 합니다.

LEN 노드가 세션을 작성할 APPN 네트워크의 LU는 LEN 노드의 네트워크 노드 서버에 상주하는 것처럼 LEN 노드에 정의해야 합니다. LEN 노드는 네트워크 노드 서버에서 LU와의 세션을 작성합니다. 네트워크 노드는 APPN 네트워크를 통해 LU가 실제로 상주하는 네트워크의 노드로 세션을 전송합니다. LEN 노드의 LU는 LEN 노드에 서비스를 제공하는 네트워크 노드에 미리 정의되어 있어야 합니다. LEN 노드의 LU 자원은 (끝 노드의 LU 자원과는 달리) 네트워크 노드 서버에 등록할 수 없습니다.

APPN 끝 노드는 중계 경로지정을 제공할 수 없습니다. LEN 노드의 유일한 링크가 APPN 끝 노드 링크일 경우, LEN 노드는 두 노드 사이의 직접 링크를 통해 끝 노드에 있는 LU와만 통신할 수 있습니다.

APPN 제어점(CP)

APPN 제어점(CP)은 유형 2.1 노드에서 물리 장치(PU) 및 논리 장치 기능을 지원하고 노드 자원을 관리하는 기능 세트입니다. APPN CP는 로컬 노드 기능(예: 어댑터와 링크의 활성화 및 비활성화)을 지시하고, 디렉토리 및 토폴로지 정보를 제공하며, 세션 시작 및 종료 시 LU를 보조합니다.

APPN 네트워크의 인접 노드는 병렬 CP-CP 세션 쌍을 사용하여 네트워크 정보를 교환하고 디렉토리 및 전송 경로 선택 서비스를 제공합니다. 상대방 CP가 상호작용을 시작하고 유지하도록 하려면 지정된 쌍의 두 세션이 모두 활동 중이어야 합니다. 서로 다른 노드 유형은 다음과 같이 이들 세션을 다르게 사용합니다.

- APPN 네트워크 노드와 각각의 인접 네트워크 노드 사이에 두 개의 병렬 CP-CP 세션이 작성됩니다. 이 CP-CP 세션은 디렉토리, 토폴로지 및 관리 서비스 데이터의 교환에 사용됩니다.
- APPN 끝 노드와 끝 노드의 서버로 작동하는 인접 네트워크 노드 사이에 두 개의 병렬 CP-CP 세션이 작성됩니다. 이 CP-CP 세션은 디렉토리, 토폴로지 및 관리 서비스 데이터의 교환에 사용됩니다.
- LEN 노드는 CP-CP 세션을 지원하지 않습니다.

CP-CP 세션에 제공되는 기능은 다음과 같이 연관된 노드 유형에 따라 다릅니다.

- 모든 CP-CP 세션은 디렉토리 검색을 수행합니다.
- 끝 노드와 네트워크 노드 사이의 CP-CP 세션은 다음 기능을 제공합니다.
 - 자원 등록
 - 끝 노드와 포컬 포인트 사이의 관리 서비스 데이터(예: 경고) 경로지정
 - 각각의 끝 노드에서 해당되는 네트워크 노드 서버로의 토폴로지 데이터 경로지정. 이 정보는 네트워크 노드 서버가 네트워크 노드 서버를 통해 흐르지 않는 전송 경로를 계산하기 위해 사용됩니다.
- 인접 네트워크 노드 사이의 CP-CP 세션은 토폴로지 정보를 교환합니다. 이 교환의 결과로, 각 네트워크 노드는 내부 네트워크 토폴로지 데이터베이스를 작성합니다.

노드를 설정할 때 CP 이름을 정의해야 합니다. CP는 또한 사용자 세션을 지원할 수 있는 LU이기도 하고, 사용자가 선택할 경우 사용자 노드에 정의된 유일한 LU가 될 수도 있습니다.

자원 찾기

TP 사이의 통신을 지원하기 위해 Linux용 Communications Server는 먼저 해당 TP를 제어하는 논리 장치 사이에 세션을 작성합니다. APPN을 사용하면 노드의 CP는 노드가 원격 LU에 대한 구성 정보를 가지고 있지 않아도 APPN 네트워크에서 LU를 찾을 수 있습니다. 네트워크에서 동적으로 LU를 찾는 APPN 기능을 디렉토리 서비스라고 합니다. 자원을 찾으면 APPN 네트워크를 통해 세션의 전송 경로가 계산됩니다.

자원 이름

각 노드에는 네트워크 이름과 제어점 이름의 두 부분으로 구성되는 고유 이름이 있습니다. 이들 이름이 전체 CP 이름을 구성합니다. 이 이름은 네트워크에 있는 다른 모든 노드에 대해 각 노드를 식별합니다. 마찬가지로, 각 논리 장치는 네트워크 이름과 LU 이름으로 구성되는 전체 LU 이름에 의해 식별됩니다.

주: 네트워크 이름 지정 규칙에 대한 자세한 정보는 *Linux용 IBM Communications Server* 빠른 시작을 참조하십시오.

디렉토리 서비스

각 APPN 노드는 네트워크 자원의 디렉토리를 유지보수합니다. 디렉토리 서비스는 로컬 디렉토리 데이터베이스를 관리하고 네트워크 노드에서 APPN 네트워크 전체를 검색하여 네트워크 자원을 찾는 노드 CP의 구성요소입니다.

노드가 시작될 때 노드에는 다음 정보가 포함됩니다.

- 노드 유형(APPN 네트워크 노드, APPN 끝 노드 또는 LEN 노드)
- 노드의 네트워크 ID
- 노드의 CP 이름

각 노드 디렉토리는 각 자원의 전체 이름, 유형 및 등록 상태를 포함하여 자원(LU 및 CP) 항목을 유지합니다. 각 로컬 디렉토리에 저장되는 특정 자원은 노드 유형에 따라 다릅니다.

- LEN 노드는 해당되는 고유 LU가 있는 디렉토리를 유지합니다. 또한 가능한 모든 상대방 LU의 디렉토리 항목을 사용하여 구성해야 합니다. LEN 노드가 세션을 작성할 APPN 네트워크의 LU는 LEN 노드의 네트워크 노드 서버에 상주하는 것처럼 LEN 노드에 정의해야 합니다. LEN 노드는 네트워크 노드 서버에서 LU와의 세션을 작성합니다. 네트워크 노드는 APPN 네트워크를 통해 네트워크의 적절한 노드로 세션을 전송합니다.

LEN 노드는 또한 디렉토리 항목에서 와일드카드를 사용하여 특정 링크를 통해 액세스할 수 있는 여러 상대방 LU를 지정할 수 있습니다.

- APPN 끝 노드는 해당되는 고유 LU가 있는 디렉토리를 유지보수합니다. 또한 인접 노드에 있는 상대방 LU의 디렉토리 항목을 저장하도록 구성할 수 있습니다. 이로써 로컬 LU는 APPN 기능을 사용하지 않고 해당 LU와의 피어간 세션을 작성할 수 있습니다.

끝 노드에 대해 자원이 로컬로 정의되어 있지 않거나 현재 끝 노드가 자원에 도달할 수 없는 경우, 끝 노드는 요청을 해당 네트워크 노드 서버에 송신하여 자원에 해당되는 APPN 네트워크를 검색하도록 요청합니다.

- APPN 네트워크 노드는 해당되는 고유 LU와 해당 도메인에 있는 끝 노드 및 LEN 노드가 있는 디렉토리를 유지보수합니다. 끝 노드는 해당되는 네트워크 노드 서버에서 해당 LU를 동적으로 등록할 수 있습니다. (LEN 노드는 네트워크 노드 서버를 사용하여 LU를 등록할 수 없으므로, LEN 노드 LU는 해당 네트워크 노드 서버에 구성해야 합니다.) 네트워크 노드 디렉토리는 또한 네트워크 노드의 도메인에 없지만 이전 검색에서 위치가 판별된 LU의 캐시 항목도 포함할 수 있습니다.

네트워크 노드는 다음 두 가지 방식으로 다른 노드에 디렉토리 서비스를 제공합니다.

- 끝 노드 또는 LEN 노드의 세션 요청에 대한 응답으로 원격 자원을 검색합니다.
- 로컬 디렉토리에서 이름 지정된 자원이 발견될 때 다른 네트워크 노드의 디렉토리 검색 요청에 대해 명확하게 응답합니다.

LEN 노드 디렉토리: 그림 5에 LEN 노드 디렉토리 예제가 있습니다. LEN 노드는 CP-CP 세션을 지원하지 않으므로, 노드 LEN1의 디렉토리는 통신하는 모든 LU를 포함해야 합니다. 노드 LEN1의 디렉토리는 해당되는 네트워크 노드 서버(NNA)를 인접 피어 끝 노드에 없는 LU의 위치로 식별합니다. 노드 LEN1은 노드 NNA를 통해서만 LU에 액세스할 수 있으므로, 네트워크 노드에서 CP를 끝 노드에 있는 LU를 포함한 모든 LU의 『소유 CP』로 정의합니다.

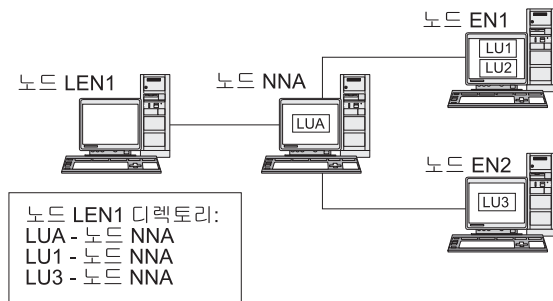


그림 5. LEN 노드 디렉토리

노드에서 직접 접속되어 있지 않은 LU와의 세션을 작성하기 위해 노드 LEN1은 해당되는 네트워크 노드 서버(노드 NNA)에 LU-LU 세션 활성화(BIND) 요청을 송신합니다. 서버는 자동으로 목적지 LU를 찾아서 BIND를 전달합니다.

주: 이 예제에서, 노드 LEN1은 노드 EN1에서 네트워크 노드 서버 NNA를 통해 LU1과의 세션을 작성할 수 있습니다. 그러나 노드 EN1의 LU2는 노드 LEN1의 디렉토리에 정의되어 있지 않으므로 노드 LEN1은 LU2와의 세션을 작성할 수 없습니다.

끝 노드 디렉토리: 끝 노드 디렉토리에 LU가 표시되지 않은 경우, 끝 노드는 원하는 LU를 찾기 위해 LOCATE 검색을 시작합니다. 원격 LU에 대한 검색을 활성화하기 위해 끝 노드는 네트워크 노드 서버의 서비스를 기동합니다. 그림 6에서는 끝 노드 디렉토리의 예제를 보여줍니다.

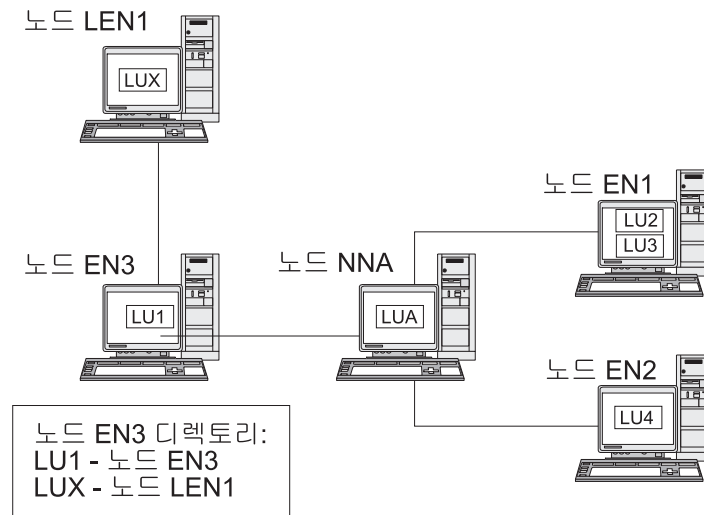


그림 6. 끝 노드 디렉토리

APPN 네트워크에서 가능한 상대방 LU는 끝 노드에 대해 정의하지 않아도 됩니다. 그러나 노드 EN3가 노드 LEN1의 LUX에 대해 세션을 작성하도록 하려면 LEN 노드의 LU가 노드 EN3에서 상대방 LU로 구성되어야 합니다.

네트워크 노드 디렉토리: 네트워크 노드는 서비스를 제공하는 끝 노드에 분산 디렉토리 서비스를 제공합니다.

22 페이지의 그림 7에 네트워크 노드 디렉토리의 예제가 있습니다.

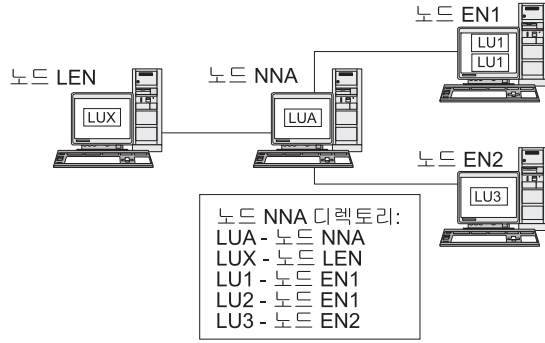


그림 7. 네트워크 노드 디렉토리

네트워크 노드는 다음과 같이 원격 LU를 찾습니다.

1. 네트워크 노드는 LU를 찾기 위한 요청을 수신합니다. 요청은 다음과 같을 수 있습니다.
 - 끝 노드 또는 LEN 노드에 의해 해당되는 네트워크 노드 서버에 송신된 목적지 LU의 이름
 - 끝 노드의 LOCATE 검색 요청에 지정된 LU 이름
 - LEN 노드의 BIND 요청에 지정된 LU 이름
 - 네트워크 노드에서 TP에 의해 지정된 LU 이름

2. 목적지 LU가 네트워크 노드에 없을 경우(그러나 디렉토리에는 표시될 경우) 네트워크 노드는 지시된 검색 요청을 목적지 네트워크 노드 서버에 송신하여 LU 위치를 검증합니다.

LU가 네트워크 노드 디렉토리에 없을 경우, 노드는 브로드캐스트 검색을 모든 인접 네트워크 노드로 송신하여 네트워크 검색을 시작합니다.

3. 각 노드는 다시 브로드캐스트를 전파하고 성공 또는 실패를 표시하는 응답을 리턴합니다.

차후의 필요에 대비하여 네트워크 노드는 성공한 브로드캐스트 검색에서 얻은 정보를 캐시합니다.

APPN 끝 노드는 또한 해당되는 네트워크 노드 서버에서 LOCATE 검색 요청을 수신(및 응답)하여 끝 노드에서 특정 LU를 검색하거나 그 존재를 계속 확인할 수 있습니다.

각 APPN 끝 노드는 네트워크 노드에 등록 메시지를 송신하여 해당되는 네트워크 노드 서버에 LU를 등록합니다. 이러한 방식으로, 네트워크 노드는 해당 도메인의 끝 노드에 대한 현재 디렉토리 정보를 유지합니다. LEN 노드는 해당되는 네트워크 노드 서버에 LU를 등록할 수 없습니다. 따라서 LEN 노드의 모든 LU는 구성을 통해 네트워크 노드 서버에 미리 정의되어 있어야 합니다.

세션 경로지정

APPN은 다음과 같은 동적 전송 경로 선택 절차를 지원합니다.

- 인접 노드가 있는 세션의 경우 직접 세션 경로지정
- 하나 이상의 중계 노드를 경유하는 세션의 경우 다음 중 하나
 - 세션 과정 동안 변경되지 않는 전송 경로를 제공하는 중계 세션 경로지정(ISR)
 - 고속 전송 프로토콜(RTP) 및 자동 네트워크 경로지정(ANR)을 포함하는 고성능 경로지정(HPR). RTP는 세션 전송 경로에서 중계 노드를 통해 네트워크 계층 패킷을 전송하는 데 필요한 주기 및 기억영역 요구사항을 최소화하고, ANR은 전송 경로 장애 또는 과잉 발생 시 세션 트래픽의 경로를 다시 지정할 수 있게 합니다.

동적 전송 경로 선택을 제공하는 APPN 기능을 토폴로지 및 경로지정 서비스(TRS)라고 합니다.

토폴로지 및 경로지정 서비스

각 APPN 노드에는 다른 APPN 노드에 대한 정보와 특정 노드 쌍 사이의 링크 세트인 전송 그룹(TG)에 대한 정보를 저장하는 토폴로지 데이터베이스가 있습니다. 특정 노드의 데이터베이스 내용은 노드 유형에 따라 다릅니다.

- 모든 네트워크 노드는 네트워크 토폴로지 데이터베이스 사본을 공유합니다. 이 공유 데이터베이스에는 다른 모든 네트워크 노드에 대한 정보(예: 네트워크 ID, CP 이름 및 기타 노드 특성)와 각 네트워크 노드 쌍 사이의 전송 그룹에 대한 정보가 있습니다. 이 데이터베이스는 완전한 네트워크 백본(backbone) 토폴로지(네트워크에 있는 노드 쌍 사이의 세션 경로지정에 사용할 수 있는 전송 그룹 및 노드) 보기를 제공합니다.

또한 각 네트워크 노드의 토폴로지 데이터베이스에는 해당 네트워크 노드에서 인접하는 끝 노드 또는 LEN 노드로의 전송 그룹에 대한 로컬 정보가 있습니다.

네트워크 노드는 토폴로지 데이터베이스를 사용하여 해당 도메인에 있는 LU와 원격 LU 사이의 세션 전송 경로를 계산하거나 다른 네트워크에 정보를 제공하여 세션 전송 경로를 계산할 수 있도록 합니다.

- 각각의 끝 노드에는 해당 끝 노드에서 인접 노드로의 전송 그룹에 대한 정보가 있는 로컬 토폴로지 데이터베이스가 있습니다.

끝 노드는 LU를 찾아서 해당 LU로의 세션 전송 경로를 계산하기 위한 요청의 일부로 네트워크 노드 서버에 이 정보를 제공합니다. 네트워크 노드 서버는 끝 노드의 세션 전송 경로를 계산할 때 끝 노드 토폴로지 정보를 사용합니다. 끝 노드는 인접 노드에서 사전 정의된 LU와의 세션을 작성할 때 이 정보를 사용합니다. 끝 노드 토폴로지 데이터베이스는 인접 노드와의 통신만 지원합니다.

주:

1. APPN 네트워크 노드 및 끝 노드는 연결 네트워크와의 링크에 대한 토폴로지 정보도 유지합니다(28 페이지의 『APPN 연결 네트워크』 참조).
2. LEN 노드는 로컬 토폴로지 정보를 유지합니다. 이 노드는 이 정보를 네트워크 노드 서버에 전달하지 않습니다.

그림 8에 표시된 것처럼, 네트워크 토폴로지 정보는 모든 네트워크 노드에서 복제되고 로컬 토폴로지 정보는 네트워크 노드와 끝 노드에서 저장됩니다.

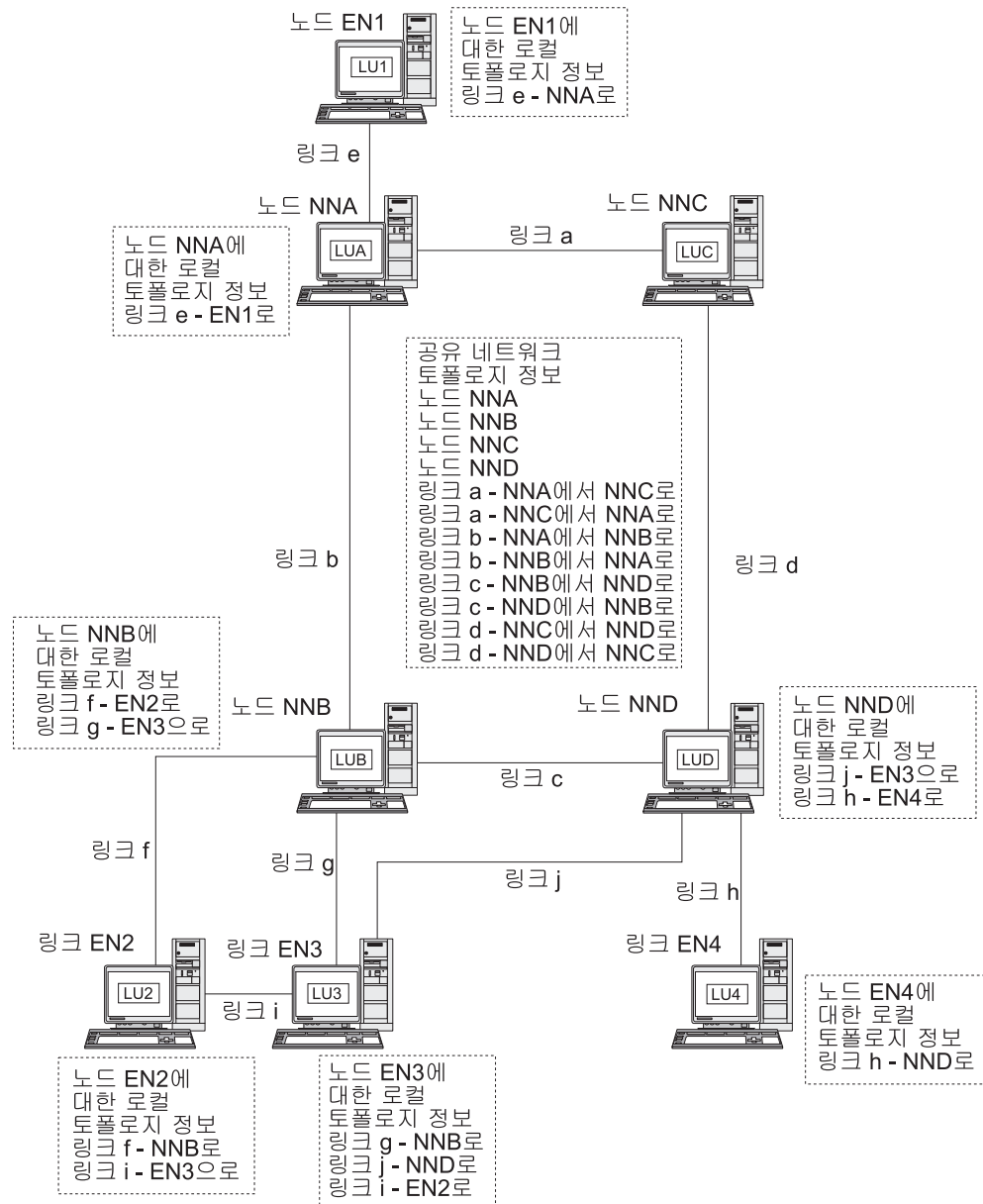


그림 8. 네트워크 노드의 네트워크 토폴로지 데이터베이스

공유 네트워크 토폴로지 데이터베이스는 노드 NNA, NNB, NNC, NND에서 복제됩니다. 또한 이들 노드는 각각 로컬 토폴로지 정보를 포함합니다(끝 노드에 대한 링크가 없어서 로컬 토폴로지 정보를 가지고 있지 않은 노드 NNC를 제외하고). 예를 들어, 노드 NNB에는 노드 EN2에 대한 링크 f와 노드 EN3에 대한 링크 g의 정보는 있지만, 노드 EN2 및 EN3을 연결하는 링크 i의 정보는 없습니다.

끝 노드에는 인접 노드에 대한 링크에 해당되는 정보만 있습니다. 예를 들어, 노드 EN2에는 노드 NNB에 대한 링크 f 및 노드 EN3에 대한 링크 i의 정보가 있습니다.

토폴로지 데이터베이스 갱신(TDU): APPN 네트워크 노드는 자원(예: 노드 또는 두 네트워크 노드 사이의 링크)을 활성화하거나 비활성화할 때, 또는 기존 자원의 특성이 변경될 때 CP-CP 세션을 사용하여 네트워크 토폴로지 정보를 교환합니다. 그러한 변경이 발생할 경우, 네트워크 노드는 노드 ID, 노드 및 링크 특성, 갱신할 자원을 식별하는 갱신 순서 번호, 자원 변경사항 등을 포함하는 토폴로지 데이터베이스 갱신(TDU)을 생성합니다. 각 TDU는 네트워크 토폴로지 데이터베이스가 네트워크 전체에서 최신 상태로 유지되도록 활동 중인 모든 네트워크 노드에 송신됩니다.

APPN 네트워크에서의 전송 경로 선택: APPN 디렉토리 서비스는 특정 세션 상대방을 찾습니다. 네트워크에서 세션 상대방을 찾은 후 토폴로지 및 경로지정 서비스는 최적의 세션 전송 경로를 계산합니다. 각 네트워크 노드는 자체의 고유 LU와 이 LU가 서비스를 제공하는 끝 노드 또는 LEN 노드에 있는 LU에 의해 시작된 세션들에 대한 전송 경로 선택 서비스를 제공합니다. 네트워크 노드는 자체의 고유 로컬 토폴로지 정보 뿐만 아니라 공유 네트워크 토폴로지 데이터베이스의 정보를 사용하여 노드 사이의 전송 경로를 동적으로 계산합니다.

세션 상대방을 찾고 나면, 네트워크 노드는 다음 단계를 수행하여 전송 경로를 선택합니다.

1. 세션 전송 경로의 필수 특성을 확보합니다.

세션을 요청하는 LU가 세션 특성을 식별하는 모드 이름을 지정합니다. 연관된 모드는 세션 트래픽 경로를 지정하기 위해 사용하는 링크의 요구사항을 지정하는 서비스 클래스(COS)를 식별합니다.

2. 가능한 전송 경로에 해당되는 모든 전송 그룹(TG) 및 네트워크 노드를 확보합니다.

- 끝 노드에서 세션 요청이 제공된 경우, 끝 노드는 해당되는 네트워크 노드 서버와 연결 네트워크(존재할 경우)에 대해 가지고 있는 링크에 대한 정보를 제공합니다.
- 세션 상대방이 인접 노드에 없으면, 세션을 요청하는 LU의 네트워크 노드 서버는 네트워크 토폴로지 데이터베이스를 사용하여 세션 상대방에 대한 전송 경로에서 중간 전송 그룹(TG)과 네트워크 노드를 식별합니다.

- 세션 상대방이 끝 노드에 있으면, 끝 노드(또는 해당되는 네트워크 노드 서버)는 네트워크 노드 서버와 해당 끝 노드 사이의 링크(또는 끝 노드와 연결 네트워크 사이의 링크)에 대한 정보를 제공합니다.
3. 세션 전송 경로에 대해 지정된 특성을 갖추고 있지 않은 모든 전송 그룹 및 네트워크 노드를 제외시킵니다.
 4. 세션에 대해 최적의 전송 경로를 계산합니다.

지정된 서비스 클래스(COS)에 따라, 전송 경로 계산 알고리즘은 각 노드 및 논리 링크에 맞는 가중치 값을 계산한 후 전송 경로마다 가중치를 합산합니다. 최적의 경로를 선택하기 위해, 네트워크 노드는 근원지 LU를 포함하는 노드에서 목적지 LU를 포함하는 노드로의 전송 경로 중 현재 가중치가 가장 적은 전송 경로를 계산합니다.

중계 경로지정

중계 경로지정을 사용하면 APPN 네트워크 노드가 다른 노드를 목적지로 하는 데이터를 수신한 후 전송할 수 있습니다. 데이터의 근원 및 목적지는 끝 노드, 다른 네트워크 노드 또는 LEN이 될 수 있습니다.

중계 경로지정은 인접 노드에 없는 LU 사이의 세션을 지원합니다. 세션의 전송 경로가 선택되고 나면, 전송 경로에 있는 APPN 네트워크 노드가 중계 경로지정을 사용하여 세션 데이터를 전송 경로에서 다음 노드로 전달합니다.

토폴로지 데이터베이스에 의해 유지되는 자원 특성에는 과잉 상태가 포함될 수 있습니다. 네트워크 노드의 과잉이 심할 경우, 네트워크 노드는 네트워크에 있는 다른 네트워크 노드에 이 정보를 전달하여 과잉 상태의 네트워크 노드가 새 세션에 대해 계산된 세션 전송 경로에 포함되는 비율이 적도록 할 수 있습니다.

APPN은 다음과 같은 두 가지 유형의 중계 경로지정을 제공합니다.

- 모든 네트워크 노드에서 사용 가능한 중계 세션 경로지정(ISR)에서는 네트워크 노드가 각 중계 세션을 추적합니다. 각 중계 노드는 세션 데이터의 페이징을 조정하여 인접 노드 사이의 데이터 흐름 속도를 제어합니다. 각 중계 노드는 단편화와 단편화된 데이터의 재조합도 수행합니다. ISR에서는 세션 전송 경로가 설정되면 해당 세션의 모든 데이터가 동일한 전송 경로를 사용합니다. 전송 경로 일부에 장애가 발생하면 세션은 종료합니다.
- APPN의 고성능 경로지정(HPR)을 지원하는 네트워크 노드에서 사용 가능한 자동 네트워크 경로지정(ANR)에서는 전송 경로 일부에 장애가 발생할 경우 중계 네트워크 노드가 동적으로 세션 트래픽 경로를 재지정할 수 있습니다. ANR은 중계 세션 페이징이나 단편화 및 재조합을 제공하지 않습니다.

ANR을 사용하면 중계 노드가 일반 APPN ISR에서 가능한 것보다 더 빨리 세션 트래픽 경로를 지정할 수 있습니다. 그러나 ANR을 사용하려면 고속 전송 프로토콜(RTP) 끝점에서 추가 오버헤드가 있어야 합니다. 중계 노드가 적은 전송 경로에서는 ANR 전

송 경로가 끝점에서의 처리 시간으로 인해 ISR 전송 경로보다 실제로 느릴 수 있습니다. 더 많은 중계 노드(홉)를 포함하는 전송 경로의 경우, ANR 전송 경로가 보통 더 빠릅니다. 속도가 빠르지도 느리지도 않게 되는 정확한 위치는 RTP 노드의 효율성에 따라 다릅니다.

직접 연결성

직접 연결성은 APPN 네트워크 노드가 세션 경로를 지정하지 않아도 세션 트래픽이 두 노드 사이에 직접 전송될 수 있도록 합니다. 일반적으로, 직접 연결된 노드 사이의 세션은 네트워크 노드를 통해 데이터를 전송하는 세션보다 더 빠르게 데이터를 교환할 수 있습니다. 공유 액세스 전송 기능(SATF) 노드의 경우(예: 그림 9에 표시된 Enterprise Extender를 사용한 IP 네트워크 또는 LAN 세그먼트의 노드) 네트워크에 있는 모든 노드 쌍 사이에 링크를 정의하여 효율성을 높일 수 있습니다. 그러나 이는 어려운 작업이 될 수 있습니다. 링크 스테이션 수는 $n \times (n-1)$ 이며, 여기서 n 은 네트워크의 노드 수입니다.

그림 9에 Enterprise Extender를 사용한 IP 네트워크 또는 LAN 세그먼트의 APPN 네트워크가 있습니다.

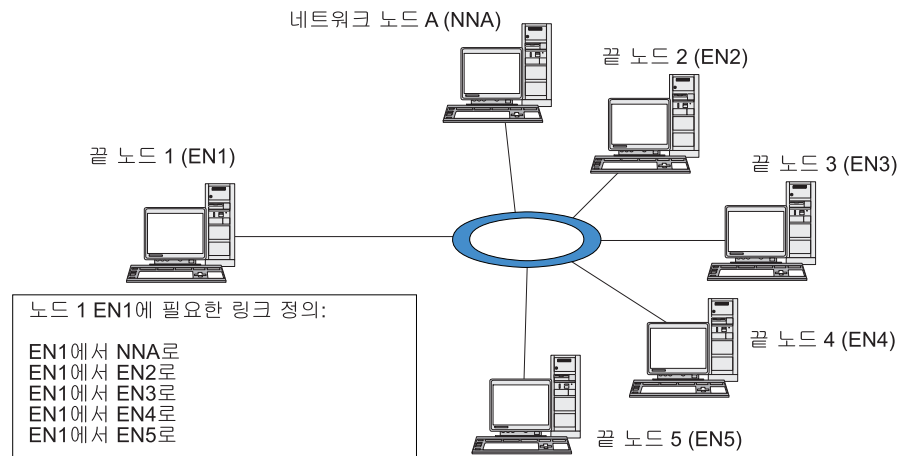


그림 9. 공유 액세스 전송 기능(SATF)을 사용하는 APPN 네트워크

노드 EN1에 네트워크의 각 링크에 대한 링크 정의가 있을 경우, 이 노드는 어떤 노드와도 직접 링크를 설정할 수 있습니다. 노드 EN1과 APPN 네트워크에 있는 다른 모든 노드 사이의 직접 링크를 지원하는 데 필요한 링크 정의는 28 페이지의 그림 10에 있습니다. 다섯 개의 다른 노드를 포함하는 네트워크의 경우 노드 EN1에는 다음 다섯 개의 링크 정의가 필요합니다.

- EN1 대 ENA
- EN1 대 EN2
- EN1 대 EN3
- EN1 대 EN4

- EN1 대 EN5

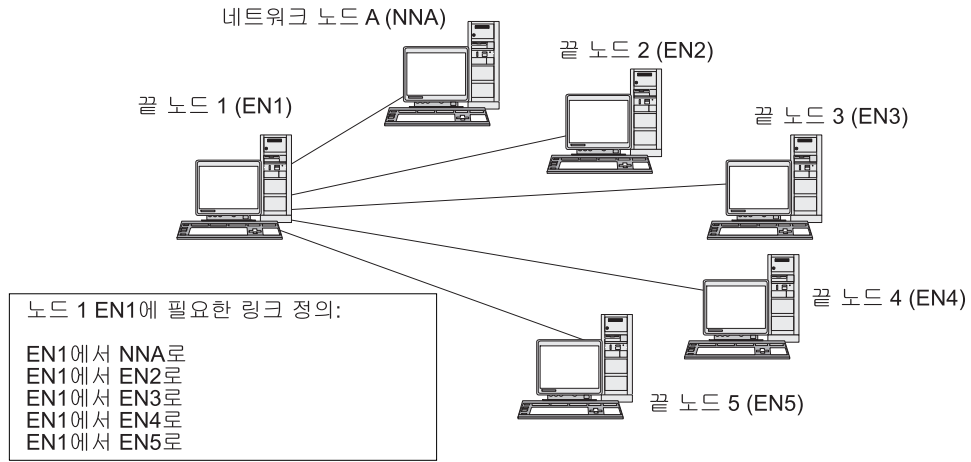


그림 10. APPN 네트워크에서 노드 EN1으로부터 모든 노드로의 직접 링크에 필요한 정의

네트워크의 모든 노드가 다른 모든 노드로의 직접 링크를 지원할 경우, 이 예제에서 6개 노드에 총 30개의 링크 정의가 필요합니다. 일반적으로 링크 정의 수는 $n \times (n-1)$ 으로 계산됩니다. 여기서 n 은 네트워크의 노드 수입니다. 대형 네트워크에서는 링크 정의 수가 급격히 증가하여 통제할 수 없는 상태가 됩니다. 네트워크 노드 사이에 링크 정의 수가 증가하면 네트워크에서 흐르는 TDU 수도 증가하여 네트워크 성능이 저하될 수 있습니다.

APPN 연결 네트워크는 이 문제점에 대한 해결책을 제공합니다.

APPN 연결 네트워크

공유 액세스 전송 기능(SATF)에 접속된 APPN 네트워크의 경우, APPN 연결 네트워크는 네트워크에 있는 노드 사이의 직접 연결성을 지원하기 위해 필요한 링크 정의 수를 크게 감소시킵니다. 연결 네트워크에서 APPN 끝 노드는 모든 노드에 대해 가능한 모든 링크를 구성하는 대신 인접 네트워크 노드 서버에 대한 단일 링크와 연결 네트워크에 대한 링크만 구성하면 됩니다.

연결 네트워크 기능을 사용하려면 APPN 네트워크가 다음 조건을 충족해야 합니다.

- APPN 네트워크의 노드는 토큰링이나 이더넷과 같은 교환식 매체를 사용하여 링크해야 합니다.
- APPN 연결 네트워크의 모든 링크는 같은 매체를 사용해야 합니다.
- 연결 네트워크를 포함하는 APPN 네트워크는 전체적으로 연결해야 합니다. 전체 연결 네트워크에서는 각 노드에 인접 노드와의 CP-CP 세션을 지원하는 최소 하나의 링크가 있습니다.

연결 네트워크에서, SATF는 연결 네트워크에 있는 각 노드에 직접 접속하는 가상 경로 지정 노드(VRN) 역할을 수행합니다. 연결 네트워크의 이름은 VRN의 제어점(CP) 이름 역할을 합니다. VRN은 연결 네트워크에서 두 노드 사이에 세션 데이터의 직접 전송을 지원하지만, 다른 노드와의 CP-CP 세션은 작성하지 않고 TDU를 생성하지도 않습니다. 연결 네트워크의 각 노드는 해당 네트워크 노드 서버와의 링크만 필요로 합니다.

연결 네트워크를 사용할 때 필요한 링크 정의는 그림 11에 있습니다. 가상 노드를 사용하면, 연결 네트워크는 노드 EN1과 APPN 네트워크에 있는 다른 모든 노드 사이의 직접 링크를 지원하지만 두 개의 링크 정의만 있으면 됩니다.

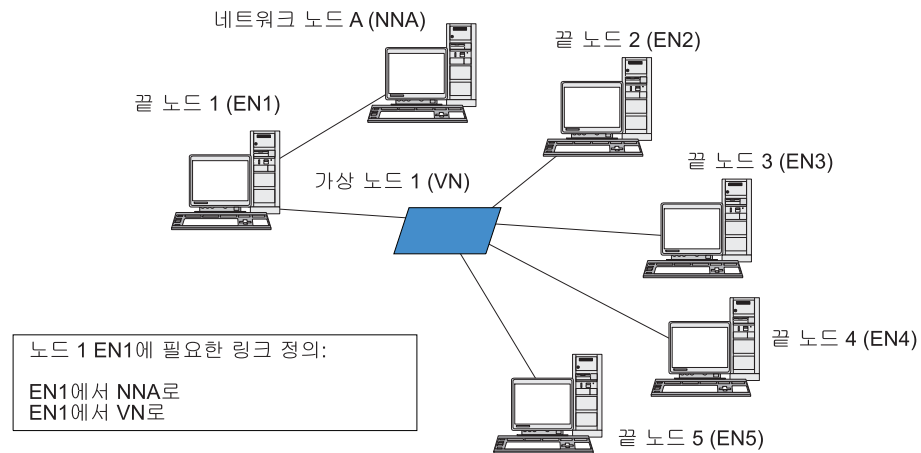


그림 11. 가상 노드를 사용한 직접 링크에 필요한 정의

APPN 네트워크에서 두 개의 끝 노드 사이에 직접 링크를 지원하려면 총 10개의 링크 정의가 필요합니다. (끝 노드마다 두 개의 링크 정의가 필요합니다. 네트워크 노드 서버에 대한 링크 정의 및 가상 노드에 대한 링크 정의입니다.) 연결 네트워크를 사용하지 않는 APPN 네트워크의 직접 연결성 요구사항(28 페이지의 그림 10 참조)에 비교하면 훨씬 적은 링크 정의 수를 가질 수 있습니다(이 예제에서는 30 대신 10개). 대형 네트워크에서는 정의 요구사항의 차이가 상당히 커집니다.

연결 네트워크에 있는 두 노드에서 LU 사이의 세션은 다음과 같이 작성됩니다.

1. 먼저 각각의 끝 노드가 해당 네트워크 노드 서버와의 CP-CP 세션을 작성합니다. (두 개의 끝 노드가 다른 네트워크 노드 서버를 갖는 경우, 해당 네트워크 노드는 CP-CP 세션을 지원하는 링크를 가지고 있어야 합니다.)
2. 끝 노드는 또한 해당되는 VRN 링크 및 로컬 주소 정보를 네트워크 노드 서버에 보고합니다. 로컬 주소 정보는 서비스 액세스점(SAP) 주소와 매체 액세스 제어(MAC) 주소가 될 수 있습니다.

3. 서버는 보통 LU-LU 세션의 최적 전송 경로로 두 개의 끝 노드 사이의 직접 링크를 선택합니다. 서버는 1차 LU가 있는 노드에 상대방 LU가 있는 노드와의 동적 링크를 설정하는 데 필요한 정보를 제공합니다.
4. 그러면 끝 노드는 중계 세션 경로지정(ISR) 없이 LU-LU 세션을 작성할 수 있습니다.

Branch Extender

이전 절에서 설명한 것처럼, APPN 네트워크의 네트워크 노드는 토폴로지 정보(네트워크 내 다른 노드들의 위치 및 이들 노드간의 통신 링크에 관한 정보)를 유지보수하고, 토폴로지가 변경될 때 이 정보를 네트워크에 전달해야 합니다. 네트워크 규모가 커짐에 따라, 저장된 정보량과 토폴로지 관련 네트워크 트래픽이 더욱 커져서 관리하기가 어려워질 수 있습니다.

네트워크를 서브네트워크로 분할하여 이러한 문제를 피할 수 있으며, 따라서 각 노드는 자체 서브네트워크의 노드에 관한 토폴로지 정보만 유지하면 됩니다. 그러나 이렇게 하면 다른 서브네트워크에서 자원을 찾으려고 할 때 네트워크 트래픽을 증가시킵니다.

그림 12에 설명된 APPN의 Branch Extender 기능은 이 문제에 대한 해결책을 제공합니다.

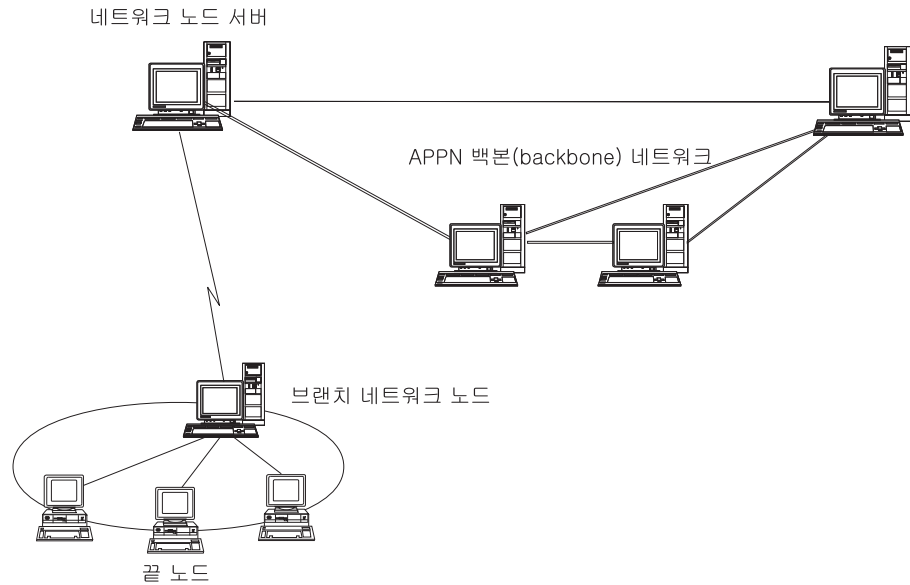


그림 12. Branch Extender

Branch Extender는 그 이름이 나타내듯이 큰 조직의 개별 지점과 같이 별개의 영역으로 나눌 수 있는 네트워크용으로 설계되었습니다. 기본 백본(backbone) APPN 네트워크(예: 조직의 본사에 있는 네트워크)에서 브랜치들을 분리시켜 작동합니다.

각 브랜치에는 브랜치 네트워크 노드(BrNN)라고 하는 새로운 유형의 노드가 있는데, 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 인식하지 못하므로 저장해야 하는 토폴로지 정보의 양이 감소됩니다.
 - BrNN은 끝 노드로 나타나기 때문에 백본(backbone) 네트워크에서 토폴로지 정보를 수신하지 않습니다. (토폴로지 정보는 네트워크 노드 사이에서만 전송됩니다.)
 - BrNN은 브랜치에 있는 자원이 BrNN 자체에 있는 것처럼, 모든 자원을 해당 NNS에 등록합니다. 이는 백본(backbone) 네트워크에 있는 노드가 브랜치에 있는 각 노드를 인식할 필요없이 브랜치에서 자원을 찾을 수 있음을 의미합니다.
- 브랜치 네트워크에 대해 BrNN은 네트워크 노드로 나타나며 브랜치의 끝 노드에 대한 NNS로 작동합니다. 브랜치에 있는 각 노드는 나머지 네트워크가 표준 NNS의 경우와 같은 방식으로 해당 NNS를 통해 연결되는 것으로 인식합니다.

APPN 네트워크에서 부속 영역 네트워크에 액세스

APPN 네트워크는 호스트가 네트워크의 자원을 제어하도록 요구하지 않지만, 호스트는 종종 APPN 네트워크에 관여합니다. APPN은 많은 호스트 플랫폼에서 구현되었으며, 호스트가 APPN 네트워크에서 네트워크 노드로 수행되도록 허용하면서 이전 부속 영역 SNA 기능을 제어하기 위해 SSCP를 제공합니다.

많은 SNA 네트워크에 부속 영역 SNA 및 APPN 요소가 있습니다. 네트워크의 백본(backbone)은 종속 LU와 호스트 기능 사이의 차이를 연결해야 하는 네트워크 노드로부터 빌드됩니다. 이를 위해 두 가지의 추가 서비스가 필요합니다.

- 호스트의 종속 LU 서버(DLUS)는 이전 SSCP 기능과 APPN 네트워크와의 인터페이스에 대한 액세스를 제공합니다.
- 네트워크 노드나 끝 노드의 종속 LU 리퀘스터(DLUR)는 APPN 네트워크를 통해 종속 LU에서 호스트로 세션 트래픽을 전송하는 수단을 제공합니다. 이 기능으로 종속 LU 세션은 APPN이 제공하는 다양한 경로지정 기능을 이용할 수 있습니다.

이러한 DLUR 및 DLUS 조합(일반적으로 간단히 DLUR이라고 함)으로 종속 LU 트래픽은 APPN 백본(backbone)을 거쳐 전송될 수 있습니다. 종속 LU를 사용하는 기존 SNA 응용프로그램은 APPN의 네트워크 관리, 동적 자원 위치 및 전송 경로 선택 기능을 이용하면서 수정없이 유지될 수 있습니다. 이를 통해 DLUR은 부속 영역 SNA에서 APPN으로 유용한 이주 경로를 제공합니다.

APPN 네트워크에서 부속 영역 네트워크에 액세스

종속 LU는 DLUR 기능을 제공하는 노드에 상주하지 않아도 됩니다. DLUR 기능이 네트워크 노드에서 제공될 경우, 종속 LU는 인접 네트워크 노드, 끝 노드 또는 LEN 노드에 있을 수 있습니다. DLUR 기능이 끝 노드에서 제공될 경우, 종속 LU는 끝 노드 자체에 있어야 합니다.

제 2 장 Linux용 Communications Server 관리

Linux용 Communications Server 관리와 제공되는 다른 관리 도구의 개요에 대해서는 『Linux용 Communications Server 관리의 개요』를 참조하십시오.

Linux용 Communications Server 관리의 첫 번째 단계는 노드와 해당 자원을 구성하는 것입니다. 42 페이지의 『Linux용 Communications Server 구성 계획』에 설명된 대로 먼저 구성을 계획하십시오.

Linux용 Communications Server를 구성하려면 먼저 43 페이지의 『로컬 시스템에서 Linux용 Communications Server 사용 가능화 및 사용 불가능화』에 설명된 대로 Linux용 Communications Server 소프트웨어를 사용 가능하도록 해야 합니다.

Linux용 Communications Server가 사용 가능하게 되면 Motif 관리 프로그램을 실행할 수 있습니다(47 페이지의 『Motif 관리 프로그램 사용』 참조). Motif 관리 프로그램은 Linux용 Communications Server를 사용하는 SNA 통신을 지원하는 데 필요한 구성 과정을 안내합니다. Motif 관리 프로그램은 제공해야 할 구성 정보를 최소화하고 여러 통신 유형(예: 3270 또는 APPC 통신)을 지원하기 위해 수행해야 하는 각 단계를 안내하므로 권장되는 관리 도구입니다.

58 페이지의 『명령행 관리 프로그램 사용』에 설명된 대로 명령행 관리 프로그램을 대신 사용할 수도 있습니다.

관리 태스크마다 이 안내서는 Motif 또는 명령행 관리에 사용할 수 있는 정보를 제공합니다. 다른 구성 방법은 34 페이지의 『관리 도구』에 설명되어 있습니다.

Linux용 Communications Server 관리의 개요

사용자는 Linux용 Communications Server 관리자로서 Linux용 Communications Server 소프트웨어를 설치하고 해당 자원을 관리하게 됩니다. Linux용 Communications Server 관리를 시작하기 전에 Linux용 Communications Server 제품의 주요 기능에 대해 알아야 합니다(Linux용 IBM Communications Server 빠른 시작 참조). 이 절에서는 수행해야 하는 관리 태스크와 이 태스크를 수행하기 위해 사용할 수 있는 도구에 대해 설명합니다.

관리 책임

Linux용 Communications Server 시스템을 관리하려면 다음을 수행해야 합니다.

Linux용 Communications Server 관리의 개요

1. 실행할 사용자 프로그램의 요구에 따라 Linux용 Communications Server 시스템 자원을 정의하십시오. Linux용 Communications Server가 통신하는 호스트 또는 피어 컴퓨터의 관리자와 함께 Linux용 Communications Server 구성이 원격 시스템의 구성과 일치하는지 확인하십시오.
2. Linux용 Communications Server 소프트웨어를 초기화하십시오.
3. 선택적으로, 사용자 요구사항이 변경되는 대로 동적으로 구성을 수정하십시오(자원을 추가 또는 제거하거나 정의된 자원을 활성화 및 비활성화하여).
4. 활성 자원 상태를 모니터링하고 진단 정보를 수집하여 발생하는 문제점을 진단하십시오.
5. 선택적으로, 응용프로그램이나 셸 스크립트를 작성하여 표준 관리 조작을 자동화하십시오.

보통 Linux용 Communications Server 시스템이 설치된 사이트에서 시스템 관리자가 이들 작업을 수행합니다. 그러나 Linux용 Communications Server는 NetView 프로그램을 사용하는 운영자가 NetView 콘솔에서 관리 명령을 실행하여 원격으로 2단계와 3단계를 수행할 수 있도록 하는 서비스점 명령 기능(SPCF)도 제공합니다. SPCF에 대한 자세한 정보는 133 페이지의 제 9 장 『NetView에서 Linux용 Communications Server 관리』를 참조하십시오.

관리 도구

Linux용 Communications Server는 시스템 관리를 위해 다양한 도구를 제공합니다. 사용자 요구사항에 따라 모든 도구를 사용하지 않아도 됩니다. 이 절에서는 각 도구가 제공하는 기능을 요약합니다.

주:

1. 이 문서는 이 절에 설명된 도구를 사용하여 수행할 수 있는 Linux용 Communications Server 관리에 대한 일반적인 정보를 제공합니다. 대부분의 목적에 따라 Motif 관리 프로그램을 사용할 것을 권장합니다. 이 프로그램은 노드 구성 및 관리에 대한 적절한 지침을 제공하기 때문입니다.
2. Linux용 Communications Server 관리 도구를 사용할 수 있는 사용자를 제어하는 방법과 그 사용자들이 사용할 수 있는 관리 기능의 범위에 대한 정보는 41 페이지의 『관리 권한』을 참조하십시오.

Linux용 Communications Server는 다음과 같은 관리 도구를 포함합니다.

- Motif 관리 프로그램(35 페이지의 『Motif 관리 프로그램』 참조)
- 웹 관리 프로그램(37 페이지의 『웹 관리 패키지』 참조)
- 명령행 관리 프로그램(37 페이지의 『명령행 관리 프로그램』 또는 *Linux용 IBM Communications Server 관리 명령 참조서* 참조)
- 서비스점 명령 기능(38 페이지의 『원격 명령 기능(RCF)』 참조)

- 구성 파일(38 페이지의 『구성 파일』 참조)
- 진단 도구(40 페이지의 『진단 도구』 참조)

모든 Linux용 Communications Server 관리 도구는 NOF API를 사용합니다. 또한 그 API를 사용하여 자신의 고유 관리 도구를 작성할 수 있습니다. 자세한 정보는 40 페이지의 『NOF 응용프로그램』을 참조하십시오.

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.

주: Motif 관리 프로그램 **xsnaadmin**은 Linux용 Communications Server 운영을 실시간으로 모니터링하는 데 사용되며 일부 성능 저하를 수반합니다. Linux for System z LPAR에는 직접 접속된 터미널이 없으므로 **xsnaadmin**은 X-Windows 프로토콜을 사용하여 표시장치를 동적으로 갱신합니다. 그러면 추가 네트워크 트래픽 및 CPU 이용이 시스템에 추가됩니다. 더 큰 네트워크 구성의 경우에는 명령행 관리 도구나 웹 관리 패키지를 사용하는 것이 더 좋습니다. 이들은 조회를 통해 갱신하며 영향이 있는 동적 트래픽을 작성하지 않습니다.

직접적인 터미널 하드웨어를 지원하는 시스템의 경우 Motif 관리 프로그램 **xsnaadmin**은 성능에 크게 영향을 미치지 않습니다. 그러나 시스템 성능에 대한 영향은 표시되는 내용 및 표시장치의 갱신 빈도에 따라 달라집니다.

다음과 같은 관리 조작을 사용할 수 있습니다.

- Linux용 Communications Server 자원 정의
- 노드 및 해당 연결성 자원 시작 및 중지
- 정의된 자원의 구성 변경
- 정의된 자원 구성 및 자원의 현재 상태 조회(활동 중인 경우)
- 자원 삭제

Motif 관리 프로그램을 사용하여 노드 자원(Linux용 Communications Server 소프트웨어가 해당 서버에서 실행 중이면 LAN의 모든 서버에 대해) 및 도메인 자원을 관리할 수 있습니다. 각각의 통신 유형(예: 3270 또는 APPC)에 대해 프로그램이 필수 자원 구성의 설정 과정을 안내합니다.

주: Motif 관리 프로그램에 있는 창과 대화상자는 특정 대화상자에서 선택한 사항에 따라 이 안내서에 나와 있는 내용과 다를 수 있습니다.

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.

Motif 관리 프로그램을 시작하려면 먼저 Linux용 Communications Server 소프트웨어가 사용 가능한지 확인하십시오(자세한 정보는 33 페이지의 제 2 장 『Linux용 Communications Server 관리』 참조). X/Motif 응용프로그램의 경우, DISPLAY 환경 변수가 적절한 X 서버를 표시하도록 설정해야 할 수도 있습니다.

백그라운드에서 Motif 관리 프로그램을 시작하려면 다음 명령을 실행하십시오.

xsnaadmin &

시작된 모든 Linux용 Communications Server가 기본 화면에 표시됩니다. 이미 구성된 서버의 경우, 프로그램을 통해 노드를 선택할 수 있으며 프로그램은 선택된 노드의 구성을 표시합니다. 그렇지 않으면, 프로그램이 노드를 선택하도록 요청하는 프롬프트를 표시하고 노드를 정의하는 데 필요한 단계를 안내합니다.

Motif 관리 프로그램을 사용하여 Linux용 Communications Server 자원을 정의 및 관리하는 방법에 대한 자세한 정보는 48 페이지의 『Motif 관리 프로그램 기동』을 참조하거나 프로그램이 제공하는 도움말 화면을 참조하십시오.

주: Motif 관리 프로그램을 사용하면 표준 Linux용 Communications Server 구성에 필요한 모든 필수 매개변수를 설정할 수 있습니다. 고급 매개변수의 경우, Motif 관리 프로그램이 디폴트 값을 제공합니다. 사용자는 필수 구성 정보만 제공하면 되며, 이로서 SNA 통신을 신속하고 쉽게 설정할 수 있습니다.

다른 Linux용 Communications Server 관리 도구(명령행 구성 및 NOF 응용프로그램 포함)는 Motif 관리 프로그램에 표시되는 것보다 더 광범위한 구성 매개변수 및 옵션에 액세스할 수 있는 권한을 제공합니다. 그러나 대부분의 경우, Motif 관리 프로그램은 구성해야 하는 주요 필드를 표시하고 대부분의 사용자가 수정하지 않아도 되는 필드는 숨기므로 이 프로그램에서 필요한 모든 구성을 수행할 수 있습니다. 명령행 구성에서 제공되는 디폴트 값은 Motif 관리 프로그램이 제공하는 디폴트 값과 다를 수 있습니다. Motif 관리 프로그램은 사용자가 수행 중인 구성 타스크에 따라 더 지능적으로 값을 선택할 수 있기 때문입니다.

이러한 추가 기능을 사용해야 할 경우, Motif 관리 프로그램을 계속 사용하여 기본 구성을 설정하고 다른 관리 도구를 사용하여 추가 기능을 지정할 수 있습니다. 나중에 Motif 관리 프로그램을 사용하여 수정된 구성을 관리할 경우, 프로그램은 사용자가 구성한 추가 기능이 Motif 프로그램에 표시되지 않아도 다른 도구를 사용하여 수행한 변경사항을 보유하고 있습니다.

웹 관리 패키지

웹 관리 패키지는 Linux용 Communications Server 지원 웹 페이지(<http://www.ibm.com/software/network/commsserver/support>)에서 다운로드 가능한 관리 도구입니다. 이 패키지는 원격 웹 브라우저에서 Linux용 Communications Server의 관리 기능에 액세스할 수 있도록 스크립트 및 지시사항을 제공합니다.

Apache 또는 IBM HTTP Server와 같은 웹 서버를 사용하는 경우, 이 패키지는 일반적으로 Motif 기반 **xnaadmin** GUI에서 발견되는 조회, 상태, 시작 및 중지와 같은 관리 기능에 대한 원격 액세스를 제공합니다. 이 패키지는 Linux용 Communications Server의 원격 관리를 위해 보안 방법이 있는 관리자를 제공합니다. Linux용 Communications Server가 지원하는 Intel, System p 및 System z 플랫폼에서 동일한 패키지가 실행됩니다.

이 패키지는 Perl-CGI 및 Perl 스크립트를 사용하여 관리 기능을 실행합니다. 거의 모든 웹 브라우저가 이 패키지에 대해 잘 작동합니다. 자원을 조회하고 관리하기 위해 실행되는 명령은 **snaadmin** 명령행 도구에 있습니다(『명령행 관리 프로그램』 참조).

웹 관리 패키지는 노드 및 도메인, 연결, 호스트 자원, TN3270E 서버, APPC, APPN, 조회, 상태 및 진단 패널에서 제어되는 자원에 대한 액세스를 제공합니다. 이 웹 인터페이스를 사용하여 노드의 자원을 조회하거나 세션 상태, 링크, LU 및 노드 조건을 가져올 수 있습니다.

웹 페이지는 표시장치 링크, PU 및 노드 조건에 대한 인터페이스를 제공합니다. 필요에 따라 노드, 링크 및 PU를 시작하거나 중지할 수 있습니다.

호스트 연결 자원은 관리 용도로 간략하게 표시되는 추가 표시와 함께 명령행 관리 호출과 동일한 방식으로 표시될 수 있습니다. 링크, LU 풀, PU 및 TN3270E 서버 세션에 대해 확장 표시가 사용 가능합니다.

원격 액세스는 Linux가 제공한 암호 보호 패키지를 사용하여 사용자 ID 및 암호 인증으로 보호됩니다.

