

IBM Communications Server for Data Center
Deployment on Linux
バージョン 7.1

管理ガイド



注記

本書および本書で紹介する製品をご使用になる前に、[173 ページの『付録 E 特記事項』](#)に記載されている情報をお読みください。

第 7 版 (2021 年 1 月)

本書は、IBM Communications Server for Data Center Deployment on Linux バージョン 7.1 (プログラム番号 5725-H32) に適用されます。また、新しい版またはテクニカル・ニュースレターで明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

資料のご注文方法については、<http://www.ibm.com/jp/manuals> の「ご注文について」をご覧ください。(URL は、変更になる場合があります)

IBM にお客様のご意見をお寄せください。本資料に関するご意見やご感想は、次の URL からお送りください。今後の参考にさせていただきます。<http://www.ibm.com/jp/manuals/main/mail.html> フォームが削除されている場合は、以下の住所までご意見をお送りください。

- International Business Machines Corporation
- Department CGMD
- P.O. Box 12195
- Research Triangle Park, North Carolina
- 27709-2195
- U.S.A.

ファックスまたはインターネットをご使用の場合は、下記宛てにお願いします。

- IBMLink: RALVM17 の CIBMORCF
- IBM Mail: IBMMAIL の USIB2HPD
- インターネット: USIB2HPD@vnet.ibm.com
- FAX: 1-800-227-5088

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

© Copyright International Business Machines Corporation 1998, 2021.

目次

表.....	vii
図.....	ix
本書について.....	xi
本書の対象読者.....	xi
本書の使用法.....	xi
本書の構成.....	xi
表記上の規則.....	xii
グラフィックの規則.....	xiii
本リリースで新しくなった部分.....	xiii
新しい機能.....	xiii
廃止された機能.....	xiv
本書の追加情報の入手先.....	xiv
第 1 章 SNA の用語と概念.....	1
システム・ネットワーク体系.....	1
SNA の基本概念.....	1
ネットワークのタイプ.....	2
SNA ノード.....	2
接続.....	4
トランザクション・プログラム.....	4
アプリケーション・プログラミング・インターフェース.....	5
ネットワーク・アクセス可能単位.....	5
セッション.....	7
会話.....	9
モード.....	10
経路選択.....	10
サービス・クラス.....	11
APPN の基本概念.....	11
APPN ノード・タイプ.....	11
APPN 制御点.....	13
リソースの検出.....	14
セッション・ルーティング.....	17
分岐エクステンダー.....	23
APPN ネットワークからサブエリア・ネットワークへのアクセス.....	24
第 2 章 CS Linux の管理.....	27
CS Linux の管理の概要.....	27
管理責任.....	27
管理ツール.....	28
管理許可.....	33
CS Linux 構成の計画.....	33
計画ワークシート.....	33
作業シート.....	34
ローカル・システムの CS Linux を使用可能および使用不可にする方法.....	34
CS Linux による使用のための環境変数の設定.....	34
CS Linux プログラムへのパスの指定.....	34
CS Linux サーバーの有効化.....	35

CS Linux サーバーを使用不可に設定する方法.....	36
Motif 管理プログラムの使用.....	37
Motif 管理プログラムの呼び出し.....	37
リソース・ウィンドウ.....	37
Resource dialogs.....	43
状況ダイアログ.....	44
ヘルプ・ウィンドウ.....	45
コマンド行管理プログラムの使用.....	46
第 3 章基本構成作業.....	47
クライアント/サーバー機能の構成.....	47
ノードの構成.....	48
ノードの構成パラメーター.....	48
追加構成.....	49
ロギングの構成.....	49
第 4 章 接続コンポーネントの定義.....	51
DLC、ポート、および接続ネットワークの定義.....	52
DLC、接続ネットワーク、およびポートの構成パラメーター.....	52
追加構成.....	56
リンク装置の定義.....	56
リンク・ステーションの構成パラメーター.....	57
追加構成.....	61
DLUR PU の定義.....	62
DLUR PU の構成パラメーター.....	62
ダウンストリーム・ノードのパススルー DLUR 用のパラメーター.....	63
追加構成.....	63
第 5 章従属 LU の構成.....	65
LU タイプ 0 から 3 の定義.....	65
LU タイプ 0 から 3 の構成パラメーター.....	65
追加構成.....	67
LU プールの定義.....	67
LU プールの構成パラメーター.....	67
第 6 章 APPC 通信の構成.....	69
ローカル LU の定義.....	70
ローカル LU の構成パラメーター.....	70
追加構成.....	71
リモートノードの定義.....	71
リモート・ノードの構成パラメーター.....	72
追加構成.....	72
パートナー LU の定義.....	72
パートナー LU の構成パラメーター.....	73
パートナー LU のリンク・ステーション・ルーティングの定義.....	74
追加構成.....	75
TP の定義.....	75
サーバー上での TP 呼び出しパラメーター.....	77
TP 定義パラメーター.....	79
モードおよびサービス・クラスの定義.....	80
モードの構成パラメーター.....	82
追加構成.....	84
CPI-C サイド情報の定義.....	84
CPI-C の構成パラメーター.....	84
追加構成.....	86
APPC セキュリティーの構成.....	86
セッションセキュリティの構成.....	86

会話セキュリティーの構成.....	87
セキュリティー・アクセス・リストの構成.....	87
第 7 章 ユーザー・アプリケーションの定義.....	89
第 8 章 パススルー・サービスの構成.....	91
TN サーバーの構成.....	91
TN サーバー・アクセス・レコードの構成.....	92
TN サーバー関連付けレコードの構成.....	94
TN リダイレクターの構成.....	95
TN リダイレクターアクセスレコードの構成.....	95
SNA ゲートウェイの構成.....	98
ダウンストリーム LU の構成パラメーター.....	99
追加構成.....	99
DLUR の構成.....	99
TN サーバーおよび TN リダイレクター: 追加のセキュリティー情報.....	100
第 9 章 NetView からの CS Linux の管理.....	103
ホスト NetView プログラムの使用.....	103
NetView 画面表示.....	104
コマンド入力域のサイズの変更.....	104
RCF コマンド構文の概要.....	104
大文字とエスケープ文字.....	105
SPCF の使用.....	105
SPCF で使用される管理コマンドに関する制約事項.....	105
SPCF コマンドの例.....	106
UCF の使用.....	106
UCF コマンド構文.....	107
使用できるコマンド.....	107
UCF コマンドの例.....	107
Linux システム・コマンドからの出力.....	108
コマンドの取り消し.....	108
UCF のセキュリティー.....	109
第 10 章 CS Linux クライアント / サーバー・システムの管理.....	111
クライアント/サーバー構成の変更.....	112
異なるドメインへのクライアントの移動.....	112
IP ネットワーキングの要件.....	113
IPv4 および IPv6 アドレッシング.....	113
クライアント/サーバー構成でのホスト名.....	113
IP ポート番号のセットアップ.....	114
LAN アクセス・タイムアウト.....	114
リモート API クライアントの HTTPS アクセス.....	115
Windows でのリモート API クライアントの管理.....	115
Windows 上の Remote API Client を使用可能に設定する.....	116
Windows でのリモート API クライアントの状況の表示.....	117
Windows 上の Remote API Client を使用不可に設定する.....	117
Windows 構成上のリモート API クライアント.....	118
Tracking SNA LU resources used by clients on a domain of servers.....	129
クライアント・アプリケーションでハードコーディングされた LU 別名をサーバーのドメイン内の LU 別名にマッピングする.....	130
AIX または Linux 上の Remote API Client の管理.....	132
AIX または Linux でのリモート API クライアントの使用可能化および使用不可.....	132
クライアント・ネットワーク・データ・ファイル (sna_clnt.net).....	133
Tracking SNA LU resources used by clients on a domain of servers.....	136
クライアント・アプリケーション上のハード・コード LU 別名をサーバーのドメイン内の LU 別名にマップする.....	136

クライアント TP の定義.....	138
付録 A 構成計画ワークシート.....	139
ノード・ワークシート.....	139
APPN ネットワーク・ノード.....	139
APPN エンド・ノード.....	140
APPN 分岐ネットワーク・ノード.....	140
LEN ノード.....	141
接続ワークシート.....	142
SDLC.....	142
トークンリング.....	144
イーサネット.....	146
QLLC (X.25).....	148
マルチパス・チャネル.....	149
Enterprise Extender (HPR/IP).....	150
パススルー・サービス・ワークシート.....	151
ローカル・ノード上の DLUR.....	151
ダウンストリーム・ノード用のパススルー DLUR.....	152
SNA ゲートウェイ.....	152
TN サーバー.....	153
TN Redirector.....	154
ユーザー・アプリケーション・サポート・ワークシート.....	155
APPC.....	155
CPI-C.....	159
5250.....	159
3270.....	160
LUA.....	160
付録 B コマンド行からの呼び出し可能 TP の構成.....	163
呼び出し可能 TP 定義のファイル形式.....	164
付録 C DDDLU 用の TN3270 LU モデルの構成.....	169
付録 D IBM へのご意見の送付方法.....	171
E メール・フィードバック・テンプレート.....	171
技術的な問題がある場合.....	171
付録 E 特記事項.....	173
商標.....	174
参考文献.....	177
CS Linux バージョン 7.1 の資料.....	177
システム・ネットワーク体系 (SNA) 関連資料.....	178
ホスト構成関連資料.....	178
z/OS Communications Server 関連資料.....	179
TCP/IP 関連資料.....	179
X.25 関連資料.....	179
APPC 関連資料.....	179
プログラミング関連資料.....	179
その他の IBM ネットワーキング関連資料.....	179
索引.....	181

表

1. 表記上の規則.....	xii
2. 標準モードおよび COS 名.....	81
3. RCF コマンドでのエスケープ文字の使用.....	105



1. SNA サブエリア・ネットワーク	3
2. 複数セッションと並列セッション	9
3. トランザクション・プログラムと論理装置との通信	10
4. サンプル APPN ネットワークの一部	12
5. LEN ノード・ディレクトリー	15
6. エンド・ノード・ディレクトリー	16
7. ネットワーク・ノード・ディレクトリー	17
8. ネットワーク・ノードのネットワーク・トポロジー・データベース	19
9. 共有アクセス転送機能を使用する APPN ネットワーク	21
10. ノード EN1 から APPN ネットワーク内の各ノードへの直接リンクに必要な定義	22
11. 仮想ノードを使用する直接リンクに必要な定義	23
12. 分岐エクステンダー	24
13. CS Linux Domain Window.....	38
14. ノード・ウィンドウ.....	40
15. CS Linux ツールバー.....	42
16. Sample Dialog.....	43
17. 状況ダイアログの例	44
18. ヘルプ・ウィンドウの例	45
19. NetView 画面の例	104

本書について

本書は、IBM Communications Server for Data Center Deployment on Linux (プログラム製品番号 5725-H32) の使用可能化、構成、および管理のためのガイドです。この IBM® ソフトウェア製品により、Linux が稼働するコンピューターで SNA (システム・ネットワーク体系) ネットワーク上の他のノードとの情報交換が可能になります。

IBM Communications Server for Data Center Deployment on Linux には、この製品が動作するハードウェアに応じて、2つの異なるインストール方法があります。

CS Linux

CS Linux は、次のハードウェアで動作します。

- Linux が稼働する 64 ビット AMD64/Intel EM64T ワークステーション (x86_64)
- Linux が稼働する IBM Power コンピューター (ppc64le)

CS Linux for IBM Z

CS Linux for IBM Z は、Linux for IBM Z が稼働する IBM Z メインフレーム上 (s390x) で作動します。

本書では、相違が明示的に記述されていない限り、CS Linux という名称はこれらの2つのバリエーションのいずれかを示すために使用され、「CS Linux コンピューター」という用語は、CS Linux が稼働しているすべてのコンピューターの種類を示す場合に使用されます。

本書は CS Linux のバージョン 7.1 に適用されます。

本書の対象読者

本書は、CS Linux を使用するシステム管理者とアプリケーション・プログラマーを対象としています。

システム管理者

システム管理者は、CS Linux をインストールし、システムをネットワーク接続用に構成し、システムの保守を行います。システム管理者は CS Linux が稼働するハードウェアおよび Linux オペレーティング・システムについて理解している必要があります。また、システム管理者は、システムを接続するネットワークに関する知識があり、SNA の一般的な概念について理解している必要があります。

アプリケーション・プログラマー

アプリケーション・プログラマーは、CS Linux プログラミング・インターフェースを使用して SNA ネットワーク上でデータを送受信するトランザクション・プログラムおよびアプリケーション・プログラムの設計とコーディングを行います。したがって、アプリケーション・プログラマーは、SNA、トランザクション・プログラムまたはアプリケーション・プログラムの通信相手のリモート・プログラム、および Linux オペレーティング・システムのプログラミング環境と操作環境について十分理解している必要があります。

アプリケーション・プログラムの作成に関する詳細情報は、各 API のマニュアルに記載されています。CS Linux の資料に関する追加情報については、『参考文献』を参照してください。

本書の使用法

本書は、CS Linux の使用可能化、構成、および管理の方法について説明しています。

本書の構成

本書は次の各章で構成されています。

- 1 ページの『[第 1 章 SNA の用語と概念](#)』では、SNA および APPN (拡張対等通信ネットワーク機能) の概念について概説します。
- 27 ページの『[第 2 章 CS Linux の管理](#)』では、CS Linux の管理ツールについて解説し、CS Linux の構成を準備する方法、サーバー上で CS Linux ソフトウェアを使用可能または使用不可にする方法、および Motif とコマンド行管理プログラムを使用する方法を説明します。

- 47 ページの『第 3 章 基本構成作業』では、CS Linux サーバーの基本的な構成タスクの実行方法を解説し、クライアント/サーバー操作の構成、SNA ノードの構成、および CS Linux 用のメッセージ・ロギングの構成についても説明します。
- 51 ページの『第 4 章 接続コンポーネントの定義』では、CS Linux ノードの接続を構成する方法を説明します。
- 65 ページの『第 5 章 従属 LU の構成』では、LU タイプ 0 から 3 と LU プールの 従属 LU (論理装置) の構成方法を説明します。
- 69 ページの『第 6 章 APPC 通信の構成』では、APPC (拡張プログラム間通信機能 (advanced program-to-program communications)) の構成方法を説明します。
- 89 ページの『第 7 章 ユーザー・アプリケーションの定義』では、ユーザー・アプリケーションの構成方法を説明します。
- 91 ページの『第 8 章 パススルー・サービスの構成』では、パススルー・サービスの構成方法を説明します。パススルー・サービスは、直接には接続できないホスト・システムとローカル・システムとの間の通信をサポートする機能です。
- 103 ページの『第 9 章 NetView からの CS Linux の管理』では、CS Linux のリモート・コマンド機能 (RCF) を使用して CS Linux を管理する方法、および NetView を実行中のホストから CS Linux ノード上でコマンドを実行する方法を説明します。
- 111 ページの『第 10 章 CS Linux クライアント / サーバー・システムの管理』では、IBM Remote API Clients の構成および管理方法を説明します。
- 139 ページの『付録 A 構成計画ワークシート』には、CS Linux 用の構成ワークシートが収録されています。
- 163 ページの『付録 B コマンド行からの呼び出し可能 TP の構成』では、TP インストール・プログラムのユーザーまたは作成者が、呼び出し可能 TP を定義できるようにする、コマンド行ユーティリティーに関する情報を説明します。
- 169 ページの『付録 C DDDLU 用の TN3270 LU モデルの構成』では、tn3270dev.dat ファイルについて説明します。このファイルにより、クライアントが DDDLU を使用する場合、TN3270 クライアントの装置タイプとホストで使用される LU モデルとの間のマッピングを変更することができます。

表記上の規則

本書では、xii ページの表 1 に示すような表記上の規則を使用しています。

表 1. 表記上の規則

内容	表記例
強調したい語句	削除する前にファイルのバックアップをとります。
資料名	<i>IBM Communications Server for Data Center Deployment on Linux 管理ガイド</i>
ファイル名またはパス名	/usr/spool/uucp/myfile.bkp
プログラムまたはアプリケーション	snaadmin
パラメーターまたは Motif のフィールド	opcode; LU name
ユーザーが入力できるリテラル値または選択値 (デフォルト値を含む)	255; 0n node startup
Motif のボタン	Status (状況)
Motif のメニュー	Services (サービス)
Motif のメニュー項目	Configure node parameters (ノード・パラメーターの構成)
ユーザー入力	0p1
コンピューター出力	CLOSE

表 1. 表記上の規則 (続き)

内容	表記例
コマンドまたは Linux ユーティリティ	define_node; cd
特定のタイプのすべてのコマンドへの一般的な参照	query_* (リソースの詳細を照会する管理コマンドをすべて示します)
オプションまたはフラグ	-i
指定する値を表す変数	filename; LU_name; user_ID
戻り値	0、-1
3270 のキー	ENTER
キーボード・キー	Ctrl+D; Enter
16 進値	0x20
環境変数	PATH
関数、コール、またはエントリー・ポイント	ioctl
プログラミング verb	GET_LU_STATUS

グラフィックの規則

UNIX

このシンボルは、AIX または Linux オペレーティング・システムのみにも適用される本文の部分の開始を示すために使用されます。これは、Linux サーバーと、AIX、Linux、Linux for Power、または Linux for IBM Z 上で稼働する IBM Remote API Client に適用されます。

WINDOWS

このシンボルは、Windows 上の IBM Remote API Client のみに適用される本文の部分の開始を示すために使用されます。



このシンボルは、オペレーティング・システムに固有の本文の部分が終了したことを示します。このシンボルよりあとにある情報は、オペレーティング・システムに関係なく適用されます。

本リリースで新しくなった部分

CS Linux バージョン 7.1 は、CS Linux バージョン 7.0 の後継製品として、引き続きサポートされています。

新しい機能

このリリースで CS Linux に追加された機能は、次のとおりです。

- Windows クライアントで使用されているアプリケーションの数の表示。

Windows クライアント・モニター・プログラム `sxclapp1.exe` が機能拡張され、この Windows クライアントを使用しているアプリケーション (APC、LUA、PCOMM など) の数が表示されるようになりました。

- クライアント・アプリケーション上のハードコーディングされた LU 別名から、サーバーのドメインの LU 別名へのマッピングの機能拡張。

アプリケーションによっては、特定の LU 別名を使用するためにハードコーディングされるものがあります。2 つ以上の Remote API Client でそのようなアプリケーションを実行する場合、この機能は (アプリケーションの各コピーに異なる LU を使用して) それらの LU 別名をサーバー上の特定の LU にマップします。これにより、SNA のフルスタック実装から移行するアプリケーションを、サーバー上の特定の LU 別

名に対して構成することができます。また、サーバー・ドメイン内のサーバーのすべてのリソースを使用して、そのサーバー・ドメイン実装内のいくつかの冗長機能を利用できます。このリリースでは、その機能にさらに構成オプションが追加されています。

- ドメイン・サーバー間のポーリング・タイマーの追加。

この機能により、マルチサーバー・クライアント・サーバー構成でのフェイルオーバーとリカバリーが迅速化されます。

- 名前に 2 桁の 10 進数を使用して、LU 0 から LU 3 の範囲を定義する機能。

この機能により、より柔軟に LU のプールに名前を付けることができます。

- tn3270 侵入検出サービスのサポート。

この機能は、VTAM の機能拡張との相互作用により、3270 データ・ストリームの特定の違反を検出できるようにします。

- Docker コンテナのサポート。

この機能により、Linux、AIX、または Windows の Remote API Client を Docker コンテナ内で実行できます。

- 最新の暗号を解く鍵のサポート。

製品の tn3270 サーバーと tn リダイレクターは、TLS 1.0、1.1、1.2、1.3 暗号化標準と、それらに関連するアルゴリズムや暗号スイートをサポートします。最新の暗号化のサポートについては、README ファイルを参照してください。

- IPv6 接続ネットワーク・リンクでの 16 進アドレスの使用可能化。

HPR/IP の IPv6 ネットワークで接続ネットワークを定義するときに、接続ネットワークの IP アドレッシングで IPv6 DNS 名のみを使用するか、IPv6 アドレスのみを使用するかを示す追加のパラメーターを指定できます。デフォルトでは、IPv6 DNS 名のみを使用します。

- Linux の Ubuntu バージョンのサポート。

RedHat Enterprise Linux と SUSE Linux Enterprise Server に加えて、Ubuntu Linux でもこの製品を使用できるようになりました。最新の O/S のサポートについては、README ファイルを参照してください。

- Linux on IBM Power の ppc64le バージョンのサポート。

この製品は、ppc64le Linux を実行している IBM Power コンピューターで使用できるようになりました。

- LLC2 DLC のイーサネット・デバイス名を構成する機能。

RHEL は通常、ens32 などの形式の名前を使用しますが、以前は eth0、eth1 などの形式の名前のみがサポートされていました。

廃止された機能

Linux (i686) を実行する 32 ビット Intel ワークステーションのサポート。

ppc64 Linux を実行する IBM Power コンピューター。ppc64le Linux を実行する IBM Power コンピューターがサポートされるようになりました。

本書の追加情報の入手先

IBM Communications Server for Data Center Deployment on Linux ライブラリーに含まれている他の資料、および SNA とワークステーションに関連するトピックに関する追加情報が記載されている資料については、『参考文献』のセクションを参照してください。

第 1 章 SNA の用語と概念

この章では、CS Linux を理解し使用するために重要な、システム・ネットワーク体系 (SNA) の用語と概念を定義します。CS Linux、その機能、およびここで説明するさまざまな SNA 概念がどのように実装されているかについては、「[IBM Communications Server for Data Center Deployment on Linux 入門](#)」を参照してください。既に SNA と CS Linux をよくご存じの場合は、[27 ページの『第 2 章 CS Linux の管理』](#)から読み始めても構いません。

この章は、次の 4 つの節に分かれています。

- [1 ページの『システム・ネットワーク体系』](#)では、SNA の定義を示します。
- [1 ページの『SNA の基本概念』](#)では、どの SNA ネットワークにも適用される用語と概念を説明します。
- [11 ページの『APPN の基本概念』](#)では、拡張対等ネットワーク機能 (APPN) をサポートする SNA ネットワークのみに適用される用語と概念を説明します。
- [24 ページの『APPN ネットワークからサブエリア・ネットワークへのアクセス』](#)では、SNA と APPN を組み合わせたネットワークに適用される用語と概念について説明します。

注：この章は、SNA の概念をすべて解説しているわけではありません。SNA の詳細な情報は、『参考文献』の節に示した SNA 資料の中に記載されています。

システム・ネットワーク体系

システム・ネットワーク体系 (SNA) は、幅広いハードウェアおよびソフトウェアのデータ通信製品間で通信を行うための共通の規則を規定する IBM データ通信アーキテクチャーです。このアーキテクチャーは、ネットワーク・コンポーネントによって交換されるメッセージのレイアウトを定義する形式、およびメッセージにตอบสนองしてネットワーク・コンポーネントが実行するアクションを定義するプロトコルの 2 種類の定義から構成されます。

SNA ネットワークは、SNA を使用して相互にリンクされ、通信するコンピューターの集合です。

元々 SNA はホストコンピュータとの通信を可能にするために設計された。各ネットワークまたはサブネットワークはホストによって制御されていました。他のコンピュータはホストと直接通信しますが、相互には通信しませんでした。この古いホスト制御スタイルのネットワークは、サブエリア SNA と呼ばれることがよくあります。SNA は、ホストを必要とせずに、ネットワーク内のコンピューター間の直接対等通信をサポートするために拡張されました。この新しいピア・レベルのネットワークは、APPN です。

多くの SNA ネットワークには、サブエリアとピアツーピア・ネットワークの両方のエレメントがあります。ネットワークがサブエリア SNA から APPN に移行するときには、APPN 対応ホストは、より新しいシステムへの対等機能として動作する一方で、古いシステムを制御することができます。同様に、単一のコンピュータは (APPN ネットワーク内の) ピアコンピュータと古いホストの両方にアクセスすることができるが、ホストとの通信はホストによって制御されるが、他のコンピュータとの通信はピアツーピアであり、ホストは関係ない。

SNA の基本概念

SNA は、メインフレームから端末装置に至るまで、さまざまなデバイスが SNA ネットワーク内で相互に通信するために使用する標準、プロトコル、および機能を定義しています。

SNA の諸機能は個別の層から成る階層構造に分割され、各層が特定の機能群を実行します。このように、ネットワーク機能を複数の層に分割することによって、ネットワーク・デバイスは、情報と処理リソースを共用できるため、ネットワーク上の各デバイスについての詳細情報を保持する必要がありません。ワークステーションのユーザーは、ネットワーク上の物理デバイスやそれらのデバイス間の接続をまったく知らなくても、他のユーザーと通信できます。

ネットワークのタイプ

SNA は、次のタイプのネットワークをサポートしています。

- サブエリア・ネットワークは、サブエリア・ノードと周辺ノードから成る、階層的に編成されたネットワークです。サブエリア・ノード (ホストや通信コントローラーなど) は、汎用ネットワーク・ルーティングを取り扱います。周辺ノード (例えば端末装置) は、汎用ネットワーク・ルーティングを認識しないでネットワークに接続します。
- 対等ネットワークは、すべてが汎用ネットワーク・ルーティングに参加する対等ノードから成る協調編成のネットワークです。
- 混合ネットワークは、ホスト制御通信と対等通信の両方をサポートするネットワークです。

注: CS Linux を実行する Linux システムは、サブエリア・ネットワーク内の周辺ノード、ピア・ネットワーク内のピア・ノード、または両方の役割を同時に果たすことができます。

SNA ノード

SNA ネットワークにおいて、ノードとは、SNA プロトコルをインプリメントし、同じネットワーク内の他のノードへの通信パスを少なくとも 1 つは持った、関連するソフトウェア・コンポーネントを伴う Linux システムまたはその他のデバイスです。各ノードは、ネットワーク通信パスのノードの終端を管理し、SNA プロトコルを使用して、各パスの反対側の終端にあるノードと通信します。

サブエリア・ネットワークと対等ネットワークではノード間の関係の定義方法が異なるので、ノード・タイプを表す (ノードがネットワーク内で果たす役割を記述するための) 用語も異なります。

サブエリア・ネットワークのノード・タイプ

SNA サブエリア・ネットワークは、次のノード・タイプをサポートしています。

- サブエリア・ノードは、すべての接続ノードの通信リソースとネットワーク・リソースを制御します。SNA では、機能と他のノードに対する制御の程度によって、サブエリア・ノードを次のように分類します。
 - タイプ 5 ノードには、ネットワーク・リソースの制御、トランザクション・プログラムのサポート、ネットワーク・オペレーターのサポート、エンド・ユーザー・サービスの提供などを行う SNA 機能があります。これらの機能はホスト・プロセッサが提供することが多いので、タイプ 5 ノードをホスト・ノードともいいます。タイプ 5 サブエリア・ノードにより制御されるデバイスおよびリソースは、そのノードのドメインを構成します。
 - タイプ 4 ノードは、ネットワークの一部でデータ・フローのルーティングと制御を行うための SNA 機能を提供します。これらの機能は通信コントローラーが提供することが多いため、タイプ 4 ノードを通信コントローラー・ノードともいいます。
- 周辺ノードは、サブエリア・ネットワーク内での従属的な役割を提供します。例えば、周辺ノードは、3270 エミュレーションまたは従属 LU 6.2 通信をサポートできます。周辺ノードは、分散プロセッサ、クラスター・コントローラー、またはワークステーションなどのデバイスで、タイプ 2.0 およびタイプ 2.1 のノードとしても分類されます。
 - タイプ 2.0 ノードは、常にタイプ 4 またはタイプ 5 のノードにより制御されます。タイプ 4 または 5 のノードが関与しなければ、タイプ 2.0 ノードは他のノードとの通信を確立できません。タイプ 2.0 ノードは、従属ノードと呼ばれます。
 - タイプ 2.1 ノードは、従属ノードの役割をしますが、他のタイプ 2.1 ノードと直接通信することもできます。

注: CS Linux を実行する Linux コンピューターは、タイプ 2.1 またはタイプ 2.0 ノードとしても機能します。

周辺ノードが接続されているタイプ 4 または 5 のサブエリア・ノードは、境界ノードとして機能します。境界ノードは、サブエリア・ノードが使用するネットワーク・アドレスと周辺ノードが使用するローカル・アドレスとの間の変換をすることによって、境界機能を実行します。

単純なサブエリア・ネットワークには次のコンポーネントが含まれています。

ホスト

ホストは、オリジナルの IBM システム /370 と互換性のあるメインフレーム・コンピューターです。標準では、ホストはタイプ 5 ノードです。ただし、CS Linux for IBM Z は、ホスト・コンピューター上でタイプ 2.1 または 2.0 のノードとして動作します。

通信コントローラー

通信コントローラーは、フロントエンド・プロセッサ (FEP) ともいい、ホストに接続された独立したプロセッサです。通信コントローラーは、ホストと他のコンピューターとの通信を管理します。

通信リンク

通信リンクは、ホスト・サイトをエンド・ユーザー・サイトと接続します。ユーザーは、通常はホストから離れた別のサイトにいます。したがって、この 2 つのサイトを通信リンクで接続する必要があります。

端末コントローラー

通信リンクのリモート・エンドにあるのが端末コントローラーです。これは、クラスター・コントローラーともいいます。端末コントローラーは、リンクの使用を制御し、データを端末装置に送ります。最も代表的な IBM 端末コントローラーは 3174 と 3274 です。

端末装置

ユーザーは、端末装置でホスト・アプリケーションを実行したり、端末装置からホストへ作業を実行依頼したりします。最も知られている IBM 端末装置は 3270 です。端末装置は、端末コントローラーを経由して、または直接に、通信コントローラーに接続できます。

プリンター

IBM 3287 などのプリンターも端末コントローラーに接続できます。このようなプリンターは、ホストからの出力を受信できます。

3 ページの図 1 に示すように、サブエリア・ネットワークの図はツリーを反転した形になります。

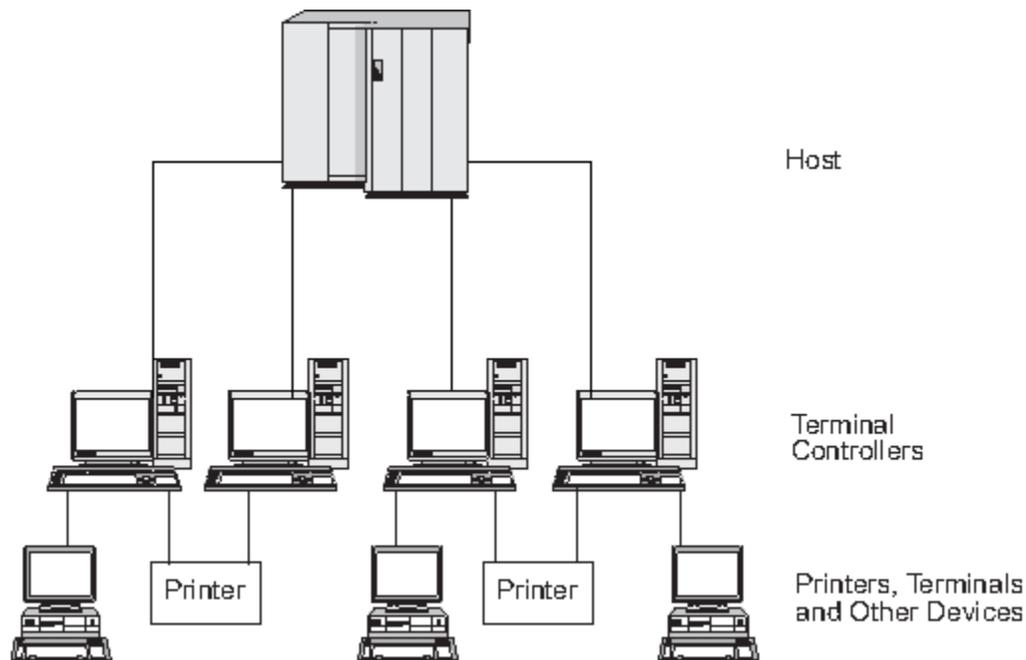


図 1. SNA サブエリア・ネットワーク

ツリーの根元に当たる部分 (図の一番上) は、ネットワークを制御するコンピューターです。枝は、このホストからネットワーク内の他のコンピューターへの通信リンク (端末コントローラー) です。葉 (図の一番下) は、ネットワーク内のコンピューターに接続されている、ユーザーがアクセスする端末装置やプリンターです。

ここに示す従来型のサブエリア SNA セットアップでは、ユーザーは 1 つのホスト・システムのリソースしか使用できません。端末装置は、端末コントローラーにデータを入力し端末コントローラーからのデータを表示する簡単な機能を備えているにすぎません。端末装置とホストとの間の SNA 通信は端末コントローラーにより処理されています。

端末コントローラーとそれに接続された端末装置は、CS Linux などのプロダクトを使用する SNA ノードに置き換えることができます。ホストの視点から見れば、このノードは端末コントローラーです。しかし、このノードは、ユーザーに、複数のホスト・システムにアクセスする機能や、画面表示をカスタマイズする機能などの追加機能を提供します。さらに、CS Linux を実行する Linux コンピューターは、SNA に無関係の他の作業にも使用できます(この点が、ホストとの通信専用で使用される 端末コントローラーとは異なります)。

対等ネットワークのノード・タイプ

対等ネットワークは、サブエリア・ネットワークの場合のようにノードを階層的に分類することはありません。他のノードとの交換は、ホストまたは他の集中プロセッサによって制御されません。どのノードも、他の任意のノードとの通信を確立できます。

対等ネットワークは、タイプ 2.1 ノードで構成されます。対等ネットワーク内のノードは、次の役割を果たすことができます。

- APPN ネットワーク・ノード (NN) は、ネットワーク・リソースの位置を識別し、ネットワーク・リソース間のセッションの経路を決定し、セッションをルーティングし、ネットワーク・ノードに直接接続されたエンド・ノード (EN) とローエントリー・ネットワーク (LEN) ノードにサービスを提供します。APPN ネットワーク・ノードのドメインは、そのノード自体と、そのノードがネットワーク・サービスを提供する相手側の任意のエンド・ノードから成ります。
- APPN エンド・ノードはリモート・リソースにアクセスでき、それらのリソースがエンド・ノード上に構成されている必要はありません。エンド・ノードは、隣接ノードとは単独で通信できますが、隣接ノード以外のノードにアクセスするには、ネットワーク・ノード・サーバーのサービスを受けることが必要です。APPN エンド・ノードのドメインに含まれるのは、そのエンド・ノード自体のみです。
- APPN ブランチ・ネットワーク・ノードにより、APPN ネットワークをブランチに分離してネットワークのトポロジを単純化し、ネットワーク管理のオーバーヘッドを低減できます。また、メイン APPN ネットワークからの分岐において、エンド・ノードとしてのネットワーク・ノード機能を提供し、メイン・ネットワーク自体がエンド・ノードとして機能します。詳しくは、[23 ページの『分岐エクステンダー』](#)を参照してください。
- ローエントリー・ネットワーク・ノード (LEN ノード) は、APPN 機能をサポートしていないタイプ 2.1 ノードです。これらのノードは、APPN ネットワークの隣接ノードと通信できますが、APPN ネットワーク内に参入してはなりません。LEN ノードでは、リモート LU を持つすべての潜在セッションを事前に定義する必要があります。この事前定義は、個別に設定するか、または特定リンクを使用してアクセスできる 1 つの隣接ネットワーク・ノード内に存在するすべてのリモート LU を指示する 1 つのデフォルト・エントリーを使用して設定します。LEN ノードのドメインに含まれるのは、そのノード自体のみです。

対等型のノード・タイプの詳細は、[11 ページの『APPN ノード・タイプ』](#)を参照してください。

接続

2 つのノードが通信するには、ノード間のデータ・フローをサポートするハードウェアとソフトウェアが各ノードに必要です。ハードウェア・コンポーネントは、各ノードのアダプターと、2 つのアダプターを接続する伝送メディアから成ります。ソフトウェア・コンポーネントは、ハードウェアとそれを介して交換されるデータを制御する機能を提供します。

ネットワークに接続されている各ノードには、1 つ以上のリンク・ステーションがあります。リンク・ステーションは、ノード内にあり、特定の隣接ノードへのデータ・フローを制御するハードウェアとソフトウェアです。2 つの隣接ノード間で通信を確立するには、リンク・ステーションのどちらか片方がまずノード間のリンクを活動化する必要があります。

トランザクション・プログラム

SNA ネットワーク経由で情報を交換するプログラムをトランザクション・プログラム (transaction program: TP) といいます。

SNA TP を組み込むことができるアプリケーション・プログラムの例を次に示します。

- エミュレーション・プログラム

- ファイル転送
- データベース・トランザクション処理
- ネットワーク管理
- 集中データ・サービス

TP は、論理装置 (logical unit: LU) を介してネットワークにアクセスし、その LU が別のノードにあるパートナー LU とのセッションを確立し維持します。論理装置の詳細は、[6 ページの『論理装置』](#)を参照してください。

注: CS Linux には、最もサポートされている API のほとんどで使えるサンプル TP が含まれています。サンプル TP の詳細は、該当の API のプログラマーズ・ガイドを参照してください。SNA TP は、他の製品の一部分としてご購入いただくこともでき、ご自分の TP を作成することもできます ([5 ページの『アプリケーション・プログラミング・インターフェース』](#)を参照)。

アプリケーション・プログラミング・インターフェース

SNA TP は、アプリケーション・プログラミング・インターフェース (API) を使用して書かれています。API が提供する特定のサブルーチンを使用して、SNA TP は、例えばデータ交換や制御機能の実行などのための、SNA 機能にアクセスすることができます。これらのサブルーチンにより、SNA TP は、リモート・ノードにある別の SNA TP と通信できます。

すべてのプラットフォーム上で、CS Linux には次の API が組み込まれています。

- APPC - LU タイプ 6.2 のみ
- CPI-C (共通プログラミング・インターフェース・コミュニケーション) - LU タイプ 6.2 のみ
- CSV (Common Service Verb) API
- LUA API

このほかに、CS Linux には、次のプロプラエタリー・プログラミング・インターフェースが組み込まれています。

- MS (管理サービス) API (AIX または Linux システムの場合のみ)
- NOF (ノード・オペレーター機能) API

ネットワーク・アクセス可能単位

TP と SNA ネットワークとの間の通信は、ネットワーク・アクセス可能単位 (NAU: 以前は「ネットワーク・アドレス可能単位」と呼ばれていました) を介して行われます。NAU は、他のネットワーク・リソースが (固有のローカル・アドレスを介して) アクセスできる固有のネットワーク・リソースです。

SNA 提供の NAU のタイプは次のとおりです。

- 物理装置 ([5 ページの『物理装置』](#)を参照)
- 論理装置 ([6 ページの『論理装置』](#)を参照)
- 制御点 ([7 ページの『制御点』](#)を参照)

注: TP は、コンポーネントではなくネットワークのユーザーとみなされるので、分類上は NAU には入りません。

物理装置

各 SNA ノードには物理装置 (physical unit: PU) が 1 つずつあります。PU は、リソース (リンク・リソースなど) を管理し、ホストとの通信をサポートします。

注: タイプ 2.1 ノード (APPN ノードの場合もある) では、制御点は、他のサービス ([7 ページの『制御点』](#)を参照) のほかに、PU サービスも提供します。2 つのタイプ 2.1 ノード (CS Linux ノードなど) は、直接通信でき、通信を確立するためにホストのサービスを必要としません。

論理装置

各 SNA ノードには、1 つ以上の論理装置 (LU) があります。LU は、TP およびエンド・ユーザーがネットワークにアクセスするために使用する一連の機能を提供します。LU は、ローカル TP およびデバイスと直接通信します。

SNA には、数タイプの LU が定義されていて、それぞれ特定クラスのアプリケーション用に最適化されています。異なるタイプの LU は互いに通信できませんが、同じタイプの LU は、異種システムに存在していても互いに通信できます。

例えば、Linux システム上で稼働する TP は、両方の TP が同じ LU タイプを使用していれば、別の Linux システム上にある TP と通信する場合と同じように簡単に、AS/400 コンピューター上の TP と通信できます。

CS Linux は、次の LU タイプをサポートしています。

LU 6.2 (APPC、5250、APPC Application Suite、および CPI-C の場合)

LU 6.2 は、分散データ処理環境でのプログラム間通信をサポートします。LU 6.2 データ・ストリームは、構造化フィールド・データ・ストリームである SNA 汎用データ・ストリーム (general data stream: GDS) か、ユーザー定義のデータ・ストリームです。LU 6.2 は、2 つのタイプ 5 ノード間の通信、タイプ 5 ノードとタイプ 2.0 または 2.1 ノードとの通信、または 2 つのタイプ 2.1 ノード間の通信に使用できます。(タイプ 2.1 ノードは APPN ノードの役割をすることができます。)

この LU タイプは、他のどの LU タイプより多数の機能と高い柔軟性を備えています。既存のハードウェアまたはソフトウェアによる制約を受けない場合は、新しいアプリケーションを開発する場合には LU 6.2 を選択するようお勧めします。

注: 独立 LU 機能を提供できるのは LU 6.2 のみです。

LU 3 (3270 印刷用)

LU 3 は、SNA 3270 データ・ストリームを使用するアプリケーション・プログラム およびプリンターをサポートします。

例えば、LU 3 は、顧客情報管理システム (CICS) で実行されており、IBM 3174 Establishment Controller に接続された IBM 3262 プリンターにデータを送信するアプリケーション・プログラムをサポートできます。

LU 2 (3270 ディスプレイ用)

LU 2 は、対話環境で SNA 3270 データ・ストリームを使用して通信するアプリケーション・プログラムとディスプレイ・ワークステーションをサポートします。タイプ 2 LU も、ファイル転送に SNA 3270 データ・ストリームを使用します。

例えば、LU 2 プロトコルは、3270 エミュレーション・プログラムをサポートできるため、ワークステーションはそれによって IBM 3270 ファミリーの端末装置の機能を実行できます。また、他のプログラムでも、通常は 3270 ディスプレイ・デバイスに出力を提供しているホスト・アプリケーションと通信するために、LU 2 を使用できます。ワークステーションは、このような TP を使用して、ホストとのある種の連携処理を実行できます。

LU 1 (SCS 印刷および RJE 用)

LU 1 は、対話式、バッチ・データ転送、または分散データ処理の環境で通信するアプリケーション・プログラム および 1 つまたは複数のデバイス・データ処理ワークステーションをサポートします。LU タイプ 1 で使用されるデータ・ストリームは、SNA 文字ストリングまたは文書コンテンツ・アーキテクチャー (DCA) に準拠します。

例えば、LU タイプ 1 は、情報管理システム/仮想ストレージ (IMS/VS) で実行されており、IBM 8100 情報システムと通信しているアプリケーション・プログラムをサポートできます。したがって、オペレーターは、アプリケーション・プログラムが保守するデータベースを訂正することができます。

LU 1 を使用するアプリケーションを、リモート・ジョブ入力 (RJE) アプリケーションともいいます。

LU 0 (LUA 用)

LU 0 は、初期の LU 定義で、プログラム間の基本的通信機能をサポートします。ある種のホスト・データベース・システム、例えば、IMS/VS (情報管理システム/仮想記憶) や一部の小売り業用および銀行用の POS システム (IBM 4680 ストア・システム・オペレーティング・システム など) では、LU 0 が使

用されます。これらの製品の現行リリースでは LU 6.2 通信もサポートされています。新規アプリケーションを作成する場合は LU 6.2 プロトコルを使用してください。

注：SNA 論理装置で使用されるデータ・ストリームについては、「*Systems Network Architecture Technical Reference*」を参照してください。

制御点

制御点 (control point: CP) は、ドメイン内のネットワーク・リソースを管理し、リソースの活動化、非活動化、状況モニターを制御する NAU です。CP は、リンクなどの物理リソースと、ネットワーク・アドレスなどの論理情報の、両方を管理します。

SNA では、次のタイプのネットワーク制御点が定義されています。

システム・サービス制御点

タイプ 5 ノードでは、CP をシステム・サービス制御点 (system services control point: SSCP) といいます。SSCP は、サブエリア・ネットワークのネットワーク・リソースを管理し制御します。例えば、SSCP は、ネットワーク・リソースのディレクトリーを使用して、制御下にある特定の LU を検出したり、ドメイン内にある 2 つの LU 間の通信を確立したりすることができます。SSCP は、他の SSCP と連携して、異なるサブエリアのドメインの LU 間の接続を確立できます。

SSCP には、ホスト・システムのネットワーク・オペレーターがネットワーク内のリソースを検査し制御するために使用できるインターフェースも組み込まれています。

物理装置制御点

タイプ 4 ノードおよびサブエリア・ノードのタイプ 2.0 ノードでは、制御点を物理装置制御点 (physical unit control point: PUCP) といいます。

制御点

タイプ 2.1 ノードでは、制御点は PU 機能と LU 機能の両方を備えています。例えば、ローカル・リンク・ステーションの活動化、ローカル・オペレーターとの対話、ローカル・リソースの管理などの機能があります。また、パートナー LU 位置、ローカル LU 用の経路の選択などのネットワーク・サービスも提供できます。

サブエリア・ネットワークでは、CS Linux ノードの CP はタイプ 2.0 PU としての役割をします。CP は、ホストの SSCP との通信を行い、サブエリア・ネットワークの他の CP との通信は行いません。

APPN ネットワークに関与したときは、CP は、隣接ノードの CP とネットワーク制御情報を交換します。CP は、タイプ 6.2 の独立 LU としての役割もします。CP は、ローカル・ノードの TP のためのデフォルト LU の役割をします。APPN 制御点の詳細は、[13 ページの『APPN 制御点』](#)を参照してください。

セッション

NAU は、セッションと呼ばれる一時論理通信チャネルを介して、他のノードの NAU と通信します。2 つの TP が通信するためには、両方の LU がセッションを確立する必要があります。ローカル・ノードでセッションを管理する LU がローカル LU です。リモート・ノードでセッションを管理する LU がパートナー LU です。

セッション・タイプ

CS Linux が関係するのは、主として次のタイプのセッションです。

LU-LU セッション

2 つの TP が通信するためには、それらの TP をサポートする LU が、まず LU-LU セッションを確立する必要があります。通常、セッションの確立が行われるのは、1 つの SNA ノードの TP が別のノードの TP と通信するときに、2 つのノードの LU 間に使用可能な既存のセッションがなかった場合です。

SSCP-LU セッション

従属 LU ([8 ページの『従属 LU と独立 LU』](#)を参照) は、まずタイプ 5 ノードの SSCP とのアクティブ SSCP-LU セッションを取得してからでないと、サブエリア・ネットワークの LU とのセッションを取得できません。SSCP-LU セッションがアクティブになると、従属 LU は LU-LU セッションを要求することができます。

SSCP-PU セッション

SSCP-LU セッションを確立するには、まず LU を制御する PU がタイプ 5 ノードの SSCP とのアクティブ SSCP-PU セッションを取得する必要があります。SSCP-PU セッションは、PU と SSCP との間で制御データおよびネットワーク管理データの受け渡しのために使用されます。

CP-CP セッション

APPN ネットワークでは、隣接ノードは CP-CP セッションを確立します。CP-CP セッションは、APPN ネットワーク内のリソースを検索するため、およびトポロジー情報を維持するために使用されます (13 ページの『APPN 制御点』を参照)

セッションの論理装置属性

論理装置には、LU-LU セッション中の対話方法を決定する属性があります。これらの属性は、SNA のアーキテクチャーにより決まります。LU は、1 次または 2 次、従属または独立のいずれかになります。

1 次 LU と 2 次 LU

セッションを確立するには、1 つの LU が、別の LU に BIND 要求を送信することによってセッションの活性化を要求します。

- 1 次 LU は、特定の LU-LU セッションのための BIND 要求を送信する LU です。
- 2 次 LU は、BIND 要求を受信する LU です。

対等ネットワークでは、固定したノード階層は使用されず、事前に決まっている 1 次 LU や 2 次 LU もありません。

注: 対等ネットワークでは、複数のセッションに関与している独立 LU (8 ページの『複数セッションと並列セッション』を参照) は、あるセッションでは 1 次 LU の役割をし、別のセッションでは 2 次 LU の役割をします。

従属 LU と独立 LU

タイプ 0、1、2、および 3 の LU は、すべて従属 LU です。タイプ 6.2 LU は、従属 LU または独立 LU のいずれかとして構成できます。

- 従属 LU (SSCP 従属 LU ともいいます) が別の LU とのセッションを確立するには、SSCP のサービスを受ける必要があります。SSCP-LU セッションを確立してからでないと、従属 LU-LU セッションは確立できません。

従属 LU は、SNA ホスト上の LU とのセッションでしか通信できません。この制限があるため、従属 LU は、一般にサブエリア・ネットワーク (ホスト介在のネットワークともいいます) を使用します。ただし、従属 LU リクエスト (DLUR) 機能を使用して、従属 LU からのセッション・トラフィックを APPN ネットワーク経由で流すことができます。DLUR の詳細は、24 ページの『APPN ネットワークからサブエリア・ネットワークへのアクセス』を参照してください。

周辺ノードの従属 LU は、常に 2 次 LU です。

- 独立 LU は、SNA ホストの支援を受けずに他の独立 LU とのセッションを確立できます。LU 6.2 が、独立 LU として使用できる唯一の LU タイプです。

独立 LU は、セッションの確立時に、1 次 LU または 2 次 LU の役割をします。

複数セッションと並列セッション

独立 LU は、複数の リモート LU とのセッションに同時に関与できます (複数セッション)。

独立 LU は、並列セッション、または同じリモート LU との複数並行セッションにも関与できます。

従属 LU (従属 LU 6.2 を含む) は、複数セッションを持つことはできません。

複数セッションの LU と並列セッションの LU を 9 ページの図 2 に示します。LUA と LUB には並列セッションがあります。LUA には、複数セッションもあります。つまり、LUB とのセッションが 2 つと、LUD とのセッションが 1 つです。LUD には、LUA と LUC との間に複数セッションがあります。

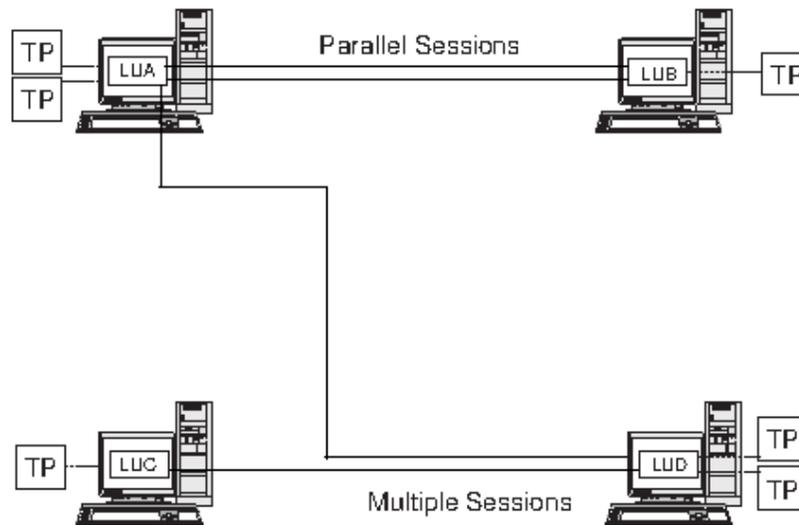


図 2. 複数セッションと並列セッション

会話

この項は LU 6.2 のみに適用されます。

2つの LU 間でセッションが確立されると、その LU-LU セッションは、2つの TP 間の情報交換をサポートします。この2つの TP ではトランザクションを実行するためにセッションを排他的に使用します。この情報交換を会話と呼びます。一時点で特定の1セッションを使用できるのは1つの会話のみですが、セッションは逐次再使用可能です(複数の会話が次々に同一セッションを使用できます)。

会話を開始するには、ソース TP がその LU に要求を送信し、リモート TP との会話を割り振るよう要求します。発呼側 TP (またはソース TP) は会話を開始する側で、電話での会話でいえば電話をかける側に当たります。呼び出し可能 TP またはターゲット TP (リモート TP) は会話のパートナーで、電話で呼び出された相手方に相当します。

10 ページの図 3 に示すように、あるノードが別のノードと通信できるようにするために、TP と LU の間で情報が交換されます。TP は直接通信しているように見えますが、どの情報交換でも、各ノードの LU が中継しています。

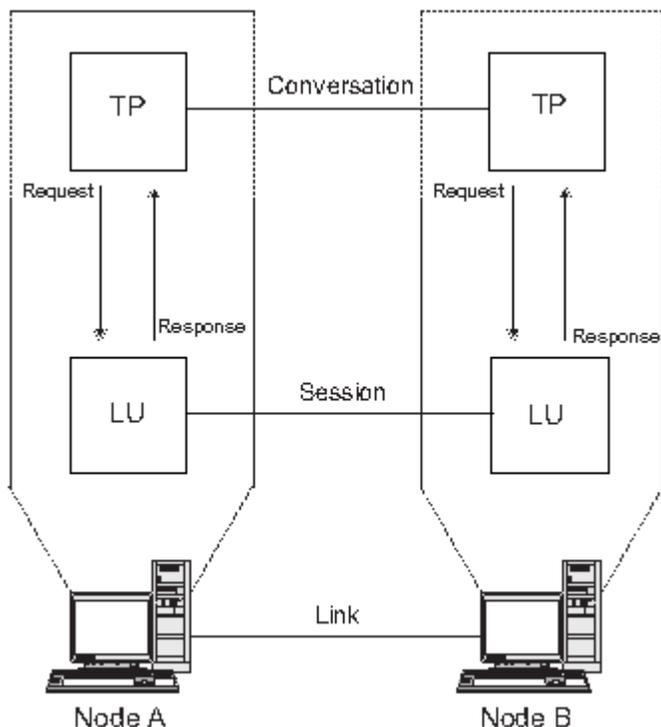


図 3. トランザクション・プログラムと論理装置との通信

SNA には、2つのタイプの会話が定義されています。基本会話とマップ式会話です。これらの2つのタイプの会話では、CS Linux と TP との間で受け渡しするために送信または受信されるデータ・パッケージの長さを示すために使用する方式が異なります。

- 基本会話では、TP はデータを SEND 関数に渡す前に、そのデータを論理レコードとして形式設定する必要があります。

論理レコードは、長さ 2 バイトのフィールドで始まる 2 バイトまたは 4 バイトのヘッダー（「LL」として表されることがあります）と、それに続く最大 32,765 バイトのデータから成っています。論理レコードは、まとめてグループ化して 1 つのブロックとして送信でき、1 つの呼び出しで複数の論理レコードを SEND 関数に伝送できます。

- マップ式会話では、情報は、1 つの不定様式データ・ブロックを指すポインターとして SEND 関数に渡されます。ブロックの長さは別のパラメーターとして渡されます。ブロックを 1 つ以上の論理レコードとして受信することはできません。必要なレコード・レベルの形式設定はすべて受信側 TP が行う必要があります。

モード

各 LU-LU セッションには、一連のセッション特性を定義する関連モードがあります。これらのセッション特性には、ペーシング・パラメーター、セッション限度 (2 つの LU 間の最大セッション数など)、メッセージ・サイズ、ルーティング・パラメーターなどがあります。

各モードは固有のモード名により識別されます。モード名は、そのノードを使用するすべての SNA ノード上で同じでなければなりません。

経路選択

LU-LU セッションを確立するには、2 つの LU が存在しているノード間の経路を計算する必要があります。経路は、リンクとノードを順序立てて並べたもので、2 つのノード間のパスを表します。

SNA ネットワークでは、次の経路選択方式がサポートされています。

- サブエリア・ネットワークの場合は、サブエリア・ノード間のすべての経路を事前に定義する必要があります。

- APPN をサポートしていない対等ネットワークの場合は、タイプ 2.1 ノードがサポートできるのは隣接ノードとのセッションだけです。中間ノードを経由してセッションを確立することはできません。
- APPN ネットワークの場合は、SNA は、セッションで使用されるモード用の指定されているサービス・クラスを使用して、セッション開始時に経路を動的に計算できます (11 ページの『サービス・クラス』を参照)。

APPN の高性能ルーティング (HPR) 機能の内容を、次に示します。

- 高速トランスポート・プロトコル (RTP) は、セッション経路で中間ノード経由でネットワーク層パケットを転送するためのサイクル数および記憶要件を最小限に抑えることができます。
- 自動ネットワーク・ルーティング (ANR) は、最初に計算された経路の一部に障害が起きた場合に、APPN ネットワークでセッションの転送が自動的にできるようにします。

サービス・クラス

サービス・クラス (class of service: COS) は、経路セキュリティ、伝送優先順位、通信量などの、ローカル・ノードが特定セッションを確立するために使用できるトランスポート・ネットワーク (データ・リンク制御およびパス制御) 特性を定義します。COS 定義では、セキュリティの許容レベル、バイト当たりのコスト、接続時間当たりのコスト、伝搬遅延、有効容量などの係数に対する相対値を割り当てます。

サブエリア・ネットワークでは、COS は、ホスト・システムに定義されている、セッションに関連したモードから導き出されます。

APPN ネットワーク・ノードでは、COS を使用して、独立 LU 間のセッション経路が計算されます。APPN ネットワークのセッションのルーティングの詳細は、17 ページの『セッション・ルーティング』を参照してください。

APPN の基本概念

拡張対等通信ネットワークング (Advanced Peer-to-Peer Networking: APPN) は、分散ネットワーク制御をサポートするネットワーク・アーキテクチャーです。これによって、ネットワークの構成と使用が容易になり、集中ネットワーク管理ができ、柔軟性に富んだ接続性をサポートできるようになります。

APPN ネットワークはタイプ 2.1 ノードで構成されます。ネットワーク内の各ノードは、APPN ネットワーク内の少なくとも 1 つの他のノードに、リンクで接続されます。これらのリンクのそれぞれを介して隣接ノードへの CP-CP セッションが確立されます (隣接ノードは、第三のノードを経由しないで直接リンクを確立できる、同一ネットワーク内のノードです)。APPN ネットワーク内のすべてのノードは、共通ネットワーク名を共有します。

APPN ノードには、Application System/400 (AS/400)、Linux を実行する Communications Server for Data Center Deployment を実行する PC、Virtual Terminal Access Method (VTAM®) を使用するシステム、CS Linux を実行する Linux サーバーなど、さまざまな規模のプロセッサを含めることができます。