명령행 관리 프로그램

명령행 관리 프로그램인 **snaadmin**을 사용하여 개별 Linux용 Communications Server 자원을 관리하기 위한 명령을 실행할 수 있습니다. Linux 명령 프롬프트나 셸 스크립트에서 직접 **snaadmin**을 사용할 수 있습니다.

특정 Linux용 Communications Server 노드에 명령을 실행하여 노드 자원을 관리하거나, SNA 네트워크 데이터 파일에 명령을 실행하여 마스터 및 백업 서버를 관리하거나, 도메인 구성 파일에 명령을 실행하여 도메인 자원을 관리할 수 있습니다.

Linux용 Communications Server 관리의 개요

서버에서 모든 관리 명령을 실행할 수 있습니다. 하지만 IBM 원격 API 클라이언트에서 명령을 실행할 때 제한사항이 있습니다.

- Windows 클라이언트에는 **snaadmin** 프로그램이 없기 때문에 명령을 실행할 수 없습니다.
- AIX 및 Linux 클라이언트에서는 **query** 또는 **status** 명령을 실행할 수 있습니다. Linux용 IBM Communications Server 관리 명령 참조서에 정의된 일부 다른 관리 명령은, IBM 원격 API 클라이언트에서 실행될 수 있음을 명시적으로 밝히고 있습니다. 그렇지 않은 경우 이 명령들은 서버에서만 사용할 수 있습니다.

다음 명령을 사용하여 명령행 관리에 대한 도움말을 볼 수 있습니다.

- **snaadmin -h**: 명령행 관리에 대한 기본적인 도움말 및 명령행 도움말에 대한 사용법 정보를 제공합니다.
- **snaadmin -h -d**: **snaadmin** 프로그램에 제공할 수 있는 명령 리스트를 제공합니다.
- **snaadmin -h** 명령: 이름 지정된 명령에 대한 도움말을 제공합니다.
- **snaadmin -h -d** 명령: 명령과 함께 지정할 수 있는 구성 매개변수 리스트를 포함하여, 이름 지정된 명령에 대한 자세한 도움말을 제공합니다.

자세한 정보는 *Linux용 IBM Communications Server 관리 명령 참조서*를 참조하십시오.

원격 명령 기능(RCF)

원격 명령 기능(RCF)은 호스트의 NetView 콘솔에서 Linux용 Communications Server 관리를 지원하기 위해 다음 기능을 제공합니다.

- 서비스점 명령 기능(SPCF)은 호스트 NetView 콘솔에 있는 운영자가 Linux용 Communications Server 관리 명령을 실행하여 NetView에서 Linux용 Communications Server를 관리할 수 있도록 합니다.
- UNIX 명령 기능(UCF)은 NetView 운영자가 Linux용 Communications Server 컴퓨터에서 표준 Linux 명령을 실행할 수 있도록 합니다.

RCF에 대한 자세한 정보는 133 페이지의 제 9 장 『NetView에서 Linux용 Communications Server 관리』를 참조하십시오.

구성 파일

Linux용 Communications Server 시스템에 대한 구성 정보는 다음 텍스트 파일에 보유됩니다.

노드 구성 파일

/etc/opt/ibm/sna/sna_node.cfg 파일에는 특정 노드에 해당되는 Linux용 Communications Server 노드 자원에 대한 정보가 있습니다. 이 파일은 노드

가 실행되는 컴퓨터에 상주합니다. 노드 자원에 대한 정보를 포함하고 있으며 노드에서 Linux용 Communications Server를 시작할 때 활성화할 자원을 지정합니다.

이 파일은 사용 가능한 자원의 초기 정의를 제공합니다. 사용자는 다른 관리 도구를 사용하여 자신의 요구사항이 변함에 따라 실행 중인 노드의 자원을 수정할 수 있습니다. 수행하는 수정사항은 자동으로 파일에 저장되므로, 노드를 중지한 후 다시 시작할 경우 수정한 구성을 다시 사용할 수 있습니다.

도메인 구성 파일

`/etc/opt/ibm/sna/sna_domn.cfg` 파일에는 Linux용 Communications Server 도메인 자원(특정 로컬 노드에 연관되지 않는 자원)에 대한 정보가 있습니다. 이 파일의 마스터 사본은 마스터 서버에 상주합니다.

가동 가능 TP 데이터 파일

`/etc/opt/ibm/sna/sna_tps` 파일에는 Linux용 Communications Server가 가동 가능(목표) TP를 시작하기 위해 필요한 정보가 있으며 다른 정보(예: TP 액세스에 필요한 보안 레벨)도 제공합니다. 이 파일은 TP가 실행되는 컴퓨터에 상주합니다.

이 파일에 대한 자세한 정보는 99 페이지의 『TP 정의』를 참조하십시오.

Motif 관리 프로그램, 명령행 관리 프로그램 또는 NOF API를 사용하여 구성을 수정할 수 있습니다. 이들 도구는 모두 노드 구성 파일이나 도메인 구성 파일에 필요한 변경사항을 적절하게 수행합니다. 구성 정보는 일반 텍스트로 저장되므로, `vi`와 같은 표준 ASCII 텍스트 편집기를 사용하거나 `awk` 또는 `sed`와 같은 Linux 유틸리티를 사용하는 셸 스크립트로 직접 파일을 수정할 수도 있습니다. 텍스트 편집기를 사용하여 수행하는 구성 파일 변경사항은 Linux용 Communications Server 시작 이전에 수행해야 합니다. Linux용 Communications Server 구성 파일 형식에 대한 자세한 정보는 *Linux용 IBM Communications Server 관리 명령 참조서*를 참조하십시오.

주: Linux용 Communications Server 구성은 동적 프로세스이므로, Linux용 Communications Server 소프트웨어를 시작하기 전에 전체 구성을 정의하지 않아도 됩니다. 구성 파일은 사용 가능한 자원의 초기 정의를 제공하지만, 사용자가 Linux용 Communications Server 소프트웨어 실행 중 필요에 따라 자원을 추가, 삭제 또는 수정할 수 있습니다. Linux용 Communications Server는 현재 정의를 저장하여 사용자가 시스템을 다시 시작해야 할 때 다시 사용할 수 있도록 합니다.

다음 파일에는 Linux용 Communications Server 클라이언트/서버 네트워크에 대한 정보가 있습니다.

SNA 네트워크 데이터 파일

`/etc/opt/ibm/sna/sna.net` 파일에는 마스터인 서버와 백업 서버로 작동할 수 있

는 서버에 대한 정보가 있습니다. 이 2진 파일은 마스터 서버에 상주합니다. 관리 프로그램이나 NOF API를 사용하여 이 파일의 내용을 수정할 수 있습니다. 이 파일에 대한 자세한 정보 및 수정 방법은 61 페이지의 『클라이언트/서버 기능 구성』을 참조하십시오.

클라이언트 네트워크 데이터 파일

sna_clnt.net 파일에는 IBM 원격 API 클라이언트에 필요한 Linux용 Communications Server에 액세스하는 방법에 대한 정보가 있습니다. 이 텍스트 파일은 클라이언트 컴퓨터에 상주합니다. 표준 ASCII 텍스트 편집기를 사용하여 이 파일의 내용을 수정할 수 있습니다.

이 파일에 대한 자세한 정보 및 수정 방법은 170 페이지의 『클라이언트 네트워크 데이터 파일(sna_clnt.net)』을 참조하십시오. Windows 클라이언트에서 해당 정보를 구성하는 방법에 대해서는 143 페이지의 제 10 장 『Linux용 Communications Server 클라이언트/서버 시스템 관리』를 참조하십시오.

NOF 응용프로그램

Linux용 Communications Server NOF API는 명령행 관리 프로그램과 같은 관리 기능을 제공하여, 사용자가 Linux용 Communications Server 자원을 정의 및 관리할 수 있게 합니다. 이는 사용자 고유의 응용프로그램을 작성하여 Linux용 Communications Server를 관리할 수 있음을 의미합니다.

자세한 정보는 *Linux용 IBM Communications Server NOF 프로그래머 안내서*를 참조하십시오.

진단 도구

Linux용 Communications Server는 Linux용 Communications Server 작동 중 발견되는 문제점을 진단하고 정정하는 데 도움이 되는 몇 가지의 진단 도구를 제공합니다.

- 문제점이나 예외(문제점의 원인을 표시할 수 있는 비정상 상태)를 발견하는 구성요소는 항목을 오류 로그 파일에 기록합니다. 또한 중요한 모든 시스템 이벤트를 감사 로그 파일에 기록할 수 있습니다. 기록되는 이벤트 유형(문제점, 예외 또는 감사)을 판별할 수 있습니다. 클라이언트/서버 네트워크 구성에서, 사용자는 모든 서버에서 기록할 이벤트 유형에 대한 글로벌 설정을 지정한 후 필요에 따라 개별 서버에서 이 설정을 대체할 수 있습니다.
- Linux용 Communications Server는 또한 Linux용 Communications Server 자원의 현재 및 최대 사용에 대한 정보를 기록하기 위해 사용되는 사용 로그 파일도 유지합니다.
- 각 유형의 로그 정보를 보유하기 위해 사용되는 파일의 이름 및 디렉토리를 지정할 수 있습니다. 오류 및 감사 로그 정보를 같은 파일로 송신할 수도 있습니다. 클라이언트

엔트/서버 시스템에서, 모든 서버에서 하나의 서버에 있는 중앙 로그 파일로 메시지를 송신하거나(중앙 로깅) 각 서버에 있는 개별 파일에 로그 메시지를 송신할 수 있습니다.

- 로그 파일은 텍스트 파일로 생성되며 vi와 같은 표준 ASCII 텍스트 편집기를 사용하여 볼 수 있습니다.
- 전체 로깅(각 메시지의 로그 파일에 로그 원인의 세부사항과 필요한 조치가 포함되는)이나 간결 로깅(메시지 텍스트 및 로그의 소스 요약만 포함되는)을 선택할 수 있습니다. 간결 로깅을 사용할 경우, 추가 정보가 필요하면 snahelp 명령행 유틸리티를 사용하여 특정 메시지 번호에 해당되는 전체 원인 및 조치 텍스트를 확보할 수 있습니다.
- 특정 이벤트가 자주 발생되어 해당 로그 파일이 동일한 로그 메시지의 많은 인스턴스를 필터링하는 경우, 하나 이상의 특정 로그 메시지가 한 번만 로그되도록 필터를 설정할 수 있습니다. 그러면 동일한 로그 메시지의 모든 후속 인스턴스가 무시되어 로그 파일에 기록되지 않습니다.
- 일부 오류 조건에 대해서는 Linux용 Communications Server가 메시지를 Linux 콘솔에 송신하여 운영자에게 경고하고 문제점 메시지를 오류 로그 파일에 기록합니다.
- 대부분의 구성요소는 해당 활동을 기록하는 추적 파일을 생성할 수 있습니다. 추적을 수행하면 Linux용 Communications Server 구성요소의 성능이 저하되므로 일반적으로 추적을 사용하지 않습니다.
- 명령행 유틸리티를 사용하면 추적 파일을 필터하여 특정 정보를 발취할 수 있으며, 추적 정보를 형식화하여 내용을 해석하거나 메시지 흐름 요약을 생성할 수 있습니다. 형식화된 출력 파일은 vi와 같은 표준 ASCII 텍스트 편집기를 사용하여 볼 수 있습니다.
- Linux용 Communications Server는 경고를 생성하여 호스트 컴퓨터에 있는 NetView 프로그램에 송신할 수 있습니다. 이 경고는 다음과 같습니다.
 - 연결 문제점에 대한 정보를 제공하기 위해 연결성 구성요소에서 생성되는 링크 경고
 - MS API를 사용하는 응용프로그램이 제공하는 경고

Linux용 Communications Server 로그 메시지, Linux용 Communications Server 추적 기능 사용 및 추적 파일 해석에 대한 정보는 *Linux용 IBM Communications Server 진단 안내서*를 참조하십시오.

MS API 사용에 대한 정보는 AIX 또는 *Linux용 IBM Communications Server MS 프로그래머 안내서*를 참조하십시오.

관리 권한

Linux용 Communications Server 관리 도구는 SNA 자원을 관리할 수 있는 권한을 가진 제한된 『SNA 관리자』 그룹이 사용하도록 만들어진 것입니다. 이를 위해 실행 파

Linux용 Communications Server 관리의 개요

일은 그룹 소유권이 sna인 시스템 관리자 로그인 root가 소유합니다. 그룹 sna의 구성원인 사용자만 Linux용 Communications Server 자원을 수정, 시작 또는 중지할 수 있습니다. SNA 관리자 권한을 가지고 있는 사용자는 이 그룹의 구성원이어야 합니다.

표준 Linux용 Communications Server 설치에서, 그룹 sna의 구성원이 아닌 사용자는 Linux용 Communications Server 관리 도구를 전혀 실행할 수 없습니다. 필요하다면, 이들 사용자가 읽기 전용 모드에서 도구를 실행하여 구성 및 상태 정보는 볼 수 있지만 자원을 수정, 시작 또는 중지하지는 못하도록 할 수 있습니다. 이렇게 하려면 **chmod**를 사용하여 해당되는 실행 파일에 대한 읽기 및 실행 권한을 사용자에게 제공하십시오.

관리 도구	실행 파일
Motif 관리 프로그램	/opt/ibm/sna/bin/X11/xsnaadmin
명령행 관리 프로그램	/opt/ibm/sna/bin/snaadmin

그러면 사용자가 해당되는 관리 도구를 실행하고 정보를 볼 수는 있지만, Linux용 Communications Server는 계속해서 sna 그룹이 아닌 사용자가 자원을 수정, 시작 또는 중지하지 못하도록 합니다.

주: 위에 설명된 대로 파일 권한을 수정할 경우, Linux용 Communications Server PTF 또는 새 릴리스를 설치한 후 이 절차를 반복해야 합니다.

Linux용 Communications Server 구성 계획

구성 변경을 수행하기 전에 철저히 계획하는 것이 중요합니다. 사용자가 수행하는 변경 사항이 로컬 노드 내의 사용자들뿐만 아니라 전체 네트워크에 걸쳐 있는 사용자들에게 피해를 줄 수 있습니다.

네트워크의 토폴로지에 대해 수행하는 변경사항의 다이어그램을 그려보는 것이 좋습니다. 다른 노드로의 연결을 추가하거나 제거할 경우, 사용자 노드와 다른 노드들을 보여주는 그림을 그려보십시오. Motif 관리 프로그램을 사용하여 기존의 모든 연결에 대한 구성 정보를 수집하고 그 정보를 다이어그램에 추가할 수 있습니다.

다이어그램에 새 자원들을 추가하면, 그것들이 기존 것과 중복되는지 혹은 이름이 충돌하지 않는지 살펴보기가 용이합니다. 마찬가지로, 다이어그램은 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.

수행해야 하는 변경사항을 판별하고 나면 필요한 구성 정보를 수집할 수 있습니다. Motif 관리 프로그램의 온라인 도움말 파일에 있는 **타스크 시트**나 『계획 워크시트』에 설명된 계획 워크시트는 특정 Linux용 Communications Server 기능에 관한 구성 정보의 수집 과정을 안내할 수 있습니다.

계획 워크시트

Linux용 Communications Server의 자원 구성을 시작하기 전에 새 자원에 대한 모든 구성 데이터를 수집하십시오. 지원해야 하는 특정 기능이나 응용프로그램에 대한 모든 정보를 기록하려면 175 페이지의 부록 A 『구성 계획 워크시트』에 있는 계획 워크시트를 사용하십시오.

네트워크 관리자, 호스트 관리자, 응용프로그램 프로그래머 및 최종 사용자와 같은 여러 소스로부터 구성 정보를 수집해야 합니다.

다른 노드로의 연결을 시도하는 경우 해당 노드에 있는 관리자가 주요 연락처입니다. 노드 관리자는 그 노드에 있는 모든 자원의 이름, 주소 및 특성을 사용자에게 알려줄 수 있습니다. 종종, 로컬 노드와 원격 노드에서 일치하는 구성 매개변수를 입력했는지 확인해야 합니다.

타스크 시트

Motif 관리 프로그램에 있는 온라인 도움말 화면에는 특정 구성 타스크를 위한 지침을 제공하는 **타스크 시트**가 있습니다. 타스크 시트는 구성 정보를 입력할 때 사용할 대화상자의 모든 도움말 화면에 대한 포인터를 포함하고 있습니다. 이 포인터를 사용하여 도움말을 찾아보고 정확히 어떤 데이터를 수집해야 하는지 확인할 수 있습니다.

타스크 시트는 또한 구성 정보를 입력할 때 사용해야 하는 개별 창 및 대화상자 각각에 대한 자세한 도움말도 언급합니다. 해당 도움말 화면들은 입력해야 하거나 선택해야 하는 각 필드를 설명합니다.

로컬 시스템에서 Linux용 Communications Server 사용 기능화 및 사용 불가능화

이 절에서는 Linux 서버에서 Linux용 Communications Server 소프트웨어를 사용 가능 및 사용 불가능하게 하는 방법을 설명합니다.

Linux용 Communications Server 도구(예: Motif 관리 프로그램)를 사용하려면 먼저 Linux용 Communications Server 소프트웨어를 사용 가능하게 해야 합니다. 보통 소프트웨어는 사용자가 Linux용 Communications Server를 설치하면 자동으로 사용 가능하게 되지만 필요할 경우 수동으로 사용 가능하게 할 수 있습니다.

Linux용 Communications Server 프로그램의 경로 지정

Linux용 Communications Server 실행 가능 프로그램은 Linux용 Communications Server의 특정 디렉토리에 저장됩니다. 프로그램을 실행할 때 이 디렉토리 경로를 지정해야 합니다. 처음으로 프로그램을 실행하기 전에 PATH 환경 변수에 디렉토리를 추가하거나 프로그램을 실행할 때마다 디렉토리를 포함하는 경로를 지정할 수 있습니다.

Motif 관리 프로그램은 `/opt/ibm/sna/bin/X11` 디렉토리에 저장되며, 기타 프로그램은 `/opt/ibm/sna/bin` 디렉토리에 저장됩니다. 이러한 디렉토리를 `.login`이나 `.profile` 파일의 PATH 환경 변수 정의에 추가하면 Linux용 Communications Server는 자동으로 프로그램을 찾습니다. 또는, 다음 예제처럼 프로그램을 실행할 때 디렉토리 이름을 지정할 수 있습니다.

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

```
/opt/ibm/sna/bin/snaadmin query_node
```

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

이 메뉴얼에 표시된 샘플 명령행은 사용자가 PATH 환경 변수에 디렉토리를 추가한 것으로 가정하므로 디렉토리 이름을 포함하지 않습니다.

Linux용 Communications Server 사용 가능화

이 절에서는 서버로 설치된(즉, SNA 노드 구성요소를 설치한) 컴퓨터에서 Linux용 Communications Server를 사용 가능하게 하는 방법을 설명합니다. 클라이언트에서 Linux용 Communications Server를 사용 가능하게 할 경우, 169 페이지의 『AIX 또는 Linux의 원격 API 클라이언트 사용 가능화 및 사용 불가능화』를 참조하십시오.

로컬 노드를 구성하거나 관리하려면(로컬로 또는 원격 Linux용 Communications Server 노드에서) 먼저 로컬 시스템에서 Linux용 Communications Server를 사용 가능하게 해야 합니다.

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

```
sna start [ -s ] [  
  -m kernel_memory_limit] [  
-t ]
```

주: `sna start` 명령을 사용하면 Linux용 Communications Server 소프트웨어는 이 명령이 실행된 디렉토리를 현재 작업 디렉토리로 사용하고 해당 디렉토리에서 하나 이상의 열린 파일 설명자를 유지보수합니다. 즉, Linux용 Communications Server

로컬 시스템에서 Linux용 Communications Server 사용 가능화 및 사용 불가능화

소프트웨어가 실행되고 있는 동안에는 해당 디렉토리가 있는 파일 시스템을 마운트 해제할 수 없습니다. 이 문제점이 발생하지 않도록 하려면 마운트 해제할 필요가 없는 파일 시스템의 디렉토리에서 Linux용 Communications Server 소프트웨어를 시작해야 합니다. 예를 들어, **sna start** 명령을 사용하기 전에, `cd /`를 사용하여 루트 디렉토리로 변경할 수 있습니다.

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.

sna start 명령의 매개변수 및 옵션은 다음과 같습니다.

-s Linux용 Communications Server가 시스템 콘솔에 메시지를 쓰지 않아야 함을 지정합니다. 이 옵션을 사용하지 않을 경우, Linux용 Communications Server는 종료할 때 콘솔에 메시지를 기록하고 콘솔과 로그 파일에 특정 오류 로그 메시지 텍스트를 기록합니다.

-m kernel_memory_limit

Linux용 Communications Server가 언제든지 사용해야 하는 최대 커널 메모리 양(KB)을 지정합니다. (커널 메모리는 내부 데이터 구조에 사용됩니다.) Linux용 Communications Server 구성요소가 현재 Linux용 Communications Server 구성요소에 할당된 총 메모리 양이 이 한계를 초과하도록 커널 메모리를 할당하려고 할 경우, 할당에 실패합니다.

이 옵션을 사용하지 않을 경우, 커널 메모리 사용에 제한을 받지 않습니다.

-t 커널 구성요소 사이의 모든 인터페이스에 대한 추적과 클라이언트/서버 추적을 활성화합니다. (이 옵션은 DLC 추적을 켜지는 않습니다.) 추적을 사용하면 시작하는 동안 발생하는 문제점을 진단할 수 있습니다. 이 옵션을 사용하지 않으면 모든 인터페이스에서 추적이 비활성화됩니다. 이 경우 사용자는 명령행 관리 프로그램 **snaadmin**을 사용하여 필요에 따라 특정 인터페이스에 대해 추적을 활성화할 수 있습니다.

모든 인터페이스에 추적을 수행하면 Linux용 Communications Server 구성요소의 성능이 저하됩니다. 소프트웨어가 사용 가능하게 되면, 명령행 관리 프로그램 **snaadmin**을 사용하여 추적이 필요하지 않은 인터페이스에 대해 추적을 중지할 수 있습니다. 추적에 대한 자세한 정보는 *Linux용 IBM Communications Server* 진단 안내서를 참조하십시오.

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

초기화에 실패하면, 메시지는 오류 원인에 대한 정보 및 (적절한 경우) Linux 운영 체제 오류 메시지와 같은 추가 정보가 포함됩니다. 표준 오류 기록 장치에 기록되는 텍스트에는 오류 로그 파일에서 추가 정보를 볼 수 있음을 표시하는 메시지도 포함됩니다. **sna start** 명령은 오류 특성을 표시하는 0이 아닌 종료 코드와 함께 종료합니다.

종료 코드 값에 대한 자세한 정보는 *Linux용 IBM Communications Server* 진단 안내서를 참조하십시오.

sna start 명령의 고급 옵션

일부 경우, 특히 Linux용 Communications Server의 새 구성을 테스트하는 경우에는 임시 파일(/etc/opt/ibm/sna/sna_node.cfg 및 /etc/opt/ibm/sna/sna_domn.cfg 파일에 있는 표준 구성이 아니라)에 저장한 구성으로 Linux용 Communications Server를 시작하는 것이 바람직합니다. **sna start** 명령에서 다음 추가 옵션을 사용하여 이렇게 할 수 있습니다.

-n *node_config_file*

-d *domain_config_file*

*node_config_file*은 /etc/opt/ibm/sna/sna_node.cfg 대신 노드 구성을 저장한 파일의 전체 경로 이름이며, *domain_config_file*은 /etc/opt/ibm/sna/sna_domn.cfg 대신 도메인 구성을 저장한 파일의 전체 경로 이름입니다.

주: 이러한 옵션은 일반 용도에 사용되지 않습니다. 일반 용도로 사용하려는 특정 요구 사항이 없으면 사용하지 마십시오.

snagetpd 명령은 항상 표준 구성 파일에서 정보를 수집하므로 Linux용 Communications Server가 이러한 옵션에서 실행할 경우에 제대로 작동하지 않습니다. **snagetpd**를 사용하기 전에 이러한 옵션 없이 Linux용 Communications Server를 시작하여 표준 구성 파일로 실행하는지 확인하십시오.

Linux용 Communications Server 사용 불가능화

서버에서 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: 3270 에뮬레이션 프로그램)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. 클라이언트에서 Linux용 Communications Server 소프트웨어를 사용 불가능하게 하면 클라이언트에서 실행 중인 프로그램이 Linux용 Communications Server 기능 액세스를 중지합니다.

로컬 시스템에서 Linux용 Communications Server 사용 가능화 및 사용 불가능화

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. Motif 관리 프로그램이나 명령행 관리 프로그램을 사용하여 활동 중인 사용자 세부사항을 보십시오.

Linux용 Communications Server 소프트웨어를 사용 불가능하게 할 때 3270 에뮬레이션 프로그램이 노드에서 LU를 사용하고 있을 경우, 해당 LU를 사용하는 모든 3270 에뮬레이션 세션이 종료됩니다. 프로그램은 계속 실행되지만 사용자는 소프트웨어가 다시 사용 가능하게 될 때까지 세션을 사용할 수 없습니다. APPC, CSV, LUA, NOF 또는 MS API를 사용하는 응용프로그램은 COMM_SUBSYSTEM_ABENDED 리턴 코드에 의해, CPI-C 응용프로그램은 CM_PRODUCT_SPECIFIC_ERROR 리턴 코드에 의해 통지됩니다.

Linux용 Communications Server 소프트웨어를 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 정보는 *Linux용 IBM Communications Server* 진단 안내서를 참조하십시오.

Motif 관리 프로그램 사용

Motif 관리 프로그램은 사용자에게 친숙한 Linux용 Communications Server 구성용 인터페이스를 제공합니다. 이 프로그램은 구성 프로세스를 안내하고 작업 가능한 구성을 작성하기 위해 제공해야 하는 정보를 최소화하기 때문에 Linux용 Communications Server 관리를 위한 권장 도구입니다.

또한 Motif 관리 프로그램을 사용하여 활동 중인 Linux용 Communications Server 시스템을 관리할 수도 있습니다. 관리 프로그램을 사용하면 Linux용 Communications Server가 활동 중일 때 변경을 수행한 후 구성에 변경을 적용할 수 있습니다. 자원을 추가, 수정 및 제거하고(대부분의 경우, 노드 및 해당 자원이 활동 중인 경우에도) 연속 조작을 위해 즉시 수정한 구성을 사용할 수 있습니다.

Motif 관리 프로그램은 구성에 사용되는 것과 동일한 인터페이스를 통해 최신 상태 정보를 표시하여, 도메인 및 노드 자원 둘 다에 대한 상태 정보에 쉽게 액세스할 수 있도록 합니다.

또한 구성 및 시스템 관리에도 Linux용 Communications Server 명령을 사용할 수 있습니다. 구성 및 관리 명령 요약은 58 페이지의 『명령행 관리 프로그램 사용』에 제공되어 있습니다.

Motif 관리 프로그램 기동

Linux용 Communications Server에 대해 Motif 관리 프로그램을 사용하려면 먼저 44 페이지의 『Linux용 Communications Server 사용 기능화』에 설명된 것처럼 Linux용 Communications Server가 사용 가능해졌는지 확인해야 합니다. (X/Motif 응용프로그램과 마찬가지로 DISPLAY 환경 변수가 적절한 X 서버를 지시하도록 설정해야 할 수도 있습니다.)

백그라운드에서 Motif 관리 프로그램 실행을 시작하려면 다음 명령을 실행하십시오.

xснаadmin &

클라이언트/서버 환경에서 Linux용 Communications Server는 도메인 창을 표시합니다.

독립형 시스템의 경우, Linux용 Communications Server는 보통 노드 창을 표시합니다. 그러나 아직 로컬 노드를 구성하지 않았으면 최초 노드 구성에 대한 도움말을 제공하는 도움말 화면을 표시합니다.

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

자원 창

도메인 창 및 노드 창은 사용자가 필요로 하는 대부분의 정보를 보여주며 추가 정보에 쉽게 액세스할 수 있도록 합니다. 이 창을 통해 로컬 네트워크에 있는 자원에 대한 정보를 쉽게 표시할 수 있습니다.

도메인 창은 정의된 모든 노드를 표시하고 노드를 추가, 삭제, 시작 및 중지할 수 있도록 합니다. 노드를 두 번 누르면 해당 노드의 노드 창이 표시됩니다.

노드 창은 특정 노드에 대한 모든 주요 자원을 보여줍니다.

도메인 및 노드 창의 메뉴는 다음 기능을 제공합니다.

선택 이 메뉴의 기능은 도메인 창에서 현재 선택된 노드나 노드 창에서 현재 선택된 항목에 관련됩니다. 이 메뉴에서 노드를 시작 또는 중지할 수 있고 노드를 확대/축소하여 노드 창을 표시할 수 있습니다. 노드 창에서 항목을 선택할 경우, 이 메뉴에 있는 제어를 사용하여 항목을 제어, 수정 또는 삭제하거나 현재 선택된 분할창에서 새 항목을 추가할 수 있습니다.

서비스 이 메뉴는 공통 타스크용 노드를 구성하는 데 필요한 모든 대화상자에 쉽게 액세스할 수 있도록 합니다. 이 메뉴를 사용하여 자원을 추가 또는 수정하거나 구성 및 관리 타스크에 대한 도움말을 볼 수 있습니다.

진단 이 메뉴의 항목에서 로깅 및 추적을 제어할 수 있습니다.

창 이 메뉴에서 다른 창에 쉽게 액세스할 수 있습니다. 이 창에는 다음이 포함됩니다.

- LU 풀 창
- CPI-C 목적지 이름 창

선택하는 자원과 옵션에 따라, 관리 프로그램은 추가 자원 창, 구성 대화상자 또는 상태 로그를 제시할 수 있습니다. 또한 구성할 특정 자원을 선택할 수 있는 문맥 대화상자, 선택사항 확인을 요청하는 확인 대화상자, 피드백이나 오류 정보를 제공하는 메시지 팝업도 표시됩니다. 각각의 창 및 대화상자에는 도움말 옵션도 포함됩니다.

도메인 창

도메인 창은 사용 중인 시스템에서 Linux용 Communications Server 도메인의 활동 중인 각 SNA 노드를 표시합니다. (노드는 Linux용 Communications Server가 노드에서 실행되고 있지 않을 경우 도메인 창에 표시되지 않습니다.) 각 노드는 시스템 이름을 사용하여 식별됩니다. 도메인 창은 도메인에서 각 노드의 현재 상태도 표시합니다.

주: 도메인 창의 노드 리스트에서 예기치 않게 서버가 누락된 경우, 서버가 켜져 있는지, 그리고 서버에서 Linux용 Communications Server 소프트웨어가 실행 중인지 확인하십시오. 필요하다면 **sna start** 명령을 사용하여 해당 노드에서 Linux용 Communications Server 소프트웨어를 시작하십시오(44 페이지의 『Linux용 Communications Server 사용 가능화』 참조).

도메인에서 하나의 노드는 항상 해당 도메인의 구성 서버로 식별됩니다. 도메인 창은 노드 옆에 『마스터』라고 하는 단어를 표시합니다. 마스터 구성 서버에는 항상 도메인 자원에 대한 구성 정보가 포함됩니다. 백업 구성 서버는 단어 『백업』으로 식별됩니다. 백업 구성 서버에는 도메인 자원의 구성 정보 사본이 있습니다.

50 페이지의 그림 13에 도메인 창의 예제가 있습니다.



그림 13. Linux용 Communications Server 도메인 창

도메인에서 활동 중인 노드(Linux용 Communications Server가 실행 중인 노드)가 구성되지 않은 경우, Linux용 Communications Server는 노드를 구성하도록 요청하는 프롬프트를 표시합니다.

주: 도메인 창에는 IBM 원격 API 클라이언트가 나열되지 않습니다. 클라이언트는 Linux용 Communications Server 자원(SNA 노드)을 사용하여 SNA 자원에 액세스합니다.

도메인 창에서 다음 관리 작업을 수행할 수 있습니다.

도메인에서 노드 시작 또는 중지

이 창에서 노드의 행을 선택하고 시작 또는 중지 버튼을 누르십시오. (또는 노드의 행을 누르고 선택 메뉴에서 노드 시작 또는 노드 중지를 선택할 수도 있습니다.)

특정 노드 관리

도메인 창에서 해당 노드의 행을 두 번 누르십시오. (또는 노드의 행을 누르고 선택 메뉴에서 등록정보를 선택할 수 있습니다. 창 메뉴에서 노드에 해당되는 창을 선택할 수도 있습니다.)

관리할 노드를 선택할 때 Linux용 Communications Server는 51 페이지의 그림 14에 표시된 대로 노드 창을 표시합니다. (독립형 시스템의 경우, 도메인에는 단 하나의 노드만 있기 때문에 Linux용 Communications Server는 도메인 창을 표시하지 않습니다. 대신, Linux용 Communications Server는 사용자가 관리 프로그램을 시작할 때 노드 창을 즉시 표시합니다.)

도메인의 서버 리스트에 노드 추가

노드에 해당되는 행을 누르고 선택 메뉴에서 구성 서버 작성을 선택하십시오.

도메인의 서버 리스트에서 노드 제거

노드에 해당되는 행을 누르고 선택 메뉴에서 구성 서버 제거를 선택하십시오.

도메인에서 모든 노드에 대한 로깅 구성

진단 메뉴에서 로깅을 선택하십시오.

특정 노드의 추적 켜기 또는 끄기

노드에 해당되는 행을 누르고 진단 메뉴에서 선택된 노드 추적 켜기를 선택하십시오.

도메인 자원에 대한 정보 얻기

창 메뉴에서 옵션을 선택하십시오. 공유 도메인 자원 외에도, 창 메뉴는 도메인에 있는 각각의 노드 창을 나열합니다.

노드 창

샘플 노드 창은 그림 14에 나타납니다. 제목 표시줄에는 시스템의 이름이 표시됩니다.

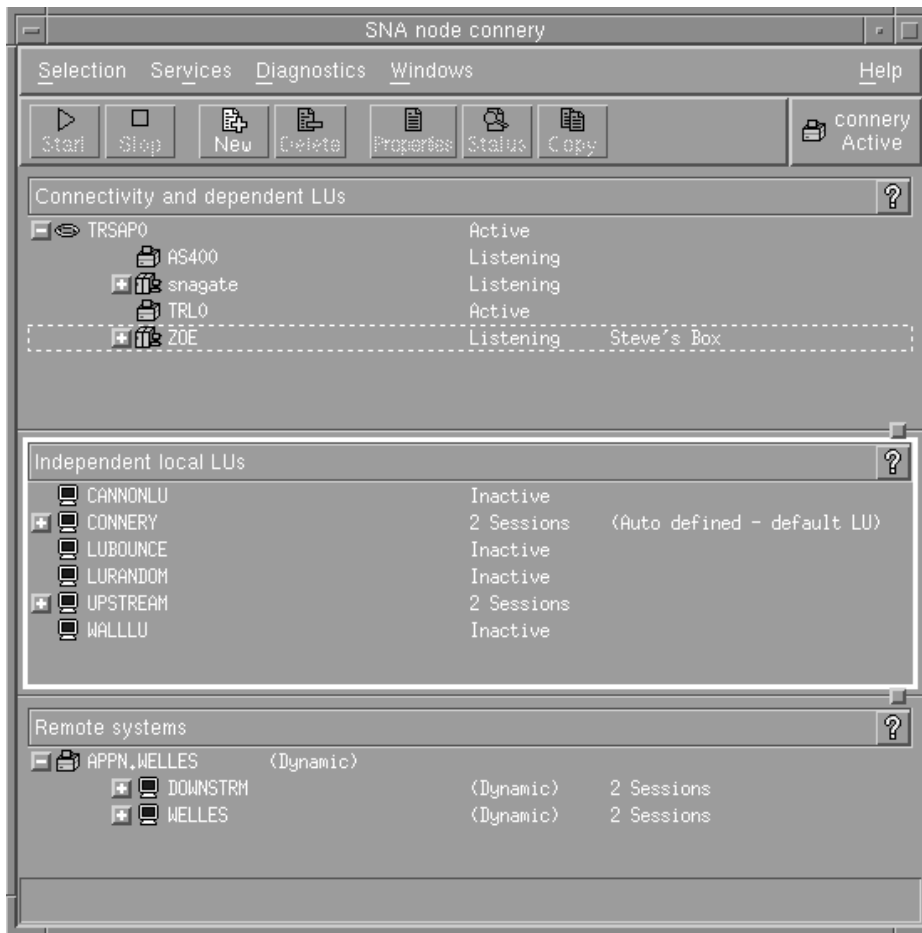


그림 14. 노드 창

노드 창에서 Linux용 Communications Server 노드의 모든 자원 및 구성요소를 추가, 삭제, 수정 및 관리할 수 있습니다. 창에 표시되는 자원 레이아웃은 자원들 간의 관계를 보여주므로 사용자는 이를 통해 표시되는 자원을 제어할 수 있습니다.

노드 창의 맨 위 오른쪽에 있는 노드 상자는 노드 상태가 활동 중인지 또는 활동 중이 아님인지 표시합니다.

노드에 정의된 포트, 로컬 LU 및 원격 노드는 항상 표시됩니다. 노드 창은 상위 포트 아래에 있는 각 링크 스테이션 및 상위 링크 스테이션 아래에 있는 각 종속 LU를 보여줍니다. 또한 로컬 LU와 원격 노드 아래에 있는 상대방 LU도 보여줍니다.

노드 창의 본문은 노드의 여러 자원 유형에 따라 다음과 같은 분할창으로 구분됩니다.

연결성 분할창

노드 창의 맨 위 분할창은 포트, 각 포트의 링크 스테이션이나 PU, 특정 링크 스테이션 또는 PU의 종속 LU를 포함하여 노드에 해당되는 연결 자원을 나열합니다. 이 창은 자원마다 현재 상태 정보를 표시합니다.

독립 로컬 LU 분할창

중간 분할창은 노드의 독립 LU를 표시합니다. 이 창은 LU마다 LU를 사용하는 세션에 대한 정보도 표시합니다.

원격 시스템 분할창

하단 분할창은 원격 노드 및 상대방 LU에 대한 정보를 표시합니다. 각 원격 노드나 상대방 LU를 위한 세션 정보 역시 보여줍니다.

분할창의 상대적인 크기를 변경하려면 분할창 사이의 경계를 누른 후 드래그하십시오.

분할창을 눌러서 선택할 수 있습니다. 자원에 해당되는 행을 눌러 분할창 내의 특정 자원을 선택할 수도 있습니다. 항목 구성을 보거나 수정하려면 항목을 두번 누르십시오. (이 창에 있는 버튼 및 메뉴를 사용하여 특정 자원에 대한 구성 정보에 액세스할 수도 있습니다.)

나열된 항목마다 항목에 속하는 자원이 해당 항목 정보 내에 내포됩니다. 예를 들어, 링크 스테이션은 속해 있는 포트 아래에 그룹화됩니다. 해당 항목의 자원이 현재 표시되지 않은 경우 항목 옆에 있는 확장 버튼



을 눌러 자원을 표시하거나, 축소 버튼



을 눌러 항목의 자원을 숨길 수 있습니다.

노드 창에서 다음과 같은 관리 작업을 수행할 수 있습니다.

자원 시작 또는 중지

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

항목의 새 자원 추가

항목을 선택하고 새로 작성 버튼을 누르십시오. (또는 선택 메뉴에서 새로 작성을 선택하십시오.) 예를 들어, 포트의 링크 스테이션을 추가하려면 포트를 선택하고 새로 작성 버튼을 누르십시오.

자원 삭제

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

자원의 구성 보기 또는 수정

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

자원의 상태 정보 보기

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

자원의 구성 복사

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

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

자원 항목

창에서 자원 레이아웃은 자원 사이의 관계를 보여줍니다.

항목에 연관된 하위 항목이 하나 이상 있으면 항목 옆에 확장 또는 축소 버튼이 표시됩니다. 확장 버튼은 연관된 하위 항목이 숨겨져 있음을 표시합니다. 확장 버튼을 눌러 하위 항목을 표시할 수 있습니다. 축소 버튼은 하위 항목이 표시되어 있음을 나타냅니다. 축소 버튼을 눌러 하위 항목을 숨길 수 있습니다. 항목 옆에 버튼이 없으면 항목에 연관된 하위 자원이 없는 것입니다.

예를 들어, 링크 스테이션이 특정 포트와 연관됩니다. 노드 창의 연결 분할 창에서, 상위 포트 아래에 링크 스테이션이 표시되며 포트와 연관된 다른 모든 링크 스테이션도 표시됩니다. 포트는 항상 표시되지만 연관된 링크 스테이션 리스트의 표시 여부를 선택할 수 있습니다. 마찬가지로, 연관된 LU 리스트를 가지고 있는 링크 스테이션을 확장하여 LU를 표시하거나 축소하여 숨길 수 있습니다.

상위 자원은 항상 하위 자원에 이전에 구성해야 하며, 상위 자원을 삭제하면 모든 하위 자원도 삭제됩니다.

도구 모음 버튼

자원 창에는 공통 기능을 쉽게 수행할 수 있는 도구 모음 버튼이 있습니다. Linux용 Communications Server의 도구 모음이 그림 15에 나타나 있습니다.



그림 15. Linux용 Communications Server 도구 모음

각 자원 창의 도구 모음에 모든 버튼이 표시되는 것은 아닙니다. 버튼의 조작이 현재 선택된 항목에 유효하지 않으면(또는 조작에 해당되는 항목을 선택해야 하는데 선택하지 않은 경우) 버튼 윤곽이 희미하게 표시되어 기능을 선택할 수 없습니다. (버튼을 누를 수 없습니다.) 다음 버튼들이 자원 창 위에 나타날 수 있습니다.



선택된 항목을 시작합니다.



선택된 항목을 종료합니다.



새 자원 항목을 추가합니다. (노드 창에서 선택한 분할창에 자원을 추가합니다.)



선택된 자원을 삭제합니다.



선택된 항목에 대한 대화상자를 열어 항목의 구성을 보거나 수정합니다.



선택된 항목을 복사합니다. 이 버튼을 누르면 해당 필드가 선택한 항목의 구성을 복제하는 대화상자가 열립니다. 대화상자의 필드를 완료하여(새 항목의 이름을 입력하여) 새 자원을 추가하십시오.



선택된 항목의 현재 상태를 표시합니다.

포트 및 링크 스테이션과 같은 많은 자원들은 활동 중에는 수정될 수 없습니다. 그러나 자원을 선택한 후 등록정보 버튼을 눌러 대화상자를 열어서 활동 중인 자원의 매개변수를 보거나, 상태 버튼을 눌러 자원에 대한 자세한 상태 정보를 볼 수 있습니다.

자원 대화상자

자원 대화상자는 자원에 대한 현재 구성 정보를 보여줍니다. 유형 0-3의 LU에 대한 샘플 대화상자는 56 페이지의 그림 16에 표시됩니다.

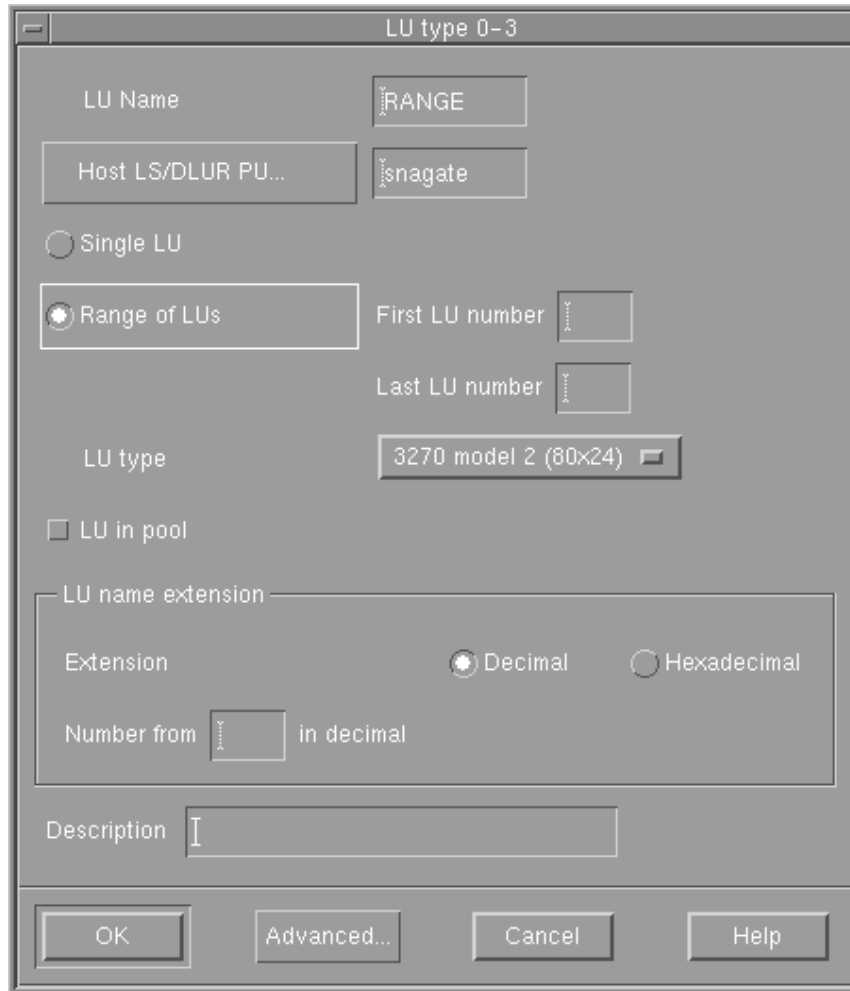


그림 16. 샘플 대화상자

자원 대화상자는 구성 프로세스를 안내하고 가능할 때마다 디폴트 값을 제공합니다. 예를 들어, 종속 LU를 추가할 경우 Motif 관리 프로그램은 자동으로 LU 번호 필드를 사용자가 지정하는 링크 스테이션에서 사용 가능한 LU 번호로 채웁니다. 필수 값을 제공하지 않으면 프로그램은 사용자가 제공해야 하는 정보를 표시하는 메시지 팝업을 표시합니다.

대부분의 대화상자는 설명 필드를 제공합니다. 이 필드에 입력하는 정보는 자원이 표시되는 창에 표시됩니다.

자원 대화상자에서 정보를 변경할 수 있는 권한이 있으면(새 항목을 추가하거나 기존 항목을 수정할 경우), 대화상자에는 확인 및 취소 버튼이 포함됩니다. 완료되면 확인 버튼을 누르십시오. 또는 자원 구성을 변경하지 않고 종료하려면 취소 버튼을 누르십시오.

자원 대화상자에서 정보를 변경할 수 없는 경우(예: 자원 구성이 활동 중일 때 수정할 수 없는 경우), 대화상자에는 확인 버튼 대신 닫기 버튼이 포함됩니다. 대화상자에서 정보 보기가 끝나면 이 버튼을 누르십시오.

대화상자에 대한 문맥 도움말을 보려면 도움말 버튼을 누르십시오.

주: 기본 Motif 대화상자는 주요 구성 필드만 표시합니다. Linux용 Communications Server는 고급 필드에 대해 디폴트 값을 제공합니다. 고급 구성 매개변수에 액세스하려면 고급 버튼을 누르십시오. 고급 매개변수를 조정하기로 결정했으면, 고급 대화상자를 열기 전에 기본 대화상자를 완료하십시오. 이 대화상자는 사용자가 기본 매개변수에 대해 입력하는 값에 따라 변경될 수 있기 때문입니다. 고급 구성 필드에 대한 정보는 Motif 관리 프로그램에 대한 온라인 도움말을 참조하십시오.

상태 대화상자

자원을 선택하고 상태 버튼을 누를 경우, Motif 관리 프로그램은 그림 17에 표시된 대로 자원에 대한 자세한 상태 정보를 표시합니다.

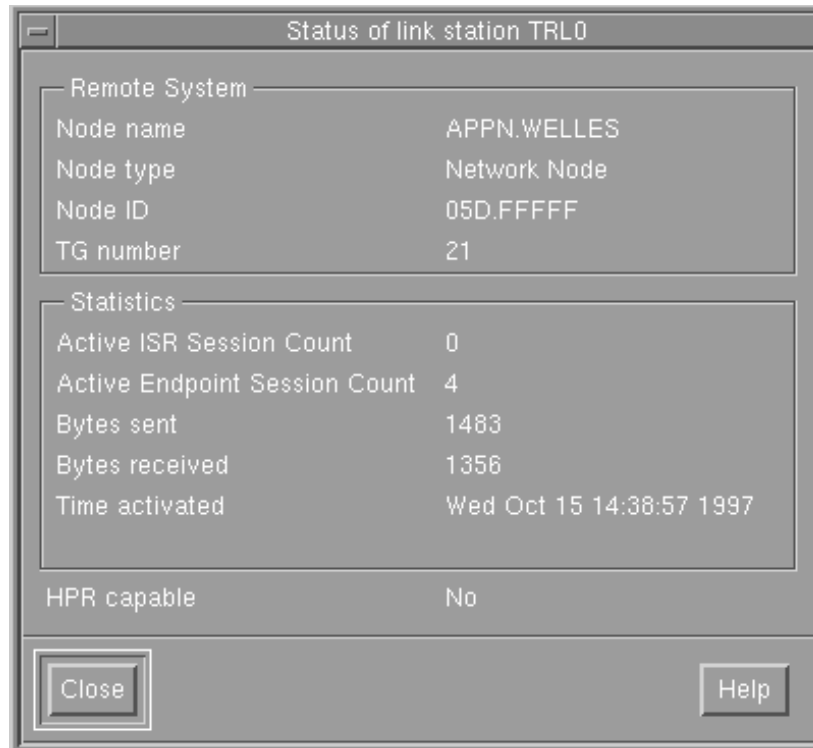


그림 17. 샘플 상태 대화상자

상태 대화상자는 자원의 현재 상태에 대한 정보를 표시합니다. 정보는 사용자가 보는 대로 동적으로 갱신됩니다.

도움말 창

Motif 관리 프로그램에 대한 온라인 도움말은 수행해야 하는 각 구성 task에 대한 자세한 지침을 제공합니다. 특히 task 시트는 특정 자원을 구성할 때 수행해야 하는 각 단계를 안내할 수 있습니다. 노드 매개변수를 구성하기 위한 task 시트(Linux용 Communications Server 구성에서 항상 첫 번째 단계)는 58 페이지의 그림 18에 표시

됩니다.

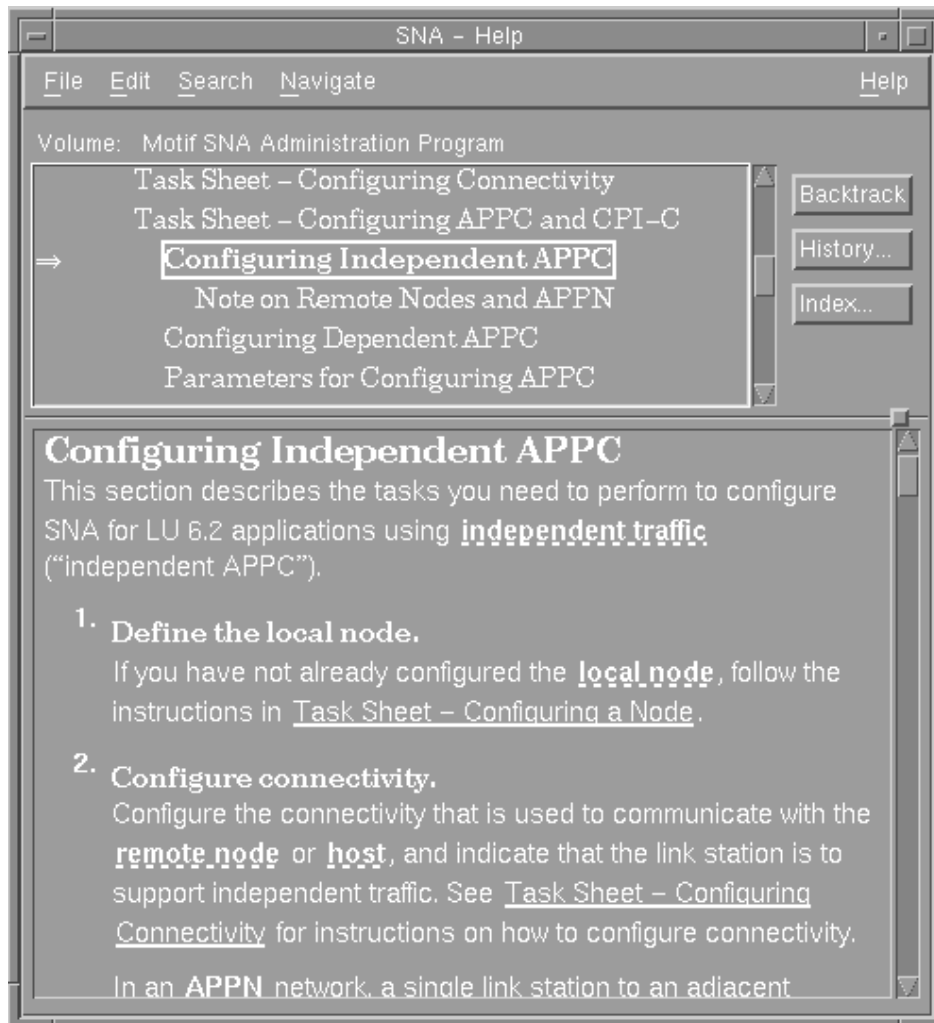


그림 18. 샘플 도움말 창

오류 메시지와 SNA 개념을 위해 각각의 창과 대화상자에 추가 도움말 창이 포함됩니다.

명령행 관리 프로그램 사용

명령행 구성을 사용하여 모든 Linux용 Communications Server 구성 매개변수를 변경할 수 있습니다. 이 구성을 사용하여 Motif 관리 프로그램을 통해 사용 가능한 자원을 구성하고, Motif 프로그램에 표시되지 않은 구성 매개변수를 설정 또는 변경할 수 있습니다. 그러나 이 관리 방법에서는 보통 Motif 관리에 필요한 것보다 많은 정보를 제공해야 합니다. 또한 제공하는 정보가 유효하고 기존 자원 정의와 일관성이 있는지 확인해야 합니다. (Motif 관리 프로그램은 사용자가 입력하는 데이터의 일관성을 보장하므로 권장되는 관리 프로그램입니다. 또한 메뉴 및 대화상자 선택사항에 따라 많은 구성 값을 추측하고 사용 가능한 정의를 기초로 값을 채울 수 있습니다.)

대부분의 관리 명령은 **snaadmin** 명령행 관리 프로그램에서 사용됩니다. 다음 양식으로 **snaadmin** 명령을 실행할 수 있습니다.

```
snaadmin command, parameter1=value1, parameter2=value2, ....
           {subrecord_name1}, sub_param1=sub_value1,
           sub_param2=sub_value2...
```

다음 명령을 사용하여 **snaadmin** 명령행 관리에 대한 도움말을 볼 수 있습니다.

- **snaadmin -h**: 명령행 관리에 대한 기본적인 도움말 및 명령행 도움말에 대한 사용법 정보를 제공합니다.
- **snaadmin -h -d**: **snaadmin** 프로그램에 제공할 수 있는 명령 리스트를 제공합니다.
- **snaadmin -h *command*** provides help for the named *command*.
- **snaadmin -h -d** 명령: 명령과 함께 지정할 수 있는 구성 매개변수 리스트를 포함하여, 이름 지정된 명령에 대한 자세한 도움말을 제공합니다.

명령에 서버 이름을 지정하기 위한 **-n** 옵션이 포함된 경우 일부 명령은 IBM 원격 API 클라이언트에서 실행할 수 있습니다. 그러한 명령의 결과는 이름 지정된 서버에서 실행한 경우와 같습니다.

이 절의 나머지 부분에서는 다른 유형의 자원에 대한 관리 명령을 요약합니다. 나열된 일부 명령 유형은 다음과 같습니다.

status_*

자원 유형에 대한 요약 정보를 제공합니다.

define_*

구성 파일에서 새 **define_*** 레코드를 작성하거나 동일 자원의 기존 레코드를 새 정의로 바꿉니다.

delete_*

파일에서 해당되는 **define_*** 레코드를 제거합니다.

query_*

해당 구성요소의 구성 파일에서 정보를 리턴하지만 파일을 수정하지는 않습니다.

set_* 추적 및 로깅 매개변수 등과 같은 관리 기능을 제어합니다.

명령행 구성에 대한 자세한 정보는 *Linux용 IBM Communications Server 관리 명령 참조서*를 참조하십시오.

제 3 장 기본 구성 TASK

이 장은 구성 TASK 개요를 제공하고 Linux용 Communications Server 노드를 구성하는 방법을 설명합니다. 또한 클라이언트/서버 환경에서 Linux용 Communications Server를 사용할 때 마스터 및 백업 서버를 구성하는 방법에 대해서도 설명합니다.

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.

포트 및 LU와 같은 많은 자원이 개별 노드에 구성됩니다. 이러한 자원을 『노드 자원』이라고 합니다.

다른 자원들은 모든 노드에 공통입니다. 해당 자원에 대한 하나의 정의만 전체 도메인에 대해 유지됩니다. 그러한 자원을 『도메인 자원』이라고 합니다. 도메인 자원 정의는 도메인의 마스터 서버에만 저장되므로 도메인의 모든 노드에서 액세스할 수 있습니다.

주: 독립형 Linux용 Communications Server 시스템에는 단 하나의 서버가 있으며, 그 서버가 항상 마스터로 작동합니다.

클라이언트/서버 환경에서 서버를 구성 서버로 표시할 수 있습니다. Linux용 Communications Server는 이 구성 서버 리스트를 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 마스터 서버를 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. 이는 구성 서버 리스트에 있는 모든 서버에 접속할 수 없을 경우에 발생합니다. 이러한 상황이 발생하면, 접속할 수 있는 서버에서만 노드 자원을 보고 구성할 수 있습니다.

주: 마스터 서버로 작동하는 노드를 직접 표시할 수는 없습니다. 마스터 서버는 구성 서버 리스트에 노드가 추가되는 순서로 선택됩니다. 서버를 리스트 맨 위로 이동하려면 리스트에서 다른 모든 노드를 제거한 후 다시 추가하십시오.

다음 관리 명령을 사용하여 구성 서버를 조회, 추가 및 삭제할 수도 있습니다.

query_sna_net

파일에 있는 서버를 나열합니다.

add_backup

리스트 끝에 새 서버를 추가합니다.

delete_backup

리스트에서 서버를 제거합니다. **delete_backup** 명령을 사용하여 마스터 서버를 삭제하거나(나열된 두 번째 서버가 마스터 서버가 되도록) 백업 서버를 삭제할 수 있습니다(더 이상 마스터로 작동하지 않도록).

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.

143 페이지의 제 10 장 『Linux용 Communications Server 클라이언트/서버 시스템 관리』는 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 노드가 APPN 네트워크에 참여하기 위해 필요한 기본적인 정보를 제공합니다. 노드를 먼저 구성한 후에 노드의 연결성과 기타 자원을 정의할 수 있습니다.

노드가 이미 구성되어 있으면, 노드 구성을 변경하기 전에 노드를 중지해야 합니다.

노드를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 노드 매개변수 구성을 선택하십시오.

명령행 관리 프로그램

define_node 명령을 실행하십시오.