APPN には次の機能があります。

- APPN のネットワーク・ノードとエンド・ノード、および APPN 以外の対等ノードのためのサポート (11 ページの『APPN ノード・タイプ』を参照)
- APPN 制御点 (APPN Control Point) 機能 (13 ページの『APPN 制御点』を参照)
- 特定の論理装置の検出をサポートするためのディレクトリー・サービス (14 ページの『リソースの検出』を参照)
- 中間セッション・ルーティング (ISR)、自動ネットワーク・ルーティング (ANR)、または接続ネットワーク (CN) を使用したセッション確立をサポートするためのトポロジー/ルーティング・サービス (17 ページの『セッション・ルーティング』、および 22 ページの『APPN 接続ネットワーク』を参照)

注: APPN ノードは、サブエリア・ネットワークに接続すると、対等ネットワーク内では APPN ノード、サブエリア・ネットワーク内では周辺ノードとして、両方の役割を果たすことができます。

APPN ノード・タイプ

APPN ネットワークの一部として使用できるのは、次のタイプのノードです。

- ネットワーク・ノード (12 ページの『APPN ネットワーク・ノード』を参照)
- エンド・ノード (13 ページの『APPN エンド・ノード』を参照)

このほかに、ローエントリー・ネットワーキング (LEN) ノードも APPN ネットワークに接続できますが、APPN 機能を使用しません (13 ページの『LEN ノード』を参照)。

サンプル APPN ネットワークは、12 ページの図 4 に示してあります。

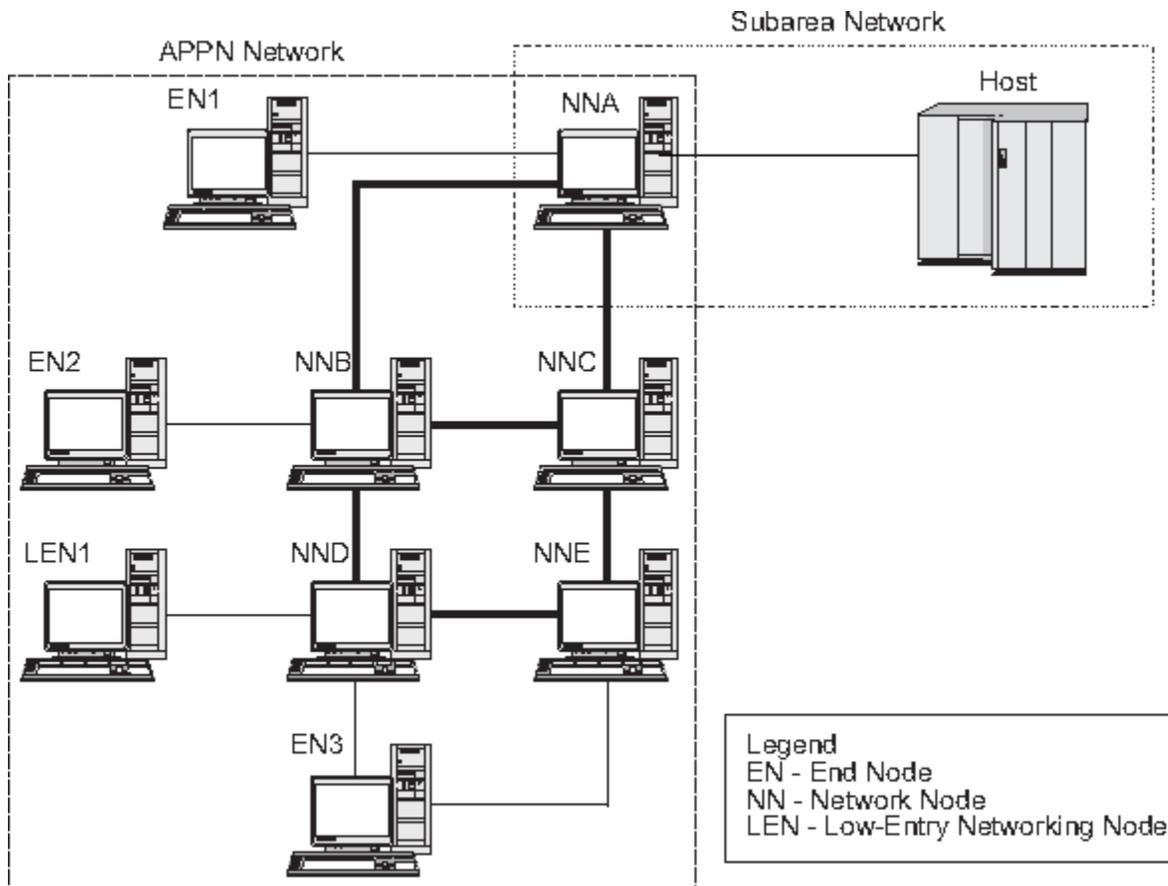


図 4. サンプル APPN ネットワークの一部

この例は、5つのネットワーク・ノード (NN)、3つのエンド・ノード (EN)、および1つの LEN ノードを含む APPN ネットワークを示しています。ネットワーク・ノードはこの APPN ネットワークのバックボーンを形成しています。エンド・ノードは、ネットワーク・ノードを介してネットワークにアクセスします。どのノードの LU 6.2 TP も、ネットワーク内の他の LU 6.2 TP と通信できます。

APPN ネットワーク・ノードの1つ (NNA) は、サブエリア・ネットワークにも関与していて、通信コントローラーを介してホストに接続されています。このノードは、APPN ネットワーク内のノードと通信するときは APPN ノードとしての役割をし、サブエリア・ネットワーク内のノードと通信するときは周辺ノードとしての役割をします。このネットワーク・ノードを介して、APPN ネットワーク内の他のノードの LU タイプ 6.2 の LU は、ホスト上の LU タイプ 6.2 の LU との LU-LU セッションを確立できます。

APPN ネットワーク・ノード

APPN ネットワーク・ノードはタイプ 2.1 のノードであり、そのドメイン内にあるすべての LU に分散ディレクトリー・サービスとルーティング・サービスを提供します。これらの LU は、ネットワーク・ノード自体にあっても、ネットワーク・ノードがサービスを提供する APPN エンド・ノードまたは LEN ノードにあっても構いません。APPN ネットワーク・ノードはエンド・ノードまたは LEN ノードのためのネットワーク・エントリー・ポイントとしての役割をするので、このネットワーク・ノードを、これらのノード用のネットワーク・ノード・サーバーとも呼びます。

ネットワーク・ノードは次のサービスを提供します。

- ローカル LU 用の LU-LU セッション・サービス
- ドメイン内のすべての LU 用のディレクトリー 検索および経路選択
- 中間セッション・ルーティング (20 ページの『[中間ルーティング](#)』を参照)
- サービス対象のエンド・ノードと MS フォーカル・ポイントとの間の、アラートなどの管理サービス (MS) データのルーティング

APPN エンド・ノード

APPN エンド・ノードは、APPN ネットワーク内のエンドポイントとしての役割をする、タイプ 2.1 ノードです。このノードは、ローカル・リソース・プライベート・ディレクトリー情報を保持しています。APPN エンド・ノードは、ローカル LU 間および隣接ノード上の LU 間のセッションを単独で確立できます。エンド・ノードに直接接続されていないノードの LU とのセッションを確立する場合は、エンド・ノードは CP-CP セッションを使用してネットワーク・ノード・サーバーからのルーティング情報とディレクトリー情報を要求します。

APPN エンド・ノードは、ローカル LU をネットワーク・ノード・サーバーに登録できます。この機能があるので、ネットワーク・ノード・サーバーで作業するネットワーク・オペレーターは、ネットワーク・ノードがサービスを提供するエンド・ノードに接続されるすべての LU の名前を事前に定義する必要はありません。

APPN エンド・ノードは、複数のネットワーク・ノードに接続できますが (12 ページの図 4 を参照)、CP-CP セッションをアクティブにできるのは、一時点で 1 つのネットワーク・ノード、つまりネットワーク・ノード・サーバーとの間にのみです。その他のネットワーク・ノードは、エンド・ノード用の中間ルーティングを提供するために使用するか、メイン・ネットワーク・ノード・サーバーが使用不可になった場合の代替ネットワーク・ノード・サーバーとして使用する場合があります。

APPN エンド・ノードには、別の APPN エンド・ノードまたは LEN ノードへの直接リンクも設定できますが、CP-CP セッションが 2 つのエンド・ノード間に確立されることはありません。

LEN ノード

ローエントリー・ネットワーキング・ノード (LEN ノード) は、独立 LU 6.2 プロトコルを使用するが CP-CP セッションはサポートしないタイプ 2.1 ノードです。LEN ノードは、APPN ネットワーク・ノードまたはエンド・ノードに接続はできますが、APPN 機能はサポートしません。

APPN ネットワーク・ノードは、接続された LEN ノードにルーティング・サービスを提供し、その LEN ノードと APPN ネットワーク内のすべてのノードとの間のリンク・ステーションを定義しなくても LEN ノードが APPN ネットワークに関与できるようにします。

APPN ネットワーク内で LEN ノードがセッションを確立する相手方の LU を、それが LEN ノードのネットワーク・ノード・サーバーに存在している場合と同じ方法で LEN ノードに定義する必要があります。LEN ノードは、ネットワーク・ノード・サーバー上の LU との間でセッションを確立します。ネットワーク・ノードは、APPN ネットワークを経由し、LU が実際に存在しているネットワーク内のノードへと、そのセッションの経路を定めます。LEN ノードの LU は、その LEN ノードにサービスを提供するネットワーク・ノードに対して事前に定義しておく必要があります。LEN ノードの LU リソースは (エンド・ノードのリソースと異なり)、ネットワーク・ノード・サーバーに登録できません。

APPN エンド・ノードは中間ルーティングを提供できません。LEN ノードが APPN エンド・ノードへの唯一のリンクである場合は、その LEN ノードは、2 つのノード間の直接リンクを介して、エンド・ノードの LU とのみ通信できます。

APPN 制御点

APPN 制御点 (APPN Control Point: CP) は、ノード・リソースを管理し、タイプ 2.1 ノードの物理装置機能と論理装置機能の両方をサポートする一連の機能です。APPN CP は、ローカル・ノード機能 (アダプターおよびリンクの活動化、非活動化など) を指示し、ディレクトリー情報およびトポロジー情報を提供し、セッションの開始時と終了時に LU を補助します。

APPN ネットワーク内の隣接ノードは、一対の並列 CP-CP セッションを使用して、ネットワーク情報を交換し、ディレクトリー・サービスと経路選択 サービスを提供します。パートナー CP が対話を開始し維持

するためには、特定ペアの両方のセッションがアクティブになっていなければなりません。次に示すように、ノード・タイプが異なれば、これらのセッションの使用法も異なります。

- APPN ネットワーク・ノードと各隣接ネットワーク・ノードの間には、2つの並列 CP-CP セッションが確立されます。これらの CP-CP セッションは、ディレクトリー、トポロジー、および管理サービス・データを交換するために使用されます。
- APPN エンド・ノードとそのエンド・ノードのサーバーとしての役割をする隣接ネットワーク・ノードの間には、2つの並列 CP-CP セッションが確立されます。これらの CP-CP セッションは、ディレクトリー、トポロジー、および管理サービス・データを交換するために使用されます。
- LEN ノードは CP-CP セッションをサポートしません。

CP-CP セッションで提供される機能は、次のように、セッションの両端のノードのタイプによって異なります。

- すべての CP-CP セッションは、ディレクトリー検索を行います。
- エンド・ノードとネットワーク・ノード間の CP-CP セッションは、次の機能を提供します。
 - リソースの登録
 - エンド・ノードとフォーカル・ポイントの間の管理サービス・データ (アラートなど) のルーティング
 - 各エンド・ノードからそのネットワーク・ノード・サーバーへのトポロジー・データのルーティング。この情報を使用して、ネットワーク・ノード・サーバーは、ネットワーク・ノード・サーバーを通らない経路を計算します。
- 隣接ネットワーク・ノード間の CP-CP セッションでは、トポロジー情報が交換されます。この交換の結果として、各ネットワーク・ノードで、内部ネットワーク・トポロジー・データベースが作成されます。

ノードを設定するときは、CP 名を定義する必要があります。CP は、ユーザー・セッションをサポートできる LU でもあります。また、ノード内に定義された唯一の LU にするような設定も可能です。

リソースの検出

TP 間の通信をサポートするために、CS Linux はまず、それらの TP を制御する論理装置間にセッションを確立します。APPN は、ノードにリモート LU についての構成情報がなくても、ノードの CP が APPN ネットワーク内の LU を検出できるようにします。ネットワーク内の LU を動的に検出する APPN 機能を、ディレクトリー・サービスと呼びます。リソースが見つかったら、APPN ネットワークを介してセッション用の経路が計算されます。

リソース名

各ノードには、ネットワーク名と制御点名の2つの部分から成る固有の名前があります。これらが合体して、CP の完全修飾名になります。この名前は、ネットワーク内の他のすべてのノードに対して、各ノードを識別します。同様に各論理装置は、ネットワーク名と LU 名で構成される、完全修飾 LU 名により識別されます。

注：ネットワーク命名規則について詳しくは、「*IBM Communications Server for Data Center Deployment on Linux 入門*」を参照してください。

ディレクトリー・サービス

各 APPN ノードには、ネットワーク・リソースのディレクトリーが保持されます。ディレクトリー・サービスは、ノード CP のコンポーネントで、ローカル・ディレクトリー・データベースを管理し、ネットワーク・ノードでは APPN ネットワーク全域にわたってネットワーク・リソースを検索します。

ノードの初期化時に、次の情報がノードに組み込まれます。

- ノード・タイプ (APPN ネットワーク・ノード、APPN エンド・ノード、または LEN ノード)
- ノードのネットワーク ID
- ノードの CP 名

各ノード・ディレクトリーには、各リソースの完全修飾名、タイプ、および登録状況が記入された、リソース (LU と CP) 用のエントリーが保持されます。各ローカル・ディレクトリーに保管される特定リソースは、ノード・タイプに応じて異なります。

- LEN ノードには、そのノード所有の LU が入ったディレクトリーが保持されます。LEN ノードの構成には、パートナーとなる可能性のある LU すべてについてのディレクトリー・エントリーも構成されている必要があります。APPN ネットワーク内で LEN ノードがセッションを確立する相手方の LU を、それが LEN ノードのネットワーク・ノード・サーバーに存在している場合と同じ方法で LEN ノードに定義する必要があります。LEN ノードは、ネットワーク・ノード・サーバー上の LU との間でセッションを確立します。ネットワーク・ノードは、APPN ネットワークを経てネットワーク内の適正なノードまで、そのセッションの経路を定めます。

LEN ノードでは、ディレクトリー・エントリーにワイルドカードを使用して、特定リンクを介してアクセスできる複数のパートナー LU を指定することもできます。

- APPN エンド・ノードには、そのノード所有の LU が含まれているディレクトリーが保持されます。このノードは、隣接ノードのパートナー LU についてのディレクトリー・エントリーも格納するように構成できます。ローカル LU は、この機能を使って、APPN 機能を使用していない LU との対等セッションを確立できます。

リソースがエンド・ノードにローカルに定義されていない場合または現在エンド・ノードからアクセスできない場合は、エンド・ノードはネットワーク・ノード・サーバーに要求を送って、APPN ネットワークでリソースを検索するよう求めます。

- APPN ネットワーク・ノードは、そのノード所有の LU と、エンド・ノードおよび LEN ノードの LU が含まれているディレクトリーを、ドメイン内に保持しています。エンド・ノードは、ネットワーク・ノード・サーバーに LU を動的に登録できます。(LEN ノードはネットワーク・ノード・サーバーに LU を登録できません。したがって、LEN ノード LU はそれぞれのネットワーク・ノード・サーバー上で構成する必要があります。) ネットワーク・ノード・ディレクトリーには、ネットワーク・ノードのドメイン内にはないが前の検索でその位置が決められている LU のキャッシュ・エントリーも入れることができます。

ネットワーク・ノードは、次の 2 とおりの方法で、他のノードにディレクトリー・サービスを提供します。

- エンド・ノードまたは LEN ノードからのセッション要求に応答して、リモート・リソースを検索する。
- 指定のリソースがローカル・ディレクトリー内で見つかる場合は、他のネットワーク・ノードからのディレクトリー検索要求に肯定応答を送る。

LEN ノード・ディレクトリー

LEN ノード・ディレクトリーの例を 15 ページの図 5 に示します。LEN ノードは CP-CP セッションをサポートしないため、ノード LEN1 用のディレクトリーには通信の相手方のすべての LU が含まれていなければなりません。ノード LEN1 用のディレクトリーは、ネットワーク・ノード・サーバー (NNA) を隣接対等エンド・ノードにないすべての LU の位置として特定します。ノード LEN1 は、これらの LU にアクセスするにはノード NNA を経由しなければならないため、エンド・ノードにある LU も含めてすべての LU の「所有側 CP」として、ネットワーク・ノード上の CP を定義します。

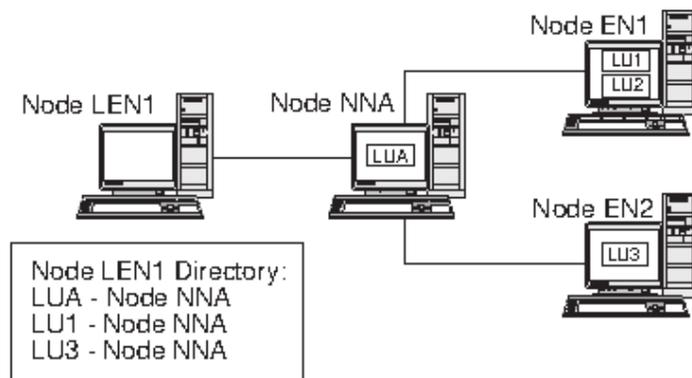


図 5. LEN ノード・ディレクトリー

直接接続されていないノードの LU とのセッションを確立するために、ノード LEN1 は、LU-LU セッション活動化 (BIND) 要求をネットワーク・ノード・サーバー (ノード NNA) に送信します。サーバーは、宛先 LU を自動的に探し出して、BIND を転送します。

注: この例では、ノード LEN1 は、ネットワーク・ノード・サーバー NNA を介して、ノード EN1 の LU1 とのセッションを確立できます。しかし、ノード EN1 の LU2 はノード LEN1 用のディレクトリーには定義されていないので、ノード LEN1 は LU2 とのセッションを確立することはできません。

エンド・ノード・ディレクトリー

エンド・ノード・ディレクトリーに含まれていない LU があつた場合、エンド・ノードは、目的の LU を見つけるために LOCATE 検索を開始します。リモート LU の検索をアクティブにするには、エンド・ノードはネットワーク・ノード・サーバーのサービスを呼び出します。エンド・ノード・ディレクトリーの例を 16 ページの図 6 に示します。

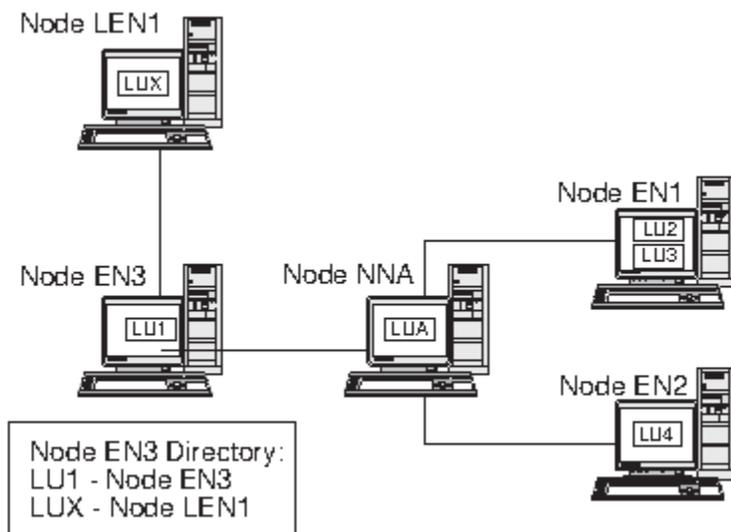


図 6. エンド・ノード・ディレクトリー

APPN ネットワーク内でパートナーとなる可能性のある LU は、エンド・ノードに定義されている必要はありません。ただし、ノード EN3 がノード LEN1 の LUX とセッションを確立するためには、LEN ノードの LU がノード EN3 のパートナー LU として構成されていなければなりません。

ネットワーク・ノード・ディレクトリー

ネットワーク・ノードは、そのサービス対象のエンド・ノードに分散ディレクトリー・サービスを提供します。

ネットワーク・ノード・ディレクトリーの例を 17 ページの図 7 に示します。

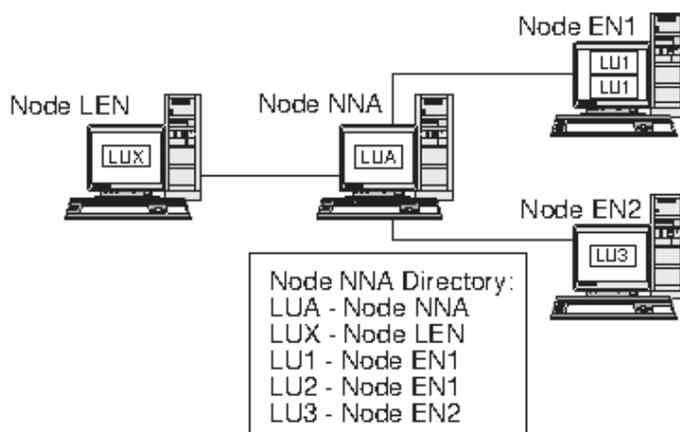


図 7. ネットワーク・ノード・ディレクトリー

ネットワーク・ノードは、次のようにしてリモート LU を探します。

1. ネットワーク・ノードが、LU を探すための要求を受信します。要求は次のいずれかです。
 - エンド・ノードまたは LEN ノードからネットワーク・ノード・サーバーへ送信された宛先 LU の名前
 - エンド・ノードからの LOCATE 検索要求に指定されている LU 名
 - LEN ノードからの BIND 要求に指定されている LU 名
 - ネットワーク・ノードの TP により指定されている LU 名
2. 宛先 LU が、ネットワーク・ノード内では見つからなかったものの、そのノードのディレクトリー内に表示される場合は、ネットワーク・ノードは、有向検索要求を宛先ネットワーク・ノード・サーバーに送信して、その LU の位置を確認します。

LU がネットワーク・ノード・ディレクトリーの中にある場合は、各隣接ネットワーク・ノードにブロードキャスト検索を送り、ネットワークの検索を開始します。

3. 各ノードはこのブロードキャストを伝搬し、成功または失敗を示す 応答を戻します。

あとで必要になる場合に備えて、ネットワーク・ノードは、成功したブロードキャスト検索から取得した情報をキャッシュに入れます。

APPN エンド・ノードは、そのエンド・ノード内の特定の LU を検索するため、または特定 LU がまだ存在していることを確認するためにネットワーク・ノード・サーバーから送られた LOCATE 検索要求を受信 (およびそれに応答) することもできます。

各 APPN エンド・ノードは、ネットワーク・ノードに登録メッセージを送信することにより、LU をネットワーク・ノード・サーバーに登録します。この方法で、ネットワーク・ノードはそのドメイン内のエンド・ノードについての現行ディレクトリー情報を保守します。LEN ノードは、ネットワーク・ノード・サーバーに LU を登録することはできません。したがって、LEN ノードのすべての LU は、構成時に、ネットワーク・ノード・サーバーに事前に登録しておくことが必要です。

セッション・ルーティング

APPN は、次の動的経路選択プロシージャをサポートしています。

- 隣接ノードとのセッションの場合は、直接セッション・ルーティング
- 1つ以上の中間ノードを通過するセッションの場合は、次のどちらかをサポートしています。
 - 中間セッション・ルーティング (ISR)。これは、セッションの実行中に変更されない経路を指定します。
 - 高性能ルーティング (HPR)。これには、高速トランスポート・プロトコル (RTP) と自動ネットワーク・ルーティング (ANR) の機能が含まれます。RTP は、セッション経路上で中間ノードを経由してネットワーク層パケットを転送するためのサイクル数および記憶要件を最小限に抑えます。また、ANR を使用すると、経路障害や輻輳 (ふくそう) を回避してセッション・トラフィックを転送できます。

動的経路選択を提供する APPN 機能は、トポロジー/ルーティング・サービス (TRS) として知られています。

トポロジー/ルーティング・サービス

各 APPN ノードには、他の APPN ノードに関する情報および伝送グループに関する情報を保管するトポロジー・データベースがあります。伝送グループとは、特定のノードのペア間のリンクのセットのことです。特定ノードについてのデータベースの内容は、ノード・タイプにより異なります。

- すべてのネットワーク・ノードは、ネットワーク・トポロジー・データベースのコピーを共有します。この共有データベースには、ネットワーク ID、CP 名、他のノードの特性など、他のすべてのネットワーク・ノードについての情報、および個々のペアのネットワーク・ノード間の伝送グループについての情報が入っています。このデータベースは、ネットワーク・バックボーン・トポロジーの完全なビュー、つまりネットワーク内のどのノード・ペアをとっても、その間でセッションをルーティングするために使用できるノードおよび伝送グループを提供します。

さらに、各ネットワーク・ノードのトポロジー・データベースには、そのネットワーク・ノードから隣接エンド・ノードまたは LEN ノードへの伝送グループに関するローカル情報も入っています。

ネットワーク・ノードは、トポロジー・データベースを使用して、ドメイン内の LU とリモート LU の間のセッションのための経路を計算したり、他のネットワーク・ノードにセッション経路を計算するための情報を提供したりします。

- 各エンド・ノードには、そのエンド・ノードから隣接ノードへの伝送グループの情報を保管するローカル・トポロジー・データベースがあります。

エンド・ノードは、LU を探しその LU へのセッション経路を計算するための要求の一部として、この情報をネットワーク・ノード・サーバーに提供します。ネットワーク・ノード・サーバーは、エンド・ノード用のセッション経路を計算するときに、そのエンド・ノードのトポロジー情報を使用します。エンド・ノードは、隣接ノードの事前定義 LU とのセッションを確立するときに、この情報を使用します。エンド・ノード・トポロジー・データベースがサポートする通信は、隣接ノードとの通信のみです。

注:

1. APPN ネットワーク・ノードおよびエンド・ノードには、接続ネットワークへのリンクに関するトポロジー情報も保持されています ([22 ページの『APPN 接続ネットワーク』](#)を参照)。
2. LEN ノードには、ローカル・トポロジー情報が保持されます。LEN ノードは、この情報をネットワーク・ノード・サーバーに転送しません。

[19 ページの図 8](#) に示すように、ネットワーク・トポロジー情報はすべてのネットワーク・ノードに複製され、ローカル・トポロジー情報はネットワーク・ノードおよびエンド・ノードに保管されます。

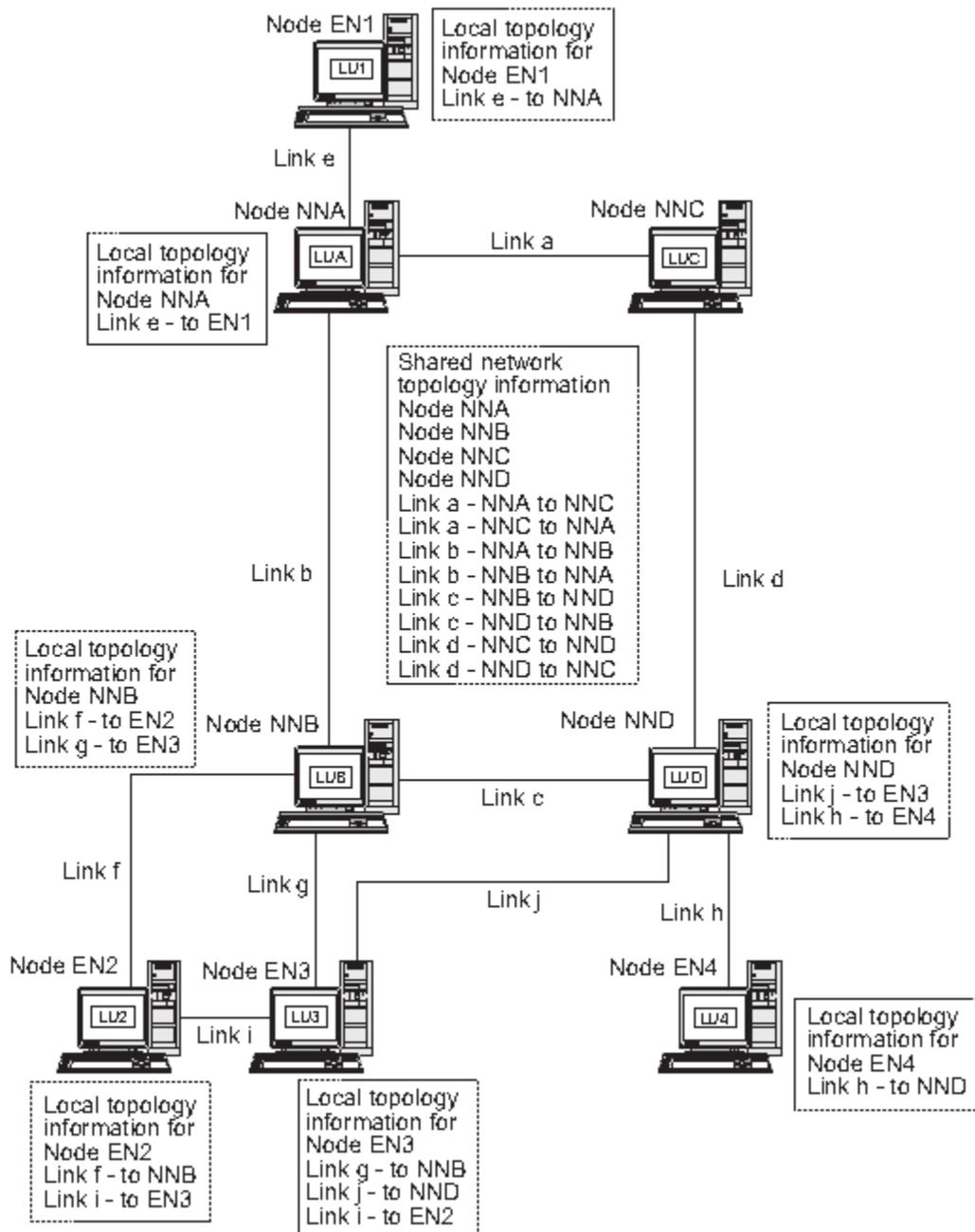


図 8. ネットワーク・ノードのネットワーク・トポロジー・データベース

共用ネットワーク・トポロジー・データベースは、ノード NNA、NNB、NNC、および NND で複製されます。また、これらのノードのそれぞれに、ローカル・トポロジー情報が含まれています。(ただし、ノード NNC を除きます。このノードにはエンド・ノードへのリンクがないので、ローカル・トポロジー情報はありません。)例えば、ノード NNB には、ノード EN2 へのリンク f および ノード EN3 へのリンク g についての情報が入っていますが、ノード EN2 と EN3 を接続するリンク i 用の情報は入っていません。

エンド・ノードに入っているのは、隣接ノードへのリンクの情報のみです。例えば、ノード EN2 にはノード NNB へのリンク f および ノード EN3 へのリンク i に関する情報が入っています。

トポロジー・データベース更新

APPN ネットワーク・ノードでは、リソース (ノードまたは 2 つのネットワーク・ノード間のリンクなど) が活性化または非活性化するとき、または既存のリソースの特性を変更するとき、CP-CP セッションを使用してネットワーク・トポロジー情報を交換します。このような変更が発生すると、ネットワーク・ノードは、トポロジー・データベース更新 (topology database update: TDU) を生成します。これには、ノー

ド識別、ノード特性とリンク特性、更新対象リソースとそのリソースの変更を識別する更新順序番号が含まれています。ネットワーク全体にわたってネットワーク・トポロジー・データベースを 現行の状態に保つために、各 TDU は、すべてのアクティブ・ネットワーク・ノード に送信されます。

APPN ネットワーク内の経路選択

APPN ディレクトリー・サービスは、特定のセッション・パートナーを探し出します。トポロジー/ルーティング・サービスは、ネットワーク内でセッション・パートナーが見つかったあとで、最適セッション経路を計算します。各ネットワーク・ノードは、それが所有する LU およびサービス対象の エンド・ノードまたは LEN ノードの LU から発信されるセッションのための 経路選択サービスを提供します。ネットワーク・ノードは、各自のローカル・トポロジー情報と、共用 ネットワーク・トポロジー・データベースからの情報を使用して、ノード間の 経路を動的に計算します。

セッション・パートナーが見つかったら、ネットワーク・ノードは次の ステップを実行して経路を選択します。

1. セッション経路の必須特性を取得する。

セッションを要求している LU が、セッション特性を識別するモード名を 指定します。そのモード名に関連したモードにより、セッション・トラフィックのルーティングに 使用するリンクに必要な条件を指定したサービス・クラスが識別されます。

2. 可能な経路についての、すべての伝送グループとネットワーク・ノード を取得する。

- エンド・ノードからのセッション要求の場合は、そのエンド・ノードが、ネットワーク・ノード・サーバーへのリンク、および接続ネットワークが存在 する場合はそのネットワークへのリンクに関する情報を提供します。
- セッション・パートナーが隣接ノードではない場合は、セッション 要求元 LU 用のネットワーク・ノード・サーバーが、ネットワーク・トポロジー・ データベースを使用して、セッション・パートナーまでの経路のネットワーク・ ノードと中間伝送グループを識別します。
- セッション・パートナーがエンド・ノードの場合は、その エンド・ノード (またはネットワーク・ノード・サーバー) が、ネットワーク・ ノード・サーバーとエンド・ノードの間のリンク (またはエンド・ノードと 接続ネットワークの間のリンク) に関する情報を提供します。

3. セッション経路の指定特性を満たさないネットワーク・ノードおよび 伝送グループをすべて除外する。

4. セッションの最適経路を計算する。

指定のサービス・クラスに応じて、経路計算アルゴリズムにより、それぞれの ノードと論理リンクごとに重み値が計算され、経路ごとに重み値が合計されます。最適パスを選択するために、ネットワーク・ノードは、発信元 LU を含む ノードから宛先 LU を含むノードまでの現在の経路のうち最小加重経路を 計算によって求めます。

中間ルーティング

APPN ネットワーク・ノードは、中間ルーティングを使用して、別のノードあての データの受信とルーティングを行います。データの発信元と宛先になるのは、 エンド・ノード、別のネットワーク・ノード、または LEN です。

中間ルーティングは、隣接ノード上にない LU 間のセッションをサポートします。セッション用の経路が選択されると、その経路内の APPN ネットワーク・ ノードは、中間ルーティングを使用して、セッション・データを経路内の次のノード に転送します。

トポロジー・データベースにより管理されるリソース特性として、輻輳(ふくそう) 状況を加えることができます。ネットワーク・ノードが重度の輻輳状態にある場合、ネットワーク・ノードはこの情報をネットワーク内の他のネットワーク・ ノードに中継して、輻輳しているネットワーク・ノードが新しいセッション用のセッション経路の計算から除外される確率を高くすることができます。

APPN には、次の 2 つのタイプの間接ルーティングがあります。

- 中間セッション・ルーティング (ISR) はすべてのネットワーク・ノードで使用可能で、このルーティングではネットワーク・ ノードが各中間セッションを追跡します。各中間ノードは、セッション・データのペーシングを調整して、隣接ノード 間でのデータ・フロー速度を制御します。各中間ノードは、データのセグメント化とセグメント化されたデータの 再組み立てもできます。ISR では、セッション経路が確

立されたら、そのセッションのすべてのデータで同じ経路が使用されます。経路の一部に障害が起これると、セッションは終了します。

- 自動ネットワーク・ルーティング (ANR) は、APPN の高性能ルーティング (HPR) 機能をサポートするネットワーク・ノードで使用可能であり、このルーティングでは、経路の一部に障害が起こった場合は中間ネットワーク・ノードが動的にセッション・トラフィックを再度、転送できます。ANR には中間セッション・ペーシングも、セグメント化および再組み立ての機能もありません。

ANR を使用すれば、中間ノードによるセッション・トラフィックのルーティングが、従来の APPN ISR の場合よりはるかに高速になります。ただし、ANR を使用すると、RTP (高速トランスポート・プロトコル) エンドポイントで余分のオーバーヘッドがかかります。中間ノードの数が少ない経路では、エンドポイントでの処理時間が増えるため、ANR 経路の方が ISR 経路より実質速度が低下する場合があります。中間ノード (ホップ) の数が多い経路では、ANR 経路の方が一般に高速になります。ANR 運用の正確な限界点は、RTP ノードの効率に応じて異なります。

直接接続

直接接続により、APPN ネットワーク・ノードでセッションのルーティングを行わなくても、セッション・トラフィックを 2 つのノード間で直接送受信できます。一般に、直接接続されたノード間のセッションでは、ネットワーク・ノード経由でデータを送付するセッションより高速でデータ交換ができます。共有アクセス転送機能 (SATF) 上のノード (例えば、21 ページの図 9 に示す LAN セグメント、または Enterprise Extender を使用する IP ネットワーク上のノード) の場合は、ネットワーク内の各ノード・ペア間のリンクを定義することによって、効率性が向上します。ただし、これは簡単な作業ではありません。リンク・ステーションの数は $n \times (n-1)$ で、 n はネットワーク内のノードの数です。

LAN セグメントまたは Enterprise Extender を使用する IP ネットワーク上の APPN ネットワークを 21 ページの図 9 に示します。

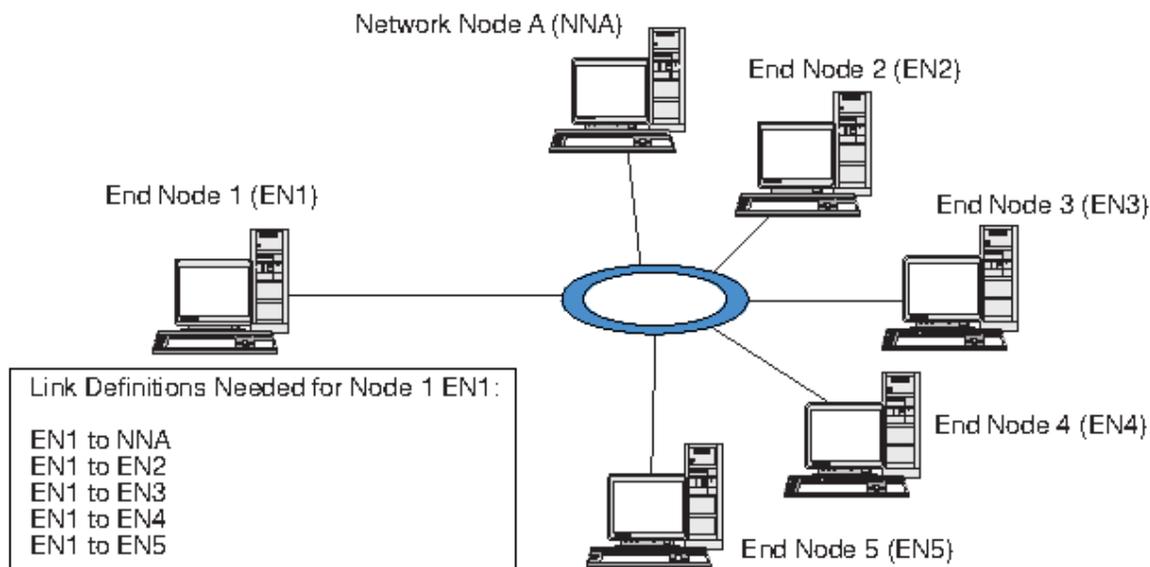


図 9. 共有アクセス転送機能を使用する APPN ネットワーク

ノード EN1 にネットワーク内の各リンク用のリンク定義がある場合は、どのノードにでも直接リンクを確立できます。ノード EN1 と APPN ネットワーク内の他の各ノードとの間の直接リンクをサポートするために必要なリンク定義を、22 ページの図 10 に示します。他に 5 つのノードがあるネットワークの場合、ノード EN1 には次の 5 つのリンク定義が必要になります。

- EN1 から NNA へ
- EN1 から EN2 へ
- EN1 から EN3 へ
- EN1 から EN4 へ
- EN1 から EN5 へ

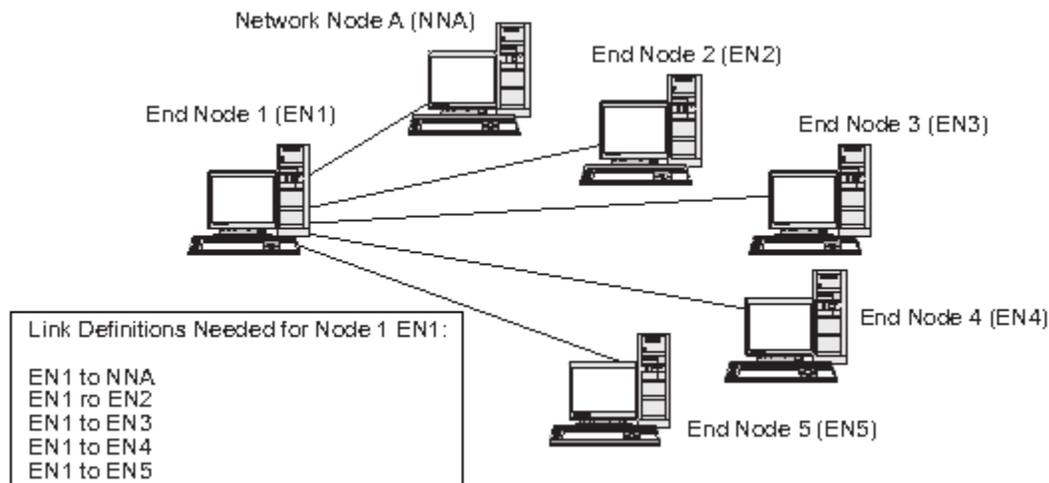


図 10. ノード EN1 から APPN ネットワーク内の各ノードへの直接リンクに必要な定義

ネットワーク内のすべてのノードが他の各ノードへの直接リンクをサポートするとすれば、この例では、6つのノードで合計 30 のリンク定義が必要です。一般に、リンク定義の数は、 $n \times (n-1)$ として計算できます。 n はネットワーク内のノード数です。ネットワークの規模がさらに大きくなると、リンク定義の数はたちまち手に負えないほど多くなります。ネットワーク・ノード間のリンク定義の数が増えると、ネットワークを通る TDU の数も増え、その結果ネットワークのパフォーマンスが低下することがあります。

APPN 接続ネットワークは、この問題に対する解決策となります。

APPN 接続ネットワーク

共有アクセス転送機能 (SATF) に接続された APPN ネットワークの場合、APPN 接続ネットワークは、ネットワーク内のノード間の直接接続をサポートするために必要なリンク定義の数を大幅に削減します。接続ネットワークでは、APPN エンド・ノードは、隣接ネットワーク・ノード・サーバーへの単一リンクと接続ネットワークへのリンクを構成するのみでよく、各ノードへの可能なリンクをすべて構成する必要はありません。

接続ネットワーク機能を使用するには、APPN ネットワークが次の条件を満たしていなければなりません。

- APPN ネットワーク内のノードは、トークンリングまたはイーサネットなどの交換メディアを使用してリンクする必要があります。
- APPN 接続ネットワーク内のすべてのリンクが、同じメディアを使用していなければなりません。
- 接続ネットワークを含んでいる APPN ネットワークは、完全接続でなければなりません。完全接続ネットワークでは、個々のノードは隣接ノードへの CP-CP セッションをサポートするリンクを 1 つ以上持っています。

接続ネットワークでは、SATF は、接続ネットワーク内の各ノードに直接接続する仮想ルーティング・ノード (VRN) として機能します。接続ネットワークの名前は、VRN 用の制御点の名前として機能します。VRN は、接続ネットワーク内にある任意の 2 つのノード間におけるセッション・データの直接ルーティングをサポートしますが、他のノードとの CP-CP セッションの確立や TDU の生成はしません。接続ネットワーク内の各ノードに必要なのは、ネットワーク・ノード・サーバーへのリンクのみです。

接続ネットワークを使用するときに必要なリンク定義を 23 ページの図 11 に示します。この接続ネットワークでは、仮想ノードを使用することによってノード EN1 と APPN ネットワーク内の他の各ノードの間の直接リンクをサポートしていて、しかも必要なリンク定義は 2 つのみです。

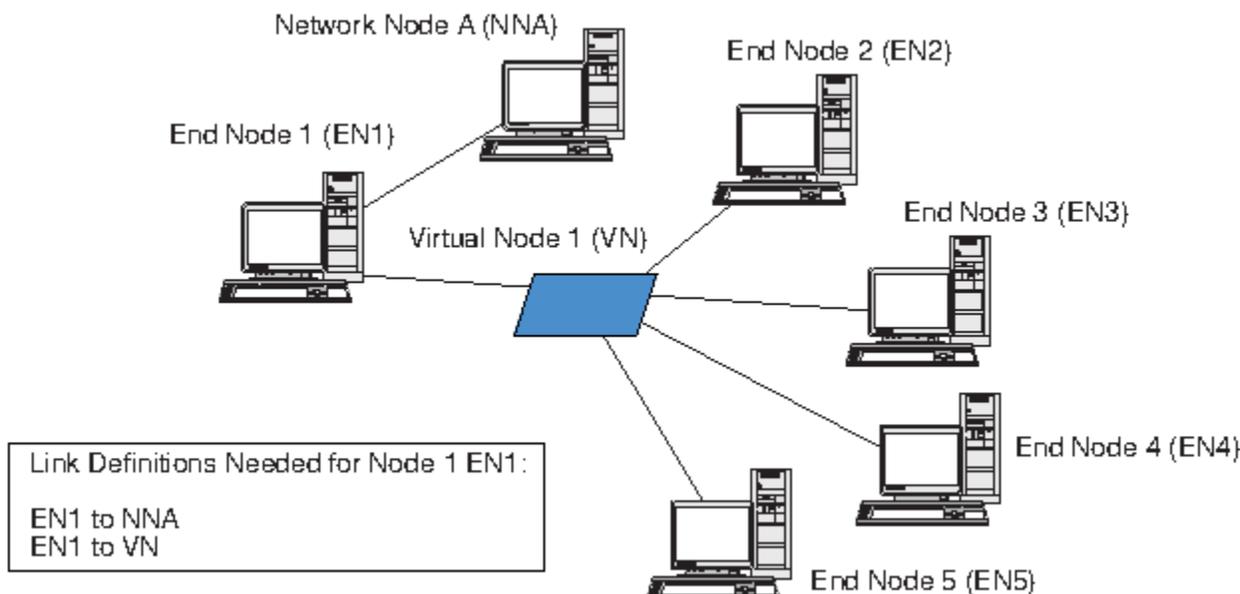


図 11. 仮想ノードを使用する直接リンクに必要な定義

APPN ネットワーク内のどの 2 つのエンド・ノードについてもその間の直接リンクをサポートするためには、合計 10 のリンク定義が必要です。(各エンド・ノードに 2 つのリンク定義が必要です。1 つのリンクはネットワーク・ノード・サーバー用、もう 1 つは仮想ノード用です。)APPN ネットワークの接続ネットワークを使用しない直接接続要件 (22 ページの図 10 を参照) に比べると、リンク定義の数ははるかに少なくなります (この例では 30 個ではなく 10 個で済みます)。ネットワークの規模がさらに大きくなると、定義に必要な条件の違いはさらに重要になります。

接続ネットワーク内の 2 つのノードの LU 間のセッションは、次のようにして確立されます。

1. 各エンド・ノードは、まずネットワーク・ノード・サーバーとの CP-CP セッションを確立します。(2 つのエンド・ノードのネットワーク・ノード・サーバーが異なる場合は、それらのネットワーク・ノードが CP-CP セッションをサポートするリンクを持っている必要があります。)
2. エンド・ノードは、エンド・ノード自体の VRN リンクとローカル・アドレス情報をネットワーク・ノード・サーバーへ報告することもできます。ローカル・アドレス情報は、サービス・アクセス・ポイント (service access point: SAP) アドレスとメディア・アクセス制御 (medium access control: MAC) アドレスにすることができます。
3. サーバーは、通常、LU-LU セッション用の最適経路として、2 つのエンド・ノード間の直接リンクを選択します。そして、1 次 LU のあるノードに、パートナー LU のあるノードへのダイナミック・リンクを確立するために必要な情報を提供します。
4. これで、エンド・ノードは、中間セッション・ルーティングを行わなくても、LU-LU セッションを確立することができます。

分岐エクステンダー

直前のセクションで説明したとおり、APPN ネットワークのネットワーク・ノードは (ネットワーク内の他のノード位置やノード間の通信リンクに関する) 情報を管理する必要があり、トポロジーが変更したときにネットワークにこの情報を転送する必要があります。ネットワークのサイズが大きくなると、保管される情報とトポロジー関連のネットワーク・トラフィックの量が多くなり、管理が困難になる場合があります。

このような問題は、ネットワークをサブネットワークに分離し、各ノードが各自のサブネットワークにあるノードのトポロジー情報のみを管理すれば、回避することができます。しかしこの結果、他のサブネットワークのリソースを検索するときに、ネットワーク・トラフィックが増加してしまいます。

24 ページの図 12 に示す APPN の分岐エクステンダー (Branch Extender) 機能には、このような問題の解決策を提供します。

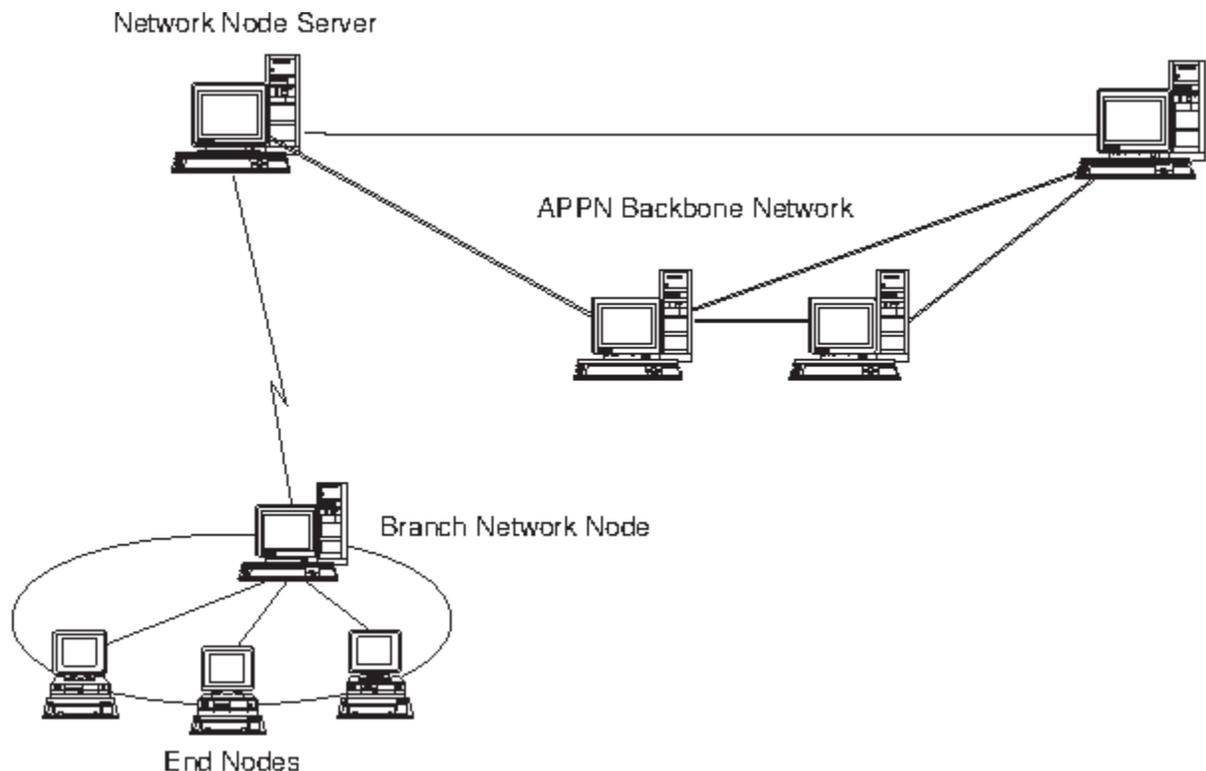


図 12. 分岐エクステンダー

名前が示すとおり、Branch Extender は、大規模な組織の個々の事業所のように、別個の領域に分割できるネットワークを対象に設計されており、メイン・バックボーン APPN ネットワークから分岐を分離させることによって機能します (例えば、組織の本社にあるネットワーク)。

各分岐には分岐ネットワーク・ノード (Branch Network Node: BrNN) という新しいタイプのノードが含まれており、メイン APPN バックボーン・ネットワークのネットワーク・ノードに接続されています。BrNN は、APPN ネットワーク・ノードと APPN エンド・ノードの機能を組み合わせたものです。

- バックボーン・ネットワークに対しては、BrNN はエンド・ノードのように機能し、バックボーン・ネットワークのネットワーク・ノード・サーバー (Network Node Server: NNS) に接続されます。
 - バックボーン・ネットワークのノードは、分岐内のノードを意識しないため、保管しなければならないトポロジー情報の量を削減できます。
 - BrNN はエンド・ノードとして機能するため、バックボーン・ネットワークからのトポロジー情報を受信しません (トポロジー情報は、ネットワーク・ノード間のみで伝送されます)。
 - BrNN は分岐の NNS とともに分岐内のすべてのリソースを登録し、あたかもそれらが BrNN 自体に存在するようにします。これにより、バックボーン・ネットワークは、分岐内の分離したノードを意識することなく、分岐内のリソースを検索することができます。
- 分岐ネットワークに対しては、BrNN はネットワーク・ノードとして機能し、分岐内のエンド・ノードに対する NNS として機能します。分岐内の各ノードは、標準の NNS の場合と同じように、残りのネットワークを分岐の NNS を通じて接続されているものと認識します。

APPN ネットワークからサブエリア・ネットワークへのアクセス

APPN ネットワークでは、ホストがネットワーク内のリソースを制御する必要はありませんが、ホストが APPN ネットワークに関与することは頻繁にあります。APPN は、これまでに多数のホスト・プラットフォームにインプリメントされていて、ホストが APPN ネットワーク内でネットワーク・ノードとして機能する一方で、従来のサブエリア SNA 機能をすべて制御するための SSCP も提供できるようにします。

多くの SNA ネットワークには、サブエリア SNA と APPN の両方のエレメントが含まれています。ネットワークのバックボーンは、ネットワーク・ノードで構築されていて、これらのネットワーク・ノードが従

属 LU とホスト上の機能の間のギャップを橋渡しする必要があります。この橋渡しを達成するためには、次の 2 つの追加サービスが必要です。

- ホストの従属 LU サーバー (Dependent LU server: DLUS) は、従来の SSCP 機能へのアクセスと APPN ネットワークへのインターフェースを提供します。
- ネットワーク・ノードまたはエンド・ノードの従属 LU リクエスター (DLUR) は、APPN ネットワークを介して従属 LU からホストへセッション・トラフィックをトランスポートするための手段を提供します。この機能により、従属 LU セッションは、APPN により提供される多様なルーティング機能を利用できます。

この DLUR と DLUS の組み合わせ(一般には単に DLUR として知られています)によって、従属 LU トラフィックを APPN バックボーンを介してトランスポートすることができます。従属 LU を使用する既存の SNA アプリケーションは、変更なしでそのまま保存でき、APPN のネットワーク管理機能、動的リソース探索機能、および経路選択機能を利用できます。このように、DLUR はサブエリア SNA から APPN への有益な移行パスを提供します。

従属 LU は、DLUR 機能を提供するノードに存在している必要はありません。DLUR 機能がネットワーク・ノードから提供される場合は、従属 LU は隣接ネットワーク・ノード、エンド・ノード、または LEN ノードにあっても構いません。DLUR 機能がエンド・ノードから提供される場合は、従属 LU はエンド・ノード自体になければなりません。

第 2 章 CS Linux の管理

CS Linux の管理の概要、および提供されているさまざまな管理ツールについては、27 ページの『CS Linux の管理の概要』を参照してください。

CS Linux の管理の最初のステップは、ノードとそのリソースを構成することです。33 ページの『CS Linux 構成の計画』に説明したように、まず構成の計画から始めてください。

CS Linux を構成するには、その前に 34 ページの『ローカル・システムの CS Linux を使用可能および使用不可にする方法』の説明に従って CS Linux ソフトウェアを使用可能にする必要があります。

CS Linux を使用可能にすれば、Motif 管理プログラムを実行できます (37 ページの『Motif 管理プログラムの使用』を参照)。Motif 管理プログラムでは、CS Linux を使用して SNA 通信をサポートするために必要な構成ができるまでの手順を説明します。Motif 管理プログラムを使用すると、指定しなければならない構成情報が最小限になり、異なる通信タイプ (3270 通信や APPC 通信など) をサポートするために必要な操作も手順ごとに説明が出るため、管理ツールとしてこのプログラムを使用することをお勧めします。

Motif 管理プログラムの代替として、WebAdmin パッケージを使用することもできます。これは、状況情報を管理、構成、および照会するための Web ブラウザー・インターフェースを提供する、別途ダウンロード可能なツールです。このツールはセキュアなアクセスを提供するため、管理者は SNA 情報への読み取り/書き込みアクセス権限を持つことも、読み取りアクセス権限のみを持つこともできます。

46 ページの『コマンド行管理プログラムの使用』で説明するコマンド行管理プログラムもあります。

本書では、管理作業ごとに、Motif 管理または コマンド行管理に使用できる情報を示してあります。その他の構成方式については、28 ページの『管理ツール』に説明があります。

CS Linux の管理の概要

CS Linux 管理者は、CS Linux ソフトウェアのインストールと、そのリソースの管理を担当します。CS Linux の管理を始める前に、CS Linux 製品の主要機能を理解しておく必要があります (『IBM Communications Server for Data Center Deployment on Linux 入門』を参照してください)。この節では、管理者が実行しなければならない管理作業と、そのために使用できるツールについて説明します。

管理責任

CS Linux システムを管理するには、次の事柄を行う必要があります。

1. 実行するユーザー・プログラムに必要な条件に従って、CS Linux システムのリソースを定義する。CS Linux の通信相手となるホストまたはピア・コンピューターの管理者と協力して作業を進めて、CS Linux の構成がリモート・システムの構成と一致するようにしてください。
2. CS Linux ソフトウェアを初期化する。
3. 条件に変更があった場合は、必要に応じて構成を動的に変更する。この変更は、リソースの追加や除去、あるいは定義されているリソースの有効/無効の切り替えによって行う。
4. アクティブ・リソースの状況を監視し、発生する問題を診断するための診断情報を収集する。
5. オプション操作として、標準管理操作を自動化するためのアプリケーション・プログラムまたはシェル・スクリプトを作成する。