노드 구성의 고급 매개변수는 정의되지 않아서 보안 실패가 보고되는 상대방 LU와 제한된 자원 시간종료가 있는 세션에 대한 제어를 제공합니다.

노드 구성 매개변수

노드 구성을 위해 다음 정보가 필요합니다.

APPN 지원

노드에 대한 APPN 지원 레벨:

- 네트워크가 APPN 네트워크가 아니면 노드를 LEN 노드로 구성하십시오.
- 다른 노드가 세션 경로지정 서비스를 제공하는 APPN 네트워크에 참여하거나 로컬 노드에서만 DLUR을 사용하려면 노드를 끝 노드로 구성하십시오.
- APPN 네트워크에서 중간 경로지정 서비스를 제공하거나 다운스트림 노드에 패스-스루 DLUR 서비스를 제공하려면 노드를 네트워크 노드로 구성하십시오.
- 브랜치 네트워크에서 기본 APPN 백본(backbone) 네트워크의 일부가 아닌 다른 노드에 세션 경로지정 서비스를 제공하려면 노드를 브랜치 네트워크 노드로 구성하십시오.

제어점 이름

로컬 노드의 전체 제어점(CP) 이름입니다. 이 이름은 네트워크에 있는 다른 노드에서 구성해야 할 수도 있으므로 SNA 네트워크 계획자에게 문의하여 이름을 판별하십시오.

제어점(CP)을 정의할 경우, Linux용 Communications Server는 자동으로 동일 이름을 사용하여 로컬 LU를 정의합니다. 그 LU는 노드의 디폴트 로컬 LU로 작동합니다.

제어점 별명

디폴트 로컬 LU의 로컬 별명입니다. 독립 LU 6.2 LU에서 기본 로컬 LU를 사용할 경우 이 값을 제공하십시오.

노드 ID

로컬 노드에서 PU의 식별자. 디폴트 (제어점) LU를 사용하는 종속 트래픽에 노드를 사용할 경우에만 값을 제공하십시오.

추가 구성

노드를 구성하고 나면 다음 구성 task로 계속하십시오.

- 67 페이지의 제 4 장 『연결 구성요소 정의』에 설명된 대로 연결성을 구성하십시오.
- 91 페이지의 제 6 장 『APPC 통신 구성』 또는 117 페이지의 제 7 장 『사용자 응용프로그램 구성』에 설명된 대로 노드 자원(LU)을 구성하십시오.
- 117 페이지의 제 7 장 『사용자 응용프로그램 구성』에 설명된 대로 응용프로그램을 구성하십시오.

로깅 구성

Linux용 Communications Server는 비정상 이벤트(선택적으로 정상 이벤트도)를 설명하는 로그 메시지를 로그 파일에 기록합니다. 문제점을 진단할 때 확인할 첫 번째 대상은 로그 파일입니다. 로그 메시지는 문제점의 원인과 취해야 할 조치에 대한 정보를 제공하기 때문입니다.

Linux용 Communications Server는 다음 이벤트 범주에 대한 메시지를 로그에 기록합니다.

문제점 사용자가 인식할 수 있을 만큼 시스템 성능을 저하시키는 비정상 이벤트(예: 세션의 비정상 종료)

예외 시스템 성능을 저하시키지만 사용자가 즉시 인식할 수 없는 비정상 이벤트(예: 자원 부족)나, 시스템 성능을 저하시키지는 않지만 나중에 발생하는 예외 또는 문제점의 원인을 표시할 수 있는 이벤트(예: 원격 시스템에서 예기치 않은 메시지 수신)

감사 정상 이벤트(예: 세션 시작)

Linux용 Communications Server는 또한 Linux용 Communications Server 자원의 현재 및 최대 사용에 대한 정보를 기록하기 위해 사용되는 사용 로그 파일도 유지합니다.

정상 및 오류 조건에 관련되는 로그들을 구별하기 위해 서로 다른 메시지 범주는 다른 파일에 기록됩니다. 문제점 및 예외 메시지는 오류 로그 파일에 기록되고, 감사 메시지는 감사 로그 파일에 기록됩니다.

특정 이벤트가 자주 발생되어 해당 로그 파일이 동일한 로그 메시지의 많은 인스턴스를 필터링하는 경우, 하나 이상의 특정 로그 메시지가 한 번만 로그되도록 필터를 설정할 수 있습니다. 그러면 동일한 로그 메시지의 모든 후속 인스턴스가 무시되어 로그 파일에 기록되지 않습니다. 이 필터링은 모든 유형의 로그(감사, 예외 및 문제점 로그)에 적용됩니다. 로깅 필터링에 대한 자세한 정보는 *Linux용 IBM Communications Server 진단 안내서*를 참조하십시오.

또한 Linux용 Communications Server는 로그 파일이 너무 커져서 디스크 자원을 소모하는 것을 방지하기 위해 백업 메커니즘을 제공합니다. 로그 파일이 허용되는 최대 크기에 도달하면, Linux용 Communications Server는 백업 파일에 현재 내용을 복사하고 로그 파일을 지웁니다.

디폴트로 Linux용 Communications Server는 다음 로그 파일을 사용합니다.

오류 로그 파일

`/var/opt/ibm/sna/sna.err`

`/var/opt/ibm/sna/bak.err`(백업)

감사 로그 파일

`/var/opt/ibm/sna/sna.aud`

`/var/opt/ibm/sna/bak.aud`(백업)

사용 로그 파일

`/var/opt/ibm/sna/sna.usage`

`/var/opt/ibm/sna/bak.usage`(백업)

텍스트 편집기나 다른 Linux 시스템 유틸리티를 사용하여 로그 파일을 볼 수 있습니다.

vi 텍스트 편집기에서 파일을 봅니다. 이 유틸리티를 사용하면 파일에서 앞뒤로 이동하고 특정 항목을 검색할 수 있습니다.

pg 파일을 한 번에 한 페이지씩 봅니다. 이 유틸리티는 사용하가 쉽고 간단하지만 로그 파일이 작을 경우에만 유용합니다.

tail 파일의 끝을 봅니다. 파일 끝은 최신 로그 메시지가 있는 곳입니다. 시스템을 실행하는 동안 로그 파일을 모니터링하려면 **-f** 옵션과 함께 이 유틸리티를 사용하십시오.

상세 로깅보다 간단한 로깅을 선택한 경우, **snahelp** 명령을 사용하여 특정 메시지 번호에 대한 원인 및 조치 정보를 판별할 수 있습니다.

대부분의 목적에 대해 디폴트 로깅 설정으로 충분하지만, 다음 유형의 변경을 수행할 수 있습니다.

- 로그에 기록할 메시지의 범주를 표시합니다.

문제점 메시지는 항상 로그에 기록되므로 사용 불가능하게 할 수 없습니다. 로깅은 다른 두 가지의 메시지 범주에 대해 보통 사용 불가능하지만 필요에 따라 사용 가능하게 할 수 있습니다.

- 로깅 메시지에 세부사항 레벨을 지정합니다.
- 도메인에 대해 중앙 로깅을 지정하거나 노드마다 로컬 로깅을 지정합니다.
- 로그 파일 이름 및 크기를 변경합니다.

로깅을 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창이나 도메인 창의 진단 메뉴에서 로깅을 선택하십시오.

명령행 관리 프로그램

다음 명령 중 하나를 실행하십시오.

- **set_central_logging**
- **set_global_log_type**

- **set_log_type**
- **set_log_file**

Motif 관리 프로그램의 로깅 대화상자는 도메인 전반에 걸쳐 로깅 설정에 영향을 줍니다. 명령행을 사용하면 특정 시스템에서 로컬 로그 설정을 구성하여 도메인 설정을 대체할 수 있습니다.

로깅에 대한 제어를 제공하는 것 외에, Motif 관리 프로그램은 추적에 대해서도 노트 레벨 제어를 제공합니다. 명령행 인터페이스는 로깅 및 추적 기능에 대해 더 큰 제어를 제공합니다. 로깅 및 추적에 대한 자세한 정보는 *Linux용 IBM Communications Server* 진단 안내서를 참조하십시오.

제 4 장 연결 구성요소 정의

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

컴퓨터에 하나 이상의 링크 프로토콜용 어댑터 카드를 설치할 수 있습니다. 연결성을 구성하기 위해 입력해야 하는 정보 중 많은 부분이 사용자가 사용 중인 링크 프로토콜에 따라 다릅니다. 원격 노드는 또한 사용자가 선택하는 것과 같은 유형의 어댑터 카드를 가지고 있어야 합니다. 그렇지 않으면 로컬 및 원격 노드 사이에 브릿지 또는 라우터가 있어야 합니다. Linux용 Communications Server가 지원하는 링크 프로토콜의 리스트는 68 페이지의 『DLC, 포트 및 연결 네트워크 정의』의 내용을 참조하십시오.

링크를 구성하려면, 68 페이지의 『DLC, 포트 및 연결 네트워크 정의』에 설명된 대로 포트를 정의해야 합니다. 또한 (대부분의 경우) 73 페이지의 『링크 스테이션 정의』에 설명된 대로 링크 스테이션을 구성해야 합니다. 로컬 노드의 LU가 DLUR를 사용하여 호스트와 통신할 경우, 82 페이지의 『DLUR PU 정의』에 설명된 대로 로컬 노드에서도 DLUR PU를 정의해야 합니다.

Motif 관리 프로그램을 사용할 경우에는 데이터 링크 제어(DLC)가 포트 구성의 일부 분으로 자동 구성됩니다. 또한 포트를 연결 네트워크의 일부로 정의하는 옵션을 사용할 수 있습니다. 명령행 구성을 사용할 때 이 구성은 포트 구성과 별개의 구성입니다.

링크 구성에 필요한 정보는 링크 프로토콜, 네트워크가 APPN 네트워크인지 여부, 그리고 링크가 종속 트래픽, 독립 트래픽, 또는 둘 다에 해당되는지 여부에 따라 다릅니다. 또한 구성해야 하는 링크는 지원해야 하는 통신 종류에 따라 다릅니다.

LUA LUA를 사용할 예정이면 호스트 컴퓨터에 대한 링크를 구성해야 합니다. 링크는 종속 트래픽에 구성해야 하며, 호스트 컴퓨터와 Linux용 Communications Server 노드에도 구성해야 하므로 SNA 네트워크 계획자에게 문의하십시오.

CPI-C 또는 APPC 사용

CPI-C 또는 APPC를 사용할 예정인데 네트워크가 APPN 네트워크가 아닌 경우, 액세스하려는 모든 인접 노드에 대해 링크를 구성해야 합니다. 이 링크는 독립 트래픽에 구성해야 하며, 인접 노드와 Linux용 Communications Server 노드에도 구성해야 하므로 SNA 네트워크 계획자에게 문의해야 할 수도 있습니다.

APPN 노드로 작동

Linux용 Communications Server 노드가 APPN 네트워크에서 끝 노드나 네트워크 노드인 경우, 구성해야 하는 링크 수가 상당히 감소될 수 있습니다. 하

나 이상의 인접 네트워크 노드에 대해 링크를 구성하고 이 링크를 사용하여 APPN 네트워크에 있는 모든 노드에 액세스할 수 있습니다. 다른 인접 노드에 직접 액세스하려면 그 노드들에 대한 링크도 구성할 수 있습니다. (이는 보통 필요하지 않지만 더 나은 성능을 제공할 수 있습니다.) 인접 노드가 Enterprise Extender를 사용하여 IP 네트워크 또는 LAN 세그먼트에 의해 연결된 경우, 직접 링크를 동적으로 설정할 수 있으므로 구성하지 않아도 됩니다. (포트를 정의할 때 네트워크를 연결 네트워크로 구성하도록 하십시오.)

APPN 네트워킹의 이점을 항상 독립 APPC에서 활용할 수 있지만 DLUR을 사용하지 않을 경우 LUA에는 적용되지 않습니다(DLUR은 호스트 및 APPN 네트워크에서 다운스트림 노드나 로컬 노드의 종속 LU 사이의 통신을 지원함). 호스트가 DLUS를 지원할 경우에만 DLUR을 사용할 수 있으므로, DLUR 사용에 관심이 있으면 SNA 네트워크 계획자에게 문의하십시오.

DLC, 포트 및 연결 네트워크 정의

포트는 네트워크에서 통신 링크의 로컬 끝을 고유한 액세스점으로 표시합니다. 각 포트는 다음과 같은 특정 링크 프로토콜과 연관됩니다.

- SDLC
- 토큰링
- 이더넷
- X.25 또는 정식 논리적 링크 제어(QLLC)
- 다중 경로 채널(MPC)(System z의 Linux용 Communications Server 전용)
- Enterprise Extender(HPR/IP)

특정 링크 프로토콜을 사용하는 하나 이상의 포트를 구성할 수 있습니다. 일반적으로, 포트는 어댑터 카드와 같은 실제 단일 액세스점에 해당하지만 일부 링크 프로토콜(예: 토큰링)에서는 단일 어댑터에 여러 포트를 정의할 수 있습니다. 각 포트는 주소(예: SAP 번호)로 구별됩니다.

Motif 관리 프로그램을 사용하여 특정 링크 프로토콜의 포트를 정의할 때, Linux용 Communications Server는 해당 유형의 DLC가 아직 정의되지 않은 경우 자동으로 포트의 DLC를 정의합니다. 명령행 구성의 경우에는 서로 다른 명령을 사용하여 포트와 DLC를 정의해야 합니다.

토큰링 링크 프로토콜을 사용하는 APPN 네트워크에서는 SAP 구성 대화상자를 사용하여 포트가 연결 네트워크의 일부임을 표시할 수도 있습니다.

SNA 게이트웨이를 사용할 경우, 암시적 링크 스테이션(명시적으로 구성되지 않은 링크 스테이션)에 대한 정의를 생성하기 위해 사용되는 템플릿을 정의할 수 있습니다. 암

시적 링크 스테이션은 다운스트림 LU를 지원할 수 있습니다. 포트가 활동 중일 때 암시적 PU 필드를 수정할 경우, 변경사항은 변경 후 생성되는 암시적 링크 스테이션 인스턴스에 적용됩니다.

포트, 연결 네트워크 및 DLC를 구성하려면, 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 연결성과 새 포트를 선택하십시오.

명령행 관리 프로그램

DLC 구성:

define_type_dlc

포트 구성:

define_type_port

이 명령에서 *type*은 링크 프로토콜 유형을 표시합니다(sd1c, tr, ethernet, ql1c, mpc, ip).

연결 네트워크 구성:

define_cn

고급 포트 구성 매개변수는 BTU 크기, 허용되는 활동 링크 수, 암시적 다운스트림 LU의 생성 및 동적 링크 스테이션 설정에 대한 제어를 제공합니다.

DLC, 연결 네트워크 및 포트 구성 매개변수

다음 매개변수는 포트 구성에 필요합니다. (Motif 관리 프로그램을 사용할 때, 포트 구성은 또한 DLC에 대한 정보를 제공하고 연결 네트워크에 포트를 지정할 수 있게 합니다.)

SNA 포트 이름

로컬로 알려진 포트 이름

어댑터 카드 번호

이 필드는 Enterprise Extender 포트에 사용되지 않습니다.

컴퓨터에 동일 유형의 카드가 여러 개 있을 경우 사용할 어댑터 카드를 식별하는 번호.

포트 번호

이 필드는 Enterprise Extender 포트에 사용되지 않습니다.

어댑터 카드가 여러 개의 포트를 지원할 수 있을 경우 사용할 포트의 번호. 유효한 포트 번호의 범위는 0에서 어댑터 카드가 지원하는 포트 수에 1을 뺀 숫자까지입니다. 어댑터 카드의 첫 번째 포트에 0을 입력하십시오.

DLC, 포트 및 연결 네트워크 정의

이 필드는 어댑터 카드가 여러 포트를 지원할 수 있을 경우에만 적용됩니다.

초기에 활동 중

노드가 시작될 때 포트를 자동으로 활성화할 것인지 여부를 표시합니다. 이 설정은 포트를 사용하는 링크 스테이션이 인접 노드의 요청에 대한 응답으로, 또는 로컬 노드의 요구로 활성화되도록 합니다. (포트를 활성화해도 링크 스테이션은 활성화되지 않습니다. 링크 스테이션은 별도로 활성화됩니다.)

다음 절에서는 링크 유형에 특정한 추가 포트 매개변수를 설명합니다. QLLC에 추가 포트 매개변수는 필요 없습니다.

SDLC의 추가 포트 매개변수

회선 세부사항

다음 매개변수는 SDLC 연결 유형을 설명합니다.

유형 다음 값 중 하나를 선택하십시오.

전용 회선

전용 회선은 이 컴퓨터와 원격 시스템 사이의 SDLC 링크에 사용됩니다.

교환식 입력

표준 전화 네트워크는 입력 호출에 사용됩니다.

1차가 아닌 포트의 경우(링크 역할 필드에 표시됨) 주소가 링크 스테이션에 구성된 전송 호출에 대해 폴 주소도 구성해야 합니다. 폴 주소는 원격 링크 스테이션에 구성된 폴 주소와 일치해야 하는 1바이트 주소(디폴트: C1)입니다. 활성화되면 포트는 이 폴 주소로 전송된 프레임에 응답합니다.

1차 포트의 경우 폴 주소를 구성하지 않아도 됩니다. 포트는 입력 호출에서 원격 링크 스테이션에서 지정한 폴 주소를 사용합니다. 다른 유형의 포트인 경우 각 링크 스테이션에서 폴 주소가 구성됩니다.

교환식 전송

표준 전화 네트워크는 전송 호출에 사용됩니다.

링크 역할

이 포트에 정의된 링크 스테이션에 대해 로컬 노드의 역할을 설명하는 값을 선택하십시오. SDLC 통신에서 한쪽 끝은 링크를 관리하고 1차 링크 스테이션이라고 합니다. 다른 한쪽 끝은 2차 링크 스테이션입니다.

이 필드에 대해 다음 값 중 하나를 사용하십시오.

2차 링크의 다른 끝은 제어가 되고 원격 시스템이 1차로 구성됩니다. 호스트 시스템에 링크를 구성할 경우 거의 항상 이 경우가 됩니다.

1차 이 포트는 링크의 SDLC 제어기 역할을 하고 원격 시스템은 2차로 구성됩니다.

조정 가능

최대 유연성을 위해 이 설정은 끝이 1차 역할을 수행하는 양쪽 끝이 조정 가능하게 합니다. 원격 시스템에 구성된 역할을 모르면 이 값을 선택하십시오.

피어 링크에 이 값을 사용할 수 있지만 역할을 조정하면 링크가 활성화될 경우 약간의 지연이 발생합니다.

1차 분산 분기 회선

링크는 전용이고 이 포트는 여러 2차 노드에 분산 분기 회선 링크의 제어기 역할을 합니다.

로컬 노드에서 다른 원격 노드에 여러 링크 스테이션을 구성할 경우 이 설정을 사용하십시오(예: 다운스트림 노드에 대한 링크). 이러한 각 노드는 2차로 구성되어야 하고 전용 회선을 사용하여야 합니다.

2차 다중 PU

로컬 포트는 원격 시스템의 포트에서 제어하는 분산 분기 회선 링크에서 2차 스테이션 중 하나입니다.

이러한 매개변수의 구성 방법을 모르면 SNA 네트워크 계획자에게 문의하십시오.

토큰링 및 이더넷에 대한 추가 포트 매개변수

로컬 SAP 번호

SAP의 주소, 보통 Intel 및 OSA2의 경우 04입니다. 카드에서 둘 이상의 SAP를 사용해야 할 경우에만 다른 값을 사용하십시오. OSA-Express 어댑터의 경우, 로컬 SAP 번호는 이 Linux 이미지에서 ethX 인터페이스에 대응하는 입출력 장치 주소의 OSA/SF에 정의된 번호와 일치해야 합니다.

SAP 번호는 4의 배수이어야 합니다.

이 필드에 대해 입력할 값을 모르면 SNA 네트워크 계획자에게 문의하십시오.

연결 네트워크에서 정의

SAP가 연결 네트워크로 LAN에 액세스할 것인지 여부를 표시합니다. 연결 네트워크를 정의하면 이전의 구성이 없어도 연결 네트워크에서 노드들 사이의 링크를 시작할 수 있습니다.

DLC, 포트 및 연결 네트워크 정의

LEN 노드는 연결 네트워크를 사용할 수 없으므로, 이 필드는 로컬 노드가 LEN 노드가 아닌 경우에만 적용됩니다.

CN 이름

연결 네트워크의 이름입니다. 연결 네트워크에서 SAP를 정의하기 위해 연결 네트워크에서 정의 옵션을 지정하지 않았으면 CN 이름을 입력하지 않아도 됩니다. CN 이름은 연결 네트워크에서 노드 사이에 연결을 설정하기 위해 가상 경로 지정 노드의 이름으로 사용됩니다.

연결 네트워크의 모든 노드에 같은 CN 이름을 지정하십시오.

이더넷 유형

이 필드는 이더넷 링크에만 적용됩니다.

네트워크가 표준 이더넷 네트워크인지 아니면 IEEE 802.3 네트워크인지 여부를 표시합니다.

Enterprise Extender(HPR/IP)의 추가 포트 매개변수

로컬 IP 인터페이스

이 필드는 선택적입니다. 이 필드를 사용하면 다중 IP 네트워크에 액세스하는 경우 로컬 IP 네트워크 인터페이스가 해당 포트에 사용되도록 지정할 수 있습니다. 하나의 IP 네트워크에만 액세스하는 경우에는 이 필드를 비워둘 수 있습니다.

인터페이스를 지정해야 하는 경우 다음 중 하나를 사용할 수 있습니다.

- 인터페이스 식별자(예: eth0 또는 en0)
- IPv4 점으로 구분된 10진수 주소(예: 193.1.11.100)
- IPv6 콜론으로 구분된 16진수 주소(예:
2001:0db8:0000:0000:0000:0000:1428:57ab 또는
2001:db8::1428:57ab)

인터페이스 식별자를 판별하려면 카드가 설치된 서버에서 **ipconfig -a** 명령을 실행하십시오. 그러면 인터페이스 식별자 및 연결된 IP 주소가 나열됩니다.

프로토콜

이 포트의 링크가 IPv4 또는 IPv6 주소를 사용하는지 여부를 표시합니다.

연결 네트워크에서 정의

어댑터가 연결 네트워크로 LAN에 액세스할 것인지 여부를 표시합니다. 연결 네트워크를 정의하면 이전의 구성이 없어도 연결 네트워크에서 노드들 사이의 링크를 시작할 수 있습니다.

LEN 노드는 연결 네트워크를 사용할 수 없으므로, 이 필드는 로컬 노드가 LEN 노드가 아닌 경우에만 적용됩니다.

CN 이름

연결 네트워크의 이름입니다. 연결 네트워크에서 포트를 정의하기 위해 연결 네트워크에서 정의 옵션을 지정하지 않았으면 CN 이름을 입력하지 않아도 됩니다. CN 이름은 연결 네트워크에서 노드 사이에 연결을 설정하기 위해 가상 경로지정 노드의 이름으로 사용됩니다.

연결 네트워크의 모든 노드에 같은 CN 이름을 지정하십시오.

암시적 링크의 추가 포트 매개변수

최대 활동 템플릿 인스턴스 수

템플릿에서 생성할 최대 링크 스테이션 인스턴스 수를 지정하십시오.

암시적 PU 액세스에 대한 다운스트림 LU 구성

이 PU를 사용하는 다운스트림 LU의 구성 여부를 표시합니다(129 페이지의 『SNA 게이트웨이 구성』 참조).

암시적 링크에서 지원되는 HPR

암시적 링크 스테이션에서 고성능 경로지정(HPR)의 지원 여부를 표시합니다.

암시적 링크에서 링크 레벨 오류 복구

링크 레벨 오류 복구를 사용하여 암시적 링크에서 HPR 트래픽을 송신할 것인지 여부를 표시합니다.

추가 구성

포트 구성을 수행하고 나면 다음 구성 태스크로 계속하십시오.

- 구성된 포트에 대해 링크 스테이션을 정의하려면 『링크 스테이션 정의』를 참조하십시오.
- DLUR PU를 정의하려면 82 페이지의 『DLUR PU 정의』를 참조하십시오.
- APPC 통신을 지원하려면 91 페이지의 제 6 장 『APPC 통신 구성』을 참조하십시오.

링크 스테이션 정의

SNA 네트워크에서 다른 노드와 통신하려면 SNA 네트워크의 인접 노드에 대해 링크 스테이션(LS) 특성을 구성해야 합니다. 링크 스테이션을 정의하려면 먼저 사용 중인 어댑터(및 링크 프로토콜)의 포트를 정의해야 합니다. 링크 스테이션을 구성하는 데 필요한 대부분의 정보는 사용하는 프로토콜에 관계없이 같습니다.

링크 스테이션은 Linux용 Communications Server 로컬 노드와 원격 컴퓨터 사이에 SNA 네트워크를 통해 논리 경로를 표시합니다. 원격 컴퓨터는 다음 중 하나가 될 수 있습니다.

링크 스테이션 정의

- Linux용 Communications Server가 3270 또는 LUA 통신을 사용하여 호스트 프로그램에 액세스하는(또는 프로그램간 통신에 APPC 또는 CPI-C를 사용하는) 호스트 컴퓨터
- 동등한 각각의 상대방으로서 통신하는 Linux용 Communications Server와 원격 컴퓨터가 있는 피어 컴퓨터(APPN 네트워크에서 일반적임)
- 호스트에 액세스하기 위해 Linux용 Communications Server SNA 게이트웨이 기능이나 DLUR 기능을 사용하는 다운스트림 컴퓨터

링크 스테이션은 특정 포트와 연관됩니다. 포트마다 하나 이상의 링크 스테이션을 정의할 수 있습니다.

중속 트래픽을 지원하는 각 링크 스테이션에는 연관된 PU(물리 장치)가 있습니다. PU는 링크 스테이션과 연관되므로, Linux용 Communications Server는 PU를 별도의 자원으로 취급하지 않습니다. PU는 링크 스테이션 구성의 일부로 구성되며 링크 스테이션 시작 및 중지 일부로 시작 및 중지됩니다. 링크 스테이션은 노드 창의 연결성 섹션에 표시되지만 PU는 어떤 창에도 표시되지 않습니다.

주: 대부분의 경우 포트에 링크 스테이션을 추가해야 합니다. 그러나 링크가 항상 원격 노드에서 활성화되는 상황을 위해 다운스트림 SNA 게이트웨이나 APPC 트래픽에 대해서만 동적으로 작성된 링크 스테이션을 사용하려면 명시적으로 링크 스테이션을 구성하지 않아도 됩니다.

원격 노드가 로컬 노드에 연결하려고 하지만 입력 호출에 지정된 주소와 일치하는 링크 스테이션이 정의되지 않았을 경우, Linux용 Communications Server는 로컬 노드에 적절한 포트가 정의되어 있으면 암시적으로 링크 스테이션을 정의할 수 있습니다. 이렇게 동적으로 작성된 링크 스테이션은 연결 지속 기간 동안 노드 창의 연결성 섹션에 표시됩니다.

링크 스테이션을 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 연결성과 새 링크 스테이션을 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

define_type_ls

이 명령에서 *type*은 링크 프로토콜 유형을 표시합니다(sd1c, tr, ethernet, q11c, mpc, ip).

링크 스테이션의 고급 매개변수는 전송 특성, XID 교환, 선택적 링크 기능, 링크를 사용하는 LU 0-3 세션의 압축 및 재활성화 프로시저에 대한 추가 제어를 제공합니다.

링크 스테이션 구성 매개변수

Motif에서는 링크 스테이션 구성 대화상자에 다음과 같은 섹션이 포함됩니다. 각 섹션은 서로 다른 구성 매개변수 범주를 포함하고 있습니다.

링크 스테이션

종속 LU, 독립 LU 또는 둘 다에 대해 LU 트래픽을 지원하는지에 관계없이 모든 링크 스테이션에 필요한 정보를 제공하려면 대화상자의 이 영역을 사용하십시오. 이 섹션에 있는 매개변수의 설명은 『공통 링크 스테이션 매개변수』를 참조하십시오.

독립 LU 트래픽

독립 트래픽에 대해 링크 스테이션을 사용할 경우에만 이 정보를 제공하십시오. 이 섹션에 있는 매개변수의 설명은 78 페이지의 『독립 LU 트래픽 매개변수』를 참조하십시오.

종속 LU 트래픽

종속 트래픽에 대해 링크 스테이션을 사용할 경우에만 이 정보를 제공하십시오. 이 섹션에 있는 매개변수의 설명은 79 페이지의 『종속 LU 트래픽 매개변수』를 참조하십시오.

공통 링크 스테이션 매개변수

다음 매개변수는 종속 트래픽, 독립 트래픽 또는 둘 다의 지원 여부에 관계없이 모든 링크 스테이션에 필수입니다.

이 대화상자에 있는 매개변수에 대한 자세한 정보는 온라인 도움말이나 *Linux*용 *IBM Communications Server* 관리 명령 참조서를 참조하십시오.

이름 링크 스테이션을 로컬로 식별할 이름

SNA 포트 이름

인접 노드에 액세스하기 위해 사용할 포트

활성화 링크 스테이션을 활성화하기 위해 사용하는 방법. 다음 방법 중 하나를 지정하십시오.

관리자에 의해

링크 스테이션은 로컬 시스템 관리자의 요청에 대해서만 활성화됩니다.

노드 시동 시

링크 스테이션은 노드가 시동될 때 자동으로 시작됩니다.

요구가 있을 때

링크 스테이션은 응용프로그램에 대한 연결성을 제공해야 할 경우에 자동으로 시작됩니다.

링크 스테이션은 포트와 별도로 활성화되므로 포트가 이미 활동 중인 경우에도 링크 스테이션을 활성화해야 합니다. 포트를 활성화해도 그 자체가 링크 스테

이션을 활성화하지는 않으며, 포트가 처음부터 활성화되도록 구성한다고 해서 노드 시작 시 해당되는 링크 스테이션이 자동으로 활성화되는 것이 아닙니다. 그러나 포트를 활성화하면 링크 스테이션을 활성화할 수 있게 됩니다. 링크 스테이션은 로컬 노드와 인접 노드 둘 다에서 포트가 활동 중이 아닌 경우 활성화할 수 없습니다.

링크 사용 비용을 부담하는 경우, 비용을 줄이기 위해 불필요하게 링크를 활성화하지 않도록 하십시오.

이 필드를 설정하는 방법을 확실히 알지 못하면 SNA 네트워크 계획자에게 문의하십시오.

LU 트래픽

링크를 통해 흐르는 LU 트래픽 유형. 이 선택사항은 링크 정의에 필요한 다른 매개변수를 판별합니다.

이 링크 유형은 독립 트래픽만 지원하므로, 이 매개변수는 Enterprise Extender(HPR/IP) 링크에 사용되지 않습니다.

무관함

독립 및 종속 LU 트래픽 둘 다에 대해 링크 스테이션을 사용할 수 있습니다. 이 옵션의 경우, 이 절에 설명된 필드 외에 78 페이지의 『독립 LU 트래픽 매개변수』 및 79 페이지의 『종속 LU 트래픽 매개변수』에 설명된 필드에 대해서도 값을 제공해야 합니다.

독립형만

독립 LU 트래픽에 대해서만 링크 스테이션을 사용할 수 있습니다. 이 옵션의 경우, 이 절에 설명된 필드 외에 78 페이지의 『독립 LU 트래픽 매개변수』에 설명된 필드에 대해서도 값을 제공해야 합니다.

종속형만

종속 LU 트래픽에 대해서만 링크 스테이션을 사용할 수 있습니다. 이 옵션의 경우, 이 절에 설명된 필드 외에 79 페이지의 『종속 LU 트래픽 매개변수』에 설명된 필드에 대해서도 값을 제공해야 합니다.

인접 노드에 접속하기 위한 주소지정 정보도 제공해야 합니다. 필요한 주소지정 정보 유형은 포트의 DLC 유형에 따라 다릅니다. 원격 노드의 주소를 제공하지 않을 경우, 링크 스테이션은 비선택적 연결대기 링크 스테이션으로 작동하여 원격 노드로부터의 입력 호출을 허용합니다.

SDLC의 추가 링크 스테이션 매개변수:

폴 주소

원격 스테이션의 폴 주소입니다. 일반적으로 C1에서 시작하는 2자리(1바이트) 16진 값으로 주소를 지정하십시오. 1차 링크 스테이션은 이 값을 사용하여 원

격 스테이션을 폴링합니다. 2차 링크 스테이션은 이 값으로 폴링에 응답합니다. 폴 주소는 다음 링크 역할에 따라 다르게 입력됩니다.

- 링크가 지점간 접속 링크(분산 분기 회선 아님)면 일반적으로 주소 C1이 사용됩니다.
- 이 링크에 대한 상위 포트가 교환식 입력이면 폴 주소는 포트에서 구성되고 각 링크 스테이션에 대해 독립적으로 구성되지 않습니다.
- 1차 교환식 전송 링크 스테이션을 구성하고 통신에 사용할 원격 2차 폴 주소를 모를 경우 1차에 0xFF의 폴 주소를 지정할 수 있습니다. 이 값은 구성된 폴 주소에 관계없이 노드가 2차에서 응답을 승인할 수 있게 합니다. 0xFF는 1차가 아닌 링크 또는 교환식 전송이 아닌 링크에 대해 유효한 주소가 아닙니다.
- 분산 분기 회선 구성을 사용할 경우 동일한 1차와 통신하는 모든 2차 링크 스테이션은 다른 폴 주소를 가져야 합니다.

링크 양 끝의 폴 주소는 일치해야 합니다. 원격 시스템에 구성된 주소를 모르면 SNA 네트워크 계획자에게 문의하십시오.

VTAM 호스트에서 폴 주소는 VTAM PU 정의에서 ADDR= 매개변수로 구성됩니다.

AS/400 시스템에서 폴 주소는 회선 설명의 STNADR 매개변수입니다.

토큰링 및 이더넷에 대한 추가 링크 스테이션 매개변수:

MAC 주소

일련의 16진수로 입력된 원격 스테이션의 MAC 주소. MAC 주소는 원격 시스템에서 어댑터 카드를 고유하게 식별합니다.

사용할 값을 모르면 SNA 네트워크 계획자에게 문의하십시오.

이 링크의 원격 끝이 VTAM 호스트이면 VTAM 포트 정의의 MACADDR= 매개변수에서 해당되는 MAC 주소를 찾을 수 있습니다.

AS/400 시스템에 대한 링크를 구성할 경우, MAC 주소는 회선 설명에 있는 ADPTADR 매개변수입니다.

SAP 번호

원격 컴퓨터에서 포트의 SAP 번호. SAP 번호는 같은 어댑터 카드를 사용하는 여러 링크를 구별합니다. 이 번호는 16진수로, 보통 04이고 4의 배수여야 합니다.

사용할 값을 모르면 SNA 네트워크 계획자에게 문의하십시오.

이 링크의 원격 끝이 VTAM 호스트이면 SAP 번호는 VTAM PU 정의의 SAPADDR= 매개변수입니다.

AS/400 시스템에 대한 링크를 구성할 경우, SAP 번호는 회선 설명에 있는 SSAP 매개변수입니다.

X.25(QLLC)의 추가 링크 스테이션 매개변수:

원격 X.25 주소

링크가 교환식 가상 회선이면 원격 DTE의 DTE 주소를 일련의 16진수로 입력하십시오.

링크가 영구 가상 회선(PVC)이면 링크 스테이션이 사용할 가상 회선을 식별하는 채널 ID를 입력하십시오. 채널 ID는 1부터 최대 1024까지 지정할 수 있습니다. 하나의 영구 가상 회선만 있는 경우 해당 채널 ID는 1이 될 가능성이 높습니다.

MPC의 추가 링크 스테이션 매개변수:

MPC 그룹

특정 채널을 식별하도록 MPC 드라이버 구성에 지정된 다중 채널 경로(MPC) 그룹 이름입니다.

Enterprise Extender(HPR/IP)의 추가 링크 스테이션 매개변수:

원격 IP 호스트 이름

이 링크에 대한 목적지 노드의 원격 호스트 이름입니다. 호스트 이름은 다음 중 하나가 될 수 있습니다. 이 링크가 사용하는 포트의 *protocol* 매개변수는 해당 주소가 IPv4 또는 IPv6 형식이 되어야 할지를 판별합니다.

- IPv4 점으로 구분된 10진수 주소(예: 193.1.11.100)
- IPv6 콜론으로 구분된 16진수 주소(예:
2001:0db8:0000:0000:0000:0000:1428:57ab 또는
2001:db8::1428:57ab)
- 이름(예: newbox.this.co.uk)
- 별명(예: newbox)

이름이나 별명을 지정할 경우, Linux 시스템은 이를 전체 이름으로 해석할 수 있어야 합니다(로컬 TCP/IP 구성을 사용하거나 도메인 이름 서버를 사용하여).

독립 LU 트래픽 매개변수

링크 스테이션을 독립 LU(APPC, 5250 또는 CPI-C 응용프로그램에서 사용하기 위한 유형 6.2 LU)에서 사용하도록 구성하려면 다음 정보가 필요합니다.

원격 노드 이름

원격 노드의 전체 CP 이름입니다.

원격 시스템이 VTAM 호스트일 경우, VTAM 시작 리스트의 *NETID* 매개변수에서 네트워크 이름(전체 이름의 처음 8자)을 찾을 수 있습니다. 마지막 8자는 VTAM 시작 리스트의 *SSCPNAME* 매개변수에 있습니다.

주: 새 원격 노드의 이름을 입력할 경우, 원격 노드의 정의를 추가하여 새 원격 노드에서 상대방 LU를 정의할 수 있습니다. 로컬 노드가 LEN 노드이면 원격 노드 유형을 지정하지 않아도 됩니다. 원격 노드 유형 필드가 적용되지 않습니다.

이 방식으로 새 원격 노드를 정의하려면, 원격 노드 이름을 지정하고 원격 노드 유형을 검색이 아닌 다른 값으로 지정하십시오.

또는 검색을 지정하고 원격 노드 이름은 빈 상태로 둘 수 있습니다. 이는 인접 노드가 링크 스테이션을 사용할 수 있음을 의미합니다. 검색 옵션은 로컬 노드가 LEN 노드인 경우 사용할 수 없습니다.

원격 노드 유형

해당 링크 스테이션을 통해 액세스되는 원격 노드에서의 APPN 지원 레벨(로컬 노드가 끝 노드이거나 네트워크 노드인 경우에만 적용됨)

원격 노드가 LEN 노드 또는 끝 노드인지, 또는 네트워크 노드인지 모를 경우에는 검색을 선택하면 됩니다. 원격 노드에서의 APPN 지원 레벨을 검색하면 링크 활성화가 다소 지연될 수 있으므로 유형을 알면 직접 지정하는 것이 좋습니다. 이는 네트워크 구성의 일관성을 유지하는 데도 도움이 됩니다.

링크 스테이션이 요구가 있을 때 활성화될 경우에는 검색을 선택할 수 없습니다.

로컬 노드가 LEN 노드이면 이 필드는 적용되지 않습니다.

브랜치 링크 유형

해당 링크 스테이션을 통해 액세스되는 원격 노드에 대한 링크의 유형(로컬 노드가 브랜치 네트워크 노드인 경우에만 적용됨)

원격 노드가 기본 APPN 백본(backbone) 내의 네트워크 노드일 경우, 업링크(백본에 대해)를 선택하십시오. 원격 노드가 브랜치 내에서 끝 노드이면, 다운링크(브랜치 내에서)를 선택하십시오.

원격 노드가 네트워크 노드로 구성되면, 브랜치 링크 유형은 자동으로 업링크(백본(backbone)에 대해)로 설정되고 이를 변경할 수 없습니다.

종속 LU 트래픽 매개변수

이 링크 유형은 독립 트래픽만 지원하므로, 다음 매개변수들은 Enterprise Extender(HPR/IP) 링크에 적용되지 않습니다.

링크 스테이션 정의

종속 LU 트래픽에 대해 링크 스테이션을 구성하면 자동으로 링크 스테이션과 같은 이름의 적절한 PU가 작성됩니다.

링크 스테이션을 종속 LU(3270 또는 LUA 응용프로그램을 위한 유형 0-3 LU)에서 사용하도록 구성하려면 다음 정보가 필요합니다.

로컬 노드 ID

SNA 네트워크에서 로컬 노드를 식별하기 위한 값

보통 같은 노드에 있는 모든 링크에 대해 같은 노드 ID(디폴트 값)를 사용할 수 있습니다. 그러나 특정 호스트에 액세스하기 위해 256개 이상의 종속 LU가 필요할 경우, 호스트에 대해 복수의 링크 스테이션(각각 최대 25다섯 개의 LU가 있고 로컬 노드 ID가 다른)을 구성해야 합니다.

원격 노드가 로컬 노드 ID를 인식하도록 구성하려면, SNA 네트워크 계획자에게 문의하십시오.

VTAM 구성에서 처음 세 자리 숫자는 PU 정의의 *IDBLK* 매개변수와 일치해야 하고 마지막 다섯 자리는 *IDNUM* 매개변수와 일치해야 합니다.

AS/400 시스템에서 노드 ID는 *EXCHID* 매개변수로 구성됩니다.

원격 노드 ID

원격 링크 스테이션의 노드 ID(선택적. 해당 링크 스테이션으로 액세스를 제한할 경우에만 적용됨). 원격 노드 ID를 지정할 경우, 링크는 원격 노드의 노드 ID가 이 정의에 지정된 값과 일치할 경우에만 활성화됩니다. 이는 교환식 포트에 몇 개의 링크 스테이션이 구성된 경우에 유용할 수 있습니다. 원격 노드에서 링크 스테이션이 활성화될 때 각 링크 스테이션을 구별할 수 있기 때문입니다. 링크 스테이션은 원격 노드의 CP 이름으로 구별할 수도 있지만 링크를 활성화할 때 해당 CP 이름을 송신하지 않는 원격 노드의 경우에는 대신 원격 노드 ID를 사용해야 합니다.

원격 노드 ID를 지정하지 않을 경우, 링크가 활성화될 때 원격 노드의 노드 ID를 확인하지 않습니다.

원격 노드 역할

원격(인접) 노드의 역할:

호스트

링크 스테이션이 호스트 컴퓨터가 있는 세션에 사용되는 종속 LU(예: 3270 LU)를 지원합니다(대부분의 경우). 링크가 SNA 게이트웨이 또는 DLUR을 사용하는 호스트 연결을 제공하는 노드에 대한 링크일 경우, 링크가 직접 호스트 컴퓨터에 연결되지 않아도 인접 노드 역할을 호스트로 설정해야 합니다.

다운스트림(SNA 게이트웨이)

링크 스테이션이 로컬 노드의 SNA 게이트웨이 성능을 사용하는 호스

트와 통신하는 다운스트림 노드에 대한 것입니다. (호스트에는 다운스트림 노드의 LU가 로컬 노드에 상주하는 것으로 표시됩니다.)

다운스트림 (DLUR)

링크 스테이션이 로컬 노드의 DLUR 성능을 사용하는 호스트와 통신할 다운스트림 노드에 대한 것입니다. (호스트에는 다운스트림 노드의 LU가 로컬 노드에 상주하는 것으로 표시됩니다.)

그러한 링크는 로컬 노드가 APPN 네트워크 노드인 경우에만 사용할 수 있습니다.

다운스트림 PU 이름

다운스트림 노드와 연관되는 PU 이름입니다. 이 값은 호스트 컴퓨터에서 다운스트림 노드에 대해 구성된 PU 이름과 일치해야 합니다. 이 이름에 사용할 값을 모르면 SNA 네트워크 계획자에게 문의하십시오.

이 필드는 이 링크 스테이션이 로컬 노드의 DLUR 성능을 사용하는 호스트와 통신할 다운스트림 PU에 대한 것임을 지정한 경우에만 적용됩니다. 원격 노드 역할 필드에 대해 다운스트림(DLUR)을 지정하여 이를 표시할 수 있습니다.

자세한 정보는 82 페이지의 『DLUR PU 정의』를 참조하십시오.

업스트림 DLUS 이름

DLUS를 지원하는 호스트 LU((다운스트림 PU가 액세스할 LU 서버)의 전체 LU 이름입니다. 이 이름에 사용할 값을 모르면 SNA 네트워크 계획자에게 문의하십시오.

이 필드는 이 링크 스테이션이 로컬 노드의 DLUR 성능을 사용하는 호스트와 통신할 다운스트림 PU에 대한 것임을 지정한 경우에만 적용됩니다. 원격 노드 역할 필드에 대해 다운스트림(DLUR)을 지정하여 이를 표시할 수 있습니다.

추가 구성

링크 스테이션 구성을 수행하고 나면 다음 구성 태스크로 계속하십시오.

- DLUR PU를 정의하려면 82 페이지의 『DLUR PU 정의』를 참조하십시오.
- 패스-스루 서비스를 구성하려면 119 페이지의 제 8 장 『패스-스루 서비스 구성』을 참조하십시오
- 특정 사용자 응용프로그램을 지원하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』을 참조하십시오.
- APPC 통신을 지원하려면 91 페이지의 제 6 장 『APPC 통신 구성』을 참조하십시오.

DLUR PU 정의

보통, 종속 LU 세션은 호스트 컴퓨터로의 직접 통신 링크를 필요로 합니다. 많은 노드 (호스트 노드 포함)가 APPN 네트워크에서 함께 연결된 경우, 일부 노드가 호스트와의 직접 연결을 가지고 있지 않고 다른 노드를 통한 간접 연결만 가지고 있을 수 있습니다. 이렇게 간접적으로 연결된 노드에서는 LU에서 호스트로의 종속 LU 세션을 작성할 수 없습니다.

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

이 절은 호스트 컴퓨터와의 연결성을 제공하는 DLUR PU를 구성하는 방법에 대해 설명합니다. DLUR PU를 구성하면 로컬 노드가 DLUR 서비스를 제공할 수 있습니다.

APPN 노드(예: Linux용 Communications Server를 실행하는 노드)의 DLUR은 호스트에 있는 종속 LU 서버(DLUS)와 함께 작동하여, APPN 네트워크를 통해 DLUR 노드의 종속 LU에서 DLUS 호스트까지 세션 경로를 지정합니다. 호스트로의 전송 경로는 여러 노드에 분산될 있으며, APPN의 네트워크 관리, 동적 자원 위치 및 전송 경로 계산 기능을 이용할 수 있습니다. DLUR은 LU가 있는 노드에서 사용 가능해야 하며, DLUS는 호스트 노드에서 사용 가능해야 합니다. 그러나 세션 전송 경로에서 어떤 중계 노드에도 DLUR이 필요하지 않습니다.

Linux용 Communications Server DLUR 노드가 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.

패스-스루 DLUR 서비스를 다운스트림 노드에 제공하려면, 먼저 다운스트림 노드와 연관되는 PU 이름을 (로컬 노드)에 구성해야 합니다. 이 값은 호스트 컴퓨터에서 다운스트림 노드에 대해 구성된 PU 이름과 일치해야 합니다.

DLUR PU를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 연결성과 새 DLUR PU를 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

```
define_internal_pu
```

DLUR PU 구성 매개변수

다음 매개변수는 DLUR PU 구성에 필요합니다.

PU 이름

로컬 노드의 DLUR PU마다 PU 이름을 지정하십시오. 이 이름은 호스트에 구성된 PU 이름과 일치해야 합니다. (Linux용 Communications Server는 PU를 식별하기 위해 PU 이름과 PU ID 모두를 호스트에 송신합니다. 호스트는 보통 PU 이름으로 PU를 식별하지만 일치하는 PU 이름을 찾지 못하면 PU ID로 식별합니다.)

DLUS 이름

DLUS를 지원하는 호스트 LU의 전체 LU 이름입니다.

DLUR을 사용하려면 Linux용 Communications Server의 DLUR 구성요소가 호스트에서 DLUS와의 LU-LU 세션을 작성해야 합니다.

호스트 LU 이름을 판별하려면 SNA 네트워크 계획자에게 문의하십시오.

백업 DLUS 이름

이 매개변수는 선택적입니다. *DLUS* 이름에서 지정한 것이 사용 불가능할 경우 사용할 수 있는 백업 호스트 LU의 전체 LU 이름입니다.

호스트 LU 이름을 판별하려면 SNA 네트워크 계획자에게 문의하십시오.

PU ID

로컬 노드에서 호스트와의 연결을 지원하는 PU의 PU ID입니다. PU ID는 두 개의 16진 문자열로 구성됩니다. 하나는 세 자릿수(블록 번호)이고 다른 하나는 다섯 자릿수입니다.

각각의 종속 LU는 PU와 연관됩니다. PU 및 LU는 둘 다 호스트 컴퓨터에서 구성됩니다. PU마다 Linux용 Communications Server 노드에서 DLUR PU를 정의해야 합니다. PU ID는 이 PU에 대해 호스트에서 구성된 PU ID와 일치해야 합니다.

많은 경우에 PU ID는 노드 ID와 같으므로 노드 ID가 디폴트입니다. 그러나 특정 호스트에 액세스하기 위해 25다섯 개 이상의 종속 LU가 필요할 경우, 복수의 DLUR PU(각각 최대 25다섯 개의 종속 LU가 있고 PU ID가 다른)를 구성해야 합니다.

이 필드를 설정하는 방법을 확실히 알지 못하면 SNA 네트워크 계획자에게 문의하십시오.

VTAM 구성에서 처음 세 자리 숫자는 PU 정의의 *IDBLK* 매개변수와 일치해야 하고 마지막 다섯 자리는 *IDNUM* 설정과 일치해야 합니다.

초기에 활동 중

노드가 시작할 때 DLUR PU가 자동으로 활성화되는지 여부를 표시합니다. 이 옵션을 설정하지 않으면 DLUR PU는 수동으로 시작해야 합니다.

압축 지원

해당 PU를 사용하는 LU 0-3 세션에 대해 데이터 압축이 지원되는지 여부를

표시합니다. 이 옵션을 설정하면 호스트가 요청할 경우 압축이 사용됩니다. 이 옵션을 설정하지 않으면 압축은 사용되지 않습니다.

DLUS 접속 무한정 재시도

첫 번째 시도에서 실패할 경우 Linux용 Communications Server가 DLUS 접속을 재시도하는지 여부를 표시합니다. 이 옵션을 설정하면 Linux용 Communications Server는 첫 번째 시도가 실패할 경우 무한정 재시도합니다. 이 옵션을 설정하지 않으면 한 번만 시도합니다.

다운스트림 노드에 대한 패스-스루 DLUR 매개변수

다운스트림 노드의 종속 LU와 호스트 사이에 트래픽을 전송하기 위해 패스-스루 DLUR을 사용하도록 Linux용 Communications Server를 구성하려면 다음 정보가 필요합니다.

다운스트림 PU 이름

다운스트림 노드와 연관되는 PU 이름입니다. PU 이름은 호스트 컴퓨터에 구성된 PU 이름과 일치해야 합니다.

다운스트림 노드는 여러 개의 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.

다운스트림 노드와 연관되는 PU 이름을 찾으려면 SNA 네트워크 계획자에게 문의하십시오.

DLUS 이름

DLUS를 지원하는 호스트 LU의 전체 LU 이름입니다. DLUR을 사용하려면 Linux용 Communications Server의 DLUR 구성요소가 호스트에서 DLUS와의 LU-LU 세션을 작성해야 합니다.

호스트 DLUS 서버에 해당되는 LU 이름을 찾으려면 SNA 네트워크 계획자에게 문의하십시오.

추가 구성

DLUR을 구성하고 나면 다음 구성 태스크로 계속하십시오.

- DLUR에 대해 LU를 구성하려면 131 페이지의 『DLUR 구성』을 참조하십시오
- 다른 패스-스루 서비스를 구성하려면 119 페이지의 제 8 장 『패스-스루 서비스 구성』을 참조하십시오
- 특정 사용자 응용프로그램을 지원하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』을 참조하십시오.
- APPC 통신을 지원하려면 91 페이지의 제 6 장 『APPC 통신 구성』을 참조하십시오.

제 5 장 종속 LU 구성

이 장은 3270, TN3270 및 LUA 통신을 사용하는 사용자 응용프로그램을 지원하도록 LU 및 LU 풀을 구성하기 위한 지시사항을 제공합니다. 이 자원들을 사용하려면 종속 LU를 구성해야 합니다.

이 장에 설명된 자원을 구성하려면 먼저 다음 구성을 수행해야 합니다.

- 62 페이지의 『노드 구성』에 설명된 대로 노드를 구성하십시오.
- 67 페이지의 제 4 장 『연결 구성요소 정의』에 설명된 대로 연결성을 구성하십시오. 3270, TN3270 및 LUA에 대해, 종속 LU 트래픽을 지원하도록 링크를 구성해야 합니다.

업스트림 SNA 게이트웨이나 DLUR을 사용할 경우에는 호스트에 대한 직접 링크를 구성하지 않아도 됩니다. 자세한 정보는 129 페이지의 『SNA 게이트웨이 구성』 및 82 페이지의 『DLUR PU 정의』를 참조하십시오.

LU 유형 0-3 정의

호스트 시스템과의 통신을 지원하려면 유형 0-3의 종속 LU를 구성해야 합니다. 이 절에 있는 정보를 사용하여 3270 또는 LUA를 지원하도록 LU를 정의할 수 있습니다. LU 범위를 정의하여, 단일 조작으로 동일 유형의 여러 LU를 구성할 수도 있습니다.

LU 유형 0-3을 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 다음 중 하나를 선택하십시오.

- 3270과 새 3270 표시장치 LU 또는 새 3270 프린터 LU
- LUA와 새 LUA LU
- TN 서버와 새 호스트 LU

명령행 관리 프로그램

다음 명령 중 하나를 실행하십시오.

```
define_lu_0_to_3
```

```
define_lu_0_to_3_range
```

고급 대화상자를 사용하여 액세스를 특정 SSCP로 제한하거나 비활성 시간종료를 지정할 수 있습니다.

LU 유형 0-3 구성 매개변수

다음 매개변수는 LU 유형 0-3 구성에 필요합니다.

LU 이름

1-8자의 LU 이름(단일 LU에 대한) 또는 1-5자의 기본 이름(LU 범위의 경우, 정의된 LU의 모든 이름을 형성하기 위해 기본 이름에 접두부가 붙습니다.)

LU 이름은 로컬로만 사용되므로, 호스트에 정의된 이름에 일치하지 않아도 됩니다.

호스트 LS/DLUR PU

호스트로의 링크를 제공하는 링크 스테이션. LU 정의는 사용자가 선택하는 링크 스테이션에 속합니다. (중속 LU가 DLUR을 지원하는 노드에 상주할 경우, 이 필드는 호스트와의 연결을 제공하는 DLUR PU를 식별합니다.)

LU 번호

LU 번호 또는 LU 번호 범위. LU 번호는 1-255가 될 수 있습니다.

LU 번호는 호스트 VTAM 구성에 있는 LU 번호와 일치해야 합니다. 호스트에 구성된 번호를 모르면 SNA 네트워크 계획자에게 문의하십시오.

LU 유형

다음 LU 유형 중 하나(구성 중인 LU 유형에 따라).

- 3270 표시장치 LU의 경우 화면 크기에 따라 적절한 모델을 지정하십시오.
 - 3270 모델 2(80x24)
 - 3270 모델 3(80x32)
 - 3270 모델 4(80x43)
 - 3270 모델 5(132x27)
- 프린터 LU의 경우 다음 중 하나를 지정하십시오.
 - 3270 프린터
 - SCS 프린터
- LU 유형을 모를 경우, 로컬 노드에서 호스트로의 SNA 게이트웨이를 지원하기 위해 LU를 사용하거나(업스트림 LU) LU가 LUA 응용프로그램용이면 무제한(명령행 구성의 경우 알 수 없음)을 지정하십시오.

LU 유형은 호스트에서의 LU 구성과 일치해야 합니다. 필요한 경우 호스트에 구성된 LU 유형이 우선합니다.

지정하는 값에 따라 Linux용 Communications Server가 표준 VTAM 테이블에 사용된 값과 일치하도록 다음 문자열 중 하나를 DDDLUNMVT의 호스트에 보냅니다.

3270002: 3270 모델 2의 경우

3270003: 3270 모델 3의 경우

- 3270004: 3270 모델 4의 경우
- 3270005: 3270 모델 5의 경우
- 3270DSC: 3270 프린터의 경우
- 3270SCS: SCS 프린터의 경우
- 3270000: RJE 워크스테이션의 경우
- 327000n: TN3270 클라이언트에서 무제한 여기서 n은 클라이언트가 제공하는 모델 번호(2-5)
- 3270000: LUA 클라이언트에서 무제한

호스트에서 TN 서버 및 DDDLU로 LU를 사용할 경우 LU는 호스트에서 구성되지 않을 수 있습니다. 이 경우, 여기에서 지정하는 LU 유형을 사용하여 호스트에서 동적으로 LU를 정의합니다. LU 모델 유형을 다운스트림 TN3270 클라이언트에서 요청한 유형과 일치하도록 정의하려면 무제한(명령행 구성에 대해 알 수 없음)을 지정하십시오. Linux용 Communications Server는 일반적으로 클라이언트가 지정한 터미널 유형(장치 유형)에서 표준 맵핑을 사용하여 LU 모델을 판별합니다. 이 맵핑을 변경해야 하는 경우에는 207 페이지의 부록 C 『DDLU용 TN3270 LU 모델 구성』에서 설명하는 **tn3270dev.dat** 파일을 사용하여 변경할 수 있습니다.

폴 내의 LU

LU가 LU 폴에 지정되는지 여부를 표시합니다.

주: 해당 LU를 LU 폴에 지정하고 사용자 세션을 해당 LU에 지정할 경우, 사용자 세션은 사용 가능할 경우 이 LU를 사용합니다. 그렇지 않으면 특정 LU 대신 LU 폴에 이를 지정하였어도 폴에서 해제된 LU를 사용합니다. LU가 이미 사용 중이면 사용자 세션을 설정할 수 없게 하기 위해 사용자가 지정된 LU만을 사용하게 할 경우, LU가 폴에 없는지 확인하십시오.

폴 이름

LU 폴의 이름

추가 구성

LU 유형 0-3 구성을 수행하고 나면 다음 구성 task로 계속하십시오.

- 3270 표시장치, TN3270 또는 LUA에 대해 종속 LU 폴을 사용하려면, 88 페이지의 『LU 폴 정의』에 설명된 대로 LU 폴을 정의하십시오.
- TN3270의 경우, 119 페이지의 『TN 서버 구성』에 설명된 대로 TN3270 클라이언트 액세스 레코드를 정의하십시오.

LU 풀 정의

3270, TN3270 및 LUA의 경우, LU 풀을 정의하여 사용자 구성을 단순화하고 호스트 세션 작성 시 보다 나은 융통성을 제공할 수 있습니다.

주: 사용자 세션을 특정 LU 또는 LU 풀로 지정할 수 있습니다.

- 풀에 있는 특정 LU에 사용자 세션을 지정하면, 세션은 사용 가능할 경우 이 LU를 사용합니다. 그렇지 않으면 특정 LU 대신 LU 풀에 이를 지정하였어도 풀에서 해제된 LU를 사용합니다.
- LU가 이미 사용 중이면 사용자 세션을 설정할 수 없게 하기 위해 사용자가 지정된 LU만을 사용하게 할 경우, LU가 풀에 없는지 확인하십시오.