これらの作業は、通常、CS Linux システムをインストールするサイトで システム管理者が行います。しかし、CS Linux には サービス・ポイント・コマンド機能 (SPCF) もあるので、NetView プログラムを使用するオペレーターは、NetView コンソールから管理コマンドを発行することによって、ステップ 2 と 3 をリモート側で実行できます。SPCF の詳細は、103 ページの『第 9 章 NetView からの CS Linux の管理』を参照してください。

管理ツール

CS Linux には、システムを管理するためのさまざまなツールがあります。どのツールを使用するかは管理上必要な条件によって決まり、必ずしもすべてのツールを使用する必要はありません。ここでは、各ツールの機能を簡単に説明します。

注：

1. 本書では、この項で説明するツールのいずれかを使用して実行できる CS Linux 管理についての一般的な情報を示します。ほとんどの目的には、Motif 管理プログラムの使用をお勧めします。Motif 管理プログラムでは、ノードの構成と管理に対しコンテキストに依存したガイダンスが提供されるためです。
2. CS Linux 管理ツールを使用できるユーザーの制御、および使用できる管理機能の範囲については、[33 ページの『管理許可』](#)を参照してください。

CS Linux には、次の管理ツールが組み込まれています。

- Motif 管理プログラム ([28 ページの『Motif 管理プログラム』](#)を参照)
- Web 管理プログラム ([29 ページの『Web 管理パッケージ』](#)を参照)
- コマンド行管理プログラム ([30 ページの『コマンド行管理プログラム』](#)、または「*IBM Communications Server for Data Center Deployment on Linux* 管理コマンド解説書」を参照してください)。
- サービス・ポイント・コマンド機能 ([30 ページの『リモート・コマンド機能』](#)を参照)
- 構成ファイル ([30 ページの『構成ファイル』](#)を参照)
- 診断ツール ([32 ページの『診断ツール』](#)を参照)

CS Linux 管理ツールはすべて NOF API を使用します。この API を使用して、独自の管理ツールを作成することもできます。詳しくは、[32 ページの『NOF アプリケーション』](#)を参照してください。

Motif 管理プログラム

CS Linux 構成を定義し、変更する最も簡単な方法は、Motif 管理プログラム (xsnaadmin)を使用することです。このプログラムは、CS Linux リソースの表示と管理を行うためのグラフィカル・ユーザー・インターフェースを提供します。

注：Motif 管理プログラム xsnaadmin は、CS Linux 操作をリアルタイムでモニターするために使用することができますが、パフォーマンスに多少の影響があります。Linux for IBM Z LPAR には、直接接続された端末がないため、xsnaadmin は X-Windows プロトコルを使用して動的に表示を更新します。これにより、ネットワーク・トラフィックと CPU 使用率がシステムに追加されます。より大規模なネットワーク構成の場合は、コマンド行管理ツールまたは Web 管理パッケージを使用する方が適しています。これらの更新は、照会を介して行われ、影響を与える動的トラフィックは作成されません。

直接端末ハードウェアをサポートするシステムの場合、Motif 管理プログラム xsnaadmin はパフォーマンスにあまり影響を与えません。ただし、システム・パフォーマンスへの影響は、表示されている内容、および表示が更新される頻度によって決まります。

以下の管理操作を使用できます。

- CS Linux リソースの定義
- ノードとその接続リソースの開始と停止
- 定義済みリソースの構成の変更
- 定義済みリソースの構成およびそれらがアクティブである場合の現在の状況の照会
- リソースの削除

Motif 管理プログラムを使用して、両方のノード・リソース (LAN 上の任意のサーバーの場合は、CS Linux ソフトウェアがそのサーバー上で実行されている場合)とドメイン・リソースの両方を管理することができます。通信のタイプ (3270 または APPC など)ごとに、プログラムは必要なリソースの構成をセットアップするためのガイドとなります。

注：Motif 管理プログラムのウィンドウおよびダイアログは、このガイドに表示されるダイアログとは異なる場合があります。これは、特定のダイアログ上での選択によって異なります。

Motif 管理プログラムには、SNA および CS Linux の概要情報、CS Linux ダイアログの参照情報、および特定のタスクを実行するためのガイダンスを提供するヘルプ画面が含まれています。

Motif 管理プログラムを開始する前に、CS Linux ソフトウェアが使用可能になっていることを確認してください (詳細については、[27 ページの『第 2 章 CS Linux の管理』](#)を参照してください)。任意の X/Motif アプリケーションと同様に、適切な X サーバーを示すために、ディスプレイ 環境変数をセットアップする必要があります。

Motif 管理プログラムをバックグラウンドで開始するには、次のコマンドを発行します。

```
xsnaadmin &
```

開始されたすべての CS Linux サーバーがメイン画面に表示されます。既に構成されているものについては、プログラムを使用してノードを選択し、選択したノードの構成を表示することができます。それ以外の場合、プログラムは、ノードを選択するように求めるプロンプトを出し、それを定義するために必要なステップを実行します。

Motif 管理プログラムを使用して CS Linux リソースを定義および管理する方法については、[37 ページの『Motif 管理プログラムの呼び出し』](#)を参照するか、プログラムによって提供されるヘルプ画面を参照してください。

注: Motif 管理プログラムを使用すると、標準の CS Linux 構成に必要なすべてのパラメーターをセットアップできます。拡張パラメーターの場合、Motif 管理プログラムはデフォルト値を提供します。必要な構成情報のみを提供する必要があります。これにより、SNA 通信を迅速かつ容易にセットアップすることができます。

その他の CS Linux 管理ツール(コマンド行構成、および NOF アプリケーション・プログラムを含む)は、Motif 管理プログラムに表示されているものよりも広い範囲の構成パラメーターとオプションにアクセスできます。ただし、ほとんどの場合、Motif 管理プログラムから必要なすべての構成を実行できます。これは、ほとんどのユーザーが変更を必要としないフィールドを構成し、非表示にするために必要なキー・フィールドを公開するためです。コマンド行構成によって提供されるデフォルト値は、Motif 管理プログラムによって提供される値とは異なる場合があります。Motif プログラムは、実行中の構成タスクのコンテキストに基づいて、よりインテリジェントに値を選択することができるためです。

これらの追加機能を使用する必要がある場合でも、Motif 管理プログラムを使用して基本構成をセットアップし、その他の管理ツールを使用して追加機能を指定することができます。後で変更後の構成を管理するために Motif 管理プログラムを使用すると、そのプログラムは、他のツールを使用して行った変更を保持しますが、構成済みの追加機能は Motif プログラムには表示されません。

Web 管理パッケージ

Web 管理パッケージは、CS Linux のサポート Web ページ <https://www.ibm.com/support/pages/web-administration-package> からダウンロード可能な管理ツールです。このパッケージは、リモートの Web ブラウザーから CS Linux の管理機能へのアクセスを可能にするスクリプトおよび命令を提供します。

Apache または IBM HTTP Server のような Web サーバーを使用することにより、このパッケージは、通常、Motif ベースの xsnaadmin の GUI にある照会、状況、開始および停止などの管理機能へのリモート・アクセスを可能にします。このパッケージは、CS Linux のリソースのリモート管理のためのセキュアで安全な方法を管理者に提供するように設計されています。このパッケージは、CS Linux がサポートする Intel、Power、および IBM Z の各プラットフォームでも実行できます。

このパッケージは、Perl-CGI および Perl スクリプトを使用して管理機能を実行します。ほとんどすべての Web ブラウザーで、このパッケージは問題なく動作します。リソースの照会および管理をするための実行コマンドは、snaadmin コマンド行ツールにあります ([30 ページの『コマンド行管理プログラム』](#)を参照してください)。

Web 管理パッケージにより、ノードおよびドメイン、接続、ホスト・リソース、TN3270E サーバー、APPC、APPN、照会、状況、および診断パネルで制御されるリソースにアクセスできます。これらの Web インターフェースにより、ノード上のすべてのリソースを照会したり、またはセッション、リンク、LU、およびノードの状況を把握したりすることができます。

Web ページは、リンク、PU、およびノード状態を表示するためのインターフェースを提供します。必要に応じて、ノード、リンク、PU を開始または停止できます。

ホスト接続リソースは、コマンド行管理の呼び出しと同じように表示でき、さらに管理用の簡潔な表示もできます。リンク、LU プール、PU、および TN3270E サーバー・セッションの詳細な表示も可能です。

リモート・アクセスは、Linux 提供のパスワード保護パッケージを使用した、ユーザー ID およびパスワードの認証により保護されます。WebAdmin プログラムは、アクセス用に 2 つの URL サイトを提供します。1 つは、トレースの構成、開始/停止、および収集と、情報の照会を可能にする読み取り/書き込みサイトです。もう 1 つは、状況を表示するための読み取り専用アクセスです。

コマンド行管理プログラム

コマンド行管理プログラム `snaadmin` を使用すると、個々の CS Linux リソースを管理するためのコマンドを出すことができます。`snaadmin` は、Linux コマンド・プロンプトから直接使用することも、シェル・スクリプト内から直接使用することもできます。

コマンドは、特定の CS Linux ノードに対して発行して、ノードのリソースを管理し、SNA ネットワーク・データ・ファイルに対して、コントローラー・サーバーとバックアップ・サーバーを管理するか、ドメイン・リソースを管理するためのドメイン構成ファイルを管理することができます。

すべての管理コマンドは、サーバー上で発行できます。ただし、IBM Remote API Client 上で発行できるコマンドには制限があります。

- On Windows clients there is no `snaadmin` プログラムでは、コマンドを発行できません。
- On AIX and Linux clients you can issue any 照会 or 状況 command. IBM Communications Server for Data Center Deployment on Linux 管理コマンド・リファレンスで定義されているその他のいくつかの管理コマンドは、IBM Remote API Client から発行できることを明示的に説明しています。それ以外の場合、これらのコマンドはサーバーからのみ使用可能です。

コマンド行管理のヘルプは、以下のコマンドのいずれかを使用して取得できます。

- `snaadmin -h` は、コマンド行の管理およびコマンド・ライン・ヘルプの使用法に関する情報について
- `snaadmin -h -d` は、`snaadmin` プログラムに提供されるコマンドのリストを提供します。
- `snaadmin -h` 指揮 は、名前付き 指揮のヘルプを提供します
- `snaadmin -h -d` 指揮 provides detailed help for the named 指揮, including a list of the configuration parameters that can be specified with the command.

詳しくは、*IBM Communications Server for Data Center デプロイメント on Linux* 管理コマンド解説書を参照してください。

リモート・コマンド機能

リモート・コマンド機能 (RCF) は、ホスト上の NetView コンソールからの CS Linux の管理をサポートする以下の機能を提供します。

- サービス・ポイント・コマンド機能 (SPCF) を使用すると、ホスト NetView コンソールのオペレーターは、CS Linux 管理コマンドを発行することにより、NetView から CS Linux を管理できます。
- UNIX コマンド機能 (UCF) を使用すると、NetView オペレーターは、CS Linux コンピューター上で標準の Linux コマンドを発行できます。

RCF の詳細は、[103 ページの『第 9 章 NetView からの CS Linux の管理』](#)を参照してください。

構成ファイル

CS Linux システムの構成情報は、以下のテキスト・ファイルに保持されています。

ノード構成ファイル

`/etc/opt/ibm/sna/sna_node.cfg` ファイルには、特定ノードの CS Linux ノード・リソースに関する情報が含まれています。このファイルは、ノードが実行されているコンピューター上にあります。

これには、ノードのリソースに関する情報が含まれ、CS Linux がノード上で開始されたときにアクティブなリソースを指定します。

このファイルは、使用可能なリソースの初期定義を提供します。その後、他の管理ツールを使用して、実行中のノードのリソースを要件の変更として変更することができます。変更内容は自動的にファイルに保存されるため、ノードの停止時および再始動時に、変更後の構成を再び使用できます。

ドメイン構成ファイル

/etc/opt/ibm/sna/sna_domn.cfg ファイルには、CS Linux ドメイン・リソース (特定のローカル・ノードに関連付けられていないリソース) に関する情報が含まれています。このファイルの制御コピーは、コントローラー・サーバー上にあります。

呼び出し可能 TP データ・ファイル

/etc/opt/ibm/sna/sna_tps ファイルには、CS Linux が呼び出し可能 (ターゲット) TP を開始する必要がある情報が含まれており、また、他の情報 (TP へのアクセスに必要なセキュリティのレベルなど) も含まれています。このファイルは、TP が実行されるコンピューター上にあります。

このファイルについて詳しくは、[75 ページの『TP の定義』](#)を参照してください。

環境パラメーター・ファイル

/etc/opt/ibm/sna/environment ファイルには、その動作を変更するときに CS Linux に提供される環境変数が含まれています。These are detailed in an appendix to the *IBM Communications Server for Data Center* デプロイメント on Linux 管理コマンド解説書。

ログ・フィルター構成ファイル

/etc/opt/ibm/sna/logfilter.txt ファイルは、CS Linux によってキャプチャーされないようにするログ・メッセージを指定します。これは、*IBM Communications Server for Data Center* デプロイメント on Linux 診断ガイドのフィルタリング・ロギング・セクションに記載されています。

構成を変更するには、Motif 管理プログラム、コマンド行管理プログラム、または NOF API を使用します。これらのツールはすべて、必要に応じて、ノード構成ファイルまたはドメイン構成ファイルに対して必要な変更を行います。構成情報はプレーン・テキストとして保管されるため、vi などの標準 ASCII テキスト・エディターを使用してファイルを直接変更することも、awk や sed などの Linux ユーティリティを使用してシェル・スクリプトを使用してファイルを変更することもできます。Any changes to configuration files using a text editor must be made **前** starting CS Linux. CS Linux 構成ファイル・フォーマットについての詳細は、*IBM Communications Server for Data Center* デプロイメント on Linux 管理コマンド解説書を参照してください。

注: CS Linux 構成は、動的プロセスです。CS Linux ソフトウェアを開始する前に、構成全体を定義する必要はありません。構成ファイルは、使用可能なリソースの初期定義を提供しますが、CS Linux ソフトウェアの実行中に、必要に応じてリソースを追加、削除、または変更することができます。CS Linux は、現在の定義を保管して、システムを再始動する必要があるときに再び使用できるようにします。

以下のファイルには、CS Linux クライアント / サーバー・ネットワークに関する情報が含まれます。

SNA ネットワーク・データ・ファイル

/etc/opt/ibm/sna/sna.net ファイルには、どのサーバーがコントローラーであるか、およびどのサーバーがバックアップ・サーバーとして機能できるかに関する情報が含まれています。このバイナリー・ファイルはコントローラー・サーバー上にあります。管理プログラムまたは NOF API を使用して、このファイルの内容を変更することができます。

このファイルとその変更方法について詳しくは、[47 ページの『クライアント/サーバー機能の構成』](#)を参照してください。

クライアントネットワークデータファイル

sna_clnt.net ファイルには、IBM Remote API Client で必要とされる CS Linux サーバーへのアクセス方法に関する情報が含まれています。このテキスト・ファイルはクライアント・コンピューター上にあります。このファイルの内容は、標準の ASCII テキスト・エディターを使用して変更できます。

このファイルとその変更方法について詳しくは、[133 ページの『クライアント・ネットワーク・データ・ファイル \(sna_clnt.net\)』](#)を参照してください。Windows クライアントで同等の情報を構成する方法については、[111 ページの『第 10 章 CS Linux クライアント / サーバー・システムの管理』](#)を参照してください。

NOF アプリケーション

CS Linux NOF API は、コマンド行管理プログラムと同じ管理機能を提供します。これらの機能を使用して、CS Linux のリソースを定義し、管理することができます。これは、CS Linux を管理するためのアプリケーション・プログラムを各自が作成できるということです。

詳しくは、「*IBM Communications Server for Data Center Deployment on Linux NOF プログラマーズ・ガイド*」を参照してください。

診断ツール

CS Linux には、CS Linux の操作中に直面する問題を診断し訂正する際に役立つツールがあります。

- 問題または例外 (問題の原因を示している可能性のある異常条件) が検出されたコンポーネントがあれば、エラー・ログ・ファイルにエントリーが書き込まれます。また、有効なシステム・イベントはすべて監査ログ・ファイルに記録できます。どのタイプのイベント (問題、例外、または監査) を記録するかは、管理者が決めます。クライアント/サーバーのネットワーク構成では、すべてのサーバー上で記録するイベントのタイプに関するグローバル設定値を指定して、必要に応じて個々のサーバー上でこれらの設定値を指定変更できます。
- CS Linux は、使用量ログ・ファイルも維持します。このファイルは、CS Linux リソースの現在の使用量とピーク使用量に関する情報を記録するために使用されます。
- ログ情報の各タイプを保持するのに使用するファイルの名前とディレクトリーを指定することができます。また、必要であればエラーおよび監査の両方のタイプのログ情報を同一ファイルに送ることができます。クライアント/サーバー・システムでは、すべてのサーバーから 1 台のサーバー上の中央ログ・ファイルにメッセージを送ることも (中央ロギング)、各サーバー上にある別々のファイルにログ・メッセージを送ることもできます。
- ログ・ファイルはテキスト・ファイルとして生成され、vi などの標準の ASCII テキスト・エディターを使用して表示できます。
- 詳細ログ (各メッセージのログ・ファイルに、ログの原因の詳細と必要なアクションが記載される)、または要約ログ (ログのソースとメッセージ・テキストの要約のみ記載される) のいずれかを選択できます。要約ログを使用する場合は、詳細な情報が必要になったら、snahelp コマンド行ユーティリティーを使用して特定のメッセージ番号についての詳細な原因と処置を取得できます。
- 特定のイベントが頻繁に発生するためログ・ファイルが同じログ・メッセージの多数のインスタンスでいっぱいになる場合は、1 つ以上の特定のログ・メッセージは 1 回のみログに記録するように、フィルターを設定できます。同じログ・メッセージの 2 回目以降のインスタンスは無視され、ログ・ファイルに書き込まれないようになります。
- 一部のエラー条件については、CS Linux は、エラー・ログ・ファイルに問題メッセージを書き込むほかに、Linux コンソールにメッセージを送ってオペレーターに警告します。
- 多くのコンポーネントは、そのコンポーネントのアクティビティーを記録するトレース・ファイルを作成することができます。トレースを実行すると、CS Linux コンポーネントのプラットフォームが低下するため、通常は使用不可になっています。
- コマンド行ユーティリティーを使用すると、トレース・ファイルをフィルターで選別して特定の情報を抽出し、さらにトレース情報を形式化してその内容を解釈したり、メッセージ・フローの要約を作成することができます。形式設定済みの出力ファイルは、vi などの標準 ASCII テキスト・エディターを使用して表示できます。
- CS Linux では、警告を生成してホスト・コンピューターの NetView プログラムに送信できます。このアラートは、次のいずれかです。
 - 接続コンポーネントからのリンク・アラート。これは、接続上の問題についての情報を提供します。
 - MS API を使用するアプリケーション・プログラムによって提供されるアラート

CS Linux ログ・メッセージ、CS Linux トレース機能の使用、およびトレース・ファイルの解釈については、「*IBM Communications Server for Data Center Deployment on Linux 診断ガイド*」を参照してください。

MS API の使用については、「*IBM Communications Server for Data Center Deployment on AIX or Linux MS プログラマーズ・ガイド*」を参照してください。

管理許可

CS Linux 管理ツールは、SNA リソースを管理する許可のある、制限されたグループの『SNA 管理者』によって使用されることを目的としています。このために、グループ所有権 `sna` を持つシステム管理者ログイン `root` が実行可能ファイルを所有します。グループ `sna` のメンバーであるユーザーのみが、CS Linux リソースの変更、開始、または停止を実行することができます。SNA 管理者許可を必要とするユーザーは、このグループのメンバーでなければなりません。

標準の CS Linux インストール済み環境では、グループ `sna` のメンバーでないユーザーは CS Linux 管理ツールをまったく実行できません。必要に応じて、これらのユーザーに読み取り専用モードでのツールの実行を許可できます。これにより、ユーザーは構成情報と状況情報を表示できますが、リソースを変更、開始、または停止することはできません。これを行うには、`chmod` を使用し、該当する実行可能ファイルに対するユーザーの読み取り許可および実行許可を付与します。

管理ツール	実行可能ファイル
Motif 管理プログラム	<code>/opt/ibm/sna/bin/X11/xsnaadmin</code>
コマンド行管理プログラム	<code>/opt/ibm/sna/bin/snaadmin</code>

これにより、任意のユーザーが適切な管理ツールを実行して情報を表示できるようになりますが、`sna` グループに含まれないユーザーがリソースを変更、開始、または停止することは、CS Linux によってこれまでどおり禁止されます。

注：上記で説明したように、ファイルへのアクセス権を変更する場合は、CS Linux PTF または新しいリリースをインストールした後で、この手順をもう一度行うことが必要になります。

CS Linux 構成の計画

構成変更をする前に十分な計画を立てておくことが非常に重要です。変更を行うと、ローカル・ノードのユーザーのみでなく、ネットワーク全体のユーザーにとっても破壊や混乱を招く原因になることがあります。

ネットワークのトポロジーに加えようとしている変更のダイアグラムを作成すると役立ちます。他のノードの接続を追加または除去する場合は、自ノードとその他のノードを示す図を描きます。Motif 管理プログラムを使用すれば、既存のすべての接続に関する構成情報を収集し、その情報をダイアグラムに追加できます。

新規リソースをダイアグラムに追加した場合、既存のリソースと重複しているかどうか、名前の競合があるかどうかを調べるのも簡単です。同様に、どのリソースを除去する必要があるかを決め、重要なリソースを削除しないようにするためにも、ダイアグラムを利用できます。

クライアント/サーバー CS Linux システムを複数のノードで構成している場合は、すべての CS Linux ノードとその接続リソースをダイアグラムに含めてください。

必要な変更を決定したら、そのために必要な構成情報の収集を開始することができます。Motif 管理プログラムのオンライン・ヘルプ・ファイルに含まれている作業シート、または [33 ページの『計画ワークシート』](#) で説明する計画ワークシートを使用すれば、その手順に従って、特定の CS Linux 機能についての構成情報を収集できます。

計画ワークシート

CS Linux 用のリソースの構成を始める前に、新規リソースのための構成データをすべて集めてください。サポートする必要がある特定の機能またはアプリケーションのためのすべての情報を記録するには、[139 ページの『付録 A 構成計画ワークシート』](#) に示す計画ワークシートを使用します。

多くの場合、ネットワーク管理者、ホスト管理者、アプリケーション・プログラマー、エンド・ユーザーなど、いくつかのソースから構成情報を集めることが必要になります。

別のノードに接続する場合は、そのノードの管理者に連絡することが重要になります。ノードの管理者から、そのノードのすべてのリソースの名前、アドレス、および特性についての情報を取得できます。多く

の場合、ローカル・ノードとリモート・ノードに入力されている構成パラメーターが一致しているかどうかを確認する必要があります。

作業シート

Motif 管理プログラムのオンライン・ヘルプ画面には、作業シートがあり、特定の構成作業のための手順説明が示されています。作業シートには、構成情報を入力するために使用するダイアログのすべてのヘルプ画面へのポインターが組み込まれています。これらのポインターを使用してヘルプを表示し、どのデータを収集しなければならないかを正確に調べることができます。

作業シートからさらに、構成情報の入力に使用しなければならない個々のウィンドウおよびダイアログについての詳細なヘルプを参照できます。これらのヘルプ画面には、入力または選択が必要な各フィールドの説明があります。

ローカル・システムの CS Linux を使用可能および使用不可にする方法

このセクションでは、Linux サーバー上で CS Linux ソフトウェアを使用可能および使用不可にする方法を説明します。

CS Linux ツール (Motif 管理プログラムを含む) を使用するには、その前に CS Linux ソフトウェアを使用可能にする必要があります。通常は、ソフトウェアは CS Linux のインストール後に自動的に使用可能になりますが、必要な場合には手動で使用可能にすることができます。

CS Linux による使用のための環境変数の設定

CS Linux は、サーバー上で CS Linux ソフトウェアを使用可能にする前に設定する必要がある、操作を制御するために多くの環境変数を使用します。これらを管理する最も簡単な方法は、CS Linux が始動時に読み込むテキスト・ファイル内に設定することです。

CS Linux 構成ディレクトリー /etc/opt/ibm/sna で、環境という名前のテキスト・ファイルを作成し、各環境変数を別々の行に設定します。例えば、

```
LANG=en_US のエクスポート
```

```
エクスポート PATH='$PATH:/opt/ibm/sna/bin'
```

```
LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib をエクスポートします
```

```
LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib をエクスポートします
```

CS Linux プログラムへのパスの指定

CS Linux 実行可能プログラムは、CS Linux に固有のディレクトリーに格納されます。プログラムを実行するときには、このディレクトリーへのパスを指定する必要があります。パスを指定するには、最初にプログラムを実行する前にディレクトリーをパス環境変数に追加するか、またはプログラムを実行するたびにディレクトリー名を指定します。

Motif 管理プログラムはディレクトリー /opt/ibm/sna/bin/X11 に保管され、その他のプログラムはディレクトリー /opt/ibm/sna/bin に保管されます。これらのディレクトリーを . ログインファイルまたは . プロファイル ファイル内のパス環境変数の定義に追加すると、CS Linux はプログラムを自動的に検索します。あるいは、以下の例のように、プログラムを実行するときにディレクトリー名を指定することもできます。

```
/opt/ibm/sna/sna/bin/sna スタート
```

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

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

このマニュアルで示されているサンプル・コマンド行では、ディレクトリーをパス環境変数に追加したことを前提としており、ディレクトリー名は含まれていません。

CS Linux サーバーの有効化

このセクションでは、サーバーとしてインストールされているコンピューター（つまり、SNA ノード・コンポーネントがインストールされている）で CS Linux を使用可能にする方法について説明します。CS Linux をクライアント上で使用可能にする場合は、[132 ページの『AIX または Linux でのリモート API クライアントの使用可能化および使用不可』](#)を参照してください。

ローカル・ノード（ローカルまたはリモートの CS Linux ノードのいずれか）を構成または管理するには、ローカル・システム上で CS Linux を使用可能にする必要があります。

CS Linux ソフトウェアを使用可能にするには、Linux コマンド・プロンプトで以下のコマンドを入力します。

```
スネアスタート [ -s ] [
```

```
  -m カーネル・メモリーの限度額 [
```

```
-t ]
```

注: スネアスタート コマンドを使用すると、CS Linux ソフトウェアは、コマンドの発行元のディレクトリを現行作業ディレクトリとして使用し、そのディレクトリ内の 1 つ以上のオープン・ファイル記述子を保守します。つまり、CS Linux ソフトウェアの実行中は、そのディレクトリを含むファイル・システムをアンマウントすることはできません。問題を回避するには、アンマウントする必要がないファイル・システム上のディレクトリから CS Linux ソフトウェアを開始する必要があります。例えば、スネアスタート コマンドを使用する前に、`cd /` を使用してルート・ディレクトリに変更することができます。

CS Linux をインストールすると、インストール・ユーティリティーは自動的に始動ファイル `/etc/rc.d/init.d/snastart` を更新して、スネアスタート コマンドを組み込みます。これにより、CS Linux がシステム始動時に自動的に開始されます。CS Linux を自動的に開始させたくない場合は、この行を削除またはコメント化してから、このセクションの指示に従って、CS Linux ソフトウェアを手動で使用可能にすることができます。

スネアスタート コマンドは以下のとおりです。のパラメーターとオプション

-s

CS Linux がシステム・コンソールにメッセージを書き込めないことを指定します。このオプションを使用しない場合、CS Linux は、メッセージの終了時にコンソールにメッセージを書き込み、ログ・ファイルだけでなく、特定のエラー・ログ・メッセージのテキストもコンソールに書き込みます。

-m カーネル・メモリーの限度額

CS Linux が任意の時点で使用するカーネル・メモリーの最大量（キロバイト単位）を指定します。（カーネル・メモリーは、内部データ構造のために使用されます。）CS Linux のコンポーネントが、CS Linux コンポーネントに現在割り振られているメモリーの合計量をこの限度を超える原因となるカーネル・メモリーを割り振ろうとすると、割り振りの試みは失敗します。

このオプションを使用しない場合、カーネル・メモリー使用量は制限されません。

-t

カーネル・コンポーネント間のすべてのインターフェース、およびクライアント/サーバー・トレースのトレースをアクティブ化します。（このオプションは、DLC トレースをオンにしません。）トレースにより、始動時に発生する問題を診断することができます。このオプションを使用しない場合、トレースはすべてのインターフェースで非アクティブになります。次に、コマンド行管理プログラム `snaadmin` を使用して、必要に応じて特定のインターフェース上でトレースを活動化することができます。

すべてのインターフェースをトレースすると、CS Linux コンポーネントのパフォーマンスが低下します。ソフトウェアが使用可能になると、コマンド行管理プログラム `snaadmin` を使用して、不要なインターフェースに対するトレースを停止することができます。トレースについて詳しくは、*IBM Communications Server for Data Center デプロイメント on Linux 診断ガイド* を参照してください。

CS Linux は、標準エラー（通常は端末の画面）にメッセージを書き込み、初期化が行われていることを示し、初期化が正常に完了したかどうかを示します。

初期化が失敗した場合、メッセージには、エラーの原因に関する情報が含まれ、(該当する場合は) Linux オペレーティング・システムのエラー・メッセージなどの追加情報が含まれます。標準エラーに書き込まれるテキストには、エラー・ログ・ファイルに追加情報を見つけることができることを示すメッセージが含まれている場合もあります。次に、スナスタート コマンドはゼロ以外の終了コードで終了します。終了コードはエラーの性質を示します。

終了コード値の詳細については、*IBM Communications Server for Data Center デプロイメント on Linux 診断ガイド*を参照してください。

sna start コマンドの拡張オプション

場合によっては、特に新しい CS Linux 構成をテストする場合には、(ファイル /etc/opt/ibm/sna/sna_node.cfg および /etc/opt/ibm/sna/sna_domn.cfg の標準構成ではなく)一時ファイルに保存した構成を使用して CS Linux を開始したい場合があります。これを行うには、スナスタート コマンドで以下の追加オプションを使用できます。

-n ノード構成ファイル

-d domain_config_file

ノード構成ファイル is the full pathname of the file to which you have saved node configuration (instead of /etc/opt/ibm/sna/sna_node.cfg)、および domain_config_file is the full pathname of the file to which you have saved domain configuration (instead of /etc/opt/ibm/sna/sna_domn.cfg).

注: これらのオプションは、一般的な使用を目的とした特定の要件を持っていない場合は、それらを使用しないでください。

これらのオプションを使用して CS Linux が実行されている場合、スナゲト pd コマンドは正しく動作しません。これは、常に標準構成ファイルから情報を収集するためです。スナゲト pd を使用する前に、これらのオプションを指定せずに CS Linux を開始することにより、標準の構成ファイルで実行している

CS Linux サーバーを使用不可に設定する方法

サーバー上で CS Linux ソフトウェアを使用不可に設定すると、CS Linux ノードと、関連した接続コンポーネントが自動的に停止します。CS Linux を使用不可にすると、他のどのプロセスも (3270 エミュレーション・プログラムなど)、このサーバー上の CS Linux リソースを使用するのを停止します。

通常は、ユーザーが個々のサービスの使用を終了したときにそれらのサービスを停止するようにし、システムを使用不可にするのは、CS Linux アクティビティーがないときのみに行ってください。クライアント上で CS Linux ソフトウェアを使用不可に設定すると、クライアント上で実行されているプログラムが CS Linux の機能にアクセスできなくなります。

まだアクティブ・ユーザーがいる間に CS Linux を使用不可にする場合は、CS Linux をこれから停止することを全ユーザーに通知し、ソフトウェアを使用不可にする前にユーザーが各自のアクティビティーを終了する時間の余裕を与えるようにしてください。アクティブ・ユーザーの詳細を表示するには、Motif 管理プログラムまたは コマンド行管理プログラムを使用します。

CS Linux ソフトウェアを使用不可にしたときに、3270 エミュレーション・プログラムでノードの LU を使用中だった場合は、それらの LU を使用していた 3270 エミュレーション・セッションはすべて終了します。プログラムは実行を続けますが、ソフトウェアが再び使用可能にされるまでは、ユーザーはセッションを使用できません。APPC、CSV、LUA、NOF、または MS API を使用中のアプリケーションには COMM_SUBSYSTEM_ABENDED 戻りコードにより通知され、CPI-C アプリケーションには CM_PRODUCT_SPECIFIC_ERROR 戻りコードにより通知されます。

CS Linux ソフトウェアを使用不可にするには、Linux コマンド・プロンプトに次のコマンドを入力します。

```
sna stop
```

CS Linux を使用不可にする処理が正常に行われると、sna stop から終了コード 0 が戻されます。これ以外の終了コードは、エラーが起こって CS Linux ソフトウェアが使用不可にされなかったことを示します。終了コードの値について詳しくは、「*IBM Communications Server for Data Center Deployment on Linux 診断ガイド*」を参照してください。

Motif 管理プログラムの使用

Motif 管理プログラムは、CS Linux を構成するための使いやすい インターフェースを提供します。このプログラムは、手順に沿って操作すれば構成プロセスを完了できるようになっていて、実用構成を作成するために指定しなければならない情報も 最小限で済むため、CS Linux 管理のツールとしてお勧めするツールです。

Motif 管理プログラムを使用して、アクティブになっている状態の CS Linux システムを管理することもできます。CS Linux がアクティブになっている間に、この管理プログラムを使用して、構成を変更しその変更を適用することができます。リソースは追加、変更、および除去でき(ほとんどの場合は、ノードとそのリソースがアクティブであっても可能です)、変更済みの構成はその後の継続操作に即時に使用できます。

Motif 管理プログラムでは、構成に使用するのと同じインターフェースを介して最新の状況情報が表示されるため、ドメイン・リソースとノード・リソースの両方の状況情報に簡単にアクセスできます。

これに代わる方法として、CS Linux コマンドを使用してシステムを構成し管理できます。構成コマンドおよび管理コマンドの要約を [46 ページの『コマンド行管理プログラムの使用』](#) に示します。

Motif 管理プログラムの呼び出し

Motif 管理プログラムを CS Linux 用に使用するには、まず、CS Linux が [35 ページの『CS Linux サーバーの有効化』](#) に説明する方法で使用可能にされていることを確認します。(他の X/Motif アプリケーションの場合と同様に、DISPLAY 環境変数を設定して適切な X サーバーを指示する必要がある場合もあります。)

Motif 管理プログラムの実行をバックグラウンドで開始するには、次のコマンドを発行します。

```
xsnaadmin &
```

クライアント/サーバー環境では、CS Linux は「ドメイン (Domain)」ウィンドウを表示します。

スタンドアロン・システムの場合、CS Linux は通常、「ノード (Node)」ウィンドウを表示します。しかし、ローカル・ノードをまだ構成していない場合は、最初にノードを構成するときのヘルプ画面が表示されます。

注: 本書では、CS Linux リソースに関する情報を表示する Motif のウィンドウを表すのに、「ウィンドウ」という用語を使用します。ウィンドウには、1 つまたは複数のセクション、つまりペイン (画面区画) が含まれていることがあります。ダイアログとは、情報を入力できる Motif のウィンドウのことです。

リソース・ウィンドウ

「ドメイン (Domain)」ウィンドウと「ノード (Node)」ウィンドウには、必要な情報のほとんどが表示され、追加情報にも簡単にアクセスできます。これらのウィンドウから、ローカル・ネットワーク内のリソースに関する情報を簡単に表示できます。

「ドメイン (Domain)」ウィンドウには、定義済みのノードがすべて表示され、このウィンドウからノードの追加、削除、開始、および停止を行うことができます。任意のノードをダブルクリックすると、そのノードの「ノード (Node)」ウィンドウが表示されます。

「ノード (Node)」ウィンドウには、特定のノードの主要リソースがすべて表示されます。

「ドメイン (Domain)」ウィンドウと「ノード (Node)」ウィンドウのメニューには、次の機能があります。

Selection (選択)

このメニュー内の機能は、「ドメイン (Domain)」ウィンドウ内で現在選択されているノード、または「ノード (Node)」ウィンドウ内で現在選択されている項目に関連しています。このメニューから、ノードを開始または停止したり、ノードの詳細を「ノード (Node)」ウィンドウに表示したりできます。「Node (ノード)」ウィンドウで項目を選択すると、このメニュー内の制御項目を使用してその項目を制御、変更、または削除でき、現在選択されているペインで新規項目を追加できます。

Services (サービス)

このメニューから、共通作業用にノードを構成するために必要なすべてのダイアログに簡単にアクセスできます。このメニューを使用して、リソースを追加または変更でき、構成作業および管理作業のためのヘルプを表示できます。

Diagnostics (診断)

このメニュー内の項目から、ロギングおよびトレースを制御できます。

Windows (ウィンドウ)

このメニューから他のウィンドウに簡単にアクセスできます。アクセスできるウィンドウには、次のものがあります。

- 「LU Pools (LU プール)」ウィンドウ
- 「CPI-C Destination Names (CPI-C 宛先名)」ウィンドウ

選択したリソースおよび指定したオプションに基づいて、管理プログラムから追加のリソース・ウィンドウ、構成ダイアログ、または状況ログが表示されます。また、構成する特定のリソースを選択できるコンテキスト・ダイアログ、選択した項目についての確認を求める確認ダイアログ、およびフィードバック情報またはエラー情報を表示するメッセージ・ポップアップも表示されます。それぞれのウィンドウおよびダイアログには、ヘルプ・オプションも含まれています。

Domain window

The Domain window shows each active SNA node in the CS Linux domain for the system you are using. (A node does not appear in the Domain window if CS Linux is not running on the node.) Each node is identified using the name of the system. The Domain window also shows the current status of each node in the domain.

Note: If a server is unexpectedly missing from the list of nodes in the Domain window, verify that the server is switched on and that the CS Linux software is running on the server. If necessary, start the CS Linux software on that node using the `sna start` command (see “CS Linux サーバーの有効化” on page 35).

One node in a domain is always identified as the configuration server for the domain. The Domain window shows the word "Controller" next to that node. The Controller configuration server always contains configuration information for domain resources. Backup configuration servers are identified by the word "Backup" on this window. Backup configuration servers contain copies of the configuration information for domain resources.

An example of a Domain window is shown in [Figure 13 on page 38](#).

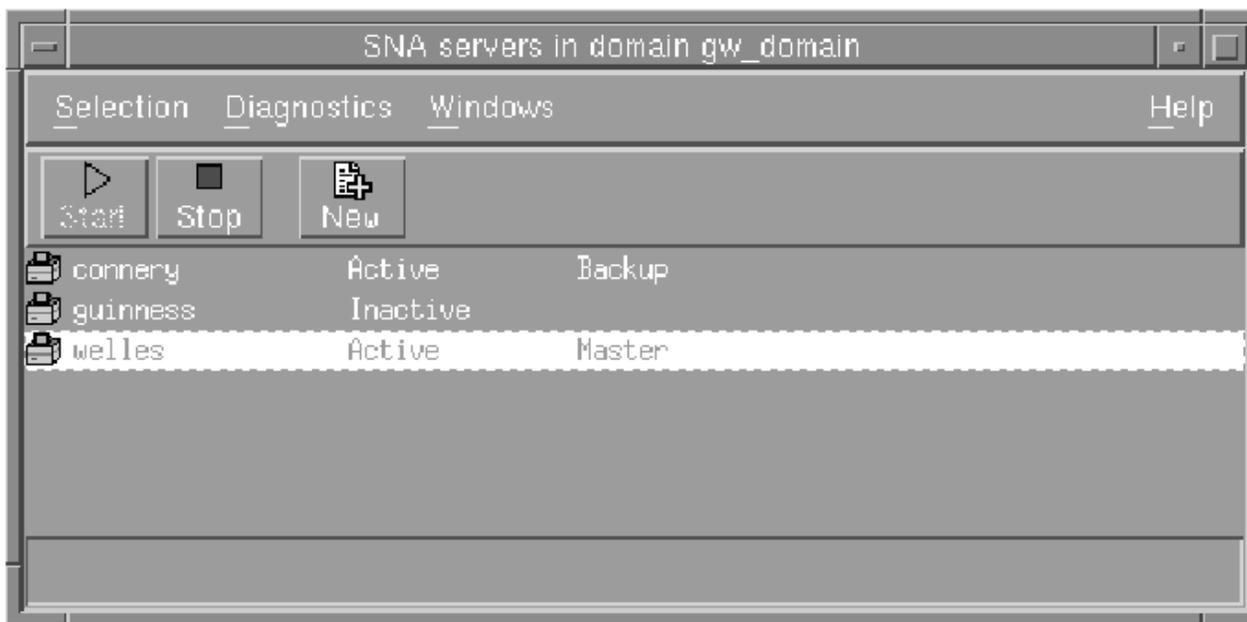


Figure 13. CS Linux Domain Window

If any active nodes in the domain (nodes on which CS Linux is running) are not configured, CS Linux prompts you to configure the node.

Note: The Domain window does not list IBM Remote API Clients. Clients use the resources of CS Linux servers (SNA nodes) to access SNA resources.

You can perform any of the following administration tasks from the Domain window:

Start or stop any node in the domain

Select the line for the node and click on the Start or Stop button on this window. (Alternatively, you can click on the line for the node, then select Start node or Stop node from the Selection menu.)

Administer a specific node

Double-click on the line for that node on the Domain window. (Alternatively, you can click on the line for the node, then select Properties from the Selection menu. You can also select the window for the node from the Windows menu.)

When you select a node to be administered, CS Linux displays the Node window as shown in [図 14 on page 40](#). (For a standalone system, CS Linux does not display the Domain window, because the domain has only one node. Instead, CS Linux immediately displays the Node window when you start the administration program.)

Add a node to the list of servers for the domain

Click on the line for the node and select Make configuration server from the Selection menu.

Remove the node from the list of servers for the domain

Click on the line for the node and select Remove configuration server from the Selection menu.

Configure logging for all nodes in the domain

Select Logging from the Diagnostics menu.

Turn tracing for a specific node on or off

Click on the line for the node and select Tracing on selected node from the Diagnostics menu.

Get information about domain resources

Choose any of the options on the Windows menu. In addition to shared domain resources, the Windows menu also lists each Node window in the domain.

ノード・ウィンドウ

サンプル・ノード・ウィンドウが [40 ページの図 14](#) に示されています。タイトル・バーには、システムの名前が表示されます。

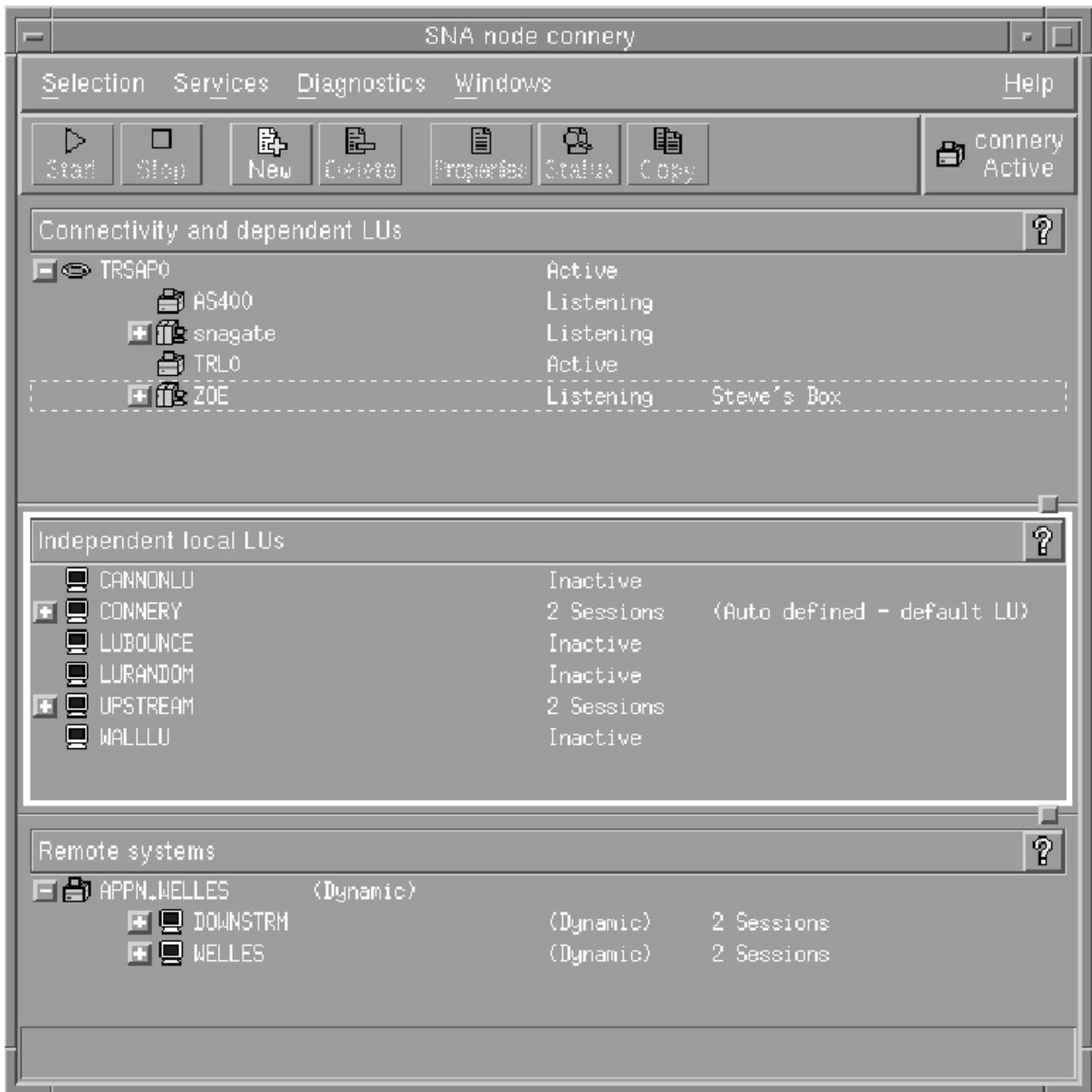


図 14. ノード・ウィンドウ

「Node (ノード)」ウィンドウから、CS Linux ノードのすべてのリソースとコンポーネントの追加、削除、変更、および管理を行うことができます。ウィンドウ内のリソースのレイアウトは、リソース間の関係を示し、どのリソースを表示するかを制御することができます。

ノード・ウィンドウの右上隅にある「ノード」ボックスは、ノードがアクティブであるか非アクティブであるかを示します。

ノード上に定義されているポート、ローカル LU、およびリモート・ノードは常に表示されます。「Node (ノード)」ウィンドウには、親ポートの下にある各リンク・ステーション、および親リンク・ステーションの下にある各従属 LU または、ローカル LU とリモート・ノードの下のパートナー LU も表示されます。

ノード・ウィンドウの本体は、ノードのさまざまなタイプのリソースについて、以下のペインに分割されます。

接続ペイン

「Node (ノード)」ウィンドウの上部ペインには、ノードの接続リソース (ポート、各ポート上のリンク・ステーションまたは PU、特定のリンク・ステーションまたは PU 上の従属 LU を含む) がリストされます。リソースごとに、このウィンドウに現在の状況情報が示されます。

「独立ローカル LU」ペイン

中央のペインには、ノードの独立 LU が示されます。LU ごとに、このウィンドウには LU を使用したセッションに関する情報も表示されます。

「リモート・システム

下部ペインには、リモート・ノードおよびパートナー LU に関する情報が表示されまた、各リモート・ノードまたはパートナー LU のセッション情報も表示されます。

ペインの相対サイズを変更するには、ペイン間の境界をクリックしてドラッグします。

ペインを選択するには、ペインをクリックします。また、リソースの行をクリックして、ペイン内の特定のリソースを選択することもできます。項目の構成を表示または変更するには、その項目をダブルクリックします。(このウィンドウのボタンおよびメニューを使用して、特定のリソースの構成情報にアクセスすることもできます。)

リストされている各項目について、その項目に属するリソースは、その項目の情報内にネストされます。例えば、リンク・ステーションは、所属するポートの下にグループ化されます。項目の横にある **展開** ボタン

 をクリックすると、その項目が現在表示されていない場合にその項目のリソースが表示されるか、

契約 ボタン  をクリックして項目のリソースを非表示にすることができます。

「Node (ノード)」ウィンドウから、以下の管理タスクを実行できます。

リソースの開始または停止

Select the resource and click on the **スタート** or **停止** button. (あるいは、セレクトションメニューから「**開始項目**」または「**停止項目**」を選択することもできます。)

アイテムの新規リソースの追加

項目を選択し、**ニュー** ボタンをクリックします (または、セレクトションメニューから **ニュー** を選択します)。例えば、ポートのリンク・ステーションを追加するには、ポートを選択し、「**ニュー**」ボタンをクリックします。

リソースの削除

リソースを選択し、**削除** ボタン (または、セレクトションメニューから **削除** を選択)。をクリックします。

任意のリソースの構成を表示または変更します。

リソースを選択し、**性質** ボタン (または、セレクトションメニューから **性質** を選択)。をクリックします。

任意のリソースの状況情報の取得

リソースを選択し、**状況** ボタン (または、セレクトションメニューから **状況** を選択)。をクリックします。

任意のリソースの構成のコピー

リソースを選択し、**コピー** ボタン (または、セレクトションメニューから **コピー** を選択)。をクリックします。

さらに、ノードの特定の構成タスクを **実行 menu, control logging (for the domain) and tracing (for the node) from the 診断 menu, and view or modify domain resources by selecting one of the items on the Windows menu.** から選択することができます。

リソース項目

ウィンドウ内のリソースのレイアウトは、リソース間の関係を表示します。

アイテムに 1 つ以上の子アイテムが関連付けられている場合は、その項目の横に **展開** ボタンまたは「**契約**」ボタンが表示されます。**展開** ボタンは、関連付けられた子項目が非表示であることを示します。「**展開**」ボタンをクリックすると、それらを表示できます。**契約** ボタンは、子項目が表示されることを示しま

す。契約 ボタンをクリックすると、それらを非表示にできます。アイテムの隣にボタンがない場合、そのアイテムには関連付けられた子リソースがありません。

例えば、リンク・ステーションが特定のポートに関連付けられているとします。「Node (ノード)」ウィンドウの「接続」ペインで、リンク・ステーションはその親ポートの下に、そのポートに関連付けられている他のすべてのリンク・ステーションとともに表示されます。ポートは常に表示されますが、関連するリンク・ステーションのリストを表示するか非表示にするかを選択することができます。同様に、関連する LU のリストを持つリンク・ステーションを拡張して、LU を表示したり、それらを非表示にしたりすることができます。

親リソースは、その子リソースの前に常に構成する必要があります。また、親リソースを削除すると、その子リソースはすべて削除されます。

ツールバー・ボタン

リソース・ウィンドウには、共通機能を簡単に実行できるようにするツールバー・ボタンがあります。CS Linux 用のツールバーは、[42 ページの図 15](#) に示されています。



図 15. CS Linux ツールバー

すべてのボタンが各リソース・ウィンドウのツールバーに表示されるわけではありません。ボタンの操作が現在選択されている項目に対して有効でない場合 (または、項目を選択する必要がありますが、何も選択されていない場合)、ボタンのアウトラインはグレーで表示され、機能を選択できません (ボタンを押すことはできません)。以下のボタンがリソース・ウィンドウに表示されます。



選択項目を開始します。



選択された項目を停止します。



新規リソース項目を追加します。(「Node (ノード)」ウィンドウで、選択したペインにリソースを追加します。)



選択したリソースを削除します



項目の構成を表示または変更するために、選択された項目のダイアログを開きます。



選択した項目をコピーします。このボタンを押すと、選択された項目の構成と重複するフィールドを持つダイアログが開きます。新しいリソースを追加するには、ダイアログのフィールド (新規項目の名前を入力) を入力します。



選択された項目の現在の状況を表示します。

ポートやリンク・ステーションなどの多くのリソースは、アクティブになっている間には変更できません。ただし、リソースを選択し、性質 ボタンをクリックしてダイアログを開くか、「状況」ボタンをクリックして、リソースの詳細な状況情報を表示します。をクリックすることにより、アクティブ・リソースのパラメーターを表示することができます。

Resource dialogs

Resource dialogs show the current configuration information for the resource. A sample dialog for an LU of types 0-3 is shown in [Figure 16 on page 43](#).

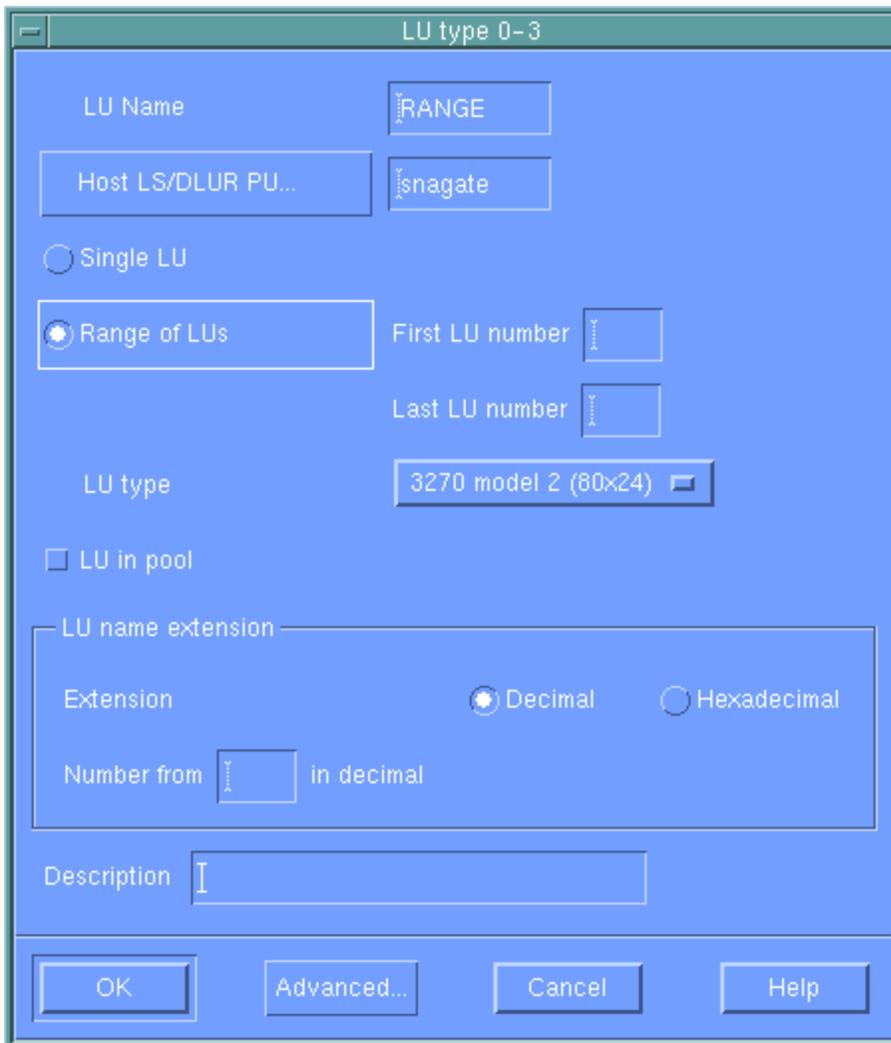


Figure 16. Sample Dialog

Resource dialogs guide you through the configuration process and supply default values whenever possible. For example, when you add a dependent LU, the Motif administration program automatically fills in the *LU number* field with an available LU number on the link station you specify. If you do not supply a required value, the program presents a message pop-up that indicates the information you need to provide.

Most dialogs provide a *Description* field; the information you enter there is displayed on the window where the resource is displayed.

If you are permitted to change the information in a resource dialog (when you are adding a new item or modifying an existing one), the dialog includes OK and Cancel buttons. Press the OK button when you are finished, or the Cancel button to exit without changing the configuration for the resource.

If you cannot change the information in a resource dialog (for example if the resource's configuration cannot be modified while it is active), the dialog includes a Close button instead of an OK button. Click this button when you are finished viewing the information in the dialog.

For context-sensitive help on the dialog, click on the Help button.

Note: The basic Motif dialogs expose only the key configuration fields; CS Linux supplies default values for advanced fields. To access advanced configuration parameters, click on the Advanced button. If you decide to adjust advanced parameters, complete the basic dialog before opening the advanced dialog, because that dialog can change depending on the values you enter for basic parameters. For information about advanced configuration fields, see the online help for the Motif administration program.

状況ダイアログ

リソースを選択して「Status (状況)」ボタンをクリックすると、44 ページの図 17 に示すように、詳細な状況情報が Motif 管理プログラムから表示されます。

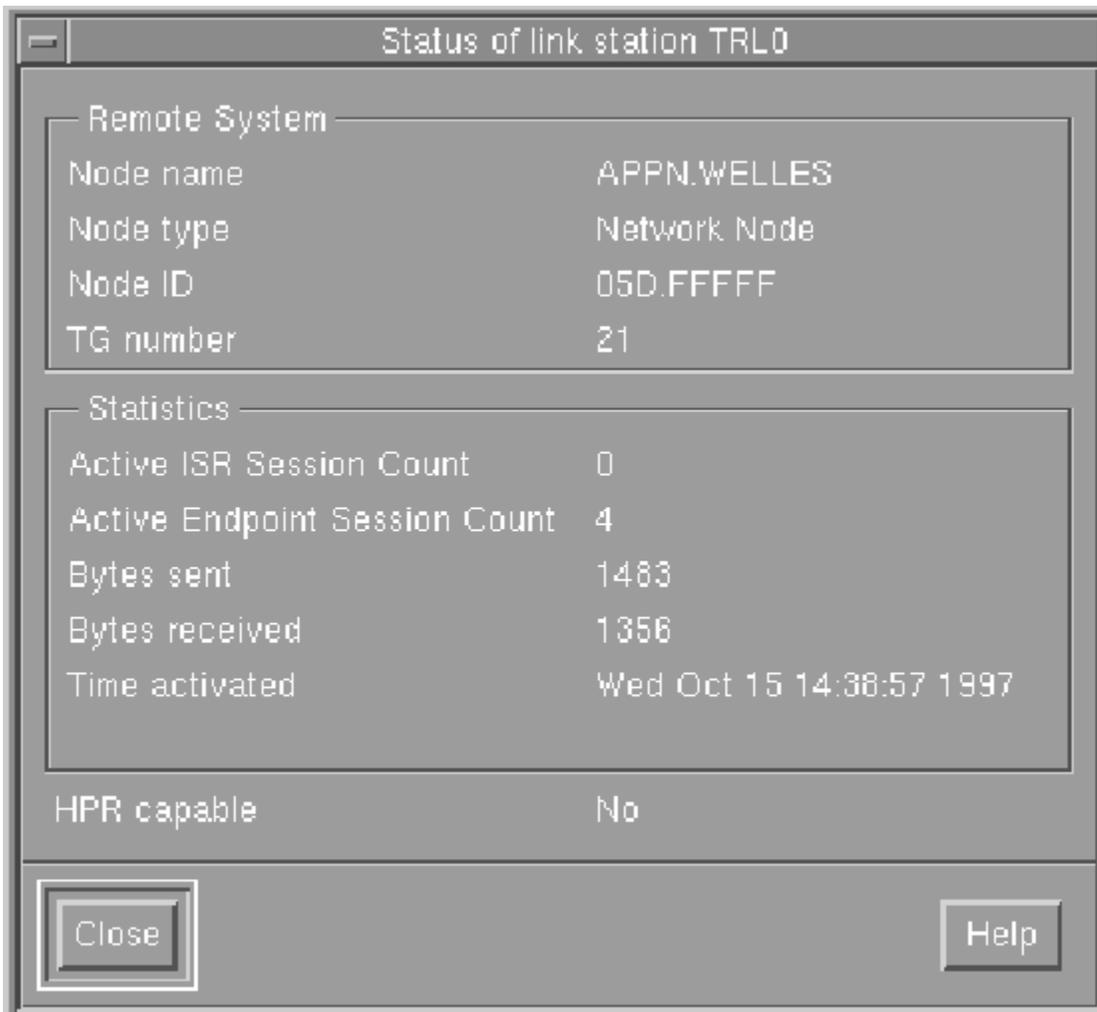


図 17. 状況ダイアログの例

状況ダイアログには、リソースの現在の状態に関する情報が表示されます。情報は、表示されている間も、動的に更新されます。

ヘルプ・ウィンドウ

Motif 管理プログラムのオンライン・ヘルプでは、実行する必要があるそれぞれの構成作業の詳細な手順が示されます。特に、作業シートには、特定のリソースを構成するために必要な操作が、順を追って示されます。ノード・パラメーターの構成用の作業シート (いつでもこれが CS Linux) の構成の最初のステップで (す) を 45 ページの図 18 に示します。

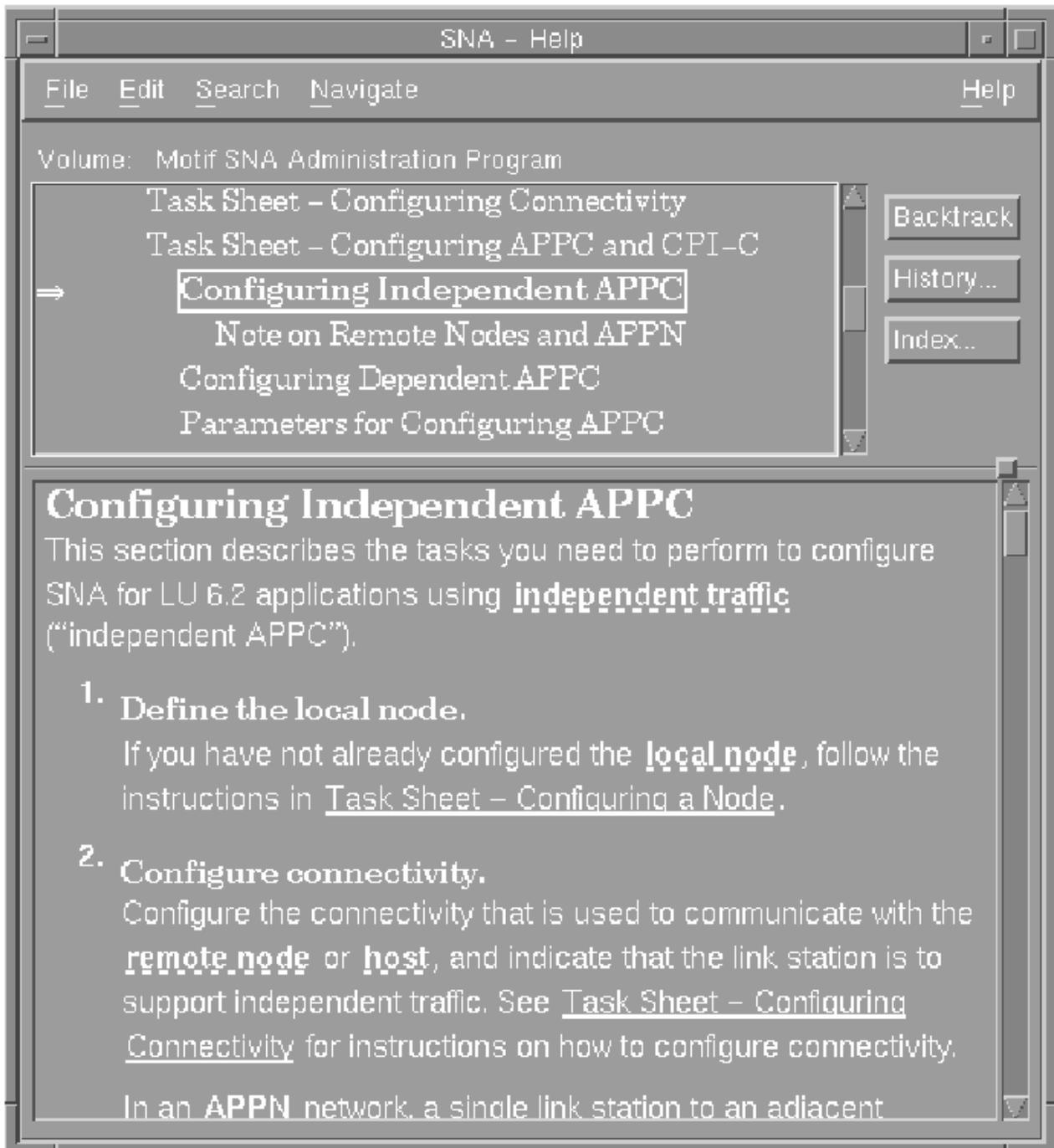


図 18. ヘルプ・ウィンドウの例

このほかに、ヘルプ・ウィンドウには、個々のウィンドウおよびダイアログ用のもの、エラー・メッセージ用のもの、および SNA の概念を説明するものなどがあります。

コマンド行管理プログラムの使用

コマンド行構成により、すべての CS Linux 構成パラメーターを変更することができます。これを使用して、Motif 管理プログラムを介して使用可能な任意のリソースを構成したり、Motif プログラムで公開されていない構成パラメーターを設定したり変更したりすることができます。ただし、この管理方式では通常、Motif 管理に必要な情報よりも多くの情報を指定する必要があります。また、指定した情報が有効で、既存のリソース定義と整合していることを確認する必要があります。(Motif 管理プログラムは、入力されたデータが整合していることを確認するため、推奨されます。さらに、メニューとダイアログの選択に基づいて多くの構成値を推測し、使用可能な定義に基づいて値を入力することができます。)

ほとんどの管理コマンドは、snaadmin コマンド行管理プログラムで使用されます。snaadmin コマンドは、以下の形式で発行できます。

snaadmin 指揮, パラメーター 1=値 1, パラメーター 2=値 2, ...

{サブレコード名 1}, サブパラメーター 1=サブ値 1,

サブパラメーター 2=サブ値 2...

You can get help for 以下のコマンドのいずれかを使用して、snaadmin コマンド行管理を実行

- snaadmin -h は、コマンド行の管理およびコマンド・ライン・ヘルプの使用法に関する情報について
- snaadmin -h -d は、snaadmin プログラムに提供されるコマンドのリストを提供します。
- snaadmin -h 指揮 は、名前付き 指揮のヘルプを提供します
- snaadmin -h -d 指揮 provides detailed help for the named 指揮, including a list of the configuration parameters that can be specified with the command.

コマンドには、サーバー名を指定するための -n オプションが含まれている場合に、IBM Remote API Client から発行できるものもあります。このようなコマンドは、指定されたサーバーで発行された場合と同じ効果を持ちます。

このセクションの残りの部分では、さまざまなタイプのリソースの管理コマンドを要約します。リストされるコマンドのタイプには、以下のものがあります。

スタウスス*

リソースのタイプに関する要約情報を提供します。

define_*

新しい define_* 構成ファイル内にレコードがあるか、または同じリソースの既存のレコードを新しい定義に置き換えます。を作成する。

削除 delete_*

対応する define_* ファイルからのレコード。を除去する

query_*

該当するコンポーネントの構成ファイルから情報を戻しますが、ファイルは変更しません。

設定値*

トレースおよびロギング・パラメーターなどの管理機能を制御します。

コマンド行構成の詳細については、*IBM Communications Server for Data Center デプロイメント on Linux* 管理コマンド解説書を参照してください。

第3章 基本構成作業

この章では、構成作業の概要を示し、CS Linux ノードを構成する方法について説明します。また、CS Linux をクライアント/サーバー環境で使用する場合に、コントローラー・サーバーとバックアップ・サーバーを構成する方法についても説明します。

クライアント/サーバー機能の構成

ここでの説明は、クライアント/サーバー環境(同じネットワークに複数の CS Linux ノードがある)で実行するために CS Linux をインストールした場合のみに該当します。

ポートや LU など、さまざまなリソースが個々のノード上で構成されます。これらのリソースは、「ノード・リソース」と呼ばれます。

その他のリソースは、すべてのノードに共通です。つまり、リソースの定義はドメイン全体で1つだけ維持されます。このようなリソースは、「ドメイン・リソース」と呼ばれます。ドメイン・リソース定義は、ドメインのコントローラー・サーバーにのみ保管され、ドメイン内のすべてのノードからアクセス可能です。

注: スタンドアロン CS Linux システムには、サーバーは1つだけ存在し、そのサーバーが常にコントローラーとして機能します。

クライアント/サーバー環境では、サーバーを構成サーバーとして指定できます。CS Linux は、これらの構成サーバーのリストを維持します。リストの先頭にあるサーバーはコントローラー・サーバーで、リストにあるその他のサーバーはバックアップ・サーバーです。サーバーは順序付きでリストされるので、コントローラー・サーバーが使用不可になった場合はリストの2番目にあるサーバー(第1バックアップ・サーバー)が引き継ぎ、コントローラーまたは第1バックアップ・サーバーのどちらも使用不可ならばリストの3番目にあるサーバー(第2バックアップ・サーバー)が引き継ぎ、以降も同様です。

ドメイン内でノードのいずれかがアクティブになると、ドメイン内で最初に使用可能な構成サーバー(通信が可能で、CS Linux ソフトウェアが実行されている最初のサーバー)がコントローラー・サーバーになります。現行のコントローラーが使用不可になった場合(ネットワーク障害などが原因で通信できない、またはコントローラー上で実行されている SNA ソフトウェアが停止したために)、リストにある次に使用可能な構成サーバーが新しくコントローラーになります。

CS Linux がコントローラーなしで稼働することがあります。この状態は、構成サーバー・リストにあるどのサーバーとも通信できない場合に発生します。この場合は、通信可能なサーバー上でのみノード・リソースを表示および構成できます。

注: コントローラー・サーバーとして機能するノードを直接指定することはできません。コントローラー・サーバーは、構成サーバー・リストにノードが追加された順序に基づいて選択されます。サーバーをリストの先頭に移動したい場合は、リストから他のノードをすべて除去してから追加し直してください。

次の管理コマンドを使用して、構成サーバーを照会、追加、および削除できます。

query_sna_net

ファイル内のサーバーをリストします。

add_backup

新規サーバーをリストの最後に追加します。

delete_backup

リストからサーバーを除去します。delete_backup コマンドを使用して、コントローラー・サーバーを削除する(リストの2番目にあるサーバーがコントローラーとしての役割を引き継ぐようにする)ことも、バックアップ・サーバーを削除する(そのサーバーがコントローラーとして機能できないようにする)こともできます。

注: サーバーが、リストの中で CS Linux ソフトウェアを実行している唯一のサーバーである場合、コントローラー・サーバーとしての役割を引き継ぐことができるサーバーが他にないので、そのサーバーは削除できません。クライアント/サーバー環境では、使用可能なコントローラー・サーバーが少なくとも1つ必要です。

ノードの構成

CS Linux は、始動時にローカル・ホスト名設定を使用してノードを識別します。代わりに、DNS 別名を使用する必要がある場合もあります。CS Linux サーバー名を DNS 別名に設定するには、次のステートメントを設定します。

```
export SNA_SERVER_NAME=DNS_alias_name_for_server
```

(環境ファイル /etc/opt/ibm/sna/environment 内)

クライアントは、Windows、AIX、および各種 Linux プラットフォームで使用できます。また、クライアントは、コンテナまたは AIX WPAR で実行できます。

111 ページの『第 10 章 CS Linux クライアント / サーバー・システムの管理』では、クライアントとサーバーを異なる CS Linux ドメインに移動する方法や、クライアント操作の詳細を構成する方法など、高度なクライアント / サーバー構成について説明しています。

ノードの構成

システム上で CS Linux を構成するための最初のステップは、ローカル・ノードを構成することです。ノード構成は、APPN ネットワークに参加するためにノードが必要とする基本情報を提供します。ノードの接続性またはその他のリソースを定義する前に、ノードを構成する必要があります。

ノードが既に構成されている場合は、ノード構成を変更する前にノードを停止する必要があります。

ノードを構成するには、以下の方法のいずれかを使用します。

Motif 管理プログラム

構成ノードのパラメーターは、「Node (ノード)」ウィンドウの「運行」メニューの選択

コマンド行管理プログラム

ノードの `define_node` コマンドを発行する

ノード構成の拡張パラメーターは、未定義のパートナー LU、セキュリティ障害の報告、および限定されたリソース・タイムアウトを持つセッションを制御します。

ノードの構成パラメーター

ノード構成には次の情報が必要です。

APPN support

ノードの APPN サポートのレベル。

- ネットワークが APPN ネットワークではない場合は、ノードを LEN ノードとして構成します。
- 別のノードがセッション・ルーティング・サービスを提供する APPN ネットワークに関与するため、または DLUR をローカル・ノードだけで使用するためには、ノードをエンド・ノードとして構成します。
- APPN ネットワークで中間ルーティング・サービスを提供するため、またはダウンストリーム・ノードにパススルー DLUR サービスを提供するためには、ノードをネットワーク・ノードとして構成します。
- メイン APPN バックボーン・ネットワークに含まれない、ブランチ・ネットワーク内の他のノードにセッション・ルーティング・サービスを提供するには、ノードをブランチ・ネットワーク・ノードとして構成します。

Control point name

ローカル・ノードの制御点の完全修飾名。この名前はネットワーク内の他のノードの構成にも必要な場合があるので、名前を決めるには SNA ネットワーク計画担当者と相談する必要があります。

制御点を定義すると、CS Linux が同じ名前のローカル LU を自動的に定義します。その LU は、ノードのデフォルト・ローカル LU として使用できます。

Control point alias

デフォルトのローカル LU のローカル別名。独立 LU 6.2 の LU でデフォルトのローカル LU が使用される場合は、この値を指定します。

Node ID

ローカル・ノードの PU の ID。ノードがデフォルト (制御点) の LU を使用する従属トラフィック用に使用される場合にのみ、値を指定します。

ノードを定義している場合は、ローカル・ホスト名と一致しない仮想 DNS 名を付けることができます。「*IBM Communications Server for Data Center Deployment on Linux* 管理コマンド解説書」の『付録』で説明している、`/etc/opt/ibm/sna/environment` ファイル内の `SNA_SERVER_NAME` 環境変数を使用します。

追加構成

ノードの構成が完了したら、次の構成作業を実行します。

- 51 ページの『第 4 章 接続コンポーネントの定義』の説明に従って、接続を構成する。
- 69 ページの『第 6 章 APPC 通信の構成』または 89 ページの『第 7 章 ユーザー・アプリケーションの定義』の説明に従って、ノード・リソース (LU) を構成する。
- 89 ページの『第 7 章 ユーザー・アプリケーションの定義』の説明に従って、アプリケーション を構成する。

ロギングの構成

CS Linux は、異常なイベント (およびオプションで通常のイベント) を記述するログ・メッセージをログ・ファイルに書き込みます。問題を診断しようとする時、ログ・ファイル内の最初の場所がログ・ファイルに表示されます。これは、ログ・メッセージに、問題の原因および実行すべきアクションに関する情報が示されているためです。

CS Linux は、以下のカテゴリーのイベントに関するメッセージを記録します

問題

ユーザー (セッションの異常終了など) に認識できないようにシステムを機能低下させる異常イベント。

例外

システムを低下させる異常なイベントですが、即時にユーザー (リソース不足など) には認識されないか、システムを低下させないイベントが発生しますが、後の例外や問題の原因 (リモート・システムから予期しないメッセージを受信するなど) を示している場合があります。

監査

通常のイベント (セッションの開始など)。

CS Linux はまた、使用量ログ・ファイルを保守します。これは、CS Linux リソースの現在の使用量とピーク使用量に関する情報を記録するために使用

通常の状態とエラー状態に関連するログを区別するために、異なるメッセージ・カテゴリーが異なるファイルに記録されます。問題メッセージおよび例外メッセージは、エラー・ログ・ファイルに記録されます。監査メッセージは監査ログ・ファイルに記録されます。

特定のイベントが頻繁に発生していることが分かり、ログ・ファイルが同じログ・メッセージの多くのインスタンスでいっぱいになっている場合は、1 つ以上の特定のログ・メッセージを 1 回だけログに記録するように指定するようにフィルターを設定することができます。これ以降、同じログ・メッセージのインスタンスは無視され、ログ・ファイルに書き込まれません。このフィルタリングは、すべてのタイプのログ (監査、例外、および問題ログ) に適用されます。ロギングのフィルタリングの詳細については、*IBM Communications Server for Data Center* デプロイメント on Linux 診断ガイドを参照してください。

CS Linux は、ログ・ファイルが大きくなりすぎてディスク・リソースを消費しないようにするためのバックアップ・メカニズムも提供します。ログ・ファイルが許可される最大サイズに達すると、CS Linux はその現行の内容をバックアップ・ファイルにコピーしてから、ログ・ファイルをクリアします。

デフォルトで、CS Linux は以下のログ・ファイルを使用します。

エラーログファイル

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

`/var/opt/ibm/sna/bak.err` (バックアップ)

監査ログ・ファイル

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

`/var/opt/ibm/sna/bak.aud` (バックアップ)

使用ログ・ファイル

`/var/opt/ibm/sna/sna. 使用法`

`/var/opt/ibm/sna/bak. 使用法` (バックアップ)

ログ・ファイルは、テキスト・エディターまたはその他の Linux システム・ユーティリティーを使用して表示できます。

vi

ファイルをテキスト・エディターで表示します。これにより、ファイル転送または逆方向にファイルを移動したり、特定のエントリーを検索したりすることができます。

パグ

一度に 1 ページずつファイルを表示します。このユーティリティーは単純で使いやすいものですが、ログ・ファイルが小さい場合にのみ役立ちます。

尾

ファイルのテール (終了) を表示します。ファイルの終わりは、最新のログ・メッセージがどこにあるかです。このユーティリティーは、システムの実行中にログ・ファイルをモニターするための `-f` オプション。と一緒に使用する

詳細ロギングではなく簡潔な項目を選択した場合は、特定のメッセージ番号の原因および処置に関する情報を判別するためのヘビ コマンド。を使用できます。

ほとんどの目的では、ロギング用のデフォルト設定で十分ですが、以下のタイプの変更を行うことができます。

- ログに記録されるメッセージのカテゴリーを示します。

問題メッセージは常にログに記録され、無効にできません。他の 2 つのメッセージ・カテゴリーでは、ロギングは通常は使用できませんが、必要に応じて使用可能にできます。

- ロギング・メッセージの詳細レベルを指定します。
- 各ノードのドメインまたはローカル・ロギングの中央ロギングを指定します
- 変更ログ・ファイルの名前およびサイズ。

ロギングを構成するには、次の方法のいずれかを使用します

Motif 管理プログラム

「ノード」ウィンドウまたは「ドメイン」ウィンドウの「診断」メニューからのロギング。の選択

コマンド行管理プログラム

次のいずれかのコマンドを実行します。

- `set_central_logging`
- `set_global_log_type`
- `set_log_type`
- `set_log_file`

Motif 管理プログラムの「ロギング」ダイアログは、ドメイン全体のログ設定に影響を与えます。コマンド行を使用すると、特定のマシンでローカル・ログ設定を構成することにより、ドメイン設定をオーバーライドできます。

Motif 管理プログラムは、ロギングを制御するだけでなく、トレースをノード・レベルで制御することもできます。コマンド行インターフェースは、ロギング機能とトレース機能の両方を制御します。ロギングおよびトレースの詳細については、「*IBM Communications Server for Data Center デプロイメント on Linux 診断ガイド*」を参照してください。

第4章 接続コンポーネントの定義

CS Linux ノードが他のノードと通信するためには、少なくとも1つの隣接ノードとの接続を構成する必要があります。接続リンクは、従属トラフィック、独立トラフィック、またはその両方を伝送するように構成できます。

コンピューターには、1つ以上のリンク・プロトコル用のアダプター・カードをインストールできます。接続を構成するために入力しなければならない情報の多くは、使用するリンク・プロトコルに応じて異なります。リモート・ノードにも、ローカル・ノードで選択したものと同一タイプのアダプター・カードが必要です。または、ローカル・ノードとリモート・ノード間のブリッジまたはルーターが必要です。CS Linux でサポートされるリンク・プロトコルのリストについては、52 ページの『DLC、ポート、および接続ネットワークの定義』を参照してください。

リンクを構成するには、52 ページの『DLC、ポート、および接続ネットワークの定義』の説明に従ってポートを定義する必要があります。さらに(ほとんどの場合には)、56 ページの『リンク装置の定義』の説明に従ってリンク・ステーションを構成する必要があります。ローカル・ノードが DLUR を使用してホストを通信する場合、62 ページの『DLUR PU の定義』の説明にしたがって、ローカル・ノードに DLUR PU も定義する必要があります。

Motif 管理プログラムを使用している場合は、データ・リンク制御 (DLC) は、ポートの構成の一部として自動的に構成されます。さらに、ポートを接続ネットワークの一部として定義するためのオプションもあります。コマンド行構成を使用している場合は、DLC 構成はポート構成とは別個に定義する必要があります。

リンク構成に必要な情報は、リンク・プロトコル、ネットワークが APPN ネットワークかどうか、およびリンクが従属トラフィック用か、独立トラフィック用か、その両用かに応じて異なります。さらに、構成が必要なリンクは、サポートしなければならない通信の種類によっても異なります。

LUA

LUA を使用する場合は、ホスト・コンピューターとのリンクを構成する必要があります。このリンクは、従属トラフィック用として構成し、CS Linux ノードとホスト・コンピューターの両方で同じように構成しなければなりません。したがって、SNA ネットワーク計画担当者に問い合わせる必要があります。

CPI-C または APCC を使用する場合

CPI-C または APCC を使用するときに、ネットワークが APPN ネットワークでない場合は、アクセスが必要なすべての隣接ノードとのリンクを構成する必要があります。これらのリンクは、独立トラフィックとして構成し、CS Linux ノードと隣接ノードの両方で同じように構成しなければなりません。したがって、SNA ネットワーク計画担当者に問い合わせる必要があります。

APPN ノードとして運用する場合

CS Linux ノードが APPN ネットワーク内のエンド・ノードまたはネットワーク・ノードの場合は、構成が必要なリンクの数は大幅に減少します。1つ以上の隣接ネットワーク・ノードとのリンクを構成し、それらのリンクを使用して APPN ネットワーク内のすべてのノードにアクセスできます。その他の隣接ノードに直接アクセスする場合は、それらのノードとのリンクも構成できます。これは通常は必要ありませんが、リンクを構成するとパフォーマンスが向上します。隣接ノードが LAN セグメントまたは Enterprise Extender を使用する IP ネットワークで接続されている場合は、直接リンクは動的にセットアップされるため、ユーザーが構成する必要はありません。ポートを定義する際に、ネットワークを接続ネットワークとして構成していることだけ確認してください。

独立 APCC の場合は、APPN ネットワーキングの利点を常に利用できますが、LUA の場合は、DLUR を使用していなければこの利点は適用されません。(DLUR は、ホストと、APPN ネットワーク内のローカル・ノードの LU または ダウンストリーム・ノードの LU との間の通信をサポートします。) DLUR を使用できるのは、ホストが DLUR をサポートしている場合のみです。したがって、DLUR の使用を考えている場合は、SNA ネットワーク計画担当者に問い合わせる必要があります。

DLC、ポート、および接続ネットワークの定義

ポートは、ネットワーク内の固有のアクセス・ポイントとして通信リンクのローカル・エンドを表します。各ポートは特定のリンク・プロトコルに関連付けられており、以下のいずれかになります。

- シュルツ
- トークンリング
- イーサネット
- X.25 または QLLC (修飾論理リンク制御)
- マルチパス・チャンネル (MPC)(CS Linux on System z のみ)
- エンタープライズ・エクステンダー (HPR/IP)

特定のリンク・プロトコルを使用する複数のポートを構成することができます。一般に、ポートはアダプター・カードなどの単一の物理アクセス・ポイントに対応しますが、いくつかのリンク・プロトコル(トークンリングなど)では、単一のアダプター用に複数のポートを定義することができます。異なるポートは、アドレス (SAP 番号など) によって区別されます。

特定のリンク・プロトコルのポートを定義するために Motif 管理プログラムを使用すると、CS Linux は、そのタイプの DLC がまだ定義されていない場合に、そのポートの DLC を自動的に定義します。コマンド行構成の場合は、異なるコマンドを使用してポートと DLC を定義する必要があります

トークンリング・リンク・プロトコルを使用する APPN ネットワークでは、「SAP 構成」ダイアログを使用して、ポートが接続ネットワークの一部であることを示すこともできます。

SNA ゲートウェイを使用している場合は、暗黙リンク・ステーション (明示的に構成されていないリンク・ステーション) の定義を生成するために使用されるテンプレートを定義できます。暗黙リンク・ステーションはダウンストリーム LU をサポートできます。ポートがアクティブな間に暗黙的な PU フィールドが変更された場合、変更は、変更後に生成された暗黙リンク・ステーション・インスタンスに影響を与えません。

ポート、接続ネットワークを構成するには、および DLC は、以下のいずれかの方法を使用します。

Motif または Web 管理プログラム

「ノード」ウィンドウの「運行」メニューから 接続性と 新規ポート。 の選択

コマンド行管理プログラム

DLC を構成するには:

```
define_タイプ_dlc
```

ポートを構成するには:

```
define_タイプ_port
```

In these commands, タイプ indicates the link protocol type (スドーク, tr (), イーサネット, クリーク, mpc, IP).

接続ネットワークを構成するには:

```
define_cn
```

拡張ポート構成パラメーターは、BTU サイズ、許可されるアクティブ・リンクの数、暗黙のダウンストリーム LU の生成、および動的リンク・ステーションの設定を制御します。

DLC、接続ネットワーク、およびポートの構成パラメーター

次のパラメーターはポート構成の必須パラメーターです。(Motif 管理プログラムを使用している場合は、ポート構成により、DLC に関する情報も表示され、接続ネットワークにポートを割り当てることができることでもあります。)

SNA port name

ローカルに定義されているポートの名前。

Adapter card number

このフィールドは、Enterprise Extender ポートには使用されません。

このコンピューターに同じタイプの複数のアダプター・カードがある場合に、使用するアダプター・カードを識別する番号。

Port number

このフィールドは、Enterprise Extender ポートには使用されません。

アダプター・カードが複数のポートをサポートできる場合に、使用するポートの番号。有効なポート番号の範囲は、0 から、アダプター・カードがサポートするポート数から 1 を引いた数までです。アダプター・カードの最初のポートには、0 を入力します。

このフィールドは、アダプター・カードが複数のポートをサポートする場合にのみ使用します。

Initially active

ノードの開始時にポートを自動的に開始させるかどうか。これを設定することにより、ポートを使用するリンク・ステーションを、隣接ノードからの要求またはローカル・ノードによる要求に応答して開始できるようになります。(ポートを開始しても、リンク・ステーションは開始されません。リンク・ステーションは別個に開始されます。)

次に、リンク・タイプに固有の追加ポート・パラメーターについて説明します。QLLC には、追加ポート・パラメーターは必要ありません。

SDLC 用の追加ポート・パラメーター**Line details**

次のパラメーターは、SDLC 接続のタイプを示します。

Type

次のいずれかの値を選択します。

Leased Line

このコンピューターとリモート・システムの間 SDLC リンクに、専用回線が使用されます。

Switched incoming

着呼に標準電話網が使用されます。

1 次ポート以外のポート (「Link role」フィールドで指定) には、ポーリング・アドレスも構成する必要があります (発呼の場合はこのアドレスはリンク・ステーションで構成されます)。ポーリング・アドレスは、1 バイトのアドレスです (デフォルトでは C1)。そのアドレスは、リモート・リンク・ステーションで構成されるポーリング・アドレスに一致していなければなりません。ポートは、アクティブになっていれば、このポーリング・アドレスと共に送信されるフレームに対して応答します。

1 次ポートの場合は、ポーリング・アドレスを構成する必要はありません。1 次ポートは、リモート・リンク・ステーションによって着呼に指定されているポーリング・アドレスを使用します。その他のタイプのポートの場合には、ポーリング・アドレスは各リンク・ステーションで構成されます。

Switched outgoing

発呼に標準電話網が使用されます。

Link role

このポート上に定義するリンク・ステーションについてのローカル・ノードの役割を示す値を選択します。SDLC 通信では、片方の終端がリンクを管理し、1 次リンク・ステーションと呼ばれます。他方の終端が、2 次リンク・ステーションです。

このフィールドには、次のいずれかの値を使用してください。

2 次

リンクの反対側の終端がコントローラーにより、リモート・システムは 1 次側として構成されます。ホスト・システムとのリンクを構成しているときは、ほとんどの場合これに該当します。

1 次

このポートはリンクの SDLC コントローラーとして動作し、リモート・システムは 2 次側として構成されます。

折衝可能

最大の柔軟性を得るために、この設定にすると、両端が折衝してどちら側が 1 次の役割を果たすかを定めることができます。リモート・システムがどの役割で構成されているかが分からないときは、この値を選択してください。

対等リンクにもこの設定を使用できますが、役割の折衝によりリンクの開始がやや遅れることに注意してください。

Primary Multi-drop

リンクは専用で、このポートはいくつかの 2 次ノードとの分岐接続リンクのコントローラーとして動作します。

ローカル・ノードから異なるリモート・ノードへの (例えば ダウンストリーム・ノードとのリンクのための) リンク・ステーションをいくつか構成する場合は、この設定を使用してください。これらの他のノードのそれぞれを、2 次側として構成する必要があり、専用回線を使用している必要があります。

Secondary Multi-PU

ローカル・ポートは、リモート・システム上のポートにより制御される分岐接続リンク上の 2 次ステーションの 1 つです。

パラメーターの構成方法が分からない場合は、SNA ネットワーク 計画担当者に問い合わせてください。

トークンリングおよびイーサネット用の追加ポート・パラメーター

Local SAP number

SAP のアドレス。通常、Intel および OSA2 アダプターの場合は 04 です。これ以外の値を使用するのは、カードで複数の SAP を使用する場合のみに行ってください。OSA Express アダプターの場合、ローカル SAP 番号は、この Linux イメージ上の ethX インターフェースに対応する入出力装置アドレスとして、OSA/SF で定義されている番号と一致していなければなりません。

SAP 番号は 4 の倍数でなければなりません。

このフィールドに入力する値が分からない場合は、SNA ネットワーク 計画担当者に問い合わせてください。

Define on connection network

SAP が接続ネットワークとして LAN にアクセスするかどうか。接続ネットワークを定義すると、接続ネットワーク上のノード間のリンクを動的に開始することができ、事前に構成しておく必要がなくなります。

LEN ノードは接続ネットワークを使用できないので、このフィールドが適用されるのは、ローカル・ノードが LEN ノードでない場合のみです。

CN name

接続ネットワークの名前。SAP を接続ネットワーク上に定義するための「Define on connection network」オプションを指定していない場合は、CN 名を入力する必要はありません。CN 名は、接続ネットワーク上のノード間にリンクを確立するために、仮想ルーティング・ノードの名前として使用されます。

接続ネットワーク上のすべてのノードに、同じ CN 名を指定してください。

Ethernet type

このフィールドは、イーサネット・リンクのみに適用されます。

ネットワークが標準イーサネット・ネットワークか、または IEEE 802.3 ネットワーク

Device name

このフィールドは、イーサネット・リンクのみに適用されます。

イーサネット・デバイスの名前。これを指定すると、アダプター番号は無視されます。例えば、RHEL システムでは、値 `device_name=ens32` はデフォルトの `x86_64` インストール済み環境で有効になります。このパラメーターは、`xснаadmin` では使用できません。

Enterprise Extender (HPR/IP) の追加ポート・パラメーター

ローカル IP インターフェース

これはオプションのフィールドです。これにより、複数の IP ネットワークにアクセスできる場合に、ポートで使用するローカル IP ネットワーク・インターフェースを指定できます。1つの IP ネットワークにのみアクセスできる場合は、このフィールドをブランクのままにすることができます。

インターフェースを指定する必要がある場合は、以下のいずれかを使用できます。

- インターフェース ID (「`eth0`」や「値 `0`」など)。
- IPv4 小数点付き 10 進数アドレス (193.1.11.100 など)。
- IPv6 コロン 16 進アドレス (2001:0db8:0000:0000:0000:0000:1428:57ab や 2001:db8::1428:57ab など)。

To determine the interface identifier, run the command `ipconfig - a` on the server where the card is installed. インターフェース ID とそれに関連付けられた IP アドレスがリストされます。

プロトコル

このポートのリンクが IPv4 または IPv6 アドレスを使用するかどうか。

上に定義 接続ネットワーク

アダプター・カードが接続ネットワークとして LAN にアクセスするかどうか。接続ネットワークを定義すると、接続ネットワーク上のノード間のリンクが、事前の構成なしで動的に開始されるようになります。

このフィールドは、LEN ノードが接続ネットワークを使用できないため、ローカル・ノードが LEN ノードではない場合にのみ適用されます。

CN 名

接続ネットワークの名前。接続ネットワーク上でポートを定義するために、接続時に定義 ネットワーク オプションを指定しない限り、CN 名を入力する必要はありません。CN 名は、接続ネットワーク上のノード間のリンクを確立するために、仮想ルーティング・ノードの名前として使用されます。

接続ネットワーク上のすべてのノードで同じ CN 名を指定してください。

暗黙リンク用の追加ポート・パラメーター

Maximum active template instances

テンプレートから生成されるリンク・ステーションのインスタンスの最大数を指定します。

Configure downstream LUs for implicit PU access

この PU を使用するダウンストリーム LU を構成するかどうか (98 ページの『SNA ゲートウェイの構成』を参照)。

HPR supported on implicit links

高性能ルーティング (HPR) を暗黙リンク・ステーションでサポートするかどうか。

Link level error recovery on implicit links

リンク・レベルのエラー回復を使用して暗黙リンクで HPR トラフィックを送信するかどうか。

接続ネットワークの追加パラメーター

IPv6 Address Only

HPR/IP の IPv6 ネットワーク上の接続ネットワークを定義する場合にこのパラメーターを指定して、接続ネットワークの IP アドレッシングで IPv6 DNS 名のみを使用するか、IPv6 アドレスのみを使用するかを指定できます。デフォルトでは、IPv6 DNS 名のみを使用します。

追加構成

ポートの構成が終わったら、次の構成作業に進みます。

- 構成したポート上にリンク・ステーションを定義するには、[56 ページの『リンク装置の定義』](#)を参照してください。
- DLUR PU を定義するには、[62 ページの『DLUR PU の定義』](#)を参照してください。
- APPC 通信をサポートするには、[69 ページの『第 6 章 APPC 通信の構成』](#)を参照してください。

リンク装置の定義

SNA ネットワーク内の他のノードと通信するには、リンク・ステーションの特性を構成する必要があります (LS)。SNA ネットワーク内の隣接ノードに移動します。リンク・ステーションを定義するには、その前に、使用しているアダプター (およびリンク・プロトコル) 用のポートを定義する必要があります。リンク・ステーションを構成するために必要な情報のほとんどは、使用されているプロトコルと同じものです。

リンク・ステーションは、CS Linux ローカル・ノードとリモート・コンピューター間の SNA ネットワークを経由する論理パスを表します。リモート・コンピューターには、以下のいずれかを指定できます。

- CS Linux が 3270 または LUA 通信を使用してホスト・プログラムにアクセスするホスト・コンピューター (または、プログラム間通信に APPC または CPI-C を使用します)
- ピア・コンピューター。CS Linux とリモート・コンピューターが同じパートナーとして通信している (APPN ネットワーク内での標準的な配置)
- ダウンストリーム・コンピューター。ホストにアクセスするために、CS Linux SNA ゲートウェイ機能または DLUR 機能を使用する。

リンク・ステーションは、特定のポートに関連付けられています。各ポート上に 1 つ以上のリンク・ステーションを定義できます。

従属トラフィックをサポートする各リンク・ステーションには、PU (物理装置) が関連付けられています。PU はリンク・ステーションに関連付けられているため、CS Linux はそれらを別個のリソースとして扱いません。これらの PU は、リンク・ステーション構成の一部として構成され、リンク・ステーションの開始および停止の一部として開始および停止されます。リンク・ステーションは、「ノード」ウィンドウの接続セクションに表示されます。PU はどのウィンドウにも表示されません。

注: ほとんどの環境では、ポートにリンク・ステーションを追加する必要があります。ただし、動的に作成されたリンク・ステーションをダウンストリーム SNA ゲートウェイまたは APPC トラフィックのみに使用したい場合は、リンクが常にリモート・ノードから活動化される状況については、明示的に 1 つのリンクを構成する必要はありません。

リモート・ノードがローカル・ノードへの接続を試行したが、着信呼び出しで指定されたアドレスと一致するリンク・ステーションが定義されていない場合、CS Linux は、適切なポートがローカル・ノード上に定義されている場合、暗黙的に定義することができます。このダイナミックに作成されたこのリンク・ステーションは、接続の期間中、「ノード」ウィンドウの接続セクションに表示されます。

リンク・ステーションを構成するには、次の方法のいずれかを使用します。

Motif または Web 管理プログラム

「ノード」ウィンドウの「運行」メニューから 接続性 と 新規リンク・ステーション。 の選択

コマンド行管理プログラム

次のコマンドを発行します。

```
define_タイプ_ls
```

In this command, タイプ indicates the link protocol type (スドーク, tr (), イーサネット, クリーク, mpc, IP).

リンク・ステーションの拡張パラメーターは、伝送特性、XID 交換、オプション・リンク機能、リンクを使用した LU 0 から 3 セッションの圧縮、および再活動化プロシージャーに対する追加制御を提供します。

リンク・ステーションの構成パラメーター

Motif では、「Link Station Configuration (リンク・ステーション構成)」ダイアログには次のセクションがあり、各セクションには異なるカテゴリーの構成パラメーターが含まれています。

リンク・ステーション

ダイアログのこの領域は、サポートする LU トラフィックが 従属 LU 用か、独立 LU 用か、または、それら両方のためのものかに関係なく、すべてのリンク・ステーションに必要な情報を提供するために使用できます。このセクションのパラメーターの説明については、[57 ページの『共通リンク・ステーション・パラメーター』](#)を参照してください。

独立 LU トラフィック

この情報は、独立トラフィック用のリンク・ステーションを使用する場合にのみ指定します。このセクションのパラメーターの説明については、[59 ページの『独立 LU トラフィック用のパラメーター』](#)を参照してください。

従属 LU トラフィック

この情報は、従属トラフィック用のリンク・ステーションを使用する場合にのみ指定します。このセクションのパラメーターの説明については、[60 ページの『従属 LU トラフィック用のパラメーター』](#)を参照してください。

共通リンク・ステーション・パラメーター

次のパラメーターは、サポートするトラフィックが従属トラフィックか、独立トラフィックか、または、その両方かに関係なく、すべてのリンク・ステーションに必須です。

このダイアログのパラメーターの詳細については、オンライン・ヘルプまたは「*IBM Communications Server for Data Center Deployment on Linux* 管理コマンド解説書」を参照してください。

Name

リンク・ステーションをローカルに識別する名前。

SNA port name

隣接ノードへアクセスするために使用するポート。

Activation

リンク・ステーションを開始するために使用する方式。次のいずれかの方式を指定します。

By administrator

ローカル・システム管理者から要求があった場合にのみ、リンク・ステーションが開始されます。

On node startup

ノードの始動時に、リンク・ステーションが自動的に開始されます。

On demand

アプリケーションの接続を提供する必要があるときに、リンク・ステーションが自動的に開始されます。

リンク・ステーションはポートとは別に開始されます。したがって、ポートが既にアクティブになっている場合でも、リンク・ステーションを開始させる必要があります。ポートを開始しても、そのみではどのリンク・ステーションも開始されません。また、ポートを最初から開始されるように構成してあっても、ノードの始動時に関連のリンク・ステーションが自動的に開始されるという訳ではありません。ポートを開始することは、リンク・ステーションを開始できる状態にするということです。ローカル・ノードと隣接ノードの両方でポートがアクティブになっていなければ、リンク・ステーションを開始することはできません。

使用量に応じて課金されるリンクの場合は、コスト節減のために、不必要にリンクを開始するのを避けてください。

このフィールドの設定方法が分からない場合は、SNA ネットワーク 計画担当者にお問い合わせください。

LU traffic

リンクを流れる LU トラフィックのタイプ。このパラメーターによって、リンク定義に必要な他のパラメーターが決まります。

Enterprise Extender (HPR/IP) リンクの場合、このリンク・タイプは独立トラフィックのみをサポートするので、このパラメーターは使用されません。

Any

リンク・ステーションは、独立と従属の両方の LU トラフィックに 使用できます。このオプションを選択した場合は、この項で説明するフィールドのほかに、[59 ページの『独立 LU トラフィック用のパラメーター』](#) および [60 ページの『従属 LU トラフィック用のパラメーター』](#) に示すフィールドに値を指定する必要があります。

Independent only

リンク・ステーションは、独立 LU トラフィックのみに使用できます。このオプションを選択した場合は、この項で説明するフィールドのほかに、[59 ページの『独立 LU トラフィック用のパラメーター』](#) に示すフィールドに値を指定する必要があります。

Dependent only

リンク・ステーションは従属 LU トラフィックのみに使用できます。このオプションを選択した場合は、この項で説明するフィールドのほかに、[60 ページの『従属 LU トラフィック用のパラメーター』](#) に示すフィールドに値を指定する必要があります。

隣接ノードに接続するためのアドレッシング情報も指定する必要があります。必要なアドレッシング情報のタイプは、ポートの DLC タイプによって異なります。リモート・ノード用のアドレスを指定しないと、リンク・ステーションは非選択 listen リンク・ステーションとして動作し、任意のリモート・ノードからの着呼を受け付けます。

SDLC 用の追加リンク・ステーション・パラメーター

Poll address

リモート・ステーションのポーリング・アドレス。アドレスを 2 桁 (1 バイト) の 16 進数で指定します。通常は C1 から始まります。1 次リンク・ステーションは、この値を使用してリモート・ステーションをポーリングします。2 次リンク・ステーションは、この値を使用してポーリングに応答します。ポーリング・アドレスの入力方法は、リンクの役割に応じて異なります。

- リンクが Point-to-Point リンクの場合 (分岐接続でない場合) は、通常はアドレス C1 が使用されます。
- このリンクの親ポートが交換着呼ポートの場合は、ポーリング・アドレスはポートで構成され、リンク・ステーションごとに独立して構成することはできません。
- 1 次交換発呼リンク・ステーションの構成中、通信相手になるリモート 2 次ステーションのポーリング・アドレスが分からない場合は、1 次側に 0xFF というポーリング・アドレスを指定します。この値を指定すると、ノードは、2 次側のポーリング・アドレスの構成値に関係なく、2 次側からの応答を受け入れることができます。1 次以外のリンクの場合または交換発呼リンク以外のリンクの場合には、アドレス 0xFF は無効です。
- 分岐接続構成を使用する場合は、同じ 1 次ステーションと通信するすべての 2 次リンク・ステーションが、異なるポーリング・アドレスを持っていない限りなりません。

リンクの両端のポーリング・アドレスは一致している必要があります。リモート・システムで構成されているアドレスが分からない場合は、SNA ネットワーク計画担当者に問い合わせてください。

VTAM ホストでは、ポーリング・アドレスは VTAM PU 定義の中の ADDR= パラメーターとして構成されます。

AS/400 システムでは、ポーリング・アドレスは Line Description (回線記述) の STNADR パラメーターです。

トークンリングおよびイーサネット用の追加のリンク・ステーション・パラメーター

MAC address

リモート・ステーションの MAC アドレス。一連の 16 進数で入力します。MAC アドレスは、リモート・システム上のアダプター・カードを一意的に識別します。

使用する値が分からない場合は、SNA ネットワーク計画担当者に問い合わせてください。

このリンクのリモート・エンドが VTAM ホストである場合は、MAC アドレスは、VTAM ポート定義の MACADDR= パラメーターに示されています。

AS/400 システムへのリンクを構成している場合は、MAC アドレスは、回線記述の中の *ADPTADR* パラメーターです。

SAP number

リモート・コンピューターのポートの SAP 番号。SAP 番号は、同じアダプター・カードを使用している異なるリンクを区別します。これは 16 進数で、通常は 04 です。番号は 4 の倍数でなければなりません。

使用する値が分からない場合は、SNA ネットワーク計画担当者に問い合わせてください。

このリンクのリモート・エンドが VTAM ホストである場合は、SAP 番号は VTAM PU 定義の *SAPADDR*=パラメーターです。

AS/400 システムへのリンクを構成している場合は、SAP 番号は、回線記述 (Line Description) の中の *SSAP* パラメーターです。

X.25 (QLLC) 用の追加のリンク・ステーション・パラメーター

Remote X.25 address

リンクがスイッチド・バーチャル・サーキットの場合は、リモート DTE の DTE アドレスを一連の 16 進数字として入力します。

リンクがパーマネント・バーチャル・サーキットの場合は、リンク・ステーションが使用するバーチャル・サーキットを識別するチャンネル ID を入力します。チャンネル ID には、1 から最大 1024 までの番号が付けられます。パーマネント・バーチャル・サーキットが 1 つしかない場合、そのチャンネル ID はおそらく 1 です。

MPC 用の追加リンク・ステーション・パラメーター

MPC group

MPC (マルチパス・チャンネル) ドライバー構成情報の場合に特定のチャンネルを識別するために指定する MPC グループ名。

Enterprise Extender (HPR/IP) 用の追加リンク・ステーション・パラメーター

Remote IP host name

このリンクの宛先ノードのリモート・ホスト名。ホスト名は次のいずれかにします。このリンクで使用するポート上の *protocol* パラメーターにより、アドレスを IPv4 形式にするか IPv6 形式にするかが決まります。

- IPv4 ドット 10 進アドレス (例えば、193.1.11.100)
- IPv6 コロン 16 進アドレス (例えば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)
- 名前 (例えば、newbox.this.co.uk)
- 別名 (例えば、newbox)

名前または別名を指定した場合、これを Linux システムが (ローカル TCP/IP 構成を使用するか、ドメイン・ネーム・サーバーを使用して) 完全修飾名に解決できなければなりません。

独立 LU トラフィック用のパラメーター

このリンク・ステーションを独立 LU (APPC、5250、または CPI-C アプリケーションで使用するためのタイプ 6.2 の LU) で使用するために構成するには、次の情報が必要です。

Remote node name

リモート・ノードの完全に修飾された CP 名です。

リモート・システムが VTAM ホストである場合は、VTAM スタート・リストの *NETID* パラメーターにネットワーク名 (完全修飾名の最初の 8 文字) があります。後半の 8 文字は、VTAM スタート・リストの *SSCPNAME* パラメーターに示されています。

注: 新規リモート・ノードの名前を入力した場合は、その新規リモート・ノードでパートナー LU を定義できるように、リモート・ノード用の定義を追加することができます。(ローカル・ノードが LEN ノ

ードである場合は、リモート・ノード・タイプを指定する必要はなく、「*Remote node type*」フィールドは適用されません。)

この方法で新規リモート・ノードを定義するには、リモート・ノード名を指定するとともに、リモート・ノード・タイプ *Discover* 以外の値としてリモート・ノード・タイプを指定します。

代わりに、*Discover* を指定して、リモート・ノード名を空にしておくこともできます。これは、どの隣接ノードでもリンク・ステーションを使用できることを意味します。*Discover* オプションは、ローカル・ノードが LEN ノードの場合は、使用できません。

Remote node type

このリンク・ステーションを介してアクセスされるリモート・ノードの APPN サポートのレベル (ローカル・ノードがエンド・ノードまたはネットワーク・ノードの場合のみ適用可能)。

リモート・ノードが LEN ノードか、エンド・ノードか、ネットワーク・ノードか分からない場合は、*Discover* を選択できます。リモート・ノード上の APPN サポートのレベルを検出する処理により、リンクの開始がやや遅くなるので、タイプが分かっている場合は、それを指定する方が賢明です。それによって、ネットワークの構成の整合性も確保されます。

リンク・ステーションがオンデマンドで開始される場合は、*Discover* は選択できません。

ローカル・ノードが LEN ノードの場合は、このフィールドは適用されません。

Branch link type

このリンク・ステーションを介してアクセスされるリモート・ノードへのリンク・タイプです。(ローカル・ノードが分岐ネットワーク・ノードの場合のみ適用可能)

リモート・ノードがメイン APPN バックボーン内のネットワーク・ノードである場合は、(バックボーンへの) *Uplink* を選択します。リモート・ノードが分岐内のエンド・ノードである場合は、(分岐内の) *Downlink* を選択します。

リモート・ノードがネットワーク・ノードに構成されている場合、分岐リンク・タイプは、自動的に (バックボーンへの) *Uplink* に設定され、これを変更することはできません。

従属 LU トラフィック用のパラメーター

これらのパラメーターは、Enterprise Extender (HPR/IP) リンクには適用されません。これは、このリンク・タイプが独立トラフィックのみをサポートするからです。

リンク・ステーションを従属 LU トラフィック用に構成すると、そのリンク・ステーションと同じ名前の適正な PU が自動的に作成されます。

リンク・ステーションを従属 LU (3270 または LUA アプリケーション用のタイプ 0 から 3 の LU) で使用するために構成するには、次の情報が必要です。

Local node ID

SNA ネットワーク内でローカル・ノードを識別する値。

通常は、同じノード上のすべてのリンクに同じノード ID (デフォルト 値) を使用できます。ただし、特定の 1 ホストにアクセスする従属 LU が 256 以上必要な場合は、ホストへの複数のリンク・ステーションを構成し、255 以下の従属 LU と異なるローカル・ノード ID を各リンク・ステーションに割り振る必要があります。

リモート・ノードがローカル・ノード ID を確実に認識できる構成にするために、SNA ネットワーク計画担当者にお問い合わせください。

VTAM 構成では、最初の 3 桁は PU 定義の *IDBLK* パラメーター に一致し、最後の 5 桁は *IDNUM* パラメーターに一致していることが必要です。

AS/400 システムでは、ノード ID は *EXCHID* パラメーター で構成されています。

Remote node ID

リモート・リンク・ステーション用のノード ID (オプション。このリンク・ステーションへのアクセスを制限する必要がある場合にのみ使用)。リモート・ノード ID を指定した場合は、リモート・ノードのノード ID がこの定義に指定された値に一致したときのみ、リンクが開始されます。これは、リモート・ノードにより開始される場合にリンク・ステーションを 区別できるので、交換ポートにいくつか

のリンク・ステーションを構成する場合に役立ちます。リンク・ステーションはリモートのノードの CP 名によって区別することもできますが、リンクを開始するときに CP 名を送信しないリモート・ノードの場合は、CP 名でなくリモート・ノード ID を使用する必要があります。

リモート・ノード ID を指定しなかった場合は、リンクのアクティブ時にリモート・ノードのノード ID は検査されません。

Remote node role

次のいずれかのリモート (隣接) ノードの役割を指定します。

Host

リンク・ステーションは、ホスト・コンピューターとのセッションのために使用される従属 LU (3270 LU など) をサポートします (最もよくあるケースです)。SNA ゲートウェイまたは DLUR を使用してホスト接続を提供するノードへのリンクの場合は、そのリンクがホスト・コンピューターに直接接続されていない場合でも、隣接ノードの役割を Host に設定する必要があります。

Downstream (SNA gateway)

リンク・ステーションは、ローカル・ノードの SNA ゲートウェイ機能を使用してホストと通信するダウンストリーム・ノードです (ホストからは、ダウンストリーム・ノードの LU はローカル・ノードに常駐しているように見えます)。

Downstream (DLUR)

リンク・ステーションは、ローカル・ノードの DLUR 機能を使用してホストと通信するダウンストリーム・ノードです。 (ホストからは、ダウンストリーム・ノードの LU はローカル・ノードに常駐しているように見えます。)

このようなリンクを使用できるのは、ローカル・ノードが APPN ネットワーク・ノードのときのみです。

Downstream PU name

ダウンストリーム・ノードに関連した PU 名。この値は、ホスト・コンピューター上でこのダウンストリーム・ノード用に構成されている PU 名に一致していなければなりません。この名前に使用する値が分からない場合は、SNA ネットワーク計画担当者にお問い合わせください。

このフィールドを使用するのは、このリンク・ステーションをローカル・ノードの DLUR 機能を使用してホストと通信するダウンストリーム PU として指定した場合のみです。これは、「Remote node role」フィールドに「Downstream (DLUR)」を指定することにより指示できます。

詳しくは、62 ページの『DLUR PU の定義』を参照してください。

Upstream DLUS name

DLUS (ダウンストリーム PU がアクセスする LU サーバー) をサポートするホスト LU の完全修飾 LU 名。この名前に使用する値が分からない場合は、SNA ネットワーク計画担当者にお問い合わせください。

このフィールドを使用するのは、このリンク・ステーションをローカル・ノードの DLUR 機能を使用してホストと通信するダウンストリーム PU として指定した場合のみです。これは、「Remote node role」フィールドに「Downstream (DLUR)」を指定することにより指示できます。

追加構成

リンク・ステーションの構成が完了したら、次の構成作業を行います。

- DLUR PU を定義するには、62 ページの『DLUR PU の定義』を参照してください。
- パススルー・サービスを構成するには、91 ページの『第 8 章 パススルー・サービスの構成』を参照してください。
- 特定のユーザー・アプリケーションをサポートするには、89 ページの『第 7 章 ユーザー・アプリケーションの定義』を参照してください。
- APPC 通信をサポートするには、69 ページの『第 6 章 APPC 通信の構成』を参照してください。

DLUR PU の定義

通常、従属 LU セッションは、ホスト・コンピューターへの直接通信リンクを必要とします。APPN ネットワーク内で多数のノード (ホスト・ノードを含む) が相互に接続されている場合、そのうちのいくつかはホストへの直接接続を持っていない可能性があります。別のノードを介した間接接続のみです。これらの間接的に接続されたこれらのノードの LU からホストへの従属 LU セッションを確立することはできません。

従属 LU リクエスター (DLUR) は、この制限を克服するように設計された APPN 機能です。

このセクションでは、ホスト・コンピューターへの接続を提供する DLUR PU を構成する方法について説明します。DLUR PU を構成すると、ローカル・ノードは DLUR サービスを提供できるようになります。

APPN ノード上の DLUR (CS Linux を実行するノードなど) は、ホストの従属 LU サーバー (DLUS) と連動して、DLUR ノード上の従属 LU から DLUS ホストへのセッションを経路指定するために、ホストで稼働します。ホストへの経路は複数のノードにまたがることができ、APPN のネットワーク管理、動的リソース・ロケーション、および経路計算機能を利用することができます。DLUR は、LU が配置されているノード上で使用可能でなければなりません。また、DLUS はホスト・ノード上で使用可能でなければなりません、DLUR はセッション経路内の中間ノード上では必要ありません。

CS Linux DLUR ノードがネットワーク・ノードである場合は、CS Linux ノードに接続されているダウンストリーム・コンピューター上の従属 LU に対して、パススルー DLUR 機能を提供することもできます。これらの LU は、CS Linux ノード上の DLUR を使用して、ノードの内部の LU の場合と同様に、ネットワークを介してホストにアクセスすることができます。

ダウンストリーム・ノードにパススルー DLUR サービスを提供するには、まず、ダウンストリーム・ノードに関連付けられた PU 名を (ローカル・ノードで) 構成する必要があります。この値は、ホスト・コンピューター上のダウンストリーム・ノードに対して構成された PU 名と一致する必要があります。

DLUR PU を構成するには、次の方法のいずれかを使用します。

Motif または Web 管理プログラム

「ノード」ウィンドウの「運行」メニューから 接続性と 新規 DLUR PU。 の選択

コマンド行管理プログラム

次のコマンドを発行します。

```
define_internal_pu
```

通常、DLUR PU は、関連 LU を除去し、DLUR PU を削除することなく再定義することはできません。ただし、ノード ptf_flag 上書き内部 PU_PARAMETERS が設定されている場合は、コマンド行管理プログラムを使用して、ノードが非活動状態の関連 LU を持つ DLUR PU を再定義することができます。

DLUR PU の構成パラメーター

次のパラメーターは DLUR PU 構成の必須パラメーターです。

PU Name

ローカル・ノード上の DLUR PU ごとに、PU 名を指定します。この名前は、ホストで構成されている PU 名と一致している必要があります (CS Linux は、PU を識別するために、PU 名と PU ID の両方をホストに送信します。ホストは通常、PU 名で PU を識別しますが、一致する PU 名が見つからないときは、PU ID によって識別します。)

DLUS Name

DLUS をサポートするホスト LU の完全修飾 LU 名。

DLUR を使用するには、CS Linux の DLUR 構成要素がホストの DLUS との間に LU-LU セッションを確立する必要があります。

ホスト LU の名前を決定するには、SNA ネットワーク計画担当者に問い合わせてください。

Backup DLUS Name

このパラメーターはオプションです。DLUS Name で指定されたホストが使用不可の場合に使用される、バックアップ・ホスト LU の完全修飾 LU 名。

ホスト LU の名前を決定するには、SNA ネットワーク計画担当者にお問い合わせください。

PU ID

ホストへの接続をサポートするローカル・ノード上の PU の PU ID。この PU ID は、1 つは 3 桁 (ブロック番号として知られています) で、もう 1 つは 5 桁の、2 つの 16 進数ストリングから成っています。

各従属 LU は PU に関連付けられます。PU と LU の両方がホスト・コンピューター上に構成されます。それぞれの PU には、CS Linux ノード上に DLUR PU を定義する必要があります。PU ID は、この PU 用にホストで構成された PU ID に一致していなければなりません。