LU 풀은 복수의 Linux용 Communications Server에 분산될 수 있습니다. (다른 서버에서 같은 이름을 사용하여 LU 풀을 정의하십시오.) 서버가 실패하거나 서비스를 받지 못할 경우, LU 풀을 사용하는 클라이언트는 다른 서버를 사용할 수 있습니다. LU 풀을 사용하면 클라이언트 구성도 단순해지고 다른 서버를 추가하거나 기존 서버에서 LU를 추가하여 용량을 쉽게 늘릴 수 있습니다.

LU 풀 창을 사용하여 Linux용 Communications Server 도메인에 해당되는 모든 LU 풀을 볼 수 있습니다. 이 창은 시스템에 구성된 LU 풀을 나열하며 이 창에서 LU 풀에 추가할 LU를 선택할 수 있습니다. LU 풀의 각 LU는 LU 풀 아래에 나열됩니다.

LU는 다음과 같이 식별됩니다.

- 3270 표시장치 LU
- 무제한 LU
- SCS 프린터
- 3270 프린터

서로 다른 유형의 LU를 같은 풀에 혼합하지 마십시오. (예: 표시장치 LU와 프린터 LU를 같은 풀에 두지 마십시오.) TN3270E 클라이언트를 지원하지 않으면 프린터 LU 풀이 필요하지 않습니다.

LU 풀을 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 창 메뉴에서 LU 풀을 선택한 후 새로 작성을 선택하여 풀을 추가하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

```
define_lu_pool
```

LU 풀 구성 매개변수

다음 매개변수는 LU 풀 구성에 필요합니다.

이름 LU 풀을 식별할 이름입니다. 이 필드는 새 LU 풀을 추가할 경우에만 적용됩니다. 기존 풀의 이름은 변경할 수 없습니다.

지정된 LU

풀에 지정할 LU. LU는 풀 하나의 멤버만 될 수 있습니다.

LU 풀 정의

제 6 장 APPC 통신 구성

APPC 응용프로그램, 5250 에뮬레이션 프로그램 및 CPI-C 응용프로그램에서는 사용자가 먼저 APPC를 구성해야 합니다. APPC 응용프로그램은 지정된 모드를 사용하여, 호스트나 피어 컴퓨터에서 다른 APPC나 CPI-C 응용프로그램과 통신하기 위해 노드의 LU 유형 6.2 자원을 사용합니다.

응용프로그램이 CPI-C를 사용하는 경우 APPC 구성 후 추가 CPI-C 구성을 수행하지 않아도 됩니다. CPI-C 응용프로그램은 노드의 LU 유형 6.2와 모드 자원을 사용하여, 호스트나 피어 컴퓨터에서 다른 APPC나 CPI-C 응용프로그램과 통신합니다. APPC 응용프로그램의 경우와 같은 자원을 CPI-C 응용프로그램에 정의합니다. 또한 Linux용 Communications Server 컴퓨터의 TP가 기동 TP(대화를 시작하는 TP)일 경우, 112 페이지의 『CPI-C 부가 정보 정의』에 설명된 대로 하나 이상의 부가 정보 항목을 정의해야 할 수도 있습니다. 이러한 항목들은 각각 상대방 TP, 상대방 TP에 액세스하기 위해 사용되는 LU 및 모드 자원, 그리고 필요한 보안 정보를 제공합니다.

APPC에 대한 구성 단계는 LU 6.2 트래픽이 종속 또는 독립 트래픽인지 여부에 따라 다릅니다. 원격 노드가 호스트가 아니면 독립 트래픽을 사용해야 합니다. 원격 노드가 호스트이면 종속 또는 독립 트래픽을 사용할 수 있습니다.

APPC 통신을 구성하려면 먼저 다음 구성을 수행해야 합니다.

- 62 페이지의 『노드 구성』에 설명된 대로 노드를 구성하십시오.
- 67 페이지의 제 4 장 『연결 구성요소 정의』에 설명된 대로 연결성을 구성하십시오.

주: APPN 네트워크에서는 인접 네트워크 노드에 대한 단일 링크 스테이션을 사용하여 네트워크에 있는 원격 노드와 통신할 수 있으므로, 원격 노드마다 별도의 링크 스테이션을 구성하지 않아도 됩니다.

많은 경우에서, APPC 응용프로그램은 로컬 및 원격 노드와 표준 모드에서 제어점(CP) LU를 사용할 수 있습니다. 이 경우, 구성은 추가 구성 없이 APPC 준비 상태가 됩니다.

다음 단계를 사용하여 로컬 노드에서 APPC 통신을 구성할 수 있습니다. 로컬 및 원격 노드 유형과, 사용자의 응용프로그램에 따라 다음 단계를 수행하지 않을 수도 있습니다.

1. 92 페이지의 『로컬 LU 정의』에 설명된 대로 로컬 LU를 정의하십시오.
2. 94 페이지의 『원격 노드 정의』에 설명된 대로 원격 노드를 정의하십시오.
3. 95 페이지의 『상대방 LU 정의』에 설명된 대로 상대방 LU를 정의하십시오.
4. 99 페이지의 『TP 정의』에 설명된 대로 기동 가능 TP를 정의하십시오.

5. 107 페이지의 『모드 및 서비스 클래스 정의』에 설명된 대로 모드를 정의하십시오.
6. 112 페이지의 『CPI-C 부가 정보 정의』에 설명된 대로 CPI-C 부가 정보를 정의하십시오.
7. 114 페이지의 『APPC 보안 구성』에 설명된 대로 APPC 보안을 정의하십시오.
8. 5250 통신을 구성하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』을 참조하십시오.

로컬 LU 정의

많은 경우에, 응용프로그램은 노드를 구성할 때 자동으로 정의되는 로컬 노드의 제어점 LU를 사용할 수 있습니다. 이것은 디폴트 LU입니다. 응용프로그램이 특정 LU를 지정하지 않으면 이 디폴트 LU를 사용할 수 있습니다. 응용프로그램이 디폴트 LU를 사용하면, 로컬 LU를 정의할 필요가 없습니다. APPC 응용프로그램 문서를 확인하거나 응용프로그램 프로그래머에게 문의하십시오.

APPC나 CPI-C 응용프로그램과 함께 사용할 수 있도록 유형 6.2 종속 LU를 구성할 경우, 이 LU를 디폴트 풀의 멤버로 정의할 수 있습니다. 특정 로컬 LU를 지정하지 않는 응용프로그램은 디폴트 LU로 정의된 LU의 풀에서 사용하지 않는 LU에 지정됩니다.

종속 LU 6.2를 디폴트 LU로 정의할 수 있습니다. (여러 노드에서 디폴트 LU를 정의할 수도 있습니다.) 디폴트 LU를 요청하는 응용프로그램은 이들 LU 중 사용 가능한 LU에 지정할 수 있습니다. LU는 응용프로그램과 같은 컴퓨터에 있어서는 안 됩니다. 그러나 응용프로그램에 대해 상대방 LU를 정의할 경우, 응용프로그램이 임의 노드에 정의된 디폴트 로컬 LU를 사용하여 올바른 상대방 LU에 접속할 수 있도록, 디폴트 LU가 정의된 모든 노드에 상대방 LU를 정의해야 합니다.

독립 APPC 및 5250은 독립 LU를 사용합니다. 각각의 LU-LU 세션에는 로컬 LU와 상대방 LU가 연관됩니다. 로컬 LU의 경우, 노드 제어점(CP)과 연관되는 사전 정의된 디폴트 LU를 사용하거나 새 로컬 LU를 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. 그러나 사용자의 네트워크가 APPN 네트워크가 아니거나 노드가 LEN 노드이면 상대방 LU를 구성해야 합니다. 이 경우 상대방 LU가 상주하는 원격 노드를 구성한 후 원격 노드에서 상대방 LU를 정의해야 합니다. (상대방 LU가 원격 노드에서 디폴트 LU이면, 원격 노드를 정의할 때 자동으로 추가되므로 명시적으로 정의하지 않아도 됩니다.)

APPC 로컬 LU를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **APPC**와 새 독립 로컬 LU 또는 새 종속 로컬 LU 를 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

define_local_lu

고급 대화상자를 사용하여 동기점 지원, 경로지정 연결 특성, 암호 대체 사용 여부, SSCP 액세스에 대한 제한사항 LU와 연관된 시스템 이름 및 보안을 지정할 수 있습니다.

로컬 LU 구성 매개변수

다음 매개변수는 로컬 LU 구성에 필요합니다.

LU 이름

로컬 LU의 LU 이름입니다.

사용할 이름을 모르면 SNA 네트워크 계획자에게 문의하십시오.

이 LU 이름은 로컬 LU의 전체 LU 이름에서 두 번째 부분입니다. 전체 LU 이름의 첫 번째 부분(네트워크 이름)은 항상 로컬 노드의 CP 이름에서 첫 번째 부분과 같습니다.

LU 별명

LU의 LU 별명입니다. 별명을 입력하지 않으면 LU 이름이 별명으로 사용됩니다.

호스트 LS/DLUR PU

LU가 속하는 DLUR PU나 호스트 링크 스테이션의 이름입니다. (이 필드는 LU가 종속 LU인 경우에만 적용됩니다.)

LU 번호

종속 LU의 LU 번호. (이 필드는 LU가 종속 LU인 경우에만 적용됩니다.)

디폴트 풀의 멤버

LU를 디폴트 종속 APPC LU 풀의 멤버로 만들것인지 여부를 표시합니다. 사용할 특정 로컬 LU를 지정하지 않는 응용프로그램은 디폴트 풀에서 사용 가능한 LU에 지정됩니다.

이 필드는 LU가 종속 LU인 경우에만 적용됩니다.

추가 구성

로컬 LU 구성을 수행하고 나면 다음 구성 task로 계속하십시오.

- 원격 노드를 정의하려면 94 페이지의 『원격 노드 정의』를 참조하십시오.
- 상대방 LU를 정의하려면 95 페이지의 『상대방 LU 정의』를 참조하십시오.

- 기동 가능 TP를 정의하려면 99 페이지의 『TP 정의』를 참조하십시오.
- 모드를 정의하려면 107 페이지의 『모드 및 서비스 클래스 정의』를 참조하십시오.
- CPI-C 부가 정보를 정의하려면 112 페이지의 『CPI-C 부가 정보 정의』를 참조하십시오.
- APPC 보안을 정의하려면 114 페이지의 『APPC 보안 구성』을 참조하십시오.
- 5250 통신을 구성하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』을 참조하십시오.

원격 노드 정의

다음과 같은 상황에서는 원격 노드와 노드의 상대방 LU를 정의해야 합니다.

- 로컬 노드가 LEN 노드이면, 모든 원격 노드 및 APPC를 사용하여 원격 노드가 통신하는 원격 노드의 모든 상대방 LU를 정의해야 합니다. LEN 노드는 동적으로 상대방 LU를 위치 지정할 수 없지만, 원격 노드 정의를 사용하면 LEN 노드가 상대방 LU를 위치 지정할 수 있습니다.
- 원격 노드가 LEN 노드이고 로컬 노드가 LEN 노드의 네트워크 노드 서버로 작동하는 네트워크 노드이면, 네트워크 노드 서버에서 LEN 노드(해당되는 상대방 LU도)를 원격 노드로 정의해야 합니다. 이 정의를 통해 APPN 네트워크의 나머지에 있는 노드는 LEN 노드에서 LU를 위치 지정할 수 있습니다.
- 원격 노드가 서로 다른 APPN 네트워크에 있을 경우, 원격 노드는 동적으로 위치 지정할 수 없으므로 사용자가 정의해야 합니다.

원격 노드를 정의해야 하는데 링크 스테이션을 정의할 때 원격 노드를 정의하지 않은 경우, 먼저 원격 노드를 정의해야 링크를 통한 APPC 통신을 사용할 수 있습니다.

원격 노드 정의를 추가할 때, 원격 노드와 이름이 같은 상대방 LU가 자동으로 추가됩니다. 이 LU는 원격 노드의 제어점 LU입니다. 응용프로그램이 이 상대방 LU를 사용하면, 상대방 LU의 LU 별명을 추가하려는 경우에도 다른 상대방 LU를 추가할 필요가 없습니다. 별명을 추가하려면 상대방 LU를 두 번 누르고 상대방 LU 구성 대화상자에 별명을 입력하십시오.

로컬 노드와 원격 노드가 둘 다 끝 노드 또는 네트워크 노드이고 APPN 네트워크의 일부일 경우, 상대방 LU는 필요할 때 동적으로 위치 지정됩니다. 이 경우, LU가 있는 원격 노드를 정의하지 마십시오. 노드를 정의하면 LU 위치를 동적으로 지정하는 APPN의 프로토콜에 결함이 발생할 수 있기 때문입니다.

이러한 결함을 방지하기 위해 Linux용 Communications Server는 사용자가 CP-CP 세션이 활동 중인(또는 과거에 CP-CP 세션을 가지고 있었던) 원격 노드를 정의하지 못하도록 합니다. 또한 이전에 원격 노드를 정의했는데 Linux용 Communications Server

가 그 원격 노드에 대해 CP-CP 세션을 작성할 경우, 항목은 임시로 동적 항목으로 변환됩니다. 노드가 활동 중이 아닐 때 원격 노드 정의를 삭제하여 결함을 수정해야 합니다.

원격 노드를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **APPC**와 새 원격 노드를 선택하십시오.

명령행 관리 프로그램

원격 노드를 정의하려면 다음 명령을 실행하십시오.

define_directory_entry

상대방 LU를 정의하려면 다음 명령을 실행하십시오.

define_partner_lu

원격 노드 구성 매개변수

다음 매개변수는 원격 노드 구성에 필요합니다.

노드의 SNA 네트워크 이름

원격 노드의 전체 CP 이름입니다. 이 대화상자에 입력한 값은 원격 노드에서 구성된 CP 이름과 일치해야 합니다.

추가 구성

원격 노드 구성을 수행하고 나면 다음 구성 태스크로 계속하십시오.

- 상대방 LU를 정의하려면 『상대방 LU 정의』를 참조하십시오.
- 기동 가능 TP를 정의하려면 99 페이지의 『TP 정의』를 참조하십시오.
- 모드를 정의하려면 107 페이지의 『모드 및 서비스 클래스 정의』를 참조하십시오.
- CPI-C 부가 정보를 정의하려면 112 페이지의 『CPI-C 부가 정보 정의』를 참조하십시오.
- APPC 보안을 정의하려면 114 페이지의 『APPC 보안 구성』을 참조하십시오.
- 5250 통신을 구성하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』을 참조하십시오.

상대방 LU 정의

로컬 노드와 원격 노드가 둘 다 네트워크 노드이거나 하나는 네트워크 노드이고 다른 하나는 끝 노드인데 사용자 응용프로그램이 상대방 LU를 참조하기 위해 LU 이름을 사용할 경우, 상대방 LU는 APPN을 사용하여 동적으로 위치 지정되므로 상대방 LU를 정의하지 않아도 됩니다. 그러나 응용프로그램이 해당되는 상대방 LU를 참조하기 위해 LU 별명을 사용하는 경우, 상대방 LU 별명 정의를 추가해야 합니다.

로컬 노드나 원격 노드가 LEN 노드인 경우, LEN 노드는 LU의 동적 위치에 참여할 수 없으므로 상대방 LU를 원격 노드의 하위로 정의해야 합니다. 응용프로그램이 상대방 LU로 원격 노드의 제어점 LU를 사용하면, 원격 노드를 정의할 때 제어점 LU가 자동으로 정의된 것입니다.

와일드카드를 사용하여 모두 동일한 원격 노드에 있고 이름이 동일한 문자로 시작하는 복수의 상대방 LU를 구성할 수 있습니다. 와일드카드를 사용한다는 것은 각각의 상대방 LU를 별도로 구성하지 않아도 됨을 의미합니다.

상대방 LU를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

Motif 관리 프로그램을 사용하여 상대방 LU 별명을 추가하거나, 특정 원격 노드에서 상대방 LU의 정의를 추가하거나, 또는 와일드카드를 사용하여 복수의 상대방 LU를 정의할 수 있습니다. 노드 창의 서비스 메뉴에서 **APPC** 및 새 상대방 LU와 다음 중 하나를 선택하십시오.

- 상대방 LU 별명
- 원격 노드의 상대방 LU
- 원격 노드의 와일드카드 상대방 LU

명령행 관리 프로그램

상대방 LU를 정의하려면 다음 명령을 실행하십시오.

define_partner_lu

LEN 노드를 상대방 LU로 정의하려면 다음 명령을 실행하십시오.

define_adjacent_len_node

define_directory_entry

상대방 LU 구성 매개변수

다음 매개변수는 상대방 LU 구성에 필요합니다.

상대방 LU 이름

상대방 LU의 전체 LU 이름입니다. 이 이름은 해당 LU에 대해 원격 노드에서 구성한 이름과 일치해야 합니다. 이름을 모르면 SNA 네트워크 계획자에게 문의하십시오.

이 필드는 특정 원격 노드에서 상대방 LU를 정의하거나 상대방 LU 별명을 정의할 때 적용됩니다.

와일드카드 상대방 LU 이름

복수 상대방 LU의 전체 LU 이름과 일치하는 이름입니다. (이 필드는 와일드

카드를 사용하는 상대방 LU를 정의할 경우에만 적용됩니다.) 와일드카드 상대방 LU 이름은 각각 1-8자인 두 개의 문자열로 구성됩니다.

- 첫 번째 문자열은 정확히 전체 상대방 LU 이름의 첫 번째 부분과 일치하는 완전한 SNA 네트워크 이름이거나, 상대방 LU의 네트워크 이름 시작 부분과 일치하는 와일드카드 접두부가 될 수 있습니다. 첫 번째 문자열의 값으로 와일드카드 접두부를 제공할 경우, 두 번째 문자열은 공백으로 두십시오. 예를 들어, 와일드카드 항목 A는 SNA 네트워크에서 이름이 A, ANT 또는 APPN(BUFFALO 또는 ZEBRA는 아님)인 모든 LU와 일치합니다.
- 첫 번째 문자열에 대해 완전한 SNA 네트워크 이름을 제공할 경우, 두 번째 문자열에도 값을 입력할 수 있습니다. (첫 번째 문자열에 대해 유효한 SNA 네트워크 이름을 제공하지 않으면 두 번째 문자열을 지정할 수 없습니다.) 두 번째 문자열은 전체 상대방 LU 이름에서 두 번째 부분의 시작 부분과 일치해야 하는 와일드카드 접두부로 처리됩니다. 예를 들어, 와일드카드 항목 A.F는 상대방 LU 이름 A.FRED 또는 A.FREDDY(APPN.FRED 또는 A.B는 아님)와 일치합니다.

두 문자열을 모두 공백으로 두면 와일드카드 상대방 LU 정의는 상대방 LU 이름과 일치합니다.

별명 로컬로 표시 가능한 상대방 LU의 별명입니다. LU 별명을 사용하여 상대방 LU를 참조하는 로컬 응용프로그램이 없을 경우 LU 별명을 지정하지 않아도 됩니다.

이 필드는 특정 원격 노드에서 상대방 LU를 정의하거나 상대방 LU 별명을 정의할 때 적용됩니다.

미해석 이름

호스트가 상대방 LU 및 로컬 LU 사이에 LU-LU 세션을 시작하도록 요청할 때 종속 로컬 LU가 사용하는 미해석 이름입니다. 이 이름을 사용하면 로컬로 구성된(응용프로그램에서 사용되는) 상대방 LU 이름이 호스트에 구성된 상대방 LU 이름과 다르게 됩니다.

디폴트 미해석 이름은 상대방 LU 이름의 두 번째 부분입니다. 이 이름은 대부분의 경우에 정확합니다. 의심이 가면 SNA 네트워크 계획자에게 문의하십시오.

이 필드는 특정 원격 노드에서 상대방 LU를 정의하거나 상대방 LU 별명을 정의할 때 적용됩니다.

병렬 세션 지원

상대방 LU가 한 번에 여러 세션을 지원할 수 있는지 여부를 표시합니다. 대부분의 경우, 상대방 LU는 한 번에 많은 세션을 지원하지 않지만 일부 LEN 노드는 병렬 세션을 지원하지 않습니다.

이 필드는 특정 원격 노드에서 상대방 LU를 정의하거나 상대방 LU 별명을 정의할 때 적용됩니다.

위치 상대방 LU가 상주하는 노드나 상대방 LU에 대한 액세스를 제공할 수 있는 노드의 전체 CP 이름입니다. 아직 정의되지 않은 원격 노드의 이름을 제공한 경우, 노드를 동적으로 검색할 수 없으면 그 노드를 정의해야 합니다.

이 필드는 특정 원격 노드에서 상대방 LU를 정의할 경우에만 적용됩니다.

상대방 LU의 링크 스테이션 경로지정 정의

링크 스테이션 경로지정을 사용하여, 도달하는 데 사용되는 링크 스테이션별로 상대방 LU의 위치를 식별할 수 있습니다.

주:

1. 링크 스테이션 경로지정은 자원을 동적으로 위치 지정할 수 있는 APPN 네트워크에는 필요하지 않습니다. APPN 네트워크는 정상적인 APPN 경로지정 메커니즘을 생략하므로 APPN 네트워크에서는 링크 스테이션 경로지정을 사용하지 않는 것이 좋습니다.
2. Enterprise Extender(HPR/IP) 링크 스테이션에서는 링크 스테이션 경로지정을 사용할 수 없습니다. 이는 이러한 링크 유형의 모든 트래픽이 특정 링크 스테이션에 고정되지 않아서 다른 경로로 전환할 수 있는 RTP 연결을 통해 전송되어야 하기 때문입니다.

상대방 LU의 링크 스테이션 경로지정을 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **APPC**, 새 상대방 LU 및 링크 스테이션 상의 상대방 LU를 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

```
define_ls_routing
```

링크 스테이션 경로지정 매개변수

다음 매개변수는 링크 스테이션 경로지정 구성에 필요합니다.

LU 이름

링크 스테이션을 제어하는 로컬 LU의 이름(상대방 LU가 특정 링크 스테이션을 통해 위치 지정될 경우)

LS 이름

링크 스테이션의 이름입니다.

상대방 LU 이름

상대방 LU의 전체 LU 이름이나 와일드카드 이름:

- 전체 LU 이름은 각각 1-8자인 두 개의 문자열로 구성됩니다.

이 이름은 해당 LU에 대해 원격 노드에서 구성한 이름과 일치해야 합니다. 이름을 모르면 SNA 네트워크 계획자에게 문의하십시오.

- 와일드카드 상대방 LU 이름은 복수 상대방 LU의 전체 LU 이름과 일치합니다. 와일드카드 상대방 LU 이름은 각각 1-8자인 두 개의 문자열로 구성됩니다.
 - 첫 번째 문자열은 정확히 전체 상대방 LU 이름의 첫 번째 부분과 일치하는 완전한 SNA 네트워크 이름이거나, 상대방 LU의 네트워크 이름 시작 부분과 일치하는 와일드카드 접두부가 될 수 있습니다. 첫 번째 문자열의 값으로 와일드카드 접두부를 제공할 경우, 두 번째 문자열은 공백으로 두십시오. 예를 들어, 와일드카드 항목 A는 SNA 네트워크에서 이름이 A, ANT 또는 APPN(BUFFALO 또는 ZEBRA는 아님)인 모든 LU와 일치합니다.
 - 첫 번째 문자열에 대해 완전한 SNA 네트워크 이름을 제공할 경우, 두 번째 문자열에도 값을 입력할 수 있습니다. (첫 번째 문자열에 대해 유효한 SNA 네트워크 이름을 제공하지 않으면 두 번째 문자열을 지정할 수 없습니다.) 두 번째 문자열은 전체 상대방 LU 이름에서 두 번째 부분의 시작 부분과 일치해야 하는 와일드카드 접두부로 처리됩니다. 예를 들어, 와일드카드 항목 A.F는 상대방 LU 이름 A.FRED 또는 A.FREDDY (APPN.FRED 또는 A.B는 아님)와 일치합니다.

두 문자열을 모두 공백으로 두면 와일드카드 상대방 LU 정의는 상대방 LU 이름과 일치합니다.

상대방 LU 이름을 와일드카드로 사용

상대방 LU 이름을 리터럴 전체 LU 이름이 아닌 와일드카드로 사용할 것인지 여부를 표시합니다.

추가 구성

상대방 LU 구성을 수행하고 나면 다음 구성 task로 계속하십시오.

- 기동 가능 TP를 정의하려면 『TP 정의』를 참조하십시오.
- 모드를 정의하려면 107 페이지의 『모드 및 서비스 클래스 정의』를 참조하십시오.
- CPI-C 부가 정보를 정의하려면 112 페이지의 『CPI-C 부가 정보 정의』를 참조하십시오.
- APPC 보안을 정의하려면 114 페이지의 『APPC 보안 구성』을 참조하십시오.
- 5250 통신을 구성하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』을 참조하십시오.

TP 정의

이 절은 APPC TP를 정의하는 방법에 대해 설명합니다.

대부분의 경우, Linux용 Communications Server 시스템에서 실행되는 TP는 정의하지 않아도 됩니다. 그러나 다음과 같은 경우에는 TP 정의를 구성해야 합니다.

APPC 특성

Linux용 Communications Server 컴퓨터의 TP가 기동 TP(또는 APPC 대화를 시작하는 소스 TP)이고 TP에 대한 액세스를 제한하지 않아도 될 경우에는 TP를 정의하지 않아도 됩니다. 그러나 다음 특성을 지정하려면 106 페이지의 『TP 정의 매개변수』에 설명된 것처럼 APPC TP를 정의할 수 있습니다.

- TP의 대화 보안을 정의할 경우
- TP가 기본 대화 또는 직접 대화 중 어느 것을 사용할 것인지 표시할 경우
- 동기점 처리를 지정할 경우
- PIP 데이터 처리를 지정할 경우

기동 가능 TP

입력되는 할당 요청에 대한 응답으로 TP가 자동으로 시작되도록 하려면, 102 페이지의 『서버에서의 TP 기동 매개변수』에 설명된 대로 기동 가능 TP로 정의하십시오.

기동 가능 TP(또는 목표 TP)는 입력되는 할당 요청에 대한 응답으로 시작되는 TP입니다. 기동 가능 TP에 대해 TP 정의를 작성해야 합니다. 기동 가능 TP는 RECEIVE_ALLOCATE를 실행하는 APPC TP가 될 수 있으며 Accept_Conversation 또는 Accept_Incoming을 실행하는 CPI-C 응용프로그램이 될 수도 있습니다.

주: 이 매뉴얼에서, 『Receive_Allocate』는 이러한 세 가지의 API 호출을 표시하기 위해 사용됩니다.

입력되는 할당 요청을 실행 중인 TP로 전송하도록 기동 가능 TP를 정의할 수도 있습니다.

기동 가능 TP의 경우, 시간종료 값을 지정하여 할당 요청에 대한 대기를 제한할 수도 있습니다. (이 옵션은 명령행 관리를 사용해야만 구성할 수 있습니다.)

Linux용 Communications Server는 다음 목적으로 기동 가능 TP 정의를 사용합니다.

- TP가 Receive_Allocate를 실행할 때, Linux용 Communications Server는 적절한 TP 이름을 사용하여 기동 가능 TP 정의를 검색합니다. 정의가 존재하고 정의에 Receive_Allocate 시간종료 값이 포함되어 있으면, Linux용 Communications Server는 Receive_Allocate를 처리할 때 이 값을 사용합니다. 그렇지 않으면 디폴트(시간종료 없음: TP가 무한정 대기하도록 함)를 사용합니다.
- 입력되는 Allocate 요청이 목표 시스템에 도달했는데 요청한 TP가 Receive_Allocate 미해결 상태로 아직 실행되고 있지 않을 경우, Linux용

Communications Server는 입력되는 Allocate에 지정된 TP 이름으로 TP 정의를 검색합니다. 정의가 존재하면 Linux용 Communications Server는 이 정의에 있는 정보를 사용하여 TP를 시작하거나(복수 인스턴스가 허용되거나 TP가 아직 실행되고 있지 않은 경우) 입력되는 Allocate를 대기행렬에 넣어야 할 것으로 판별합니다(TP가 이미 실행 중이고 복수 인스턴스가 허용되지 않을 경우).

필요하면 같은 TP에 대해 두 유형의 정의를 구성할 수 있습니다(예: 기동 가능 TP에 대한 대화 보안을 정의하기 위해).

TP 정의를 구성하려면 다음 방법 중 하나를 사용하십시오.

APPC 특성을 정의할 경우:

다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **APPC**와 트랜잭션 프로그램을 선택하십시오. Linux용 Communications Server가 TP 창을 표시하면, 맨 아래 분할창을 선택하고 새로 작성 버튼을 누르거나 기존 TP 정의를 선택하고 등록정보 버튼을 누르십시오.

명령행 관리 프로그램

snaadmin define_tp 명령을 실행하십시오.

기동 가능 TP를 정의할 경우:

서버와 클라이언트에 대한 구성 방법은 다릅니다.

- 서버에서는 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **APPC**와 트랜잭션 프로그램을 선택하십시오. Linux용 Communications Server가 TP 창을 표시하면, 맨 위 분할창을 선택하고 새로 작성 버튼을 누르거나 기존 기동 가능 TP 정의를 선택하고 등록정보 버튼을 누르십시오.

명령행 관리

snatpinstall 명령을 실행하십시오.

-

AIX, LINUX

AIX 또는 Linux의 IBM 원격 API 클라이언트에서 **snatpinstall** 명령을 실행하십시오.

•

WINDOWS

Windows 클라이언트에서는 클라이언트 소프트웨어가 설치된 디렉토리로 변경한 후 **tpinst32** 명령을 실행하십시오. (이 명령은 32비트 및 x64 버전의 Windows모두에 적용됩니다.)



snatpinstall 또는 **tpinst32** 명령 사용에 대해서는 197 페이지의 부록 B 『명령행에서 기동 가능 TP 구성』을 참조하십시오.

서버에서의 TP 기동 매개변수

이 절에서는 서버에서 기동 가능 TP를 구성할 때 Motif 관리 프로그램이나 명령행 관리 프로그램이 요구하는 매개변수에 대해 설명합니다. 클라이언트에서의 기동 가능 TP 구성에 대한 정보는 197 페이지의 부록 B 『명령행에서 기동 가능 TP 구성』을 참조하십시오.

다음 매개변수는 로컬 노드에서 기동할 수 있는 TP에 필수입니다.

TP 이름

다음 양식 중 하나로 된 TP 이름

응용프로그램 TP

TP가 사용자 응용프로그램일 경우, 이름을 정상 문자로 제공하십시오 (최대 길이 64자).

서비스 TP

TP가 SNA 서비스 트랜잭션 프로그램이면 이름을 16진수로 입력하십시오(최대 4바이트를 표시하는 8자리의 16진수).

각각의 TP 정의가 다른 LU 별명을 지정할 경우, TP 이름이 같은 여러 APPC 기동 가능 TP를 정의할 수 있습니다. 사용할 특정 LU 별명을 지정할 수 없으므로 CPI-I 기동 가능 TP에 대해서는 이를 수행할 수 없습니다. 각각의 CPI-I 기동 가능 TP는 다른 이름이어야 합니다.

매개변수는 임의 LU에서/특정 LU에서의 기동에 대한 것임

TP가 APPC TP인 경우, 이 매개변수는 임의 LU나 특정 LU에서 TP를 기동 가능하도록 만들 것인지 여부를 지정합니다. 디폴트로, TP는 어떤 LU에서도 기동할 수 있습니다.

주: TP가 CPI-C 응용프로그램일 경우, 이 필드는 임의 LU에서 TP를 기동 가능하도록 설정해야 합니다. CPI-C는 특정 로컬 LU로부터 입력되는 Attach

의 승인을 지원하지 않으므로, CPI-C 응용프로그램에 이 옵션을 지정하려고 하면 입력되는 Attach를 TP에 전송할 때 오류가 발생합니다.

LU 별명

이 필드는 TP가 CPI-C 응용프로그램일 경우 사용하지 않아야 합니다. TP가 APPC 응용프로그램일 경우, 이 TP 정의에 대한 매개변수가 임의 LU에서의 기동에 해당됨을 지정할 경우에만 적용됩니다.

TP가 입력되는 Attach를 승인할 로컬 LU 별명입니다. 이 이름은 Linux용 Communications Server 노드에서 로컬 APPC LU의 이름과 일치해야 합니다. LU 별명을 지정하지 않을 경우, TP는 로컬 LU로부터 입력되는 Attach를 승인합니다.

공백이 아닌 LU 별명을 지정한 경우, TP는 RECEIVE_ALLOCATE Verb의 확장 양식을 사용하고 이 LU 별명을 Verb에 대한 매개변수로 지정해야 합니다. 그러면 Linux용 Communications Server가 입력되는 Attach를 올바른 TP에 전송할 수 있습니다. 다른 형식의 RECEIVE_ALLOCATE에 대한 자세한 정보는 AIX 또는 *Linux용 IBM Communications Server APPC 프로그래머 안내서*를 참조하십시오. TP가 응용프로그램에 빌드하기 보다는 런타임에서 올바른 LU 별명을 판별할 수 있도록 허용하려면, 적절한 LU 별명을 포함하도록 환경 변수를 설정하고(환경 매개변수 사용), 응용프로그램이 RECEIVE_ALLOCATE를 실행할 방법을 판별하기 위해 환경 변수를 읽도록 설계하면 됩니다.

각각의 TP 정의가 다른 LU 별명을 지정할 경우, TP 이름이 같은 여러 TP를 정의할 수 있습니다.

지원되는 복수 인스턴스

이 옵션을 선택하지 않을 경우 TP는 대기행렬에 넣어진 TP입니다. TP를 실행하는 동안 입력되는 Allocate 요청은 TP가 다른 Receive_Allocate를 실행할 때까지 또는 실행을 완료하고 다시 시작될 수 있을 때까지 대기행렬에 있게 됩니다. 입력되는 Allocate 요청은 입력되는 Allocate 요청이 컴퓨터에 전송되도록 구성된 LU가 요청을 수신할 경우, 또는 컴퓨터에서 정보를 전송하지 않도록 구성된 LU가 요청을 수신할 경우에만 TP에 전송됩니다.

이 옵션을 선택할 경우 TP는 대기행렬에 없는 TP입니다. Linux용 Communications Server는 입력되는 Allocate 요청이 도달할 때마다 새 TP 사본을 시작합니다. 대기행렬에 없는 TP를 운영자가 시작할 수 없습니다. 그러한 TP는 항상 Linux용 Communications Server에 의해 자동으로 시작됩니다. 대기행렬에 없는 TP의 경우, Linux용 Communications Server는 한 번에 여러 개의 TP 사본을 실행할 수 있도록 허용합니다. 모든 사본은 같은 사용자 및 그룹 ID와 같은 작업 디렉토리를 사용하여 실행됩니다(사용자 ID 및 그룹 ID 매개변수에 정의된 대로). TP가 로컬 시스템에서 파일에 쓸 경우, 서로 다른 TP 사본이 서로의 파일 위에 겹쳐쓰지 않도록 해야 합니다.

대기행렬에 없는 TP가 대화를 종료하고 나면, 그 TP는 종료되거나 다른 RECEIVE_ALLOCATE를 발행할 수 있습니다. 자주 사용하는 프로그램의 경우, 이렇게 하면 대화마다 프로그램의 새 인스턴스를 시작하므로서 발생하는 성능 오버헤드를 피할 수 있습니다. 대기행렬에 없고 자동으로 시작되는 TP에 대해 Attach가 수신될 때마다, Linux용 Communications Server는 해당 TP 인스턴스로부터의 미해결 RECEIVE_ALLOCATE가 있는지 확인합니다. 있으면 TP는 입력되는 대화에 사용되고, 그렇지 않으면 Linux용 Communications Server가 프로그램의 새 인스턴스를 시작합니다.

입력 Allocate를 실행 중인 TP로 전송

이 옵션은 복수 인스턴스가 지원되지 않을 경우에만 적용됩니다.

TP가 대기행렬에 놓인 브로드캐스트 TP일 경우에 이 옵션을 선택하십시오. TP를 실행하는 동안 입력되는 Allocate 요청은 TP가 다른 Receive_Allocate를 실행할 때까지 또는 실행을 완료하고 다시 시작될 수 있을 때까지 대기행렬에 있게 됩니다. TP가 시작되면 TP에 관한 정보가 LAN의 모든 서버로 브로드캐스트됩니다. 다른 컴퓨터의 LU가 입력되는 Allocate 요청을 수신하고 어떤 경로지정 정보도 구성되어 있지 않은 경우, 그 LU는 동적으로 TP를 찾아서 Allocate 요청을 전송할 수 있습니다.

이 옵션을 사용하면 LU에 대해 명시적 경로지정 정보를 구성하지 않아도 되므로, 여러 컴퓨터에서 동일 TP의 여러 사본을 실행하여 로드 균형을 이룰 수 있습니다. 그러나 LAN 트래픽을 줄이기 위해 정보를 브로드캐스트하지 않거나 특정 LU에 도달하는 입력 Allocate 요청이 항상 TP의 동일 사본에 전송되도록 하려면 이 옵션을 선택하지 마십시오.

TP 실행 프로그램의 전체 경로

TP용 실행 파일의 전체 경로 및 파일 이름입니다.

파일은 사용자 ID 매개변수에 지정한 사용자에게 실행 권한을 가지고 있어야 합니다. 또한 사용자 ID가 루트로 설정된 상태에서 실행 파일을 실행할 경우, 파일을 루트가 소유해야 하고 setuid 및 setgid 권한을 가지고 있어야 Linux용 Communications Server에 의해 자동으로 시작될 수 있습니다.

인수 TP에 전달할 명령행 인수입니다(공백으로 구분). 인수는 여기에 표시되는 순서와 동일하게 TP에 전달됩니다.

이 값은 선택적입니다. 이 값을 포함하지 않을 경우, TP는 명령행 인수 없이 가동됩니다.

사용자 ID

Linux용 Communications Server가 TP를 시작하기 위해 사용하는 사용자 ID입니다. 이 행은 필수이므로 반드시 지정해야 합니다. ID는 Linux용 Communications Server 컴퓨터에서 유효한 Linux 로그인 ID여야 합니다.

TP는 이 사용자 ID와 연관되는 홈 디렉토리에서 시작됩니다. 이 홈 디렉토리는 또한 추적 파일과 TP가 액세스하는 다른 파일에 대한 디폴트 경로이기도 합니다(응용프로그램이 전체 경로를 지정하여 대체하지 않는 한). 응용프로그램이 경로 없이 파일 이름을 지정할 경우, Linux용 Communications Server는 이 홈 디렉토리에서 파일을 검색합니다. 응용프로그램이 상대 경로를 사용하여 파일 이름을 지정하면, Linux용 Communications Server는 이 홈 디렉토리에 상대적으로 지정된 디렉토리에서 파일을 검색합니다.

TP 실행 프로그램의 전체 경로 매개변수에 지정된 TP용 실행 파일은 지정된 사용자에게 대해 실행 권한을 가지고 있어야 합니다. 또한 사용자 ID가 루트로 설정된 경우, 파일을 루트가 소유해야 하고 setuid 및 setgid 권한을 가지고 있어야 Linux용 Communications Server에 의해 자동으로 시작될 수 있습니다.

그룹 ID

Linux용 Communications Server가 TP를 시작하기 위해 사용하는 그룹 ID입니다. Linux용 Communications Server 컴퓨터에서 유효한 Linux 그룹 ID여야 합니다.

이 매개변수는 선택적입니다. 이 매개변수를 포함하지 않을 경우, 디폴트는 sna입니다.

표준 입력

TP가 사용하는 표준 입력 파일이나 장치의 전체 경로 이름을 지정하십시오.

이 매개변수는 선택적입니다. 이 매개변수를 포함하지 않을 경우, 디폴트는 /dev/null입니다.

표준 출력

TP가 사용하는 표준 출력 파일이나 장치의 전체 경로 이름을 지정하십시오.

이 매개변수는 선택적입니다. 이 매개변수를 포함하지 않을 경우, 디폴트는 /dev/null입니다.

표준 오류

TP가 사용하는 표준 오류 파일 또는 장치의 전체 경로 이름을 지정하십시오.

이 매개변수는 선택적입니다. 이 매개변수를 포함하지 않을 경우, 디폴트는 /dev/null입니다.

환경

TP에 필요한 환경 변수를 지정하십시오.

각 변수는 *environment_variable=value* 양식으로 지정되며, 255자까지 가능합니다. *environment_variable=value* 문자열에서는 = 문자 앞뒤에 공백이나 탭 문자가 있어서는 안됩니다.

TP 정의

Motif 관리 프로그램에서 여러 환경 변수를 지정해야 할 경우(최대 64개) | 문자를 사용하여 구분하십시오. 변수는 여기에 표시되는 순서와 동일하게 설정됩니다.

TP가 CPI-C 응용프로그램일 경우, 이 필드를 사용하여 환경 변수 APPCLLU를 설정할 수 없습니다. 로컬 LU는 자동으로 로드되는 CPI-C 응용프로그램의 경우 지정할 수 없습니다.

이 필드는 선택적입니다. 이 필드를 포함하지 않을 경우, 어떤 환경 변수도 사용되지 않습니다.

TP 정의 매개변수

APPC TP를 구성하여 대화 보안, 대화 유형, 동기 레벨 및 PIP 데이터 처리를 지정할 수 있습니다. APPC 통신용 TP를 정의할 경우 다음 매개변수가 필요합니다.

TP 이름

다음 양식 중 하나로 된 TP 이름입니다.

응용프로그램 TP

TP가 사용자 응용프로그램일 경우, 이름을 정상 문자로 제공하십시오 (최대 길이 64자).

서비스 TP

TP가 SNA 서비스 트랜잭션 프로그램이면 이름을 16진수로 입력하십시오(4바이트씩 표시하는 최대 8자리 16진수).

대화 레벨 보안 필요

할당 요청이 유효한 사용자 이름 및 암호(또는 이미 암호를 확인했음을 표시하는 표시기)를 포함해야 할 경우 이 옵션을 선택하십시오. 이 옵션을 지정하지 않을 경우 어떤 확인도 필요하지 않습니다.

액세스 제한

보안 액세스 리스트에 사용자 이름을 포함시켜야 할 경우 이 옵션을 선택하십시오. 이 필드는 대화 레벨 보안 필요 옵션을 선택할 경우에만 적용됩니다.

보안 액세스 리스트

TP에 액세스할 수 있는 사용자 ID가 있는 보안 액세스 리스트의 이름입니다. 액세스 제한 옵션을 선택하면, 이 값을 제공해야 합니다.

대화 유형

TP가 기본 대화만 허용하는지, 직접 대화만 허용하는지, 아니면 두 대화 유형 중 아무거나 허용하는지 여부를 지정하십시오.

동기 레벨

TP가 승인하는 동기화 확인 레벨을 지정하십시오. 확인 동기화에 대한 자세한 정보는 AIX 또는 Linux용 IBM Communications Server APPC 프로그래머 안내서를 참조하십시오. 다음 값 중 하나를 선택하십시오.

- 없음
- 확인
- 동기점
- 없음 또는 확인
- 없음, 확인 또는 동기점

PIP 허용

TP가 PIP 데이터(프로그램 초기화 매개변수)를 승인할 경우 이 옵션을 선택하십시오.

모드 및 서비스 클래스 정의

모드는 로컬 LU(LU 유형 6.2)가 해당되는 상대방 LU와 통신하기 위해 사용하는 특성 세트를 지정합니다. 이 특성에는 두 LU 사이에 데이터가 전송되는 방법(예: 최대 RU 길이 및 페이징 창 크기)과 LU가 병렬 세션을 설정할 수 있는지 여부에 대한 정보가 포함됩니다.

또한, LU 사이의 통신 경로에 대한 요구사항(예: 특정 네트워크 보안 레벨 시행, 전송 시간 최소화 또는 비용이 드는 통신 링크 사용 금지)을 지정해야 할 수도 있습니다. 특성(예: 전송 시간, 전송 비용 및 네트워크 보안)에 대한 최소 및 최대 승인 가능 값을 지정하는 서비스 클래스(COS)를 사용하여 이러한 요구사항을 정의할 수 있습니다. COS는 또한 이러한 여러 범위의 값과 연관되는 가중치를 지정할 수 있습니다. 이렇게 하면 노드가 동일 원격 LU에 대해 두 개 이상의 전송 경로가 사용 가능할 경우 네트워크에서의 최상의 전송 경로를 계산할 수 있습니다.

Linux용 Communications Server 노드가 네트워크 노드일 경우, 각 모드의 정의에는 해당 모드에 필요한 COS의 이름이 있습니다. Linux용 Communications Server 노드가 LEN 노드 또는 끝 노드일 경우, COS를 모드와 연관짓지 않아도 됩니다. COS 이름이 동적으로 판별되기 때문입니다.

SNA는 대다수 시스템의 요구사항을 다루는 표준 모드 및 연관된 COS 수를 정의합니다. 사용자는 일반적으로 추가 모드와 COS를 정의하지 않아도 됩니다. 필요한 모드가 사전 정의된 표준 모드(모드 창에서 볼 수 있는) 중 하나가 아닐 경우에만 모드를 정의해야 합니다.

디폴트 모드는 입력 대화에 있는 모드 이름을 인식하지 못할 경우에 사용됩니다. 디폴트 모드를 지정하지 않을 경우, 디폴트 모드는 공백 모드 이름입니다.

표준 모드 이름 및 연관된 COS 이름이 108 페이지의 표 2에 나와 있습니다. 이 표준 이름들과 연관되는 매개변수에 대한 자세한 정보는 IBM SNA 매뉴얼 *LU 6.2 Reference—Peer Protocols*(모드에 대한 경우) 및 *APPN Architecture Reference*(COS에 대한 경우)를 참조하십시오.

모드 및 서비스 클래스 정의

표 2. 표준 모드 및 COS 이름

모드 이름	연관된 COS 이름	목적
(공백)	#CONNECT	모드 이름을 지정하지 않는 세션(기본적인 디폴트 COS 매개변수)
#BATCH	#BATCH	일괄처리 응용프로그램에 사용되는 세션
#BATCHSC	#BATCHSC	최소 레벨의 경로지정 보안 상태로 일괄처리 응용프로그램에 사용되는 세션
#BATCHC	#BATCH	일괄처리 응용프로그램에서 압축을 사용하는 세션
#BATCHCS	#BATCH	최소 레벨의 경로지정 보안 상태로 일괄처리 응용프로그램에서 압축을 사용하는 세션
#INTER	#INTER	대화식 응용프로그램에 사용되는 세션
#INTERSC	#INTERSC	최소 레벨의 경로지정 보안 상태로 대화식 응용프로그램에 사용되는 세션
#INTERC	#INTER	대화식 응용프로그램에서 압축을 사용하는 세션
#INTERCS	#INTER	최소 레벨의 경로지정 보안 상태로 대화식 응용프로그램에서 압축을 사용하는 세션
SNASVCMG	SNASVCMG	CNOS(세션 수 변경) 및 관리 서비스 세션
CPSVCMG	CPSVCMG	노드 사이의 CP-CP 세션
CPSVRMGR	CPSVRMGR	중속 LU 리퀘스터(DLUR)에 사용되는 CP-CP 세션
QPCSUPP	#CONNECT	5250 에뮬레이션에 사용되는 세션

모드가 구성되고 나면, 로컬 LU와 상대방 LU 사이에 세션을 활성화하기 위해 APPC 또는 CPI-C 응용프로그램에서 그 모드를 사용할 수 있습니다. APPC 응용프로그램은 사용할 모드를 지정해야 하지만, CPI-C 응용프로그램은 CPI-C 부가 정보(모드 이름을 포함하는)를 사용할 수 있습니다. CPI-C 부가 정보 구성에 대한 자세한 정보는 112 페이지의 『CPI-C 부가 정보 정의』를 참조하십시오.

모드 또는 서비스 클래스(COS)를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **APPC** 및 **모드**를 선택한 후 **모드** 창에서 새로 작성을 선택하십시오.

명령행 관리 프로그램

모드를 정의하려면 다음 명령을 실행하십시오.

define_mode

디폴트 모드를 변경하려면 다음 명령을 실행하십시오.

define_defaults

서비스 클래스를 정의하려면 다음 명령을 실행하십시오.

define_cos

모드 구성 매개변수

다음 매개변수는 모드 구성에 필요합니다.

이름 정의할 모드의 이름입니다. 모드 이름은 1-8자의 문자열입니다.

이 모드를 사용하는 APPC 응용프로그램(로컬 및 원격 응용프로그램)도 이 이름을 사용할 수 있으므로, 응용프로그램 개발자와 함께 이름을 확인하십시오. (써드-파티 응용프로그램의 경우에는 해당 제품 문서를 참조하십시오.)

COS 이름

모드의 서비스 클래스(COS) 이름입니다. 이름은 1-8자의 문자열입니다. 보통 대화식 데이터 교환에 사용되는 모드의 경우에는 #INTER를, 대량 데이터 전송에 사용되는 모드의 경우에는 #BATCH를 지정할 수 있습니다.

이 필드는 네트워크 노드에만 적용됩니다.

지정할 값을 모르면 SNA 네트워크 계획자에게 문의하십시오.

세션 한계

다음의 필드를 사용하여 세션 한계를 지정하십시오.

초기 세션 한계

CNOS를 사용하여 다른 최대값으로 조정하지 않는 한 해당 모드를 사용하면서 LU 쌍이 가질 수 있는 최대 세션 수입니다(최대 세션 한계까지).

보통 이 필드에는 8 값을 사용하십시오. 확실하지 않으면, SNA 네트워크 계획자나 APPC 응용프로그램 개발자에게 문의하십시오. (써드-파티 응용프로그램의 경우에는 제품 문서를 참조하십시오.)

최대 세션 한계

CNOS 조정과 함께 이 모드를 사용하는 LU 쌍 사이에 허용되는 최대 세션 수입니다(최대 32,767까지).

이 필드는 보통 초기 세션 한계와 같은 값으로 설정됩니다. 확실하지 않으면, SNA 네트워크 계획자나 APPC 응용프로그램 개발자에게 문의하십시오. (써드-파티 응용프로그램의 경우에는 제품 문서를 참조하십시오.)

최소 회선경합 성공 세션 수

Linux용 Communications Server가 회선경합 성공 세션으로 로컬 LU에서 사용하도록 예약해야 하는 세션 수입니다(최대 세션 한계까지).

이 필드는 보통 0으로 설정하면 안전하지만, 확실하지 않을 경우 SNA 네트워크 계획자에게 문의하십시오.

최소 회선경합 성공 세션 수와 최소 회선경합 실패 세션 수의 합은 초기 세션 한계를 초과할 수 없습니다.

최소 회선경합 실패 세션 수

Linux용 Communications Server가 회선경합 실패 세션으로 로컬 LU

에서 사용하도록 예약해야 하는 최소 세션 수입니다. 최소 회선경합 성공 세션 수 필드의 값과 함께, 이 값은 세션 경합을 분석하는 방법을 판별합니다.

이는 보통 0으로 설정하면 안전하지만, 확실하지 않을 경우 SNA 네트워크 계획자에게 문의하십시오.

최소 회선경합 성공 세션 수와 최소 회선경합 실패 세션 수의 합은 초기 세션 한계를 초과할 수 없습니다.

자동 활성 세션 수

로컬 LU와 이 모드를 사용하는 상대방 LU 사이의 세션에 CNOS 조정이 발생한 후 자동으로 활성화되는 세션 수입니다(최소 회선경합 성공 세션 수까지). 이 필드에 값을 지정하면 이 모드를 사용하는 LU가 즉시 대화 할당을 요구하는 TP로부터의 요청에 대한 응답으로 세션을 자동으로 시작할 수 있게 됩니다.

수신 페이싱 창

SNA 페이싱 응답이 전송되기 전에 수신할 수 있는 RU 수를 지정하려면 다음 필드를 사용하십시오.

초기 창 크기

로컬 LU가 원격 LU에 페이싱 응답을 전송하기 전에 수신할 수 있는 요청 단위(RU) 수에 대한 초기 설정. 이는 4로 설정하면 안전합니다. 더 높은 값으로 설정하면 일부 상황에서 성능이 개선될 수 있지만 메모리 사용이 증가할 수도 있습니다.

최대 창 크기

로컬 LU가 원격 LU에 페이싱 응답을 전송하기 전에 수신할 수 있는 최대 요청 단위(RU) 수입니다.

이 값은 선택적입니다. 이 값을 제공하지 않으면, 최대 수신 페이싱 창은 무제한 상태가 됩니다. 값을 제공하면, 그 값을 사용하여 적합 페이싱에 맞게 수신 페이싱 창 크기를 제한하게 됩니다. 적합 페이싱을 사용하지 않을 경우에는 이 값이 무시됩니다.

페이싱 창은 0 - 32767 바이트가 될 수 있습니다. 0 값은 무제한 창을 지정합니다.

인접 노드가 고정 페이싱만 지원할 경우, 이 값들은 고정 페이싱 창 크기를 판별합니다. 그러나 인접 노드는 계속해서 조정을 통해 창 크기를 설정할 수 있습니다. 인접 노드가 적합 페이싱을 사용할 경우, 이 값들은 초기 창 크기를 설정합니다.

시간종료 지정

이 모드를 사용하는 LU 6.2 세션이 시간종료되기 전에 비활성 상태로 있어야

하는 시간(0 - 65535초)을 지정하려면 이 옵션을 선택하십시오. 이 값을 변경하면 이 정의를 사용하여 활성화되는 세션(이미 활성화된 세션은 아님)에만 적용됩니다.

0 값을 사용하면, 세션은 해제되는 대로 시간종료됩니다.

최대 RU 크기 제한

상대방 LU에 전송하기 전에 버퍼를 채우는 데이터 양을 판별하는 최대 RU 크기를 지정하려면 이 옵션을 선택하십시오.

상한은 256 - 62440 바이트가 될 수 있습니다. 상한을 1024바이트로 설정하면 안전합니다. 더 높은 값으로 설정하면 일부 상황에서 성능이 개선될 수 있지만 메모리 사용이 증가할 수도 있습니다.

하한은 0이나, 256과 사용자가 지정하는 상한 사이의 값이 될 수 있습니다.

이 필드의 값이 원격 노드에 대해 정의된 RU 크기와 다를 경우, 해당 노드가 있는 세션에 사용되는 크기를 조정하여 세션에 적절한 RU 크기를 설정할 수 있습니다. 실제 값은 하한 필드보다 낮은 값이 될 수 없습니다.

전송 및 수신 값과 함께 이 수치들을 사용하여 로컬 및 상대방 LU 사이의 세션 레벨 단위 처리량을 조정할 수 있습니다. 사용할 값을 모르면, 디폴트 값으로 시작하여 필요에 따라 단위 처리량이 최대화되도록 조정하십시오.

압축 지원

이 모드를 사용하는 세션에 대해 데이터 압축이 지원되는지 여부를 표시합니다. 이 옵션을 설정하지 않으면 압축은 사용되지 않습니다.

이 옵션을 설정할 경우, 인바운드 데이터 및 아웃바운드 데이터에 사용할 최대 압축 레벨을 지정할 수 있습니다. 이들 옵션은 별도의 옵션이므로, 두 방향에 대해 다른 레벨을 지정하거나, 하나의 방향에서는 압축을 사용하지만 다른 방향에서는 사용하지 않을 수 있습니다. 각 방향에서, 압축을 사용하지 않을 경우 없음을 선택하거나, 값 RLE(최소 압축), LZ9 또는 LZ10(최대 압축) 중 하나를 선택할 수 있습니다.

SNA를 정의된 값으로 재설정

Motif 대화상자를 사용하여 표준 모드를 수정할 경우, 이 버튼을 눌러서 모드 매개변수 값을 SNA 정의 값으로 재설정할 수 있습니다.

추가 구성

모드 구성을 수행하고 나면 다음 구성 태스크로 계속하십시오.

- CPI-C 부가 정보를 정의하려면 112 페이지의 『CPI-C 부가 정보 정의』를 참조하십시오.
- APPC 보안을 정의하려면 114 페이지의 『APPC 보안 구성』을 참조하십시오.

- 5250 통신을 구성하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』을 참조하십시오.

CPI-C 부가 정보 정의

CPI-C 기호식 목적지 이름을 사용하는 CPI-C 응용프로그램을 지원할 경우, CPI-C 부가 정보를 정의해야 합니다. 부가 정보는 기호식 목적지 이름을 대화를 위한 상대방 TP, 상대방 LU, 모드 및 보안에 관한 정보와 연관시킵니다.

CPI-C에 대한 기호식 목적지 이름을 판별하려면, 응용프로그램 개발자에게 문의하십시오. (씨드-파티 응용프로그램의 경우, 제품 문서를 참조하십시오.)

CPI-C 부가 정보를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **APPC**와 **CPI-C**를 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

```
define_cplic_side_info
```

CPI-C 구성 매개변수

응용프로그램에서 사용하는 CPI-C 기호식 목적지 이름마다 다음 정보를 수집하십시오.

이름 실행할 CPI-C 응용프로그램(TP라고도 함)에서 사용하는 기호식 목적지 이름입니다. 이 이름은 1-8자가 될 수 있습니다.

응용프로그램 개발자(씨드-파티 응용프로그램의 경우에는 제품 문서)가 이 이름을 제공할 수 있습니다.

로컬 LU

다음 방법 중 하나를 사용하여 이 부가 정보를 사용하는 TP에 의해 초기화된 대화에 해당되는 로컬 LU입니다.

로컬 LU 별명

로컬 LU의 별명입니다.

디폴트 LU 사용

디폴트 풀(존재할 경우)의 멤버를 사용하거나 노드 제어점 LU(디폴트 풀이 정의되지 않은 경우)를 사용하려면 이 옵션을 지정하십시오.

APPCLLU 환경 변수를 설정하면, 사용자가 제공하는 로컬 LU 정보가 무시되고 환경 변수에 지정된 LU가 대신 사용됩니다.

상대방 LU

이 부가 정보를 사용하는 로컬 TP에 의해 초기화된 대화에 해당되는 전체 상대방 LU 이름이나 별명입니다. 상대방 LU는 상대방 TP를 실행하는 컴퓨터에 구성된 LU여야 합니다.

모드 상대방 LU에 액세스하기 위해 사용할 APPC 모드의 이름입니다. 대부분의 경우, 모드는 다음의 사전 정의된 모드 중 하나입니다.

- 공백 이름
- #BATCH
- #BATCHSC
- #INTER
- #INTERSC
- QPCSUPP

상대방 TP

CPI-C 응용프로그램이 통신하는 트랜잭션 프로그램의 이름입니다.

- TP가 사용자 응용프로그램일 경우, 이름을 정상 문자로 지정하십시오(최대 길이 64자).
- TP가 서비스 TP일 경우, 이름을 16진수로 입력하십시오(4바이트씩 표시하는 최대 8자리 16진수).

응용프로그램 개발자(써드-파티 응용프로그램의 경우에는 제품 문서)가 이 정보를 제공할 수 있습니다.

보안 사용할 대화 레벨 보안의 레벨입니다. 옵션은 다음과 같습니다.

없음 상대방 TP가 보안 매개변수 확인을 요구하지 않습니다.

동일 상대방 TP가 보안을 사용하지만, 초기화 TP에서 제공하는 사용자 ID 및 암호의 로컬 TP에 의한 확인을 승인합니다. 보안 레벨로 동일을 선택할 경우, 상대방 TP가 승인하는 유효한 사용자 ID도 지정해야 합니다.