多くの場合、PU ID はノード ID と同じなので、ノード ID がデフォルトです。ただし、特定の 1 ホストにアクセスする従属 LU が 256 以上必要な場合は、複数の DLUR PU を構成し、255 以下の従属 LU と、異なる PU ID を各 DLUR PU に割り振る必要があります。

このフィールドの設定方法が分からない場合は、SNA ネットワーク計画担当者にお問い合わせください。

VTAM 構成では、最初の 3 桁は PU 定義の *IDBLK* パラメーターに一致し、最後の 5 桁は *IDNUM* 設定に合致している必要があります。

Initially active

ノードの始動時に DLUR PU を自動的に開始するかどうか。このオプションを設定しない場合、DLUR PU を手動で開始する必要があります。

Compression supported

この PU を使用する LU 0 から 3 セッションに対して、データ圧縮がサポートされるかどうか。このオプションを設定すると、ホストが要求した場合に圧縮が使用されます。このオプションを設定しない場合、圧縮は使用されません。

Retry contacting DLUS indefinitely

CS Linux が DLUS との通信を最初に試行して失敗した場合に、再試行するかどうか。このオプションを設定すると、CS Linux は最初の試行が失敗した場合に無制限に再試行します。このオプションを設定しない場合、再試行は 1 回のみ行います。

ダウンストリーム・ノードのパススルー DLUR 用のパラメーター

パススルー DLUR を使用してダウンストリーム・ノードの従属 LU とホスト の間でトラフィックをトランスポートできるように CS Linux を構成するには、次の情報が必要です。

Downstream PU name

ダウンストリーム・ノードに関連した PU 名。この PU 名は、ホスト・コンピューター上で構成された PU 名に一致していなければなりません。

1 つのダウンストリーム・ノードが複数の PU をサポートできます。この場合、各ダウンストリーム PU は異なるリンクに関連付けられるため、CS Linux DLUR ノードとダウンストリーム・ノードの間に複数のリンクを構成する必要があり、それぞれのリンクのダウンストリーム PU 名を知っている必要があります。

ダウンストリーム・ノードに関連付けられている PU 名を調べるには、SNA ネットワーク計画担当者にお問い合わせください。

DLUS name

DLUS をサポートするホスト LU の完全修飾 LU 名。DLUR を使用するには、CS Linux の DLUR 構成要素がホストの DLUS との間に LU-LU セッションを確立する必要があります。

ホスト DLUS サーバーの LU 名を調べるには、SNA ネットワーク計画担当者にお問い合わせください。

追加構成

DLUR の構成が完了したら、次の構成作業を実行します。

- DLUR 用の LU を構成するには、[99 ページの『DLUR の構成』](#)を参照してください。
- その他のパススルー・サービスを構成するには、[91 ページの『第 8 章 パススルー・サービスの構成』](#)を参照してください。

- 特定のユーザー・アプリケーションをサポートするには、[89 ページの『第7章 ユーザー・アプリケーションの定義』](#)を参照してください。
- APPC 通信をサポートするには、[69 ページの『第6章 APPC 通信の構成』](#)を参照してください。

第 5 章 従属 LU の構成

この章では、3270、TN3270、および LUA 通信を使用するユーザー・アプリケーションをサポートする LU および LU プールの構成方法を説明します。これらを使用するには、従属 LU を構成する必要があります。

この章で説明するリソースを構成する前に、次の構成を済ませておく必要があります。

- 48 ページの『ノードの構成』の説明に従って、ノードを構成する。
- 51 ページの『第 4 章 接続コンポーネントの定義』の説明に従って、接続を構成します。3270、TN3270、および LUA の場合は、従属 LU トラフィックをサポートするためのリンクを構成する必要があります。

アップストリーム SNA ゲートウェイまたは DLUR を使用している場合は、ホストへの直接リンクを構成する必要はありません。詳しくは、98 ページの『SNA ゲートウェイの構成』および 62 ページの『DLUR PU の定義』を参照してください。

LU タイプ 0 から 3 の定義

ホスト・システムとの通信をサポートするには、タイプ 0 から 3 の従属 LU を構成する必要があります。この節に示す情報を使用して、3270 または LUA をサポートする LU を定義できます。LU の範囲を定義して、同じタイプの複数の LU を一度の操作で構成することもできます。

タイプ 0 から 3 の LU を構成するには、次のどちらかの方式を使用します。

Motif または Web 管理プログラム

「Node (ノード)」ウィンドウの「Services (サービス)」メニューから、次のいずれかの項目を選択します。

- 「3270」と、「New 3270 display LU (新規 3270 ディスプレイ LU)」または「New 3270 printer LU (新規 3270 プリンター LU)」のどちらか
- 「LUA」と「New LUA LU (新規 LUA LU)」
- 「TN server (TN サーバー)」と「New host LU (新規 ホスト LU)」

コマンド行管理プログラム

次のいずれかのコマンドを発行します。

```
define_lu_0_to_3
define_lu_0_to_3_range
```

拡張ダイアログを使用して、特定の SSCP へのアクセスを制限したり、非活動タイムアウトを指定したりできます。

LU タイプ 0 から 3 の構成パラメーター

次のパラメーターは、LU タイプ 0 から 3 の構成の必須パラメーターです。

LU 名

1 から 8 文字の LU 名 (単一の LU の場合)、または 1 から 5 文字のベース名 (LU の範囲を指定すると、定義されたすべての LU は、ベース名に接頭部を追加することにより作成されます)。

この LU 名はローカルにのみ使用されるものなので、ホストに定義されている名前に対応している必要はありません。

Host LS/DLUR PU

ホストへのリンクを提供するリンク・ステーション。LU 定義は、選択するリンク・ステーションに所属します。(従属 LU が DLUR をサポートするノード上にある場合は、このフィールドには、ホストへの接続を提供する DLUR PU を指定します。)

LU numbers

単独の LU 番号または LU 番号の範囲。有効な LU 番号は 1 から 255 です。

指定する LU 番号は、ホスト VTAM 構成の中の番号に対応していなければなりません。ホストに構成されている番号が分からない場合は、SNA ネットワーク 計画担当者にお問い合わせください。

LU type

次の LU タイプの 1 つ (構成中の LU のタイプにより決まります)。

- 3270 ディスプレイ LU の場合は、画面サイズに応じて適切なモデルを指定します。
 - 3270 model 2 (80x24)
 - 3270 model 3 (80x32)
 - 3270 model 4 (80x43)
 - 3270 model 5 (132x27)
- プリンター LU の場合は、次のどちらかを指定します。
 - 3270 printer
 - SCS printer
- LU タイプが分からない場合、ローカル・ノードからホスト (アップストリーム LU) への SNA ゲートウェイをサポートするために使用する LU の場合、または LUA アプリケーション用の LU の場合は、Unrestricted (コマンド行を構成する場合は unknown) を指定してください。

LU タイプは、ホストの LU の構成に一致していなければなりません。必要な場合は、ホストで構成されている LU タイプが優先的に使用されます。

ユーザーが指定する値に応じて、CS Linux は、標準の VTAM テーブルで使用される値と一致する、以下のストリングのいずれかを DDDLU NMVT のホストへ送信します。

- 3270002 (3270 モデル 2 の場合)
- 3270003 (3270 モデル 3 の場合)
- 3270004 (3270 モデル 4 の場合)
- 3270005 (3270 モデル 5 の場合)
- 3270DSC (3270 プリンターの場合)
- 3270SCS (SCS プリンターの場合)
- 3270000 (RJE ワークステーションの場合)
- 327000n (Unrestricted、TN3270 クライアントの場合)。n はクライアント提供のモデル番号 (2 から 5) です。
- 327000@ (Unrestricted、LUA クライアントの場合)

ホストでこの LU を TN サーバーおよび DDDLU と共に使用する場合、この LU がホストで構成されていないことがあります。その場合は、ホストで動的に LU を定義するときに、ここに指定した LU タイプが使用されます。ダウンストリーム TN3270 クライアントが要求したタイプに一致するように LU モデル・タイプを定義したい場合は、Unrestricted (コマンド行構成の場合は unknown) を指定します。CS Linux は、通常、クライアントが指定する端末型式 (装置タイプ) から標準マッピングを使用して LU モデルを確定します。このマッピングの変更が必要な場合は、169 ページの『付録 C DDDLU 用の TN3270 LU モデルの構成』に説明がある `tn3270dev.dat` ファイルを使用して変更が可能です。

LU in pool

LU を LU プールに割り当てるかどうか。

注: この LU を LU プールに割り当て、ユーザーのセッションをこの LU に割り当てた場合、LU が使用可能ならばユーザーのセッションはこの LU を使用します。そうでなければ、セッションを特定の LU でなく LU プールに割り当てた場合と同様に、プールから任意の空き LU を使用します。ユーザーが使用する LU を指定のもののみ限定して、その LU が既に使用中の場合はユーザーのセッションを確立できないようにしたい場合は、その LU がプールに含まれていないことを確認してください。

Pool name

LU プールの名前。

追加構成

LU タイプ 0 から 3 の構成が完了したら、次の構成作業を行います。

- 3270 ディスプレイ、TN3270、または LUA 用の従属 LU のプールを使用するには、[67 ページの『LU プールの定義』](#)の説明に従って、LU プールを定義します。
- TN3270 の場合は、[91 ページの『TN サーバーの構成』](#)の説明に従って TN3270 のクライアント・アクセス・レコードを定義します。

LU プールの定義

3270、TN3270、および LUA の場合、LU プールを定義してユーザー構成を単純化し、ホスト・セッションの確立における柔軟性を高めます。

注: ユーザーのセッションは、特定の LU または LU プールに割り当てることができます。

- プール内にある特定の LU にユーザーのセッションを割り当てると、セッションは、その LU が使用可能であれば、この LU を使用します。それ以外の場合は、特定の LU ではなく LU プールに割り当てた場合と同様に、プールからの空き LU が使用されます。
- ユーザーが指定された LU のみを使用するようにしたい場合は、LU が既に使用されている場合にユーザーのセッションを確立できないようにするために、LU がプール内にないことを確認してください。

LU プールは、複数の CS Linux サーバーにまたがっても構いません。異なるサーバー上の同じ名前を持つ LU プールを定義するだけです。サーバーで障害が発生したり、サービスが提供されていない場合は、LU プールを使用するクライアントは別のサーバーを使用できます。また、LU プールを使用すると、クライアント構成が単純化され、別のサーバーを追加するか、既存のサーバーに LU を追加することによって、容量を簡単に拡張できます

「LU プール」ウィンドウを使用して、CS Linux ドメインのすべての LU プールを表示することができます。このウィンドウには、システムで構成されている LU プールがリストされ、LU プールに追加する LU を選択することができます。LU プール内の個々の LU は、LU プールの下にリストされます。

LU は次のように識別されます。

- 3270 ディスプレイ LU
- 制限なし LU
- SCS プリンター
- 3270 プリンター

同じプール内に異なるタイプの LU を混在させないでください (例えば、ディスプレイとプリンター LU を同じプールに入れないでください)。TN3270E クライアントをサポートしていなければ、プリンター LU のプールが必要になる可能性はありません。

LU プールを構成するには、以下の方法のいずれかを使用します。

Motif または Web 管理プログラム

「ノード」ウィンドウの「Windows」メニューから LU プールを選択し、ニューを選択してプールを追加します。の選択

コマンド行管理プログラム

次のコマンドを発行します。

プールの `define_lu_pool`

LU プールの構成パラメーター

次のパラメーターは、LU プールの構成の必須パラメーターです。

LU プールの定義

Name

LU プールを識別する名前。このフィールドは、新規 LU プールを追加する場合にのみ適用されます。既存のプールの変更することはできません。

Assigned LUs

プールに割り当てる LU。同じ LU を複数のプールのメンバーにすることはできません。

第 6 章 APPC 通信の構成

APPC アプリケーション、5250 エミュレーション・プログラム、および CPI-C アプリケーションのどれを使用するときも、まず必要なのは APPC を構成することです。APPC アプリケーションは、指定のモードを使用してホストまたは対等コンピューター上の他の APPC アプリケーションや CPI-C アプリケーションと通信するために、ノードの LU タイプ 6.2 (LU 6.2) のリソースを使用します。

CPI-C を使用するアプリケーションの場合は、APPC を構成したあとで、さらに CPI-C 構成も実行する必要があります。CPI-C アプリケーションでは、ノードの LU タイプ 6.2 リソースおよびモード・リソースを使用して、ホストまたは対等コンピューター上の別の APPC または CPI-C アプリケーションと通信します。CPI-C アプリケーションと APPC アプリケーションには、同じリソースを定義します。さらに、CS Linux コンピューター上の TP が呼び出し側 TP (会話を開始する TP) である場合は、その TP に関する 1 つ以上のサイド情報エントリーを定義することが必要な場合もあります (方法については、[84 ページの『CPI-C サイド情報の定義』](#)を参照)。これらのエントリーは、それぞれ、1 つのパートナー TP、その TP にアクセスするために使用する LU、モード・リソース、および必要なセキュリティ情報に関する情報を提供します。

APPC の構成手順は、LU 6.2 トラフィックが従属か独立かによって異なります。リモート・ノードがホストでない場合は、独立トラフィックを使用する必要があります。リモート・ノードがホストである場合は、従属または独立のどちらのトラフィックも使用できます。

APPC 通信を構成するには、その前に次の構成をしておくことが必要です。

- [48 ページの『ノードの構成』](#)の説明に従って、ノードを構成する。
- [51 ページの『第 4 章 接続コンポーネントの定義』](#)の説明に従って、接続を構成します。

注：APPN ネットワークでは、1 つの隣接ノードへの 1 つのリンク・ステーションを使用して、ネットワーク内のすべてのノードと通信できるので、各リモート・ノードへのリンク・ステーションを別々に構成する必要はありません。

多くの場合、APPC アプリケーションは、ローカルとリモートの両ノード上で制御点 LU を使用でき、そして標準モードを使用できます。その場合は、既に APPC 用の構成は済んでいるので、これ以上の構成は必要ありません。

ローカル・ノード上での APPC 通信は、次の手順で構成できます。ローカル・ノードおよびリモート・ノードのタイプ、そして使用するアプリケーションによっては、次の手順は不要な場合もあります。

1. [70 ページの『ローカル LU の定義』](#)の説明に従って、ローカル LU を定義します。
2. [71 ページの『リモートノードの定義』](#)の説明に従って、リモート・ノードを定義します。
3. [72 ページの『パートナー LU の定義』](#)の説明に従って、パートナー LU を定義します。
4. [75 ページの『TP の定義』](#)の説明に従って、呼び出し可能 TP を定義します。
5. [80 ページの『モードおよびサービス・クラスの定義』](#)の説明に従って、モードを定義します。
6. [84 ページの『CPI-C サイド情報の定義』](#)の説明に従って、CPI-C サイド情報を定義します。
7. [86 ページの『APPC セキュリティの構成』](#)の説明に従って、APPC セキュリティを定義します。
8. 5250 通信を構成するには、[89 ページの『第 7 章 ユーザー・アプリケーションの定義』](#)を参照してください。

さらに、2 つ以上のクライアント・コンピューター上で同じ APPC または CPI-C アプリケーションを実行している場合、それが指定するローカル LU 別号または TP 名のオーバーライドが必要になることがあります。これにより、2 つ以上のアプリケーションのコピーを、アプリケーションを変更せずに異なるローカル LU を使用するように割り当てたり、同じアプリケーションの 2 つ以上のコピーを区別したりできます。詳しくは、[136 ページの『クライアント・アプリケーション上のハード・コード LU 別号をサーバーのドメイン内の LU 別号にマップする』](#) (AIX または Linux 上の Remote API Client の場合) および [130 ページの『クライアント・アプリケーションでハードコーディングされた LU 別号をサーバーのドメイン内の LU 別号にマップする』](#) (Windows クライアントの場合) を参照してください。

ローカル LU の定義

多くの場合、アプリケーションはローカル・ノードの制御点 LU を使用することができます。この LU は、ノードの構成時に自動的に定義されます。これはデフォルトの LU です。アプリケーションが特定の LU を指定していない場合は、この LU を使用できます。アプリケーションがデフォルト LU を使用する場合は、ローカル LU を定義する必要はありません。APPC アプリケーションの資料を確認するか、アプリケーション・プログラマーに連絡してください。

APPC または CPI-C アプリケーションで使用するタイプ 6.2 の従属 LU を構成している場合は、それらをデフォルト・プールのメンバーとして定義することができます。特定のローカル LU を指定しないアプリケーションには、デフォルト LU として定義された LU のプールから、未使用の LU が割り当てられます。

従属 LU 6.2 をデフォルト LU として定義することができます (また、デフォルトの LU をより多くのノードに定義することもできます)。デフォルト LU を要求するアプリケーションは、これらの LU のいずれにも使用できるように割り当てることができます。LU は、アプリケーションと同じコンピューター上に存在する必要はありません。ただし、アプリケーションのパートナー LU を定義する場合、パートナー LU は、デフォルト LU が定義されているすべてのノード上で定義されている必要があります。これにより、アプリケーションは、任意のノードで定義されているデフォルトのローカル LU を使用して、正しいパートナー LU に接続することができます。

独立 APPC および 5250 は独立 LU を使用します。各 LU-LU セッションには、ローカル LU とパートナー LU が関係します。ローカル LU の場合は、ノード制御点に関連付けられている事前定義のデフォルト LU を使用するか、または新しいローカル LU を構成することができます。APPN は、パートナー LU を動的に見つけることができるため、CS Linux ノードが APPN ネットワーク内のエンド・ノードまたはネットワーク・ノードである場合、パートナー LU を構成する必要はありません。ただし、ネットワークが APPN ネットワークでない場合、またはノードが LEN ノードである場合は、パートナー LU を構成する必要があります。この場合は、パートナー LU が常駐するリモート・ノードを構成してから、リモート・ノード上でパートナー LU を定義する必要があります。(パートナー LU がリモート・ノード上のデフォルト LU である場合は、リモート・ノードを定義するときに自動的に追加されるので、それを明示的に定義する必要はありません。)

APPC ローカル LU を構成するには、以下の方法のいずれかを使用します。

Motif または Web 管理プログラム

「ノード」ウィンドウの「運行」メニューの新しい独立したローカル LU 「アプリケーション・アプリケーション」または「新しい従属ローカル LU」。の選択

コマンド行管理プログラム

次のコマンドを発行します。

```
define_local_lu
```

拡張ダイアログを使用して、同期点サポートの指定、ルーティング特性の付加、パスワード置換を使用するかどうか、SSCP アクセスの制限、LU に関連するシステム名、およびセキュリティーを指定することができます。

ローカル LU の構成パラメーター

次のパラメーターは、ローカル LU の構成の必須パラメーターです。

LU 名

ローカル LU の LU 名。

使用する名前が分からないときは、SNA ネットワーク計画担当者に問い合わせてください。

この LU 名は、ローカル LU の完全修飾 LU 名の 2 番目の部分です。完全修飾 LU 名の最初の部分 (ネットワーク名) は、常にローカル・ノードの CP 名の最初の部分と同じです。

LU alias

LU の LU 別名。別名を入力しなかった場合は、LU 名が別名として使用されます。

Host LS/DLUR PU

LU が所属するホスト・リンク・ステーションまたは DLUR PU の名前。(このフィールドは、LU が従属 LU である場合に限り適用されます。)

LU number

従属 LU の LU 番号。(このフィールドは、LU が従属 LU である場合に限り適用されます。)

Member of default pool

LU をデフォルトの従属 APPC LU プールのメンバーにするかどうかを指示します。特定のローカル LU を指定していないアプリケーションには、デフォルトの プール内の使用可能な LU のどれかが割り当てられます。

このフィールドは、LU が従属 LU である場合に限り適用されます。

System name

LU に関連付けられている Remote API Client の DNS コンピューター名。(Allocate を発行するリモート APPC アプリケーションから) 受け取ったすべての LU6.2 接続は、このコンピューター名にルーティングされます。

このフィールドは、構成がクライアント/サーバーの場合にのみ適用されます。

追加構成

ローカル LU の構成が完了したら、次の構成作業を行います。

- リモート・ノードを定義するには、71 ページの『リモートノードの定義』を参照してください。
- パートナー LU を定義するには、72 ページの『パートナー LU の定義』を参照してください。
- 呼び出し可能 TP を定義するには、75 ページの『TP の定義』を参照してください。
- モードを定義するには、80 ページの『モードおよびサービス・クラスの定義』を参照してください。
- CPI-C サイド情報を定義するには、84 ページの『CPI-C サイド情報の定義』を参照してください。
- APPC セキュリティーを定義するには、86 ページの『APPC セキュリティーの構成』を参照してください。
- 5250 通信を構成するには、89 ページの『第 7 章 ユーザー・アプリケーションの定義』を参照してください。

リモートノードの定義

以下の状況では、リモート・ノード (およびノード上のパートナー LU) を定義する必要があります。

- ローカル・ノードが LEN ノードである場合は、すべてのリモート・ノードと、APPC を使用して通信するリモート・ノード上のパートナー LU をすべて定義する必要があります。LEN ノードは、パートナー LU を動的に見つけることができません。リモート・ノード定義を使用すると、パートナー LU を見つけることができます。
- リモート・ノードが LEN ノードであり、ローカル・ノードが LEN ノードのネットワーク・ノード・サーバーとして機能するネットワーク・ノードである場合は、LEN ノード (およびそのパートナー LU) をネットワーク・ノード・サーバー上のリモート・ノードとして定義する必要があります。この定義により、残りの APPN ネットワーク内のノードは LEN ノード上の LU を見つけることができます。
- リモート・ノードが別の APPN ネットワーク内にある場合は、リモート・ノードを動的に配置できないため、リモート・ノードを定義する必要があります。

リモート・ノードを定義する必要があり、リンク・ステーションを定義したときにそれを行わなかった場合は、リンク上で APPC 通信を使用する前に、リモート・ノードを定義する必要があります。

リモート・ノード定義を追加すると、リモート・ノードと同じ名前を持つパートナー LU が自動的に追加されます。これは、リモート・ノードの制御点 LU です。アプリケーションがこのパートナー LU を使用している場合は、パートナー LU に LU 別名を追加することはできますが、別のパートナー LU を追加する必要はありません。別名を追加するには、パートナー LU をダブルクリックして、「パートナー LU 構成」ダイアログに別名を入力します。

ローカル・ノードとリモート・ノードの両方がエンド・ノードまたはネットワーク・ノードであり、APPN ネットワークの一部である場合、パートナー LU は、必要に応じて動的に配置されます。この場合は、LU が配置されているリモート・ノードを定義しないでください。これは、ノードを定義すると、LU を動的に探し出す APPN 内のプロトコルが誤動作する原因になる可能性があるからです。

この誤動作を防止するために、CS Linux では、CP-CP セッションが活動状態になっている（または過去に CP-CP セッションを持っている）リモート・ノードを定義することはできません。さらに、以前にリモート・ノードを定義していて、CS Linux がそのリモート・ノードとの CP-CP セッションを確立している場合には、項目は一時的に動的ノードに変換されます。ノードが非アクティブの場合は、リモート・ノード定義を削除して、障害を修正する必要があります。

リモート・ノードを構成するには、次の方法のいずれかを使用します。

Motif または Web 管理プログラム

「ノード」ウィンドウの「運行」メニューから アプリケーション・アプリケーション と 新規リモート・ノード。 の選択

コマンド行管理プログラム

リモート・ノードを定義するには、次のコマンドを発行します。

```
define_directory_entry
```

パートナー LU を定義するには、次のコマンドを発行します。

```
define_partner_lu
```

リモート・ノードの構成パラメーター

次のパラメーターは、リモート・ノード構成の必須パラメーターです。

Node's SNA network name

リモート・ノードの完全に修飾された CP 名です。このダイアログに入力する値は、リモート・ノードで構成されている CP 名と同じでなければなりません。

追加構成

リモート・ノードの構成が完了したら、次の作業を行います。

- パートナー LU を定義するには、[72 ページの『パートナー LU の定義』](#)を参照してください。
- 呼び出し可能 TP を定義するには、[75 ページの『TP の定義』](#)を参照してください。
- モードを定義するには、[80 ページの『モードおよびサービス・クラスの定義』](#)を参照してください。
- CPI-C サイド情報を定義するには、[84 ページの『CPI-C サイド情報の定義』](#)を参照してください。
- APPC セキュリティーを定義するには、[86 ページの『APPC セキュリティーの構成』](#)を参照してください。
- 5250 通信を構成するには、[89 ページの『第 7 章 ユーザー・アプリケーションの定義』](#)を参照してください。

パートナー LU の定義

ローカル・ノードとリモート・ノードの両方がネットワーク・ノードである場合、または一方がネットワーク・ノードで、もう一方がエンド・ノードで、アプリケーションがパートナー LU を参照するために LU 名を使用している場合、パートナー LU を定義する必要はありません。これは、APPN を使用して動的に配置できるためです。ただし、アプリケーションがパートナー LU を参照するために LU 別名を使用している場合は、パートナー LU 別名定義を追加する必要があります。

ローカル・ノードまたはリモート・ノードのいずれかが LEN ノードである場合は、LEN ノードは LU の動的ロケーションには参加できないので、リモート・ノードの子としてパートナー LU を定義する必要があります。アプリケーションがリモート・ノードの制御点 LU をそのパートナー LU として使用する場合、制御点 LU は、リモート・ノードの定義時に自動的に定義されます。

ワイルドカードを使用して、すべて同じリモート・ノード上にある複数のパートナー LU を構成し、その名前が同じ文字で始まるようにすることができます。ワイルドカードを使用することは、各パートナー LU を個別に構成する必要がないことを意味します。

パートナー LU を構成するには、以下の方法のいずれかを使用します。

Motif または Web 管理プログラム

Motif 管理プログラムを使用して、パートナー LU 別名を追加したり、特定のリモート・ノード上にパートナー LU の定義を追加したり、あるいはワイルドカードを使用して複数のパートナー LU を定義したりすることができます。アプリケーション・アプリケーション、新規パートナー LU、および以下のいずれかが「ノード」ウィンドウの「運行」メニューから表示されます。の選択

- パートナー LU の別名
- リモート・ノード上のパートナー LU
- リモート・ノード上のワイルドカード・パートナー LU

コマンド行管理プログラム

パートナー LU を定義するには、次のコマンドを発行します。

```
define_partner_lu
```

LEN ノードをパートナー LU として定義するには、以下のコマンドを発行します。

```
define_adjacent_len_node
```

```
define_directory_entry
```

パートナー LU の構成パラメーター

次のパラメーターは、パートナー LU 構成の必須パラメーターです。

Partner LU name

パートナー LU の完全修飾 LU 名。この名前は、この LU のリモート・ノードで構成されている名前と同じでなければなりません。該当の名前が分からないときは、SNA ネットワーク計画担当者に問い合わせてください。

このフィールドが適用されるのは、特定のリモート・ノード上にパートナー LU を定義する場合、またはパートナー LU 別名を定義する場合です。

Wildcard partner LU name

複数のパートナー LU の完全修飾 LU 名に一致する名前。(このフィールドが適用されるのは、ワイルドカードを使用してパートナー LU を定義する場合のみです。)ワイルドカード・パートナー LU 名は、1 から 8 文字から成る 2 つのストリングで設定されます。

- 最初のストリングには、パートナー LU の完全修飾 LU 名の最初の部分に正確に一致する完全な SNA ネットワーク名か、または、パートナー LU のネットワーク名の先頭部分に一致するワイルドカード接頭部を指定できます。最初のストリングの値としてワイルドカード接頭部を指定する場合は、2 番目のストリングはブランクのままにします。例えば、A というワイルドカード・エントリーは、SNA ネットワーク内の A、ANT、または APPN などの名前のすべての LU に一致します (BUFFALO や ZEBRA などには一致しません)。
- 最初のストリングに完全な SNA ネットワーク名を指定する場合は、2 番目のストリングの値も入力できます。(最初のストリングに有効な SNA ネットワーク名を指定しない限り、2 番目のストリングは指定できません。)2 番目のストリングはワイルドカード接頭部として処理されます。この接頭部は、パートナー LU の完全修飾名の後半の部分の先頭に一致しなければなりません。例えば、A.F というワイルドカード・エントリーは、A.FRED や A.FREDDY などのパートナー LU 名に一致します (APPN.FRED や A.B などには一致しません)。

ストリングを両方ともブランクにした場合は、ワイルドカード・パートナー LU 定義はすべての LU 名に一致することになります。

Alias

ローカルで表示できる、パートナー LU の別名。LU 別名を使用してパートナー LU を参照するローカル・アプリケーションがない場合は、LU 別名を指定する必要はありません。

このフィールドが適用されるのは、特定のリモート・ノード上にパートナー LU を定義する場合、またはパートナー LU 別名を定義する場合です。

Uninterpreted Name

従属ローカル LU が、パートナー LU とローカル LU の間の LU-LU セッションの開始をホストに要求するとき使用する非解釈名。これにより、ホストで構成されているパートナー LU 名とは異なる名前を、ローカルで構成する (そしてアプリケーションで使用する) パートナー LU 名として使用できます。

デフォルトの非解釈名は、パートナー LU 名の 2 番目の部分です。ほとんどの場合は、これを使用すれば間違いはありません。疑わしいときは、SNA ネットワーク計画担当者に問い合わせてください。

このフィールドが適用されるのは、特定のリモート・ノード上にパートナー LU を定義する場合、またはパートナー LU 別名を定義する場合です。

Supports parallel sessions

パートナー LU が同時に複数のセッションをサポートするかどうかを指示します。ほとんどの場合、パートナー LU は同時に複数のセッションをサポートしますが、一部の LEN ノードでは並列セッションはサポートされません。

このフィールドが適用されるのは、特定のリモート・ノード上にパートナー LU を定義する場合、またはパートナー LU 別名を定義する場合です。

Location

パートナー LU が存在するノード、またはパートナー LU へのアクセスを提供するノードにある CP の完全修飾名。未定義のリモート・ノードの名前を指定した場合、そのノードを動的に検索できない場合にはそのノードを定義する必要があります。

このフィールドが適用されるのは、特定のリモート・ノード上にパートナー LU を定義する場合のみです。

パートナー LU のリンク・ステーション・ルーティングの定義

リンク・ステーション・ルーティングを使用して、リンク・ステーションに到達するために使用されるリンク・ステーションによってパートナー LU の位置を識別することができます。

注:

1. リンク・ステーション経路指定は、リソースが動的に位置指定できる APPN ネットワーク内では必要ありません。APPN ネットワークでは、通常の APPN ルーティング・メカニズムをバイパスするため、リンク・ステーション・ルーティングを使用することはお勧めできません。
2. Enterprise Extender (HPR/IP) リンク・ステーションとのリンク・ステーション・ルーティングを使用することはできません。これは、このリンク・タイプのすべてのトラフィックが RTP 接続を経由して流れている必要があるためです。RTP 接続は、特定のリンク・ステーションに固定されておらず、別のパスに切り替えることができます。

パートナー LU のリンク・ステーション・ルーティングを構成するには、以下の方法のいずれかを使用します。

Motif または Web 管理プログラム

アプリケーション・アプリケーション、新規パートナー LU、and リンク・ステーション上のパートナー LU from the 運行 menu on the Node window. の選択

コマンド行管理プログラム

次のコマンドを発行します。

ルーティングのデフラグ

リンク・ステーション・ルーティング・パラメーター

次のパラメーターは、リンク・ステーション・ルーティング構成の必須パラメーターです。

LU 名

リンク・ステーションを制御するローカル LU の名前 (特定のリンク・ステーションによりパートナー LU を見つける場合)。

LS name

リンク・ステーションの名前。

Partner LU name

パートナー LU の完全修飾 LU 名、またはワイルドカード名。

- 完全修飾 LU 名は、1 から 8 文字から成る 2 つのストリングで設定されます。

この名前は、この LU のリモート・ノードで構成されている名前と同じでなければなりません。該当の名前が分からないときは、SNA ネットワーク計画担当者に問い合わせてください。

- ワイルドカード・パートナー LU 名は、複数のパートナー LU の完全修飾 LU 名に一致します。ワイルドカード・パートナー LU 名は、1 から 8 文字から成る 2 つのストリングで設定されます。
 - 最初のストリングには、パートナー LU の完全修飾 LU 名の最初の部分に正確に一致する完全な SNA ネットワーク名か、または、パートナー LU のネットワーク名の先頭部分に一致するワイルドカード接頭部を指定できます。最初のストリングの値としてワイルドカード接頭部を指定する場合は、2 番目のストリングはブランクのままにします。例えば、A というワイルドカード・エントリーは、SNA ネットワーク内の A、ANT、または APPN などの名前のすべての LU に一致します (BUFFALO や ZEBRA などには一致しません)。
 - 最初のストリングに完全な SNA ネットワーク名を指定する場合は、2 番目のストリングの値も入力できます。(最初のストリングに有効な SNA ネットワーク名を指定しない限り、2 番目のストリングは指定できません。) 2 番目のストリングはワイルドカード接頭部として処理されます。この接頭部は、パートナー LU の完全修飾名の後半の部分の先頭に一致しなければなりません。例えば、A.F というワイルドカード・エントリーは、A.FRED や A.FREDDY などのパートナー LU 名に一致します (APPN.FRED や A.B などには一致しません)。

ストリングを両方ともブランクにした場合は、ワイルドカード・パートナー LU 定義はすべての LU 名に一致することになります。

Use partner LU name as a wildcard

パートナー LU 名を、リテラルの完全修飾 LU 名ではなくワイルドカードとして使用するかどうかを指示します。

追加構成

パートナー LU の構成が完了したら、次の構成作業を行います。

- 呼び出し可能 TP を定義するには、[75 ページの『TP の定義』](#)を参照してください。
- モードを定義するには、[80 ページの『モードおよびサービス・クラスの定義』](#)を参照してください。
- CPI-C サイド情報を定義するには、[84 ページの『CPI-C サイド情報の定義』](#)を参照してください。
- APPC セキュリティーを定義するには、[86 ページの『APPC セキュリティーの構成』](#)を参照してください。
- 5250 通信を構成するには、[89 ページの『第 7 章 ユーザー・アプリケーションの定義』](#)を参照してください。

TP の定義

このセクションでは、APPC TP の定義方法を説明します。

ほとんどの場合、CS Linux システムで実行される TP を定義する必要はありませんが、以下の場合には TP 定義を構成する必要があります。

APPC の特性

CS Linux コンピューター上の TP が呼び出し元 TP である場合(またはソース TP - APPC 会話を開始する TP)で、TP へのアクセスを制限する必要がない場合は、TP を定義する必要はありません。ただし、[79 ページの『TP 定義パラメーター』](#)で説明されているように、APPC TP を定義して、以下の特性を指定することができます。

- TP の会話セキュリティを定義します。
- TP が基本会話またはマップ式会話を使用するかどうかを示す
- 同期点処理を指定します。
- PIP データの処理を指定します。

呼び出し可能 TP

着信割り振り要求にตอบสนองして TP を自動的に開始できるようにするには、77 ページの『サーバー上で
の TP 呼び出しパラメーター』で説明されているように、呼び出し可能 TP として TP を定義します。

呼び出し可能 TP (またはターゲット TP) 着信割り振り要求に対する応答として開始されるものです。
呼び出し可能 TP 用の TP 定義を作成する必要があります。呼び出し可能 TP は、RECEIVE_ALLOCATE
を発行する APPC TP、または Accept_Conversation を発行する CPI-C アプリケーション、または
Accept_Incoming とすることができます。

注: 本書では、"受信側割り振り" という句を使用して、これらの 3 つの API 呼び出しのいずれかを示
します。

また、呼び出し可能 TP を定義して、着信割り振り要求を実行中の TP に経路指定することもできます。

呼び出し可能 TP の場合は、タイムアウト値を指定して、割り振り要求の待機を制限することもできま
す。(このオプションは、コマンド行管理を使用してのみ構成できます。)

CS Linux では、以下の目的で呼び出し可能 TP 定義が使用されます。

- TP が Receive_Allocate を発行すると、CS Linux は、適切な TP 名を持つ呼び出し可能 TP 定義を検索
します。定義が存在し、かつ Receive_Allocate タイムアウトの値が含まれている場合、CS Linux は
Receive_Allocate; の処理時にこの値を使用します。それ以外の場合は、デフォルトを使用します(タ
イムアウトなし、TP は無期限に待機することになります)。
- 着信割り振り要求がターゲット・システムに到着し、要求された TP がまだ未処理の Receive_Allocate
を使用して実行されていない場合、CS Linux は、着信割り振り指定された TP 名を持つ TP 定義を
検索します。定義が存在する場合、CS Linux は、この定義内の情報を使用して TP を開始します(複
数のインスタンスが許可されている場合、または TP がまだ実行されていない場合)。または、着信
割り振りをキューに入れる必要があるかどうかを判別します(TP が既に実行中で、複数のインスタン
スが許可されていない場合)。

必要な場合は、同じ TP に対して両方のタイプの定義を構成することができます(例えば、呼び出し可能 TP
用の会話セキュリティーを定義するため)。

TP 定義を構成するには、以下の方法のいずれかを使用します。

APPC 特性を定義するには

次のいずれかの方法を使用します。

Motif または Web 管理プログラム

Select アプリケーション・アプリケーション and トランザクション・プログラム from the 運行
menu on the Node window. CS Linux が「TP」ウィンドウを表示する場合は、下部ペインを選択し
て「ニュー」ボタンをクリックするか、既存の TP 定義を選択して、「性質」ボタンをクリックし
ます。

コマンド行管理プログラム

snaadmin define_tp コマンドを発行する

呼び出し可能 TP を定義する場合:

サーバーおよびクライアントの構成メソッドは、以下のように異なります。

- サーバーでは、以下のいずれかの方法を使用します。

Motif または Web 管理プログラム

Select アプリケーション・アプリケーション and トランザクション・プログラム from the 運行
menu on the Node window. CS Linux で「TP」ウィンドウが表示されたら、上部ペインを選択し
て「ニュー」ボタンをクリックするか、既存の呼び出し可能 TP 定義を選択し、「性質」ボタ
ンをクリックします。

コマンド行管理

snatpinstall コマンドを発行する

- 

AIX または Linux 上の IBM リモート API クライアントで、以下を実行します snatpinstall コマン
ド

WINDOWS

Windows クライアントでは、クライアント・ソフトウェアがインストールされているディレクトリーに移動して、`tpinst32` コマンドを発行します。(このコマンドは、32 ビット版と x64 バージョンの Windows の両方に適用されます。)

`snatpinstall` または `tpinst32` コマンドを参照してください。163 ページの『付録 B コマンド行からの呼び出し可能 TP の構成』の使用については、

サーバー上での TP 呼び出しパラメーター

ここでは、サーバー上で呼び出し可能 TP を構成する際に、Motif 管理プログラムまたはコマンド行管理プログラムに指定する必要があるパラメーターについて説明します。クライアント上での呼び出し可能 TP の構成については、163 ページの『付録 B コマンド行からの呼び出し可能 TP の構成』を参照してください。

次のパラメーターは、ローカル・ノードで呼び出せる TP のための必須パラメーターです。

TP name

次のいずれかの形式の TP 名。

Application TP

TP がユーザー・アプリケーションである場合は、通常の文字で名前を指定します (長さは最大 64 文字)。

Service TP

TP が SNA サービス・トランザクション・プログラムである場合は、16 進数で名前を入力します (最大 8 桁の 16 進数、つまり 4 バイト)。

同じ TP 名を持つ複数の APPC 呼び出し可能 TP を定義することもできます。ただし、その場合は、各 TP 定義でそれぞれ異なる LU 別名を指定する必要があります。CPI-C 呼び出し可能 TP に対してはこれを行うことはできません。それは、使用する特定の LU 別名が指定できないためです。各 CPI-C 呼び出し可能 TP は名前が異なっている必要があります。

Parameters are for invocation on any LU/on specific LU

TP が APPC TP の場合、このパラメーターは、任意の LU で TP を呼び出し可能にするか、または特定の LU でのみ呼び出し可能にするかを指定します。デフォルトでは、TP は任意の LU で呼び出し可能です。

注: TP が CPI-C アプリケーションの場合、このフィールドは、任意の LU で TP が呼び出し可能であるよう設定されている必要があります。CPI-C は、特定のローカル LU からの着呼接続の受け入れをサポートしていないため、CPI-C アプリケーションに対してこのオプションを指定すると、TP への着呼接続のルーティングにエラーが生じることがあります。

LU alias

TP が CPI-C アプリケーションである場合、このフィールドは、使用してはなりません。TP が APPC アプリケーションである場合、この TP 定義のパラメーターが任意の LU での呼び出し用であることを指定した場合のみ、このフィールドが適用されます。

TP がどこからの着呼接続を受け入れるかをローカル LU 別名で指定します。この名前は、CS Linux ノード上のローカル APPC LU の名前と同じでなければなりません。LU 別名を指定しなかった場合は、TP はどのローカル LU からも着呼接続を受け入れます。

非ブランクの LU 別名が指定されている場合、TP は RECEIVE_ALLOCATE verb の拡張フォームを使用し、その verb へのパラメーターとしてこの LU 別名を指定する必要があります。これにより、CS Linux は、着信接続を正しい TP に送ることができるようになります。さまざまな形式の RECEIVE_ALLOCATE について詳しくは、「IBM Communications Server for Data Center Deployment on AIX or Linux APPC プログラマーズ・ガイド」を参照してください。アプリケーションに LU 別名を事前に組み込まないで、TP が実行時に正しい LU 別名を判別できるようにする場合は、該当の LU 別名が入る環境変数を設定し (Environment パラメーターを使用)、RECEIVE_ALLOCATE の発行方法を決定するための環境変数をアプリケーションが読み込むように設計します。

同じ TP 名を持つ複数の TP を定義することもできます。ただし、その場合は、各 TP 定義でそれぞれ異なる LU 別名を指定する必要があります。

Multiple instances supported

このオプションを選択しなかった場合は、TP は待機 TP となります。この TP の実行中に到着した着呼割り振り要求は、TP が新しい `Receive_Allocate` を発行するか、または TP の実行が終わって再始動が可能になるまで、待機状態になります。着信割り振り要求がこの TP に送付されるのは、着信割り振り要求をこのコンピューターに送付するように構成されている LU がその要求を受信した場合、または、このコンピューター上にあってルーティング情報が構成されていない LU がその要求を受信した場合だけです。

このオプションを選択した場合は、TP は非待機 TP になります。CS Linux は、この TP を宛先とする着呼割り振り要求が到着するたびに、この TP の新しいコピーを開始します。非待機 TP はオペレーターが開始することはできません。常に CS Linux により自動的に開始されます。非待機 TP の場合は、CS Linux は TP の複数のコピーを同時に実行できます。すべてのコピーは、「`User ID`」および「`Group ID`」パラメーターに定義されている、同じユーザー ID と同じグループ ID、および同じ作業ディレクトリーを使用して実行されます。TP がローカル・システムのファイルへの書き込みをする場合は、TP の複数のコピーが互いに相手のファイルへ上書きしないように注意する必要があります。

非待機の TP が会話を終了したあとで、TP を終了させるか、または別の `RECEIVE_ALLOCATE` を発行することができます。頻繁に使用されるプログラムについては、各会話でプログラムの新しいインスタンスを開始するという効率のオーバーヘッドを回避できます。非待機の自動開始済み TP に対する接続を受信するたびに、CS Linux は、この TP のインスタンスから出された未処理の `RECEIVE_ALLOCATE` が既に存在するかどうか検査します。未解決がある場合、この TP は着呼の会話に使用されます。そうでない場合は、CS Linux はプログラムの新しいインスタンスを開始します。

Route incoming Allocates to running TP

このオプションは、複数のインスタンスがサポートされない場合にのみ適用されます。

TP がブロードキャスト待機 TP である場合は、このオプションを選択します。この TP の実行中に到着した着呼割り振り要求は、TP が新しい `Receive_Allocate` を発行するか、または TP の実行が終わって再始動が可能になるまで、待機状態になります。この TP が開始されると、この TP に関する情報が LAN 上のすべてのサーバーにブロードキャストされます。別のコンピューター上の LU が着信割り振り要求を受信した場合に、ルーティング情報が構成されていないときは、その LU は動的にこの TP を見つけて、それに対して割り振り要求を送付します。

このオプションを使用すると、LU のルーティング情報を明示的に構成する必要がなくなり、また同じ TP の複数のコピーを別々のコンピューター上で実行することで負荷のバランスを取ることができます。ただし、LAN トラフィックを減らすために情報のブロードキャストを避けたい場合、または、特定の LU に到着した着信割り振り要求が常に TP の同じコピーに送付されるようにしたい場合は、このオプションを選択しないでください。

Full path to TP executable

この TP 用の実行可能ファイルの絶対パスとファイル名。

このファイルは、「`User ID`」パラメーターに指定されたユーザーに対する実行許可を持つものでなければなりません。さらに、`User ID` を `root` に設定してこの実行可能ファイルを実行する場合は、このファイルは `root` が所有しているものでなければならず、また、CS Linux がこのファイルを自動開始できるようにするには、`setuid` 許可および `setgid` 許可が設定されていることも必要です。

Arguments

TP に渡すコマンド行引数を、スペースで区切って指定します。これらの引数は、ここで入力したのと同じ順序で TP に渡されます。

この値はオプションです。これを省略した場合は、コマンド行引数なしで TP が呼び出されます。

User ID

TP を開始するために CS Linux が使用するユーザー ID。この行は必須であり、指定する必要があります。この ID は、CS Linux コンピューター上の有効な Linux ログイン ID でなければなりません。

TP は、このユーザー ID に関連したホーム・ディレクトリーの中で開始されます。このホーム・ディレクトリーは、トレース・ファイル、および TP がアクセスするその他のファイルのデフォルトのパスでもあります (アプリケーションで、絶対パスの指定により上書きされた場合を除きます)。アプリケ

ーションで、パスなしのファイル名が指定された場合は、CS Linux はこのホーム・ディレクトリーの中でそのファイルを検索します。アプリケーションで、ファイル名と相対パスが指定されている場合は、CS Linux は、このホーム・ディレクトリーを基準として相対的に指定されたディレクトリーの中で、そのファイルを検索します。

ここで指定するユーザーには、「*Full path to TP executable*」パラメーターで指定された実行可能ファイルの実行許可が与えられていることが必要です。さらに、*User ID* を *root* に設定する場合は、ファイルは *root* が所有しているものでなければならず、また、CS Linux がこのファイルを自動開始できるようにするには、*setuid* 許可および *setgid* 許可が設定されていることも必要です。

Group ID

TP を開始するために CS Linux が使用するグループ ID。この ID は、CS Linux コンピューター上の有効な Linux グループ ID でなければなりません。

このパラメーターはオプションです。これが含まれていない場合のデフォルトは *sna* です。

Standard input

TP が使用する標準入力ファイルまたはデバイスの絶対パス名を指定します。

このパラメーターはオプションです。これが含まれていない場合のデフォルトは */dev/null* です。

Standard output

TP が使用する標準出力ファイルまたはデバイスの絶対パス名を指定します。

このパラメーターはオプションです。これが含まれていない場合のデフォルトは */dev/null* です。

Standard error

TP が使用する標準エラー・ファイルまたはデバイスの絶対パス名を指定します。

このパラメーターはオプションです。これが含まれていない場合のデフォルトは */dev/null* です。

Environment

TP が必要とする環境変数を指定します。

各変数は、*environment_variable=value* の形式で指定し、最大長は 255 文字です。

environment_variable=value のストリングの中で、= 文字の前後にスペースまたはタブ文字を使用しないでください。

Motif 管理プログラム内で、複数の環境変数 (最大数 64) を指定する必要がある場合は、| 文字を使用してこれらの変数を区切ります。変数は、ここでの入力と同じ順序で設定されます。

TP が CPI-C アプリケーションである場合は、このフィールドを使用して環境変数 *APPCLLU* を設定できないので注意してください。自動的にロードされた CPI-C アプリケーションに対して、ローカル *LU* を指定することはできません。

このフィールドはオプションです。これを省略した場合、環境変数は使用されません。

TP 定義パラメーター

APPC TP を構成することにより、会話セキュリティー、会話タイプ、同期レベル、および、PIP データの取り扱いを指定できます。次のパラメーターは、APPC 通信用の TP を定義するための必須パラメーターです。

TP name

次のいずれかの形式の TP 名。

Application TP

TP がユーザー・アプリケーションである場合は、通常の文字で名前を指定します (長さは最大 64 文字)。

Service TP

TP が SNA サービス・トランザクション・プログラムである場合は、16 進数で名前を入力します (最大 8 桁の 16 進数、つまり 4 バイト)。

Conversation level security required

割り振り要求に、有効なユーザー名およびパスワード (またはパスワードが既に検査済みであることを示すインディケーター) が含まれていなければならない場合は、このオプションを選択します。このオプションを選択しなかった場合は、検査は不要とみなされます。

Restrict access

ユーザー名がセキュリティー・アクセス・リストに含まれていなければならない場合は、このオプションを選択します。このフィールドが適用されるのは、「Conversation level security required」オプションを選択した場合のみです。

Security access list

この TP へのアクセスを許可されているユーザー ID を含むセキュリティー・アクセス・リストの名前。「Restrict access」オプションを選択した場合は、この値を指定する必要があります。

Conversation type

TP が、基本会話のみ、マップ式会話のみ、または両方のタイプの会話を受け入れるかを指定します。

Sync level

TP が受け入れる確認同期のレベルを指定します。確認同期について詳しくは、「IBM Communications Server for Data Center Deployment on AIX or Linux APPC プログラマーズ・ガイド」を参照してください。次のいずれかの値を選択します。

- None
- Confirm
- Sync-point
- None または Confirm
- None、Confirm、または Sync-point

PIP allowed

TP が PIP データ (プログラム初期設定パラメーター (PIP)) を受け入れる場合はこのオプションを選択します。

モードおよびサービス・クラスの定義

A モードローカル LU (LU タイプ 6.2) がパートナー LU と通信するために使用する特性のセットを指定します。これらの特性には、2 つの LU 間でのデータ伝送方法に関する情報 (RU の最大長さやペーシング・ウィンドウ・サイズなど)、および LU が並列セッションを確立できるかどうかに関する情報が含まれます。

さらに、特定のレベルのネットワーク・セキュリティーの実施、伝送時間の最小化、または高価な通信リンクの使用を回避するなど、LU 間の通信パスの要件を指定する必要がある場合があります。これらの要件は、サービス・クラス (COS) を使用して定義できます。これは、伝送時間、伝送コスト、ネットワーク・セキュリティーなどの特性の許容値の最小値と最大値を指定します。また、COS はこれらの値の異なる範囲に関連する重み付けを指定する。これにより、同じリモート LU への 2 つ以上の経路が使用可能になったときに、ノードはネットワーク全体の最良の経路を計算することができます。

CS Linux ノードがネットワーク・ノードである場合、各モードの定義には、そのモードに必要な COS の名前が含まれます。CS Linux ノードが LEN ノードまたはエンド・ノードである場合は、COS をモードと関連付ける必要はありません。COS 名は動的に決定されます。

SNA は、ほとんどのシステムの要件をカバーする標準モードおよび関連する COS の数を定義します。通常は、追加のモードと COS を定義する必要はありません。モードを定義する必要があるのは、必要なモードが事前定義の標準モードではなく、「モード」ウィンドウで表示されるモードの場合のみです。

着信会話内のモード名が認識されない場合は、デフォルト・モードが使用されます。デフォルト・モードを指定しない場合、デフォルト・モードはブランク・モード名になります。

標準モード名とそれに関連する COS 名は、[81 ページの表 2](#) に示されています。これらの標準名に関連付けられているパラメーターについて詳しくは、[IBM SNA マニュアル LU 6.2 参照 - ピア・プロトコル \(モードの場合\)](#) および [APPN アーキテクチャー参照 \(COS の場合\)](#) を参照してください。

表 2. 標準モードおよび COS 名

モード名	関連付けられた COS 名	目的
(ブランク)	#CONNECT	モード名を指定しないセッション (基本デフォルト COS パラメーター)
#BATCH	#BATCH	バッチ処理アプリケーションで使用されるセッション
#BATCHSC	#BATCHSC	最小レベルのルーティング・セキュリティを持つ、バッチ処理アプリケーションによって使用されるセッション
#BATCHC	#BATCH	バッチ処理アプリケーションで圧縮を使用するセッション
#BATCHCS	#BATCH	バッチ処理アプリケーションで圧縮を使用するセッション (最小レベルのルーティング・セキュリティを使用)
#INTER	#INTER	対話式アプリケーションが使用するセッション
#INTERSC	#INTERSC	対話式アプリケーションによって使用されるセッション (最小レベルのルーティング・セキュリティ)
#INTERC	#INTER	対話式アプリケーションで圧縮を使用するセッション
#INTERCS	#INTER	対話式アプリケーションで圧縮を使用するセッション (最小レベルのルーティング・セキュリティ)
スナスフコマン	スナスフコマン	CNOS (セッション数の変更) および管理サービス・セッション
CPSVCMG	CPSVCMG	ノード間の CP-CP セッション
CPSVRMGR	CPSVRMGR	従属 LU リクエスター (DLUR) に使用される CP-CP セッション
QPCSUPP	#CONNECT	5250 エミュレーションに使用されるセッション

モードが構成されると、任意の APPC アプリケーションまたは CPI-C アプリケーションで使用して、ローカル LU とパートナー LU の間のセッションを活動化することができます。APPC アプリケーションは使用するモードを指定する必要がありますが、CPI-C アプリケーションは CPI-C サイド情報 (モード名を含む) を使用することができます。CPI-C サイド情報の構成について詳しくは、[84 ページの『CPI-C サイド情報の定義』](#)を参照してください。

モードまたはサービス・クラスを構成するには、以下の方法のいずれかを使用します。

Motif または Web 管理プログラム

アプリケーション・アプリケーション and モーデス from the 運行 menu on the Node window, then choose ニュー on the Mode window. の選択

コマンド行管理プログラム

モードを定義するには、次のコマンドを発行します。

```
define_mode
```

デフォルト・モードを変更するには、次のコマンドを発行します。

```
define_defaults
```

サービス・クラスを定義するには、次のコマンドを発行します。

```
define_cos
```

モードの構成パラメーター

次のパラメーターはモード構成の必須パラメーターです。

Name

定義するモードの名前。モード名は1から8文字のストリングです。

このモードを使用する APPC アプリケーション (ローカル・アプリケーションおよびリモート・アプリケーションの両方を含む) でも、この名前が使用されることがあるため、アプリケーション開発者に問い合わせるこの名前をチェックしてください (または、サード・パーティー・アプリケーションの場合は、該当製品の資料を参照してください)。

COS name

このモードのサービス・クラスの名前。この名前は1から8文字のストリングです。通常ここに指定するのは、対話型データ交換に使用するモードを表す #INTER か、または大量データ転送に使用するモードを表す #BATCH です。

このフィールドはネットワーク・ノードのみに適用されます。

指定する値が分からないときは、SNA ネットワーク計画担当者にお問い合わせください。

セッション限度

次のフィールドを使用してセッション限度を指定します。

Initial session limit

CNOS を使用して別個に最大セッション数の折衝が行われていない場合に、一对の LU がこのモードを使用して持つことのできる最大セッション数 (最大セッション限度までの範囲内)。

通常は、このフィールドの値には8を指定します。確信がないときは、SNA ネットワーク計画担当者または APPC アプリケーション開発者に問い合わせてください (または、サード・パーティー・アプリケーションの場合は製品の資料を参照してください)。

Maximum session limit

CNOS 折衝の有無に関係なく、このモードを使用する一对の LU 間に許される最大セッション数 (最高 32,767)。

通常は、このフィールドは初期セッション限度と同じ値に設定されます。確信がないときは、SNA ネットワーク計画担当者または APPC アプリケーション開発者に問い合わせてください (または、サード・パーティー・アプリケーションの場合は製品の資料を参照してください)。

Minimum contention winner sessions

コンテンション勝者としてのローカル LU が使用するために CS Linux が確保する必要があるセッション数 (セッション限度までの範囲内)。

通常は、このフィールドは0に設定しておけば安全ですが、確信がない場合は、SNA ネットワーク計画担当者にお問い合わせください。

コンテンション勝者セッションの最小数とコンテンション敗者セッションの最小数の合計が、初期セッション限度を超えてはなりません。

Minimum contention loser sessions

コンテンション敗者としてのローカル LU が使用するために CS Linux が確保する必要がある最小セッション数。この値は、「Minimum contention winner sessions」フィールドの値と共に、セッションの競合の解決方法を決定するために使用されます。

通常は、この値は0に設定しておけば安全ですが、確信がない場合は、SNA ネットワーク計画担当者にお問い合わせください。

コンテンション勝者セッションの最小数とコンテンション敗者セッションの最小数の合計が、初期セッション限度を超えてはなりません。

Auto-activated sessions

このモードを使用するローカル LU とパートナー LU の間のセッションに関する CNOS 折衝が行われたあとで、自動的に開始されるセッション数 (コンテンツン勝者セッションの最小数の範囲内)。このフィールドに値を指定すると、このモードを使用する LU は、会話の即時割り振りを求める TP からの要求にこたえて、自動的にセッションを開始できます。

受信ペーシング・ウィンドウ

これらのフィールドには、SNA ペーシング応答を送信する前に受信できる RU の数を指定します。

Initial window size

ローカル LU が受け取ることができる要求単位 (RU) の数の初期設定です。この数を超えると、ローカル LU はリモート LU にペーシング応答を送信することが必要になります。これは 4 に設定しておけば安全です。

状況によっては、この値をもっと大きくするとパフォーマンスが向上することがありますが、代わりにメモリー使用量が増加します。

Maximum window size

ローカル LU が受け取ることができる要求単位 (RU) の最大数です。この数を超えると、ローカル LU はリモート LU にペーシング応答を送信することが必要になります。

この値はオプションです。これを指定しなかった場合は、最大受信ペーシング・ウィンドウのサイズは無制限となります。この値を指定した場合は、適応ペーシング用の受信ペーシング・ウィンドウのサイズが、この値に基づいて制限されます。適応ペーシングが使用されていない場合は、この値は無視されます。

ペーシング・ウィンドウのサイズの範囲は、0 から 32767 バイトです。0 の値は無制限のウィンドウを意味します。

隣接ノードが固定ペーシングのみをサポートしている場合は、これらの値によって固定ペーシング・ウィンドウ・サイズが決まります。ただし、隣接ノードは、折衝により別のウィンドウ・サイズを設定することもできます。隣接ノードが適応ペーシングを使用している場合は、これらの値により初期ウィンドウ・サイズが設定されます。

Specify timeout

このモードを使用する LU 6.2 セッションでタイムアウトが生じるまでの非アクティブ状態の秒数 (0 から 65535) を指定する場合は、このオプションを選択します。この値の変更による影響を受けるのは、この定義を使用して開始されるセッションだけです (既にアクティブになっているセッションは影響を受けません)。

0 の値を指定した場合は、セッションは解放されると同時にタイムアウトになります。

Restrict maximum RU size

パートナー LU に送信するデータをどれだけバッファに入れることができるのかを決定する、最大 RU サイズを指定する場合、このオプションを選択します。

上限は、256 から 62440 バイトの範囲内です。上限は 1024 バイトに設定しておけば安全です。状況によっては、この値をもっと大きくするとパフォーマンスが向上することがありますが、代わりにメモリー使用量が増加します。

下限は、0 か、または 256 から指定した上限までの範囲内の値です。

このフィールドの値がリモート・ノード用に定義された RU サイズと異なる場合は、そのノードとのセッションに使用するサイズを折衝し、セッションのための適切な RU サイズを確立できます。この実際の値を、下限フィールドの値より小さくすることはできません。

これらの値を、送受信ペーシング値と一緒に使用することにより、ローカル LU とパートナー LU の間のセッション・レベルのスループットを調整できます。どの値を使用すべきか分からない場合は、まずデフォルト値から始めて、必要に応じて、最大限のスループットが得られるように調整してください。

Compression supported

このモードを使用するセッションに対して、データ圧縮がサポートされるかどうか。このオプションを設定しない場合、圧縮は使用されません。

このオプションを設定すると、インバウンド・データおよびアウトバウンド・データに使用される最大圧縮レベルを指定できます。これらのレベルは別々のオプションなので、2つの方向で異なるレベルを指定したり、片方向では圧縮を使用して片方向では使用しないように指定したりできます。それぞれの方向で、圧縮を行わない場合は「なし (None)」、または「RLE」(最小圧縮)、「LZ9」、「LZ10」(最大圧縮)のいずれかの値を選択できます。

Reset to SNA defined values

Motif ダイアログを使用して標準モードを変更している場合は、このボタンをクリックすることにより、モード・パラメーターの値を SNA 定義の値にリセットできます。

追加構成

モードの構成が完了したら、次の構成作業を行います。

- CPI-C サイド情報を定義するには、84 ページの『CPI-C サイド情報の定義』を参照してください。
- APPC セキュリティーを定義するには、86 ページの『APPC セキュリティーの構成』を参照してください。
- 5250 通信を構成するには、89 ページの『第 7 章 ユーザー・アプリケーションの定義』を参照してください。

CPI-C サイド情報の定義

CPI-C 記号宛先名を使用する CPI-C アプリケーションをサポートしている場合は、CPI-C サイド情報を定義する必要があります。サイド情報は、シンボリック宛先名と、会話のパートナー TP、パートナー LU、モード、およびセキュリティーに関する情報を関連付けます。

CPI-C のシンボリック宛先名を判別するには、アプリケーション開発者(またはサード・パーティー・アプリケーションの場合は、製品資料を参照してください)を参照してください。

CPI-C サイド情報を構成するには、以下の方法のいずれかを使用します。

Motif または Web 管理プログラム

「ノード」ウィンドウの「運行」メニューからアプリケーション・アプリケーションと CPI-C。の選択

コマンド行管理プログラム

次のコマンドを発行します。

キャッシュ・サイド情報のデフラグ

CPI-C の構成パラメーター

アプリケーションで使用するそれぞれの CPI-C シンボリック宛先名ごとに、次の情報を収集します。

Name

実行したい CPI-C アプリケーション (TP と呼ばれます) で使用するシンボリック宛先名。この名前前の長さは 1 から 8 文字です。

この名前が分からないときは、アプリケーション開発者に問い合わせてください(または、サード・パーティー・アプリケーションの場合は製品の資料を参照してください)。

Local LU

このサイド情報を使用する TP が、次のいずれかの方法を使用して開始する会話のローカル LU。

Local LU alias

ローカル LU の別名。

Use default LU

このオプションは、デフォルト・プールのメンバーを使用するか(デフォルト・プールがある場合)、またはノード制御点 LU を使用するよう(デフォルト・プールが定義されていない場合)指定します。

APPCLLU 環境変数が設定されている場合は、ここで指定する ローカル LU 情報は無視され、この環境変数に指定された LU が代わりに使用されます。

Partner LU

このサイド情報を使用するローカル TP が開始する会話に使用される 別名またはパートナー LU の完全修飾 LU 名。パートナー LU は、パートナー TP を実行するコンピューターで構成されている LU でなければなりません。

Mode

パートナー LU にアクセスするために使用する APPC モードの名前。ほとんどの場合、モードは次の定義済みモードのいずれかです。

- ブランク名
- #BATCH
- #BATCHSC
- #INTER
- #INTERSC
- QPCSUPP

Partner TP

CPI-C アプリケーションが通信する相手方のトランザクション・プログラムの名前。

- TP がユーザー・アプリケーションである場合は、通常の文字で名前を指定します (長さは最大 64 文字)。
- TP がサービス TP である場合は、16 進数で名前を入力します (最大 8 個の 16 進数字、つまり 4 バイト)。

この情報は、アプリケーション開発者 (または、サード・パーティー・アプリケーションの場合は製品の資料) から入手できます。

Security

使用したい会話レベル・セキュリティーのレベル。これには次のオプションがあります。

None

パートナー TP は、セキュリティー・パラメーターの検査を必要としません。

Same

パートナー TP はセキュリティーを使用しますが、開始側 TP が提供するユーザー ID およびパスワードに対するローカル TP の検査を受け入れます。security level として「Same」を選択した場合は、パートナー TP が受け入れる有効なユーザー ID も指定する必要があります。

Program

パートナー TP がユーザー ID およびパスワードを必要としています。security level として「Program」を指定した場合は、パートナー TP が受け入れる有効なユーザー ID およびパスワードを指定する必要があります。

Program strong

パートナー TP がユーザー ID およびパスワードを必要としています。パスワードが暗号化されるようにするために、ローカル・ノードとリモート・ノードの両方が拡張セキュリティーをサポートしていることが必要です。

使用するセキュリティー・パラメーターについては、CPI-C アプリケーション 資料を参照するか、アプリケーション・プログラマーに問い合わせてください。

User ID

security level として、「Same」、「Program」、「Program strong」のいずれかを選択した場合は、開始メッセージに付随してリモート・アプリケーションに送るユーザー ID を指定します。この値は、アプリケーションが受け入れるものとして定義されているユーザー ID のどれかに一致していなければなりません。

このユーザー ID は、ローカル・ノードおよびリモート・ノードのどちらの Linux ログイン・ユーザー ID とも関係ありません。リモート・ノードで CS Linux が実行されている場合は、リモート・ノードで

「会話セキュリティーの構成 (Conversation Security Configuration)」ダイアログを使用して、ユーザー ID を構成する必要があります。

Password

security level として「Program」または「Program strong」を指定した場合は、会話の割り振り時に送信するパスワードを指定します。この値は、指定されたユーザー名と共に使用するものとして、リモート・アプリケーションで定義されているパスワードに一致していなければなりません。

このパスワードは、ローカル・ノードおよびリモート・ノードのどちらの Linux ログイン・パスワードとも関係ありません。リモート・ノードで CS Linux が実行されている場合は、リモート・ノードで「会話セキュリティーの構成 (Conversation Security Configuration)」ダイアログを使用して、パスワードを構成する必要があります。

追加構成

CPI-C 構成が完了したら、次の構成作業を行います。

- APPC セキュリティーを定義するには、[86 ページの『APPC セキュリティーの構成』](#)を参照してください。
- 5250 通信を構成するには、[89 ページの『第 7 章 ユーザー・アプリケーションの定義』](#)を参照してください。

APPC セキュリティーの構成

APPC セキュリティーについては次の構成作業を実行できます。

- セッション・セキュリティーの構成 ([86 ページの『セッションセキュリティーの構成』](#))
- 会話セキュリティーの構成 ([87 ページの『会話セキュリティーの構成』](#))
- セキュリティー・アクセス・リストの構成 ([87 ページの『セキュリティー・アクセス・リストの構成』](#))

セッションセキュリティーの構成

LU-LU セッションを検証するには、セッション・レベルのセキュリティーを使用します。各定義は、ローカル LU 名、パートナー LU 名、およびパスワードで構成されます。

CS Linux は、このパスワードを使用して、ローカル LU とパートナー LU の間のセッションを検証します。(パスワードは、Linux ログオン・パスワードとは関連していません。)

セッション・セキュリティーを構成するには、次の方法のいずれかを使用します

Motif または Web 管理プログラム

アプリケーション・アプリケーション, セキュリティー, and セッション・レベル・セキュリティー from the 運行 menu on the Node window. の選択

コマンド行管理プログラム

次のコマンドを発行します。

```
lu_lu_password の define_lu_password
```

セッション・セキュリティーの構成パラメーター

次のパラメーターは、セッション・セキュリティー構成の必須パラメーターです。

Local LU

ローカル LU の LU 名。この名前は 1 から 8 文字のストリングです。

Partner LU

パートナー LU の完全修飾 LU 名。

Password

ローカル LU とパートナー LU の間のセッションの妥当性を検査するために CS Linux が使用するパスワード。パスワードは、EBCDIC 形式の文字ストリング (16 桁の 16 進数表示) です。パスワードからセ

セッション確立時に交換されるキーが作成されます。このパスワードは、ローカル・ノードおよびリモート・ノードのどちらの Linux ログイン・パスワードとも関係ありません。

追加構成

セッション・セキュリティ構成が完了したら、次の構成作業を行います。

- 会話セキュリティを構成するには、[87 ページの『会話セキュリティの構成』](#)を参照してください。
- 5250 通信を構成するには、[89 ページの『第 7 章 ユーザー・アプリケーションの定義』](#)を参照してください。

会話セキュリティの構成

会話セキュリティは、着信会話を検証するために使用されます。各定義は、ユーザー ID とパスワードで構成されます。このユーザー ID は、ローカル・ノードまたはリモート・ノードのいずれかにある Linux ログイン・ユーザー ID には関連していません。

会話セキュリティを構成するには、次の方法のいずれかを使用します。

Motif または Web 管理プログラム

アプリケーション・アプリケーション, セキュリティ, and 会話レベル・セキュリティ from the 運行 menu on the Node window. の選択

コマンド行管理プログラム

次のコマンドを発行します。

ユーザー ID パスワードのデフラグ

会話セキュリティの構成パラメーター

次のパラメーターは、会話セキュリティ構成の必須パラメーターです。

User ID

リモート・ノードからの着呼会話において受け入れるユーザー ID。ユーザー ID の長さは 10 文字までです。

Password

リモート・ノードからの着呼会話において受け入れるパスワード。パスワードの長さは 10 文字までです。

追加構成

会話セキュリティの構成が終わったら、次に、[89 ページの『第 7 章 ユーザー・アプリケーションの定義』](#)の説明に従って 5250 通信を構成します。

セキュリティ・アクセス・リストの構成

APPC セキュリティー・アクセス・リストを定義して、LU または TP (あるいはその両方) へのアクセスを制御することができます。このリストは、APPC ローカル LU または TP の定義によって参照することができます。

セキュリティ・アクセス・リストを構成するには、次の方法のいずれかを使用します。

Motif または Web 管理プログラム

アプリケーション・アプリケーション, セキュリティ, and 会話レベル・セキュリティ from the 運行 menu on the Node window, then select the Security Access Lists pane and choose ニュー. の選択

コマンド行管理プログラム

次のコマンドを発行します。

セキュリティ・アクセス・リストの定義

セキュリティー・アクセス・リストの構成パラメーター

次のパラメーターは、セキュリティー・アクセス・リスト構成用の 必須パラメーターです。

Name

セキュリティー・アクセス・リストの名前。APPC TP またはローカル LU の定義の中で、この名前を使用してアクセス・リストを参照できます。

Users in access list

セキュリティー・アクセス・リストに含めるユーザーの名前。

追加構成

セキュリティー・アクセス・リストの構成が完了したら、次の構成作業を行います。

- TP アクセスの構成 ([75 ページの『TP の定義』](#))

第7章 ユーザー・アプリケーションの定義

この章では、3270、5250、または LUA のいずれかの通信を使用するユーザー・アプリケーションをサポートするための SNA リソースを構成する方法を紹介します。この種のアプリケーションに必要な SNA リソースは LU です。

3270、LUA、および従属 APPC 通信の場合は、従属 LU を構成する必要があります。独立 APPC および 5250 通信の場合は、デフォルトの制御点 LU (ローカル・ノードの構成時に自動的に定義される) を使用することも、独立 LU を定義することもできます。

この章で説明するリソースを構成する前に、次の構成を済ませておく必要があります。

- [48 ページの『ノードの構成』](#)の説明に従って、ノードを構成する。
- [51 ページの『第4章 接続コンポーネントの定義』](#)の説明に従って、接続を構成します。3270、LUA、および従属 APPC 通信の場合は、従属 LU トラフィックをサポートするリンクを構成する必要があります。独立 APPC および 5250 通信の場合は、リンクは独立 LU トラフィックをサポートしていることが必要です。

アップストリーム SNA ゲートウェイまたは DLUR を使用している場合は、ホストへの直接リンクを構成する必要はありません。詳しくは、[98 ページの『SNA ゲートウェイの構成』](#)および [62 ページの『DLUR PU の定義』](#)を参照してください。

次のリストは、必要な構成作業を、ユーザー・アプリケーションのタイプ 別に示しています。

3270 アプリケーション

3270 通信の場合は、次のリソースを構成します。

1. 3270 ディスプレイまたはプリンターの場合は、[65 ページの『LU タイプ 0 から 3 の定義』](#)の説明に従って従属 LU を定義します。
2. 3270 ディスプレイで LU のプールから選択できるようにするには、[67 ページの『LU プールの定義』](#)の説明に従って LU プールを定義します。専用 LU を使用するディスプレイの場合は、このステップは省略できます。

5250 アプリケーション

5250 通信の場合は、次のリソースを構成します。

1. APPC 通信用のノードを構成します。
 - a. ローカル・ノードの制御点 LU を使用できる場合は、ローカル LU を構成する必要はありません。ローカル LU 定義が必要な場合は (例えば、セッション・セキュリティーを使用するため)、[70 ページの『ローカル LU の定義』](#)の説明に従ってローカル LU を定義します。
 - b. ローカル・ノードが LEN ノードである場合は、[71 ページの『リモートノードの定義』](#)の説明に従って、AS/400 システムをリモート・ノードとして定義する必要があります。

ローカル・ノードが APPN エンド・ノードまたはネットワーク・ノードである場合は、AS/400 システム上の制御点 LU をパートナー LU として使用できるので、他のパートナー LU を構成する必要はありません。

5250 は標準モード QPCSUPP を使用するので、モードを定義する必要はありません。

LUA アプリケーション

LUA アプリケーションをサポートするには、次のリソースを構成します。

1. [65 ページの『LU タイプ 0 から 3 の定義』](#)の説明に従って従属 LU を定義します。
2. LUA アプリケーションが LU のプールから選択できるようにするには、[67 ページの『LU プールの定義』](#)の説明に従って LU プールを定義します。アプリケーションが専用 LU を使用する場合は、このステップは省略できます。

さらに、2 つ以上のクライアント・コンピューター上で同じ LUA アプリケーションを実行している場合、それが指定する LUA LU 名のオーバーライドが必要になることがあります。これにより、2 つ以上のアプリケーションのコピーを、アプリケーションを変更せずに異なるローカル LU を使用するよう

割り当てることができます。詳しくは、[136 ページの『クライアント・アプリケーション上のハード・コード LU 別名をサーバーのドメイン内の LU 別名にマップする』](#) (AIX または Linux 上の Remote API Client の場合) および [130 ページの『クライアント・アプリケーションでハードコーディングされた LU 別名をサーバーのドメイン内の LU 別名にマッピングする』](#) (Windows クライアントの場合) を参照してください。

LUA アプリケーションはノードの LU 0 から 3 リソースを使用して、ホスト・アプリケーションと通信します。追加リソースを定義する必要はありません。

第 8 章 パススルー・サービスの構成

CS Linux を実行するサーバー上のパススルー・サービスにより、SNA ホストと、そのホストに直接接続されていないローカル・システムとの間での通信ができます。

CS Linux には、TN3270、TN3287、および TN3270E クライアント (総称的に「TN3270 クライアント」と呼ばれます) のための TN サーバー・サポートが組み込まれています。この機能の構成方法については、[91 ページの『TN サーバーの構成』](#)を参照してください。

CS Linux には、集散的に「Telnet クライアント」と呼ばれる TN3270、TN3270E、TN5250、および VT クライアントへのパススルー TCP/IP ホスト・アクセスのための TN リダイレクター・サポートが組み込まれています。この機能の構成方法については、[95 ページの『TN リダイレクターの構成』](#)を参照してください。

SNA ゲートウェイは、ホストとローカル・システムの間での接続を提供します。ローカル・ノード上で、この機能をサポートする LU を構成することができます ([98 ページの『SNA ゲートウェイの構成』](#)を参照)。また、明示的に構成されていないダウンストリーム LU をサポートするためのテンプレートも定義できます ([52 ページの『DLC、ポート、および接続ネットワークの定義』](#)を参照)。

DLUR は、APPN ネットワーク内のホストとノードの間の従属 LU セッションをサポートしています。この機能の構成方法については、[99 ページの『DLUR の構成』](#)を参照してください。

TN サーバーの構成

TN サーバーを使用して、TN3270 クライアントは、TN サーバーをインプリメントする 中間 CS Linux ノードを介してホストと通信できます。TN3270 クライアントは TCP/IP を使用して TN サーバーに接続され、TN サーバーで定義されている LU を使用します。TN サーバー LU は、クライアント用の TN3270 セッションをサポートするために、ホストの LU とのセッションを確立します。

TN サーバーを構成するには、次の構成作業を行う必要があります。

- [48 ページの『ノードの構成』](#)の説明に従って、ローカル・ノードを定義します。
- [51 ページの『第 4 章 接続コンポーネントの定義』](#)の説明に従って、ローカル・ノードとホスト間の、従属トラフィック用のポートとリンク・ステーションを構成します。
- ホストとの通信に使用するローカル・ノード上の TN3270 LU を定義します。LU を追加する方法については、[65 ページの『LU タイプ 0 から 3 の定義』](#)を参照してください。
- LU プールを使用する場合は、[67 ページの『LU プールの定義』](#)の説明に従って LU プールを定義します。

TN サーバーを構成するには、次の作業を行います。

- TN サーバーを使用する TN3270 クライアント ごとに、TN サーバー・アクセス・レコードを 1 つずつ構成するか、すべてのクライアントがサーバーにアクセスできるためのデフォルト のレコードを構成します ([92 ページの『TN サーバー・アクセス・レコードの構成』](#)を参照)。
- TN3270E または TN3287 クライアントをサポートしようとする場合は、ディスプレイ LU およびプリンター LU 用のアソシエーション・レコードを定義できます ([94 ページの『TN サーバー関連付けレコードの構成』](#)を参照)。このレコードがあれば、TN3270E または TN3287 クライアントは、関連のディスプレイ LU を選択することにより特定のプリンターを選択できます。クライアントは、TN サーバー・アクセス・レコード内の LU を選択する権限を持っていることが必要です。

TN サーバー用の追加オプションを使用すると、プリンターの応答を強制したり、すべての TN3270 セッションに対してキープアライブ方式を指定したり、TN3270 クライアントの許可検査に使用される取り消しリストを維持する外部 LDAP サーバーへのアクセス方法を指定したりできます。これらのオプションにアクセスするには、「TN Server (TN サーバー)」ウィンドウの「Service (サービス)」メニューを使用してください。

注: TN サーバーを構成する前に、[100 ページの『TN サーバーおよび TN リダイレクター: 追加のセキュリティ情報』](#)にある情報を検討してください。そこでは一部の TN3270 クライアントまたは Telnet クライアントとの相互作用に影響を与える可能性があるプロトコル変更について説明されています。

TN サーバー・アクセス・レコードの構成

TN サーバー・アクセス・レコードは、どの TN3270 クライアントが TN サーバーにアクセスできるか、およびどの LU が使用するかを示します。各アクセス・レコードは、TN サーバーへのアクセスを許可されている TN3270 クライアント、クライアントが接続する TCP/IP ポート、およびクライアントが使用する LU または LU プールを識別します。

また、任意の TN3270 クライアント（すべてのクライアントに同じ LU または LU プールを持つ）によるアクセスを可能にするデフォルト・レコードを定義することもできます。

TN3270 クライアントは、ノード、ポート、およびリンク・ステーションがアクティブになっている場合にのみ TN サーバーを使用できます。

TN サーバー・アクセス・レコードを構成するには、以下の方法のいずれかを使用します。

Motif または Web 管理プログラム

Select TN サーバー from the 運行 menu on the Node window, and TN サーバー from the submenu. 表示されたウィンドウで、「TN サーバー・クライアント・アクセス許可」ペインを選択して、ニューを選択します。

コマンド行管理プログラム

次のコマンドを発行します。

```
アクセスの定義 _tn3270_-----
```

注: If you define a TN server access record using the command-line administration program, snaadmin、または NOF アプリケーションを使用すると、*listen_local_address* parameter to specify an address on the local TN Server computer to which the TN3270 client will connect. これを行うと、アクセス・レコードは Motif 管理プログラムに表示されないため、そのプログラムを使用してそのプログラムを表示または管理することはできません。コマンド行管理プログラムまたは NOF アプリケーションを使用して、それを引き続き管理することができます。

TN サーバー・アクセス・レコードの構成パラメーター

次のパラメーターは、TN サーバー・アクセス・レコードの構成用の 必須パラメーターです。

TN3270 client address

アクセス・レコードが適用される TN3270 クライアントを識別する アドレス。

デフォルト・レコード

どの TN3270 にもアクセスが許可されます。

TCP/IP 名または別名 (TCP/IP name or alias)

指定した TN3270 クライアントによるアクセスが許可されます。クライアントの TCP/IP 名が分かっている場合は、このオプションを選択し、その名前を入力してください。多くのコンピューターでは、hostname コマンドを使用すると、そのコンピューターの TCP/IP 名が分かります。

TCP/IP アドレス

特定の TCP/IP アドレスからのアクセスが許可されます。TN3270 クライアントの TCP/IP アドレスが分かっている場合は、このオプションを選択し、そのアドレスを入力してください。指定できるのは、以下のいずれかです。

- IPv4 ドット 10 進アドレス (例えば、193.1.11.100)
- IPv6 コロン 16 進アドレス (例えば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)

Support TN3270E

ノードが提供する TN3270 サポートのレベル。

TN3270

TN3270 プロトコルのみをサポートします。このオプションを選択すると、クライアントが TN3270E プロトコルをサポートしている場合でも、サーバーでの TN3270E サポートはありません。

TN3270E

TN3270 および TN3270E の両方のプロトコルをサポートします (デフォルト)。

TN3270 および TN3287 プロトコルは、このオプションの選択に関係なく、常にサポートされます。

AS/400 の TN3270 クライアントについては、このオプションを TN3270E に設定する必要があります。

TCP/IP port number

TN3270 クライアントを接続するポートの TCP/IP ポート番号 (TN サーバー上の)。

注: TCP/IP ポートは、SNA ポートとはまったく関係がありません。

TN3270 サービス用の予約済みポート番号は 23 です。TN サーバー上の使用されていない別のポート番号を選択する場合は、TN3270 クライアント上でもそのポート番号を構成する (または、そのポート番号を指定するオプションを使用して TN3270 クライアントを開始する) 必要があります。使用できる可能性が高いのは、2000 より大きいポート番号です。256 から 1023 の範囲内のポート番号を使用すると、セキュリティは多少向上しますが、これらのポート番号は既に使用されていることが多いです。

1 つの TN3270 クライアントが複数の LU または LU プールを使用できるようにする場合は、複数のアクセス・レコードを定義し、それぞれに異なる TCP/IP ポート番号を割り当てて、各 LU または LU プールをそれぞれ異なるポート番号で識別できるようにします。

Display LU assigned

アクティブ状態にあるときに TN3270 クライアントがアクセスする LU の名前。この LU は、ローカル・ノード上の従属 LU でなければなりません。特定の LU の名前の代わりに、LU プールの名前を指定することもできます。

Printer LU assigned

このアクセス・レコードを使用するクライアント用のデフォルトのプリンター LU または LU プールの名前。この LU は、ローカル・ノード上の従属 LU として定義されていなければなりません。

Allow access to specific LU

TN3270E および TN3287 クライアントがセッション用に特定の LU を要求できるようにしたい場合は、このオプションを指定します。(TN3270 クライアントの場合、このオプションは使用できません。)

SSL secure session

このセッションがサーバーへのアクセスに Secure Sockets Layer (SSL) を使用していることを示します。

このオプションは、サーバーに SSL をサポートするのに必要な追加ソフトウェアをインストールしている場合のみ使用できます。それ以外はこのオプションを選択することはできません。

注: このセッションの *TCP/IP port number* パラメーターが、セッションで Telnet デーモンの TCP/IP ポートが使用されていることを示す場合は、このセッションに対して SSL を使用しないでください。Telnet デーモンの TCP/IP ポートを使用するセッションで SSL を使用すると、ノードが活動状態であるときに、Telnet クライアントが telnet を使用して CS Linux コンピューターにアクセスすることができなくなります。

Perform client authentication

このオプションは、ユーザーが「*SSL secure session*」オプションを選択した場合のみ表示されます。

このオプションを指定して、TN サーバーがクライアント認証を使用するセッションを要求していることを示します。クライアントは、有効な証明書 (TN サーバーを使用する許可のある有効なクライアントとして、クライアントを識別できる情報) を送信する必要があります。

TN Server は、この証明書が有効かどうかを検査します。また、この証明書を外部 LDAP サーバーの証明書取り消しリストに照らして検査して、ユーザーの許可が取り消されていないことを確認する必要があります。この場合はさらに、TN サーバーの「*Advanced Parameters (拡張パラメーター)*」ダイアログを使用して、このサーバーへのアクセス方法を指定する必要があります。

security level

このセッションに必要な SSL セキュリティ・レベルを示します。セッションでは、クライアントおよびサーバーでサポートされている最もレベルの高いセキュリティが使用されます。クライアントが必要なセキュリティ・レベルがサポートされていない場合、セッションは開始されません。

このオプションは、ユーザーが「*SSL secure session*」オプションを選択した場合のみ表示されます。可能な値は以下のとおりです。

Authenticate Only

証明書を交換する必要がありますが、暗号化は使用しません。このオプションは、通常、クライアントがセキュア・イントラネットに接続している場合に、暗号化のオーバーヘッドを回避するために使用します。

Authenticate Minimum

クライアントはサーバーからの証明書を要求し、証明書の有効性を検査する必要があります。暗号化はクライアントが要求しても、必要ありません (使用することはできません)。

40 Bit Minimum

クライアントは最低 40 ビットの暗号化でサポートする必要があります。

56 Bit Minimum

クライアントは最低 56 ビットの暗号化でサポートする必要があります。

128 Bit Minimum

クライアントは少なくとも 128 ビット暗号をサポートしている必要があります。

168 Bit Minimum

クライアントは少なくとも 168 ビット暗号をサポートしている必要があります。

256 Bit Minimum

クライアントは少なくとも 256 ビット暗号をサポートしている必要があります。

注: 暗号化を使用する場合は、CS Linux に追加のソフトウェアをインストールする必要があります。詳しくは、「*IBM Communications Server for Data Center Deployment on Linux 入門*」を参照してください。暗号化レベルをサポートするために必要なソフトウェアが利用できない国があるため、場所によっては、リストされている暗号化レベルをすべて使用できないことがあります。

追加構成

TN サーバー・アクセスの構成が完了したら、次の構成作業を行います。

- 94 ページの『[TN サーバー関連付けレコードの構成](#)』の説明に従って、TN サーバーのアソシエーション・レコードを構成する。

TN サーバー関連付けレコードの構成

TN サーバー関連付けレコードは、プリンター LU とディスプレイ LU との間の関連を定義します。これにより、TN3270E または TN3287 プロトコルが 2 つのプロトコルを接続できるようになります。クライアントのアクセス・レコードが特定の LU の選択を許可する場合、このレコードは、関連するディスプレイ LU を指定することによって、クライアントが特定のプリンターを選択できるようにします。

TN サーバー関連付けレコードを構成するには、以下の方法のいずれかを使用します。

Motif または Web 管理プログラム

「ノード」ウィンドウの「運行」メニューから TN サーバーを選択し、「TN サーバー」ウィンドウで「アソシエーション・レコード」ペインを選択し、ニューを選択します。の選択

コマンド行管理プログラム

次のコマンドを発行します。

```
非 323270_関連付けの数
```

TN サーバー・アソシエーション・レコードの構成パラメーター

次のパラメーターは、TN サーバー・アソシエーション・レコードの構成用の必須パラメーターです。

Display LU

ディスプレイ LU の名前 (ローカル・ノードで定義されているものでなければなりません)。

Printer LU

プリンター LU の名前 (ローカル・ノードで定義されているものでなければなりません)。他の TN サーバー・アソシエーション・レコードに既に入力されている プリンター LU は指定しないでください。

TN リダイレクターの構成

TN リダイレクターを使用すると、TN3270、TN3270E、TN5250、および VT クライアント (総称して Telnet クライアントと呼ばれます) が、TN リダイレクターをインプリメントした中間 CS Linux ノードを介してホストと通信できます。クライアントは、TCP/IP を使用して TN リダイレクターに接続し、TN リダイレクターは、ホストに対して別の TCP/IP 接続を確立します。

TN リダイレクターを構成するには、次の手順で行います。

- サーバーを使用する Telnet クライアント ごとに、TN リダイレクター・アクセス・レコードを 1 つずつ構成するか、すべてのクライアントがサーバーにアクセスするためのデフォルト のレコードを構成します (95 ページの『TN リダイレクターアクセスレコードの構成』を参照)。

注: TN リダイレクターを構成する前に、100 ページの『TN サーバーおよび TN リダイレクター: 追加のセキュリティ情報』にある情報を検討してください。そこでは一部の TN3270 クライアントまたは Telnet クライアントとの相互作用に影響を与える可能性があるプロトコル変更について説明されています。

TN リダイレクターアクセスレコードの構成

TN リダイレクター・アクセス・レコードは、TCP/IP リンク上で TN リダイレクターにアクセスできる Telnet クライアントを示します。各アクセス・レコードは、TN リダイレクターへのアクセスを許可されている Telnet クライアントを識別します。クライアントが CS Linux への接続に使用する TCP/IP ポート、CS Linux がホストに接続するために使用する TCP/IP ポート、および SSL セキュリティ設定を識別します。任意のクライアントによるアクセスを可能にするデフォルト・レコードを定義することもできます。

すべてのクライアントが TN リダイレクターを使用することを許可し、すべてのクライアントが同じホスト・アクセス構成を使用するようにしたい場合は、デフォルト・レコードを定義できます。

Telnet クライアントは、ノードがアクティブの場合にのみ TN リダイレクターを使用できます。

TN リダイレクター・アクセス・レコードを構成するには、次の方法のいずれかを使用します

Motif または Web 管理プログラム

Select TN サーバー from the 運行 menu on the Node window, and TN サーバー from the submenu. 結果のウィンドウで、TN リダイレクター・クライアント・アクセス権限ペインを選択して、ニューを選択します。

コマンド行管理プログラム

次のコマンドを発行します。

```
define_tn_リダイレクト
```

注: If you define a TN redirector access record using the command-line administration program, snaadmin、または NOF アプリケーションを使用すると、listen_local_address parameter to specify an address on the local TN Server computer to which the TN3270 client will connect. これを行うと、アクセス・レコードは Motif 管理プログラムに表示されないため、そのプログラムを使用してそのプログラムを表示または管理することはできません。コマンド行管理プログラムまたは NOF アプリケーションを使用して、それを引き続き管理することができます。

TN リダイレクターのアクセス・レコード構成パラメーター

TN リダイレクターのアクセス・レコードの構成には、クライアントとホスト TCP/IP 接続用の 2 つのグループのパラメーターから構成されます。

クライアント・パラメーターは以下のとおりです。

Telnet client address

アクセス・レコードが適用される Telnet クライアントを識別するアドレスです。

デフォルト・レコード

任意の Telnet クライアントによるアクセスを許可します。

TCP/IP 名または別名 (TCP/IP name or alias)

名前付きの Telnet クライアントによるアクセスを許可します。クライアントの TCP/IP 名が分かっている場合は、このオプションを選択し、その名前を入力してください。多くのコンピューターでは、hostname コマンドを使用すると、そのコンピューターの TCP/IP 名が分かります。

TCP/IP アドレス

特定の TCP/IP アドレスからのアクセスが許可されます。クライアントの TCP/IP アドレスが分かっている場合は、このオプションを選択し、そのアドレスを入力してください。指定できるのは、以下のいずれかです。

- IPv4 ドット 10 進アドレス (例えば、193.1.11.100)
- IPv6 コロン 16 進アドレス (例えば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)

TCP/IP port number

クライアントを接続するポートの TCP/IP ポート番号 (TN サーバー上の)。

注: TCP/IP ポートは、SNA ポートとはまったく関係がありません。

クライアント上のこのポート番号も構成する必要があります (またはオプションを使用してポート番号を指定し、クライアントを開始します)。使用できる可能性が強いのは、2000 より大きいポート番号です。256 から 1023 の範囲内のポート番号を使用すると、セキュリティは多少向上しますが、これらのポート番号は既に使用されている場合が多いです。

SSL secure session

このセッションがサーバーへのアクセスに Secure Sockets Layer (SSL) を使用していることを示します。

このオプションは、サーバーに SSL をサポートするのに必要な追加ソフトウェアをインストールしている場合のみ使用できます。それ以外はこのオプションを選択することはできません。

Perform client authentication

このオプションは、ユーザーが「SSL secure session」オプションを選択した場合のみ表示されます。

このオプションを指定して、TN サーバーがクライアント認証を使用するセッションを要求していることを示します。クライアントは、有効な証明書 (TN サーバーを使用する許可のある有効なクライアントとして、クライアントを識別できる情報) を送信する必要があります。

TN Redirector は、証明書が有効であるかどうかを検査すると同時に、そのユーザーの権限が取り消されないように、外部 LDAP サーバーの証明書取り消しリストに照らしてその証明書を検査する必要があります。この場合は、さらに TN サーバーの「Advanced Parameters (拡張パラメーター)」ダイアログを使用して、このサーバーへのアクセス方法を指定する必要があります。(このダイアログは、「TN Server (TN サーバー)」ウィンドウの「Service (サービス)」メニューから選びます。)

security level

クライアント・セッションに必要な SSL のセキュリティ・レベルを指定します。セッションでは、クライアントおよびサーバーでサポートされている最もレベルの高いセキュリティが使用されます。クライアントで必要なセキュリティ・レベルがサポートされていない場合、セッションは開始されません。

このオプションは、ユーザーが「SSL secure session」オプションを選択した場合のみ表示されます。可能な値は以下のとおりです。

Authenticate Only

証明書を交換する必要がありますが、暗号化は使用しません。このオプションは、通常、クライアントがセキュア・イントラネットに接続している場合に、暗号化のオーバーヘッドを回避するために使用します。

Authenticate Minimum

クライアントはサーバーからの証明書を要求し、証明書の有効性を検査する必要があります。暗号化はクライアントが要求しても、必要ありません (使用することはできません)。

40 Bit Minimum

クライアントは最低 40 ビットの暗号化でサポートする必要があります。

56 Bit Minimum

クライアントは最低 56 ビットの暗号化でサポートする必要があります。

128 Bit Minimum

クライアントは少なくとも 128 ビット暗号をサポートしている必要があります。

168 Bit Minimum

クライアントは少なくとも 168 ビット暗号をサポートしている必要があります。

256 Bit Minimum

クライアントは少なくとも 256 ビット暗号をサポートしている必要があります。

注: 暗号化を使用する場合は、CS Linux に追加のソフトウェアをインストールする必要があります。詳しくは、「*IBM Communications Server for Data Center Deployment on Linux 入門*」を参照してください。暗号化レベルをサポートするために必要なソフトウェアが利用できない国があるため、場所によっては、リストされている暗号化レベルをすべて使用できないことがあります。

宛先ホストのパラメーターは以下のとおりです。

Address

アクセス・レコードが適用されるホストを識別するアドレスです。

TCP/IP 名または別名 (TCP/IP name or alias)

名前付きのホストへのアクセスです。ホストの TCP/IP 名が分かっている場合は、このオプションを選択し、その名前を入力してください。多くのコンピューターでは、hostname コマンドを使用すると、そのコンピューターの TCP/IP 名が分かります。

TCP/IP アドレス

特定の TCP/IP アドレスへのアクセスです。ホストの TCP/IP アドレスが分かっている場合は、このオプションを選択し、そのアドレスを入力してください。指定できるのは、以下のいずれかです。

- IPv4 ドット 10 進アドレス (例えば、193.1.11.100)
- IPv6 コロン 16 進アドレス (例えば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)

TCP/IP port number

TN リダイレクターがホストへのアクセスに使用する TCP/IP ポート番号です。

注: TCP/IP ポートは、SNA ポートとはまったく関係がありません。

ホスト上のこのポート番号も構成する必要があります。使用できる可能性が強いのは、2000 より大きいポート番号です。256 から 1023 の範囲内のポート番号を使用すると、セキュリティは多少向上しますが、これらのポート番号は既に使用されている場合が多いです。

SSL secure session

このオプションを指定して、TN リダイレクターがホストへのアクセスに Secure Sockets Layer (SSL) を使用することを示します。

このオプションは、ホストが SSL をサポートする場合のみ使用できます。

security level

ホスト・セッションに必要な SSL のセキュリティ・レベルを指定します。セッションには、ホストとサーバーの両方でサポートできるセキュリティ・レベルの中で最高レベルのものを使用します。ホストが必須レベル以上のセキュリティをサポートできない場合、セッションは開始されません。

このオプションは、ユーザーが「SSL secure session」オプションを選択した場合のみ表示されます。

可能な値は以下のとおりです。

Authenticate Only

証明書を交換する必要がありますが、暗号化は使用しません。このオプションは、通常、ホストがセキュア・イントラネットに接続している場合に、暗号化のオーバーヘッドを回避するために使用します。

Authenticate Minimum

ホストはサーバーからの証明書を要求し、証明書の有効性を検査する必要があります。暗号化はホストが要求しても、必要ありません (使用することはできます)。

40 Bit Minimum

ホストは最低 40 ビットの暗号化でサポートする必要があります。

56 Bit Minimum

ホストは最低 56 ビットの暗号化でサポートする必要があります。

128 Bit Minimum

ホストは少なくとも 128 ビット暗号をサポートしている必要があります。

168 Bit Minimum

ホストは少なくとも 168 ビット暗号をサポートしている必要があります。

256 Bit Minimum

ホストは少なくとも 256 ビット暗号をサポートしている必要があります。

注: 暗号化を使用する場合は、CS Linux に追加のソフトウェアをインストールする必要があります。詳しくは、「*IBM Communications Server for Data Center Deployment on Linux 入門*」を参照してください。暗号化レベルをサポートするために必要なソフトウェアが利用できない国があるため、場所によっては、リストされている暗号化レベルをすべて使用できないことがあります。

SNA ゲートウェイの構成

通常、従属 LU セッションは、ホスト・コンピューターへの直接通信リンクを必要とします。ただし、CS Linux を実行するノードでは、ホストへの直接通信リンクを持つノードが、ダウンストリーム・コンピューターの LU に SNA ゲートウェイ機能を提供し、CS Linux ノードからの通信リンクを介してホストにアクセスできるようにすることもできます。ダウンストリーム・コンピューターには、ホストとの依存通信をサポートする SNA PU タイプ 2.0 または 2.1 が含まれている必要があります。例えば、ダウンストリーム・コンピューターは、スタンドアロン構成で CS Linux を実行する別のコンピューターである場合があります。

SNA ゲートウェイ機能を使用すると、ホストとダウンストリーム・コンピューターとの間で転送されるすべてのデータが、CS Linux ローカル・ノードを経由してルーティングされます。これにより、ダウンストリーム・コンピューターは、直接リンクを必要とせずに、CS Linux または他のダウンストリーム・コンピューターとのホスト接続を共有することができます。例えば、ローカル・トークンリング・ネットワーク上の CS Linux に接続されたいくつかのダウンストリーム・コンピューターをセットアップして、CS Linux からホストへの同じ長距離 SDLC 専用回線にアクセスするようにすることができます。

SNA ゲートウェイを使用すると、ホストでの構成も単純化されます。ホスト構成には、CS Linux コンピューターとそのホスト通信リンクのみを含める必要があります。ダウンストリーム・コンピューターの LU は、CS Linux コンピューターのリソースの一部として構成されています。ホスト・コンピューターは SNA ゲートウェイが使用されていることを認識していません。

SNA ゲートウェイを構成する前に、次の構成タスクを実行する必要があります。

- [48 ページの『ノードの構成』](#)の説明に従ってローカル・ノードを定義します。
- [51 ページの『第 4 章 接続コンポーネントの定義』](#)で説明されているように、ローカル・ノードとホスト間の従属トラフィック用にポートおよびリンク・ステーションを構成します。また、ローカル・ノードとダウンストリーム・ノードの間の従属トラフィック用にポートおよびリンク・ステーションを構成します。ダウンストリーム・リンクの場合、暗黙のダウンストリーム LU (ローカル・ノード上で明示的に定義されていない LU) をサポートするように、ポート上でテンプレートを構成できます。
- ホスト (アップストリーム LU) との通信に使用されるローカル・ノード上の LU を定義します。従属 LU 6.2 LU を含むアップストリーム LU は、LU タイプ 0-3 構成ダイアログを使用して定義しなければなりません。これは、LU タイプが無制限 (不明)であることを指定します。LU を追加するには、[65 ページの『LU タイプ 0 から 3 の定義』](#)を参照してください
- いずれかの LU プールを使用する場合は、[67 ページの『LU プールの定義』](#)の説明に従って LU プールを定義します。

SNA ゲートウェイを使用可能にするには、ダウンストリーム・ワークステーションとのセッションをサポートするように、ローカル・ノード上の LU を構成する (暗黙のダウンストリーム LU をサポートするためにポートでテンプレートを構成した場合は、ダウンストリーム LU を明示的に定義する必要はありません)

ん。) ローカル・ノード上で定義される LU は、"ダウンストリーム LU。" と呼ばれます。ダウンストリーム LU を構成するには、ダウンストリーム・ノードで使用されている LU 番号、およびホスト LU の名前が必要です。(ダウンストリーム・ノード上に定義されている LU は、任意の従属 LU タイプにすることができます。)

ダウンストリーム LU を構成するには、次の方法のいずれかを使用します

Motif または Web 管理プログラム

「ノード」ウィンドウの「運行」メニューから SNA ゲートウェイと新しいダウンストリーム LU の選択

コマンド行管理プログラム

次のいずれかのコマンドを実行します。

ダウンストリームの `define_downstream_lu`

ダウンロード範囲が不足しています

ダウンストリーム LU の構成パラメーター

次のパラメーターは、ダウンストリーム LU 構成用の必須パラメーターです。

Downstream LU name

各ダウンストリーム LU の名前。この LU 名はローカルで LU を識別するために使用されるのみなので、ダウンストリーム・ノード上の構成と一致している必要はありません。

LU の範囲を定義する場合は、1 から 5 文字のベース名を指定します。CS Linux は、3 桁の 10 進数ストリングをベース名に追加して、指定された LU 番号ごとに LU 名を作成します。

Downstream PU name

ダウンストリーム・ノードへのリンク・ステーションの名前。

LU number

この LU 番号は、ダウンストリーム・ノードで定義されている LU 番号と同じでなければなりません。使用する LU 番号が分からない場合は SNA ネットワーク計画担当者に問い合わせてください。

LU の範囲を定義することにより、連続した LU 番号を持つ複数の LU を構成できます。

Upstream LU name

ダウンストリーム LU が通信するホスト LU または LU プールの名前。

Delayed logon

CS Linux は、ユーザーの起動時間を減らすために、アップストリーム LU を割り当てずにログオン画面を表示します。3270 のユーザーは、アップストリーム LU に関連付けられる前に、いずれかのキーを押す必要があります。

Allow timeout

必要な LU の数を減らすために、アクティブの PLU-SLU セッションを持たない LU は、ここで指定する秒数が経過すると、アップストリーム LU から切り離されます。

追加構成

SNA ゲートウェイ用のダウンストリーム LU の構成が完了したら、次の構成作業を行います。

- ユーザー・アプリケーションを構成するには、[89 ページの『第 7 章 ユーザー・アプリケーションの定義』](#)を参照してください。

DLUR の構成

通常は、従属 LU セッションでは、ホスト・コンピューターへの直接通信リンクが必要です。APPN ネットワーク内で多数のノード (ホスト・ノードも含む) が相互に接続される場合、一部のノードには、ホストへの直接接続がなく、代わりに別のノードを経由する間接接続があります。直接接続がない限り、このような間接接続ノード内の LU からホストへの従属 LU セッションを確立することはできません。

従属 LU リクエスター (Dependent LU requester: DLUR) は、この制限を解消するために設計された APPN 機能です。DLUR は、APPN ノード (CS Linux を実行するノードなど) で構成できます。DLUR はホストの従属 LU サーバー (DLUS) と共に動作し、DLUR ノード上の従属 LU から、APPN ネットワークを経由して DLUS ホストまでのセッションの経路を定めます。

ホストへの経路は、複数のノードにまたがることができ、APPN のネットワーク管理機能、動的リソース探索機能、および経路計算機能を利用して設定されます。DLUR は LU が定義されているノード上で使用でき、DLUS はホスト・ノード上で使用できることが必要ですが、セッション経路内の中間ノード上で DLUR を使用可能にする必要はありません。

注: LEN ノード上では DLUR は構成できません。

CS Linux DLUR ノードがネットワーク・ノードの場合は、CS Linux ノードに接続されているダウンストリーム・コンピューターの従属 LU に、パススルー DLUR 機能を提供することができます。(この機能をサポートするのはネットワーク・ノードのみです。) このようなダウンストリーム LU は、CS Linux ノードの DLUR を使用して、ノード内部の LU と同じ方法で、ネットワーク経由でホストにアクセスできます。

注: エンド・ノード上でパススルー DLUR を構成することはできません。

DLUR を構成するために必要な作業は、従属 LU がローカル・ノードにあるかダウンストリーム・ノードにあるかによって決まります。

ローカル・ノード上の DLUR サポートを構成するには、次の構成作業を行う必要があります。

1. 48 ページの『ノードの構成』の説明に従って、ローカル・ノードを定義します。ダウンストリーム・ノード用のパススルー DLUR サポートを提供することを予定している場合は、このノードを APPN ネットワーク・ノードとして定義してください。
2. APPN ネットワークへの接続を構成します。APPN 接続では、ローカル・ノードと隣接 APPN ネットワーク・ノードの間の独立トラフィック用のポートとリンク・ステーションが、少なくとも 1 つずつ必要です (51 ページの『第 4 章 接続コンポーネントの定義』を参照)。
3. 62 ページの『DLUR PU の定義』の説明に従って、ローカル・ノード上に DLUR PU を定義します。(DLUR PU はホストへの接続をサポートします。)
4. ローカル・ノード上の LU をサポートするように DLUR を構成するには、89 ページの『第 7 章 ユーザー・アプリケーションの定義』の説明に従って、ローカル・ノード上の LU を追加する必要があります。LU は、3270 ディスプレイ、3270 プリンター、または LUA をサポートするように構成できます。LU がサポートするユーザー・アプリケーションに必要な条件によっては、さらに追加の構成が必要になることがあります。

ダウンストリーム・ノード用のパススルー DLUR サポートを構成するには、次の構成作業を行う必要があります。

1. ローカル・ノードを、APPN ネットワーク・ノードとして定義する (48 ページの『ノードの構成』を参照してください)。
2. ダウンストリーム・ノードへの接続を構成します。51 ページの『第 4 章 接続コンポーネントの定義』の説明に従って、ローカル・ノードと各ダウンストリーム・ノードの間の従属トラフィック用のポートとリンク・ステーションを構成します (ダウンストリーム・ノード用の DLUR をサポートするためには、DLUR PU を定義する必要はありません)。
3. 1 つのダウンストリーム・ノードが複数の PU をサポートできます。この場合、各ダウンストリーム PU は異なるリンクに関連付けられるため、CS Linux DLUR ノードとダウンストリーム・ノードの間に複数のリンクを構成する必要があり、それぞれのリンクのダウンストリーム PU 名を知っている必要があります。

TN サーバーおよび TN リダイレクター: 追加のセキュリティ情報

このセクションでは、TN サーバーおよび TN リダイレクター接続の SSL セキュリティを維持するためのプロトコル変更に関する追加情報を提供します。これは、一部の TN3270 クライアントまたは Telnet クライアントとの対話に影響する場合があります。

CS Linux には、RFC 5746、<http://tools.ietf.org/html/rfc5746> (GSKIT パッケージのバージョン 8 で実装) に記載されている機能が含まれています。これにより、SSL セキュア TN3270 および TN リダイレクター接続でのハンドシェイク再ネゴシエーションのためのセキュリティの脆弱性が修正されます。ただし、

クライアントがこのプロトコルを実装せず、再ネゴシエーションを使用しようとする、クライアントからの保護された接続が失敗する可能性があります。

RFC 5746 プロトコルを実装していないクライアントからの TN3270 および TN リダイレクター接続で問題が発生した場合は、CS Linux の動作を変更して、このプロトコルが変更される前の古い操作に戻すことができます (再ネゴシエーションを許可します)。これを行うには、CS Linux を開始する前に環境変数 GSKIT_RENEGOTIATION をシェルに設定するか、環境ファイル /etc/opt/ibm/sna/environment: を使用します。

- 再ネゴシエーションを許可するには、GSKIT_テルの再ネゴシエーション=フル または GSKIT_ネゴシエーション = 省略形 は、クライアントの要件に従っています。を
- 再ネゴシエーションを防ぐために、GSKIT_ネゴシエーション=なし. を設定
- 環境変数を設定しない場合、CS Linux は再ネゴシエーションを許可しません。

注: 再ネゴシエーションを許可する場合は、<http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-3555> を参照し、その説明にある可能性のあるセキュリティー問題を認識している必要があります。

第 9 章 NetView からの CS Linux の管理

CS Linux に組み込まれているリモート・コマンド機能 (RCF) は、ホスト・コンピューターの NetView プログラムと連携して動作し、NetView のオペレーターがホスト NetView プログラムから CS Linux コンピューターにコマンドを発行できるようにします。(NetView および RCF コマンドの概要については、[103 ページの『ホスト NetView プログラムの使用』](#)を参照してください。)

CS Linux RCF には次の 2 つの機能があります。

- サービス・ポイント・コマンド機能 (SPCF) は、NetView オペレーターが、コマンド行管理プログラム `snaadmin` の場合と同じ構文を使用して、NetView から CS Linux 管理コマンドを発行できるようにします。この機能については、[105 ページの『SPCF の使用』](#)で説明します。
- UNIX コマンド機能 (UNIX command facility: UCF) は、NetView オペレーターが NetView から Linux オペレーティング・システム・コマンドを発行できるようにします。この機能については、[106 ページの『UCF の使用』](#)で説明します。

これらの機能は、どちらも NetView コンソールから同じ方法でアクセスでき、コマンド発行のための全体的な構文も同じです。

ホスト NetView プログラムの使用

CS Linux RCF は、ホスト・コンピューターにある NetView プログラムと連携して動作します。ホストでは、バージョン 1 リリース 2 以降の NetView を実行していることが必要です。CS Linux は、NetView バージョン 1 リリース 1 をサポートしていません。

NetView プログラムを使用するには、次のものがが必要です。

- ホスト NetView プログラム用のログイン ID およびパスワード (この情報はホストの担当者から入手してください)。
- CS Linux のサービス・ポイント名。NetView プログラム用のホストで定義されます (この情報についてはホストの担当者に問い合わせてください)。
- NetView プログラムを実行しているホスト・コンピューターにアクセスするための、DLC、ポート、およびリンク・ステーション。

RCF 機能をテストするには、ホストから直接 NetView にアクセスする代わりに、3270 エミュレーションを使用して CS Linux からアクセスすることができます。テストする場合は次のものも必要です。

- ホストで構成されている 3270 LU
- この LU を使用する 3270 セッション

必要な構成情報は、ホストの管理者から入手してください。

NetView プログラムにアクセスするには、次の手順を使用します。

1. CS Linux ソフトウェアが、RCF アクセス・パラメーター (`define_rcf_access` レコード) を含むノード構成ファイルを使用して開始されていることを確認します。
2. 3270 エミュレーションを使用して NetView プログラムにアクセスする場合は、3270 エミュレーション・プログラムを開始し、ホストへのセッションをアクティブにします。
3. ホスト管理者の指示に従って、NetView を開始し、ログオンします。(操作手順は NetView のバージョンによって異なる場合があります。)
4. 必要に応じて SPCF または UCF のコマンドを発行します。
5. 3270 エミュレーションを使用して NetView にアクセスしている場合は、コマンドの発行を終了してから、3270 の資料の指示に従って 3270 エミュレーションを終了させてください。

NetView 画面表示

NetView 画面のレイアウトは、どのホストにあるどのバージョンの NetView かによって異なります。代表的なレイアウトを、104 ページの図 19 に示します。

この画面には、下部に入力域があります。コマンドはこの領域に入力します。??? の行は、メインスクリーン域 (NetView がコマンドに対する 応答を表示する場所) と入力域の境界を示しています。

```

MCCF          N E T V I E W          [SCAN DDAC12 07/18/95 13:52:24 A]

RUNCMD SP=ABCDPU01,APPL=NODE,START_DLC,DLC_NAME=TOKR01
COMMAND ISSUED SUCCESSFULLY

???
runcmd sp=abcdpu01,appl=node,query_node
    
```

図 19. NetView 画面の例

コマンド入力域のサイズの変更

デフォルトでは入力域は 1 行ですが、長いコマンドの場合は複数の行にまたがる場合もあります。NetView のバージョンによっては、input コマンドを使用して 1 行、2 行、または 3 行の入力域を指定できます。この入力域を指定する場合には、次のコマンドを入力します。

```
input n
```

このコマンドで、*n* は使用したい行数で、その値は 1、2、または 3 です。使用している NetView のバージョンでこのコマンドが機能しない場合は、NetView のサポート担当者に連絡してください。

RCF コマンド構文の概要

SPCF コマンドと UCF コマンドのどちらも RCF コマンド構文を使用します。

実行コマンド・スペース =*spname*、 *appl*=コンポーネント、 コマンド・テキスト

NetView は、runcmd ユーティリティーを使用して、コマンド・ストリングをリモート・システムに送信します。このコマンドには以下のパラメーターが含まれ

sp=spname

CS Linux ノードに対応するサービス・ポイント名 (NetView で定義されている) を示します。ホスト NetView の担当者は、この情報を提供できます。

アプリケーション=コンポーネント

NetView がコマンドを送信する先の CS Linux コンポーネントの名前を以下のように示します。

ノード

サービス・ポイント名 *spname* に関連付けられた CS Linux ノード (SPCF コマンドの場合)

ユニックス

サービス・ポイント名 *spname* に関連付けられた CS Linux コンピューター上で実行される UCF デモン・プログラム (UCF コマンドの場合)

コマンド・テキスト

発行されるコマンドのテキストを指定します。SPCF の場合、これは CS Linux コマンド行管理プログラムに発行されるコマンドです。UCF の場合、これは Linux オペレーティング・システム用のコマンドです。For more information about the commands that can be used, see 105 ページの『SPCF で使用される管理コマンドに関する制約事項』 or 107 ページの『使用できるコマンド』.