프로그램

상대방 TP는 사용자 ID 및 암호를 요구합니다. 보안 레벨로 프로그램을 선택할 경우, 상대방 TP가 승인하는 유효한 사용자 ID 및 암호를 지정해야 합니다.

프로그램 강화

상대방 TP는 사용자 ID 및 암호를 요구합니다. 로컬 노드와 원격 노드 둘 다 암호가 암호화되도록 보안 향상을 지원해야 합니다.

CPI-C 응용프로그램 관련 문서를 참조하거나 응용프로그램 프로그래머에게 문의하여 사용할 보안 매개변수를 찾아보십시오.

사용자 ID

보안 레벨로 동일, 프로그램 또는 프로그램 강화를 선택한 경우, 초기화 메시지에서 원격 응용프로그램에 전송할 사용자 ID를 지정하십시오. 이 값은 응용프로그램이 승인하도록 정의한 사용자 ID와 일치해야 합니다.

이 사용자 ID는 로컬 또는 원격 노드에서의 Linux 로그인 사용자 ID와 관련되지 않습니다. 원격 노드가 Linux용 Communications Server를 실행 중인 경우, 대화 보안 구성 대화상자를 사용하여 원격 노드에 사용자 ID를 구성해야 합니다.

암호 보안 레벨을 프로그램 또는 프로그램 강화로 지정할 경우, 대화 할당 시 전송할 암호를 지정하십시오. 이 값은 제공된 사용자 이름으로 사용할 원격 응용프로그램에서 정의한 암호와 일치해야 합니다.

이 암호는 로컬 또는 원격 노드에서의 Linux 로그인 암호와 관련되지 않습니다. 원격 노드가 Linux용 Communications Server를 실행 중인 경우, 대화 보안 구성 대화상자를 사용하여 원격 노드에 암호를 구성해야 합니다.

추가 구성

CPI-C 구성을 수행하고 나면 다음 구성 task로 계속하십시오.

- APPC 보안을 정의하려면 『APPC 보안 구성』을 참조하십시오.
- 5250 통신을 구성하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』을 참조하십시오.

APPC 보안 구성

APPC 보안에 대해 다음 구성 task를 수행할 수 있습니다.

- 『세션 보안 구성』에 설명된 대로 세션 보안 구성
- 115 페이지의 『대화 보안 구성』에 설명된 대로 대화 보안 구성
- 116 페이지의 『보안 액세스 리스트 구성』에 설명된 대로 보안 액세스 리스트 구성

세션 보안 구성

세션 레벨 보안은 LU-LU 세션의 유효성을 확인하기 위해 사용됩니다. 각각의 정의는 로컬 LU 이름, 상대방 LU 이름 및 암호로 구성됩니다.

Linux용 Communications Server는 암호를 사용하여 로컬 LU 및 상대방 LU 사이의 세션 유효성을 확인합니다. (암호는 Linux 로그인 암호와 관련되지 않습니다.)

세션 보안을 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **APPC**, 보안 및 세션 레벨 보안을 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

define_lu_lu_password**세션 보안 구성 매개변수**

다음 매개변수는 세션 보안 구성에 필요합니다.

로컬 LU

로컬 LU의 LU 이름입니다. 이름은 1-8자의 문자열입니다.

상대방 LU

상대방 LU의 전체 LU 이름입니다.

암호 Linux용 Communications Server가 로컬 LU 및 상대방 LU 사이의 세션 유효성을 확인하기 위해 사용하는 암호. 암호는 세션 설정 시 교환되는 키를 작성하기 위해 사용되는 EBCDIC 형식화 문자열(16자리의 16진수로 표시)입니다. 이 암호는 로컬 또는 원격 노드에서의 Linux 로그인 암호와 관련되지 않습니다.

추가 구성

세션 보안 구성을 수행하고 나면 다음 구성 태스크로 계속하십시오.

- 대화 보안을 구성하려면 『대화 보안 구성』을 참조하십시오.
- 5250 통신을 구성하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』을 참조하십시오.

대화 보안 구성

대화 보안은 입력 대화의 유효성을 확인하기 위해 사용됩니다. 각 정의는 사용자 ID 및 암호로 구성됩니다. 이 사용자 ID는 로컬 또는 원격 노드에서의 Linux 로그인 사용자 ID와 관련되지 않습니다.

대화 보안을 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **APPC**, 보안 및 대화 레벨 보안을 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

define_userid_password**대화 보안 구성 매개변수**

다음 매개변수는 대화 보안 구성에 필요합니다.

사용자 ID

원격 노드로부터의 입력 대화에서 승인할 사용자 ID입니다. 사용자 ID는 10자까지 가능합니다.

암호 원격 노드로부터의 입력 대화에서 승인할 암호. 암호는 10자까지 가능합니다.

추가 구성

대화 보안을 구성하고 나면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』에 설명된 대로 5250 통신을 구성할 수 있습니다.

보안 액세스 리스트 구성

APPC 보안 액세스 리스트를 정의하여 LU 또는 TP(또는 둘 다)에 대한 액세스를 제어할 수 있습니다. 이 리스트는 APPC 로컬 LU나 TP의 정의에서 참조할 수 있습니다.

보안 액세스 리스트를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **APPC**, 보안 및 대화 레벨 보안을 선택한 후, 보안 액세스 리스트 분할창을 선택하고 새로 작성을 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

```
define_security_access_list
```

보안 액세스 리스트 구성 매개변수

다음 매개변수는 보안 액세스 리스트 구성에 필요합니다.

이름 보안 액세스 리스트 이름입니다. APPC TP 또는 로컬 LU의 정의에서 이 이름을 사용하여 액세스 리스트를 참조할 수 있습니다.

액세스 리스트 내의 사용자

보안 액세스 리스트에 포함되는 사용자 이름입니다.

추가 구성

보안 액세스 리스트 구성을 수행하고 나면 다음 구성 task로 계속하십시오.

- 99 페이지의 『TP 정의』에 설명된 대로 TP 액세스를 구성하십시오.

제 7 장 사용자 응용프로그램 구성

이 장은 3270, 5250 및 LUA 통신을 사용하는 사용자 응용프로그램을 지원하도록 SNA 자원을 구성하기 위한 지시사항을 제공합니다. 그러한 응용프로그램에서 요구하는 SNA 자원은 LU입니다.

3270, LUA 및 종속 APPC 통신의 경우, 종속 LU를 구성해야 합니다. 독립 APPC 및 5250 통신의 경우, 디폴트 제어점 LU(로컬 노드를 구성할 때 자동으로 정의되는)를 사용하거나 독립 LU를 정의할 수 있습니다.

이 장에 설명된 자원을 구성하려면 먼저 다음 구성을 수행해야 합니다.

- 62 페이지의 『노드 구성』에 설명된 대로 노드를 구성하십시오.
- 67 페이지의 제 4 장 『연결 구성요소 정의』에 설명된 대로 연결성을 구성하십시오. 3270, LUA 및 독립 APPC 통신의 경우, 종속 LU 트래픽을 지원하도록 링크를 구성해야 합니다. 독립 APPC 및 5250 통신의 경우, 링크는 독립 LU 트래픽을 지원해야 합니다.

업스트림 SNA 게이트웨이나 DLUR을 사용할 경우에는 호스트에 대한 직접 링크를 구성하지 않아도 됩니다. 자세한 정보는 129 페이지의 『SNA 게이트웨이 구성』 및 82 페이지의 『DLUR PU 정의』를 참조하십시오.

다음 리스트는 각 유형의 사용자 응용프로그램에 필요한 구성 작업을 설명합니다.

3270 응용프로그램

3270 통신의 경우, 다음 자원을 구성하십시오.

1. 3270 표시장치 또는 프린터의 경우, 85 페이지의 『LU 유형 0-3 정의』에 설명된 대로 종속 LU를 정의하십시오.
2. 3270 표시장치를 사용 가능하도록 하여 LU 풀에서 선택하려면, 88 페이지의 『LU 풀 정의』에 설명된 대로 LU 풀을 정의하십시오. 표시장치가 전용 LU를 사용하면 이 단계를 건너뛸 수 있습니다.

5250 응용프로그램

5250 통신의 경우, 다음 자원을 구성하십시오.

1. APPC 통신용 노드를 구성하십시오.
 - a. 로컬 노드의 제어점 LU를 사용할 수 있으면, 로컬 LU를 구성하지 않아도 됩니다. 로컬 LU 정의가 필요할 경우(예: 세션 보안을 사용하기 위해), 92 페이지의 『로컬 LU 정의』에 설명된 대로 로컬 LU를 정의하십시오.

- b. 로컬 노드가 LEN 노드이면 94 페이지의 『원격 노드 정의』에 설명된 대로 AS/400 시스템을 원격 노드로 정의해야 합니다.

로컬 노드가 APPN 끝 노드이거나 네트워크 노드일 경우, AS/400 시스템에서 제어점(CP) LU를 상대방 LU로 사용할 수 있으므로 다른 상대방 LU를 구성하지 않아도 됩니다.

5250은 표준 모드 QPCSUPP를 사용하므로 모드를 정의하지 않아도 됩니다.

LUA 응용프로그램

LUA 응용프로그램을 지원하려면 다음 자원을 구성하십시오.

1. 85 페이지의 『LU 유형 0-3 정의』에 설명된 종속 LU를 정의하십시오.
2. LUA 응용프로그램을 사용 가능하도록 하여 LU 풀에서 선택하려면, 88 페이지의 『LU 풀 정의』에 설명된 대로 LU 풀을 정의하십시오. 응용프로그램이 전용 LU를 사용하면 이 단계를 건너뛸 수 있습니다.

LUA 응용프로그램은 노드의 LU 0-3 자원을 사용하여 호스트 응용프로그램과 통신합니다. 추가 자원은 정의하지 않아도 됩니다.

제 8 장 패스-스루 서비스 구성

Linux용 Communications Server를 실행 중인 서버의 패스-스루 서비스는 직접 호스트에 연결되어 있지 않은 로컬 시스템과 SNA 호스트 사이의 통신을 사용 가능하게 합니다.

Linux용 Communications Server에는 TN3270, TN3287 및 TN3270E 클라이언트(총체적으로 『TN3270 클라이언트』라고 함)에 대한 TN 서버 지원이 포함되어 있습니다. 이 기능을 구성하려면 『TN 서버 구성』을 참조하십시오.

Linux용 Communications Server에는 또한 TN3270, TN3270E, TN5250 및 VT 클라이언트(총체적으로 『텔넷 클라이언트』라고 함)에 대한 패스-스루 TCP/IP 호스트 액세스를 위한 TN 방향 재지정자 지원도 포함되어 있습니다. 이 기능을 구성하려면 124 페이지의 『TN 방향 재지정자 구성』을 참조하십시오.

SNA 게이트웨이는 호스트와 로컬 시스템 사이의 연결을 제공합니다. 로컬 노드에 LU를 구성하여 이 기능을 지원하거나(129 페이지의 『SNA 게이트웨이 구성』 참조) 명시적으로 구성되지 않은 다운스트림 LU를 지원하기 위해 사용되는 템플릿을 정의할 수 있습니다(68 페이지의 『DLC, 포트 및 연결 네트워크 정의』 참조).

DLUR은 APPN 네트워크에서 호스트와 노드 사이의 종속 LU 세션을 지원합니다. 이 기능을 구성하려면 131 페이지의 『DLUR 구성』을 참조하십시오.

TN 서버 구성

TN 서버는 TN3270 클라이언트가 TN 서버를 구현하는 중간 Linux용 Communications Server 노드를 통해 호스트와 통신할 수 있게 합니다. TN3270 클라이언트는 TCP/IP를 사용하여 TN 서버에 연결하고, TN 서버에 정의된 LU를 사용합니다. TN 서버 LU는 호스트에 있는 LU로 세션을 설정하여 클라이언트용 TN3270 세션을 지원합니다.

TN 서버를 구성하려면 먼저 다음 구성 작업을 수행해야 합니다.

- 62 페이지의 『노드 구성』에 설명된 대로 로컬 노드를 정의하십시오.
- 67 페이지의 제 4 장 『연결 구성요소 정의』에 설명된 대로 로컬 노드와 호스트 사이의 종속 트래픽을 위한 포트 및 링크 스테이션을 구성하십시오.
- 호스트와의 통신에 사용되는 TN3270 LU를 로컬 노드에 정의하십시오. LU를 추가하려면 85 페이지의 『LU 유형 0-3 정의』를 참조하십시오.
- LU 풀을 사용하려는 경우 88 페이지의 『LU 풀 정의』에 설명된 대로 LU 풀을 정의하십시오.

TN 서버를 구성하려면 다음 작업을 수행하십시오.

- 서버를 사용할 TN3270 클라이언트마다 TN 서버 액세스 레코드를 구성하거나, 클라이언트가 서버에 액세스할 수 있도록 하는 디폴트 레코드를 구성하십시오(『TN 서버 액세스 레코드 구성』 참조).
- TN3270E 또는 TN3287 클라이언트를 지원할 경우, 표시장치 및 프린터 LU의 연결 레코드를 정의할 수 있습니다(124 페이지의 『TN 서버 연결 레코드 구성』 참조). 이 레코드로 TN3270E 또는 TN3287 클라이언트는 특정 프린터를 선택할 수 있습니다(연관된 표시장치 LU를 선택하여). 클라이언트는 TN 서버 액세스 레코드에서 LU를 선택할 수 있는 권한이 있어야 합니다.

TN 서버의 추가 옵션을 사용하면 프린터 응답을 강요하고, 모든 TN3270 세션에 활성화 지속 방법을 지정하고, TN3270 클라이언트에 대한 권한을 확인하기 위해 사용되는 취소 리스트를 보유하는 외부 LDAP 서버에 액세스하는 방법을 지정할 수 있습니다. 이 옵션에 액세스하려면, TN 서버 창에서 서비스 메뉴를 사용하십시오.

TN 서버 액세스 레코드 구성

TN 서버 액세스 레코드는 TN 서버에 액세스할 수 있는 TN3270 클라이언트와 이 클라이언트가 사용해야 하는 LU를 표시합니다. 각 액세스 레코드는 TN 서버에 대한 액세스가 허용되는 TN3270 클라이언트, 클라이언트가 연결하는 TCP/IP 포트, 클라이언트가 사용하는 LU 또는 LU 풀을 식별합니다.

모든 TN3270 클라이언트가 액세스할 수 있도록 하는 디폴트 레코드를 정의할 수도 있습니다(모든 클라이언트에 대해 같은 LU 또는 LU 풀 사용).

TN3270 클라이언트는 노드, 포트 및 링크 스테이션이 활동 중인 경우에만 TN 서버를 사용할 수 있습니다.

TN 서버 액세스 레코드를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **TN** 서버를 선택하고 하위 메뉴에서 **TN** 서버를 선택하십시오. 결과 창에서, TN 서버 클라이언트 액세스 권한 분할창을 선택한 후 새로 작성을 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

```
define_tn3270_access
```

주: 명령행 관리 프로그램, **snaadmin** 또는 NOF 응용프로그램을 사용하여 TN 서버 액세스 레코드를 정의하는 경우 *listen_local_address* 매개변수를 사용하여 로컬 TN 서버 컴퓨터에 TN3270 클라이언트가 연결될 주소를 지정할 수 있습니다. 이렇게 하면 Motif 관리 프로그램에 액세스 레코드가 표시되지 않아 해당 프로그램을 사

용하여 액세스 레코드를 보거나 관리할 수 없습니다. 명령행 관리 프로그램 또는 NOF 응용프로그램을 사용하여 액세스 레코드를 관리할 수는 있습니다.

TN 서버 액세스 레코드 구성 매개변수

다음 매개변수는 TN 서버 액세스 레코드 구성에 필요합니다.

TN3270 클라이언트 주소

액세스 레코드를 적용할 TN3270 클라이언트를 식별하는 주소입니다.

디폴트 레코드

모든 TN3270 클라이언트의 액세스를 허용합니다.

TCP/IP 이름 또는 별명

이름 지정된 TN3270 클라이언트의 액세스를 허용합니다. 클라이언트의 TCP/IP 이름을 알고 있으면, 이 옵션을 선택한 후 이름을 입력하십시오. 많은 컴퓨터에서, **hostname** 명령을 사용하여 컴퓨터의 TCP/IP 이름을 찾을 수 있습니다.

TCP/IP 주소

특정 TCP/IP 주소에서의 액세스를 허용합니다. TN3270 클라이언트의 TCP/IP 주소를 알고 있으면, 이 옵션을 선택한 후 주소를 입력하십시오. 다음 중 하나를 입력할 수 있습니다.

- IPv4 점으로 구분된 10진수 주소(예: 193.1.11.100)
- IPv6 콜론으로 구분된 16진수 주소(예:
2001:0db8:0000:0000:0000:0000:1428:57ab 또는
2001:db8::1428:57ab)

TN3270E 지원

노드에서 제공하는 TN3270 지원 레벨입니다.

TN3270

TN3270 프로토콜만 지원합니다. 이 옵션을 선택하면 TN3270E 프로토콜에 대한 서버 지원이 클라이언트에서 지원되더라도 사용 불가능하게 됩니다.

TN3270E

TN3270 및 TN3270E 프로토콜 둘 다를 지원합니다(디폴트).

선택하는 옵션에 관계없이 TN3270 및 TN3287 프로토콜은 항상 지원됩니다. AS/400 TN3270 클라이언트의 경우, 이 옵션은 TN3270E로 설정해야 합니다.

TCP/IP 포트 번호

TN3270 클라이언트가 연결하는 포트의 TCP/IP 포트 번호(TN 서버에서).

주: TCP/IP 포트는 SNA 포트에 관련되어 있지 않습니다.

TN3270 서비스에 자주 사용되는 포트 번호는 23이지만, TN 서버에서 사용하고 있지 않는 다른 포트 번호를 선택하면, TN3270 클라이언트에서 이 포트 번호도 구성해야 합니다. (또는 포트 번호를 지정하기 위한 옵션을 사용하여 TN3270 클라이언트를 시작합니다.) 2000 위의 포트 번호를 사용할 수 있을 것입니다. 범위 256-1023 내의 포트 번호는 약간 더 나은 보안을 제공할 수 있지만 사용 중일 가능성이 높습니다.

TN3270 클라이언트가 두 개 이상의 LU 또는 LU 풀을 사용할 수 있도록 하려면, 다른 포트 번호를 지정하여 다른 LU 또는 LU 풀을 식별할 수 있도록 각각 다른 TCP/IP 포트 번호를 사용하는 여러 개의 액세스 레코드를 정의하십시오.

지정된 표시장치 LU

TN3270 클라이언트가 활동 중일 때 액세스하는 LU의 이름입니다. LU는 로컬 노드에서 종속 LU여야 합니다. 특정 LU의 이름이 아닌 LU 풀 이름을 지정할 수 있습니다.

지정된 프린터 LU

액세스 레코드를 사용하는 클라이언트의 디폴트 프린터 LU 또는 LU 풀 이름입니다. 이 LU는 로컬 노드에서 종속 LU로 정의되어 있어야 합니다.

특정 LU로의 액세스 허용

TN3270E 및 TN3287 클라이언트가 세션에 대해 특정 LU를 요청할 수 있도록 하려면 이 옵션을 지정하십시오. (이 옵션은 TN3270 클라이언트에 사용할 수 없습니다.)

SSL 보안 세션

세션이 서버에 액세스하기 위해 SSL을 사용함을 표시하려면 이 옵션을 지정하십시오.

이 옵션은 서버에서 SSL을 지원하는 데 필요한 추가 소프트웨어를 설치한 경우에만 사용할 수 있습니다. 그렇지 않은 경우에는 선택할 수 없습니다.

주: 이 세션의 TCP/IP 포트 번호 매개변수가 세션이 텔넷 디먼의 TCP/IP 포트를 사용함을 표시할 경우, 이 세션에 SSL을 사용하지 마십시오. 텔넷 디먼의 TCP/IP 포트를 사용하는 세션에서 SSL을 사용할 경우, 텔넷 클라이언트는 노드가 활동 중일 때 텔넷을 사용하여 Linux용 Communications Server 컴퓨터에 액세스할 수 없습니다.

클라이언트 인증 수행

이 옵션은 SSL 보안 세션 옵션을 선택한 경우에만 표시됩니다.

TN 서버에서 세션이 클라이언트 인증을 사용해야 함을 표시하려면 이 옵션을 지정하십시오. 클라이언트는 유효한 인증(TN 서버를 사용할 수 있는 권한이 부여된 유효한 클라이언트로 식별하는 정보)을 전송해야 합니다.

인증이 유효한지 확인하는 것 외에도, TN 서버는 외부 LDAP 서버에서 인증 취소 목록에 대해 인증을 확인하여 사용자 권한이 취소되지 않았는지 확인해야 할 수도 있습니다. 이 경우, TN 서버 고급 매개변수 대화상자를 사용하여 이 서버에 액세스하는 방법을 지정해야 할 수도 있습니다.

보안 레벨

이 세션에 필요한 SSL 보안 레벨을 표시합니다. 세션은 클라이언트 및 서버 모두 지원할 수 있는 최상위 보안 레벨을 사용합니다. 클라이언트가 요청된 보안 레벨이나 그 이상을 지원할 수 없으면, 세션이 시작되지 않습니다.

이 옵션은 SSL 보안 세션 옵션을 선택한 경우에만 표시됩니다.

가능한 값은 다음과 같습니다.

인증만 인증을 교환해야 하지만 암호화는 사용되지 않습니다. 이 옵션은 보통 클라이언트가 보안 인트라넷을 거쳐 연결할 때 암호화 오버헤드를 피하기 위해 사용됩니다.

최소 인증

클라이언트는 서버로부터의 인증을 요청하여 유효성을 확인해야 합니다. 암호화는 필요하지 않습니다. (그러나 클라이언트가 요청할 경우에는 사용할 수 있습니다.)

최소 40비트

클라이언트는 최소 40비트 암호화를 지원해야 합니다.

최소 56비트

클라이언트는 최소 56비트 암호화를 지원해야 합니다.

최소 128비트

클라이언트는 최소 128비트 암호화를 지원해야 합니다.

최소 168비트

클라이언트는 최소 168비트 암호화를 지원해야 합니다.

주: 암호화를 사용하려면 Linux용 Communications Server와 함께 추가 소프트웨어를 설치해야 합니다. 자세한 정보는 *Linux용 IBM Communications Server 빠른 시작*을 참조하십시오. 암호화 레벨을 지원하기 위해 필요한 소프트웨어를 사용자 국가에서 사용하지 못할 수도 있으므로, 위치에 따라 나열되는 모든 암호화 레벨을 사용하지 못할 수도 있습니다.

추가 구성

TN 서버 액세스 구성을 수행하고 나면 다음 구성 태스크로 계속하십시오.

- 124 페이지의 『TN 서버 연결 레코드 구성』에 설명된 대로 TN 서버 연결 레코드를 구성하십시오.

TN 서버 연결 레코드 구성

TN 서버 연결 레코드는 TN3270E 또는 TN3287 프로토콜이 두 개를 연결할 수 있도록 프린터 LU와 표시장치 LU 사이의 연결을 정의합니다. 클라이언트의 액세스 레코드가 특정 LU 선택을 허용하면, 이 레코드는 클라이언트가 연관된 표시장치 LU를 지정하여 특정 프린터를 선택할 수 있도록 합니다.

TN 서버 연결 레코드를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **TN 서버**를 선택한 후, TN 서버 창에서 연결 레코드 분할창을 선택하고 **새로 작성**을 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

```
define_tn3270_association
```

TN 서버 연결 레코드 구성 매개변수

다음 매개변수는 TN 서버 연결 레코드 구성에 필요합니다.

표시장치 LU

표시장치 LU의 이름(로컬 노드에 정의해야 하는).

프린터 LU

프린터 LU의 이름(로컬 노드에 정의해야 하는). 다른 TN 서버 연결 레코드에 입력한 프린터 LU는 지정하지 마십시오.

TN 방향 재지정자 구성

TN 방향 재지정자는 TN3270, TN3270E, TN5250 및 VT 클라이언트(총체적으로 텔넷 클라이언트라고 함)가 TN 방향 재지정자를 구현하는 중간 Linux용 Communications Server 노드를 통해 호스트와 통신할 수 있게 합니다. 클라이언트는 TCP/IP를 사용하여 TN 방향 재지정자에 연결합니다. 그러면 TN 방향 재지정자가 호스트에 대해 별도의 TCP/IP 연결을 설정합니다.

TN 방향 재지정자를 구성하려면 다음 작업을 수행하십시오.

- 서버를 사용할 텔넷 클라이언트마다 TN 방향 재지정자 액세스 레코드를 구성하거나, 클라이언트가 서버에 액세스할 수 있도록 하는 디폴트 레코드를 구성하십시오(『TN 방향 재지정자 액세스 레코드 구성』 참조).

TN 방향 재지정자 액세스 레코드 구성

TN 방향 재지정자 액세스 레코드는 TCP/IP 링크를 거쳐 TN 방향 재지정자에 액세스할 수 있는 텔넷 클라이언트를 표시합니다. 각 액세스 레코드는 TN 방향 재지정자에 대한 액세스가 허용되는 텔넷 클라이언트, 클라이언트가 Linux용 Communications

Server에 연결하기 위해 사용하는 TCP/IP 포트, Linux용 Communications Server가 호스트에 연결하기 위해 사용하는 TCP/IP 포트, SSL 보안 설정을 식별합니다. 어떤 클라이언트도 액세스할 수 있는 디폴트 레코드를 정의할 수도 있습니다.

어떤 클라이언트도 TN 방향 재지정자를 사용할 수 있도록 허용하고 모든 클라이언트가 같은 호스트 액세스 구성을 사용하도록 하려면 디폴트 레코드를 정의하면 됩니다.

텔넷 클라이언트는 노드가 활동 중일 경우에만 TN 방향 재지정자를 사용할 수 있습니다.

TN 방향 재지정자 액세스 레코드를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **TN** 서버를 선택하고 하위 메뉴에서 **TN** 서버를 선택하십시오. 결과 창에서, TN 방향 재지정자 클라이언트 액세스 권한 분할창을 선택한 후 새로 작성을 선택하십시오.

명령행 관리 프로그램

다음 명령을 실행하십시오.

`define_tn_redirect`

주: 명령행 관리 프로그램, **snaadmin** 또는 NOF 응용프로그램을 사용하여 TN 방향 재지정자 액세스 레코드를 정의하는 경우 `listen_local_address` 매개변수를 사용하여 로컬 TN 서버 컴퓨터에 TN3270 클라이언트가 연결될 주소를 지정할 수 있습니다. 이렇게 하면 Motif 관리 프로그램에 액세스 레코드가 표시되지 않아 해당 프로그램을 사용하여 액세스 레코드를 보거나 관리할 수 없습니다. 명령행 관리 프로그램 또는 NOF 응용프로그램을 사용하여 액세스 레코드를 관리할 수는 있습니다.

TN 방향 재지정자 액세스 레코드 구성 매개변수

TN 방향 재지정자 액세스 레코드 구성은 클라이언트 및 호스트 TCP/IP 연결에 해당되는 두 그룹의 매개변수로 구성됩니다.

클라이언트 매개변수는 다음과 같습니다.

텔넷 클라이언트 주소

액세스 레코드를 적용할 텔넷 클라이언트를 식별하는 주소입니다.

디폴트 레코드

모든 텔넷 클라이언트의 액세스를 허용합니다.

TCP/IP 이름 또는 별명

이름 지정된 텔넷 클라이언트의 액세스를 허용합니다. 클라이언트의 TCP/IP 이름을 알고 있으면, 이 옵션을 선택한 후 이름을 입력하십시오. 많은 컴퓨터에서, **hostname** 명령을 사용하여 컴퓨터의 TCP/IP 이름을 찾을 수 있습니다.

TCP/IP 주소

특정 TCP/IP 주소에서의 액세스를 허용합니다. 클라이언트의 TCP/IP 주소를 알고 있으면, 이 옵션을 선택한 후 주소를 입력하십시오. 다음 중 하나를 입력할 수 있습니다.

- IPv4 점으로 구분된 10진수 주소(예: 193.1.11.100)
- IPv6 콜론으로 구분된 16진수 주소(예:
2001:0db8:0000:0000:0000:0000:1428:57ab 또는
2001:db8::1428:57ab)

TCP/IP 포트 번호

클라이언트가 연결하는 포트의 TCP/IP 포트 번호입니다(TN 서버에서).

주: TCP/IP 포트는 SNA 포트에 관련되어 있지 않습니다.

클라이언트에서 이 포트 번호를 구성해야 할 수도 있습니다. (또는 포트 번호를 지정하기 위한 옵션을 사용하여 클라이언트를 시작합니다.) 2000 위의 포트 번호를 사용할 수 있을 것입니다. 범위 256-1023 내의 포트 번호는 약간 더 나은 보안을 제공할 수 있지만 사용 중일 가능성이 높습니다.

SSL 보안 세션

세션이 서버에 액세스하기 위해 SSL을 사용함을 표시하려면 이 옵션을 지정하십시오.

이 옵션은 서버에서 SSL을 지원하는 데 필요한 추가 소프트웨어를 설치한 경우에만 사용할 수 있습니다. 그렇지 않은 경우에는 선택할 수 없습니다.

클라이언트 인증 수행

이 옵션은 SSL 보안 세션 옵션을 선택한 경우에만 표시됩니다.

TN 서버에서 세션이 클라이언트 인증을 사용해야 함을 표시하려면 이 옵션을 지정하십시오. 클라이언트는 유효한 인증(TN 서버를 사용할 수 있는 권한이 부여된 유효한 클라이언트로 식별하는 정보)을 전송해야 합니다.

인증이 유효한지 확인하는 것 외에도, TN 방향 재지정자는 외부 LDAP 서버에서 인증 취소 목록에 대해 인증을 확인하여 사용자 권한이 취소되지 않았는지 확인해야 할 수도 있습니다. 이 경우, TN 서버 고급 매개변수 대화상자(TN 서버 창의 서비스 메뉴에서 액세스)를 사용하여 이 서버에 액세스하는 방법을 지정해야 할 수도 있습니다.

보안 레벨

클라이언트 세션에 필요한 SSL 보안 레벨을 표시합니다. 세션은 클라이언트 및 서버 모두 지원할 수 있는 최상위 보안 레벨을 사용합니다. 클라이언트가 요청된 보안 레벨이나 그 이상을 지원할 수 없으면, 세션이 시작되지 않습니다.

이 옵션은 SSL 보안 세션 옵션을 선택한 경우에만 표시됩니다.

가능한 값은 다음과 같습니다.

인증만 인증을 교환해야 하지만 암호화는 사용되지 않습니다. 이 옵션은 보통 클라이언트가 보안 인트라넷을 거쳐 연결할 때 암호화 오버헤드를 피하기 위해 사용됩니다.

최소 인증

클라이언트는 서버로부터의 인증을 요청하여 유효성을 확인해야 합니다. 암호화는 필요하지 않습니다. (그러나 클라이언트가 요청할 경우에는 사용할 수 있습니다.)

최소 40비트

클라이언트는 최소 40비트 암호화를 지원해야 합니다.

최소 56비트

클라이언트는 최소 56비트 암호화를 지원해야 합니다.

최소 128비트

클라이언트는 최소 128비트 암호화를 지원해야 합니다.

최소 168비트

클라이언트는 최소 168비트 암호화를 지원해야 합니다.

주: 암호화를 사용하려면 Linux용 Communications Server와 함께 추가 소프트웨어를 설치해야 합니다. 자세한 정보는 *Linux용 IBM Communications Server 빠른 시작*을 참조하십시오. 암호화 레벨을 지원하기 위해 필요한 소프트웨어를 사용자 국가에서 사용하지 못할 수도 있으므로, 위치에 따라 나열되는 모든 암호화 레벨을 사용하지 못할 수도 있습니다.

목적지 호스트 매개변수는 다음과 같습니다.

주소 액세스 레코드를 적용할 호스트를 식별하는 주소입니다.

TCP/IP 이름 또는 별명

이름 지정된 호스트에 액세스합니다. 호스트의 TCP/IP 이름을 알고 있으면, 이 옵션을 선택한 후 이름을 입력하십시오. 많은 컴퓨터에서, **hostname** 명령을 사용하여 컴퓨터의 TCP/IP 이름을 찾을 수 있습니다.

TCP/IP 주소

특정 TCP/IP 주소에 액세스합니다. 호스트의 TCP/IP 주소를 알고 있으면, 이 옵션을 선택한 후 주소를 입력하십시오. 다음 중 하나를 입력할 수 있습니다.

- IPv4 점으로 구분된 10진수 주소(예: 193.1.11.100)
- IPv6 콜론으로 구분된 16진수 주소(예: 2001:0db8:0000:0000:0000:0000:1428:57ab 또는 2001:db8::1428:57ab)

TCP/IP 포트 번호

TN 방향 재지정자가 호스트에 액세스하기 위해 사용하는 TCP/IP 포트 번호입니다.

주: TCP/IP 포트는 SNA 포트에 관련되어 있지 않습니다.

호스트에서 이 포트 번호를 구성해야 할 수도 있습니다. 2000 위의 포트 번호를 사용할 수 있을 것입니다. 범위 256-1023 내의 포트 번호는 약간 더 나은 보안을 제공할 수 있지만 사용 중일 가능성이 높습니다.

SSL 보안 세션

TN 방향 재지정자가 호스트에 액세스하기 위해 SSL을 사용함을 표시하려면 이 옵션을 지정하십시오.

이 옵션은 호스트가 SSL을 지원할 경우에만 사용할 수 있습니다.

보안 레벨

호스트 세션에 필요한 SSL 보안 레벨을 표시합니다. 세션은 호스트 및 서버 모두 지원할 수 있는 최상위 보안 레벨을 사용합니다. 호스트가 요청된 보안 레벨이나 그 이상을 지원할 수 없으면, 세션이 시작되지 않습니다.

이 옵션은 SSL 보안 세션 옵션을 선택한 경우에만 표시됩니다.

가능한 값은 다음과 같습니다.

인증만 인증을 교환해야 하지만 암호화는 사용되지 않습니다. 이 옵션은 보통 호스트 연결이 보안 인트라넷을 거쳐갈 경우 암호화 오버헤드를 피하기 위해 사용됩니다.

최소 인증

호스트는 서버로부터의 인증을 요청하여 유효성을 확인해야 합니다. 암호화는 필요하지 않습니다. (그러나 호스트가 요청할 경우에는 사용할 수 있습니다.)

최소 40비트

호스트는 최소 40비트 암호화를 지원해야 합니다.

최소 56비트

호스트는 최소 56비트 암호화를 지원해야 합니다.

최소 128비트

호스트는 최소 128비트 암호화를 지원해야 합니다.

최소 168비트

호스트는 최소 168비트 암호화를 지원해야 합니다.

주: 암호화를 사용하려면 Linux용 Communications Server와 함께 추가 소프트웨어를 설치해야 합니다. 자세한 정보는 *Linux용 IBM Communications Server* 빠른 시작을 참조하십시오. 암호화 레벨을 지원하기 위해 필요한 소

소프트웨어를 사용자 국가에서 사용하지 못할 수도 있으므로, 위치에 따라 나열되는 모든 암호화 레벨을 사용하지 못할 수도 있습니다.

SNA 게이트웨이 구성

보통, 종속 LU 세션은 호스트 컴퓨터로의 직접 통신 링크를 필요로 합니다. 그러나 호스트와의 직접 통신 링크를 가지고 있고 Linux용 Communications Server를 실행 중인 노드도 다운스트림 컴퓨터의 LU에 대해 SNA 게이트웨이 기능을 제공하여 LU가 Linux용 Communications Server 노드로부터 통신 링크를 거쳐 호스트에 액세스할 수 있도록 합니다. 다운스트림 컴퓨터는 호스트와의 종속 통신을 지원하기 위해 SNA PU 유형 2.0 또는 2.1을 포함하고 있어야 합니다. 예를 들어, 다운스트림 컴퓨터는 독립형 구성에서 Linux용 Communications Server를 실행하는 또 다른 컴퓨터일 수 있습니다.

SNA 게이트웨이 기능을 사용할 경우, 호스트와 다운스트림 컴퓨터 사이에 전송되는 모든 데이터가 Linux용 Communications Server 로컬 노드를 통해 전송됩니다. 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 SDLC 전용 회선에 액세스할 수 있습니다.

SNA 게이트웨이를 사용하면 호스트에서 구성을 단순화할 수도 있습니다. 호스트 구성에는 Linux용 Communications Server 컴퓨터 및 해당 호스트 통신 링크만이 포함되어야 합니다. 다운스트림 컴퓨터의 LU는 Linux용 Communications Server 컴퓨터 자원의 일부로 구성됩니다. 호스트 컴퓨터는 SNA 게이트웨이가 사용되고 있음을 인식하지 못합니다.

SNA 게이트웨이를 구성하기 전에 다음 구성 작업을 수행해야 합니다.

- 62 페이지의 『노드 구성』에 설명된 대로 로컬 노드를 정의하십시오.
- 67 페이지의 제 4 장 『연결 구성요소 정의』에 설명된 대로 로컬 노드와 호스트 사이의 종속 트래픽을 위한 포트 및 링크 스테이션을 구성하십시오. 또한 로컬 노드와 다운스트림 노드 사이의 종속 트래픽에 대해 포트 및 링크 스테이션을 구성하십시오. 다운스트림 링크의 경우, 포트에서 암시적 다운스트림 LU(로컬 노드에 명시적으로 정의되지 않은 LU)를 지원하도록 템플릿을 구성해야 합니다.
- 호스트(업스트림 LU)와의 통신에 사용되는 로컬 노드에서 LU를 정의하십시오. 종속 LU 6.2 LU를 포함하여 업스트림 LU는 LU 유형 0-3 구성 대화상자를 사용하여 LU 유형 무제한(알 수 없음)을 지정하여 정의해야 합니다. LU를 추가하려면 85 페이지의 『LU 유형 0-3 정의』를 참조하십시오.

- LU 풀을 사용하려는 경우 88 페이지의 『LU 풀 정의』에 설명된 대로 LU 풀을 정의하십시오

SNA 게이트웨이를 사용 가능하게 하려면, 로컬 노드에서 다운스트림 워크스테이션이 있는 세션을 지원하도록 LU를 구성해야 합니다. (암시적 다운스트림 LU를 지원하도록 포트에서 템플릿을 구성한 경우에는 다운스트림 LU를 명시적으로 정의하지 않아도 됩니다.) 로컬 노드에 정의된 LU를 『다운스트림 LU』라고 합니다. 다운스트림 LU를 구성하려면, 다운스트림 노드에서 사용되는 LU 번호와 호스트 LU의 이름이 필요합니다. (다운스트림 노드에 정의된 LU는 종속 LU 유형이 될 수 있습니다.)

다운스트림 LU를 구성하려면 다음 방법 중 하나를 사용하십시오.

Motif 관리 프로그램

노드 창의 서비스 메뉴에서 **SNA 게이트웨이**와 새 **다운스트림 LU**를 선택하십시오.

명령행 관리 프로그램

다음 명령 중 하나를 실행하십시오.

```
define_downstream_lu
```

```
define_downstream_lu_range
```

다운스트림 LU 구성 매개변수

다음 매개변수는 다운스트림 LU 구성에 필요합니다.

다운스트림 LU 이름

각 다운스트림 LU의 이름입니다. LU 이름은 LU를 로컬로 식별할 경우에만 사용되므로, 다운스트림 노드의 구성과 일치시킬 필요는 없습니다.

LU의 범위를 정의할 경우 1-5자의 기본 이름을 지정하십시오. Linux용 Communications Server는 세 자리의 십진 문자열을 기본 이름에 추가하여 사용자가 지정하는 LU 번호마다 LU 이름을 작성합니다.

다운스트림 PU 이름

다운스트림 노드에 대한 링크 스테이션의 이름입니다.

LU 번호

LU 번호는 다운스트림 노드에 정의된 LU 번호와 일치해야 합니다. 사용할 LU 번호를 모르면 SNA 네트워크 계획자에게 문의하십시오.

LU의 범위를 정의하여 연속 LU 번호를 사용하는 몇 개의 LU를 구성할 수 있습니다.

업스트림 LU 이름

다운스트림 LU가 통신할 호스트 LU 또는 LU 풀의 이름입니다.

지연된 로그인

사용자 시작 시간을 줄이기 위해, Linux용 Communications Server는 업스트림 LU를 지정하지 않고 로그인 화면을 표시합니다. 3270 사용자는 사용자가 업스트림 LU와 연관되기 전에 키를 눌러야 합니다.

시간종료 허용

필요한 LU 수를 줄이기 위해, 활동 중인 PLU-SLU 세션이 없는 LU는 이 시간(초)이 지난 후 업스트림 LU와의 연관이 해제됩니다.

추가 구성

SNA 게이트웨이의 다운스트림 LU 구성을 수행하고 나면 다음 구성 태스크로 계속하십시오.

- 사용자 응용프로그램을 지원하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』을 참조하십시오.

DLUR 구성

보통, 종속 LU 세션은 호스트 컴퓨터로의 직접 통신 링크를 필요로 합니다. 많은 노드(호스트 노드 포함)가 APPN 네트워크에서 함께 연결된 경우, 일부 노드가 호스트와의 직접 연결을 가지고 있지 않고 다른 노드를 통한 간접 연결만 가지고 있을 수도 있습니다. 이러한 직접 연결이 없고 간접적으로 연결된 노드에서는 LU에서 호스트로의 종속 LU 세션을 설정할 수 없습니다.

종속 LU 리퀘스터(DLUR)는 이러한 제한을 극복하기 위해 설계된 APPN 기능입니다. DLUR은 APPN 노드(예: Linux용 Communications Server를 실행 중인 노드)에서 구성할 수 있습니다. DLUR은 호스트에 있는 종속 LU 서버(DLUS)와 함께 작동하여, APPN 네트워크를 통해 DLUR 노드의 종속 LU에서 DLUS 호스트로 세션 경로를 지정합니다.

호스트로의 전송 경로는 여러 노드에 분산될 있으며, APPN의 네트워크 관리, 동적 자원 위치 및 전송 경로 계산 기능을 이용할 수 있습니다. DLUR은 LU가 정의된 노드에서 사용 가능해야 하며, DLUS는 호스트 노드에서 사용 가능해야 합니다. 그러나 세션 전송 경로에서 어떤 중계 노드에도 DLUR이 필요하지 않습니다.

주: LEN 노드에서는 DLUR을 구성할 수 없습니다.

Linux용 Communications Server DLUR 노드가 네트워크 노드이면 Linux용 Communications Server 노드에 연결된 다운스트림 컴퓨터의 종속 LU에 패스-스루 DLUR 기능을 제공할 수도 있습니다. (네트워크 노드만 이 기능을 지원합니다.) 이러한 다운스트림 LU는 Linux용 Communications Server 노드의 DLUR을 사용하여 노드 내부의 LU에서와 같은 방법으로 네트워크를 거쳐 호스트에 액세스할 수 있습니다.

주: 끝 노드에서는 패스-스루 DLUR을 구성할 수 없습니다.

DLUR을 구성하기 위해 수행해야 하는 작업은 종속 LU가 로컬 노드에 있는지, 아니면 다운스트림 노드에 있는지 여부에 따라 다릅니다.

로컬 노드에서 DLUR 지원을 구성하려면, 다음 구성 작업을 수행해야 합니다.

1. 62 페이지의 『노드 구성』에 설명된 대로 로컬 노드를 정의하십시오. 다운스트림 노드에 대해 패스-스루 DLUR 지원을 제공할 경우, 노드를 APPN 네트워크 노드로 정의하십시오.
2. APPN 네트워크와의 연결성을 구성하십시오. APPN 연결을 위해서는 67 페이지의 제 4 장 『연결 구성요소 정의』에 설명된 대로 로컬 노드와 인접 APPN 네트워크 노드 사이의 독립 트래픽을 위한 최소 하나의 포트 및 링크 스테이션이 필요합니다.
3. 82 페이지의 『DLUR PU 정의』에 설명된 대로 로컬 노드에서 DLUR PU를 정의하십시오. (DLUR PU는 호스트와의 연결성을 지원합니다.)
4. 로컬 노드에서 LU를 지원하도록 DLUR을 구성하려면 117 페이지의 제 7 장 『사용자 응용프로그램 구성』에 설명된 대로 로컬 노드에 LU를 추가해야 합니다. LU는 3270 표시장치, 3270 프린터 또는 LUA를 지원하도록 구성할 수 있습니다. LU가 지원하는 사용자 응용프로그램의 요구사항에 따라 추가 구성을 수행해야 할 수도 있습니다.

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

1. 로컬 노드를 APPN 네트워크 노드로 정의하십시오.(62 페이지의 『노드 구성』 참조).
2. 다운스트림 노드와의 연결성을 구성하십시오. 67 페이지의 제 4 장 『연결 구성요소 정의』에 설명된 대로 로컬 노드와 각 다운스트림 노드 사이의 종속 트래픽을 위한 포트 및 링크 스테이션을 구성하십시오. (다운스트림 노드에 대해 DLUR을 지원하도록 DLUR PU를 정의할 필요는 없습니다.)
3. 다운스트림 노드는 여러 개의 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.

제 9 장 NetView에서 Linux용 Communications Server 관리

Linux용 Communications Server에는 호스트 컴퓨터에서 NetView 프로그램과 함께 작동하여 NetView 운영자가 호스트 NetView 프로그램에서 Linux용 Communications Server 컴퓨터로 명령을 실행할 수 있도록 하는 원격 명령 기능(RCF)이 포함되어 있습니다. (NetView 및 RCF 명령에 대한 간단한 개요는 『호스트 NetView 프로그램 사용』을 참조하십시오.)

Linux용 Communications Server RCF는 다음의 두 가지 기능을 제공합니다.

- NetView 운영자는 서비스점 명령 기능(SPCF)을 사용하여 명령행 관리 프로그램 **snaadmin**과 같은 구문으로 NetView에서 Linux용 Communications Server 관리 명령을 실행할 수 있습니다. 이 기능은 136 페이지의 『SPCF 사용』에 설명되어 있습니다.
- NetView 운영자는 UNIX 명령 기능(UCF)을 사용하여 NetView에서 Linux 운영 체제 명령을 실행할 수 있습니다. 이 기능은 138 페이지의 『UCF 사용』에 설명되어 있습니다.

두 기능에는 모두 같은 방법으로 NetView 콘솔에서 액세스할 수 있으며, 명령을 실행하기 위한 전반적인 구문도 같습니다.

호스트 NetView 프로그램 사용

Linux용 Communications Server RCF는 호스트 컴퓨터에 있는 NetView 프로그램과 함께 작동합니다. 호스트는 NetView 버전 1, 릴리스 2 이상을 실행하고 있어야 합니다. Linux용 Communications Server는 NetView 버전 1 릴리스 1을 지원하지 않습니다.

NetView 프로그램을 사용하려면 다음이 필요합니다.

- 호스트 NetView 프로그램에 대한 로그인 ID 및 암호(이 정보에 대해서는 호스트 담당자에게 문의하십시오.)
- NetView 프로그램의 호스트에 정의된 Linux용 Communications Server의 서비스점 이름(이 정보에 대해서는 호스트 담당자에게 문의하십시오.)
- NetView 프로그램이 실행 중인 호스트 컴퓨터에 액세스하기 위한 DLC, 포트 및 링크 스테이션

호스트에서 직접 액세스하는 대신 Linux용 Communications Server에서 NetView에 액세스하기 위해 3270 에뮬레이션을 사용하여 RCF 기능을 검사할 수도 있습니다. 이 경우 다음 정보도 필요합니다.

- 호스트에 구성된 3270 LU
- 이 LU를 사용하는 3270 세션

필요한 구성 정보를 확보하려면 호스트 관리자에게 문의하십시오.

NetView 프로그램에 액세스하려면 다음 단계를 수행하십시오.

1. RCF 액세스 매개변수 정의(define_rcf_access 레코드)를 포함하는 노드 구성 파일을 사용하여 Linux용 Communications Server 소프트웨어가 시작되어 있는지 확인하십시오.
2. 3270 에뮬레이션을 사용하여 NetView 프로그램에 액세스하는 경우, 3270 에뮬레이션 프로그램을 시작하여 호스트에 대해 세션을 활성화하십시오.
3. NetView 시작 및 로그온에 대해서는 호스트 관리자가 제공하는 지시사항에 따르십시오. (조작 순서는 NetView 버전에 따라 다를 수 있습니다.)
4. 필요에 따라 SPCF 또는 UCF 명령을 실행하십시오.
5. 3270 에뮬레이션을 사용하여 NetView에 액세스할 경우, 명령 실행을 완료했을 때 3270 에뮬레이션을 종료하는 방법에 대해서는 3270 문서의 지시사항을 따르십시오.

NetView 화면 표시장치

NetView 화면 레이아웃은 각기 다른 호스트에 있는 NetView 버전에 따라 다릅니다. 일반적인 레이아웃은 그림 19에 나와 있습니다.

표시장치에는 화면 맨 아래에 명령을 입력할 수 있는 입력 영역이 있습니다. ??? 행은 입력 영역에서 기본 화면 영역(NetView가 명령에 대한 응답을 표시하는)을 나눕니다.

```
NCCF          N E T V I E W          [SCAN DDAC12  07/18/95  13:52:24 A
              RUNCMD SP=ADCDPU01,APL=NODE,START_DLC,DLC_NAME=TKR01
              COMMAND ISSUED SUCCESSFULLY

              ???
              runcmd sp=abcdpu01,appl=node,query_node
```

그림 19. NetView 화면 예제

명령 입력 영역 크기 변경

디폴트로 입력 영역은 한 줄로 되어 있지만, 일부 긴 명령을 입력하기 위해서는 여러 행이 필요합니다. 일부 NetView 버전에서는 **input** 명령을 사용하여 하나, 둘 또는 세 행의 입력 영역을 지정할 수 있습니다. 이를 수행하려면 다음 명령을 입력하십시오.

input n

이 명령에서 *n*은 원하는 행 수를 표시하는 1, 2 또는 3입니다. 이 명령이 사용 중인 NetView 버전에서 작동하지 않을 경우, NetView 지원 담당자에게 문의하십시오.

RCF 명령 구문 개요

SPCF 및 UCF 명령 둘 다 RCF 명령 구문을 사용합니다.

runcmd sp=*spname*, appl=*component*, *commandtext*

NetView는 **runcmd** 유틸리티를 사용하여 명령 문자열을 원격 시스템에 보냅니다. 명령에 포함되는 매개변수는 다음과 같습니다.

sp=*spname*

Linux용 Communications Server 노드에 해당되는 서비스점 이름(NetView에 정의된)을 표시합니다. 호스트 NetView 담당자가 이 정보를 제공할 수 있습니다.

appl=*component*

NetView가 다음과 같이 명령을 전송해야 하는 Linux용 Communications Server 구성요소의 이름을 표시합니다.

node 서비스점 이름 *spname*(SPCF 명령의 경우)과 연관되는 Linux용 Communications Server 노드

unix 서비스점 이름 *spname*(UCF 명령의 경우)과 연관되는 Linux용 Communications Server 컴퓨터에서 실행 중인 UCF 디먼 프로그램

commandtext

실행하는 명령의 텍스트를 제공합니다. SPCF의 경우, 이는 Linux용 Communications Server 명령행 관리 프로그램에 대해 실행되는 명령입니다. UCF의 경우, 이는 Linux 운영 체제에 해당되는 명령입니다. 사용할 수 있는 명령에 대한 자세한 정보는 136 페이지의 『SPCF에서 사용되는 관리 명령의 제한사항』 또는 139 페이지의 『사용할 수 있는 명령』을 참조하십시오.

대문자 및 이탈 문자

Linux는 대문자와 소문자를 구별하지만, NetView 프로그램은 그렇지 않습니다. NetView **netvaxis** 명령에서는 대문자와 소문자를 혼합하여 **runcmd**를 입력할 수 있지만 Linux용 Communications Server RCF에서는 **netvaxis**가 현재 사용 중인지 여부를 판별할 수 없습니다. RCF는 호스트에서 수신된 영문자가 원래 대문자인지, 아니면 소문자인지 판별할 수 없으므로 수신된 문자가 소문자인 것으로 간주합니다. 또한 호스트 문자 세트는 일부 명령에 반드시 필요한 대괄호 문자 [및]를 지원하지 않을 수도 있습니다.

RCF는 다음과 같이 백슬래시 문자 \를 사용하여 대괄호 문자와 대문자에 대한 지원을 제공합니다.

- 명령 문자열에 대문자를 포함시키려면, 대문자 앞에 백슬래시를 포함시키십시오. 앞에 백슬래시가 없는 영문자는 소문자인 것으로 해석됩니다.
- 대괄호 문자 [및]를 포함시키려면, 순서 \< (및 \)를 각각 사용하십시오.
- 백슬래시 문자 \ 자체를 포함시키려면 두 번 입력하십시오.

단일 백슬래시 다음에 영문자가 아닌 다른 문자가 있을 경우, 백슬래시는 무시되고 문자는 변경되지 않은 상태로 남습니다.

일부 예가 표 3에 나와 있습니다.

표 3. RCF 명령에서 이탈 문자 사용

생성할 문자	입력
ABcd	\a\bcd
[]	\(\)
\a	\\a
\[\\\[

Linux 명령행에서 Linux 셸이 특수 문자를 해석하지 못하도록 하기 위해 정상적으로 사용하는 이탈 문자는 RCF에서 필요하지 않습니다. 예를 들어, 문자 * 또는 \$를 포함하는 문자열에서는 Linux 명령행에서 입력하는 것처럼 이탈 문자를 사용하지 마십시오. 또한 SPCF를 사용하여 관리 명령을 실행할 경우 LIST_FROM_NEXT와 같은 상수 이름에서 대소문자가 구분되지 않는다는 점에 유의하십시오. 이 문자들을 대문자로 만들기 위해 이탈시킬 필요가 없습니다.

SPCF 사용

SPCF를 사용하면 NetView 콘솔에서 명령을 실행하여 실행 중인 Linux용 Communications Server 시스템을 관리할 수 있습니다. 이들 명령은 Linux용 Communications Server 명령행 관리 프로그램 **snaadmin**을 사용하여 실행하는 명령들과 같습니다(Linux용 IBM Communications Server 관리 명령 참조서 참조).

SPCF 명령 구문에 대해서는 135 페이지의 『RCF 명령 구문 개요』를 참조하십시오. *appl=node* 매개변수 다음에 오는 명령 텍스트는 Linux 명령행에서 **snaadmin** 프로그램에 대해 지정하는 것과 같은 형식으로 Linux용 Communications Server 명령행 관리 프로그램에 대해 실행되는 명령입니다. 개별 명령의 관리 명령 구문과 매개변수에 대한 자세한 정보는 *Linux용 IBM Communications Server 관리 명령 참조서*를 참조하십시오.

SPCF에서 사용되는 관리 명령의 제한사항

명령행 옵션 **-i**를 사용하여 파일이나 표준 입력에서 입력을 지정할 수 없습니다. 모든 명령은 NetView 콘솔에서 직접 입력해야 합니다.

query_* 명령을 사용할 경우, Linux 명령행에서 명령을 입력할 경우와 같은 방식으로 명령행 옵션 **-a**(모든 항목 리턴) 및 **-d**(자세한 정보 리턴)를 사용할 수 있습니다.

보안을 제공하기 위해, SPCF에서 특정 유형의 명령만 허용되도록 Linux용 Communications Server 구성을 설정할 수 있습니다. 예를 들어, 원격 사용자가 **query_*** 명령은 실행할 수 있지만 Linux용 Communications Server 구성요소를 활성화 또는 비활성화할 수 없도록 제한할 수 있습니다. 다음 명령 그룹마다 별도로 액세스를 제어할 수 있습니다.

- **define_***, **set_***, **delete_***, **add_*** 및 **remove_*** 명령과 **init_node** 명령
- **query_*** 명령
- 『조치』 명령: **start_***, **stop_***, **activate_***, **deactivate_*** 명령과, **aping**, **initialize_session_limit**, **change_session_limit** 및 **reset_session_limit** 명령

SPCF의 보안 옵션 설정에 대한 자세한 정보는 *Linux용 IBM Communications Server 관리 명령 참조서*에서 **define_rcf_access** 명령에 대한 설명을 참조하십시오.

SPCF 명령 예제

다음 예제는 SPCF를 사용하여 **define_lu_0_to_3** 명령을 실행할 수 있는 방법을 보여줍니다. 이 예제에서는 두 문자열 LU\$01 및 PU2에서 대문자를 표시하기 위해 백슬래시 문자를 사용하고 있습니다. 상수 이름 3270_display_model_2에 있는 문자들은 대문자로 만들지 않아도 됩니다. **snaadmin** 프로그램은 이 문자열을 소문자로 승인하기 때문입니다.

```
runcmd sp=mypname, appl=node, define_lu_0_to_3, lu_name=\lu$01,
```

```
    nau_address=1, pu_name=\p\u2, lu_model=3270_display_model_2
```

다음 예는 SPCF를 사용하여 **query_lu_0_to_3** 명령을 실행할 수 있는 방법을 보여줍니다. **-a** 옵션은 『모든 항목 리턴』을 표시하므로, LU 이름이나 PU 이름을 지정하지 않아도 됩니다. **-d** 옵션은 『자세한 정보 리턴』을 표시하므로, *list_options* 매개변수를 사용하여 이를 지정하지 않아도 됩니다. 두 옵션은 **snaadmin** 프로그램과 동일한 방식으로 작동합니다.

```
runcmd sp=mypname, appl=node, -a -d query_lu_0_to_3
```

UCF 사용

NetView 운영자는 UCF를 사용하여 NetView 콘솔에서 명령 텍스트를 입력함으로써 Linux용 Communications Server를 실행 중인 컴퓨터에서 Linux 명령을 실행하고 명령들의 출력을 볼 수 있습니다. 이 기능은 Linux용 Communications Server에 관련된 명령으로 제한됩니다. 139 페이지의 『사용할 수 있는 명령』에 설명된 제한사항에 따라 임의 유형의 명령을 실행할 수 있습니다.

원격 운영자는 UCF를 사용하여 Linux용 Communications Server 컴퓨터에서의 활동을 모니터하고 문제점을 진단하며, 어떤 경우에는 정정 조치도 취할 수 있습니다.

define_rcf_access 명령을 사용하여 Linux용 Communications Server가 UCF를 지원 하는지 여부를 지정할 수 있습니다(Linux용 IBM Communications Server 관리 명령 참조서 참조). 구성에 UCF 지원이 지정되어 있으면 Linux용 Communications Server는 노드를 시작할 때 UCF 디먼 프로그램을 시작합니다. UCF 디먼은 명령마다 새 Linux 셸을 시작하고 해당 셸에서 명령을 실행하여 UCF로부터 Linux 명령을 처리합니다. UCF 지원이 포함되지 않으면 Linux용 Communications Server는 이 프로그램을 시작하지 않습니다.

구성에서 UCF 사용자의 이름을 지정하며 이 이름은 Linux용 Communications Server 컴퓨터에서 유효한 로그인 이름이어야 합니다. UCF 셸은 해당 사용자에게 지정된 셸 프로그램, 로그인 ID, 권한, 그리고 **.login** 또는 **.profile**을 사용하여 시작합니다. (셸 프로그램을 지정하지 않으면 **/bin/sh**가 사용됩니다.) 이는 정상적인 Linux 시스템 보안 기능을 사용하여 파일 및 명령에 대한 UCF 사용자 액세스 권한을 제한하며 UCF에서 사용할 수 있는 명령 범위를 제한할 수 있음을 의미합니다.