大文字とエスケープ文字

Linux は大文字と小文字の英字を区別しますが、NetView プログラムはそれを行うことができません。NetView ネットヴァシス コマンドを使用して、大 / 小文字混合の入力を `runcmd` に提供することができますが、CS Linux RCF は、ネットヴァシスが使用中かどうかを判別する方法がありません。RCF は、ホストから受信した英字が元々大文字でも小文字であっても判別できないので、受信された文字は小文字になることを想定しています。また、ホスト文字セットは、一部のコマンドで必要とされる大括弧文字 [および] をサポートしていない可能性があります。

RCF は、以下のように、バックスラッシュ文字 `\` を使用して大文字と大括弧の文字をサポートします。

- コマンド・ストリングに大文字を含めるには、その前に円記号文字を含めてください。円記号が前に付いていない英字は、小文字として解釈されます。
- 大括弧の文字 [と] を含めるには、シーケンス `\(と \)` をそれぞれ使用します。
- 円記号 `\` 自体を組み込むには、その文字を 2 回入力します。

単一の円記号の後に他の英字以外の文字が続く場合、円記号は無視され、文字は変更されません。

いくつかの例を [105 ページの表 3](#) に示します

表 3. RCF コマンドでのエスケープ文字の使用

生成する文字	入力
ABcd	<code>\a\bcd</code>
[]	<code>\(\)</code>
<code>\a</code>	<code>\\ a</code>
<code>\[</code>	<code>\\ \(</code>

Linux コマンド行で通常使用するエスケープ文字は、Linux シェルが特殊文字を解釈できないようにするために、RCF では必要ありません。例えば、Linux コマンド行で入力する場合と同様に、文字 `*` または `$` を含む文字列でエスケープ文字を使用することはできません。また、SPCF を使用して管理コマンドを発行する場合は、次の値を削除のような定数名では大 / 小文字を区別しないことに注意してください。これらの文字をエスケープして、大文字にする必要はありません。

SPCF の使用

SPCF により、NetView コンソールから、CS Linux システムの実行を管理するためのコマンドを発行できます。これらのコマンドは、CS Linux コマンド行管理プログラム `snaadmin` を使用して発行できるコマンドと同じです（「*IBM Communications Server for Data Center Deployment on AIX* 管理コマンド解説書」を参照してください）。

SPCF コマンドの構文については、[104 ページの『RCF コマンド構文の概要』](#)を参照してください。`appl=node` パラメーターのあとに続くコマンド・テキストは、CS Linux コマンド行管理プログラムに対して発行するコマンドで、その形式は、Linux コマンド行で `snaadmin` プログラムに対して指定する場合と同じです。管理コマンドの構文および各コマンドのパラメーターについては、「*IBM Communications Server for Data Center Deployment on Linux* 管理コマンド解説書」を参照してください。

SPCF で使用される管理コマンドに関する制約事項

コマンド行オプション `-i` を使用して、ファイルまたは標準入力からの入力を指定することはできません。すべてのコマンドは、NetView コンソールで直接入力する必要があります。

`query_*` コマンドを使用すると、コマンド行オプションを使用できます `-a` (すべての項目を戻す) および `-d` (詳細情報の戻し) は、Linux コマンド行でコマンドを入力するときと同じ方法で実行します。

セキュリティを提供するために、SPCF から特定のタイプのコマンドのみが許可されるように CS Linux 構成をセットアップすることができます。例えば、リモート・ユーザーは、`query_*` コマンドを発行でき

UCF の使用

ますが、CS Linux コンポーネントのアクティブ化または非アクティブ化は許可されません。以下のコマンド・グループのそれぞれについて、アクセスを別々に制御することができます。

- `define_*`、設定値*、削除 `delete_*`、`add_*`、および `remove_*` コマンド、および `init_node*`
- `query_*` コマンド
- "アクション" コマンド: `びっくり*`、`ストップ_*`、`アクティブ (_)*`、`非アクティブ化 _*`、`and also 開口`、`初期セッション限度`、`変更セッション限度`、`and リセット・セッション限度`

SPCF のセキュリティ・オプションのセットアップについて詳しくは、アクセスの `define_rcf_access` のコマンド *IBM Communications Server for Data Center デプロイメント on Linux* 管理コマンド解説書の説明を参照してください。

SPCF コマンドの例

以下の例は、SPCF を使用して `lu_lu_0_to_3` をフラグメント化する コマンドを発行する方法を示しています。この例では、2 文字のストリングル `$01` と `パブリッシュ 2` で大文字を示すために、円記号文字を使用します。 `snaadmin` プログラムは、このストリングを小文字で受け入れるので、定数名 `3270_display_model_2` の大文字で文字を作る必要はありません。

```
runcmd sp=myspname、 appl=ノード、 define_lu_0_to_3、 lu_name=\l \u$01
```

```
ノード・アドレス =1、 pu_name=\p\u2、 lu_model=3270_display_model_2
```

以下の例は、SPCF を使用して 3 つの要求を照会 コマンドを発行する方法を示しています。 `-a` オプションは、"すべてのエントリを返す" を指定するため、LU 名または PU 名を指定する必要はありません。 `-d` オプションは、"詳細情報を戻します。" は、`list_options` パラメーターを使用してこれを指定する必要はありません。これらの 2 つのオプションは、`snaadmin` プログラム。の場合とまったく同じように働きます。

```
runcmd sp=myspname、 appl=ノード、 -a -d query_lu_0_to_3
```

UCF の使用

UCF を使用すると、NetView オペレーターは、NetView コンソールでコマンド・テキストを入力し、これらのコマンドから出力を表示することによって、CS Linux を実行しているコンピューターで Linux コマンドを発行することができます。この機能は CS Linux に関連するコマンドに制限されません。 [107 ページの『使用できるコマンド』](#) の制限の対象となる場合は、どのようなタイプのコマンドでも発行できます。

UCF を使用すると、リモート・オペレーターは CS Linux コンピューター上のアクティビティをモニターし、問題を診断することができ、場合によっては修正処置を行うことができます。

CS Linux がアクセスの `define_rcf_access` コマンド (*IBM Communications Server for Data Center デプロイメント on Linux* 管理コマンド解説書) を使用して UCF をサポートするかどうかを指定することができます。構成で UCF がサポートされていることを指定する場合、CS Linux は UCF デーモン・プログラムを開始します。ノードが開始されたとき、UCF デーモンは、コマンドごとに新しい Linux シェルを開始し、そのシェルでコマンドを実行することによって、UCF から Linux コマンドを処理します。UCF サポートが組み込まれていない場合、CS Linux はこのプログラムを開始しません。

この構成は、UCF ユーザーの名前を指定します。これは、CS Linux コンピューター上の有効なログイン名でなければなりません。The UCF shell is started using the shell program, login ID, permissions, and . ログイン or . プロファイル specified for that user. (シェル・プログラムを指定しなかった場合は、`/bin/sh` が使用されます。)これは、通常の Linux システム・セキュリティ機能を使用して、UCF ユーザーのファイルおよびコマンドへのアクセスを制限し、そのため UCF から使用可能なコマンドの範囲を制限することができます。ことを意味します。

UCF 構成のセットアップの詳細については、アクセスの `define_rcf_access` のコマンド *IBM Communications Server for Data Center デプロイメント on Linux* 管理コマンド解説書の説明を参照してください。

UCF コマンド構文

UCF コマンドの構文は次のとおりです。

```
runcmd sp=spname, appl=unix, unix_command
```

NetView は runcmd ユーティリティーを使用して、リモート・システムにコマンドを送ります。このコマンドには次のパラメーターがあります。

sp=*spname*

spname を指定します。これは、NetView で定義されている サービス・ポイントの名前です。この情報は、ホスト NetView の担当者から入手できます。

appl=unix

サービス・ポイント名 *spname* に関連した CS Linux コンピューター上の UCF デーモン・プログラムにコマンドを送信するように、NetView に指示します。

unix_command

Linux オペレーティング・システム・コマンドを入力します。このコマンドは、Linux コマンド行で入力する場合と同じように入力します。ただし、大文字または大括弧を指示するためにエスケープ文字を使用する点異なります ([104 ページの『RCF コマンド構文の概要』](#)を参照)。

Linux コマンド行では、Linux シェルが特殊文字の固有の解釈をするのを防止するためにエスケープ文字を使いますが、UCF ではエスケープ文字は必要ありません。例えば、Linux コマンド行では * または \$ の文字を含んだ文字列を入力したいときにエスケープ文字を使用しますが、これは必要ありません。

使用できるコマンド

UCF は、ユーザーとの対話を必要とせずに完了するコマンドと組み合わせて使用するために設計されています (出力の発生の有無とは関係なく)。例えば、*filename* の内容を表示した後に完了するコマンド `catfilename` や、またはエラーが発生しなければ出力なしに完了するコマンド `mvfilename1 filename2` を発行することができます。

UCF コマンドにより生成された出力は、Linux オペレーティング・システム・コマンドの完了時に UCF に戻されます。そのため、次のような制約があります。

- コマンドの完了後に生成された出力は、UCF に戻されません。例えば、バックグラウンドで実行するために、あとに & を付けたコマンドを発行したとすると、UCF は、そのバックグラウンド・コマンドのプロセス ID を示すオペレーティング・システム・メッセージを受け取りますが、そのあとで生成される出力は受け取りません。同様に、UCF を使用してデーモン・プロセスを開始することはできますが、そのプロセスにより生成される出力を参照することはできません。
- UCF は、完了前にユーザーが追加入力をしなければならないコマンドには使用できません (例えば、対話型プロセスを開始する `vifilename` や、ユーザーが停止するまで完了しない `tail -ffilename` などのコマンドには使用できません)。

すべての Linux コマンドは、構成されている UCF ユーザーのログイン ID および許可に従って実行されるので、使用できるコマンドは、UCF ユーザーのログインのアクセス権により制限されます。特に、root またはスーパーユーザーのコマンドは許可されません。詳しくは、[109 ページの『UCF のセキュリティ』](#)を参照してください。

UCF コマンドの例

次に示すのは、NetView から入力する UCF コマンドの例です。

```
runcmd sp=myspname, appl=unix, grep \temp \((ab\)*)\.c >\t\e\m\p.out
```

Linux コンピューターで実行されるコマンドは次のようになります。

```
grep Temp [ab]*.c >TEMP.out
```

Linux システム・コマンドからの出力

コマンドが正常に発行されると、NetView 画面に以下のメッセージが表示されます。

```
=== UNIX コマンドの実行 ===(エラー・メッセージを含む、コマンドからの出力)
```

```
=== UNIX コマンドが完了しました ===
```

これらのメッセージは、NetView 画面に同時に表示されない場合があります。UNIX コマンドを実行中メッセージは、UCF デモン・プログラムがコマンドを受信し、NetView オペレーターに制御を戻すとすぐに表示されます。コマンドからの出力は、作成時に NetView に送信され、一連の個別のメッセージとして表示される場合があります。UNIX コマンドの完了メッセージは、Linux コマンドが終了し、シェルが終了したときに表示されます。

Linux コマンドからの出力にタブ文字が含まれている場合、CS Linux は、NetView に出力を送信する前に各タブをスペース文字に変換します。それ以外の場合は、出力は未変更

前のコマンドがまだ進行中であるときに(つまり、UNIX コマンドの完了メッセージを受信されます。以下のメッセージが表示されます。の前に)コマンドを発行すると、

```
=== 待機コマンド ===
```

2 番目のコマンドはキューに入れられ、前のコマンドが完了した時点で実行されます。

コマンドの取り消し

UCF は、まだ進行中のコマンドをキャンセルする方法を提供します。これは、現在のコマンドの実行を停止したり、さらに入力を行わずに完了できない対話式コマンド(vi ファイル名など)を取り消すために使用できます。これは、Ctrl + C は、端末上で実行されているプロセスを停止するか、または Linux 強制終了コマンドを使用してプロセスを停止します。のような割り込みシーケンスを使用することと同じです。

CS Linux は、現在実行中のコマンドの取り消しに加えて、キューに入れられたコマンドをすべて取り消します。

コマンド構文は Linux コマンドの場合と同じですが、コマンド・テキストの代わりに ux-キャンセル というストリングが指定されています。例えば、

```
runcmd sp=myspname、 appl=unix、 ux-cancel
```

未処理の各コマンド(現在実行中のコマンドおよびキューに入れられたコマンド)ごとに、次のメッセージが表示されます。

```
=== UNIX コマンドがキャンセルされました ===
```

このメッセージは、コマンドが実行されていた Linux シェルが停止していることを示しています。追加の Linux コマンドは、必要に応じて発行できます。

コマンドが Linux コンピューター上でデモン・プロセスを開始する場合、このプロセスは ux-キャンセルによって停止されない可能性があります。このようなプロセスを明示的に停止するには、Linux 強制終了コマンド(端末上または UCF を使用して)を使用する必要がある場合があります。

ux-キャンセルを使用すると、UCF は次のメッセージを表示します。が実行されている UCF コマンドがない場合

```
未解決コマンドなし
```

この場合、ux-キャンセル コマンドは無視されます。アクションは必要ありません。このメッセージは、ux-キャンセル コマンドが、直前のコマンドが終了した後で、UNIX コマンドの完了メッセージを受信される前に発行された場合に表示されます。

UCF のセキュリティ

UCF は、リモート・オペレーターが Linux コンピューターでコマンドを発行し、そのコマンドからの出力を受け取れるようにするためのものなので、セキュリティ 関連事項について考慮することが重要です。例えば、オペレーターがプライベート情報にアクセスしたり、他のユーザーを妨害するような Linux コマンドを発行したりできないようにする必要があります。

CS Linux 構成情報には、UCF ユーザーとして特定の Linux システム・ユーザー名が指定されています。この名前は、CS Linux コンピューター上の有効なログイン ID でなければなりません。すべての UCF コマンドは、このユーザー ID に基づいて実行されます。つまり、このユーザーのアクセス許可に基づいて実行されます。

一般に、Linux が提供する通常のセキュリティ機能を使用して、UCF ユーザーがアクセスできるコマンドを制限することにより、妥当と認められるコマンドのみを UCF から使用できるようにします。そのためには次に示すガイドラインが役に立ちます。

- UCF ユーザー名には UCF 専用の名前を使用するようにし、他の目的にも使用する既存のログイン名は使用しないでください。このようにすれば、このユーザーの権限を定義するときに、UCF 用として妥当と認められる権限のみを定義することができます。また、UCF を使用して開始されたプロセスを識別しやすくなります。
- UCF ユーザーがユーザー ID またはグループ ID を変更するとき、その対象となるユーザーおよびグループに制約を設けることが必要な場合もあります。特に、次のことを UCF ユーザーに許可しないように注意してください。
 - root のユーザーまたはスーパーユーザーになること。
 - グループ ID system を使用すること。snaadmin プログラムへアクセスできます。(この章で述べたように、このプログラムの機能にアクセスする場合は、UCF ではなく SPCF を使用してください。)

第 10 章 CS Linux クライアント / サーバー・システムの管理

CS Linux は、単一の Linux システム上のすべての SNA コンポーネントとアプリケーションを備えたスタンダード・システムとして稼働することも、クライアント/サーバー・ドメインの一部として稼働することもできます。クライアント/サーバー・ドメインに両方のサーバーが含まれる(SNA ノード) および IBM リモート API クライアント(これは、サーバーを介した SNA 接続にアクセスできます)。

複数の CS Linux サーバーを持つドメインでは、1 つのサーバーが CS Linux ドメイン構成ファイルの制御コピーを保持します。このサーバーは、コントローラー・サーバーと呼ばれます。ドメイン内の他のサーバーをバックアップ・サーバーとして定義できます。ドメイン構成ファイルは、開始時にバックアップ・サーバーにコピーされるか、または制御コピーが変更されたときに、すべてのバックアップ・サーバーが最新情報のコピーを保持するようにコピーされます。

リモート API クライアントは、AIX、Linux、Linux for Power、Linux for IBM Z、または Microsoft Windows を実行するコンピューターにすることができます。また、リモート API クライアントは、AIX、Linux、または Windows システム上のコンテナで実行することもできます。これは、Java for-CPI-C アプリケーションを使用する Web アプリケーション・サーバーのようなコンテナで実行されるアプリケーション用の SNA API を提供します。ドメイン・サーバーがコンテナの IP アドレスを解決できるようにするには、コンテナに DNS 名を指定する必要があります。(カーネルの依存関係のために、CS Linux サーバーをコンテナにインストールすることはできません。)

サーバーおよびクライアントは、TCP/IP を使用して CS Linux ドメイン間で通信します。IPv4 と IPv6 の両方のアドレッシングがサポートされます。クライアントは同時に 1 つ以上のサーバーにアクセスことができ、必要に応じて並行アプリケーションを実行できます。クライアント/サーバー構成のネットワーク要件については、[113 ページの『IP ネットワーキングの要件』](#)を参照してください。

クライアントとサーバーの間で使用される TCP/IP 接続は、VM の下で稼働するサーバー間の物理 LANS、WAN、または仮想パスにわたって流れることがあります。CS Linux の資料では、LAN という用語がこれらすべてに使用されています。

UNIX

AIX または Linux 上のリモート API クライアントの場合は、CS Linux ネットワークおよびサーバーに関する情報を指定する必要があります。この機能について、およびクライアントでの CS Linux ソフトウェアの使用可能化および使用不可化に関する説明については、[132 ページの『AIX または Linux 上の Remote API Client の管理』](#)を参照してください。

すべての管理コマンドは、サーバー上で発行できます。ただし、どのコマンドを AIX および Linux クライアントで発行できるかについては、制約事項があります。

- 任意の 照会 または 状況 コマンドは、AIX または Linux クライアント上で使用します。を発行できます。
- IBM Communications Server for Data Center Deployment on Linux 管理コマンド・リファレンスで定義されているその他のいくつかの管理コマンドは、IBM Remote API Client から発行できることを明示的に説明しています。それ以外の場合、これらのコマンドはサーバーからのみ使用可能です

WINDOWS

Windows クライアントの場合、クライアント・ソフトウェアを使用可能にするために CS Linux が使用できる情報を指定する必要があります。Windows クライアント上に呼び出し可能 TP を使用する予定の場合は、TP に関する情報も提供する必要があります。これらの機能について、および Windows クライアントでの CS Linux ソフトウェアの使用可能化および使用不可化に関する説明については、[115 ページの『Windows でのリモート API クライアントの管理』](#)を参照してください。

IBM Communications Server for Data Center Deployment on Linux Administration Command リファレンスに定義されている管理コマンドは、Windows クライアントから発行できません。

クライアント/サーバー構成の変更

「IBM Communications Server for Data Center Deployment on Linux 入門」の説明に従って CS Linux ソフトウェアをインストールすると、最初は、スタンドアロン・モードとしてインストールされます (単一の Linux コンピューターにすべてのコンポーネントがあります)。CS Linux をクライアント/サーバー・システムとして実行する場合は、1 つのサーバーをコントローラー・サーバーとして構成し、その他のサーバーをバックアップ・サーバーとして構成できます。(コントローラー以外のすべてのサーバーをバックアップ・サーバーとして構成することをお勧めします。)

CS Linux には、サーバーをクライアント/サーバー・ドメインの一部にするためのコマンド行アプリケーション・プログラム `snanetutil` があります。これを行うために、それぞれのサーバーで以下のコマンドを使用します (コントローラー・サーバーから始めます)。

```
sna stop
snanetutil controller_name [domain_name]
sna start
```

`snanetutil` コマンドのパラメーターは次のとおりです。

controller_name

サーバーが属することになるドメインのコントローラー・サーバーの名前。サーバーを既存のドメインに移動する場合、この名前はそのドメイン内にある既存のコントローラー・サーバーの名前と一致している必要があります。

domain_name

サーバーが属することになるドメインの名前。このパラメーターはオプションです。このオプションを指定しない場合、CS Linux はデフォルトのドメイン・ネーム `ibmcs_domain` を使用します。

コントローラー以外の各サーバーをバックアップ・サーバーとして構成するには、以下のコマンドを発行します。これは、それぞれのバックアップ・サーバーで行うことも、コントローラー・サーバーで行うこともできます。しかし、いずれにしても、CS Linux ソフトウェアはコントローラー・サーバー上で実行する必要があります。

```
snaadmin add_backup, backup_name=server_name
```

`server_name` は、バックアップ・サーバーとして追加するサーバーの名前です。

CS Linux は、始動時にローカル・ホスト名設定を使用してノードを識別します。代わりに、DNS 別名を使用する必要がある場合もあります。CS Linux サーバー名を DNS 別名に設定するには、環境ファイル `/etc/opt/ibm/sna/environment` で、次のステートメントを設定します。

```
export SNA_SERVER_NAME=DNS_alias_name_for_server
```

また、`snanetutil` プログラムを使用して、既存のドメインからサーバーを移動し、それがスタンドアロン・システムとして実行されるようにすることもできます。

注: クライアント/サーバー・システムとしての CS Linux の稼働を停止して、単にスタンドアロン・ノードとして使用する必要がない限り、このオプションは使用しないでください。すべてのサーバーを既存のドメインから除去すると、そのドメインに残ったクライアントは SNA リソースにアクセスできなくなります。

サーバーがスタンドアロン・システムとして稼働するように、それをドメイン外に移動するには、次のコマンドを使用します。

```
snanetutil -d
```

異なるドメインへのクライアントの移動

`snanetutil` プログラムを使用して、異なるクライアント/サーバー・ドメイン間でサーバーを移動できます。ドメイン間でクライアントを移動するには、クライアント構成を変更する必要があります。

Windows 上の移動するそれぞれの Remote API Client 上で、クライアント構成ユーティリティを使用して、`domain` パラメーターを変更して新しいドメイン・ネームに一致させる必要があります。詳細については、[118 ページの『Windows 構成上のリモート API クライアント』](#)を参照してください。

AIX または Linux 上の移動するそれぞれの Remote API Client 上で、クライアント・ネットワーク・データ・ファイルの Configuration セクションの *domain* エントリーを変更して、新しいドメイン・ネームに一致させる必要があります。詳細については、[133 ページの『クライアント・ネットワーク・データ・ファイル \(sna_clnt.net\)』](#)を参照してください。

IP ネットワーキングの要件

Remote API Client は、TCP/IP を使用して、または WebSphere サーバーを介した HTTPS を使用して CS Linux サーバーと通信できます。HTTPS 接続の使用に関する詳細は、[115 ページの『リモート API クライアントの HTTPS アクセス』](#)を参照してください。

Remote API Client を実行するには、その前に、ネットワーク内のクライアントとサーバーの両方に対して TCP/IP ポート・アドレスを構成する必要があります。デフォルトのポート割り当てを使用して問題が生じた場合は、[114 ページの『IP ポート番号のセットアップ』](#)で説明するように競合を解決する必要があります。

また、[114 ページの『LAN アクセス・タイムアウト』](#)で説明するように、クライアントが CS Linux の使用を終了したときに TCP/IP 接続が自動的にドロップされるようにクライアントを設定できます。

ドメイン・トポロジー内の変更をより素早く検出できるようにするために、サーバー・ペア間でサーバー間キープアライブ・ポーリングが使用されます。デフォルトのポーリング時間は 5 秒です。これは、環境ファイル `/etc/opt/ibm/sna/environment` 内のパラメーター `SNA_SLIM_SERVER_POLL=nnn` (0-7200 秒) を使用して変更できます。値 0 は、ポーリングがないことを示します。

IPv4 および IPv6 アドレッシング

CS Linux クライアント/サーバー・ドメインでは、IPv4 アドレスまたは IPv6 アドレスを使用できますが、ドメイン内のすべてのサーバーに同じアドレッシング・フォーマット (IPv4 または IPv6) を使用する必要があります。

- サーバーに IPv4 を使用する場合は、クライアントにも IPv4 を使用しなければなりません。
- サーバーに IPv6 を使用する場合は、クライアントには IPv6 または IPv4 のいずれも使用できます。

IPv4 および IPv6 アドレス指定のセットアップと使用の方法について詳しくは、「[IBM Communications Server for Data Center Deployment on Linux 入門](#)」を参照してください。

クライアント/サーバー構成でのホスト名

CS Linux は、サーバーとクライアントの間の内部通信に完全修飾 IP ホスト名を使用します。通常、ローカル・システムは、ネットワーク構成 (DNS など) からこれらの名前を判別できます。これが可能でない場合は、構成内でホスト名が必要なときは常に、別名 (`newbox` など) ではなく完全修飾名 (`newbox.this.co.uk` など) を使用する必要があります。

各コンピューターのローカル・サーバー名は、`/etc/hosts` ファイルから取得されます。このファイル内のエントリーは、最初に IP アドレスを指定し、次に完全修飾名を指定し、最後に別名を指定する必要があります。例えば、次のようになります。

```
9.42.108.28    newbox.this.co.uk    newbox
```

サーバーがマルチホームである場合 (例えば、2 つ以上の TCP/IP ネットワーク・インターフェースが異なる IP アドレスで表示されるようになっている場合)、`/etc/hosts` ファイル内のエントリーはすべてのアドレスに同じ IP 名を指定して、すべてのネットワーク・インターフェースで名前を正しく解決できるようにする必要があります。例えば、

```
9.42.108.28    newbox.this.co.uk    newbox
9.42.80.127    newbox.this.co.uk    newbox
```

また、行マルチオンを `/etc/host.conf` ファイルは、サーバーがマルチホームであることを示します。に組み込む必要があります。

IP ポート番号のセットアップ

CS Linux では TCP/IP と UDP/IP の両方の通信を使用クライアント / サーバーのデータを LAN 経由で送信します。デフォルトでは、両方のタイプの通信にポート番号 1553 が使用されます。ほとんどのインストール済み環境では、このポート番号は適切なものである必要があります。変更する必要はありません。

CS Linux ソフトウェアを使用可能にする際に問題が発生した場合は、エラー・ログ・ファイルで、CS Linux が使用するポート番号が、別のプログラムで使用されているポート番号と競合していることを示すメッセージを確認してください。このようなメッセージが見つかった場合は、以下の手順を実行します。

1. エラーが発生したコンピューター上の `/etc/services` ファイルを調べて、TCP/IP または UDP/IP 通信のいずれかのポート番号 1553 を使用して、別のプログラムがリストされているかどうかを確認してください。この場合は、別のポートを使用するようにもう一方のプログラムを変更してください。
2. これを行うことができない場合、またはポート 1553 を使用しているプログラムがリストされていない場合は、どのプログラムでも使用されているように、ファイルにリストされていない別のポート番号を見つけてください。同じドメイン内の他のすべての CS Linux コンピューター上の `/etc/services` ファイル。他のコンピューターでその番号が使用されていないことを確認します。の検査
3. ドメイン内の各コンピューター上の `/etc/services` ファイルで、以下の形式で 2 行を追加します。の場合、

```
sna-cs      nnnn/tcp
sna-cs      nnnn/udp
```

`nnnn` 項目は新しいポート番号です。これは、CS Linux ドメイン内のすべてのコンピューターで同じ値に設定する必要があります。

4.

CS Linux ドメインに Windows クライアントが含まれている場合は、各 Windows コンピューター上のサービスファイルに同じ 2 行を追加します。サービスファイルは Linux ファイルと同じフォーマットになっており、通常は Windows TCP/IP ソフトウェアのホーム・ディレクトリーに保管されます。必要に応じて、詳細については Windows TCP/IP の資料を参照してください。

5. CS Linux サーバーおよびリモート API クライアント・ソフトウェアを再度使用可能にします。

注: ファイアウォールを使用して、無許可アクセスを防止するために、サーバー上のポート 1553 (またはクライアント/サーバー通信用に指定した新しいポート番号) を保護することをお勧めします。TCP および UDP の両方のトラフィックは、他の CS Linux サーバーおよびリモート API クライアントとの間で許可される必要がありますが、ポートへのアクセスは他のコンピューターに許可されていない必要があります。

LAN アクセス・タイムアウト

接続料金が支払われるネットワークを介してクライアントが CS Linux サーバーと通信している場合は、クライアント上のアプリケーションが CS Linux リソースを使用して停止した後で、クライアントからの TCP/IP 接続が自動的にドロップされるようにすることができます。これにより、クライアント上の SNA ソフトウェアが自動的に使用不可にされることはありません。アクティブのまま、後でアプリケーションが必要とする場合は、サーバーとの接続を再確立しようとします。

`_l_access_timeout` パラメーター (AIX または Linux 上のリモート API クライアント用の `sna_clnt.net` ファイル、または Windows 上のリモート API クライアントのレジストリー) を使用すると、クライアント上の SNA ソフトウェアを使用不可にすることができます。クライアント上で指定された時間に以下のいずれのイベントも発生しなかった場合、TCP/IP 接続はドロップされます。

- APPC または CPI-C 会話がアクティブである (または会話を開始しようとする)
- アクティブな LUA セッション
- Windows クライアントからの CSV TRANSFER_MS_DATA verbsfrom a
- MS verb (Linux クライアントのみ)
- NOF verb (query_central_ロガー または すべての照会ノード verb) を除く)

- Administration commands (except the following events, which do **しない** cause the client to restart the connection):
 - クライアントによって記録されたエラーまたは監査メッセージ (これらは、中央ロギングが使用されている場合でも、クライアント上でローカルにログに記録されます)
 - 管理コマンド query_central_ロガー または すべての照会ノード (これらは、TCP/IP 接続が切断される前に使用可能であった情報を戻し、したがって、LAN の現在の状況と一致しない場合があります)。
 - NOF verb query_central_ロガー または すべての照会ノード (同等の管理コマンドの場合と同様)

特に、SNA ソフトウェアを使用可能にしても、指定されたタイムアウト内にクライアント上で CS Linux アプリケーションを開始しない場合は、TCP/IP 接続がドロップされます。

TCP/IP 接続がダウンしている間にこれらのイベントのいずれかが発生すると、クライアントは、[133 ページの『クライアント・ネットワーク・データ・ファイル \(sna_clnt.net\)』](#)の *パラメーターおよびサーバー名 [121 ページの『Servers』](#)パラメーターの説明に従って、サーバーに接続する試みを再始動します。

このクライアント上で呼び出された TP の着信接続は、TCP/IP 接続がダウンしている間は受け入れられません。接続は、ターゲット・システムが非アクティブの場合と同じように拒否されます。これは、クライアント上の他のアプリケーションが実行されておらず、TCP/IP 接続がタイムアウトになった場合に、クライアント上で自動的に開始された TP を使用できないことを意味します。しかし、TP によって発行された Receive_Allocate verb は TCP/IP 接続を再確立するので、クライアント上のオペレーター開始 TP はいつでも使用できます。

リモート API クライアントの HTTPS アクセス

リモート API クライアントが HTTPS を使用して CS Linux サーバーに接続するクライアント/サーバー・システムを実行している場合は、これらのクライアントからサーバーへの HTTPS アクセスを提供するために、WebSphere Application Server を実行するコンピューターが必要になります。このサーバーをインストールして構成する方法については、[Linux スタートアップ・ガイド](#)でのデータ・センター・デプロイメントのための *IBM Communications Server* を参照してください。

新しいサーバーを CS Linux ドメインに追加し、リモート API クライアントが HTTPS を使用してこれらのサーバーにアクセスできるようにしたい場合は、これらのサーバーを組み込むように WebSphere サーバー構成ファイルを更新する必要があります。このファイルの名前は `snahttpsrv.cfg` で、ユーザー・インストール・ルート 環境変数で指定されたディレクトリーにある WebSphere サーバー上に保管されます。この場所がわからない場合は、以下の手順を実行します。

1. WebSphere 管理コンソールを開始します。
2. 管理コンソールのメニュー・バーで「環境」、「WebSphere 変数の管理」を選択します
3. このリストにある ユーザー・インストール・ルート 変数を探し、その値 (WebSphere サーバー上のディレクトリーのパス) をメモします。環境変数のリストは複数のページにまたがる場合があるため、「次へ」ボタンを使用してリスト内をスクロールする必要がある場合があります。

テキスト・エディターを使用して構成ファイルを編集し、HTTPS を使用してリモート API クライアントによってアクセスできるすべての CS Linux サーバーのリストを組み込みます。各サーバーは、以下の形式で、ファイルの個別の行に指定する必要があります。

サーバー = `servername.domainname.com`

新規サーバーにアクセスするリモート API クライアントごとに、クライアント・ネットワーク・データ・ファイル内のサーバーのリスト (または Windows クライアント用の Windows レジストリー) に新しいサーバー名を追加する必要もあります。この章の後半では、該当するクライアント・タイプのセクションを参照してください。

Windows でのリモート API クライアントの管理



CS Linux は、Microsoft Windows を実行するマシンが CS Linux ドメイン内のクライアントとして機能するようにします。CS Linux クライアント・ソフトウェアには、Microsoft Host Integration Server と互換性のある API ライブラリー、Windows Open Systems Architecture (WOSA) が含まれています。および、IBM パーソナル・コミュニケーションズおよび Communications Server for Windows によって提供されるインターフェース。これにより、これらの実装に対して作成されたアプリケーションを、Windows 上の Remote API Client 上で変更しない

Windows 上のリモート API クライアントは、以下の WOSA API をサポートします。

- Windows APPC
- Windows CPI-C 型
- Windows LUA
- Windows CSV

Windows SNA API について詳しくは、Microsoft Host Integration Server に付属の資料を参照してください。

Windows 上で Remote API Client が必要とする SNA ネットワーク情報、およびその他の情報は、Windows レジストリーに保持されます。

クライアントで CS Linux アプリケーションまたはエミュレーション・プログラムを使用するには、その前にクライアントを使用可能にする必要があります。詳しくは、[116 ページの『Windows 上の Remote API Client を使用可能に設定する』](#)を参照してください。クライアントが使用可能になっていると、CS Linux 機能にアクセスするために、CS Linux を実行しているサーバーに TCP/IP ネットワーク経由で連絡します。

クライアントの操作は、Windows レジストリー内の情報によっても制御されます。Windows レジストリーには、以下の情報が含まれます。

- Windows でのリモート API クライアントに固有の構成情報
- クライアントがアクセスできるサーバー
- クライアント上で実行されているアプリケーションのロギングおよびトレースのオプション
- クライアント上で実行される CPI-C および CSV アプリケーションの追加オプション
- クライアント上で実行できる呼び出し可能 TP (APPC または CPI-C)

最も一般的に使用されるパラメーターは、クライアント構成ユーティリティーを使用して変更することもできます。これは、クライアント構成ユーティリティーを変更するための推奨方法です。詳しくは、[118 ページの『Windows 構成上のリモート API クライアント』](#)を参照してください。

注: クライアントが HTTPS を使用してサーバーにアクセスする場合は、クライアントを使用する前に、クライアント構成を変更して、これらのサーバーの名前と、それらのサーバーへの HTTPS アクセスを提供する WebSphere サーバーを指定する必要があります。詳しくは、[118 ページの『Windows 構成上のリモート API クライアント』](#)を参照してください。

Windows 上の一部のレガシー SLI アプリケーションは、SNA スタックの完全実装を処理するために作成されました。ここでは、戻りコードが COMM_SUBSYSTEM_XXX の場合、SNA スタックがもはや稼働していないか、使用可能でないことを示します。リモート API クライアント / サーバーへのマイグレーション時にこのメッセージが発生する可能性があるのは、TCP/IP 接続の障害が発生したときですが、ソケットのリカバリーが可能な場合です。このようなアプリケーションがこの状態を「致命的」として処理しないようにするには、これらの 3 つの戻りコードを LUA_SESSION_FAILURE にマップするように Remote API Client を構成することができます。これを構成するには、ファイルをリモート API クライアント・インストール・ディレクトリー内の slicommap (通常は c:\ibmcs\w64cli)。してください

Windows 上の Remote API Client を使用可能に設定する

Windows 上の Remote API Client は、Windows サービスとして実行されます。インストール・プログラムにより、コンピューターが始動するとクライアントが自動的に開始されるように構成されます。必要に応じて、次のどちらかの方法でクライアントを手動で開始できます。

- 「コントロール・パネル (Control Panel)」の「管理ツール (Administrative Tool)」にある「サービス (Services)」アプレットから、クライアントを開始します。

- コマンド・ウィンドウ、または「開始」/「ファイル名を指定して実行」アイコンから、`net start sxclient` を入力します。

クライアントは、Windows レジストリーの情報を使用して CS Linux を実行するサーバーを見つけます。これらの情報はクライアント構成ユーティリティを使用して定義され、これについては、[118 ページの『Windows 構成上のリモート API クライアント』](#)に説明されています。

注: Microsoft Windows Vista を使用する場合は、管理者権限のあるコマンド・プロンプトから `net start sxclient` を実行する必要があります。このコマンド・プロンプトにアクセスするには、右マウス・ボタンで「コマンド プロンプト」アイコンをクリックし、「管理者として実行」を選択して、プロンプトに管理者パスワードを入力します。

Windows でのリモート API クライアントの状況の表示

Client Monitor は、マウス・ポインターを移動したときに、クライアントの状況を表示するアイコンをシステム・トレイに配置します。モニターは、コンピューターの始動時に自動的に実行されるようにセットアップされますが、以下のいずれかの方法で手動で実行することもできます。

- 「スタート」メニューから、Windows プログラム・グループのリモート API クライアントから「クライアント・モニター」を選択します。
- コマンド・プロンプトから、クライアント・ソフトウェアがインストールされているディレクトリーに移動して、`sxclappl` コマンドを実行します。

クライアント・モニターは、通常、クライアント・ソフトウェアのインストール時に選択した言語で情報を表示します。この情報を別の言語で表示したい場合は、コマンド・プロンプトからクライアント・モニターを開始することによって、クライアント・ソフトウェアがインストールされているディレクトリーに移動し、次に、`sxclappl` コマンドを実行する前に、希望する言語に対応するサブディレクトリーに移動することができます。例えば、フランス語で情報を表示するには、クライアント・ソフトウェアがインストールされているディレクトリーの下の `fr_FR` サブディレクトリーにある `sxclappl` コマンドを実行します。

クライアント・モニターは、状況を以下のいずれかとして表示します。

非アクティブ

クライアントは開始されていません。

未接続

クライアントは開始されていますが、まだサーバーとの接続を確立していません (または連絡先が失われています)。

サーバー名 (nnn)

クライアントは指定されたサーバーに接続されています。nnn は、この Windows クライアントを使用しているアプリケーション (APPC、LUA、PCOMM など) の数を指定します。サポートできるアプリケーションの最大数は、Linux 管理コマンド・リファレンスの IBM Communications Server for Data Center デプロイメントで説明されているパラメーター `maximum_process_count` によって指定されません。`sxclsfce.exe` サービスおよびモニター・プログラム自体がアプリケーションとしてカウントされるため、最小値は 2 のアプリケーションが実行されていない場合に発生します。

Windows 上の Remote API Client を使用不可に設定する

クライアントを使用不可に設定する前に、クライアント上のすべての CS Linux アプリケーション (3270 および 5250 エミュレーション・プログラム、または CS Linux API を使用するアプリケーション) が停止していることを確認してください。

クライアントを使用不可に設定するには、次のいずれかの方法でクライアント・サービスを停止します。

- 「コントロールパネル (Control Panel)」の「管理ツール (Administrative Tools)」にある「サービス (Service)」アプレットから、クライアント・サービスを停止します。
- コマンド・ウィンドウ、または「開始」/「ファイル名を指定して実行」アイコンから、`net stop sxclient` を入力します。

これにより、Windows Terminal Services を実行するコンピューター上で、すべてのユーザーがクライアントを使用できなくなります。

注: Microsoft Windows Vista を使用する場合は、管理者権限のあるコマンド・プロンプトから `net stop sxclient` を実行する必要があります。このコマンド・プロンプトにアクセスするには、右マウス・ボタンで「コマンドプロンプト」アイコンをクリックし、「管理者として実行」を選択して、プロンプトに管理者パスワードを入力します。

Windows 構成上のリモート API クライアント

Windows 上のリモート API クライアントでは、構成情報は Windows レジストリーに保持されます。このレジストリーには、SNA ネットワーク情報(AIX または Linux 上のリモート API クライアント上のクライアント・ネットワーク・データ・ファイルに保持されている情報と類似)が含まれています。また、Windows 上の Remote API Client に固有の追加の構成情報が含まれています。

注: CPI-C アプリケーション(ローカル TP 名およびローカル LU 別名)の構成情報は、環境変数またはレジストリーのいずれかで指定できます。Windows Terminal Server を使用していて、異なるローカル LU を使用して同じアプリケーションの複数のコピーを実行する必要がある場合は、環境変数を使用する必要があります。詳しくは、[128 ページの『Appl_Name』](#)を参照してください。

クライアント構成ユーティリティー最も一般的に使用されるクライアント構成パラメーターを変更する簡単な方法を提供し、これらのパラメーターを変更するための推奨方法を提供します。このプログラムは、以下のいずれかの方法で開始することができます。

- 「スタート」メニューから、Windows プログラム・グループのリモート API クライアントから構成ユーティリティーを選択します。
- コマンド・プロンプトから、クライアント・ソフトウェアがインストールされているディレクトリーに移動して、`sxclconf` コマンドを実行します。

通常、構成ユーティリティーは、クライアント・ソフトウェアのインストール時に選択した言語に情報を表示します。If you prefer to see this information in a different language, you can do this by starting the Configuration Utility from a command prompt: change to the directory where the client software is installed, and then to the subdirectory corresponding to your preferred language, before running the `sxclconf` command. 例えば、フランス語で情報を表示するには、クライアント・ソフトウェアがインストールされているディレクトリーの下に `fr_FR` サブディレクトリーにある `sxclconf` コマンドを実行します。

プログラムは、初期インストール・プロセスに表示されたのと同じ構成ウィンドウを表示します。これらの構成パラメーターを変更する方法の詳細については、[IBM Communications Server for Data Center Deployment の Windows インストール・チャプタにあるリモート API クライアント](#)を参照してください。

注: クライアント構成パラメーターを変更した後、変更を有効にするには、クライアントを停止して再始動する必要があります。これを行う方法の詳細については、[117 ページの『Windows 上の Remote API Client を使用不可に設定する』](#) and [116 ページの『Windows 上の Remote API Client を使用可能に設定する』](#)を参照してください。

レジストリーでは、以下のキーのサブキーの下に構成された値に情報が含まれています。

```
\\HKEY_LOCAL_MACHINE\SOFTWARE\SNA Client\SxClient\Parameters
```

各レジストリー・サブキーの可能な値は、以下のとおりです。

```
Configuration
domain = domain_name
maximum_process_count = nn
maximum_header_count = nn
maximum_element_count = nn
invoked_tps = YES | NO
lan_access_timeout = nn
poll_timer = 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
CSVTLBG = table_G_filename

```

注: ドメイン = ドメイン名 値は、レジストリーで必要な唯一の値です。

以下のセクションで構成パラメーターについて説明します。パラメーターが値 そうだ または 違う。を取る場合は、Y または y で始まるストリングはすべて そうだと解釈され、N または n で始まるストリングはすべて 違う。と解釈されます。

Configuration

Configuration サブキーは、次のようにクライアントに関する構成情報を指定します。

domain

この値のレジストリー・データ・タイプは REG_SZ です。

domain_name 値は、クライアントのインストール時に指定された CS Linux LAN のドメイン・ネームを示します。この行は必要です。

maximum_process_count

この値のレジストリー・データ・タイプは REG_SZ です。

このクライアントで同時に実行できる APPC、CPI-C、LUA、および NOF アプリケーションの最大総数を指定します。

このパラメーターはオプションです。デフォルト値は 240 で、通常はこの値で十分と考えられます。IPC 制御ブロックの割り振りの失敗を報告するエラー・メッセージが表示された場合は、このパラメーターを指定して、最大プロセス・カウントを増やしてください。指定できる最大値は 1024 です。

maximum_header_count、maximum_element_count

これらの値のレジストリー・データ・タイプは REG_SZ です。

この 2 つのパラメーターはオプションです。デフォルト値は 20000 と 30000 で、通常はこれで十分と考えられます。サポート担当者の指示がない限り、これらのパラメーターに値を指定する必要は通常ありません。

invoked_tps

この値のレジストリー・データ・タイプは REG_SZ です。

値は次のいずれかです。

YES

このクライアントは、呼び出された TP (RECEIVE_ALLOCATE を実行する APPC TP、または Accept_Conversation か Accept_Incoming を実行する CPI-C アプリケーション) の実行に使用されます。この場合は、さらにこのクライアント上で TP を定義する必要が生じることがあります。詳細は、[75 ページの『TP の定義』](#) または [163 ページの『付録 B コマンド行からの呼び出し可能 TP の構成』](#) を参照してください。

NO

このクライアントは、呼び出された TP の実行には使用されません。

この行はオプションです。指定しない場合、デフォルトは NO です。

lan_access_timeout

この値のレジストリー・データ・タイプは REG_SZ です。

クライアント上のアプリケーションが CS Linux リソースを使用していないときに、クライアントからサーバーへの IP または HTTPS 接続をアクティブに保つ時間を秒数で指定します。詳しくは、[114 ページの『LAN アクセス・タイムアウト』](#) を参照してください。

有効な範囲は 0 から 65535 です。最小タイムアウトは 60 秒です (これより小さい値は、60 秒に丸められます)。接続を迅速に活動停止するには、クライアントを使用不可に設定します。

このパラメーターはオプションです。指定しない場合、デフォルトでタイムアウトは設定されず、クライアントが稼働しているかぎり接続はアクティブに保たれます。

poll_timer

このパラメーターは、クライアントがファイアウォールを通過してサーバーに接続する場合にのみ使用されます。

この値のレジストリー・データ・タイプは REG_SZ です。

クライアントのアプリケーションが一定の期間、CS Linux リソースを使用しなかった場合は、アクティビティがないため、ファイアウォールによって接続が終了される可能性があります。このパラメーターによって、クライアントには接続をアクティブにしておくために、一定間隔でサーバーにポーリング・メッセージを送信することが強制されます。サーバーはこのメッセージを他の用途では使用しません。

ポーリング・メッセージの間隔を分単位で指定します。範囲は、1 から 1440 (1 分から 1 日) です。標準的には値 10 が、ほとんどのファイアウォールに適しています。

ポーリング・メッセージを指示せず、アクティビティがないことを検出した場合にファイアウォールが接続をドロップできるようにするには、0 (ゼロ) を指定します。

このパラメーターはオプションです。指定されない場合、クライアントはポーリング・メッセージを送信しません。

broadcast_attempt_count

この値のレジストリー・データ・タイプは REG_SZ です。

クライアントがブロードキャスト方式を使用してサーバーに接続する場合 (121 ページの『Servers』で説明する * エントリーによって指定)、このパラメーターは、サーバーへの接続を 1 回試行する間に行うブロードキャストの最大回数を指定します。

有効な範囲は 1 から 65535 です。最小値は 1 です。これより大きな値を指定すると、クライアントはサーバーに接続するかこのカウントが満了するまで、10 秒ごとに再試行します。サーバーに接続できずにカウントが満了した場合、クライアントは指定されたサーバーへの接続を試行します (121 ページの『Servers』を参照)。

このパラメーターはオプションです。指定しない場合、デフォルトは 5 です。

server_lost_timeout

この値のレジストリー・データ・タイプは REG_SZ です。

クライアントがサーバーへの接続を失って再接続する必要があるか、ブロードキャストまたは指定されたサーバー (121 ページの『Servers』を参照) を使用してサーバーに接続できなかった場合に、このパラメーターはクライアントがサーバーへの接続を試行するまで待つ秒数を指定します。クライアントがサーバーへの接続を失った場合、CS Linux はタイムアウト期間が満了するまで待たずに、5 秒から指定されたタイムアウトまでの間でランダムに決められた期間が経過すると再試行します。これは、多数のクライアントが同時にサーバーへの接続を試行することによって、ネットワーク・トラフィックのバーストが起こることを防ぐためです。

このパラメーターはオプションです。有効な範囲は 5 から 65535 です。指定しない場合、デフォルトは 200 (秒) です。

client_start_timeout

この値のレジストリー・データ・タイプは REG_DWORD です。

クライアントが始動してサーバーへの接続を試行している間、アプリケーションが待つ時間を秒数で指定します。0 から 300 の範囲の値が有効です。この範囲外の値は、強制的に範囲内に修正されます。デフォルト値は 10 秒です。

アプリケーションとクライアントの両方をシステム起動時に開始するように構成した (「スタートアップ」フォルダーに入れるか、自動開始サービスにすることによって) 場合に、このパラメーターを使用してイベントを制御できます。アプリケーションは、このフィールドに指定した秒数だけ待ち、クライアントを先に始動できるようにします。これにより、クライアントがサーバーに接続してアプリケーションに必要なリソースを提供するので、これらのリソースの不足によってアプリケーションが失敗することを事前に防止できます。

Servers

Servers サブキーには、クライアントがリソースにアクセスすることのできる CS Linux 構成サーバーに関する情報が含まれています。このリストには、クライアントと同じドメイン内のコントローラー構成サーバーとすべてのバックアップ・サーバーの名前が含まれます。コントローラー・サーバーとバックアップ・サーバーの構成に関する情報は、47 ページの『クライアント/サーバー機能の構成』を参照してください。

注: このサブキーの形式と意味は、クライアントがサーバーと同じプライベート・ネットワーク上にあるか、あるいは、以下に説明するように、HTTPS を使用して公衆ネットワーク経由で接続されるのかによって異なります。

Server1

この値のレジストリー・データ・タイプは REG_SZ です。

サーバー名を以下のいずれかの形式で指定します。

- クライアントが HTTPS を使用してそのサーバーにアクセスする場合、HTTPS サポートを提供する WebSphere サーバーの名前と CS Linux サーバーの名前を、次の形式で指定します。

`webservername : servername1`

これは、WebSphere が HTTPS 接続にデフォルトのポート 443 を使用するように設定されていると想定しています。ネットワーク管理者が異なるポート番号を使うように WebSphere を構成している場合は、以下の形式でポート番号を指定します。

`webservername : portnumber : servername1`

WebSphere が HTTPS 接続をサポートするように構成する方法について詳しくは、「[IBM Communications Server for Data Center Deployment on Linux 入門](#)」を参照してください。

- クライアントがそのサーバーへのアクセスに HTTPS を使用しない場合、最初に接続を試行するサーバーの名前を指定します。 `webservername` および `portnumber` パラメーターは使用されず、指定してはなりません。
- クライアントがサーバーと同じプライベート・ネットワーク上にある場合、TCP/IP サブネット上 (または、クライアント・コンピューターに複数の LAN アダプター・カードがあれば、アクセス可能なすべてのサブネット上) のすべてのコンピューターに対して UDP ブロードキャスト・メッセージを使用して、CS Linux を実行するサーバーを検索するようにクライアントに指示できます。 これを実行するには、サーバー名の代わりに * (アスタリスク文字) を指定します。

このオプションは、クライアントが IPv4 をアドレッシングを使用する場合にのみ使用できます。UDP ブロードキャストは、IPv6 についてはサポートされません。

クライアントは、サーバーに接続するまでブロードキャストを 10 秒ごとに再試行し、`broadcast_attempt_count` パラメーターに指定された試行回数に達するまで続けます。サーバーに接続する前に `broadcast_attempt_count` に指定された限度に達した場合、クライアントは、送信されたメッセージを使用して、指名された 1 つ以上のサーバー (ファイルの以降の行に指定された) への接続を試行します。

注: UDP ブロードキャストを使用していない場合は、`Server2` から `Server9` のパラメーターを使用して、このクライアントからアクセスが必要なその他のサーバー名をすべて指定する必要があります。クライアントは、このファイルで指定したサーバー上のリソースは使用できますが、他のサーバー上のリソースは使用できません。

Server2-Server9

この値のレジストリー・データ・タイプは REG_SZ です。

このクライアントが使用するリソースを含む、追加の CS Linux サーバーの名前を指定します。 `Server1` の場合と同じ形式を使用します。

クライアントが UDP ブロードキャストを使用してサーバーへの接続を試行 (または、`Server1` に指定されたサーバーへの接続を試行) して、応答を受け取らなかった場合、クライアントは指定メッセージを使用して `Server2` に指定されたサーバーへの接続を試行します。これも失敗した場合、クライアントは `Server3` に指定されたサーバーへの接続を試行し、以降も同様です。これらのサーバー名はオプションですが、ブロードキャスト方式でのサーバーの検索に失敗した場合や、`Server1` に指定されたサーバーが使用できない場合のバックアップ手段になります。

リストされたサーバーすべての試行が成功しなかった場合、クライアントは `server_lost_timeout` パラメーターに指定された秒数だけ待ってから、サーバーへの接続の試行プロセスを再開します (UDP ブロードキャストを使用するか、リストの先頭にあるサーバーを使用して)。

パラメーター `Server2` から `Server9` を * に設定して、UDP ブロードキャストの使用を指示することはできません。* 値はファイル内でどのサーバー名よりも前に指定する必要があるため、この指示に使用できるパラメーターは `Server1` のみです。

ロギング

ロギングサブキーは、クライアントのロギング・オプションを指定します。これらのオプションを使用して、ドメイン全体に指定されたロギング・オプションをオーバーライドするクライアント・ロギング設定を指定することができます。ドメイン・ロギング・オプションの指定について詳しくは、[49 ページの『ロギングの構成』](#)を参照してください。

中央ロギングが有効になっている場合は、すべてのログ・メッセージがサーバー上の中央ファイルに書き込まれます。この場合、ここで指定した例外ログ・ロギングが使用可能パラメーターおよび `audit_logging_enabled` パラメーターのみが使用され、残りのパラメーターは無視されます。

ロギング・オプションは以下のように指定されます。

例外ログ・ロギングが使用可能

この値のレジストリー・データ・タイプは REG_SZ です。

このパラメーターは、以下のいずれかの値に設定します。

そうだ

例外メッセージを記録する。

違う。

例外メッセージを記録しません。

このパラメーターはオプションです。指定されていない場合、クライアントはグローバル・ドメイン設定を使用して、例外メッセージが記録されるかどうかを判別します。(初期のデフォルトは、例外メッセージが記録されることです。)

audit_logging_enabled

この値のレジストリー・データ・タイプは レグスズです。

このパラメーターは、以下のいずれかの値に設定します。

そうだ

監査メッセージを記録する。

違う。

監査メッセージを記録しません。

このパラメーターはオプションです。指定されていない場合、クライアントはグローバル・ドメイン設定を使用して、監査メッセージが記録されるかどうかを判別します。(初期のデフォルトは、監査メッセージが記録されることです。)

ログ・ディレクトリー

この値のレジストリー・データ・タイプは レグスズです。

このクライアント上でログ・ファイルが保管されるディレクトリーの絶対パス。すべてのログ・ファイルおよびバックアップ・ログ・ファイル(以下のパラメーターで指定)は、このディレクトリーに保管されます。*IBM Communications Server for Data Center* デプロイメント *on Linux* 診断ガイドで説明されているログ・フィルター機能を使用している場合は、ファイル `logfilter.txt` (この機能を制御する) もこのディレクトリーに保管されます。

このパラメーターはオプションです。これが指定されていない場合、ファイルは Windows のインストール・ディレクトリーに保管されます。

error_file

この値のレジストリー・データ・タイプは レグスズです。

エラー・メッセージが書き込まれるファイルの名前。このパラメーターはオプションです。これが指定されていない場合、デフォルトは `ドルー.` です。

エラー・メッセージと監査メッセージを 1つのファイルに記録するには、このパラメーターと `audit_file` パラメーターの両方に同じファイル名を指定します。

バックアップ・エラー・ファイル

この値のレジストリー・データ・タイプは レグスズです。

バックアップ・エラー・ログ・ファイルの名前。エラー・ログ・ファイルがエラー・ファイル・ラップ・サイズで指定されたサイズに達すると、CS Linux はその内容をバックアップ・ファイルにコピーし(既存のファイルを上書きします)、エラー・ログ・ファイルをクリアします。

このパラメーターはオプションです。これが指定されていない場合、デフォルトは `バーク.` です。

エラー・メッセージと監査メッセージを 1つのファイルに記録するには、このパラメーターと `audit_file` パラメーターの両方に同じファイル名を指定します。

エラー・ファイル・ラップ・サイズ

この値のレジストリー・データ・タイプは `REG_DWORD` です。

`error_file` によって指定されるログ・ファイルの最大サイズ。ファイルに書き込まれたメッセージによってファイル・サイズがこの制限を超えると、CS Linux はログ・ファイルの現在の内容をバックアップ・ログ・ファイルにコピーしてから、ログ・ファイルをクリアします。これは、エラー・ログ・ファイルによって取得されるディスク・スペースの最大量が、エラー・ファイル・ラップ・サイズパラメーターの値の約 2 倍であることを意味します。

このパラメーターはオプションです。これが指定されていない場合、デフォルトは 1000000 (バイト) です。エラー・メッセージと監査メッセージが同じファイルに記録されている場合、このパラメーターは 監査ファイル・ラップ・サイズ パラメーターと同じ値に設定する必要があります。

audit_file

この値のレジストリー・データ・タイプは レグスズです。

監査メッセージが書き込まれるファイルの名前。このパラメーターはオプションです。これが指定されていない場合、デフォルトは `スール・ワード` です。

エラー・メッセージと監査メッセージを 1 つのファイルに記録するには、このパラメーターと `error_file` パラメーターの両方に同じファイル名を指定します。

audit_file

この値のレジストリー・データ・タイプは レグスズです。

バックアップ監査ログ・ファイルの名前。監査ログ・ファイルが 監査ファイル・ラップ・サイズ で指定されたサイズに達すると、CS Linux はその内容をバックアップ・ファイルにコピーし (既存のファイルを上書きします)、監査ログ・ファイルをクリアします。

このパラメーターはオプションです。これが指定されていない場合、デフォルトは `bak.aud` です。

エラー・メッセージと監査メッセージを 1 つのファイルに記録するには、このパラメーターと バックアップ・エラー・ファイル パラメーターの両方に同じファイル名を指定します。

監査ファイル・ラップ・サイズ

この値のレジストリー・データ・タイプは REG_DWORD です。

`audit_file` によって指定されるログ・ファイルの最大サイズ。ファイルに書き込まれたメッセージによってファイル・サイズがこの制限を超えると、CS Linux はログ・ファイルの現在の内容をバックアップ・ログ・ファイルにコピーし、ログ・ファイルをクリアします。これは、監査ログ・ファイルによって取得されるディスク・スペースの最大量が、監査ファイル・ラップ・サイズ パラメーターの値の約 2 倍であることを意味します。

このパラメーターはオプションです。これが指定されていない場合、デフォルトは 1000000 (バイト) です。エラー・メッセージと監査メッセージが同じファイルに記録されている場合、このパラメーターはエラー・ファイル・ラップ・サイズ パラメーターと同じ値に設定する必要があります。

正常性エラー

この値のレジストリー・データ・タイプは レグスズです。

エラー・ログ・ファイルで簡潔ロギングを使用するか、詳細ロギングを使用するかを指定します。この設定は例外ログと問題ログの両方に適用されます。以下のいずれかの値を指定できます。

そうだ

簡潔なロギングを使用します。ログ・ファイル内の各メッセージには、メッセージ・ヘッダー情報 (メッセージ番号やログ・タイプなど) の要約、およびメッセージ・テキスト・ストリングとパラメーターが含まれています。ログの原因、および必要なアクションの詳細を取得するには、Linux を実行しているコンピューター上の `ヘビユーティリティー` を使用できます。

違う。

詳細ロギングを使用します。ログ・ファイル内の各メッセージには、メッセージ・ヘッダー情報の完全なリスト、メッセージ・テキスト・ストリングとパラメーター、およびログの原因に関する追加情報と、必要なアクションが含まれます。

このパラメーターはオプションです。これが指定されていない場合、デフォルトは、コントローラー・サーバーに発行された直前の `set_global_log_type` コマンド (または Motif 管理プログラムを使用して設定) から取得されます。 `set_global_log_type` コマンドが発行される前の初期デフォルトは、簡潔ロギングを使用することです。

セントラル・ロギングを使用している場合、すべてのコンピューターからのメッセージの簡潔なロギングまたは詳細ロギングの選択は、中央ロガーとして機能するサーバー上でこのパラメーターの設定によって決定されます。この設定は、 `set_global_log_type` コマンド、またはデフォルトをオーバーライドするために、そのサーバーに発行された `set_log_type` コマンドからのコマンド。のいずれかにすることができます。

正常に監査されます

この値のレジストリー・データ・タイプは REG_SZ です。

監査ログ・ファイルで簡潔なロギングを使用するか、詳細ロギングを使用するかを指定します。許可されている値とその意味は、正常性エラー パラメーターの場合と同じです。

API_tracing

API_tracing サブキーは、クライアント上で実行されるアプリケーションの API トレース・オプションを指定します。トレースの詳細については、「*IBM Communications Server for Data Center Deployment on Linux 診断ガイド*」を参照してください。トレース・オプションは、次のように指定します。

file1

この値のレジストリー・データ・タイプは REG_SZ です。

トレース・ファイルの絶対パス名、またはトレースの出力先ファイルが 2 つある場合の最初のトレース・ファイル (*file2* パラメーターの説明を参照)。

API トレースを使用可能に設定した場合は、このパラメーターが必要です。

file2

この値のレジストリー・データ・タイプは REG_SZ です。

2 番目のトレース・ファイルの絶対パス名。このパラメーターはオプションです。トレースの出力先を 2 つのファイルでなく 1 つのファイルにする場合は、この行を指定しないでください。

file1 と *file2* の両方を指定すると、トレースの出力先は 2 つのファイルになります。最初のファイルが *flip_size* パラメーターに指定されたサイズに達すると、2 番目のファイルがクリアされ、トレースは 2 番目のファイルを出力先として続きます。その後、このファイルが *flip_size* に指定されたサイズに達すると、最初のファイルがクリアされ、トレースは最初のファイルを出力先として続きます。これにより、過大なディスク・スペースを使用せずにトレースを長期間継続できるようになります。必要な最大スペースは、*flip_size* パラメーターの値の約 2 倍です。

flip_size

この値のレジストリー・データ・タイプは REG_DWORD です。

トレース・ファイルの最大サイズ。2 つのファイル名を指定した場合、現在のファイルがこのサイズに達すると、トレースは 2 つのファイルの間で切り替わります。ファイル名を 1 つだけ指定した場合、このパラメーターは無視され、ファイル・サイズは制限されません。