UCF 구성 설정에 대한 자세한 정보는 *Linux용 IBM Communications Server 관리 명령 참조서*에서 **define_rcf_access** 명령에 대한 설명을 참조하십시오.

UCF 명령 구문

UCF 명령 구문은 다음과 같습니다.

runcmd sp=spname, appl=unix, unix_command

NetView는 **runcmd** 유틸리티를 사용하여 명령을 원격 시스템에 보냅니다. 명령에 포함되는 매개변수는 다음과 같습니다.

sp=spname

NetView에서 정의된 서비스점의 이름인 *spname*을 지정합니다. 호스트 NetView 담당자가 이 정보를 제공할 수 있습니다.

appl=unix

NetView가 서비스점 이름 *spname*과 연관되는 Linux용 Communications Server 컴퓨터에서 UCF 디먼 프로그램에 명령을 전송하도록 지시합니다.

unix_command

Linux 운영 체제 명령을 제공합니다. 이 명령은 대문자나 대괄호 문자를 표시하기 위한 이탤 문자(135 페이지의 『RCF 명령 구문 개요』에 설명됨)를 제외하고, Linux 명령행에 입력한 대로 입력됩니다.

Linux 명령행에서 Linux 셸이 특수 문자를 해석하지 못하도록 하기 위해 정상적으로 사용하는 이탤 문자는 UCF에서 필요하지 않습니다. 예를 들어, 문자 * 또는 \$를 포함하는 문자열에서는 Linux 명령행에서 입력하는 것처럼 이탤 문자를 사용하지 마십시오.

사용할 수 있는 명령

UCF는 사용자의 추가 상호작용 없이 완료되는(출력 생성 여부에 관계없이) 명령과 함께 사용하도록 설계된 것입니다. 예를 들어, *filename*의 내용을 표시한 후 완료하는 **cat filename** 명령이나, 오류가 발생하지 않으면 출력 없이 완료하는 **mv filename1 filename2** 명령을 실행할 수 있습니다.

UCF 명령이 생성하는 출력은 Linux 운영 체제 명령이 완료될 때 UCF에 리턴됩니다. 이 때 다음 제한사항이 발생합니다.

- 명령이 완료된 후 생성되는 출력은 UCF에 전송되지 않습니다. 예를 들어, 뒤에 &가 있는 명령이 백그라운드에서 실행되도록 할 경우, UCF는 백그라운드 명령의 프로세스 ID를 제공하는 운영 체제 메시지는 수신하지만 생성되는 후속 출력은 수신하지 않습니다. 마찬가지로, UCF를 사용하여 디먼 프로세스를 시작할 수 있지만 프로세스에서 생성되는 출력은 볼 수 없습니다.
- UCF는 완료되기 전에 사용자의 추가 입력이 필요한 명령(예: 대화식 프로세스를 시작하는 **vi filename** 명령이나 사용자가 중지할 때까지 완료되지 않는 **tail -f filename** 명령)과 함께 사용할 수 없습니다.

모든 Linux 명령은 구성된 UCF 사용자의 로그인 ID와 권한을 사용하여 실행되므로, UCF 사용자의 로그인 액세스 권한에 따라 유효한 명령이 제한됩니다. 특히, 루트 또는 상위 사용자 명령은 사용할 수 없습니다. 자세한 정보는 141 페이지의 『UCF 보안』을 참조하십시오.

UCF 명령 예제

다음은 NetView에서 입력하는 UCF 명령의 예제입니다.

```
runcmd sp=mypname, appl=unix, grep \temp \{ab\}*.c >\{t\em\p.out
```

Linux에서 실행되는 명령어는 다음과 같습니다.

```
grep Temp [ab]*.c >TEMP.out
```

Linux 시스템 명령의 출력

명령을 성공적으로 실행한 경우, NetView 화면에 다음 메시지가 표시됩니다.

=== EXECUTING UNIX COMMAND ===

(오류 메시지를 포함하여 명령의 출력 표시)

=== UNIX COMMAND COMPLETED ===

이들 메시지는 NetView 화면에 동시에 표시되지 않을 수도 있습니다. **EXECUTING UNIX COMMAND** 메시지는 UCF 디먼 프로그램이 명령을 수신하고 제어를 NetView 운영자에게 리턴하는 즉시 표시됩니다. 명령 출력은 생성되는 대로 NetView에 전송되고 일련의 개별 메시지로 표시될 수 있습니다. **UNIX COMMAND COMPLETED** 메시지는 Linux 명령이 완료되어 해당 셸이 종료된 경우에 표시됩니다.

Linux 명령 출력에 탭 문자가 있으면 Linux용 Communications Server는 각 탭을 공백 문자로 변환한 후 출력을 NetView에 전송합니다. 그렇지 않으면 출력은 변경되지 않은 상태로 전송됩니다.

이전 명령이 계속 진행되고 있을 때 명령을 실행하면(즉, **UNIX COMMAND COMPLETED** 메시지가 수신되기 전에) 다음 메시지가 표시됩니다.

=== COMMAND QUEUED ===

두 번째 명령은 대기행렬에 놓이고, 이전 명령이 완료될 때 실행됩니다.

명령 취소

UCF는 계속 진행 중인 명령을 취소하는 방법을 제공합니다. 이는 현재 명령의 실행을 중지하거나 대화식 명령(예: 추가 입력 없이는 완료할 수 없는 **vi filename**)을 취소하기 위해 사용할 수 있습니다. 이는 **Ctrl + C**와 같은 인터럽트 순서를 사용하여 터미널에서 실행 중인 프로세스를 중지하거나, Linux **kill** 명령을 사용하여 프로세스를 중지하는 것과 같습니다.

현재 실행 중인 명령을 취소하는 것 외에도, Linux용 Communications Server는 그 명령 이후에 대기행렬에 놓여진 명령도 취소합니다.

명령 구문은 명령 텍스트 대신 **ux-cancel** 문자열을 사용하는 것을 제외하고 Linux 명령과 같습니다. 예를 들면 다음과 같습니다.

runcmd sp=myspname, appl=unix, ux-cancel

각각의 미해결 명령(현재 실행 중인 명령과 대기행렬에 있는 명령)에 대해 다음 메시지가 표시됩니다.

=== UNIX COMMAND CANCELLED ===

이 메시지는 명령이 실행 중인 Linux 셸이 중지되었음을 표시합니다. 필요에 따라 추가 Linux 명령을 실행할 수 있습니다.

명령이 Linux 컴퓨터에서 디먼 프로세스를 시작할 경우, 이 프로세스는 **ux-cancel**에 의해 중지할 수 없습니다. 그러한 프로세스를 명시적으로 중지하기 위해서는 Linux **kill** 명령을 사용해야 할 수도 있습니다(터미널에서 또는 UCF를 사용하여).

ux-cancel을 사용할 때 실행 중인 UCF 명령이 없으면, UCF는 다음 메시지를 표시합니다.

NO OUTSTANDING COMMANDS

이 경우 **ux-cancel** 명령은 무시되고, 어떤 조치도 필요하지 않습니다. 이 메시지는 이전 명령이 완료된 후 **UNIX COMMAND COMPLETED** 메시지가 수신되기 전에 **ux-cancel** 명령을 실행할 때 표시될 수 있습니다.

UCF 보안

UCF는 원격 운영자가 Linux 컴퓨터에서 명령을 실행하고 명령의 출력을 수신할 수 있도록 하므로, 보안 관련 사항을 고려하는 것이 중요합니다. 예를 들어, 운영자가 개인 정보에 액세스하거나 다른 사용자를 방해하는 Linux 명령을 실행할 수 없도록 해야 합니다.

Linux용 Communications Server 구성에는 UCF 사용자로서 특정 Linux 시스템 사용자 이름이 포함됩니다. 이 이름은 Linux용 Communications Server 컴퓨터에서 유효한 로그인 ID여야 합니다. 모든 UCF 명령은 이 사용자 ID와 사용자의 액세스 권한으로 실행됩니다.

이는 UCF에서 사용 가능한 명령으로 간주되는 명령들만 허용되도록 Linux에서 제공하는 정상적인 보안 기능을 사용하여 UCF 사용자가 액세스할 수 있는 명령을 제한하도록 하기 위한 것입니다. 다음 지침이 유용할 수 있습니다.

- UCF 사용자 이름은 UCF에만 사용되는 이름이어야 합니다. 다른 목적으로도 사용되는 기존 로그인은 사용할 수 없습니다. 이로서 UCF에 대해 허용되는 권한만 포함되도록 사용자 권한을 쉽게 정의할 수 있고, UCF를 사용하여 시작한 프로세스를 식별할 수 있습니다.
- UCF 사용자가 사용자 ID나 그룹 ID를 변경할 수 있는 사용자 및 그룹을 제한해야 할 수도 있습니다. 특히, UCF 사용자는
 - 루트 또는 상위 사용자가 될 수 없습니다.
 - **snaadmin** 프로그램에 액세스할 수 있는 그룹 ID **system**을 사용할 수 없습니다. (이 프로그램의 기능은 UCF 대신 이 장의 앞 부분에 설명된 SPCF를 사용하여 액세스해야 합니다.)

제 10 장 Linux용 Communications Server 클라이언트/서버 시스템 관리

Linux용 Communications Server는 단일 Linux 시스템의 모든 SNA 구성요소 및 응용프로그램과 독립형 시스템으로 작동하거나 클라이언트/서버 도메인의 일부로 작동할 수 있습니다. 클라이언트/서버 도메인에는 서버(SNA 노드)와 IBM 원격 API 클라이언트(서버를 통해 SNA 연결에 액세스할 수 있는) 둘 다 포함될 수 있습니다.

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 정의할 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.

원격 API 클라이언트는 AIX, Linux, pSeries용 Linux, System z용 Linux 또는 Microsoft Windows를 실행하는 컴퓨터가 될 수 있습니다.

서버 및 클라이언트는 Linux용 Communications Server 도메인에서 TCP/IP를 사용하여 통신합니다(IPv4 및 IPv6 주소지정 둘 다 지원). 클라이언트는 동시에 하나 이상의 서버에 액세스할 수 있고, 필요에 따라 동시 응용프로그램을 실행할 수 있습니다. 클라이언트/서버 구성에 대한 네트워킹 요구사항에 대해서는 146 페이지의 『IP 네트워킹 요구사항』을 참조하십시오.

클라이언트와 서버 간에 사용되는 TCP/IP 연결은 물리적 LANS, WAN 또는 VM에서 실행되고 있는 서버 간의 가상 경로에서 설정됩니다. Linux용 Communications Server 책에서, LAN 용어는 이러한 항목 모두에 사용됩니다.

AIX, LINUX

AIX 또는 Linux의 원격 API 클라이언트의 경우 Linux용 Communications Server 네트워크 및 서버에 대한 정보를 제공해야 합니다. 이 기능에 대한 정보와 클라이언트에서 Linux용 Communications Server 소프트웨어를 사용 가능 및 사용 불가능하게 하는 방법에 관한 지시사항은 169 페이지의 『AIX 또는 Linux의 원격 API 클라이언트 관리』를 참조하십시오.

서버에서 모든 관리 명령을 실행할 수 있습니다. 하지만 AIX 및 Linux 클라이언트에서 명령을 실행할 때 제한사항이 있습니다.

- **query** 또는 **status** 명령을 AIX 또는 Linux 클라이언트에서 실행할 수 있습니다.

- Linux용 IBM Communications Server 관리 명령 참조서에 정의된 일부 다른 관리 명령은, IBM 원격 API 클라이언트에서 실행될 수 있음을 명시적으로 밝히고 있습니다. 그렇지 않은 경우 이 명령들은 서버에서만 사용할 수 있습니다.

WINDOWS

Windows 클라이언트의 경우, Linux용 Communications Server가 사용하여 클라이언트 소프트웨어를 사용 가능하게 할 수 있는 정보를 제공해야 합니다. Windows 클라이언트에서 기동 가능 TP를 사용할 예정이면 TP에 대한 정보도 제공해야 합니다. 이 기능에 대한 정보와 Windows 클라이언트에서 Linux용 Communications Server 소프트웨어를 사용 가능 및 사용 불가능하게 하는 방법에 관한 지시사항은 150 페이지의 『Windows에서 원격 API 클라이언트 관리』를 참조하십시오.

Linux용 IBM Communications Server 관리 명령 참조서에 정의된 관리 명령은 Windows 클라이언트에서 실행할 수 없습니다.



클라이언트/서버 구성 변경

Linux용 IBM Communications Server 빠른 시작에 설명된 것처럼 Linux용 Communications Server 소프트웨어를 설치할 때, 소프트웨어는 초기에 독립형 모드(모든 구성요소가 단일 Linux 컴퓨터에 있음)에서 설치됩니다. Linux용 Communications Server를 클라이언트/서버 시스템으로 실행하려면 한 서버를 마스터 서버로 구성한 다음 다른 서버를 백업 서버로 구성할 수 있습니다. (마스터 이외의 모든 서버를 백업 서버로 구성하는 것이 좋습니다.)

Linux용 Communications Server는 명령행 응용프로그램인 **snetutil**을 포함하여 클라이언트/서버 도메인의 서버 부분으로 만듭니다. 이렇게 하려면 각 서버에서 다음 명령을 사용하십시오(마스터 서버로 시작).

```
sna stop
```

```
snetutil master_name [domain_name]
```

```
sna start
```

snetutil 명령의 매개변수는 다음과 같습니다.

master_name

서버가 속하는 도메인에서 마스터 서버의 이름입니다. 서버를 기존 도메인으로 이동할 경우, 이는 해당 도메인에 있는 기존 마스터 서버의 이름과 일치해야 합니다.

domain_name

서버가 속하는 도메인의 이름입니다. 이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, Linux용 Communications Server는 디폴트 도메인 이름인 `ibmcs_domain`을 사용합니다.

마스터 이외의 각 서버를 백업 서버로 구성하려면 다음 명령을 발행하십시오. 백업 서버 자체나 마스터 서버에서 수행할 수 있지만 Linux용 Communications Server 소프트웨어는 두 경우 모두 마스터 서버에서 실행해야 합니다.

snaadmin add_backup, backup_name=server_name

*server_name*은 백업 서버로 추가할 서버 이름입니다.

독립형 시스템으로 실행하도록 **snautil** 프로그램을 사용하여 서버를 기존 도메인 밖으로 이동할 수 있습니다.

주: Linux용 Communications Server를 클라이언트/서버 시스템으로 실행하는 것을 중지하고 이를 단지 독립형 노드로 사용하려는 것이 아니면 이 옵션을 사용하지 마십시오. 기존 도메인에서 모든 서버를 제거하면, 해당 도메인에 남아 있는 클라이언트는 SNA 자원에 액세스할 수 없습니다.

독립형 시스템으로 실행하도록 서버를 도메인 밖으로 이동하려면 다음 명령을 사용하십시오.

snautil -d**클라이언트를 다른 도메인으로 이동**

snautil 프로그램을 사용하면 서로 다른 클라이언트/서버 도메인 사이에 서버를 이동할 수 있습니다. 도메인 사이에 클라이언트를 이동하려면, 클라이언트 구성을 수정하여 이를 수행해야 합니다.

이동할 Windows의 각 원격 API 클라이언트에서 클라이언트 구성 유틸리티를 사용하여 *domain* 매개변수를 새 도메인 이름과 일치하도록 변경하십시오. 자세한 내용은 152 페이지의 『Windows의 원격 API 클라이언트 구성』을 참조하십시오.

이동할 AIX 또는 Linux의 각 원격 API 클라이언트에서, 클라이언트 네트워크 데이터 파일의 구성 섹션에 있는 *domain* 항목을 새 도메인 이름과 일치하도록 변경하십시오. 자세한 내용은 170 페이지의 『클라이언트 네트워크 데이터 파일(sna_clnt.net)』을 참조하십시오.

IP 네트워킹 요구사항

원격 API 클라이언트는 TCP/IP를 사용하거나 WebSphere 서버를 통한 HTTPS를 사용하여 Linux용 Communications Server와 통신할 수 있습니다. HTTPS 연결을 사용하는 방법에 대한 자세한 내용은 149 페이지의 『원격 API 클라이언트에 대한 HTTPS 액세스』를 참조하십시오.

원격 API 클라이언트를 실행하기 전에 먼저 네트워크에서 클라이언트 및 서버 둘 다에 TCP/IP 포트 주소를 구성해야 합니다. 디폴트 포트 지정에 문제점이 발생하면 147 페이지의 『IP 포트 번호 설정』에 설명된 대로 충돌을 해결해야 합니다.

또한 클라이언트가 Linux용 Communications Server 사용을 완료했을 때 자동으로 TCP/IP 연결이 중단되도록 클라이언트를 설정할 수 있습니다(148 페이지의 『LAN 액세스 시간종료』 참조).

IPv4 및 IPv6 주소지정

Linux용 Communications Server 클라이언트/서버 도메인의 컴퓨터에서 IPv4 또는 IPv6 주소를 사용할 수 있지만 도메인의 모든 서버에서는 동일한 주소지정 형식(IPv4 또는 IPv6)을 사용해야 합니다.

- 서버가 IPv4를 사용하는 경우, 클라이언트도 IPv4를 사용해야 합니다.
- 서버가 IPv6를 사용하는 경우, 클라이언트가 IPv6 또는 IPv4를 사용할 수 있습니다.

IPv4 및 IPv6 주소지정 설정 및 사용 방법에 대한 자세한 정보는 *Linux용 IBM Communications Server* 빠른 시작을 참조하십시오.

클라이언트/서버 구성의 호스트 이름

Linux용 Communications Server는 서버와 클라이언트 간의 내부 통신에 전체 IP 호스트 이름을 사용합니다. 일반적으로 로컬 시스템이 네트워크 구성(예: DNS)에서 이러한 이름을 판별할 수 있습니다. 판별할 수 없는 경우, 구성에 호스트 이름이 필요할 때마다 별명(예: newbox)이 아닌 완전한 이름(예: newbox.this.co.uk)을 사용해야 합니다.

각 컴퓨터의 로컬 서버 이름은 **/etc/hosts** 파일에서 비롯됩니다. 이 파일을 항목은 IP 주소를 먼저 지정한 다음, 완전한 이름을 지정하고, 마지막으로 별명을 지정해야 합니다. 예를 들면 다음과 같습니다.

```
9.42.108.28    newbox.this.co.uk    newbox
```

서버가 다중호인 경우(예: 서버에 둘 이상의 TCP/IP 네트워크 인터페이스가 있어 다른 IP 주소로 표시되는 경우), **/etc/hosts** 파일의 항목이 모든 주소에 대해 동일한 IP 이

름을 지정하여 해당 이름이 모든 네트워크 인터페이스에 대해 올바르게 해석될 수 있도록 해야 합니다. 예를 들면 다음과 같습니다.

```
9.42.108.28  newbox.this.co.uk  newbox
9.42.80.127  newbox.this.co.uk  newbox
```

또한 multi on 행을 **/etc/host.conf** 파일에 포함시켜 서버가 다중호임을 표시해야 합니다.

IP 포트 번호 설정

Linux용 Communications Server는 LAN에서 클라이언트/서버 데이터를 전송하기 위해 TCP/IP 및 UDP/IP 통신을 사용합니다. 디폴트로, 두 유형의 통신에 대해 포트 번호 1553을 사용합니다. 대부분의 설치에서는 이 포트 번호가 적합하므로 변경하지 않아도 됩니다.

Linux용 Communications Server 소프트웨어를 사용 가능하게 할 때 문제점이 발생하면, 오류 로그 파일을 확인하여 Linux용 Communications Server에서 사용하는 포트 번호가 다른 프로그램에서 사용하는 포트 번호와 충돌함을 표시하는 메시지가 있는지 살펴보십시오. 그러한 메시지가 있으면 다음 단계를 수행하십시오.

1. 오류가 발생한 컴퓨터에서 **/etc/services** 파일을 확인하여 다른 프로그램이 TCP/IP 또는 UDP/IP 통신용으로 포트 번호 1553을 사용하는 것으로 나열되는지 살펴보십시오. 그러한 경우, 먼저 다른 프로그램이 다른 포트를 사용하도록 변경하십시오.
2. 이를 수행할 수 없거나 포트 1553을 사용하는 것으로 나열되는 프로그램이 없으면, 파일에서 어떤 프로그램에도 사용되지 않는 것으로 나열되는 다른 포트 번호를 찾으십시오. 같은 도메인에 있는 다른 모든 Linux용 Communications Server 컴퓨터에서 **/etc/services** 파일을 확인하여 다른 컴퓨터에서 그 포트 번호가 사용되지 않는지 확인하십시오.
3. 도메인에 있는 각 컴퓨터의 **/etc/services** 파일에서 다음 양식의 두 행을 추가하십시오.

```
sna-cs      nnnn/tcp
sna-cs      nnnn/udp
```

nnnn 항목은 새 포트 번호입니다. 이 번호는 Linux용 Communications Server 도메인에 있는 모든 컴퓨터에서 같은 값으로 설정해야 합니다.

- 4.

WINDOWS

Linux용 Communications Server 도메인에 Windows 클라이언트가 있으면 동일한 두 행을 각 Windows 컴퓨터에 있는 **services** 파일에 추가하십시오. **services**

파일은 Linux 파일과 같은 형식으로 되어 있고, 일반적으로 Windows TCP/IP 소프트웨어의 홈 디렉토리에 저장됩니다. 필요할 경우 자세한 정보는 Windows TCP/IP 문서를 참조하십시오.



5. Linux용 Communications Server 및 원격 API 클라이언트 소프트웨어를 다시 사용 가능화하십시오.

주: 서버의 포트 1553(또는 클라이언트/서버 통신에 지정한 새 포트 번호)을 보호하여 무단 액세스를 방지하기 위해서 방화벽을 사용해야 합니다. Linux용 Communications Server 및 원격 API 클라이언트에서는 TCP와 UDP 트래픽 둘 다 허용해야 하지만 다른 컴퓨터에서는 포트 액세스를 허용하지 않아야 합니다.

LAN 액세스 시간종료

클라이언트가 연결 비용을 지불하는 네트워크를 거쳐 Linux용 Communications Server 와 통신하는 경우, 클라이언트의 응용프로그램이 Linux용 Communications Server 자원 사용을 중지한 후에 자동으로 클라이언트의 TCP/IP 연결이 중단될 수도 있습니다. 이로 인해 클라이언트에서 SNA 소프트웨어가 자동으로 사용 불가능하게 되는 것은 아닙니다. 이것은 활동 상태로 유지되며 응용프로그램이 나중에 필요로 할 경우 서버와의 접속을 다시 설정합니다.

lan_access_timeout 매개변수(AIX나 Linux에서 원격 API 클라이언트의 경우 **sna_clnt.net** 파일, Windows에서 원격 API 클라이언트의 경우 레지스트리에서)를 사용하여 클라이언트에서 SNA 소프트웨어를 사용 불가능하게 할 수 있습니다. TCP/IP 연결은 지정된 시간 동안 클라이언트에서 다음의 어떤 이벤트도 발생하지 않을 경우에 중단됩니다.

- APPC 또는 CPI-C 대화 활동 중(또는 대화를 시작하려고 시도함)
- LUA 세션 활동 중
- Windows 클라이언트로부터의 CSV TRANSFER_MS_DATA Verb
- MS verbs(Linux 클라이언트만)
- NOF Verb(**query_central_logger** 또는 **query_node_all** Verb 제외)
- 관리 명령(클라이언트가 연결을 재시도하지 않도록 하는 다음 이벤트 제외)
 - 클라이언트가 기록한 오류 또는 감사 메시지(이 메시지들은 중앙 로깅을 사용할 경우에도 클라이언트에 로컬로 기록됨)
 - 관리 명령 **query_central_logger** 또는 **query_node_all**(이 명령은 TCP/IP 연결이 끊어지기 전에 사용 가능했던 정보를 리턴하므로 현재 LAN 상태와 일치하지 않을 수 있음)

- NOF Verb **query_central_logger** 또는 **query_node_all**(해당되는 관리 명령과 같음)

특히, SNA 소프트웨어를 사용 가능하게 하지만 지정된 시간종료 이내에 클라이언트에서 Linux용 Communications Server 응용프로그램을 시작하지 않을 경우에 TCP/IP 연결이 끊어집니다.

TCP/IP 연결이 중단되어 있을 때 이 이벤트 중 하나가 발생하면, 클라이언트는 170 페이지의 『클라이언트 네트워크 데이터 파일(sna_clnt.net)』 또는 157 페이지의 『Servers』에서 * 및 *servername* 매개변수에 대해 설명한 대로 서버 접속을 재시도합니다.

이 클라이언트에서 기동된 TP에 대한 입력 Attach는 TCP/IP 연결이 중단되어 있을 경우 승인할 수 없습니다. Attach는 목표 시스템이 비활성 상태인 것으로 간주하고 거부됩니다. 이는 클라이언트에서 자동으로 시작된 TP는 클라이언트에서 실행 중인 다른 응용프로그램이 없고 TCP/IP 연결이 시간종료 값을 초과한 경우에 사용할 수 없음을 의미합니다. 그러나 클라이언트에서 운영자가 시작한 TP는 언제든지 사용할 수 있습니다. TP가 발행한 Receive_Allocate Verb가 TCP/IP 연결을 재설정하기 때문입니다.

원격 API 클라이언트에 대한 HTTPS 액세스

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. 이 서버의 설치 및 구성 방법에 대한 지시사항은 *Linux용 IBM Communications Server* 빠른 시작을 참조하십시오.

새 서버를 Linux용 Communications Server 도메인에 추가하고 원격 API 클라이언트가 HTTPS를 사용하여 이러한 서버에 액세스할 수 있게 하려면 이러한 서버를 포함하도록 WebSphere 서버 구성 파일을 갱신해야 합니다. 이 파일은 **snahttpsrv.cfg**로 명명되고 **USER_INSTALL_ROOT** 환경 변수에서 지정한 디렉토리의 WebSphere 서버에 저장됩니다. 위치가 확실하지 않으면 다음 단계를 수행하십시오.

1. WebSphere 관리 콘솔을 시작하십시오.
2. 관리 콘솔 메뉴 표시줄에서 환경, WebSphere 변수 관리를 선택하십시오.
3. 이 리스트에서 **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.

HTTPS를 사용하여 원격 API 클라이언트에서 액세스할 수 있는 모든 Linux용 Communications Server 리스트를 포함하도록 텍스트 편집기를 사용하여 구성 파일을 편집하십시오. 각 서버는 파일의 별도 행에 다음 형식으로 지정되어야 합니다.

server=servername.domainname.com

새 서버에 액세스하는 각 원격 API 클라이언트에서 클라이언트 네트워크 데이터 파일의 서버 리스트(또는 Windows 클라이언트에 대한 Windows 레지스트리)에도 새 서버 이름을 추가해야 합니다. 이 장의 뒷부분에 있는 적절한 클라이언트에 대한 절을 참조하십시오.

Windows에서 원격 API 클라이언트 관리

WINDOWS

Linux용 Communications Server는 Microsoft Windows를 실행하는 시스템이 Linux용 Communications Server 도메인에서 클라이언트로 작동할 수 있게 합니다. Linux용 Communications Server 클라이언트 소프트웨어에는 Microsoft 호스트 통합 서버, WOSA(Windows Open Systems Architecture), 그리고 Windows용 IBM Personal Communications and Communications Server에서 제공하는 인터페이스와 호환 가능한 API 라이브러리가 있습니다. 이는 이러한 구현을 위해 작성된 응용프로그램이 Windows의 원격 API 클라이언트에서 변경 없이 실행되도록 합니다.

Windows의 원격 API 클라이언트는 다음 WOSA API를 지원합니다.

- Windows APPC
- Windows CPI-C
- Windows LUA
- Windows CSV

Windows SNA API에 대한 자세한 정보는 Microsoft 호스트 통합 서버와 함께 제공되는 문서를 참조하십시오.

SNA 네트워크 정보와 Windows의 원격 API 클라이언트에 필요한 기타 정보는 Windows 레지스트리에 보유됩니다.

클라이언트는 사용자가 클라이언트에서 Linux용 Communications Server 응용프로그램이나 애플리케이션 프로그램을 사용하기 전에 사용 가능 상태에 있어야 합니다. 자세한 정보는 151 페이지의 『Windows의 원격 API 클라이언트 사용 기능화』를 참조하십시오. 클라이언트가 사용 가능하게 되면, 클라이언트는 TCP/IP 네트워크를 거쳐 Linux용 Communications Server를 실행하는 서버에 접속하여 Linux용 Communications Server 기능에 액세스합니다.

클라이언트 조작도 Windows 레지스트리에 있는 정보에 의해 제어됩니다. Windows 레지스트리에는 다음에 관한 정보가 있습니다.

- Windows의 원격 API 클라이언트 특정 구성 정보
- 클라이언트가 액세스할 수 있는 서버

- 클라이언트에서 실행 중인 응용프로그램의 로깅 및 추적 옵션
- 클라이언트에서 실행 중인 CPI-C 및 CSV 응용프로그램의 추가 옵션
- 클라이언트에서 실행할 수 있는 기동 가능 TP(APPC 또는 CPI-C)

가장 일반적으로 사용되는 매개변수는 기본적인 수정 방법인 클라이언트 구성 유틸리티를 사용하여 수정할 수도 있습니다. 자세한 정보는 152 페이지의 『Windows의 원격 API 클라이언트 구성』을 참조하십시오.

주: 클라이언트가 HTTPS를 사용하여 서버에 액세스할 경우 클라이언트 구성을 수정하여 HTTPS 액세스를 제공하는 WebSphere 서버와 이러한 서버의 이름을 지정해야 클라이언트를 사용할 수 있습니다. 자세한 내용은 152 페이지의 『Windows의 원격 API 클라이언트 구성』을 참조하십시오.

Windows의 원격 API 클라이언트 사용 가능화

Windows의 원격 API 클라이언트는 Windows 서비스로 실행합니다. 설치 프로그램은 컴퓨터가 시작될 때 CS Linux Win32 클라이언트가 자동으로 시작되도록 구성합니다. 필요하다면 다음 방법 중 하나를 사용하여 수동으로 시작할 수 있습니다.

- 제어판, 관리 도구 아래의 서비스 애플릿에서 클라이언트 서비스를 시작하십시오.
- 명령 창이나 시작/실행 아이콘에서 `net start sxclient`를 입력하십시오.

그러면 클라이언트는 152 페이지의 『Windows의 원격 API 클라이언트 구성』에 설명된 대로 클라이언트 구성 유틸리티를 사용하여 정의된 Windows 레지스트리에 있는 정보를 사용하여 Linux용 Communications Server를 실행 중인 서버를 찾습니다.

주: Microsoft Windows Vista를 사용하는 경우, 관리자 권한이 있는 명령 프롬프트의 `net start sxclient`를 실행해야 합니다. 이 명령 프롬프트에 액세스하려면 명령 프롬프트 아이콘에서 마우스 오른쪽 단추를 사용하여 '관리자로 실행'을 선택하고 프롬프트에 관리자 암호를 입력하십시오.

Windows의 원격 API 클라이언트 상태 보기

클라이언트 모니터는 사용자가 마우스 포인터를 그 위로 이동할 때 클라이언트 상태를 표시하는 아이콘을 시스템 트레이에 위치시킵니다. 모니터는 컴퓨터가 시작할 때 자동으로 실행되도록 설정되지만, 다음 방법 중 하나로 수동으로 실행할 수도 있습니다.

- 시작 메뉴의 Windows 프로그램 그룹에 있는 원격 API 클라이언트에서 클라이언트 모니터를 선택하십시오.
- 명령 프롬프트에서 클라이언트 소프트웨어가 설치된 디렉토리로 변경한 후 `sxclappl` 명령을 실행하십시오.

클라이언트 모니터는 일반적으로 클라이언트 소프트웨어를 설치할 때 선택한 언어로 정보를 표시합니다. 이 정보를 다른 언어로 표시하려면 명령 프롬프트에서 클라이언트 모니터를 시작하고 클라이언트 소프트웨어가 설치된 디렉토리로 변경한 후 선호하는 언어에 해

Windows에서 원격 API 클라이언트 관리

당하는 서브디렉토리로 변경하고 **sxclappl** 명령을 실행하십시오. 예를 들어, 프랑스로 정보를 표시하려면 클라이언트 소프트웨어가 설치된 디렉토리 아래의 **fr_FR** 서브디렉토리어서 **sxclappl** 명령을 실행하십시오.

클라이언트 모니터는 다음 중 하나로 상태를 표시합니다.

활동 중이 아님

클라이언트가 시작되지 않았습니다.

연결되지 않음

클라이언트가 시작되었으나, 아직 서버와 접속되지 않았습니다. (또는 접속이 끊어졌습니다.)

Server_Name

클라이언트가 이름 지정된 서버에 연결되어 있습니다.

Windows의 원격 API 클라이언트 사용 불가능화

클라이언트를 사용 불가능하게 하기 전에, 클라이언트에 있는 모든 Linux용 Communications Server 응용프로그램(3270 및 5250 에뮬레이션 프로그램 또는 Linux용 Communications Server API를 사용하는 응용프로그램)이 중지되었는지 확인하십시오.

클라이언트를 사용 불가능하게 하려면 다음 방법 중 하나로 클라이언트 서비스를 중지하십시오.

- 제어판, 관리 도구 아래의 서비스 애플릿에서 클라이언트 서비스를 중지하십시오.
- 명령 창이나 시작/실행 아이콘에서 **net stop sxclient**를 입력하십시오.

Windows 터미널 서비스를 실행 중인 컴퓨터에서, 이는 모든 사용자가 클라이언트를 사용할 수 없음을 의미합니다.

주: Microsoft Windows Vista를 사용하는 경우, 관리자 권한이 있는 명령 프롬프트의 **net stop sxclient**를 실행해야 합니다. 이 명령 프롬프트에 액세스하려면 명령 프롬프트 아이콘에서 마우스 오른쪽 단추를 사용하여 '관리자로 실행'을 선택하고 프롬프트에 관리자 암호를 입력하십시오.

Windows의 원격 API 클라이언트 구성

Windows의 원격 API 클라이언트에서 구성 정보는 Windows 레지스트리에 보존됩니다. 레지스트리에는 SNA 네트워크 정보(AIX 또는 Linux에서 클라이언트 네트워크 데이터 파일에 보유된 정보와 유사한)가 있습니다. Windows에서 원격 API 클라이언트에 해당되는 추가 구성 정보도 일부 있습니다.

주: CPI-C 응용프로그램에 대한 구성 정보(로컬 TP 이름 및 로컬 LU 별명)는 환경 변수 또는 레지스트리에 지정할 수 있습니다. Windows 터미널 서버를 사용하고 다

른 로컬 LU를 사용하여 동일한 응용프로그램의 여러 사본을 실행할 경우 환경 변수를 사용해야 할 수 있습니다. 자세한 정보는 167 페이지의 『Appl_Name』을 참조하십시오.

클라이언트 구성 유틸리티는 가장 일반적으로 사용되는 클라이언트 구성 매개변수를 수정하는 간단한 방법을 제공하며 이러한 매개변수를 수정하는 기본 방법입니다. 다음 방법 중 하나로 이 프로그램을 시작할 수 있습니다.

- 시작 메뉴의 Windows 프로그램 그룹에 있는 원격 API 클라이언트에서 구성 유틸리티를 선택하십시오.
- 명령 프롬프트에서 클라이언트 소프트웨어가 설치된 디렉토리로 변경한 후 **sxclconf** 명령을 실행하십시오.

구성 유틸리티는 일반적으로 클라이언트 소프트웨어를 설치할 때 선택한 언어로 정보를 표시합니다. 이 정보를 다른 언어로 표시하려면 명령 프롬프트에서 구성 유틸리티를 시작하고 클라이언트 소프트웨어가 설치된 디렉토리로 변경한 후 선호하는 언어에 해당하는 서브디렉토리로 변경하고 **sxclconf** 명령을 실행하십시오. 예를 들어, 프랑스로 정보를 표시하려면 클라이언트 소프트웨어가 설치된 디렉토리 아래의 **fr_FR** 서브디렉토리에서 **sxclconf** 명령을 실행하십시오.

프로그램이 초기 설치 프로세스에 표시되었던 동일한 구성 창을 표시합니다. 이러한 구성 매개변수를 수정하는 방법에 대한 자세한 내용은 Linux용 IBM Communications Server 빠른 시작에서 Windows의 원격 API 클라이언트 설치 장을 참조하십시오.

주: 클라이언트 구성 매개변수를 변경한 후 클라이언트를 중지하고 다시 시작해야 변경 사항이 적용됩니다. 자세한 내용은 152 페이지의 『Windows의 원격 API 클라이언트 사용 불가능화』 및 151 페이지의 『Windows의 원격 API 클라이언트 사용 가능화』를 참조하십시오.

레지스트리에서 정보는 다음 키의 서브키 아래에 구성된 값에 포함됩니다.

```
\\HKEY_LOCAL_MACHINE\SOFTWARE\SNA Client\SxClient\Parameters
```

레지스트리 서브키마다 가능한 값은 다음과 같습니다.

```
Configuration
domain = domain_name
    maximum_process_count = nn
    maximum_header_count = nn
    maximum_element_count = nn
invoked_tps = YES | NO
lan_access_timeout = nn
broadcast_attempt_count = nn
server_lost_timeout = nn
    client_start_timeout = nn

Servers
Server1 = * | [ webservername : [ portnumber : ] ]servername1
Server2 = [ webservername : [ portnumber : ] ]servername2
```

```

.
.
.
Server9 = [ webservername : [ portnumber : ] ]servername9
Logging
exception_logging_enabled = YES | NO
audit_logging_enabled = YES | NO
log_directory = directory
error_file = error_filename
backup_error_file = backup_error_filename
error_file_wrap_size = error_file_size
audit_file = audit_filename
backup_audit_file = backup_audit_filename
audit_file_wrap_size = audit_file_size
succinct_errors = YES | NO
succinct_audits = YES | NO

API_tracing
file1 = trace_filename_1
file2 = trace_filename_2
flip_size = filesize
truncation_length = length
all_api = YES | NO
appc = YES | NO
cpic = YES | NO
csv = YES | NO
rui = YES | NO
nof = YES | NO

CS_tracing
file1 = cs_trace_filename_1
file2 = cs_trace_filename_2
flip_size = filesize
admin_msg = YES | NO
datagram = YES | NO
data = YES | NO
send = YES | NO
receive = YES | NO

Internal_tracing
file1 = internal_trace_filename_1
file2 = internal_trace_filename_2
flip_size = filesize
trace_level = nn
trace_flushing = YES | NO

Appl_Name
APPCTPN = tp_name
APPCLLU = lu_name

CSV_data
CSVTBLG = table_G_filename

```

주: domain = *domain_name* 값은 레지스트리에서 유일한 필수값입니다.

다음 절에서는 구성 매개변수를 설명합니다. 매개변수는 YES 또는 NO 값을 사용하므로, Y 또는 y로 시작하는 문자열은 YES로, N 또는 n으로 시작하는 문자열은 NO로 해석됩니다.

Configuration

Configuration 서브키에는 클라이언트에 대한 다음과 같은 구성 정보가 있습니다.

domain

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

domain_name 값은 클라이언트 설치 시 지정한 대로, Linux용 Communications Server LAN의 도메인 이름을 표시합니다. 이 행은 필수입니다.

maximum_process_count

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

이 클라이언트에서 한 번에 실행할 수 있는 최대 APPC, CPI-C, LUA 및 NOF 응용프로그램 수를 지정하십시오.

이 매개변수는 선택적입니다. 일반적으로 디폴트 값은 240이면 충분합니다. IPC 제어 블록 할당에 실패했다는 오류 메시지가 표시되면 이 매개변수를 지정하여 최대 프로세스 개수를 늘려야 합니다. 최대 1024까지 지정할 수 있습니다.

maximum_header_count, maximum_element_count

이러한 값의 레지스트리 데이터 유형은 REG_SZ입니다.

이 두 매개변수는 선택적입니다. 일반적으로 디폴트 값은 20000 및 30000이면 충분합니다. 지원 담당자의 지시가 있는 경우를 제외하고 이러한 매개변수의 값을 입력할 필요가 없습니다.

invoked_tps

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

다음 값 중 하나를 지정하십시오.

YES 이 클라이언트는 기동된 TP(RECEIVE_ALLOCATE를 발행하는 APPC TP나 Accept_Conversation 또는 Accept_Incoming을 발행하는 CPI-C 응용프로그램)를 실행하기 위해 사용됩니다. 이 경우 클라이언트에서 TP를 정의해야 할 수도 있습니다. 자세한 정보는 99 페이지의 『TP 정의』 또는 197 페이지의 부록 B 『명령행에서 기동 가능 TP 구성』을 참조하십시오.

NO 이 클라이언트는 기동된 TP를 실행하는 데 사용되지 않습니다.

이 행은 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 NO입니다.

lan_access_timeout

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

클라이언트의 어떤 응용프로그램도 Linux용 Communications Server 자원을 사용하지 않는 동안 클라이언트에서 서버로의 IP 또는 HTTPS 연결을 활동 중 상태로 유지해야 하는 시간(초)을 지정하십시오. 자세한 정보는 148 페이지의 『LAN 액세스 시간종료』를 참조하십시오.

유효 범위는 0-65535입니다. 최소 시간종료는 60초(더 낮은 값은 60초로 반올림됨)입니다. 연결을 더 빨리 비활성화하려면, 클라이언트를 사용 불가능하게 하십시오.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 시간종료를 사용하지 않는 것이므로 연결은 클라이언트가 실행되는 동안 활동 중 상태로 유지됩니다.

broadcast_attempt_count

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

클라이언트가 브로드캐스트 방법을 사용하여 서버(157 페이지의 『Servers』에 설명된 * 항목으로 지정된)에 접속할 경우, 이 매개변수는 서버에 접속하려는 한 번의 시도에서 수행할 최대 브로드캐스트 수를 지정합니다.

유효 범위는 1-65535입니다. 최소 값은 1입니다. 더 높은 값을 지정하면, 클라이언트는 서버에 접속할 때까지, 또는 이 계수에 도달할 때까지 10초마다 재시도합니다. 서버에 접속하지 못한 상태로 계수에 도달하면, 클라이언트는 이름 지정된 서버에 접속하려고 합니다(157 페이지의 『Servers』에 설명된 대로).

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 5입니다.

server_lost_timeout

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

클라이언트가 서버와의 접속이 끊어져서 다시 연결해야 하거나, 브로드캐스트 또는 이름 지정된 서버를 사용하여 서버에 접속하는 데(157 페이지의 『Servers』에 설명된 대로) 실패하면, 이 매개변수는 클라이언트가 서버 접속을 시도하기 전에 기다릴 시간(초)을 지정합니다. 클라이언트가 서버와의 접속이 끊어진 경우, Linux용 Communications Server는 전체 시간종료 시간 동안 기다리지 않지만 5초와 지정된 시간종료 값 사이의 임의의 시간 후 재시도합니다. 이렇게 하면 많은 클라이언트가 동시에 서버에 접속하려고 하여 발생하는 네트워크 트래픽 버스트를 피할 수 있습니다.

이 매개변수는 선택적입니다. 유효 범위는 5-65535입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 200(초)입니다.

client_start_timeout

이 값의 레지스트리 데이터 유형은 REG_DWORD입니다.

클라이언트가 서버 접속을 시작하고 재시도하는 동안 응용프로그램이 기다리는 시간(초)을 지정하십시오. 0 및 300 사이의 값이 유효합니다. 이 범위 밖의 값은 범위 내의 값이 되도록 조정됩니다. 디폴트 값은 10초입니다.

이 매개변수는 응용프로그램과 클라이언트 모두 시스템 시동 시 시작되도록 구성된 경우(시작 폴더에 놓거나 자동으로 시작되는 서비스로 만들어서) 이벤트를 제어하기 위해 사용할 수 있습니다. 응용프로그램은 이 필드에 지정된 초 수 동안 기다려서 클라이언트를 먼저 가져올 수 있도록 합니다. 이 방법에서, 클라이언트는 응용프로그램이 자원 부족으로 실패하기 전에 서버에 연결하여 응용프로그램이 필요로 하는 자원을 제공할 수 있습니다.

Servers

Servers 서브키에는 클라이언트가 자원에 액세스할 수 있는 Linux용 Communications Server 구성 서버에 대한 정보를 포함합니다. 이 목록에는 클라이언트와 같은 도메인에 백업 서버 및 마스터 구성 서버의 이름을 포함해야 합니다. 마스터 및 백업 서버 구성에 대한 정보는 61 페이지의 『클라이언트/서버 기능 구성』을 참조하십시오.

주: 이 서브키의 형식 및 의미는 클라이언트가 서버와 같은 사설 네트워크에 있는지 또는 아래 설명된 대로 HTTPS를 사용하여 공용 네트워크를 통해 연결되어 있는지에 따라 다릅니다.

Server1

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

별표(*)나 서버 이름을 입력하십시오.

- 서버와 동일한 사설 네트워크의 클라이언트가 TCP/IP 서브넷(또는 클라이언트 컴퓨터에 두 개 이상의 LAN 어댑터 카드가 있을 경우, TCP/IP 서브넷이 액세스할 수 있는 모든 서브넷)에 있는 모든 컴퓨터에 대해 UDP 브로드캐스트 메시지를 사용하여 Linux용 Communications Server를 실행 중인 서버에 접속해야 함을 표시하려면 *를 지정하십시오.

이 옵션은 클라이언트가 IPv4 주소지정을 사용할 경우에만 사용할 수 있습니다. IPv6에 대해 UDP 브로드캐스트는 지원되지 않습니다.

클라이언트는 서버에 접속할 때까지 *broadcast_attempt_count* 매개변수에 지정된 시도 횟수까지 10초마다 브로드캐스트를 재시도합니다. 서버에 접속되기 전에 *broadcast_attempt_count*에 지정한 한계에 도달하면, 클라이언트는 하나 이상의 이름 지정된 서버(파일의 다음 행에 지정된)에 대해 지시된 메시지 사용을 시도합니다.

- 클라이언트가 서버와 동일한 사설 네트워크에 있지만 UDP 브로드캐스트를 사용하여 어떤 서버에도 도달할 수 없어서 지정된 메시지를 사용해야 하는

경우 접속하려고 시도해야 하는 첫 번째 서버의 이름을 지정하십시오. *webservername* 및 *portnumber* 매개변수는 사용되지 않으며 지정하면 안 됩니다. 이는 다음 경우에 적용됩니다.

- 클라이언트가 IPv6 주소지정을 사용하는 경우(IPv6에 대해 UDP 브로드캐스트는 지원되지 않음)
- Linux용 Communications Server LAN이 여러 TCP/IP 서브넷에 분산되어 있고 클라이언트가 UDP를 사용하여 액세스할 수 있는 어떤 TCP/IP 서브넷에도 Linux용 Communications Server가 없을 경우
- 클라이언트에 UDP 지원이 설치되지 않은 경우

다른 경우에는 반드시 UDP 브로드캐스트를 사용하지 않아도 됩니다. 브로드캐스트를 시도하지 않도록 지정하려면 * 대신 첫 번째 서버의 이름을 지정하십시오.

- 클라이언트가 HTTPS를 사용하여 서버에 액세스하면 UDP 브로드캐스트가 지원되지 않습니다. 이 경우 HTTPS 지원을 제공하는 WebSphere 서버의 이름 및 Linux용 Communications Server의 이름을 다음 형식으로 지정하십시오.

webservername : *servername1*

여기서는 WebSphere가 HTTPS 연결에 디폴트 포트 443을 사용한다고 가정합니다. 네트워크 관리자가 다른 포트 번호를 사용하도록 WebSphere를 구성한 경우, 다음 형식으로 포트 번호를 포함하십시오.

webservername : *portnumber* : *servername1*

HTTPS 연결을 지원하도록 WebSphere를 구성하는 방법에 대한 자세한 정보는 *Linux용 IBM Communications Server* 빠른 시작을 참조하십시오.

- 주: UDP 브로드캐스트를 사용 중이 아닌 경우에는 *Server2-Server9* 매개변수를 사용하여 이 클라이언트가 액세스해야 하는 다른 서버의 이름을 지정해야 합니다. 클라이언트는 이 파일에 지정된 서버의 자원을 사용할 수 있으나 다른 서버의 자원은 사용할 수 없습니다.

Server2-Server9

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

이 클라이언트에서 사용하는 자원을 포함하는 추가 Linux용 Communications Server 이름을 지정하십시오. *Server1*과 같은 형식을 사용하십시오.

클라이언트가 UDP 브로드캐스트를 사용하여 서버에 접속하려고 시도하였지만 (또는 *Server1*에 지정된 서버에 접속하려고 했지만) 어떤 응답도 수신하지 못한 경우, 클라이언트는 지시된 메시지를 사용하여 *Server2*에 지정된 서버에 접속하려고 합니다. 이 접속이 실패하면, 클라이언트는 *Server3*에 지정된 서버에

접속하려고 합니다. 이 서버 이름들은 선택적이지만, 서버를 찾는 브로드캐스트 방법이 실패하거나 *Server1*에 지정된 서버를 사용할 수 없을 경우 백업 메커니즘을 제공해야 합니다.

클라이언트가 나열된 모든 서버에 접속을 시도했으나 성공하지 못한 경우, 클라이언트는 *server_lost_timeout* 매개변수에 지정된 시간(초) 동안 기다린 후 서버에 접속하려는 시도 프로세스를 다시 시작합니다(UDP 브로드캐스트를 사용하거나 나열된 첫 번째 서버를 사용하여).

Server2-Server9 매개변수는 *로 설정하여 UDP 브로드캐스트를 사용할 것을 표시할 수 없습니다. *Server1* 매개변수만 이를 표시하기 위해 사용할 수 있습니다. * 값은 파일에서 서버 이름 앞에 있어야 하기 때문입니다.

Logging

Logging 서브키는 클라이언트에 대한 로깅 옵션을 지정합니다. 이 옵션을 사용하여 도메인에 대해 전체적으로 지정된 로깅 옵션을 대체하는 클라이언트 로깅 설정을 지정할 수 있습니다. 도메인 로깅 옵션 지정에 대한 자세한 정보는 64 페이지의 『로깅 구성』을 참조하십시오.

중앙 로깅이 사용 가능할 경우, 모든 로그 메시지는 서버의 중앙 파일에 기록됩니다. 이 경우, 여기에 지정된 *exception_logging_enabled* 및 *audit_logging_enabled* 매개변수만 사용됩니다. 나머지 매개변수는 무시됩니다.

로깅 옵션은 다음과 같이 지정합니다.

exception_logging_enabled

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

이 매개변수를 다음 값 중 하나로 설정하십시오.

YES 예외 메시지를 기록합니다.

NO 예외 메시지를 기록하지 않습니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않으면, 클라이언트는 글로벌 도메인 설정을 사용하여 예외 메시지의 기록 여부를 판별합니다. (초기 디폴트는 예외 메시지를 기록하는 것입니다.)

audit_logging_enabled

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

이 매개변수를 다음 값 중 하나로 설정하십시오.

YES 감사 메시지를 기록합니다.

NO 감사 메시지를 기록하지 않습니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않으면, 클라이언트는 글로벌 도메인 설정을 사용하여 감사 메시지의 기록 여부를 판별합니다. (초기 디폴트는 감사 메시지를 기록하는 것입니다.)

log_directory

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

클라이언트에서 로그 파일이 저장되는 디렉토리의 전체 경로. 모든 로그 파일 및 백업 로그 파일(다음 매개변수에 지정된)이 이 디렉토리에 저장됩니다. Linux용 IBM Communications Server 진단 안내서에 설명된 로그 필터링 기능을 사용하는 경우, 이 기능을 제어하는 **logfilter.txt** 파일도 이 디렉토리에 저장됩니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 파일은 Windows 설치 디렉토리에 저장됩니다.

error_file

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

오류 메시지를 기록할 파일의 이름입니다. 이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 **sna.err**입니다.

오류 및 감사 메시지를 단일 파일에 기록하려면, 이 매개변수와 *audit_file* 매개변수 둘 다에 대해 같은 파일 이름을 지정하십시오.

backup_error_file

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

백업 오류 로그 파일의 이름입니다. 오류 로그 파일이 *error_file_wrap_size*에 지정한 크기에 도달하면 Linux용 Communications Server는 그 내용을 백업 파일에 복사한 후(기존 파일 위에 겹쳐쓰기) 오류 로그 파일을 지웁니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 **bak.err**입니다.

오류 및 감사 메시지를 단일 파일에 기록하려면, 이 매개변수와 *backup_audit_file* 매개변수 둘 다에 대해 같은 파일 이름을 지정하십시오.

error_file_wrap_size

이 값의 레지스트리 데이터 유형은 REG_DWORD입니다.

*error_file*에 지정한 로그 파일의 최대 크기입니다. 파일에 기록되는 메시지로 인해 파일 크기가 이 한계를 초과할 경우, Linux용 Communications Server는 로그 파일의 현재 내용을 백업 로그 파일에 복사한 후 로그 파일을 지웁니다. 이는 오류 로그 파일에 사용되는 최대 디스크 공간이 대략 *error_file_wrap_size* 매개변수 값의 두 배임을 의미합니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 1000000(바이트)입니다. 오류 및 감사 메시지를 같은 파일에 기록할 경우, 이 매개변수는 *audit_file_wrap_size* 매개변수와 같은 값으로 설정해야 합니다.

audit_file

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

감사 메시지를 기록할 파일의 이름입니다. 이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 **sna.aud**입니다.

오류 및 감사 메시지를 단일 파일에 기록하려면, 이 매개변수와 *error_file* 매개변수 둘 다에 대해 같은 파일 이름을 지정하십시오.

backup_audit_file

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

백업 감사 로그 파일의 이름입니다. 감사 로그 파일이 *audit_file_wrap_size*에 지정한 크기에 도달하면 Linux용 Communications Server는 그 내용을 백업 파일에 복사한 후(기존 파일 위에 겹쳐쓰기) 감사 로그 파일을 지웁니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 **bak.aud**입니다.

오류 및 감사 메시지를 단일 파일에 기록하려면, 이 매개변수와 *backup_error_file* 매개변수 둘 다에 대해 같은 파일 이름을 지정하십시오.

audit_file_wrap_size

이 값의 레지스트리 데이터 유형은 REG_DWORD입니다.

*audit_file*에 지정한 로그 파일의 최대 크기입니다. 파일에 기록되는 메시지로 인해 파일 크기가 이 한계를 초과할 경우, Linux용 Communications Server는 로그 파일의 현재 내용을 백업 로그 파일에 복사한 후 로그 파일을 지웁니다. 이는 감사 로그 파일에 사용되는 최대 디스크 공간이 대략 *audit_file_wrap_size* 매개변수 값의 두 배임을 의미합니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 1000000(바이트)입니다. 오류 및 감사 메시지를 같은 파일에 기록할 경우, 이 매개변수는 *error_file_wrap_size* 매개변수와 같은 값으로 설정해야 합니다.

succinct_errors

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

오류 로그 파일에서 간결 로깅 또는 상세 로깅 중 어느 유형을 사용할 것인지 지정합니다. 이 설정은 예외 로그와 문제점 로그 둘 다에 적용됩니다. 다음 값 중 하나를 지정할 수 있습니다.

YES 간결 로깅을 사용합니다. 로그 파일의 각 메시지에는 메시지 헤더 정보(예: 메시지 번호 및 로그 유형) 및 메시지 텍스트 문자열과 매개변

수의 요약이 들어 있습니다. 로그의 원인과 필요한 조치에 대해 자세히 보려면 Linux를 실행하는 컴퓨터에서 **snahelp** 유틸리티를 사용하면 됩니다.

NO 상세 로깅을 사용합니다. 로그 파일의 각 메시지는 메시지 헤더 정보, 메시지 텍스트 문자열 및 매개변수의 전체 리스트, 로그 원인과 필요한 조치에 대한 추가 정보가 포함됩니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 마스터 서버에 대해 실행된 이전 **set_global_log_type** 명령에서 취합니다. (또는 Motif 관리 프로그램을 사용하여 설정합니다.) **set_global_log_type** 명령이 실행되기 이전의 초기 디폴트는 간결 로깅을 사용하는 것입니다.

중앙 로깅을 사용할 경우, 모든 컴퓨터의 메시지에 대해 간결 로깅을 선택할 것인지 상세 로깅을 선택할 것인지의 여부는 중앙 로거로 작동하는 서버에서 이 매개변수를 설정하여 결정됩니다. 이 설정은 **set_global_log_type** 명령이나, 디폴트를 대체하기 위해 서버에 실행된 **set_log_type** 명령에서 수행할 수 있습니다.

succinct_audits

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

감사 로그 파일에서 간결 로깅 또는 상세 로깅 중 어느 로깅 유형을 사용할 것인지 지정합니다. 허용되는 값과 그 의미는 *succinct_errors* 매개변수와 같습니다.

API_tracing

API_tracing 서브키는 클라이언트에서 실행되는 응용프로그램에 대한 API 추적 옵션을 지정합니다. 추적에 대한 자세한 정보는 Linux용 IBM Communications Server 진단 안내서를 참조하십시오. 추적 옵션은 다음과 같이 지정합니다.

file1 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

추적 파일의 전체 경로 이름 또는 두 파일에 대해 추적할 경우 첫 번째 추적 파일의 전체 경로 이름(*file2* 매개변수 설명 참조).

이 매개변수는 사용자가 API 추적을 사용 가능하게 할 경우에 필요합니다.

file2 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

두 번째 추적 파일의 전체 경로 이름입니다. 이 매개변수는 선택적입니다. 추적이 두 파일 대신 하나의 파일에 대해 수행됨을 표시하려면 이 행을 포함하지 마십시오.

file1 및 *file2* 둘 다 지정할 경우, 추적은 두 파일에 대해 수행됩니다. 첫 번째 파일이 *flip_size* 매개변수에 지정된 크기에 도달하면, 두 번째 파일을 지우고 두 번째 파일로 추적을 계속합니다. 이 파일이 *flip_size*에 지정된 크기에 도달하면, 첫 번째 파일을 지우고 첫 번째 파일로 추적을 계속합니다. 이렇게 하면

초과 디스크 공간을 사용하지 않아도 장시간 추적을 계속할 수 있습니다. 필요한 최대 공간은 대략 *flip_size* 매개변수 값의 두 배입니다.

flip_size

이 값의 레지스트리 데이터 유형은 REG_DWORD입니다.

추적 파일의 최대 크기입니다. 두 개의 파일 이름을 지정하면, 추적은 현재 파일이 이 크기에 도달할 때 두 파일 사이에 전환됩니다. 하나의 파일 이름만 지정하면 이 매개변수는 무시되고 파일 크기에는 제한이 없습니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 1000000(바이트)입니다.

truncation_length

이 값의 레지스트리 데이터 유형은 REG_DWORD입니다.

각 메시지에 대해 추적 파일에 기록될 정보의 최대 길이입니다(바이트). 메시지가 이 길이보다 더 길면 Linux용 Communications Server는 메시지 시작 부분만 추적 파일에 기록하고 *truncation_length*를 벗어나는 데이터는 버립니다. 이렇게 하면 각 메시지에 대해 대부분의 중요한 정보를 기록하면서 파일이 긴 메시지로 채워지는 것을 방지할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않으면 Linux용 Communications Server는 메시지를 자르지 않습니다. (각 메시지의 모든 데이터가 파일에 기록됩니다.)

all_api

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

모든 API의 메시지를 추적하려면, 이 매개변수를 YES로 설정하십시오. 이 경우, Linux용 Communications Server는 *appc - nof*의 매개변수를 무시합니다.

모든 API에 대해 추적을 사용 불가능하게 하려면, *all_api* 및 *appc - nof*의 모든 매개변수를 NO로 설정하십시오.

특정 API에 해당되는 메시지만 추적하려면, *all_api*를 NO로 설정하고 *appc - nof*의 매개변수를 사용하여 추적할 API를 표시하십시오.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 NO입니다.

appc

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

APPC API 메시지를 추적하려면 이 매개변수를 YES로 설정하고, 그렇지 않으면 NO로 설정하십시오.

이 매개변수는 선택적입니다. 매개변수 값을 지정하지 않으면 디폴트 값은 NO입니다. *all_api* 매개변수를 YES로 설정하면 이 매개변수는 무시되고 APPC 메시지를 추적합니다.

- cpic* 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.
CPI-C API 메시지를 추적하려면 이 매개변수를 YES로 설정하고, 그렇지 않으면 NO로 설정하십시오.
이 매개변수는 선택적입니다. 매개변수 값을 지정하지 않으면 디폴트 값은 NO입니다. *all_api* 매개변수를 YES로 설정하면 이 매개변수는 무시되고 CPI-C 메시지를 추적합니다.
- csv* 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.
CSV API 메시지를 추적하려면 이 매개변수를 YES로 설정하고, 그렇지 않으면 NO로 설정하십시오.
이 매개변수는 선택적입니다. 매개변수 값을 지정하지 않으면 디폴트 값은 NO입니다. *all_api* 매개변수를 YES로 설정하면 이 매개변수는 무시되고 CSV 메시지를 추적합니다.
- rui* 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.
LUA RUI 메시지를 추적하려면 이 매개변수를 YES로 설정하고, 그렇지 않으면 NO로 설정하십시오.
이 매개변수는 선택적입니다. 매개변수 값을 지정하지 않으면 디폴트 값은 NO입니다. *all_api* 매개변수를 YES로 설정하면 이 매개변수는 무시되고 LUA RUI 메시지를 추적합니다.
- nof* 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.
NOF API 메시지를 추적하려면 이 매개변수를 YES로 설정하고 그렇지 않으면 NO로 설정하십시오. NOF 메시지는 Windows 클라이언트의 응용프로그램에서 직접 사용되지는 않지만 Linux용 Communications Server 구성요소에서 구성 정보를 얻는 데 내부적으로 사용됩니다.
이 매개변수는 선택적입니다. 매개변수 값을 지정하지 않으면 디폴트 값은 NO입니다. *all_api* 매개변수를 YES로 설정하면 이 매개변수는 무시되고 NOF 메시지를 추적합니다.

CS_tracing

CS_tracing 서브키는 클라이언트/서버 추적(클라이언트 및 Linux용 Communications Server 사이의 메시지 추적)에 대한 옵션을 지정합니다. 추적에 대한 자세한 정보는 *Linux용 IBM Communications Server* 진단 안내서를 참조하십시오. 추적 옵션은 다음과 같이 지정합니다.

- file1* 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.
추적 파일의 전체 경로 이름 또는 두 파일에 대해 추적할 경우 첫 번째 추적 파일의 전체 경로 이름(*file2* 매개변수 설명 참조).

이 매개변수는 클라이언트/서버 추적을 사용 가능하게 할 경우에 필요합니다. *trace_flags* 매개변수도 설정해야 합니다.

file2 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

두 번째 추적 파일의 전체 경로 이름입니다. 이 매개변수는 선택적입니다. 추적이 두 파일 대신 하나의 파일에 대해 수행됨을 표시하려면 이 행을 포함하지 마십시오.

file1 및 *file2* 둘 다 지정할 경우, 추적은 두 파일에 대해 수행됩니다. 첫 번째 파일이 *flip_size* 매개변수에 지정된 크기에 도달하면, 두 번째 파일을 지우고 두 번째 파일로 추적을 계속합니다. 이 파일이 *flip_size*에 지정된 크기에 도달하면, 첫 번째 파일을 지우고 첫 번째 파일로 추적을 계속합니다. 이렇게 하면 초과 디스크 공간을 사용하지 않아도 장시간 추적을 계속할 수 있습니다. 필요한 최대 공간은 대략 *flip_size* 매개변수 값의 두 배입니다.

flip_size

이 값의 레지스트리 데이터 유형은 REG_DWORD입니다.

추적 파일의 최대 크기입니다. 두 개의 파일 이름을 지정하면, 추적은 현재 파일이 이 크기에 도달할 때 두 파일 사이에 전환됩니다. 하나의 파일 이름만 지정하면 이 매개변수는 무시되고 파일 크기에는 제한이 없습니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 1000000(바이트)입니다.

admin_msg

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

클라이언트/서버 토폴로지에 관련되는 내부 메시지를 추적하려면 이 매개변수를 YES로 설정하고, 그렇지 않으면 NO로 설정하십시오.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 NO입니다.

datagram

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

데이터그램 메시지를 추적하려면 이 매개변수를 YES로 설정하고, 그렇지 않으면 NO로 설정하십시오.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 NO입니다.

data 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

데이터 메시지를 추적하려면 이 매개변수를 YES로 설정하고, 그렇지 않으면 NO로 설정하십시오.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 NO입니다.

send 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

클라이언트에서 서버로 전송된 모든 데이터 메시지를 추적하려면 이 매개변수를 YES로 설정하고, 그렇지 않으면 NO로 설정하십시오.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 NO입니다.

receive

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

클라이언트가 서버로부터 수신한 모든 데이터 메시지를 추적하려면 이 매개변수를 YES로 설정하고, 그렇지 않으면 NO로 설정하십시오.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 NO입니다.

Internal_tracing

Internal_tracing 서브키는 클라이언트의 내부 조작을 추적하기 위한 옵션을 지정합니다. 추적에 대한 자세한 정보는 *Linux용 IBM Communications Server 진단 안내서*를 참조하십시오. 추적 옵션은 다음과 같이 지정합니다.

file1 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

추적 파일의 전체 경로 이름 또는 두 파일에 대해 추적할 경우 첫 번째 추적 파일의 전체 경로 이름(*file2* 매개변수 설명 참조).

이 매개변수는 내부 추적을 사용 가능하게 할 경우에 필요합니다. *trace_level* 매개변수도 설정해야 합니다.

file2 이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

두 번째 추적 파일의 전체 경로 이름입니다. 이 매개변수는 선택적입니다. 추적이 두 파일 대신 하나의 파일에 대해 수행됨을 표시하려면 이 행을 포함하지 마십시오.

file1 및 *file2* 둘 다 지정할 경우, 추적은 두 파일에 대해 수행됩니다. 첫 번째 파일이 *flip_size* 매개변수에 지정된 크기에 도달하면, 두 번째 파일을 지우고 두 번째 파일로 추적을 계속합니다. 이 파일이 *flip_size*에 지정된 크기에 도달하면, 첫 번째 파일을 지우고 첫 번째 파일로 추적을 계속합니다. 이렇게 하면 초과 디스크 공간을 사용하지 않아도 장시간 추적을 계속할 수 있습니다. 필요한 최대 공간은 대략 *flip_size* 매개변수 값의 두 배입니다.

flip_size

이 값의 레지스트리 데이터 유형은 REG_DWORD입니다.

추적 파일의 최대 크기입니다. 두 개의 파일 이름을 지정하면, 추적은 현재 파일이 이 크기에 도달할 때 두 파일 사이에 전환됩니다. 하나의 파일 이름만 지정하면 이 매개변수는 무시되고 파일 크기에는 제한이 없습니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 1000000(바이트)입니다.

trace_level

이 값의 레지스트리 데이터 유형은 REG_DWORD입니다.

추적에 포함할 세부사항의 레벨입니다. 유효값 범위는 0(모두 추적) - 20(추적 안함)입니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 20(추적 안함)입니다.

trace_flushing

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

이 매개변수를 YES로 설정할 경우, 각 추적 명령문은 즉시 디스크로 플러시됩니다. 이렇게 되면 조작성이 어느 정도 느려질 수 있지만 충돌이 발생할 경우 추적 데이터가 유실되지 않습니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 NO입니다.

Appl_Name

Appl_Name 서브키는 CPI-C 응용프로그램의 옵션을 지정합니다.

주: 이러한 옵션은 환경 변수 또는 레지스트리에서 지정할 수 있습니다. Linux용 Communications Server는 먼저 환경 변수를 확인하고 지정되어 있을 경우 이 정보를 사용합니다. 환경 변수가 지정되어 있지 않은 경우에만 레지스트리 항목을 사용합니다. Windows 터미널 서버를 사용하고 다른 로컬 LU를 사용하여 동일한 응용프로그램의 여러 사본을 실행할 경우 환경 변수를 사용해야 할 수 있습니다.

레지스트리에서 하나 이상의 응용프로그램에 대해 이 옵션을 설정하려면, 응용프로그램마다 해당 형식으로 섹션을 포함시키고 *Appl_Name* 변수를 응용프로그램의 실행 파일 이름(.exe 파일 이름 확장자는 제외하고)으로 대체하십시오.

CPI-C에 대한 자세한 정보는 AIX 또는 Linux용 IBM Communications Server CPI-C 프로그래머 안내서를 참조하십시오.

옵션은 다음과 같이 지정합니다.

APPCLLU

이 옵션은 레지스트리 대신 APPCLLU 환경 변수를 사용하여 지정할 수 있습니다.

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

응용프로그램이 사용하는 로컬 LU의 이름입니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 응용프로그램은 디폴트 LU(로컬 노드의 제어점과 연관되는 LU)를 사용하려고 합니다.

APPCTPN

이 옵션은 레지스트리 대신 APPCTPN 환경 변수를 사용하여 지정할 수 있습니다.

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

응용프로그램의 TP 이름입니다. 이 이름은 응용프로그램을 식별하기 위해 로그 및 추적 파일에서 사용됩니다. 기동된 응용프로그램(issues Accept_Conversation을 발행한 응용프로그램)의 경우, 이 이름은 또한 올바른 응용프로그램과 입력되는 Allocate 요청의 TP 이름을 일치시키는 데도 사용됩니다. 기동된 응용프로그램은 또한 Specify_Local_TP_Name 호출을 사용하여 입력되는 Allocate 요청과 일치될 추가 이름을 지정할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 CPIC_DEFAULT_TPNAME입니다.

CSV_data

CSV_data 서브키는 CSV 인터페이스를 사용하는 응용프로그램의 옵션을 지정합니다. 이 서브키는 CONVERT Verb를 사용하여 사용자 정의 변환 테이블(테이블 G)로 문자 변환을 수행하는 응용프로그램에만 적용됩니다. CONVERT Verb에 대한 자세한 정보는 AIX 또는 Linux용 IBM Communications Server CSV 프로그래머 안내서를 참조하십시오.

클라이언트의 어떤 응용프로그램도 이 기능을 사용하지 않을 경우, 이 섹션을 포함시키지 않아도 됩니다.

이 섹션에서 유일한 옵션은 다음과 같습니다.

CSVTLG

이 값의 레지스트리 데이터 유형은 REG_SZ입니다.

사용자 정의 테이블 G 변환 테이블을 포함하는 파일의 전체 경로 이름입니다. 이 매개변수는 CSV 응용프로그램이 테이블 G 문자 변환을 수행해야 할 경우 필수입니다(디폴트 없음). 그렇지 않으면 선택적입니다.



AIX 또는 Linux의 원격 API 클라이언트 관리

AIX, LINUX

원격 API 클라이언트는 AIX, Linux, pSeries용 Linux 또는 System z용 Linux에서 실행할 수 있습니다.

AIX 또는 Linux의 원격 API 클라이언트에 대한 클라이언트 정보는 클라이언트에서 SNA 소프트웨어를 설치할 때 작성되는 **sna_clnt.net** 파일에 저장됩니다. 그 파일이 있어야 클라이언트 소프트웨어를 사용 가능하게 할 수 있습니다.

주: 클라이언트가 HTTPS를 사용하여 서버에 액세스할 경우 **sna_clnt.net** 파일을 수정하여 HTTPS 액세스를 제공하는 WebSphere 서버와 이러한 서버의 이름을 지정해야 클라이언트를 사용할 수 있습니다. 자세한 내용은 170 페이지의 『클라이언트 네트워크 데이터 파일(sna_clnt.net)』을 참조하십시오.

AIX 또는 Linux의 원격 API 클라이언트 사용 가능화 및 사용 불가능화

AIX 또는 Linux에서 원격 API 클라이언트 소프트웨어를 사용 가능하게 하려면 명령 프롬프트에 다음 명령을 입력하십시오.

```
sna start [ -t ]
```

클라이언트를 설치할 때 설치 유틸리티는 시작 파일 **/etc/rc.sna**(AIX) 또는 **/etc/rc.d/init.d/snastart**(Linux)를 자동으로 갱신하여 **sna start** 명령을 포함시킵니다. 이것은 시스템이 시작될 때 클라이언트가 자동으로 시작되도록 합니다. 자동으로 시작되지 않도록 하려면 이 행을 제거하거나 주석으로 표시한 다음 이 절에 있는 지시사항을 수행하여 소프트웨어를 수동으로 사용 가능하게 하십시오.

유일한 옵션은 다음과 같습니다.

-t 클라이언트/서버 추적을 활성화합니다. 이 옵션을 사용하면 클라이언트가 서버에 연결하려고 시도하는 동안 발생하는 문제점을 진단할 수 있습니다. 이 옵션을 사용하지 않으면 모든 인터페이스에서 클라이언트/서버 추적이 비활성화됩니다. 이 경우 사용자는 명령행 관리 프로그램 **snaadmin**을 사용하여 필요에 따라 추적을 활성화할 수 있습니다.

이 옵션은 DLC 추적을 사용 가능하게 하지 않는 것을 제외하고, Motif 관리 프로그램에서 모든 추적 설정 필드를 선택하는 것과 같습니다.

추적을 수행하면 Linux용 Communications Server 구성요소의 성능이 저하됩니다. 소프트웨어가 사용 가능하게 되면, 명령행 관리 프로그램 **snaadmin**을 사

용하여 추적이 필요하지 않을 때 추적을 중지할 수 있습니다. 추적에 대한 자세한 정보는 *Linux*용 *IBM Communications Server* 진단 안내서를 참조하십시오.

원격 API 클라이언트를 중지하려면 명령 프롬프트에 다음 명령을 입력하십시오.

```
sna stop
```

클라이언트 네트워크 데이터 파일(*sna_clnt.net*)

sna_clnt.net 파일은 AIX 또는 Linux의 원격 API 클라이언트에서 사용 가능한 Linux용 Communications Server 기능 및 클라이언트가 액세스할 수 있는 서버를 정의합니다. (클라이언트에 해당되는 파일에 대한 정보는 143 페이지의 제 10 장 『Linux용 Communications Server 클라이언트/서버 시스템 관리』를 참조하십시오.)

또한 Linux용 Communications Server가 클라이언트/서버 통신에 대해 사용하는 IP 포트 번호 설정에 대한 정보도 포함됩니다. 디폴트 포트 번호는 대부분의 경우에 적합해야 합니다. Linux용 Communications Server가 Linux 컴퓨터에 다른 프로그램과의 포트 이름 충돌이 있음을 표시하는 오류 메시지를 기록할 경우에만 이 정보를 참조해야 합니다.

클라이언트 컴퓨터는 도메인 구성 파일이나 SNA 네트워크 데이터 파일의 사본을 보유하지 않습니다. 컴퓨터는 Linux용 Communications Server LAN의 서버에 액세스하기 위해 필요하고, 필요한 구성 정보를 제공하기 위해 서버에 의존하는 정보만 보유합니다.

필요한 SNA 네트워크 정보는 AIX의 */etc/sna* 또는 Linux의 */etc/opt/ibm/sna* 디렉토리에 저장되는 *sna_clnt.net* 파일에 보유됩니다. 이 파일은 클라이언트 설치 프로세스에서 설정되며, 필요에 따라 표준 텍스트 편집기를 사용하여 나중에 수정할 수 있는 ASCII 텍스트 파일입니다.

주: 이 파일의 매개변수를 변경한 후 클라이언트를 중지하고 다시 시작해야 변경사항이 적용됩니다. 자세한 내용은 169 페이지의 『AIX 또는 Linux의 원격 API 클라이언트 사용 가능화 및 사용 불가능화』를 참조하십시오.

파일 내용은 다음과 같습니다.

```
domain = domain_name
maximum_process_count = nn
maximum_header_count = nn
maximum_element_count = nn
invoked_tps = YES | NO
lan_access_timeout = nn
broadcast_attempt_count = nn
server_lost_timeout = nn
*
[ webservername : [ portnumber : ] ]servername1
```



```
[ webservername : [ portnumber : ] ]servername2
.
.
.
```

다음 리스트는 파일의 각 행에 있는 매개변수를 설명합니다.

domain

domain_name 매개변수 값은 Linux용 Communications Server LAN의 도메인 이름을 표시합니다. 이 이름은 클라이언트 설치 시 `ibmcs_domain`에 설정됩니다. 이 행은 필수입니다.

maximum_process_count

이 클라이언트에서 한 번에 실행할 수 있는 최대 APPC, CPI-C, LUA 및 NOF 응용프로그램 수를 지정하십시오.

이 매개변수는 선택적입니다. 일반적으로 디폴트 값은 240이면 충분합니다. IPC 제어 블록 할당에 실패했다는 오류 메시지가 표시되면 이 매개변수를 지정하여 최대 프로세스 개수를 늘려야 합니다. 최대 4096까지 지정할 수 있습니다.

maximum_header_count, maximum_element_count

이 두 매개변수는 선택적입니다. 일반적으로 디폴트 값은 1250 및 1800이면 충분합니다. 지원 담당자의 지시가 있는 경우를 제외하고 이러한 매개변수의 값을 입력할 필요가 없습니다.

invoked_tps

클라이언트가 기동된 TP(RECEIVE_ALLOCATE Verb를 발행하는 APPC TP나 Accept_Conversation 또는 Accept_Incoming Verb를 발행하는 CPI-C 응용프로그램)를 실행하기 위해 사용될 경우 `invoked_tps = YES`를 지정하십시오. 이 경우 클라이언트에서 TP를 정의해야 할 수도 있습니다. 자세한 정보는 99 페이지의 『TP 정의』를 참조하십시오.

클라이언트가 기동된 TP를 실행하는 데 사용되지 않을 경우 `invoked_tps = NO`를 지정하십시오.

이 행은 선택적입니다. 이 행을 포함하지 않을 경우, 디폴트는 NO입니다.

lan_access_timeout

클라이언트의 어떤 응용프로그램도 Linux용 Communications Server 자원을 사용하지 않는 동안 클라이언트에서 서버로의 IP 또는 HTTPS 연결을 활동 중 상태로 유지해야 하는 시간(초)을 지정하십시오. 자세한 정보는 148 페이지의 『LAN 액세스 시간종료』를 참조하십시오.

최소 시간종료는 60초(더 낮은 값은 60초로 반올림됨)입니다. 연결을 더 빨리 중단하려면 클라이언트에서 Linux용 Communications Server 소프트웨어를 사용 불가능하게 하십시오.

시간종료가 사용되지 않음을 표시하여 Linux용 Communications Server 소프트웨어가 클라이언트에서 실행 중일 때 연결이 활동 중 상태로 유지되도록 하려면, 이 매개변수를 지정하지 마십시오.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 시간종료를 사용하지 않는 것입니다.

broadcast_attempt_count

클라이언트가 브로드캐스트 방법을 사용하여 서버(* 항목으로 지정된)에 접속할 경우, 이 매개변수는 서버에 접속하려는 한 번의 시도에서 수행할 최대 브로드캐스트 수를 지정합니다. 최소 값은 1입니다. 더 높은 값을 지정하면, 클라이언트는 서버에 접속할 때까지, 또는 이 계수에 도달할 때까지 10초마다 재시도합니다. 서버에 접속하지 못한 상태로 계수에 도달하면, 클라이언트는 이름 지정된 서버에 접속하려고 합니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 5입니다.

server_lost_timeout

클라이언트가 서버와의 접속이 끊어져서 다시 연결해야 하거나, 브로드캐스트 또는 이름 지정된 서버를 사용하여 서버에 접속하는 데 실패하면, 이 매개변수는 클라이언트가 서버 접속 시도를 시작하거나 다시 시작하기 전에 기다릴 시간(초)을 지정합니다. 클라이언트가 서버와의 접속이 끊어진 경우, Linux용 Communications Server는 전체 시간종료 시간 동안 기다리지 않지만 5초와 지정된 시간종료 값 사이의 임의 시간 후 재시도합니다. 이렇게 하면 많은 클라이언트가 동시에 서버에 접속하려고 하여 발생하는 네트워크 트래픽 버스트를 피할 수 있습니다.

이 매개변수는 선택적입니다. 이 매개변수를 지정하지 않을 경우, 디폴트는 200초입니다.

- * 이 행은 서버와 동일한 사설 네트워크의 클라이언트가 TCP/IP 서브넷(또는 클라이언트 컴퓨터에 두 개 이상의 LAN 어댑터 카드가 있을 경우, TCP/IP 서브넷이 액세스할 수 있는 모든 서브넷)에 있는 모든 컴퓨터에 대해 UDP 브로드캐스트 메시지를 사용하여 Linux용 Communications Server를 실행 중인 서버에 접속해야 함을 표시합니다.

이 옵션은 클라이언트가 IPv4 주소지정을 사용할 경우에만 사용할 수 있습니다. IPv6에 대해 UDP 브로드캐스트는 지원되지 않습니다.

클라이언트는 서버에 접속할 때까지 *broadcast_attempt_count* 매개변수에 지정된 시도 횟수까지 10초마다 브로드캐스트를 재시도합니다. 서버에 접속되기 전에 *broadcast_attempt_count*에 지정한 한계에 도달하면, 클라이언트는 하나 이상의 이름 지정된 서버(파일의 다음 행에 지정된)에 대해 지시된 메시지 사용을 시도합니다.

UDP 브로드캐스트의 사용이 선택적입니다. 브로드캐스트를 시도하지 않도록 지정하려면 이 행을 포함하지 마십시오. 이 행을 포함할 경우, 파일에서 서버 이름을 앞에 와야 합니다.

클라이언트가 서버와 동일한 사설 네트워크에 있지만 UDP 브로드캐스트를 사용하여 어떤 서버에도 도달할 수 없는 경우, 이 행을 포함하지 마십시오. 이는 다음 경우에 적용됩니다.

- Linux용 Communications Server LAN이 여러 TCP/IP 서브넷에 분산되어 있고 클라이언트가 UDP를 사용하여 액세스할 수 있는 어떤 TCP/IP 서브넷에도 Linux용 Communications Server가 없을 경우
- 클라이언트에 UDP 지원이 설치되지 않은 경우

클라이언트가 HTTPS를 사용하여 서버에 액세스하면 UDP 브로드캐스트가 지원되지 않습니다. 이 경우 아래 설명한 대로 서버 이름을 명확히 지정하십시오.

주: UDP 브로드캐스트를 사용 중이 아닌 경우에는 아래 설명에 따라 이 클라이언트가 액세스해야 하는 다른 서버의 이름을 지정해야 합니다. 클라이언트는 지정된 서버의 자원을 사용할 수 있으나 다른 서버의 자원은 사용할 수 없습니다.

server names

이 클라이언트에서 사용하는 자원을 포함하는 하나 이상의 Linux용 Communications Server 이름을 지정하십시오. 이 목록에는 클라이언트와 같은 도메인에 백업 서버 및 마스터 구성 서버의 이름을 포함해야 합니다. 마스터 및 백업 서버 구성에 대한 정보는 61 페이지의 『클라이언트/서버 기능 구성』을 참조하십시오. 또한 이 클라이언트가 액세스해야 하는 다른 모든 서버의 이름을 포함해야 합니다. 클라이언트는 지정된 서버의 자원을 사용할 수 있으나 다른 서버의 자원은 사용할 수 없습니다.

클라이언트가 IPv6 주소지정을 사용하는 경우 UDP 브로드캐스트가 지원되지 않습니다. * 옵션을 사용하는 대신 최소 하나의 서버 이름을 지정해야 합니다.

클라이언트가 HTTPS를 사용하여 서버에 액세스하면 UDP 브로드캐스트가 지원되지 않습니다. 이 경우 HTTPS 지원을 제공하는 WebSphere 서버의 이름 및 Linux용 Communications Server의 이름을 다음 형식으로 지정하십시오.

webservername : servername1

여기서는 WebSphere가 HTTPS 연결에 디폴트 포트 443을 사용한다고 가정합니다. 네트워크 관리자가 다른 포트 번호를 사용하도록 WebSphere를 구성한 경우, 다음 형식으로 포트 번호를 포함하십시오.

webservername : portnumber : servername1

HTTPS 연결을 지원하도록 WebSphere를 구성하는 방법에 대한 자세한 정보는 Linux용 IBM Communications Server 빠른 시작을 참조하십시오.

* 행(UDP 브로드캐스트의 사용을 표시하기 위한)이 없거나 클라이언트가 이 방법을 사용하여 서버에 접속하려고 했지만 응답을 수신하지 못한 경우, 클라이언트는 지시된 메시지를 사용하여 나열된 첫 번째 서버에 접속하려고 합니다. 이 접속이 실패하면, 클라이언트는 나열된 두 번째 서버에 접속하려고 합니다. 즉, 서버가 나열된 순서를 변경하여 두 개 이상의 구성 서버 간 밸런스를 조절할 수 있습니다.

* 행(UDP 브로드캐스트 사용을 표시하기 위한)이 없을 경우, 최소 하나의 서버 이름을 지정해야 합니다. 그렇지 않으면 서버 이름은 선택적입니다.

클라이언트가 나열된 모든 서버에 접속을 시도했지만 성공하지 못한 경우, 클라이언트는 *server_lost_timeout* 매개변수에 지정된 시간(초) 동안 기다린 후 서버에 접속하려는 시도 프로세스를 다시 시작합니다(UDP 브로드캐스트를 사용하거나 나열된 첫 번째 서버를 사용하여).

sna_clnt.net뿐만 아니라 추가 파일 **server.current**도 같은 이름의 디렉토리(AIX의 */var/sna* 또는 Linux의 */var/opt/ibm/sna*)에 저장됩니다. 이것은 클라이언트가 현재 연결된 서버(있을 경우) 이름을 포함하는 텍스트 파일입니다. 이 파일에서 도메인에서 클라이언트의 연결 지점 역할을 하는 서버를 확인할 수 있습니다.



클라이언트 TP 정의

원격 API 클라이언트 시스템에서의 TP 정의에 대한 정보는 99 페이지의 『TP 정의』 또는 197 페이지의 부록 B 『명령행에서 기동 가능 TP 구성』을 참조하십시오.

부록 A. 구성 계획 워크시트

이 부록은 Linux용 Communications Server의 특정 기능을 구성하기 위한 워크시트를 제공합니다. 워크시트는 각 기능을 사용 가능하게 하는 데 필요한 기본 구성 매개변수를 요약합니다. 고급 구성 매개변수에 대한 정보는 이 책의 본문에서 해당 절을 참조하거나 *Linux용 IBM Communications Server 관리 명령 참조서*를 참조하십시오.

노드를 구성하는 데 필요한 모든 정보를 수집하려면 다음 범주의 워크시트를 완료해야 합니다.

노드 구성

노드 성능과 노드가 작동하는 네트워크의 특성에 따라 『노드 워크시트』에 있는 워크시트 중 하나를 완료하십시오.

연결성 구성

네트워크에서 다른 시스템과 통신하기 위해 사용하는 링크 프로토콜에 따라 178 페이지의 『연결성 워크시트』에 포함된 워크시트 중 하나 이상을 완료하십시오.

패스-스루 서비스 구성

노드에서 지원하는 패스-스루 서비스에 대해 187 페이지의 『패스-스루 서비스 워크시트』에 있는 워크시트를 완료하십시오.

응용프로그램 지원 구성

노드에서 지원하는 사용자 응용프로그램 유형에 따라 190 페이지의 『사용자 응용프로그램 지원 워크시트』에 포함된 워크시트 중 하나 이상을 완료하십시오.

노드 워크시트

다음 워크시트 중 하나 이상을 완료하십시오.

- 『APPN 네트워크 노드』
- 176 페이지의 『APPN 끝 노드』
- 176 페이지의 『APPN 브랜치 네트워크 노드』
- 177 페이지의 『LEN 노드』

APPN 네트워크 노드

로컬 노드가 APPN 네트워크 노드(APPN 네트워크에서 경로지정 서비스를 제공하는 노드)일 경우 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 매개변수 대화상자		
APPN 지원	네트워크 노드	

Motif 필드	유효한 항목/주의사항	사용자 구현값
제어점 이름	NETNAME.CPNAME(각각 1-8자의 유형 A EBCDIC 문자)	
제어점 별명	최대 8자	
노드 ID	8자리의 16진수	
연결성 구성: 178 페이지의 『연결성 워크시트』를 참조하십시오.		
클라이언트/서버 구성: 독립형 노드에는 필요하지 않습니다.		
구성 서버?	Linux용 Communications Server LAN에서 도메인 자원에 대한 정보를 저장하려면 노드가 구성 서버로 작동해야 합니까?	
응용프로그램 구성: 190 페이지의 『사용자 응용프로그램 지원 워크시트』를 참조하십시오.		

APPN 끝 노드

로컬 노드가 APPN 끝 노드(동적 경로지정 정보를 사용할 수 있지만 다른 노드에 대한 경로지정 서비스는 제공하지 않는 노드)일 경우 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 매개변수 대화상자		
APPN 지원	끝 노드	
제어점 이름	NETNAME.CPNAME(각각 1-8자의 유형 A EBCDIC 문자)	
제어점 별명	최대 8자	
노드 ID	8자리의 16진수	
연결성 구성: 178 페이지의 『연결성 워크시트』를 참조하십시오.		
클라이언트/서버 구성: 독립형 노드에는 필요하지 않습니다.		
구성 서버?	Linux용 Communications Server LAN에서 도메인 자원에 대한 정보를 저장하려면 노드가 구성 서버로 작동해야 합니까?	
응용프로그램 구성: 190 페이지의 『사용자 응용프로그램 지원 워크시트』를 참조하십시오.		

APPN 브랜치 네트워크 노드

로컬 노드가 APPN 브랜치 네트워크 노드(기본 APPN 네트워크에서 구분된 브랜치에 있는 끝 노드에 네트워크 노드 기능을 제공하지만 기본 네트워크 자체에서는 끝 노드의 역할을 수행하는 노드)일 경우 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 매개변수 대화상자		
APPN 지원	브랜치 네트워크 노드	
제어점 이름	NETNAME.CPNAME(각각 1-8자의 유형 A EBCDIC 문자)	
제어점 별명	VTAM 호스트에 연결하려면 이 이름이 VTAM PU문의 NETID= 및 CPNAME= 항목과 일치해야 합니다.	
노드 ID	최대 8자	
연결성 구성: 178 페이지의 『연결성 워크시트』를 참조하십시오.	8자리의 16진수	
클라이언트/서버 구성: 독립형 노드에는 필요하지 않습니다.		
구성 서버?	Linux용 Communications Server LAN에서 도메인 자원에 대한 정보를 저장하려면 노드가 구성 서버로 작동해야 합니까?	
응용프로그램 구성: 190 페이지의 『사용자 응용프로그램 지원 워크시트』를 참조하십시오.		

LEN 노드

로컬 노드가 LEN 노드(APPN 기능을 지원하지 않는 노드나 호스트 컴퓨터와만 통신하는 독립형 시스템)일 경우 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 매개변수 대화상자		
APPN 지원	LEN 노드	
제어점 이름	NETNAME.CPNAME(각각 1-8자의 유형 A EBCDIC 문자)	
제어점 별명	VTAM 호스트에 연결하려면 이 이름이 VTAM PU문의 NETID= 및 CPNAME= 항목과 일치해야 합니다.	
노드 ID	최대 8자	
연결성 구성: 178 페이지의 『연결성 워크시트』를 참조하십시오.	8자리의 16진수	
클라이언트/서버 구성: 독립형 노드에는 필요하지 않습니다.		
구성 서버?	Linux용 Communications Server LAN에서 도메인 자원에 대한 정보를 저장하려면 노드가 구성 서버로 작동해야 합니까?	
응용프로그램 구성: 190 페이지의 『사용자 응용프로그램 지원 워크시트』를 참조하십시오.		

연결성 워크시트

다른 노드와 통신하기 위해 사용하는 링크 프로토콜마다 다음 워크시트 중 하나를 완료하십시오. 필요하다면 하나의 포트에서 여러 개의 링크 스테이션을 구성할 수 있습니다.

- 『SDLC』
- 180 페이지의 『토큰링』
- 182 페이지의 『이더넷』
- 183 페이지의 『QLLC(X.25)』
- 185 페이지의 『다중 경로 채널』
- 185 페이지의 『Enterprise Extender(HPR/IP)』

SDLC

SDLC 링크 프로토콜을 사용하는 연결성을 지원하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
SDLC 포트 대화상자		
SNA 포트 이름	최대 8자	
SDLC 카드 번호	0 - <i>number_of_cards_minus_1</i>	
포트 번호	0 - <i>number_of_ports_on_card_minus_1</i>	
초기에 활동 중	필요할 경우 선택	
회선 세부사항		
유형	전용 회선	
	교환식 전송	
링크 역할	교환식 입력 조정 가능	
	1차	
	1차 분산 분기 회선	
	2차	
	2차 다중 PU	
교환식 입력 또는 전용 회선의 경우:		
폴 주소	1차가 아닌 교환식 입력 포트의 경우에만	
	VTAM 호스트에서 폴 주소는 VTAM PU 정의에서 ADDR= 매개변수로 구성됩니다.	
	AS/400 시스템에서 폴 주소는 회선 설명의 STNADR 매개변수입니다.	
SDLC 링크 스테이션 대화상자		
링크 스테이션 필드		

Motif 필드	유효한 항목/주의사항	사용자 구현값
이름 SNA 포트 이름 활성화	최대 8자 최대 8자 관리자에 의해	
LU 트래픽	노드 시동 시 요구가 있을 때 무관함 독립형만 종속형만	
독립 LU 트래픽		
원격 노드	NETNAME.CPNAME(각각 1-8자의 유형 A EBCDIC 문자. 선택적)	
원격 노드 유형	원격 시스템이 VTAM 호스트일 경우, VTAM 시작 리스트의 NETID 매개변수에서 네트워크 이름(전체 이름의 처음 8자)을 찾 을 수 있습니다. 마지막 8자는 VTAM 시작 리스트의 SSCPNAME 매개변수에 있습니다. 검색 네트워크 노드 끝 또는 LEN 노드	
종속 LU 트래픽		
원격 노드 역할	호스트 다운스트림 (SNA 게이트웨이)	
로컬 노드 ID	다운스트림 (DLUR) 8자리의 16진수(다폴트는 노드 이름) VTAM 구성에서 처음 세 자리 숫자는 PU 정의의 IDBLK 매개변수와 일치해야 하고 마지막 다섯 자리는 IDNUM 매개변수와 일 치해야 합니다.	
원격 노드 ID 다운스트림 PU 이름 업스트림 DLUS 이름	AS/400 시스템에서 노드 ID는 EXCHID 매 개변수로 구성됩니다. 8자리의 16진수(선택적) 1-8자의 유형 A EBCDIC 문자 NETNAME.LUNAME(각각 1-8자의 유형 A EBCDIC 문자)	
접속 정보		

Motif 필드	유효한 항목/주의사항	사용자 구현값
폴 주소	<p>포트에서만 구성된 교환식 입력 포트의 경우</p> <p>2자리의 16진수</p> <ul style="list-style-type: none"> • 포인트간 C1 • 1차 교환식 전송의 0xFF(목적지 주소 알 수 없음) • 1차 분산 분기 회선의 고유 주소 <p>VTAM 호스트에서 폴 주소는 VTAM PU 정의에서 ADDR= 매개변수로 구성됩니다.</p> <p>AS/400 시스템에서 폴 주소는 회선 설명의 STNADR 매개변수입니다.</p>	

토큰링

토큰링 링크 프로토콜을 사용하는 연결성을 지원하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
토큰링 SAP 대화상자		
SNA 포트 이름	최대 8자	
토큰링 카드 번호	0 - number_of_cards_minus_1	
로컬 SAP 번호	16진수(4의 배수)	
초기에 활동 중	필요할 경우 선택	
연결 네트워크에서 정의	필요할 경우 선택	
CN 이름	NETNAME, CNNAME(각각 1-8자의 유형 A EBCDIC 문자)	
토큰링 링크 스테이션 대화상자		
링크 스테이션 필드		
이름	최대 8자	
SNA 포트 이름	최대 8자	
활성화	관리자에 의해	
	노드 시동 시	
LU 트래픽	요구가 있을 때 무관함	
	독립형만	
	종속형만	
독립 LU 트래픽		

Motif 필드	유효한 항목/주의사항	사용자 구현값
원격 노드	<i>NETNAME.CPNAME</i> (각각 1-8자의 유형 A EBCDIC 문자. 선택적)	
원격 노드 유형	원격 시스템이 VTAM 호스트일 경우, VTAM 시작 리스트의 <i>NETID</i> 매개변수에서 네트워크 이름(전체 이름의 처음 8자)을 찾을 수 있습니다. 마지막 8자는 VTAM 시작 리스트의 <i>SSCPNAME</i> 매개변수에 있습니다. 검색 끝 또는 LEN 노드 네트워크 노드	
종속 LU 트래픽		
원격 노드 역할	호스트 다운스트림 (SNA 게이트웨이)	
로컬 노드 ID	다운스트림 (DLUR) 8자리의 16진수(디폴트는 노드 이름) VTAM 구성에서 처음 세 자리 숫자는 PU 정의의 <i>IDBLK</i> 매개변수와 일치해야 하고 마지막 다섯 자리는 <i>IDNUM</i> 매개변수와 일치해야 합니다.	
원격 노드 ID	AS/400 시스템에서 노드 ID는 <i>EXCHID</i> 매개변수로 구성됩니다.	
다운스트림 PU 이름	8자리의 16진수(선택적)	
업스트림 DLUS 이름	1-8자의 유형 A EBCDIC 문자 <i>NETNAME.LUNAME</i> (각각 1-8자의 유형 A EBCDIC 문자)	
접속 정보		
MAC 주소	16진수 이 링크의 원격 끝이 VTAM 호스트이면 VTAM 포트 정의의 <i>MACADDR=</i> 매개변수에서 해당되는 MAC 주소를 찾을 수 있습니다. AS/400 시스템에 대한 링크를 구성할 경우, MAC 주소는 회선 설명에 있는 <i>ADPTADR</i> 매개변수입니다.	

Motif 필드	유효한 항목/주의사항	사용자 구현값
SAP 번호	16진수(4의 배수) 이 링크의 원격 끝이 VTAM 호스트이면 SAP 번호는 VTAM PU 정의의 SAPADDR= 매개변수입니다. AS/400 시스템에 대한 링크를 구성할 경우, SAP 번호는 회선 설명에 있는 SSAP 매개변수입니다.	

이더넷

이더넷 링크 프로토콜을 사용하는 연결성을 지원하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
이더넷 SAP 대화상자		
SNA 포트 이름	최대 8자	
이더넷 카드 번호	0 - <i>number_of_cards_minus_1</i>	
로컬 SAP 번호	16진수(4의 배수)	
초기에 활동 중	필요할 경우 선택	
연결 네트워크에서 정의	필요할 경우 선택	
CN 이름	NETNAME.CNNAME(각각 1-8자의 유형 A EBCDIC 문자)	
이더넷 유형	표준 또는 802.3 선택	
이더넷 링크 스테이션 대화상자		
링크 스테이션 필드		
이름	최대 8자	
SNA 포트 이름	최대 8자	
활성화	관리자에 의해	
	노드 시동 시	
	요구가 있을 때	
LU 트래픽	무관함	
	독립형만	
	종속형만	
독립 LU 트래픽		
원격 노드	NETNAME.CPNAME(각각 1-8자의 유형 A EBCDIC 문자. 선택적) 원격 시스템이 VTAM 호스트일 경우, VTAM 시작 리스트의 NETID 매개변수에서 네트워크 이름(전체 이름의 처음 8자)을 찾을 수 있습니다. 마지막 8자는 VTAM 시작 리스트의 SSCPNAME 매개변수에 있습니다.	

Motif 필드	유효한 항목/주의사항	사용자 구현값
원격 노드 유형	검색 네트워크 노드 끝 또는 LEN 노드	
중속 LU 트래픽		
원격 노드 역할	호스트 다운스트림 (SNA 게이트웨이) 다운스트림 (DLUR)	
로컬 노드 ID	8자리의 16진수(다폴트는 노드 이름) VTAM 구성에서 처음 세 자리 숫자는 PU 정의의 <i>IDBLK</i> 매개변수와 일치해야 하고 마지막 다섯 자리는 <i>IDNUM</i> 매개변수와 일치해야 합니다. AS/400 시스템에서 노드 ID는 <i>EXCHID</i> 매개변수로 구성됩니다.	
원격 노드 ID 다운스트림 PU 이름 업스트림 DLUS 이름	8자리의 16진수(선택적) 1-8자의 유형 A EBCDIC 문자 <i>NETNAME.LUNAME</i> (각각 1-8자의 유형 A EBCDIC 문자)	
접속 정보		
MAC 주소	16진수 이 링크의 원격 끝이 VTAM 호스트이면 VTAM 포트 정의의 <i>MACADDR=</i> 매개변수에서 해당되는 MAC 주소를 찾을 수 있습니다. AS/400 시스템에 대한 링크를 구성할 경우, MAC 주소는 회선 설명에 있는 <i>ADPTADR</i> 매개변수입니다.	
SAP 번호	16진수(4의 배수) 이 링크의 원격 끝이 VTAM 호스트이면 SAP 번호는 VTAM PU 정의의 <i>SAPADDR=</i> 매개변수입니다. AS/400 시스템에 대한 링크를 구성할 경우, SAP 번호는 회선 설명에 있는 <i>SSAP</i> 매개변수입니다.	

QLLC(X.25)

QLLC(X.25) 링크 프로토콜을 사용하는 연결성을 지원하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
QLLC 포트 대화상자		
SNA 포트 이름	최대 8자	
X.25 카드 번호	0 - <i>number_of_cards_minus_1</i>	
초기에 활동 중	필요할 경우 선택	
QLLC 링크 스테이션 대화상자		
링크 스테이션 필드		
이름	최대 8자	
SNA 포트 이름	최대 8자	
활성화	관리자에 의해	
	노드 시동 시	
LU 트래픽	요구가 있을 때 무관함	
	독립형만	
	종속형만	
독립 LU 트래픽		
원격 노드	<i>NETNAME.CPNAME</i> (각각 1-8자의 유형 A EBCDIC 문자. 선택적)	
	원격 시스템이 VTAM 호스트일 경우, VTAM 시작 리스트의 <i>NETID</i> 매개변수에서 네트워크 이름(전체 이름의 처음 8자)을 찾을 수 있습니다. 마지막 8자는 VTAM 시작 리스트의 <i>SSCPNAME</i> 매개변수에 있습니다.	
원격 노드 유형	검색	
	네트워크 노드	
	끝 또는 LEN 노드	
종속 LU 트래픽		
원격 노드 역할	호스트	
	다운스트림 (SNA 게이트웨이)	
로컬 노드 ID	다운스트림 (DLUR) 8자리의 16진수(다폴트는 노드 이름)	
	VTAM 구성에서 처음 세 자리 숫자는 PU 정의의 <i>IDBLK</i> 매개변수와 일치해야 하고 마지막 다섯 자리는 <i>IDNUM</i> 매개변수와 일치해야 합니다.	
	AS/400 시스템에서 노드 ID는 <i>EXCHID</i> 매개변수로 구성됩니다.	
원격 노드 ID	8자리의 16진수(선택적)	
다운스트림 PU 이름	1-8자의 유형 A EBCDIC 문자	
업스트림 DLUS 이름	<i>NETNAME.LUNAME</i> (각각 1-8자의 유형 A EBCDIC 문자)	

Motif 필드	유효한 항목/주의사항	사용자 구현값
접속 정보		
원격 X.25 주소	16진수(SVC 전용), 1-4096(PVC 전용)	

다중 경로 채널

다중 경로 채널 링크 프로토콜을 사용하는 연결성을 지원하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
다중 경로 채널 포트 대화상자		
SNA 포트 이름	최대 8자	
포트 번호	다중 경로 채널 장치의 장치 번호와 일치해야 함	
초기에 활동 중	필요할 경우 선택	
다중 경로 채널 링크 스테이션 대화상자		
링크 스테이션 필드		
이름	최대 8자	
SNA 포트 이름	최대 8자	
활성화	관리자에 의해 노드 시동 시 요구가 있을 때	
독립 LU 트래픽		
원격 노드	NETNAME.CPNAME(각각 1-8자의 유형 A EBCDIC 문자. 선택적) 원격 시스템이 VTAM 호스트일 경우, VTAM 시작 리스트의 NETID 매개변수에서 네트워크 이름(전체 이름의 처음 8자)을 찾을 수 있습니다. 마지막 8자는 VTAM 시작 리스트의 SSCPNAME 매개변수에 있습니다.	
원격 노드 유형	검색 네트워크 노드 끝 또는 LEN 노드	

Enterprise Extender(HPR/IP)

Enterprise Extender 링크 프로토콜을 사용하는 연결성을 지원하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
Enterprise Extender 포트 대화상자		
SNA 포트 이름	최대 8자	

Motif 필드	유효한 항목/주의사항	사용자 구현값
초기에 활동 중	필요할 경우 선택	
프로토콜	이 포트의 링크 스테이션이 IPv4 또는 IPv6 주소를 사용하는지 여부를 표시합니다.	
로컬 IP 인터페이스	다중 IP 네트워크에 액세스하는 경우 IP 링크에 사용할 로컬 네트워크 어댑터 카드의 식별자. 하나의 IP 네트워크에만 액세스하는 경우에는 이 필드를 비워둘 수 있습니다.	
연결 네트워크에서 정의	필요할 경우 선택	
CN 이름	NETNAME.CNNAME(각각 1-8자의 유형 A EBCDIC 문자)	
Enterprise Extender 링크 스테이션 대화상자		
링크 스테이션 필드		
이름	최대 8자	
SNA 포트 이름	최대 8자	
활성화	관리자에 의해	
	노드 시동 시	
	요구가 있을 때	
독립 LU 트래픽		
원격 노드	NETNAME.CPNAME(각각 1-8자의 유형 A EBCDIC 문자. 선택적)	
	원격 시스템이 VTAM 호스트일 경우, VTAM 시작 리스트의 NETID 매개변수에서 네트워크 이름(전체 이름의 처음 8자)을 찾을 수 있습니다. 마지막 8자는 VTAM 시작 리스트의 SSCPNAME 매개변수에 있습니다.	
원격 노드 유형	검색	
	끝 또는 LEN 노드	
	네트워크 노드	
접속 정보		
원격 IP 호스트 이름	IPv4 점으로 구분된 10진수 주소(예: 193.1.11.100), IPv6 콜론으로 구분된 16진수 주소(예: 2001:0db8:0000:0000:0000:0000:1428:57ab 또는 2001:db8::1428:57ab), 이름(예: newbox.this.co.uk), 별명(예: newbox). 포트의 protocol 매개변수는 이 주소가 IPv4 또는 IPv6 주소인지 여부를 판별합니다.	
	이름이나 별명을 지정할 경우, Linux 시스템은 이를 전체 이름으로 해석할 수 있어야 합니다(로컬 TCP/IP 구성을 사용하거나 도메인 이름 서버를 사용하여).	

패스-스루 서비스 워크시트

패스-스루 서비스가 로컬 노드에서 지원될 경우, 다음 절에 설명된 패스-스루 서비스에 대한 워크시트를 완료하십시오.

- 『로컬 노드의 DLUR』
- 『다운스트림 노드에 대한 패스-스루 DLUR』
- 188 페이지의 『SNA 게이트웨이』
- 188 페이지의 『TN 서버』
- 189 페이지의 『TN 방향 재지정자』

로컬 노드의 DLUR

로컬 노드에서 DLUR을 지원하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 구성:	175 페이지의 『노드 워크시트』를 참조하십시오.	
연결성 구성:	178 페이지의 『연결성 워크시트』를 참조하십시오. 로컬 노드에서 DLUR을 지원하려면 APPN 네트워크와의 연결성을 구성하십시오.	
DLUR PU:	.	
PU 이름	1-8자의 유형 A EBCDIC 문자	
DLUS 이름	NETNAME.LUNAME(각각 1-8자의 유형 A EBCDIC 문자)	
백업 DLUS 이름	이 매개변수는 선택적입니다.	
PU ID	NETNAME.LUNAME(각각 1-8자의 유형 A EBCDIC 문자) 8자리의 16진수 VTAM 구성에서 처음 세 자리 숫자는 PU 정의의 IDBLK 매개변수와 일치해야 하고 마지막 다섯 자리는 IDNUM 설정과 일치해야 합니다. AS/400 시스템에서 PU ID는 EXCHID 매개변수로 구성됩니다.	
초기에 활동 중	필요할 경우 선택	
압축 지원	필요할 경우 선택	
DLUS 접속 무한정 재시도	필요할 경우 선택	
로컬 LU 및 응용프로그램 구성:	190 페이지의 『사용자 응용프로그램 지원 워크시트』를 참조하십시오. 로컬 종속 LU와 필요한 응용프로그램 지원을 구성해야 합니다.	

다운스트림 노드에 대한 패스-스루 DLUR

로컬 노드가 APPN 네트워크 노드일 경우, 다운스트림 노드에 대해 패스-스루 DLUR 서비스를 제공할 수 있습니다. DLUR을 지원하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 구성:	노드를 네트워크 노드로 구성하십시오(175 페이지의 『APPN 네트워크 노드』 참조).	

Motif 필드	유효한 항목/주의사항	사용자 구현값
연결성 구성: 178 페이지의 『연결성 워크시트』를 참조하십시오. APPN 네트워크와의 연결성뿐만 아니라 종속 트래픽과 다운스트림 노드 사이의 연결성도 구성하십시오.		

SNA 게이트웨이

로컬 노드가 SNA 게이트웨이를 지원하도록 하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 구성: 175 페이지의 『노드 워크시트』를 참조하십시오.		
연결성 구성: 178 페이지의 『연결성 워크시트』를 참조하십시오. 종속 트래픽과 호스트 사이의 연결성, 그리고 종속 트래픽과 각 다운스트림 노드 사이의 연결성을 구성하십시오.		
로컬 LU 및 응용프로그램 구성: 190 페이지의 『사용자 응용프로그램 지원 워크시트』를 참조하십시오.		
LU 풀 대화상자		
풀 이름	1-8자의 유형 AE EBCDIC 문자	
LU 리스트	풀에 지정할 LU(유형 0-3) 이름	
다운스트림 LU 대화상자		
다운스트림 LU 이름	1-8자의 유형 A EBCDIC 문자(LU의 범위에 해당되는 기본 이름의 경우 1-5자)	
다운스트림 PU 이름	유형 A EBCDIC 문자열	
LU 번호	1-255(범위의 경우, 첫 번째 및 마지막 숫자 제공)	
업스트림 LU 이름	유형 A EBCDIC 문자열(LU 이름의 경우)이나 유형 AE EBCDIC 문자열(LU 풀 이름의 경우)	

TN 서버

로컬 노드가 TN3270 클라이언트를 지원하도록 하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 구성: 175 페이지의 『노드 워크시트』를 참조하십시오.		
연결성 구성: 178 페이지의 『연결성 워크시트』를 참조하십시오.(종속 LU 트래픽과 호스트 사이의 연결성 구성).		
로컬 LU 및 응용프로그램 구성: 190 페이지의 『사용자 응용프로그램 지원 워크시트』를 참조하십시오.		
LU 풀 대화상자		
풀 이름	1-8자의 유형 AE EBCDIC 문자	
LU 리스트	풀에 지정할 LU(유형 0-3) 이름	
TN 서버 액세스 대화상자		

Motif 필드	유효한 항목/주의사항	사용자 구현값
TN3270 클라이언트 주소	다음 중 하나 지정: <ul style="list-style-type: none"> • 디폴트 레코드(임의의 TN3270 클라이언트) • TCP/IP 주소(클라이언트의 IP 주소, IPv4 또는 IPv6 주소) • TCP/IP 이름 또는 별명 	
TN3270E 지원	TN3270E(TN3270 및 TN3287 외에)를 지원할 경우에 선택	
TN3270 포트 및 LU		
TCP/IP 포트 번호	보통 23.	
지정된 표시장치 LU	LU 또는 풀 이름	
지정된 프린터 LU	LU 또는 풀 이름	
특정 LU로의 액세스 허용	필요할 경우 선택	
SSL 보안 세션	필요할 경우 선택	
클라이언트 인증 수행	필요할 경우 선택	
암호화 수준	다음 중 하나 지정: <ul style="list-style-type: none"> • 인증만 • 최소 인증 • 최소 40비트 • 최소 56비트 • 최소 128비트 • 최소 168비트 	
TN 서버 연결 대화상자		
표시장치 LU	LU 이름	
프린터 LU	LU 이름	

TN 방향 재지정자

로컬 노드가 TN 방향 재지정자를 사용하는 텔넷 클라이언트를 지원하도록 하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 구성: 175 페이지의 『노드 워크시트』를 참조하십시오.		
TN 방향 재지정자 액세스 대화상자		
텔넷 클라이언트 주소	다음 중 하나 지정: <ul style="list-style-type: none"> • 디폴트 레코드(임의의 텔넷 클라이언트) • TCP/IP 주소(클라이언트의 IP 주소, IPv4 또는 IPv6 주소) • TCP/IP 이름 또는 별명 	

Motif 필드	유효한 항목/주의사항	사용자 구현값
TCP/IP 포트 번호 SSL 보안 세션 클라이언트 인증 수행 암호화 수준	보통 23. 필요할 경우 선택 필요할 경우 선택 다음 중 하나 지정: • 인증만 • 최소 인증 • 최소 40비트 • 최소 56비트 • 최소 128비트 • 최소 168비트	
호스트 주소	다음 중 하나 지정: • TCP/IP 주소(호스트의 IP 주소, IPv4 또는 IPv6 주소) • TCP/IP 이름 또는 별명	
TCP/IP 포트 번호 SSL 보안 세션 암호화 수준	필요할 경우 선택 다음 중 하나 지정: • 인증만 • 최소 인증 • 최소 40비트 • 최소 56비트 • 최소 128비트 • 최소 168비트	

사용자 응용프로그램 지원 워크시트

로컬 노드가 해당되는 사용자 레벨 응용프로그램을 지원하도록 하려면 다음 워크시트를 완료하십시오.

- 191 페이지의 『APPC』
- 193 페이지의 『CPI-C』
- 194 페이지의 『5250』
- 194 페이지의 『3270』
- 195 페이지의 『LUA』

APPC

로컬 노드가 APPC 응용프로그램을 지원하도록 하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 구성: 175 페이지의 『노드 워크시트』를 참조하십시오.		
연결성 구성: 178 페이지의 『연결성 워크시트』를 참조하십시오.		
로컬 LU 대화상자: 디폴트 제어점(CP) LU를 사용할 수 있는 경우에는 필요하지 않습니다.		
LU 이름 LU 별명	1-8자의 유형 A EBCDIC 문자 최대 8자	
중속 LU 매개변수		
호스트 <i>LS/DLUR PU</i>	호스트 또는 DLUR PU와의 중속 링크 스테이션 이름(LU 정의 이전에 정의해야 함)	
LU 번호	1-255	
디폴트 풀의 멤버	이 값은 VTAM/NCP LU 자원 정의문의 <i>LOCADDR</i> 매개변수와 일치해야 합니다. 필요할 경우 선택(중속 LU의 경우에만)	
로컬 LU 매개변수		
동기점 지원	필요할 경우 선택	
암호 대체 사용 불가능	필요할 경우 선택	
특정 SSCP로 제한	필요할 경우 선택(중속 LU의 경우에만)	
	SSCP ID는 VTAM 시작 리스트의 <i>SSCPID=</i> 필드에 정의됩니다.	
원격 노드 대화상자: 로컬 노드가 LEN 노드일 경우에만 구성하십시오.		
노드의 SNA 네트워크 이름	<i>NETNAME.CPNAME</i> (각각 1-8자의 유형 A EBCDIC 문자)	
상대방 LU 대화상자: LEN 노드와의 통신에, 상대방 LU 별명을 정의하기 위해, 또는 로컬 노드가 LEN 노드일 경우에만 필요합니다.		
상대방 LU 이름 별명 미해석 이름	<i>NETNAME.LUNAME</i> (각각 1-8자의 유형 A EBCDIC 문자) 최대 8자 1-8자의 유형 AE EBCDIC 문자(호스트 LU 이름이 로컬로 사용하는 PLU 이름과 다를 경우)	
병렬 세션 지원 위치	지원될 경우 선택 <i>NETNAME.CPNAME</i> (각각 1-8자의 유형 A EBCDIC 문자)	
LS 경로지정 대화상자: 링크 스테이션이 상대방 LU를 찾을 경우에만 필요합니다.		
LU 이름 LS 이름 상대방 LU 이름	1-8자의 유형 A EBCDIC 문자 최대 8자 <i>NETNAME.LUNAME</i> (각각 1-8자의 유형 A EBCDIC 문자)	
상대방 LU 이름을 와일드카드 사용	필요할 경우 선택	
모드 대화상자: 비표준 모드를 사용할 경우에만 필요합니다.		

Motif 필드	유효한 항목/주의사항	사용자 구현값
이름	1-8자의 유형 A EBCDIC 문자	
COS 이름	1-8자의 유형 A EBCDIC 문자	
세션 한계		
초기 세션 한계	최대 세션 한계. 권장값은 8입니다.	
최대 세션 한계	최대 32767	
최소 회선경합 성공 세션 수	최대 세션 한계. 권장값은 0입니다.	
최소 회선경합 실패 세션 수	권장값은 0입니다.	
자동 활성 세션 수	0 - <i>minimum_contention_winners</i>	
수신 페이싱 창		
초기 창 크기	권장값은 4입니다.	
최대 창 크기	선택적	
세션 시간종료		
최대 RU 크기	권장 상한은 1024입니다.	
압축 지원		
최대 인바운드 압축	없음	
	RLE	
	LZ9	
	LZ10	
최대 아웃바운드 압축	없음	
	RLE	
	LZ9	
	LZ10	
세션 보안 대화상자: 특정 로컬 및 상대방 LU 사이의 세션에 세션 보안이 요구될 경우에만 필요합니다.		
로컬 LU	1-8자의 유형 A EBCDIC 문자	
상대방 LU	1-8자의 유형 A EBCDIC 문자	
암호	16자리의 16진수	
TP 로드 대화상자: 로컬 TP가 원격 시스템의 요청에 대한 응답으로 시작될 경우에만 필요합니다.		
TP 이름	사용자 응용프로그램: 최대 64자의 ASCII 문자	
특정 LU로 제한	서비스 TP: 최대 8자리의 16진수 필요할 경우 선택	
LU 별명	최대 8자	
지원되는 복수 인스턴스	대기행렬에 없는 TP의 경우 선택. 선택하지 않으면 TP가 이미 실행 중일 때 입력되는 Allocate 요청이 대기행렬에 놓임	
입력 Allocate를 실행 중인 TP로 전송	대기행렬에 놓인 브로드캐스트 TP의 경우 선택	
TP 실행 프로그램의 전체 경로	실행 파일의 경로 및 파일 이름(디폴트는 TP 이름)	
인수	실행 파일에 유효한 인수	

Motif 필드	유효한 항목/주의사항	사용자 구현값
사용자 ID	최대 64자	
그룹 ID	최대 64자	
TP 정의 대화상자: APPC 특성을 정의합니다.		
TP 이름	사용자 응용프로그램: 최대 64자의 ASCII 문자	
대화 레벨 보안 필요	서비스 TP: 최대 8자리의 16진수 할당 요청에 대해 유효한 사용자 이름 및 암호가 필요할 경우 선택	
액세스 제한	사용자 이름을 보안 액세스 리스트에 포함해야 할 경우에 선택	
보안 액세스 리스트 대화 유형	보안 액세스 리스트 이름 기본 맵핑 아무거나 없음	
동기 레벨	확인 동기점 없음 또는 확인 없음, 확인 또는 동기점 필요할 경우 선택	
PIP 허용		
대화 보안 대화상자: 원격 시스템의 요청에 대한 응답으로 시작되는 로컬 TP에 대화 보안이 요구될 경우에만 필요합니다.		
사용자 ID	최대 10자	
암호	최대 10자	

CPI-C

로컬 노드가 CPI-C 응용프로그램을 지원하도록 하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 구성: 175 페이지의 『노드 워크시트』를 참조하십시오.		
연결성 구성: 178 페이지의 『연결성 워크시트』를 참조하십시오.		
APPC 구성: 191 페이지의 『APPC』를 참조하십시오.		
CPI-C 목적지 대화상자		
기호식 목적지 이름	1-8자	
로컬 LU	별명(8자까지) 또는 전체 이름 (NETNAME.LUNAME. 각각 1-8자의 유형 A EBCDIC 문자)	
상대방 LU	별명(8자까지) 또는 전체 이름 (NETNAME.LUNAME. 각각 1-8자의 유형 A EBCDIC 문자)	

Motif 필드	유효한 항목/주의사항	사용자 구현값
모드 상대방 TP 이름	유형 A EBCDIC 문자열 사용자 응용프로그램: 최대 64자	
보안	서비스 TP: 최대 8자리의 16진수 없음 동일	
사용자 ID	프로그램 보안 레벨이 동일 또는 프로그램일 경우에 만(사용자 로그인 ID에 관련되지 않음)	
암호	보안 레벨이 프로그램일 경우에만(사용자 로그인 암호에 관련되지 않음)	

5250

로컬 노드가 5250 통신을 지원하도록 하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 구성:	175 페이지의 『노드 워크시트』를 참조하십시오.	
연결성 구성:	178 페이지의 『연결성 워크시트』를 참조하십시오(독립 트래픽에 구성).	
APPC 구성:	191 페이지의 『APPC』를 참조하십시오.	

3270

로컬 노드가 3270 통신을 지원하도록 하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 구성:	175 페이지의 『노드 워크시트』를 참조하십시오.	
연결성 구성:	178 페이지의 『연결성 워크시트』를 참조하십시오(중속 트래픽에 구성).	
LU 유형	0-3 대화상자	
LU 이름	1-8자의 유형 A EBCDIC 문자(또는 LU의 범위에 해당되는 기본 이름의 경우 1-5자)	
호스트 <i>LS/DLUR/PU</i>	호스트 또는 DLUR/PU와의 중속 링크 스테 이션 이름(LU 정의 이전에 정의해야 함)	
LU 번호	1-255(범위의 경우, 첫 번째 및 마지막 숫자 제공) 이 값은 VTAM/NCP LU 자원 정의문의 <i>LOCADDR</i> 매개변수와 일치해야 합니다.	

Motif 필드	유효한 항목/주의사항	사용자 구현값
LU 유형	3270 모델 2(80x24) 표시장치 3270 모델 3(80x32) 표시장치 3270 모델 4(80x43) 표시장치 3270 모델 5(132x27) 표시장치 3270 프린터 SCS 프린터	
풀 내의 LU	원하는 옵션 선택(표시장치 및 무제한 LU의 경우에만)	
풀 이름	1-8자의 유형 AE EBCDIC 문자	
LU 풀 대화상자		
풀 이름	1-8자의 유형 AE EBCDIC 문자	
LU 리스트	풀에 지정할 LU(유형 0-3) 이름	

LUA

로컬 노드가 LUA 응용프로그램을 지원하도록 하려면 이 워크시트를 완료하십시오.

Motif 필드	유효한 항목/주의사항	사용자 구현값
노드 구성:	175 페이지의 『노드 워크시트』를 참조하십시오.	
연결성 구성:	178 페이지의 『연결성 워크시트』를 참조하십시오(중속 트랙에 구성).	
LU 유형 0-3 대화상자		
LU 이름	1-8자의 유형 A EBCDIC 문자(또는 LU의 범위에 해당되는 기본 이름의 경우 1-5자)	
호스트 LS/DLUR PU	호스트 또는 DLUR PU와의 중속 링크 스테이션 이름(LU 정의 이전에 정의해야 함)	
LU 번호	1-255(범위의 경우, 첫 번째 및 마지막 숫자 제공)	
LU 유형	무제한	
풀 내의 LU	원하는 옵션 선택(표시장치 및 무제한 LU의 경우에만)	
풀 이름	1-8자의 유형 AE EBCDIC 문자	
LU 풀 대화상자		
풀 이름	1-8자의 유형 AE EBCDIC 문자	
LU 리스트	풀에 지정할 LU(유형 0-3) 이름	

부록 B. 명령행에서 기동 가능 TP 구성

Linux용 Communications Server에는 사용자나 TP 설치 프로그램 작성자가 기동 가능 TP를 정의할 수 있는 명령행 유틸리티가 있습니다. 이 명령행 유틸리티는 서버나 클라이언트에서 실행할 수 있습니다.

명령 구문은 TP 정의를 정의, 제거 또는 조회 중인지 여부에 따라 다르고 Windows에서 원격 API 클라이언트에 대해 다릅니다.

WINDOWS

tpinst32 명령을 사용하기 전에 클라이언트 소프트웨어가 설치된 디렉토리로 변경하십시오.

tpinst32 명령은 일반적으로 클라이언트 소프트웨어를 설치할 때 선택한 언어로 정보를 표시합니다. 이 정보를 다른 언어로 표시하려면 선호하는 언어에 해당하는 서브디렉토리로 변경한 후 명령을 실행하십시오. 예를 들어, 프랑스로 정보를 표시하려면 클라이언트 소프트웨어가 설치된 디렉토리 아래의 **fr_FR** 서브디렉토리로 변경하십시오.

기동 가능 TP 정의:

AIX, LINUX

snatpinstall -a file_name

WINDOWS

tpinst32 -a file_name

주: **tpinst32** 명령은 32비트 및 x64 버전의 Windows 모두에 적용됩니다.

명령행에서 기동 가능 TP 구성

이 명령은 지정된 *file_name*에서 하나 이상의 TP 정의를 추가합니다. 파일에 이름이 지정된 TP가 이미 정의된 경우, 파일에 있는 정보는 기존 정의를 대체됩니다. 필수 파일 형식에 대해서는 199 페이지의 『기동 가능 TP 정의의 파일 형식』을 참조하십시오.

기동 가능 TP 정의 제거:

AIX, LINUX

```
snatpinstall -r -t TP_name [ -l LU_alias ]
```

이 명령은 지정된 TP 이름을 가지고 있는 항목과 (같은 TP 이름으로 여러 개의 APPC TP가 정의된 경우) 지정된 LU 별명을 제거합니다. 항목이 CPI-C 응용프로그램에 해당되거나 지정한 TP 이름으로 정의된 APPC TP가 단 하나인 경우 **-l LU_alias** 옵션을 생략하십시오.

WINDOWS

```
tpinst32 -r -t TP_name
```

이 명령은 지정된 TP 이름을 갖는 항목을 제거합니다.

기동 가능 TP 정의 조회:

AIX, LINUX

```
snatpinstall -q [ -t TP_name ] [ -l LU_alias ]
```

이 명령은 지정된 TP 이름을 가지고 있는 항목과 (같은 TP 이름으로 여러 개의 APPC TP가 정의된 경우) 지정된 LU 별명을 조회합니다. 항목이 CPI-C 응용프로그램에 해당되거나 지정한 TP 이름으로 정의된 APPC TP가 단 하나인 경우 **-l LU_alias** 옵션을 생략하십시오. **-t TP_name** 옵션을 포함하지 않으면 명령은 기동 가능한 모든 TP 정의를 조회합니다.

WINDOWS

```
tpinst32 -q -t TP_name
```

이 명령은 지정된 TP 이름을 갖는 항목을 조회합니다. **-t TP_name** 옵션을 포함하지 않으면 명령은 기동 가능한 모든 TP 정의를 조회합니다.

기동 가능 TP 정의의 파일 형식

기동 가능 TP에 대한 구성 정보를 제공하는 파일은 표준 텍스트 편집기를 사용하여 수정할 수 있는 ASCII 텍스트 파일입니다. 파일의 각 항목 형식은 다음과 같습니다.

```
[TPname]
PATH           = full_pathname_of_executable_file
ARGUMENTS     = command-line_arguments_separated_by_spaces
TYPE          = QUEUED | QUEUED-BROADCAST | NON-QUEUED
TIMEOUT       = nnn
```

AIX, LINUX

```
USERID        = user_ID
GROUP         = group_ID
LUALIAS       = LU_alias
ENV           = environment_variable=value
              .
              .
              .
ENV           = environment_variable=value
```

WINDOWS

```
SHOW          = MAXIMIZED | MINIMIZED | HIDDEN | NORMAL | NOACTIVATE | MINNOACTIVATE
SECURITY_TYPE = APPLICATION | SERVICE
SERVICE_NAME = name_of_installed_service
USERID        = domain_name\user_ID
```

매개변수는 다음과 같습니다. 운영자가 시작한 TP의 경우, 사용되는 유일한 매개변수는 TP 이름, TP 유형, 시간종료 값 및 (AIX 또는 Linux의 APPC TP일 경우) LU 별명입니다. 다른 매개변수는 자동으로 시작된 TP에만 적용됩니다.

AIX, LINUX

AIX 또는 Linux에서 잘못된 매개변수를 입력하면 Linux용 Communications Server는 오류 메시지를 리턴합니다.

WINDOWS

기동 가능 TP 정의의 파일 형식

Windows 시스템에서 Linux용 Communications Server는 잘못된 매개변수를 무시합니다.



TPname

TP의 이름(사이에 공백 없이 1-64자). Receive_Allocate 또는 입력되는 Allocate 요청에서 지정한 TP 이름을 이 이름과 일치시킵니다. TP가 자동으로 시작되는 TP일 경우, TP가 시작될 때 Receive_Allocate에 이 TP 이름을 지정하여 Linux용 Communications Server가 입력되는 Attach를 해당되는 TP로 전송하도록 해야 합니다.

이 이름은 대괄호로 묶어야 하며 큰따옴표 안에 ASCII 문자열로 지정할 수 있습니다(예: ["TPNAME1"]). 또는, TP 이름의 EBCDIC 문자를 나타내는 16진 배열 (예: [<53504E414D45F1>])이나 두 형식의 조합(예: [<3f>"TP1"])으로 지정할 수 있습니다. 이 예에서, 첫 번째 문자는 인쇄할 수 없는 문자인 0x3f이며 다음 문자는 "TP1"입니다.

Linux용 Communications Server는 제공된 ASCII 문자열을 EBCDIC으로 변환하지만, 16진 문자열(이미 EBCDIC 형식인 것으로 간주되는)에 대해서는 변환을 수행하지 않습니다. 그런 다음 지정한 TP 이름과 일치시키기 전에 이름 오른쪽을 EBCDIC 영역으로 채웁니다(총 64자까지).

PATH TP용 실행 파일의 경로 및 파일 이름입니다.

이 행은 선택적입니다. 이 행이 포함되지 않으면 Linux용 Communications Server는 실행 파일 이름이 TP 이름과 같은 것으로 간주합니다. 경로 없이 파일 이름을 지정하면 AIX 또는 Linux 시스템의 디폴트 경로는 **/etc/opt/ibm/sna**입니다. Windows 클라이언트의 경우, Linux용 Communications Server는 실행 파일을 찾기 위해 정상적인 Windows 메커니즘을 사용합니다.

ARGUMENTS

TP에 전달할 명령행 인수(공백으로 구분). 이 인수는 명령행에 표시된 것과 같은 순서로 TP에 전달됩니다.

이 행은 선택적입니다. 이 값을 포함하지 않을 경우, TP는 명령행 인수 없이 기동됩니다.

TYPE 다음 값 중 하나를 지정하십시오.

QUEUED

TP는 대기행렬에 놓인 TP입니다. TP를 실행하는 동안 입력되는 Allocate 요청은 TP가 다른 Receive_Allocate를 실행할 때까지 또는

실행을 완료하고 다시 시작될 수 있을 때까지 대기행렬에 있게 됩니다. 입력되는 Allocate 요청은 이 요청이 컴퓨터에 전송되도록 구성된 LU가 수신할 경우에만 TP에 전송됩니다.

QUEUED-BROADCAST

TP는 대기행렬에 놓인 브로드캐스트 TP입니다. TP를 실행하는 동안 입력되는 Allocate 요청은 TP가 다른 Receive_Allocate를 실행할 때까지 또는 실행을 완료하고 다시 시작될 수 있을 때까지 대기행렬에 있게 됩니다. TP가 시작되면 TP에 관한 정보가 LAN의 모든 서버로 브로드캐스트됩니다. 다른 컴퓨터의 LU가 입력되는 Allocate 요청을 수신하고 어떤 경로지정 정보도 구성되어 있지 않은 경우, 그 LU는 동적으로 TP를 찾아서 Allocate 요청을 전송할 수 있습니다.

QUEUED 대신 QUEUED-BROADCAST를 사용하면 LU에 대해 명시적 경로지정 정보를 구성하지 않아도 되므로, 여러 컴퓨터에서 동일 TP의 여러 사본을 실행하여 로드 균형을 이룰 수 있습니다. 그러나 LAN 트래픽을 줄이기 위해 정보를 브로드캐스트하지 않거나 특정 LU에 도달하는 입력 Allocate 요청이 항상 TP의 동일 사본에 전송되도록 하려면 QUEUED를 사용해야 합니다.

NON-QUEUED

TP는 대기행렬에 없는 TP입니다. Linux용 Communications Server는 입력되는 Allocate 요청이 도달할 때마다 새 TP 사본을 시작합니다. 대기행렬에 없는 TP의 경우 *TIMEOUT* 매개변수를 지정하지 마십시오.

대기행렬에 없게 정의된 TP를 운영자가 시작할 수 없습니다. 그러한 TP는 항상 Linux용 Communications Server에 의해 자동으로 시작됩니다. 운영자가 TP를 시작할 경우에는 NON-QUEUED를 지정하지 마십시오. 사용자가 대기행렬에 없는 TP를 시작하려고 할 경우, Linux용 Communications Server는 입력되는 어떤 Allocate 요청도 기다리지 않으므로 Receive_Allocate를 거부합니다.

대기행렬에 없는 TP가 대화를 종료하고 나면, 그 TP는 종료되거나 다른 Receive_Allocate를 발행할 수 있습니다. 자주 사용하는 프로그램의 경우, 이렇게 하면 대화마다 프로그램의 새 인스턴스를 시작하므로서 발생하는 성능 오버헤드를 피할 수 있습니다. 대기행렬에 없고 자동으로 시작되는 TP에 대해 Attach가 수신될 때마다, Linux용 Communications Server는 해당 TP 인스턴스로부터의 미해결 Receive_Allocate가 있는지 확인합니다. 있으면 TP는 입력되는 대화에 사용되고, 그렇지 않으면 Linux용 Communications Server가 프로그램의 새 인스턴스를 시작합니다.

NON-QUEUED를 사용할 경우, 한 번에 여러 개의 TP 사본을 실행할 수 있습니다. TP가 파일에 쓸 경우, 서로 다른 TP 사본이 서로의 파일 위에 겹쳐쓰지 않도록 해야 합니다. 이렇게 하려면 다음 방법 중 하나를 사용하십시오.

- TP가 파일을 작성하는 대신 기존 파일에 데이터를 추가하도록 하십시오 (모든 TP 사본이 같은 파일에 데이터를 추가하도록).
- TP가 실행 중 사용하는 프로세스 ID를 기반으로 런타임 시 파일 이름을 생성하도록 TP를 설계하십시오(TP의 각 사본이 다른 파일에 기록하도록).

이 행은 선택적입니다. 이 행을 포함하지 않거나 유효하지 않은 값을 지정할 경우 디폴트는 QUEUED입니다.

TIMEOUT

대기 상태의 입력되는 Allocate 요청이 없을 경우 TP가 발행한 Receive_Allocate 호출이 블록화해야 하는 최대 시간(초). 이 시간 내에 입력되는 어떤 Allocate도 수신되지 않으면, 호출은 『상태 확인 - Allocate 대기 상태 아님』을 표시하는 리턴 코드로 실패합니다.

0 시간종료 값은 호출을 발행할 때 입력되는 Allocate가 아직 대기 상태가 아니면 호출이 항상 실패함을 표시합니다. 시간종료 값 -1은 호출이 입력되는 Allocate를 무한정 기다리며 시간종료되지 않음을 표시합니다.

이 행은 선택적입니다. 이 행이 포함되지 않거나 유효하지 않은 값(숫자가 아닌 값)이 지정되면 디폴트는 -1(무한정)입니다.

TYPE 매개변수가 NON-QUEUED로 설정되면 이 매개변수를 지정하지 마십시오. Linux용 Communications Server는 대기행렬에 없는 TP에 대해 시간종료 값 0을 사용합니다. TP는 항상 입력되는 Allocate에 대한 응답으로 시작되므로 항상 하나의 대기가 있기 때문입니다.

AIX, LINUX

USERID

Linux용 Communications Server가 TP를 시작하기 위해 사용하는 사용자 ID를 지정하십시오. TP는 이 사용자 ID와 연관되는 홈 디렉토리에서 시작됩니다. 이 홈 디렉토리는 또한 추적 파일과 TP가 액세스하는 다른 파일에 대한 디폴트 경로이기도 합니다(응용프로그램이 전체 경로를 지정하여 대체하지 않는 한). 응용프로그램이 경로 없이 파일 이름을 지정할 경우, Linux용 Communications Server는 이 홈 디렉토리에서 파일을 검색합니다. 응용프로그램이 상대 경로를 사용하여 파일 이름을 지정하면, Linux용 Communications Server는 이 홈 디렉토리에 상대적으로 지정된 디렉토리에서 파일을 검색합니다.

이 행은 필수이므로 반드시 지정해야 합니다. ID는 Linux용 Communications Server 컴퓨터에서 유효한 로그인 ID여야 합니다. 이 ID는 AIX 또는 Linux 구성에서 사용자 이름 길이를 더 작은 숫자로 제한하지 않는 한 64자까지 가능합니다.

PATH 매개변수에 지정된 TP용 실행 파일은 지정된 사용자에게 대해 실행 권한을 가지고 있어야 합니다. 또한 *USERID*가 루트로 설정된 경우, 파일을 루트가 소유해야 하고 *setuid* 및 *setgid* 권한을 가지고 있어야 Linux용 Communications Server에 의해 자동으로 시작될 수 있습니다.

GROUP

Linux용 Communications Server가 TP를 시작하기 위해 사용하는 그룹 ID를 지정하십시오. ID는 Linux용 Communications Server 컴퓨터에서 유효한 그룹 ID여야 합니다. 이 ID는 AIX 또는 Linux 구성에서 사용자 이름 길이를 더 작은 숫자로 제한하지 않는 한 64자까지 가능합니다.

이 행은 선택적입니다. 이 행을 포함하지 않을 경우, 디폴트는 기타입니다.

LUALIAS

TP가 입력되는 Attach를 승인할 로컬 LU 별명을 지정하십시오.

주: 이 매개변수는 TP가 APPC TP인 경우에만 사용할 수 있습니다. TP가 CPI-C 응용프로그램일 경우, 이 매개변수를 지정하지 마십시오. CPI-C는 특정 로컬 LU로부터 입력되는 Attach의 승인을 지원하지 않으므로, CPI-C 응용프로그램에 LU 별명을 지정하려고 하면(공백 LU 별명이어도) 입력되는 Attach를 TP에 전송할 때 오류가 발생합니다.

이는 Linux용 Communications Server 로컬 APPC LU의 이름과 일치해야 하는 8자 이름입니다.

TP가 로컬 LU에서 입력되는 Attach를 승인함을 표시하려면, 이 매개변수를 두 개의 큰 따옴표("")로 설정하여 공백 LU 별명임을 표시하십시오. 기동 가능 TP 데이터 파일에 같은 TP 이름에 해당되는 두 개 이상의 항목이 있을 경우, 이 항목 중 하나만 공백 LU 별명을 지정할 수 있습니다. 다른 항목들은 각각 서로 다른 명시적 LU 별명을 지정해야 합니다. Linux용 Communications Server는 이 TP 이름에 해당되는 입력 Attach를 적절한 LU 별명을 지정하는 TP와 일치시키거나, 일치하는 LU 별명이 없을 경우 공백 LU 별명을 지정하는 TP와 일치시킵니다.

공백이 아닌 LU 별명을 파일에 지정한 경우, TP는 APPC RECEIVE_ALLOCATE Verb의 확장 양식을 사용하고 이 LU 별명을 Verb에 대한 매개변수로 지정해야 합니다. 그러면 Linux용 Communications Server가 입력되는 Attach를 올바른 TP에 전송할 수 있습니다. 다른 형식의 RECEIVE_ALLOCATE에 대한 자세한 정보는 AIX 또는 Linux용 IBM Communications Server APPC 프로그래머 안내서를 참조하십시오. TP가 응

응용프로그램에 빌드하기 보다는 런타임에서 올바른 LU 별명을 판별할 수 있도록 허용하려면, 적절한 LU 별명을 포함하도록 환경 변수를 설정하고(ENV 매개변수 사용), 응용프로그램이 RECEIVE_ALLOCATE를 발행할 방법을 판별하기 위해 이 환경 변수를 읽도록 설계하면 됩니다.

이 행은 선택적입니다. 이 행을 포함하지 않을 경우, 디폴트는 로컬 LU에서 입력되는 Attach를 승인하는 것이며 TP는 두 가지의 APPC RECEIVE_ALLOCATE Verb 양식 중 하나를 사용할 수 있습니다.

ENV TP에 필요한 환경 변수를 지정하십시오. 각 변수는 개별 *ENV* 행에 *environment_variable=value* 양식으로 지정됩니다. 최대 64개의 *ENV* 행을 포함할 수 있습니다. 변수는 여기에 표시되는 것과 같은 순서로 설정됩니다.

environment_variable=value 문자열에서는 = 문자 앞뒤에 공백이나 탭 문자가 있어서는 안됩니다.

WINDOWS

SHOW

이 매개변수는 응용프로그램이 GUI 응용프로그램인 경우에만 적용되며, 응용프로그램이 콘솔 응용프로그램인 경우에는 무시됩니다. 응용프로그램을 시작할 때 이 응용프로그램의 표시 방법을 지정하십시오. 이 매개변수는 응용프로그램에 전달되며, Linux용 Communications Server가 처리하지 않습니다. 응용프로그램이 이 매개변수를 해석하고 그에 따라 작동합니다. 다음 값을 입력할 수 있습니다.

MAXIMIZED

응용프로그램이 최대화됩니다.

MINIMIZED

응용프로그램이 최소화됩니다.

HIDDEN

응용프로그램이 화면에 표시되지 않습니다.

NORMAL

응용프로그램이 정상 크기로 제 위치에 표시됩니다.

NOACTIVATE

응용프로그램이 정상 크기로 제 위치에 표시되지만, 초점은 이전 활성 창에 남습니다. 이 응용프로그램의 창은 활성 창이 되지 않습니다.

MINNOACTIVATE

응용프로그램이 최소화되지만 초점은 이전 활성 창에 남습니다.

이 매개변수는 선택적입니다. 이 매개변수를 포함하지 않을 경우, 디폴트는 NORMAL입니다.

SECURITY_TYPE

TP 실행 파일의 보안 유형을 지정하십시오.

APPLICATION

TP 실행 파일은 CreateProcess 시스템 호출을 사용하여 응용프로그램으로 시작됩니다.

SERVICE

TP 실행 파일은 StartService 시스템 호출을 사용하여 서비스로 시작됩니다. 이 경우, 서비스는 이전에 *SERVICE_NAME* 매개변수에 지정된 이름을 사용하여 서비스 제어 관리자로서 설치되어 있어야 합니다.

이 값은 Windows 서비스로 실행 중인 TP(16진수로 지정된 4자 이름의 SNA 서비스 TP가 아닌)를 가리킵니다. Windows에서는 한 번에 하나의 서비스 사본만 실행할 수 있으므로 *TYPE* 매개변수를 NON-QUEUED로 설정하면 안됩니다. 이 값을 지정할 경우, 대신 QUEUED-BROADCAST 값이 사용됩니다.

SERVICE_NAME

서비스 제어 관리자와 함께 설치된 서비스의 이름입니다. 이 매개변수는 *SECURITY_TYPE*이 SERVICE일 경우에만 사용됩니다.

USERID

*SECURITY_TYPE*이 APPLICATION일 때 클라이언트가 TP를 시작하기 위해 사용해야 하는 도메인 및 사용자 ID를 지정하십시오. 이 매개변수의 형식은 Windows 클라이언트 컴퓨터가 도메인의 일부이면 *domain_name\user_ID*이고, Windows 클라이언트 컴퓨터가 도메인의 일부가 아니면 *computer_name\user_ID* (도메인 이름 대신 Windows 클라이언트의 고유 컴퓨터 이름을 나타냄)입니다. 클라이언트는 지정된 사용자 로그인 세션에서 TP를 시작하려고 합니다. *USERID*가 공백이거나 지정하지 않을 경우, TP는 콘솔 세션에서 시작됩니다. 지정한 사용자가 로그인하지 않았거나 콘솔에 로그인한 사용자가 없을 경우, TP는 시작되지 않으며 Linux용 Communications Server는 실패 통지를 받습니다.



이 항목들의 형식에 대해 다음 사항을 유의하십시오.

- 행의 첫 번째 문자로 #를 입력하여 설명을 포함시킬 수 있습니다. 그러면 Linux용 Communications Server가 이 행을 무시합니다. Linux용 Communications Server는 아무것도 없는 빈 줄도 무시합니다.
- 각각의 *parameter=value* 항목은 한 행에 있어야 합니다. 줄 바꾸기 문자가 있으면 안됩니다. 행의 최대 길이는 255자입니다. 추가 문자는 무시됩니다.

기동 가능 TP 정의의 파일 형식

- 행의 맨 앞과 맨 끝 또는 = 앞뒤에 있는 여백(공백 문자와 탭 문자)은 무시됩니다 (*ENV* 매개변수에 대한 *environment_variable=value* 문자열 제외).
- 각각의 TP 정의는 TP 이름을 식별하는 행으로 시작하고 파일 끝나거나 다음 TP 이름으로 끝납니다.
- 64번까지 발생할 수 있는 *ENV* 행을 제외하고, 동일 TP에 동일 매개변수를 두 번 이상 지정하지 마십시오. 동일 매개변수를 여러 번 지정하면, 각 키워드의 마지막 인스턴스만 사용됩니다.

부록 C. DDDLU용 TN3270 LU 모델 구성

TN3270 클라이언트가 DDDLU를 사용하여 Linux용 Communications Server TN 서버를 통해 호스트에 연결하는 경우, Linux용 Communications Server는 클라이언트에 필요한 LU 모델에 대한 정보를 호스트로 전송해야 합니다. 일반적으로 클라이언트가 지정한 터미널 유형(장치 유형)에서 표준 맵핑을 사용하여 LU 모델을 판별합니다.

TN3270 장치 유형과 LU 모델 간의 맵핑을 변경해야 하는 경우에는 **tn3270dev.dat** 파일을 사용하여 변경할 수 있습니다. 이 파일의 샘플 버전은 /opt/ibm/sna/samples에서 제공됩니다. 이 파일을 /etc/opt/ibm/sna로 복사한 후 이 위치에서 **vi** 등의 표준 ASCII 텍스트 편집기를 사용하여 변경하십시오. 변경사항은 다음 번에 Linux용 Communications Server를 다시 시작할 때 적용됩니다.

이 파일의 각 행은 TN3270 장치와 LU 모델 간을 맵핑하는 문자열(이 장치에 대해 호스트로 보내야 하는)을 나타냅니다. 각 행은 공백으로 분리되는 다음과 같은 항목으로 구성됩니다.

- 첫 번째 항목은 이 맵핑이 TN3270E 클라이언트용인 경우(TN3270 확장 사용) 단일 문자 Y이며 표준 TN3270 클라이언트용인 경우 N입니다.
- 두 번째 항목은 클라이언트에서 지정된 터미널 유형(장치 유형)입니다. 이 항목은 최대 40자의 텍스트 문자열이며 유효한 문자는 문자 A-Z, 숫자 0-9, / 및 -입니다.
- 세 번째 항목은 이 클라이언트에 맞는 LU 모델을 식별하도록 Linux용 Communications Server가 호스트로 전송하는 7문자의 LU 모델 문자열입니다.

Linux용 Communications Server가 제공하는 샘플 파일에는 TN3270과 TN3270E 버전 둘 다에 대해 18개의 표준 장치 유형에 대한 맵핑(모두 36개 항목)이 있습니다.

- 이 표준 장치 중 하나 이상에 대해 다른 LU 모델 문자열을 지정해야 하는 경우에는 파일의 해당 행에서 세 번째 항목을 변경하여 필요한 7문자의 문자열을 지정하십시오.
- 샘플 파일에 포함되지 않은 추가 장치 유형을 지원해야 하는 경우에는 각 장치 유형에 대해 위에서 설명한 형식으로 파일에 새 행을 추가하십시오.

부록 D. 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.

135-700

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.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예: 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.의 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. 2000, 2005, 2006, 2007, 2008, 2009. All rights reserved.

상표

IBM, IBM 로고 및 ibm.com은 미국 또는 기타 국가에서 사용되는 International Business Machines Corp.의 상표 또는 등록상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 해당 회사의 상표입니다. 현재 IBM 상표 목록은 웹 “저작권 및 상표 정보”(www.ibm.com/legal/copytrade.shtml)에 있습니다.

Adobe는 미국 또는 기타 국가에서 사용되는 Adobe Systems Incorporated의 등록상표입니다.

Intel 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation 또는 자회사의 상표 또는 등록상표입니다.

Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.

Microsoft 및 Windows는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 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 6.4
- 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 제목과 주문 번호만 표시됩니다.

Linux용 Communications Server 버전 6.4 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 빠른 시작*을 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 및 GA30-2280)

이 책은 지원되는 네트워크 특성, 설치, 구성 및 운영에 관한 정보를 포함하고 있으며 Linux용 Communications Server에 대해 개괄적으로 설명합니다. 이 서적에는 두 가지 버전이 있습니다.

GA30-2285는 i686, x86_64 및 ppc64 플랫폼의 Linux용 Communications Server를 위한 것입니다.

GA30-2280은 System z의 Linux용 Communications Server를 위한 것입니다.

- *Linux용 IBM Communications Server 관리 안내서*(SA30-2286)

이 책은 SNA 및 Linux용 Communications Server 개요와 Linux용 Communications Server 구성 및 운영에 관한 정보를 제공합니다.

- *IBM Communications Server for Linux Administration Command Reference (SC31-6770)*

이 책은 SNA 및 Linux용 Communications Server 명령에 관한 정보를 제공합니다.

- *IBM Communications Server for AIX or Linux CPI-C Programmer's Guide(SC23-8691)*

이 책은 숙련된 “C” 또는 Java™ 프로그래머에게 Linux용 Communications Server CPI 통신 API를 사용하여 SNA 트랜잭션 프로그램을 작성할 수 있는 정보를 제공합니다.

- *IBM Communications Server for AIX or Linux APPC Programmer's Guide(SC23-8692)*

이 책에는 고급 프로그램간 통신(APPC)을 사용하여 응용프로그램을 작성하는 데 필요한 정보가 들어 있습니다.

- *IBM Communications Server for AIX or Linux LUA Programmer's Guide(SC23-8690)*

이 책에는 일반 LU 응용프로그램 프로그래밍 인터페이스(LUA)를 사용하여 응용프로그램을 작성하는 데 필요한 정보가 들어 있습니다.

- *IBM Communications Server for AIX or Linux CSV Programmer's Guide(SC23-8689)*

이 책에는 공통 서비스 Verb(CSV) API를 사용하여 응용프로그램을 작성하는 데 필요한 정보가 들어 있습니다.

- *IBM Communications Server for AIX or Linux MS Programmer's Guide(SC23-8596)*

이 책에는 관리 서비스(MS) API를 사용하여 응용프로그램을 작성하는 데 필요한 정보가 들어 있습니다.

- *IBM Communications Server for Linux NOF Programmer's Guide(SC31-6778)*

이 책에는 노드 연산자 기능(NOF) API를 사용하여 응용프로그램을 작성하는 데 필요한 정보가 들어 있습니다.

- *Linux용 IBM Communications Server 진단 안내서(GA30-2279)*

이 책에서는 SNA 네트워크 문제점 해결에 대한 정보를 제공합니다.

- *IBM Communications Server for AIX or Linux APPC Application Suite User's Guide(SC23-8595)*

이 책은 Linux용 Communications Server와 함께 사용되는 APPC 응용프로그램에 관한 정보를 제공합니다.

- *Linux용 IBM Communications Server 용어집*(GA30-2281)

이 책은 Linux용 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]

가상 경로지정 노드(VRN) 29
가상 터미널 액세스 방법(VTAM) 15
간결 로깅 41
감사 로그 파일 40
경계 노드 3
계획 워크시트 43
고객 정보 제어 시스템(CICS) 8
고급 피어간 통신(APPN) 1
고성능 경로지정(HPR) 14, 23
고속 전송 프로토콜(RTP) 14, 23
공유 액세스 전송 기능(SATF) 27
관리
 도구 34
 책임 33
관리 서비스(MS) 17
관리 서비스(MS) API 7
구성
 노드 62
 보안 액세스 리스트 116
 연결 네트워크 68
 연결성 67
 중속 LU 85
 타스크 61
 파일 38
 패스-스루 서비스 119
 포트 68
 APPC 보안 114
 APPC 통신 91
 CPI-C 부가 정보 112
 DLC 68
 DLUR 131
 SNA 게이트웨이 129
 TN 방향 재지정자 액세스 레코드 125
 TN 서버 액세스 레코드 120
 TN 서버 연결 레코드 124
 TP 99
구성 서버 61
 제거 62
 추가 62
그룹 ID 매개변수 105
기동 가능 TP 12
 데이터 파일 39

기동 가능 TP (계속)
 Linux용 Communications Server 정의
 100
 snatpinstall 사용 197
기동 TP 12, 100
기본 대화 13
끝 노드
 디렉토리 20, 21
 샘플 APPN 네트워크에서 16
 설명 5
 APPN 17

[나]

내부 추적
 Windows의 원격 API 클라이언트 166
네트워크
 관리 133
 유형 2
 토폴로지 데이터베이스 23
 혼합 31
네트워크 관리 데이터 10
네트워크 노드
 디렉토리 20, 21
 샘플 구성 16
네트워크 노드 서버 5, 16
네트워크 데이터 파일
 설명 39
 AIX 또는 Linux의 원격 API 클라이언트
 170
네트워크 액세스 가능 장치(NAU) 7
네트워크 액세스 프로세스(NAP) 150
네트워크 주소 지정 가능 장치 7
노드
 구성 방법 62
 구성 파일 38
 매개변수 63
 목적 62
 부속 영역 2
 워크시트 175
 유형 3, 5
 주변 2
 추가 구성 필요성 63
 피어 2

노드 (계속)
 SNA 2
노드 연산자 기능(NOF) API 7
노드 자원 61
노드 창 51
노드 ID 매개변수 63
노드의 SNA 네트워크 이름 매개변수 95
논리 레코드 13
논리 장치(LU) 7

[다]

다운스트림 LU 이름 매개변수 130
다운스트림 PU 이름 매개변수 81, 84
다중 경로 채널
 워크시트 185
다중 경로 채널(MPC)
 포트 구성 68
단말기 4
대화
 보안 115
 설명 12
대화 레벨 보안 필요 매개변수 106
대화 보안
 구성 방법 115
 매개변수 116
대화 유형 매개변수 106
대화상자 48
데이터 링크 제어(DLC) 67
데이터 파일
 기동 가능 TP 39
 노드 구성 38
 도메인 구성 39
 클라이언트 네트워크 40
 SNA 네트워크 39
 TP 정의 39
도메인 이름
 변경 144
도메인 자원 61
도메인 창 49
도움말
 명령행 관리 프로그램 59
 Motif 관리 프로그램 57

- 독립 LU
 - 구성 91
 - 설명 11
- 동기 레벨 매개변수 106
- 디렉토리
 - 끝 노드 20, 21
 - 네트워크 노드 20, 21
 - LEN 노드 19, 20
 - Linux용 Communications Server 실행 가능 프로그램 44
- 디폴트 폴의 멤버 매개변수 93
- 디폴트 LU 사용 매개변수 112

[라]

- 로그 메시지 40
- 로그 파일
 - 구성 64
 - 유형 64
- 로깅 65
 - Windows의 원격 API 클라이언트 159
- 로컬 노드
 - LU 10
- 로컬 노드 ID 매개변수 80
- 로컬 토폴로지 데이터베이스 23
- 로컬 IP 인터페이스 매개변수 72
- 로컬 LU
 - 구성 방법 92
 - 매개변수 93
 - 설명 10
 - 정의 92
 - 추가 구성 필요성 93
- 로컬 LU 매개변수 112, 115
- 로컬 LU 별명 매개변수 112
- 로컬 LU 이름 매개변수 98
- 로컬 SAP 번호 매개변수 71
- 링크 스테이션
 - 구성 73, 74
 - 매개변수 75, 76, 77, 78, 79, 80, 81
 - 설명 6
 - 추가 구성 필요성 81
- 링크 스테이션 경로지정
 - 매개변수 98
- 링크 스테이션 이름 매개변수 98

[마]

- 마스터 서버 61, 143
 - 지정 144
- 매개변수는 임의 LU 매개변수에서의 기동에 대한 것임 102
- 매체 액세스 제어(MAC) 29
- 명령
 - 구성 서버 수정 62
- 명령행 관리 프로그램
 - 도움말 59
 - 명령 유형 59
 - 사용 58
 - 설명 37
 - 클라이언트에서 59
- 모드 107
 - 구성 108
 - 매개변수 109, 110, 111
 - 설명 13
 - 추가 구성 필요성 111
 - 표준 107
- 모드 매개변수 113
- 목적지 호스트 주소 매개변수 127
- 목표 TP 12, 100
- 문서 내용 아키텍처(DCA) 8
- 문자, RCF 명령에서 135
- 문제점 판별 지원
 - 개요 40
 - 로깅 64
- 문제점 해결 40
- 물리 장치 제어점(PUCP) 9
- 물리 장치(PU) 7
- 미해석 이름 매개변수 97

[바]

- 백업 마스터 서버 61
- 백업 서버 143
- 백업 DLUS 이름 매개변수 83
- 버전 번호, NetView 133
- 버전, IP 주소 146
- 별명 매개변수 97
- 별명, 상대방 LU 96
- 병렬 세션 11
- 병렬 세션 지원 매개변수 97
- 보안
 - 대화 115
 - 세션 114

- 보안 (계속)
 - APPC 114
 - UCF 138, 141
- 보안 매개변수 113
- 보안 소켓 계층(SSL) 122, 126
 - 데이터 암호화 123, 126
 - 서버 인증 123, 127, 128
 - 클라이언트 인증 122, 126
- 보안 액세스 리스트
 - 구성 방법 116
 - 매개변수 116
 - 목적 116
 - 추가 구성 필요성 116
- 보안 액세스 리스트 매개변수 106
- 복수 세션 11
- 부속 영역 네트워크
 - 노드 유형 3
 - 설명 2
 - 예제 4
 - 전송 경로 선택 14
- 부속 영역 노드 3
- 부속 영역 SNA 1
- 브랜치 네트워크 노드 5, 30
- 브랜치 링크 유형 매개변수 79
- 브로드캐스트 검색 22

[사]

- 사용자 응용프로그램 지원 워크시트 190
- 사용자 ID 매개변수 104, 114, 116
- 상대방 LU 10
 - 구성 방법 96
 - 매개변수 96, 97, 98
 - 별명, 정의 96
 - 복수, 와일드카드로 정의 96
 - 원격 노드, 정의 96
 - 추가 구성 필요성 99
- 상대방 LU 매개변수 113, 115
- 상대방 LU 이름 매개변수 96, 98
- 상대방 TP 매개변수 113
- 상태
 - Windows의 원격 API 클라이언트 152
- 서버
 - 사용 가능화 44
 - 사용 불가능화 46
 - 제거 62
 - 추가 62
 - 클라이언트와의 관계 143

서비스 액세스점(SAP) 29
 서비스 클래스(COS) 14
 서비스점 133
 서비스점 명령 기능(SPCF) 38, 133
 세션
 경로지정 23
 설명 10
 유형 10
 세션 보안
 구성 방법 114
 매개변수 115
 추가 구성 필요성 115
 세션 시간종료 매개변수 110
 소스 TP 12, 100
 소프트웨어 사용 불가능화 46
 수신 페이징 창 매개변수 110
 시간종료 지정 매개변수 110
 시간종료 허용 매개변수 131
 시스템 서비스 제어점(SSCP) 9

[아]

암시적 링크에서 링크 레벨 오류 복구 매개변수 73
 암시적 링크에서 지원되는 HPR 매개변수 73
 암시적 PU 액세스에 대한 다운스트림 LU 구성 매개변수 73
 암호 매개변수 114, 115, 116
 압축 지원 매개변수 83, 111
 액세스 리스트 내의 사용자 매개변수 116
 액세스 제한 매개변수 106
 어댑터 카드 번호 매개변수 69
 업스트림 DLUS 이름 매개변수 81
 업스트림 LU 이름 매개변수 130
 연결 네트워크
 구성 68, 69
 구성 방법 69
 설명 15
 추가 구성 필요성 73
 토폴로지 정보 24
 APPN 28
 연결 네트워크에서 정의 매개변수 71, 72
 연결성
 구성 67
 설명 6
 워크시트 175, 178
 직접 27
 오류 로그 파일 40

와일드카드 96
 와일드카드 상대방 LU 이름 매개변수 96
 요청 단위(RU) 110
 워크시트 43
 원격 노드
 구성 방법 95
 노드의 SNA 네트워크 이름 매개변수 95
 상대방 LU 96
 정의 94
 추가 구성 필요성 95
 LU 10
 원격 노드 역할 매개변수 80
 원격 노드 유형 매개변수 79
 원격 노드 이름 매개변수 78
 원격 노드 ID 매개변수 80
 원격 명령 기능(RCF) 38
 원격 작업 입력(RJE) 9
 원격 X.25 주소 매개변수 78
 웹 관리 패키지
 설명 37
 위치 매개변수 98
 유형 2.0 노드 3
 유형 2.1 노드 3
 유형 4 노드 3
 유형 5 노드 3
 응용프로그램
 워크시트 175
 프로그램 6
 응용프로그램 프로그래밍 인터페이스(API) 6
 이더넷
 워크시트 182
 포트 구성 68
 이더넷 유형 매개변수 72
 이름 매개변수
 링크 스테이션 75
 모드 109
 보안 액세스 리스트 116
 CPI-C 기호식 목적지 112
 LU 풀 89
 이탈 문자, RCF 135
 인수 매개변수 104
 인접 노드 15
 일반 데이터 스트림(GDS) 8
 입력 Allocate를 실행 중인 TP로 전송 매개변수 104

[자]

자동 네트워크 경로지정(ANR) 14, 23
 자동 활성 세션 수 매개변수 110
 자원 이름 19
 자원 찾기 19
 자원, 찾기 19
 전송 경로 14
 전송 경로 선택 14, 23, 25
 전송 그룹(TG) 23
 전송 네트워크 14
 전체 로깅 41
 전체 CP 이름 19
 전체 LU 이름 19
 정보 관리 시스템/가상 기억영역(IMS/VS) 9
 제어 데이터 10
 제어점 별명 매개변수 63
 제어점 이름 매개변수 63
 제어점(CP) 9
 종속 노드 3
 종속 LU
 구성 85
 설명 11
 종속 LU 서버(DLUS) 31
 주변 노드 3
 중계 경로지정 26
 중계 세션 경로지정(ISR) 23, 26
 중앙 로깅 41
 중지 명령 47
 지시된 검색 22
 지연된 로그인 매개변수 131
 지원되는 복사 인스턴스 매개변수 103
 지정된 표시장치 LU 매개변수 122
 지정된 프린터 LU 매개변수 122
 지정된 LU 매개변수 89
 직접 대화 13
 직접 연결성 27
 진단 도구 40

[차]

창
 노드 48, 51
 도구 모음 버튼 54
 도메인 48, 49
 메뉴 48
 설명 48
 자원 48

창 (계속)
 자원 항목 53
 CPI-C 목적지 이름 49
 LU 풀 49
 초기 세션 한계 매개변수 109
 초기 창 크기 매개변수 110
 초기에 활동 중 매개변수 70, 83
 최대 세션 한계 매개변수 109
 최대 창 크기 매개변수 110
 최대 활동 템플릿 인스턴스 수 73
 최대 RU 크기 매개변수 111
 최대 RU 크기 제한 매개변수 111
 최소 회선경합 성공 세션 수 매개변수 109
 최소 회선경합 실패 세션 수 매개변수 109
 추적
 클라이언트/서버 169
 LAN 169
 추적 파일 41

[카]

커널 구성요소 추적 45
 커널 구성요소, 추적 45
 커널 메모리 한계 45
 클라이언트
 관리 143
 기동 가능 TP 구성 197
 네트워크 데이터 파일 40
 네트워킹 요구사항 146
 ARGUMENTS 매개변수 200
 PATH 매개변수 200
 SECURITY_TYPE 매개변수 205
 SERVICE_NAME 매개변수 205
 SHOW 매개변수 204
 TIMEOUT 매개변수 202
 TP 정의 174
 TPname 매개변수 200
 TYPE 매개변수 200
 클라이언트 구성 유틸리티, Windows 153
 클라이언트/서버
 구성 61
 추적 164, 169
 클러스터 제어기 4

[타]

타스크 시트 43
 터미널 제어기 4

텔넷 클라이언트 주소 매개변수 125
 토큰링
 워크시트 180
 포트 구성 68
 토폴로지 데이터베이스 갱신(TDU) 25
 토폴로지 및 경로지정 서비스(TRS) 23
 토폴로지 정보 10
 로컬 24
 연결 네트워크 24
 통신 링크 4
 통신 제어기 3
 통신 제어기 노드 3
 트랜잭션 프로그램(TP) 6
 특정 LU로의 액세스 허용 매개변수 122

[파]

패스-스루 서비스
 구성 119
 워크시트 175, 187
 패스-스루 DLUR 84
 포트
 구성 68, 69
 매개변수 69, 70, 71, 72, 73
 추가 구성 필요성 73
 포트 번호 매개변수 69
 풀 주소 매개변수 76
 표시장치 LU 매개변수 124
 표준 오류 매개변수 105
 표준 입력 매개변수 105
 표준 출력 매개변수 105
 풀 내의 LU 매개변수 87
 풀 이름 매개변수 87
 프론트 엔드 프로세서(FEP) 3
 프로토콜 1
 프로토콜 매개변수 72
 프린터 4
 프린터 LU 매개변수 124
 피어 네트워크 2
 노드 유형 5
 전송 경로 선택 14
 피어간 통신 1

[하]

하위 네트워크(LEN) 노드 5
 형식 1
 호스트 3

호스트 노드 3
 호스트 LS/DLUR PU 매개변수 86, 93
 혼합 네트워크 2, 31
 환경 매개변수 105
 활성화 매개변수 75
 회선 세부사항 매개변수 70

[숫자]

1차 LU 11
 2차 LU 11
 3270
 워크시트 194
 풀 구성 88
 LU 구성 85
 5250
 워크시트 194

A

AIX 또는 Linux의 원격 API 클라이언트
 관리 169
 서버 이름 173
 broadcast_attempt_count 172
 invoked_tps 171
 lan_access_timeout 171
 server_lost_timeout 172
 * 172
 ANR
 동적 경로 재지정 26
 설명 14, 23
 API
 설명 6
 소유권 7
 Linux용 Communications Server와 함께
 포함됨 6
 API 추적
 Windows의 원격 API 클라이언트 162
 APPC
 구성 91
 보안 114
 워크시트 191
 APPCLLU
 Windows의 원격 API 클라이언트 167
 APPCTPN
 Windows의 원격 API 클라이언트 167
 Application System/400(AS/400) 15

APPN
 기능 15
 끝 노드 5, 17, 176
 네트워크 14, 15, 27
 네트워크 노드 5, 16, 175
 네트워크 예제 15
 노드 유형 15
 브랜치 네트워크 노드 5, 176
 설명 1, 14
 연결 네트워크 28
 전송 경로 선택 25
 제어점(CP) 18
 APPN 지원 매개변수 63
 AS/400(Application System/400) 15

B

BIND 요청 11
 Branch Extender 30

C

CICS(고객 정보 제어 시스템) 8
 CN 이름 매개변수 72, 73
 CN(연결 네트워크) 15
 COS
 목적 107
 설명 14
 유형 107
 COS 이름 매개변수 109
 CPI-C 부가 정보
 구성 방법 112
 매개변수 112, 113, 114
 추가 구성 필요성 114
 CPI-C(통신용 공통 프로그래밍 인터페이스)
 부가 정보 112
 워크시트 193
 CP(제어점) 9
 CP-CP 세션 10
 CSVTBLG
 Windows의 원격 API 클라이언트 168

D

DCA(문서 내용 아키텍처) 8
 DDDL
 TN3270용 LU 모델 207

DLC
 구성 67, 69
 구성 방법 69
 추가 구성 필요성 73

DLUR
 구성 131
 설명 31
 워크시트 187
 추가 구성 필요성 84

DLUR PU
 구성 방법 82
 매개변수 83, 84

DLUR 다운스트림 노드 84

DLUS
 설명 31

DLUS 이름 매개변수 83, 84

DLUS 접속 무한정 재시도 매개변수 84

domain
 구성 파일 39
 설명 3

E

Enterprise Extender(HPR/IP)
 워크시트 185
 포트 구성 68
 ENV 매개변수 204
 EN(끝 노드) 5

F

FEP(프린트 엔드 프로세서) 3

G

GDS(일반 데이터 스트림) 8
 GROUP 매개변수 203

H

HPR
 설명 14, 23
 HTTPS 149

I

IMS/VS(정보 관리 시스템/가상 기억영역) 9
 IP 주소 형식 146

IP 포트 번호 147
 IPv4 주소 146
 IPv6 주소 146
 ISR 23, 26

L

LAN 액세스 시간종료 148

LAN 추적
 클라이언트에서 169

LEN 노드
 기능 15
 디렉토리 19, 20
 설명 5, 17
 워크시트 177

Linux 명령 133

Linux 클라이언트

도메인 이름 171
 maximum_element_count 171
 maximum_header_count 171
 maximum_process_count 171

Linux용 Communications Server 소프트웨어

사용 가능화

서버에서 44

초기화 중 문제점 46

Windows의 원격 API 클라이언트 151

Linux용 Communications Server 소프트웨어

사용 불가능화

Windows의 원격 API 클라이언트 152

Linux용 Communications Server 실행 기능

프로그램의 경로 44

LS(링크 스테이션) 73

LU

설명 7
 유형 8

LU 0

설명 9

LU 1 8

LU 2 8

LU 3 8

LU 6.2

구성 91

설명 8

LU 번호 매개변수 86, 93, 130

LU 별명 매개변수 93, 103

LU 유형 매개변수 86

LU 유형 0-3

구성 방법 85

LU 유형 0-3 (계속)
 매개변수 86, 87
 추가 구성 필요성 87
 LU 이름 매개변수 86, 93
 LU 트래픽 매개변수 76
 LU 풀
 구성 방법 88
 매개변수 89
 보기 88
 정의 88
 LUA
 구성 85
 워크시트 195
 풀 구성 88
 LUALIAS 매개변수 203
 LU-LU 세션 10

M

MAC 주소 매개변수 77
 MAC(매체 액세스 제어) 29
 Motif 관리 프로그램
 기동 48
 노드 창 51
 대화상자 55, 57
 도구 모음 버튼 54
 도메인 창 49
 도움말 57
 사용 47
 설명 35
 자원 창 48
 자원 항목 53
 MPC
 포트 구성 68
 MPC 그룹 매개변수 78
 MS(관리 서비스) 17

N

NAP(네트워크 액세스 프로세스) 150
 NAU(네트워크 액세스 가능 장치) 7
 NetView
 명령 133
 명령 입력 영역 크기 변경 134
 버전 번호 133
 서비스점 133
 설명 133
 프로그램 133

NetView (계속)
 화면 표시장치 134
 NN(네트워크 노드) 5
 NOF(노드 연산자 기능) API 40

P

PIP 허용 매개변수 107
 PU
 설명 7
 DLUR용 82
 PU ID 매개변수 83
 PU 이름 매개변수 83
 PUCP(물리 장치 제어점) 9

Q

QLLC
 워크시트 183
 포트 구성 68

R

RCF
 기능 38
 명령 구문 135
 유효 문자 135
 RJE(원격 작업 입력) 9
 RTP
 끝점 26
 설명 14, 23
 RU(요청 단위) 110

S

SAP 번호 매개변수 77
 SAP(서비스 액세스점) 29
 SATF
 직접 연결성 27
 APPN 네트워크에서 28
 SDLC
 워크시트 178
 포트 구성 68
 SEND 함수 13
 SNA
 계층 2
 계층 구조 2
 기본 개념 2

SNA (계속)
 네트워크 1
 네트워크 데이터 파일 39, 170
 네트워크 유형 2
 부속 영역 1
 설명 1
 APPN 개념 14
 SNA 게이트웨이
 목적 129
 워크시트 188
 SNA 게이트웨이의 다운스트림 LU
 구성 방법 130
 매개변수 130, 131
 추가 구성 필요성 131
 SNA 네트워크 정보
 Windows의 원격 API 클라이언트 152
 SNA 소프트웨어 사용 기능화
 AIX 또는 Linux의 원격 API 클라이언트 169
 SNA 포트 이름 매개변수 69, 75
 snaadmin 프로그램 37
 snanetutil 프로그램 144
 SNA를 정의된 값으로 재설정 매개변수 111
 sna_clnt.net 파일 170
 SPCF
 명령 136
 명령 구문 135
 설명 38, 133
 SSCP 종속 LU 11
 SSCP(시스템 서비스 제어점) 9
 SSCP-LU 세션 10
 SSCP-PU 세션 10
 start 명령 45

T

TCP/IP 포트 번호 매개변수 121, 126, 128
 TDU(토폴로지 데이터베이스 갱신) 25
 TN 방향 재지정자
 액세스 레코드 125
 액세스 레코드 매개변수 125, 126, 127, 128
 워크시트 189
 TN 서버
 액세스 레코드 120, 123
 액세스 레코드 매개변수 121, 122
 연결 레코드 124
 연결 레코드 매개변수 124

TN 서버 (계속)
 워크시트 188
 TN3270 클라이언트
 DDDLU 207
 DDDLU용 LU 모델 207
 TN3270 클라이언트 주소 매개변수 121
 TN3270E 지원 매개변수 121
 TP
 구성 99
 구성 방법 101
 기동 12, 100
 기동 가능 12, 100
 기동 매개변수 102, 103, 104, 105
 목표 12, 100
 설명 6
 소스 12, 100
 클라이언트 174
 APPC 정의 매개변수 106, 107
 TP 구성 매개변수
 ENV 204
 GROUP 203
 LUALIAS 203
 USERID, AIX 또는 Linux 202
 USERID, Windows 205
 TP 실행 프로그램의 전체 경로 매개변수 104
 TP 이름 매개변수 102, 106
 TRS(토폴로지 및 경로지정 서비스) 23

U

UCF
 권한 138
 디먼 프로그램 138
 명령 구문 135, 138
 명령 취소 140
 보안 138, 141
 사용 138
 사용자 138
 사용자 이름 141
 사용할 수 있는 명령 139
 샘플 명령 139
 설명 38, 133
 유효한 명령 139
 출력 140
 파일 액세스 141
 UDP/IP 통신 147
 UNIX 명령 기능(UCF) 38

USERID 매개변수
 AIX 또는 Linux 202
 Windows 205
 ux-cancel 명령 140

V

VRN
 설명 29
 VTAM(가상 터미널 액세스 방법) 15

W

WebSphere Application Server 149
 Windows Open Systems
 Architecture(WOSA) 150
 Windows 클라이언트
 네트워크 액세스 프로세스(NAP) 150
 Windows의 원격 API 클라이언트
 구성 152
 구성 정보 155
 가동된 TP 155
 내부 추적 정보 166
 로깅 정보 159
 사용 가능화 151
 사용 불가능화 152
 상태 152
 서버 정보 157
 클라이언트 구성 유틸리티 153
 클라이언트/서버 추적 정보 164
 admin_msg 165
 all_api 163
 API 추적 정보 162
 appc 163
 APPCLLU 168
 APPCTPN 168
 audit_file 161
 audit_file_wrap_size 161
 audit_logging_enabled 159
 backup_audit_file 161
 backup_error_file 160
 broadcast_attempt_count 156
 client_start_timeout 156
 cpic 164
 CPI-C 응용프로그램 데이터 167
 csv 164
 CSV 응용프로그램 데이터 168
 CSVTBLG 168

Windows의 원격 API 클라이언트 (계속)

data 165
 datagram 165
 domain 155
 error_file 160
 error_file_wrap_size 160
 exception_logging_enabled 159
 file1 162
 file1(CS_tracing) 164
 file1(Internal_tracing) 166
 file2 162
 file2(CS_tracing) 165
 file2(Internal_tracing) 166
 flip_size 163
 flip_size(CS_tracing) 165
 flip_size(Internal_tracing) 166
 lan_access_timeout 155
 log_directory 160
 maximum_element_count 155
 maximum_header_count 155
 maximum_process_count 155
 nof 164
 receive 166
 rui 164
 send 166
 Server1 157
 Server2-Server9 158
 server_lost_timeout 156
 succinct_audits 162
 succinct_errors 161
 trace_flushing 167
 trace_level 167
 truncation_length 163
 WOSA(Windows Open Systems
 Architecture) 150

X

xsnaadmin 프로그램 35



SA30-2286-03