このパラメーターはオプションです。指定しない場合、デフォルトは 1000000 (バイト) です。

truncation_length

この値のレジストリー・データ・タイプは REG_DWORD です。

それぞれのメッセージについて、トレース・ファイルに書き込まれる情報の最大長 (バイト単位)。メッセージがこれより長い場合、CS Linux はメッセージの先頭のみをトレース・ファイルに書き込み、*truncation_length* を超えるデータは破棄します。これにより、長いメッセージでファイルがいっぱいなることを防ぎながら、それぞれのメッセージの最も重要な情報を記録できます。

このパラメーターはオプションです。指定しない場合、CS Linux はメッセージの切り捨てを行いません (各メッセージのデータがすべてファイルに書き込まれます)。

all_api

この値のレジストリー・データ・タイプは REG_SZ です。

すべての API に関するメッセージをトレースするには、このパラメーターを YES に設定します。この場合、CS Linux は *appc* から *nof* までのパラメーターを無視します。

すべての API のトレースを使用不可に設定するには、*all_api* を設定し、*appc* から *nof* のパラメーターすべてを NO に設定します。

特定の API に関するメッセージのみをトレースするには、*all_api* を NO に設定し、*appc* から *nof* のパラメーターを使用してトレース対象の API を指示します。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。

appc

この値のレジストリー・データ・タイプは REG_SZ です。

APPC API メッセージをトレースするには、このパラメーターを YES に設定します。そうでなければ、NO に設定します。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。all_api パラメーターが YES に設定されている場合、このパラメーターは無視され、APPC メッセージがトレースされます。

cpic

この値のレジストリー・データ・タイプは REG_SZ です。

CPI-C API メッセージをトレースするには、このパラメーターを YES に設定します。そうでなければ、NO に設定します。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。all_api パラメーターが YES に設定されている場合、このパラメーターは無視され、CPI-C メッセージがトレースされます。

csv

この値のレジストリー・データ・タイプは REG_SZ です。

CSV API メッセージをトレースするには、このパラメーターを YES に設定します。そうでなければ、NO に設定します。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。all_api パラメーターが YES に設定されている場合、このパラメーターは無視され、CSV メッセージがトレースされます。

rui

この値のレジストリー・データ・タイプは REG_SZ です。

LUA RUI メッセージをトレースするには、このパラメーターを YES に設定します。そうでなければ、NO に設定します。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。all_api パラメーターが YES に設定されている場合、このパラメーターは無視され、LUA RUI メッセージがトレースされます。

nof

この値のレジストリー・データ・タイプは REG_SZ です。

NOF API メッセージをトレースするには、このパラメーターを YES に設定します。そうでなければ、NO に設定します。NOF メッセージは、Windows クライアント上のアプリケーションによって直接使用されることはありませんが、構成情報を取得するために CS Linux コンポーネントによって内部的に使用されます。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。all_api パラメーターが YES に設定されている場合、このパラメーターは無視され、NOF メッセージがトレースされます。

CS_tracing

CS_tracing サブキーは、クライアント/サーバーのトレース (クライアントと CS Linux サーバーとの間のメッセージのトレース) のオプションを指定します。トレースの詳細については、「*IBM Communications Server for Data Center Deployment on Linux 診断ガイド*」を参照してください。トレース・オプションは、次のように指定します。

file1

この値のレジストリー・データ・タイプは REG_SZ です。

トレース・ファイルの絶対パス名、またはトレースの出力先ファイルが 2 つある場合の最初のトレース・ファイル (file2 パラメーターの説明を参照)。

このパラメーターは、クライアント/サーバー・トレースを使用可能に設定する必要がある場合に必要です。さらに、trace_flags パラメーターも設定する必要があります。

file2

この値のレジストリー・データ・タイプは REG_SZ です。

2 番目のトレース・ファイルの絶対パス名。このパラメーターはオプションです。トレースの出力先を 2 つのファイルでなく 1 つのファイルにする場合は、この行を指定しないでください。

file1 と *file2* の両方を指定すると、トレースの出力先は 2 つのファイルになります。最初のファイルが *flip_size* パラメーターに指定されたサイズに達すると、2 番目のファイルがクリアされ、トレースは 2 番目のファイルを出力先として継続します。その後、このファイルが *flip_size* に指定されたサイズに達すると、最初のファイルがクリアされ、トレースは最初のファイルを出力先として継続します。これにより、過大なディスク・スペースを使用せずにトレースを長期間継続できるようになります。必要な最大スペースは、*flip_size* パラメーターの値の約 2 倍です。

flip_size

この値のレジストリー・データ・タイプは REG_DWORD です。

トレース・ファイルの最大サイズ。2 つのファイル名を指定した場合、現在のファイルがこのサイズに達すると、トレースは 2 つのファイルの間で切り替わります。ファイル名を 1 つだけ指定した場合、このパラメーターは無視され、ファイル・サイズは制限されません。

このパラメーターはオプションです。指定しない場合、デフォルトは 1000000 (バイト) です。

admin_msg

この値のレジストリー・データ・タイプは REG_SZ です。

クライアント/サーバー・トポロジーに関連する内部メッセージをトレースするには、このパラメーターを YES に設定します。そうでなければ、NO に設定します。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。

datagram

この値のレジストリー・データ・タイプは REG_SZ です。

データグラム・メッセージをトレースするには、このパラメーターを YES に設定します。そうでなければ、NO に設定します。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。

data

この値のレジストリー・データ・タイプは REG_SZ です。

データ・メッセージをトレースするには、このパラメーターを YES に設定します。そうでなければ、NO に設定します。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。

send

この値のレジストリー・データ・タイプは REG_SZ です。

クライアントからサーバーに送信するデータ・メッセージをすべてトレースするには、このパラメーターを YES に設定します。そうでなければ、NO に設定します。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。

receive

この値のレジストリー・データ・タイプは REG_SZ です。

クライアントがサーバーから受信したデータ・メッセージをすべてトレースするには、このパラメーターを YES に設定します。そうでなければ、NO に設定します。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。

Internal_tracing

Internal_tracing サブキーは、クライアントの内部操作をトレースするためのオプションを指定します。トレースの詳細については、「*IBM Communications Server for Data Center Deployment on Linux 診断ガイド*」を参照してください。トレース・オプションは、次のように指定します。

file1

この値のレジストリー・データ・タイプは REG_SZ です。

トレース・ファイルの絶対パス名、またはトレースの出力先ファイルが 2 つある場合の最初のトレース・ファイル (*file2* パラメーターの説明を参照)。

このパラメーターは、内部トレースを使用可能に設定する場合に必要です。さらに、*trace_level* パラメーターを設定する必要があります。

file2

この値のレジストリー・データ・タイプは REG_SZ です。

2 番目のトレース・ファイルの絶対パス名。このパラメーターはオプションです。トレースの出力先を 2 つのファイルでなく 1 つのファイルにする場合は、この行を指定しないでください。

file1 と *file2* の両方を指定すると、トレースの出力先は 2 つのファイルになります。最初のファイルが *flip_size* パラメーターに指定されたサイズに達すると、2 番目のファイルがクリアされ、トレースは 2 番目のファイルを出力先として継続します。その後、このファイルが *flip_size* に指定されたサイズに達すると、最初のファイルがクリアされ、トレースは最初のファイルを出力先として継続します。これにより、過大なディスク・スペースを使用せずにトレースを長期間継続できるようになります。必要な最大スペースは、*flip_size* パラメーターの値の約 2 倍です。

flip_size

この値のレジストリー・データ・タイプは REG_DWORD です。

トレース・ファイルの最大サイズ。2 つのファイル名を指定した場合、現在のファイルがこのサイズに達すると、トレースは 2 つのファイルの間で切り替わります。ファイル名を 1 つだけ指定した場合、このパラメーターは無視され、ファイル・サイズは制限されません。

このパラメーターはオプションです。指定しない場合、デフォルトは 1000000 (バイト) です。

trace_level

この値のレジストリー・データ・タイプは REG_DWORD です。

トレース対象に含める詳細のレベル。有効な値の範囲は、0 (すべてのトレース) から 20 (トレースなし) です。

このパラメーターはオプションです。指定しない場合、デフォルトは 20 (トレースなし) です。

trace_flushing

この値のレジストリー・データ・タイプは REG_SZ です。

このパラメーターを YES に設定した場合、それぞれのトレース記述はディスクに即時に書き出されます。このため操作の速度が大幅に低下しますが、クラッシュが発生した場合にトレース・データが確保されます。

このパラメーターはオプションです。指定しない場合、デフォルトは NO です。

Appl_Name

Appl_Name サブキーは、CPI-C アプリケーションのオプションを指定します。

注：これらのオプションは、環境変数またはレジストリーで指定できます。CS Linux はまず環境変数を検査し、指定されている場合はその情報を使用します。環境変数が指定されていない場合は、レジストリー項目のみを使用します。Windows Terminal Server を使用し、同じアプリケーションの複数のコピーを異なるローカル LU を使用して実行する必要がある場合、環境変数を使用する必要がある場合もあります。

レジストリーの中でこれらのオプションを 1 つ以上のアプリケーションに対して設定するには、それぞれのアプリケーションごとにこの形式のセクションを組み込み、*Appl_Name* 変数をアプリケーション・プログラムの実行可能名 (.exe ファイル名拡張子を付けない) に置き換えます。

CPI-C の詳細については、「*IBM Communications Server for Data Center Deployment on AIX or Linux CPI-C プログラマーズ・ガイド*」を参照してください。

オプションは次のように指定します。

APPCLLU

このオプションは、レジストリー内ではなく、APPCLLU 環境変数を使用して指定できます。

この値のレジストリー・データ・タイプは REG_SZ です。

このアプリケーションが使用するローカル LU の名前。

このパラメーターはオプションです。指定しない場合、アプリケーションはデフォルト LU (ローカル・ノードの制御点に関連した LU) の使用を試みます。

APPCTPN

このオプションは、レジストリー内ではなく、APPCTPN 環境変数を使用して指定できます。

この値のレジストリー・データ・タイプは REG_SZ です。

アプリケーションの TP 名。この名前は、ログ・ファイルとトレース・ファイル内でアプリケーションの識別に使用されます。呼び出されたアプリケーション (Accept_Conversation を実行するアプリケーション) の場合は、着信割り振り要求の TP 名を正しいアプリケーションと突き合わせるためにも使用されます。呼び出されたアプリケーションは、Specify_Local_TP_Name 呼び出しを使用して、着信割り振り要求と突き合わせる名前を追加指定することもできます。

このパラメーターはオプションです。指定しない場合、デフォルトは CPIC_DEFAULT_TPNAME です。

CSV_data

CSV_data サブキーは、CSV インターフェースを使用するアプリケーションのオプションを指定します。このサブキーは、CONVERT verb を使用して、ユーザー定義の変換テーブル (テーブル G) による文字変換を実行するアプリケーションのみに適用されます。CONVERT verb の詳細については、「*IBM Communications Server for Data Center Deployment on AIX or Linux 共通サービス Verb プログラマーズ・ガイド*」を参照してください。

クライアント上のアプリケーションがこの機能を使用しない場合は、このセクションを組み込む必要はありません。

このセクションのオプションは、次の 1 つだけです。

CSVTLG

この値のレジストリー・データ・タイプは REG_SZ です。

ユーザー定義のテーブル G 変換テーブルを含むファイルの絶対パス名。CSV アプリケーションがテーブル G 文字変換を実行する必要がある場合は、このパラメーターが必要です (デフォルトはありません)。そうでなければ、このパラメーターはオプションです。

Tracking SNA LU resources used by clients on a domain of servers

An APPC application may require two or more conversations with its partner application in order to complete a task, especially if the partner application cannot complete its processing immediately. For example, application A sends data to application B for processing that will take some time to complete, and terminates without waiting for the results. At a later time, when it has completed the processing, application B starts a new conversation with application A to return the results.

If an application that operates in this way runs on two or more clients, CS Linux needs to ensure that the new conversation is routed back to the original client that made the request. In particular, if the client application uses LUs in an LU pool, it needs to maintain an association with the LU that it used for the original request, so that the new conversation can be routed from this LU to the correct client application.

In order to use this function, you need to set the `CLI_OVERWRITE_SYS_NAME` option in the `ptf_flags` parameter of the `define_node` command for the node that owns the LU pool. Then use a standard text editor to create a file named `tpnamehost`, in the directory where the client software is installed. The file can be empty: CS Linux only checks whether the file exists, and does not take any notice of the contents.

If the `tpnamehost` file exists on the client:

- CS Linux modifies the `sys_name` of the LU used by the application to match this IP hostname. This ensures that, when the partner application requests a new conversation using this LU, CS Linux will connect it with the application running on the specified client (the application that made the original request using this LU).
- CS Linux also takes the TP name specified by the application, and appends a % character followed by the IP hostname of the client computer. This modified TP name string is then used in logging and tracing for the application running on this client, so that you can distinguish between two or more copies of the application on different clients.

Note: When you create the `tpnamehost` file on the client, ensure that there is not already a file named `tpnameipv4` in the same location. (The use of this file is described in “異なるクライアント上で実行する TP を区別するためのローカル TP 名のオーバーライド” on page 132.) If both files exist, the `tpnameipv4` file takes precedence: CS Linux appends the IP address and not the IP hostname, and does not create an association between the client application and its LU.

クライアント・アプリケーションでハードコーディングされた LU 別名をサーバーのドメイン内の LU 別名にマッピングする

2 つ以上の Remote API Client on Windows で同じ APPC、CPI-C、または LUA アプリケーションを実行している場合、それが指定するローカル LU 別名、LUA LU 名、または TP 名のオーバーライドが必要になることがあります。これにより、2 つ以上のアプリケーションのコピーを、アプリケーションを変更せずに異なるローカル LU を使用するように割り当てたり、同じアプリケーションの 2 つ以上のコピーを区別したりできます。(AIX または Linux 上の Remote API Client での同等の機能については、136 ページの『クライアント・アプリケーション上のハード・コード LU 別名をサーバーのドメイン内の LU 別名にマップする』を参照してください。)

Windows 上の Remote API Client の `lualiasmap.txt` ファイルは、このクライアント上で実行しているアプリケーションによって指定される LU と、アプリケーションが使用する必要がある実際の LU の間のマッピングを定義します。これにより、同じアプリケーションを複数のクライアントで実行し、それぞれが異なる LU を使用するよう強制できます。

- APPC アプリケーションの場合、これは `TP_STARTED verb` でアプリケーションによって指定されたローカル LU 別名と、このクライアントのアプリケーションが使用する必要があるローカル LU とをマップします。
- CPI-C アプリケーションの場合、これはローカル LU 別名 (`Set_Local_LU_Name` を使用してアプリケーションにより指定されたか、または CPI-C 構成から取られたもの) と、このクライアントのアプリケーションが使用する必要があるローカル LU とをマップします。
- LUA アプリケーションの場合、`RUI_INIT` または `RUI_INIT_PRIMARY verb` でアプリケーションによって指定される `lua_luname` パラメーターと、このクライアント上のアプリケーションが使用する必要があるタイプ 0 から 3 の LU の間のマッピングを行います。アプリケーションが拡張形式の `RUI_INIT` (LU をその LU 名ではなく、その PU 名と LU 番号で識別する) を使用している場合には、このマッピングには効果がありません。

この機能を使用する必要がある場合、標準のテキスト・エディターを使用して、クライアント・ソフトウェアのインストール先ディレクトリーに `lualiasmap.txt` という名前の ASCII テキスト・ファイルを作成します。

ファイルの各行には、2 つの 8 文字ストリングがシングル・スペース区切りで含まれています。最初のストリングは列 1 で開始する必要があり、アプリケーションで指定されたローカル LU 別名または LUA LU 名になります (8 文字より短い場合はスペースが埋め込まれます)。2 番目のストリングは列 10 で始まる必要があります (最初の 8 文字ストリングに 1 個のスペースを付けた後です)。また、これはこのクライアント上のアプリケーションが使用する必要がある実際のローカル LU あるいはタイプ 0 から 3 の LU です。こ

のファイルの最大行数は 256 です(旧バージョンの Windows クライアントの最大行数は 64 でした)。これにより、255 個の LU を持つ PU 全体をマップできます。

以下に示すのは、マッピング・ファイルの例です。

```
MyAppLU1 APPCLu5
MyAppLU2 APPCLu6
LUALU3   NewLUA11
LUALU4   NewLUA12
```

この例で、それぞれの意味は次のとおりです。

- ローカル LU 別名 MyAppLU1 または MyAppLU2 を指定するこのクライアント上で稼働する APPC または CPI-C アプリケーションは、lu_alias が APPCLu5 または APPCLu6 の実ローカル LU を使用するようにマップされます。
- このクライアント上で実行している、LU 名 LUALU3 または LUALU4 を指定する LUA アプリケーションは、NewLUA11 または NewLUA12 という名前のタイプ 0 から 3 の実際の LU を使用するようにマップされます。これら 2 つの項目間の追加のスペースに注意してください。これはアプリケーションによって指定された LU 名が 8 文字より短いためです。

さらに、luaaliasmap.txt ファイルでは、以下の文字を使用できます。

- + 文字 - 一部のレガシー LUA アプリケーション (LANDP など) は、RUI_INIT() 応答で返された session_id と LU を無視します。その場合、マッピングは有効になりません。luaaliasmap.txt ファイルの行 1 の位置 1 に + 文字が含まれている場合、LU マッピングはアプリケーションによって発行されるすべての RUI_*() verb (RUI_INIT() だけではない) に適用されます。
- * 文字 - luaaliasmap.txt ファイルでは、ワイルドカード接尾部を指定できます。最初のストリング内で * を使用することで、1 つの LU (またはプール) 名に複数の LU 別名をマップすることができます。例えば、行

```
pref*   LUALU
```

は、pref で始まるすべての別名をその 1 つの LU にマップします。両方の位置で * を使用することで、複数の別名をマップすることもできます。例えば、行

```
alias*  LUNAM*
```

は、alias01 を LUALU01 に、あるいは alias7 を LUALU7 にといったようにマップします。名前が 8 文字以内である限り、接頭部の長さは任意です。* を 2 つ目の位置でのみ使用することはサポートされていません。

- - 文字 - アプリケーションがサーバー (任意のサーバー) で認識されない LU 別名を使用してセッションを開こうとすると、RUI_INIT() verb がクライアントからそのサーバーに送信され、クライアントにエラーが返されます。場合によって、アプリケーションは、認識されないいくつかの LU 別名は使用せずに再試行します。これにより、不要なトラフィックと CPU の使用が発生します。- を列 10 (2 つ目のストリングではなく) に配置すると、これを防ぐことができます。これは、* ワイルドカード構文と一緒に使用できます。例えば、ファイル

```
alias1  alias1
alias2  alias2
*       -
```

は、2 つの別名 alias1 および alias2 のみをサーバーに送信することを許可します (この例のように、マッピングをノーオペレーションにすることができます)。他の別名を指定する RUI_INIT() または TP_STARTED() は、サーバーにメッセージを送信することなく、クライアント・ライブラリー・コードから拒否されます。

異なるクライアント上で実行する TP を区別するためのローカル TP 名のオーバーライド

Windows 上の Remote API Client の `tpnameipv4` ファイルは、クライアント上で実行している APPC または CPI-C アプリケーションによって指定された TP 名を変更するよう CS Linux に指示します。これは、各クライアントが固有の TP 名を使用するためです (その IPv4 アドレスで区別される)。これにより、2 つ以上のクライアント上で同じアプリケーションを実行し、それらが区別できるように異なる TP 名を使用することを強制できます。(アプリケーションによって指定された TP 名は、ローカル ID としてのみ使用され、CS Linux 構成内でどのパラメーターとも突き合わせる必要がないため、アプリケーションはその元の TP 名を使用している場合と同じように稼働します。)

- APPC アプリケーションの場合、TP 名は `TP_STARTED verb` でアプリケーションによって指定されます。
- CPI-C アプリケーションの場合、TP 名は、`Specify_Local_TP_Name` または `Set_CPIC_Side_Information` を使用してアプリケーションによって指定されるか、あるいは CPI-C 構成から取られます。

この機能を使用する必要がある場合、標準のテキスト・エディターを使用して、クライアント・ソフトウェアのインストール先ディレクトリーに `tpnameipv4` という名前のファイルを作成します。このファイルは空であってもかまいません。CS Linux はファイルが存在しているかどうかを確認するだけで、その内容については注意を払いません。

`tpnameipv4` ファイルがクライアント上にある場合、CS Linux はアプリケーションによって指定された TP 名を選んで下線文字を追加し、その後にクライアント・コンピューターの IP アドレスを続けます。この変更された TP 名ストリングは、このクライアント上で実行しているアプリケーションのログギングおよびトレースに使用されます。これにより、異なるクライアント上のアプリケーションの複数コピーを区別できます。



AIX または Linux 上の Remote API Client の管理

LINUX

Remote API Client は、AIX、Linux、Linux for Power、または Linux for IBM Z 上で実行できます。

AIX または Linux 上の Remote API Client のクライアント情報は、`sna_clnt.net` ファイルに保管されます。このファイルは、クライアントに SNA ソフトウェアをインストールするときに作成されます。クライアント・ソフトウェアを使用可能に設定するには、事前にそのファイルが存在する必要があります。

注: クライアントが HTTPS を使用してサーバーにアクセスする場合、クライアントを使用する前に、`sna_clnt.net` ファイルを変更して、これらのサーバーの名前と、サーバーへの HTTPS アクセスを提供する WebSphere サーバーの名前を指定する必要があります。詳しくは、[133 ページの『クライアント・ネットワーク・データ・ファイル \(sna_clnt.net\)』](#)を参照してください。

AIX または Linux でのリモート API クライアントの使用可能化および使用不可

AIX または Linux 上で Remote API Client ソフトウェアを使用可能にするには、コマンド・プロンプトで以下のコマンドを入力します。

```
スネアスタート [-t]
```

When you install the client, the installation utility automatically updates the startup file `/etc/rc.sna` (AIX) or `/etc/rc.d/init.d/snastart` (Linux) to include the スネアスタート command. これにより、クライアントがシステムの始動時に自動的に開始されます。自動的に開始したくない場合は、この行を削除またはコメント化してから、このセクションの指示に従って手動でソフトウェアを使用可能にすることができます。

唯一のオプションは以下のとおりです。

-t

クライアント / サーバーのトレースをアクティブにします。これにより、クライアントがサーバーに接続しようとしたときに発生する問題を診断することができます。このオプションを使用しない場合、

クライアント / サーバー・トレースはすべてのインターフェースで非アクティブになります。その後、コマンド行管理プログラム `snaadmin` を使用して、このトレースを必要に応じて活動化することができます。

このオプションは、Motif 管理プログラムでトレースをすべて設定 オン フィールドを選択するのと同様です。ただし、DLC トレースを使用可能にしない点は異なります。

トレースを実行すると、CS Linux コンポーネントのパフォーマンスが低下します。ソフトウェアが使用可能になると、コマンド行管理プログラム `snaadmin` を使用して、不要になったトレースを停止することができます。トレースについて詳しくは、*IBM Communications Server for Data Center デプロイメント on Linux 診断ガイド* を参照してください。

Remote API Client を停止するには、コマンド・プロンプトで以下のコマンドを入力します。

スナ・ストップ

クライアント・ネットワーク・データ・ファイル (sna_clnt.net)

`sna_clnt.net` ファイルは、AIX または Linux 上のリモート API クライアント上で使用可能な CS Linux 機能を定義し、クライアントがアクセスできるサーバーを定義します。(Windows クライアント上の同等のファイルについては、[111 ページの『第 10 章 CS Linux クライアント / サーバー・システムの管理』](#)を参照してください。)

また、CS Linux がクライアント/サーバー通信に使用する IP ポート番号の設定に関する情報も含まれています。デフォルトのポート番号は、ほとんどの場合に適しています。この情報を参照する必要があるのは、CS Linux が、同じコンピューター上の別のプログラムと競合するポート番号があることを示すエラー・メッセージである場合のみです。

クライアント・コンピューターは、ドメイン構成ファイルまたは SNA ネットワーク・データ・ファイルのコピーを保持しません。これは、CS Linux LAN 上のサーバーにアクセスするために必要な情報のみを保持し、必要な構成情報を提供するためにサーバーに依存します。

必要な SNA ネットワーク情報は、ファイル `sna_clnt.net` に保持されています。このファイルは、AIX 上のディレクトリー `/etc/sna`、または Linux 上の `/etc/opt/ibm/sna` に保管されています。このファイルは、クライアントのインストール・プロセス中にセットアップされます。このファイルは、標準のテキスト・エディターを使用して後で必要に応じて変更できる ASCII テキスト・ファイルです。

注: このファイル内のパラメーターを変更した後、変更を有効にするには、クライアントを停止して再始動する必要があります。これを行う方法の詳細については、[132 ページの『AIX または Linux でのリモート API クライアントの使用可能化および使用不可』](#)を参照してください。

ファイルの内容は以下のとおりです。

```
domain = domain_name
maximum_process_count = nn
maximum_header_count = nn
maximum_element_count = nn
invoked_tps = YES | NO
lan_access_timeout = nn
poll_timer = nn
broadcast_attempt_count = nn
server_lost_timeout = nn
* | [ webservername : [ portnumber : ] ]servername1
[ webservername : [ portnumber : ] ]servername2
.
.
```

以下のリストでは、ファイルの各行のパラメーターについて説明します。

ドメイン

ドメイン名パラメーター値は、CS Linux LAN のドメイン・ネームを示します。この名前は、クライアントのインストール時に `ibmcs_domain` に設定されます。この行は必須です。

最大 `process_count`

このクライアント上で一度に実行できる APPC、CPI-C、LUA、および NOF アプリケーションの最大合計数を指定します。

このパラメーターはオプションです。デフォルト値は 240 ですが、通常はこれで十分です。IPC 制御ブロックの割り振りの失敗を報告するエラー・メッセージが表示される場合は、このパラメーターを指定することによって、最大プロセス数を増やす必要がある場合があります。指定できる最大値は 4096 です。

最大ヘッド・カウント, 最大エレメント数の最大値

これらの 2 つのパラメーターはオプションです。デフォルト値は 1250 および 1800 です。これは、通常は十分です。通常、これらのパラメーターに値を提供する必要はありません。ただし、サポート担当者に指示された場合

呼び出し側 tps

このクライアントが呼び出された TP (RECEIVE_ALLOCATE verb を発行する APPC TP、または Accept_Conversation verb または Accept_Incoming verb を発行する CPI-C アプリケーション) を実行するために使用される場合は、呼び出しの呼び出し = YES を指定します。この場合、このクライアント上で TP を定義する必要がある場合もあります。詳しくは、75 ページの『TP の定義』を参照してください。

このクライアントが呼び出された TP の実行に使用されない場合は、呼び出しの呼び出し = NO を指定します

この行はオプションです。組み込まれていない場合、デフォルトは違う。です。

l_access_timeout

クライアント上のアプリケーションが CS Linux リソースを使用していないときに、クライアントからサーバーへの IP または HTTPS 接続がアクティブに保たれる時間を秒単位で指定します。詳しくは、114 ページの『LAN アクセス・タイムアウト』を参照してください。

最小タイムアウトは 60 秒です (これより小さい値は 60 秒に切り上げられます)。接続をより速く停止するには、クライアント上の CS Linux ソフトウェアを使用不可にしてください。

タイムアウトがないことを示すために、CS Linux ソフトウェアがクライアント上で実行されている限り、接続がアクティブに保持されるようにするには、このパラメーターを指定しないでください。

このパラメーターはオプションです。これが指定されていない場合、デフォルトはタイムアウトになりません。

poll_タイマー

このパラメーターは、クライアントがファイアウォールを経由してサーバーに接続する場合にのみ使用されます。クライアント上のアプリケーションが一定期間 CS Linux リソースを使用していない場合は、アクティビティーがないため、ファイアウォールが接続を終了する可能性があります。このパラメーターは、接続をアクティブに保つために、クライアントが一定の間隔でサーバーにポーリング・メッセージを送信するように強制します。サーバーは、このメッセージを他のユーザーに対して使用しません。

ポーリング・メッセージとポーリング・メッセージの間の時間を 1 から 1440 (1 分の 1 分) の範囲で指定します。通常、値 10 はほとんどのファイアウォールに適しています。

ポーリング・メッセージがないことを示すために、ファイアウォールがアクティビティーを検出しなかった場合に接続をドロップする可能性があるため、0 (ゼロ) を指定します。

このパラメーターはオプションです。これが指定されていない場合、クライアントはポーリング・メッセージを送信しません。

ブロードキャスト試行回数 (t_count)

クライアントがブロードキャスト・メソッドを使用してサーバー (最初のサーバー名ではなく * で指定) に接続する場合、このパラメーターは、サーバーに接続しようとする 1 回の試行で実行されるブロードキャストの最大数を指定します。最小値は 1 です。高い値が指定されている場合、クライアントはサーバーに接続するか、この数に達するまで、10 秒ごとに再試行します。サーバーに接続せずにカウントに達すると、クライアントは、指定されたサーバーへの接続を試行します。

このパラメーターはオプションです。これが指定されていない場合、デフォルトは 5 です。

サーバー・キャッシュ・タイムアウト

クライアントがサーバーとの接続を失い、再接続する必要がある場合、または、ブロードキャストまたは指定されたサーバーを使用してサーバーへの接続が失敗した場合、このパラメーターは、サーバーへ

の接続の試行を開始または再始動する前に、クライアントが待機する時間を秒単位で指定します。クライアントがサーバーとの接続を失った場合、CS Linux はフル・タイムアウト期間を待機しませんが、5 秒から指定されたタイムアウトまでのランダム期間の経過後に再試行します。これは、多数のクライアントが同時にサーバーに接続しようとしていることによるネットワーク・トラフィックの負荷を回避するためです。

このパラメーターはオプションです。これが指定されていない場合、デフォルトは 200 秒です。

サーバー名

1 つ以上の CS Linux サーバーの名前を指定します。このクライアントが使用するリソースを含みます。このリストには、コントローラー構成サーバーの名前と、クライアントと同じドメイン内のすべてのバックアップ・サーバーの名前が含まれている必要があります。コントローラーおよびバックアップ・サーバーの構成については、47 ページの『クライアント/サーバー機能の構成』を参照してください。また、このクライアントがアクセスする必要のある他のサーバーの名前も含まれている必要があります。クライアントは、指定されたサーバー上のリソースを使用できますが、他のサーバー上のリソースは使用できません。

サーバー名は、次のいずれかの形式で入力します。

- クライアントが HTTPS を使用してサーバーにアクセスする場合は、HTTPS サポートを提供する WebSphere サーバーの名前、および CS Linux サーバーの名前を以下の形式で指定します。

Web サーバー名 : サーバー名 1

これは、HTTPS 接続にデフォルト・ポート 443 を使用するように WebSphere がセットアップされていることを想定しています。ご使用のネットワーク管理者が別のポート番号を使用するように WebSphere を構成している場合は、ポート番号を以下の形式で指定します。

Web サーバー名 : ポート番号 : サーバー名 1

HTTPS 接続をサポートするように WebSphere を構成する方法については、Linux スタートアップ・ガイドでのデータ・センター・デプロイメントのための *IBM Communications Server* を参照してください。

- クライアントがサーバーにアクセスするために HTTPS を使用していない場合は、接続しようとする最初のサーバーの名前を指定します。Web サーバー名パラメーターおよびポート番号パラメーターは使用されず、指定することはできません。
- クライアントがサーバーと同じプライベート・ネットワーク上にある場合は、その TCP/IP サブネット上のすべてのコンピューター（または、クライアント・コンピューターに複数の LAN アダプター・カードが含まれている場合は、アクセス可能なすべてのサブネット上）に対して UDP ブロードキャスト・メッセージを使用して、CS Linux を実行するサーバーを検出する必要があることを指定できます。これを行うには、最初のサーバー名の代わりに * (アスタリスク文字) を指定します。

このオプションは、クライアントが IPv4 アドレッシングを使用する場合にのみ使用可能です。UDP ブロードキャストは IPv6 ではサポートされません。

クライアントは、サーバーに接続するまで、10 秒ごとに、ブロードキャスト試行回数 (*t_count*) パラメーターによって指定された試行回数まで、ブロードキャストを再試行します。サーバーが接続される前にブロードキャスト試行回数 (*t_count*) によって指定された制限に達すると、クライアントは、指定されたメッセージを 1 つ以上の指定されたサーバー（ファイルの以下の行で指定）を使用して試行します。

注: UDP ブロードキャストを使用していない場合は、このクライアントがアクセスする必要のあるすべてのサーバーの名前を指定する必要があります。クライアントは、このファイルに指定されているサーバー上のリソースを使用できますが、他のサーバー上のリソースを使用することはできません。

クライアントは、成功なしでリストされたすべてのサーバーを試行すると、上記のサーバー・キャッシュ・タイムアウトによって指定された時間を待機してから、サーバーへの接続を試行するプロセスを再開します (UDP ブロードキャストで、または最初のサーバーがリストされている場合)。

sna_clnt.net のほかに、追加のファイルサーバー . 現行が同じディレクトリーに保管されます (AIX 上の場合は /var/sna、Linux 上の場合は /var/opt/ibm/sna)。これは、クライアントが現在接続しているサーバー（ある場合）の名前が入っているテキスト・ファイルです。このファイルを確認して、ドメインへのクライアントの接続ポイントとして機能しているサーバーを判別することができます。

Tracking SNA LU resources used by clients on a domain of servers

An APPC application may require two or more conversations with its partner application in order to complete a task, especially if the partner application cannot complete its processing immediately. For example, application A sends data to application B for processing that will take some time to complete, and terminates without waiting for the results. At a later time, when it has completed the processing, application B starts a new conversation with application A to return the results.

If an application that operates in this way runs on two or more clients, CS Linux needs to ensure that the new conversation is routed back to the original client that made the request. In particular, if the client application uses LUs in an LU pool, it needs to maintain an association with the LU that it used for the original request, so that the new conversation can be routed from this LU to the correct client application.

In order to use this function, you need to set the `CLI_OVERWRITE_SYS_NAME` option in the `ptf_flags` parameter of the `define_node` command for the node that owns the LU pool. Then use a standard text editor to create a file named `tpnamehost`, in the directory `/etc/sna` on AIX or `/etc/opt/ibm/sna` on Linux. The file can be empty: CS Linux only checks whether the file exists, and does not take any notice of the contents.

If the `tpnamehost` file exists on the client:

- CS Linux modifies the `sys_name` of the LU used by the application to match this IP hostname. This ensures that, when the partner application requests a new conversation using this LU, CS Linux will connect it with the application running on the specified client (the application that made the original request using this LU).
- CS Linux also takes the TP name specified by the application, and appends a % (percentage sign) character followed by the IP hostname of the client computer. This modified TP name string is then used in logging and tracing for the application running on this client, so that you can distinguish between two or more copies of the application on different clients.

Note: When you create the `tpnamehost` file on the client, ensure that there is not already a file named `tpnameipv4` in the same location. (The use of this file is described in “異なるクライアントで実行されている TP を区別するためにローカル TP 名をオーバーライドする” on page 138.) If both files exist, the `tpnameipv4` file takes precedence: CS Linux appends the IP address and not the IP hostname, and does not create an association between the client application and its LU.

クライアント・アプリケーション上のハード・コード LU 別名をサーバーのドメイン内の LU 別名にマップする

AIX または Linux 上の 2 つ以上のリモート API クライアント上で同じ APPC、CPI-C、または LUA アプリケーションを実行している場合は、ローカル LU 別名、LUA LU 名、またはそれが指定する TP 名をオーバーライドする必要がある場合があります。これにより、アプリケーションを変更せずに異なるローカル LU を使用したり、同じアプリケーションの 2 つ以上のコピーを区別したりするために、アプリケーションの複数のコピーを割り当てることができます。(Windows クライアント上の同等の機能については、130 ページの『クライアント・アプリケーションでハードコーディングされた LU 別名をサーバーのドメイン内の LU 別名にマッピングする』を参照してください。)

AIX または Linux 上のリモート API クライアント上の `lualiasmap.txt` ファイルは、このクライアントで実行されているアプリケーションと、アプリケーションが使用する実際の LU によって指定された LU 間のマッピングを定義します。これにより、同じアプリケーションを複数のクライアントで実行し、それらのクライアントが異なる LU を強制的に使用することができます。

- APPC アプリケーションの場合は、`TP_STARTED` verb のアプリケーションによって指定されたローカル LU 別名と、このクライアント上のアプリケーションが使用するローカル LU との間でマップされます。
- CPI-C アプリケーションの場合は、ローカル LU 別名 (アプリケーションが `Set_Local_LU_Name` を使用して指定するか、CPI-C 構成から取得される) と、このクライアント上のアプリケーションが使用するローカル LU との間のマップを作成します。
- LUA アプリケーションの場合、このアプリケーションは、`RUI_INIT` または `RUI_INIT_PRIMARY` verb のアプリケーションによって指定された `l_a_luname` パラメーターと、このクライアント上のアプリケー

セッションが使用するタイプ 0 から 3 の LU をマップします。アプリケーションが RUI_INIT の拡張形式を使用している場合は、このマッピングは効果がありません。この形式では、LU 名ではなく、PU 名と LU 番号によって LU を識別します。

この機能を使用する必要がある場合は、標準のテキスト・エディターを使用して `lualiasmap.txt`。AIX 上のディレクトリ `/etc/sna`、または Linux 上の `/etc/opt/ibm/sna`。という名前の ASCII テキスト・ファイルを作成し

ファイルの各行には、1 つのスペースで区切られた 2 つの 8 文字ストリングが含まれます。最初のストリングは 1 桁目から開始する必要があり、アプリケーションによって指定されるローカル LU の別名または LUA LU 名です (8 文字より短い場合はスペースが埋め込まれます)。2 番目のストリングは、10 桁目から開始する必要があります (最初の 8 文字ストリングの後にスペースがある)。2 番目のストリングは、このクライアント上のアプリケーションが使用する必要がある実際のローカル LU またはタイプ 0 から 3 の LU です。このファイルには、最大 256 行 (UNIX クライアントの前のバージョンの最大 64 個) を含めることができます。これにより、255 の LU の全 PU をマップすることができます。

マッピング・ファイルの例を以下に示します。

```
MyAppLU1 APPCLu5
MyAppLU2 APPCLu6
LUALU3    NewLUA11
LUALU4    NewLUA12
```

この例では、

- ローカル LU 別名 MyAppLU1 または MyAppLU2 を指定する、このクライアント上で実行されている APPC または CPI-C アプリケーションは、`lu_alias APPCLu5` または `APPCLu6` で実際のローカル LU を使用するようにマップされます。
- LU 名 LUALU3 または LUALU4 を指定する、このクライアント上で実行されている LUA アプリケーションは、`NewLUA11` または `NewLUA12` という名前のタイプ 0 から 3 の実際の LU を使用するようにマップされます。アプリケーションによって指定された LU 名が 8 文字より短いため、これらの 2 つの項目の追加スペースに注意してください。

さらに、以下の文字を `lualiasmap.txt` ファイル: 内で使用することもできます。

- + 文字 - 一部のレガシー LUA アプリケーション (LANDP など) は、戻された `session_id` および戻された LU を `RUI_INIT()` 応答で無視します。これが発生した場合、マッピングは有効になりません。`lualiasmap.txt` ファイルには、位置 1 に + 文字が含まれています。その後、LU マッピングは、`RUI_*()` のすべての verb の問題に適用されます (`RUI_INIT()` だけではありません)。の最初の行の場合
- * の文字 - ワイルドカードは、`lualiasmap.txt` ファイルに指定できます。最初のストリング内の * を使用して、複数の LU 別名を単一の LU (またはプール) 名にマップすることができます。例えば、次のような行があります。

```
pref*    LUALU
```

プリフで開始された別名を、その単一 LU にマップします。複数の別名を、両方の位置で * を使用してマップすることもできます。例えば、

```
alias*   LUNAM*
```

`will map alias01 to ルアルー 01 or alias7 to ルアルー 7 etc.` 名前の長さが 8 文字に制限されている場合は、接頭部の長さを変更できます。2 番目の位置にある * はサポートされていません。

- 文字 - サーバー上で認識されない LU 別名を使用してアプリケーションがセッションを開こうとすると (または、実際には任意のサーバー)、その `RUI_INIT()` verb はクライアントからサーバーに送信され、エラーはクライアントに戻されます。場合によっては、アプリケーションは再試行され、認識されない LU 別名がいくつかありません。これにより、不必要なトラフィックと CPU 使用率が - が 10 桁目 (2 番

目のストリングの代わりに)に配置されている場合は、これを防止できます。これは*ワイルドカード構文と一緒に使用することができます。例えば、次のようなファイルを使用します。

```
alias1  alias1
alias2  alias2
*       -
```

2つの別名別名 1 と alias2 をサーバーに送信することができます (マッピングは、この例では、ノーオペレーションになります)。他の別名を指定している RUI_INIT() または TP_STARTED() は、メッセージをサーバーに送信せずに、クライアント・ライブラリー・コードから拒否されます。

異なるクライアントで実行されている TP を区別するためにローカル TP 名をオーバーライドする

AIX または Linux 上のリモート API クライアント上の tpnameipv4 ファイルは、クライアント上で実行されている APPC または CPI-C アプリケーションによって指定される TP 名を変更するように CS Linux に指示します。これにより、各クライアントは、固有の TP 名 (その IPv4 アドレスによって識別される) を使用します。これにより、同じアプリケーションを複数のクライアントで実行し、それらを強制的に使用して異なる TP 名を使用することができます。これにより、それらを区別できます。(アプリケーションによって指定された TP 名はローカル ID としてのみ使用され、CS Linux 構成内のどのパラメーターとも一致する必要はありません。そのため、アプリケーションは元の TP 名と同じ方法で実行されます。)

- APPC アプリケーションの場合、TP 名は TP_STARTED verb でアプリケーションによって指定されます。
- CPI-C アプリケーションの場合、TP 名は、Specify_Local_TP_Name または Set_CPIC_Side_Information を使用してアプリケーションによって指定するか、または CPI-C 構成から取得することができます。

If you need to use this function, use a standard text editor to create a file named tpnameipv4, in the directory /etc/sna on AIX or /etc/opt/ibm/sna on Linux. このファイルは空にすることができます。CS Linux は、ファイルが存在するかどうかのみを検査し、内容の通知は行いません。

クライアント上に tpnameipv4 ファイルが存在する場合、CS Linux はアプリケーションによって指定された TP 名を取り、下線文字とその後にクライアント・コンピューターの IP アドレスを付加します。この変更された TP 名ストリングは、このクライアント上で実行されるアプリケーションのロギングおよびトレースで使用されるため、異なるクライアント上で 2 つ以上のアプリケーション・コピーを区別できます。



クライアント TP の定義

Remote API Client システムでの TP の定義の詳細は、75 ページの『TP の定義』または 163 ページの『付録 B コマンド行からの呼び出し可能 TP の構成』を参照してください。

付録 A 構成計画ワークシート

この付録には、CS Linux の特定の機能を構成するためのワークシートが収録されています。ワークシートは、各機能を使用可能にするために必要な基本構成パラメーターをまとめたものです。拡張構成パラメーターについては、本書の本文の中のそれぞれ該当の項か、「*IBM Communications Server for Data Center Deployment on Linux* 管理コマンド解説書」を参照してください。

ノードを構成するために必要なすべての情報を収集するには、次のカテゴリーのワークシートを作成する必要があります。

ノード構成

ノードの機能と、そのノードが属しているネットワークの特性に応じて、[139 ページの『ノード・ワークシート』](#)に含まれるワークシートの 1 つに記入します。

接続の構成

ネットワーク内の他のシステムと通信するために使用するリンク・プロトコルに応じて、[142 ページの『接続ワークシート』](#)に含まれるワークシートの 1 つまたはいくつかに記入します。

パススルー・サービスの構成

ノードがサポートする必要があるパススルー・サービスについて、[151 ページの『パススルー・サービス・ワークシート』](#)に含まれるワークシートに記入します。

アプリケーション・サポートの構成

ノードがサポートする必要があるユーザー・アプリケーションのタイプに応じて、[155 ページの『ユーザー・アプリケーション・サポート・ワークシート』](#)に含まれるワークシートの 1 つまたはいくつかに記入します。

ノード・ワークシート

次のワークシートの 1 つに記入します。

- [139 ページの『APPN ネットワーク・ノード』](#)
- [140 ページの『APPN エンド・ノード』](#)
- [140 ページの『APPN 分岐ネットワーク・ノード』](#)
- [141 ページの『LEN ノード』](#)

APPN ネットワーク・ノード

ローカル・ノードが、APPN ネットワーク・ノード (APPN ネットワーク・ノード内のルーティング・サービスを提供するノード) である場合は、このワークシートに記入します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
「Node Parameters (ノード・パラメーター)」ダイアログ		
<i>APPN support</i>	Network node	
<i>Control point name</i>	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字) VTAM ホストに接続するには、この名前は、VTAM PU ステートメントの中の NETID= エントリーおよび CPNAME= エントリーに一致している必要があります。	
<i>Control point alias</i>	最大 8 文字	
<i>Node ID</i>	8 桁の 16 進数	

Motif フィールド	有効なエントリー/注	ユーザーの設定値
接続の構成: 142 ページの『接続ワークシート』 を参照		
クライアント/サーバー構成: スタンドアロン・ノードの場合は不要		
Configuration server?	ノードが、CS Linux LAN 内のドメイン・リソースに関する情報を保管する構成サーバーとして機能するかどうか。	
アプリケーションの構成: 155 ページの『ユーザー・アプリケーション・サポート・ワークシート』 を参照		

APPN エンド・ノード

ローカル・ノードが APPN エンド・ノード (動的ルーティング情報を使用できるが、他のノードのためのルーティング・サービスは提供しないノード) である場合は、このワークシートに記入します。

Motif フィールド	有効なエントリー/注	ユーザーの設定値
「Node Parameters (ノード・パラメーター)」ダイアログ		
APPN support	End node	
Control point name	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字) VTAM ホストに接続するには、この名前は、VTAM PU ステートメントの中の NETID= エントリーおよび CPNAME= エントリーに一致している必要があります。	
Control point alias	最大 8 文字	
Node ID	8 桁の 16 進数	
接続の構成: 142 ページの『接続ワークシート』 を参照		
クライアント/サーバー構成: スタンドアロン・ノードの場合は不要		
Configuration server?	ノードが、CS Linux LAN 内のドメイン・リソースに関する情報を保管する構成サーバーとして機能するかどうか。	
アプリケーションの構成: 155 ページの『ユーザー・アプリケーション・サポート・ワークシート』 を参照		

APPN 分岐ネットワーク・ノード

ローカル・ノードが APPN 分岐ネットワーク・ノードである場合 (メイン APPN ネットワークから分離された分岐で、エンド・ノードへのネットワーク・ノード機能を提供する一方で、メイン・ネットワーク自体ではエンド・ノードとして機能するノード)、このワークシートを完了します。

Motif フィールド	有効なエントリー/注	ユーザーの設定値
「Node Parameters (ノード・パラメーター)」ダイアログ		
APPN support	Branch network node	

Motif フィールド	有効なエントリー/注	ユーザーの設定値
<i>Control point name</i>	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字) VTAM ホストに接続するには、この名前は、VTAM PU ステートメントの中の NETID= エントリーおよび CPNAME= エントリーに一致している必要があります。	
<i>Control point alias</i>	最大 8 文字	
<i>Node ID</i>	8 桁の 16 進数	
接続の構成: 142 ページの『接続ワークシート』を参照		
クライアント/サーバー構成: スタンドアロン・ノードの場合は不要		
<i>Configuration server?</i>	ノードが、CS Linux LAN 内のドメイン・リソースに関する情報を保管する構成サーバーとして機能するかどうか。	
アプリケーションの構成: 155 ページの『ユーザー・アプリケーション・サポート・ワークシート』を参照		

LEN ノード

ローカル・ノードが LEN ノード (APPN 機能をサポートしないノード、またはホスト・コンピューターのみと通信するスタンドアロン・システム) である場合は、このワークシートを完成します。

Motif フィールド	有効なエントリー/注	ユーザーの設定値
「Node Parameters (ノード・パラメーター)」ダイアログ		
<i>APPN support</i>	LEN ノード	
<i>Control point name</i>	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字) VTAM ホストに接続するには、この名前は、VTAM PU ステートメントの中の NETID= エントリーおよび CPNAME= エントリーに一致している必要があります。	
<i>Control point alias</i>	最大 8 文字	
<i>Node ID</i>	8 桁の 16 進数	
接続の構成: 142 ページの『接続ワークシート』を参照		
クライアント/サーバー構成: スタンドアロン・ノードの場合は不要		
<i>Configuration server?</i>	ノードが、CS Linux LAN 内のドメイン・リソースに関する情報を保管する構成サーバーとして機能するかどうか。	
アプリケーションの構成: 155 ページの『ユーザー・アプリケーション・サポート・ワークシート』を参照		

接続ワークシート

他のノードとの通信に使用するそれぞれのリンク・プロトコルごとに、次のワークシートの内の1つを完成します。必要があれば、1つのポート上に複数のリンク・ステーションを構成することもできます。

- [142 ページの『SDLC』](#)
- [144 ページの『トークンリング』](#)
- [146 ページの『イーサネット』](#)
- [148 ページの『QLLC \(X.25\)』](#)
- [149 ページの『マルチパス・チャネル』](#)
- [150 ページの『Enterprise Extender \(HPR/IP\)』](#)

SDLC

SDLC リンク・プロトコルを使用する接続をサポートするには、このワークシートを完成します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
「SDLC Port (SDLC ポート)」ダイアログ		
<i>SNA port name</i>	最大 8 文字	
<i>SDLC card number</i>	0 から <i>number_of_cards_minus_1</i>	
<i>Port number</i>	0 から <i>number_of_ports_on_card_minus_1</i>	
<i>Initially active</i>	必要な場合に選択します。	
回線詳細		
<i>Type</i>	Leased line Switched outgoing Switched incoming	
<i>Link role</i>	折衝可能 1 次 Primary multi-drop 2 次 Secondary multi-PU	
Switched incoming (交換着呼回路) または Leased line (専用回線) の場合		
<i>Poll address</i>	非 1 次交換着呼ポートの場合のみ VTAM ホストでは、ポーリング・アドレスは VTAM PU 定義の中の <i>ADDR=</i> パラメーターとして構成されます。 AS/400 システムでは、ポーリング・アドレスは Line Description (回線記述) の <i>STNADR</i> パラメーターです。	
「SDLC Link Station (SDLC リンク・ステーション)」ダイアログ		
リンク・ステーションのフィールド		
<i>Name</i>	最大 8 文字	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>SNA port name</i>	最大 8 文字	
<i>Activation</i>	By administrator On node startup On demand	
<i>LU traffic</i>	Any Independent only Dependent only	
独立 LU トラフィック		
<i>Remote node</i>	<p><i>NETNAME.CPNAME</i> (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字。オプション)</p> <p>リモート・システムが VTAM ホストである場合は、VTAM スタート・リストの <i>NETID</i> パラメーターにネットワーク名 (完全修飾名の最初の 8 文字) があります。後半の 8 文字は、VTAM スタート・リストの <i>SSCPNAME</i> パラメーターに示されています。</p>	
<i>Remote node type</i>	Discover Network node End or LEN node	
従属 LU トラフィック		
<i>Remote node role</i>	Host Downstream (SNA Gateway) Downstream (DLUR)	
<i>Local node ID</i>	<p>8 桁の 16 進数 (デフォルトはノード名)</p> <p>VTAM 構成では、最初の 3 桁は PU 定義の <i>IDBLK</i> パラメーターに一致し、最後の 5 桁は <i>IDNUM</i> パラメーターに一致していることが必要です。</p> <p>AS/400 システムでは、ノード ID は <i>EXCHID</i> パラメーターで構成されています。</p>	
<i>Remote node ID</i>	8 桁の 16 進数 (オプション)	
<i>Downstream PU name</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>Upstream DLUS name</i>	<i>NETNAME.LUNAME</i> (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
連絡先情報		

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
Poll address	<p>交換着呼ポートの場合は、そのポート上でのみ構成されます。</p> <p>2 桁の 16 進数:</p> <ul style="list-style-type: none"> Point-to-Point の場合は C1 1 次交換発呼 (宛先アドレスは不明) の場合は 0xFF 1 次対マルチドロップの場合は固有アドレス <p>VTAM ホストでは、ポーリング・アドレスは VTAM PU 定義の中の ADDR= パラメーターとして構成されます。</p> <p>AS/400 システムでは、ポーリング・アドレスは Line Description (回線記述) の STNADR パラメーターです。</p>	

トークンリング

トークンリング・リンクを使用する接続をサポートするには、このワークシートを完成します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
「Token Ring SAP (トークンリング SAP)」ダイアログ		
SNA port name	最大 8 文字	
Token ring card number	0 から number_of_cards_minus_1	
Local SAP number	16 進数字 (4 の倍数)	
Initially active	必要な場合に選択します。	
Define on connection network	必要な場合に選択します。	
CN name	NETNAME.CNNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	

「Token Ring Link Station (トークンリング・リンク・ステーション)」ダイアログ

リンク・ステーションのフィールド		
Name	最大 8 文字	
SNA port name	最大 8 文字	
Activation	By administrator On node startup On demand	
LU traffic	Any Independent only Dependent only	

独立 LU トラフィック

Motif フィールド	有効なエントリー/注	ユーザーの設定値
<i>Remote node</i>	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字。オプション) リモート・システムが VTAM ホストである場合は、VTAM スタート・リストの NETID パラメーターにネットワーク名 (完全修飾名の最初の 8 文字) があります。後半の 8 文字は、VTAM スタート・リストの SSCPNAME パラメーターに示されています。	
<i>Remote node type</i>	Discover End or LEN node Network node	
従属 LU トラフィック		
<i>Remote node role</i>	Host Downstream (SNA Gateway) Downstream (DLUR)	
<i>Local node ID</i>	8 桁の 16 進数 (デフォルトはノード名) VTAM 構成では、最初の 3 桁は PU 定義の IDBLK パラメーターに一致し、最後の 5 桁は IDNUM パラメーターに一致していることが必要です。 AS/400 システムでは、ノード ID は EXCHID パラメーターで構成されています。	
<i>Remote node ID</i>	8 桁の 16 進数 (オプション)	
<i>Downstream PU name</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>Upstream DLUS name</i>	NETNAME.LUNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
連絡先情報		
<i>MAC address</i>	16 進数字 このリンクのリモート・エンドが VTAM ホストである場合は、MAC アドレスは、VTAM ポート定義の MACADDR= パラメーターに示されています。 AS/400 システムへのリンクを構成している場合は、MAC アドレスは、回線記述の中の ADPTADR パラメーターです。	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>SAP number</i>	16 進数字 (4 の倍数) このリンクのリモート・エンドが VTAM ホストである場合は、SAP 番号は VTAM PU 定義の <i>SAPADDR</i> = パラメーターです。 AS/400 システムへのリンクを構成している場合は、SAP 番号は、回線記述 (Line Description) の中の <i>SSAP</i> パラメーターです。	

イーサネット

イーサネット・リンク・プロトコルを使用する接続をサポートするには、このワークシートを完成します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
「Ethernet SAP (イーサネット SAP)」 ダイアログ		
<i>SNA port name</i>	最大 8 文字	
<i>Ethernet card number</i>	0 から <i>number_of_cards_minus_1</i>	
<i>Device name</i>	オプションでカード番号の代わりに使用します。	
<i>Local SAP number</i>	16 進数字 (4 の倍数)	
<i>Initially active</i>	必要な場合に選択します。	
<i>Define on connection network</i>	必要な場合に選択します。	
<i>CN name</i>	<i>NETNAME.CNNAME</i> (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
<i>Ethernet type</i>	標準、または 802.3 を選択します。	
「Ethernet Link Station (イーサネット・リンク・ステーション)」 ダイアログ		
リンク・ステーションのフィールド		
<i>Name</i>	最大 8 文字	
<i>SNA port name</i>	最大 8 文字	
<i>Activation</i>	By administrator On node startup On demand	
<i>LU traffic</i>	Any Independent only Dependent only	
独立 LU トラフィック		

Motif フィールド	有効なエントリー/注	ユーザーの設定値
<i>Remote node</i>	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字。オプション) リモート・システムが VTAM ホストである場合は、VTAM スタート・リストの NETID パラメーターにネットワーク名 (完全修飾名の最初の 8 文字) があります。後半の 8 文字は、VTAM スタート・リストの SSCPNAME パラメーターに示されています。	
<i>Remote node type</i>	Discover Network node End or LEN node	
従属 LU トラフィック		
<i>Remote node role</i>	Host Downstream (SNA Gateway) Downstream (DLUR)	
<i>Local node ID</i>	8 桁の 16 進数 (デフォルトはノード名) VTAM 構成では、最初の 3 桁は PU 定義の IDBLK パラメーターに一致し、最後の 5 桁は IDNUM パラメーターに一致していることが必要です。 AS/400 システムでは、ノード ID は EXCHID パラメーターで構成されています。	
<i>Remote node ID</i>	8 桁の 16 進数 (オプション)	
<i>Downstream PU name</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>Upstream DLUS name</i>	NETNAME.LUNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
連絡先情報		
<i>MAC address</i>	16 進数字 このリンクのリモート・エンドが VTAM ホストである場合は、MAC アドレスは、VTAM ポート定義の MACADDR= パラメーターに示されています。 AS/400 システムへのリンクを構成している場合は、MAC アドレスは、回線記述の中の ADPTADR パラメーターです。	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
SAP number	16 進数字 (4 の倍数) このリンクのリモート・エンドが VTAM ホストである場合は、SAP 番号は VTAM PU 定義の SAPADDR= パラメーターです。 AS/400 システムへのリンクを構成している場合は、SAP 番号は、回線記述 (Line Description) の中の SSAP パラメーターです。	

QLLC (X.25)

QLLC (X.25) リンク・プロトコルを使用する接続をサポートするには、このワークシートを完成します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
「QLLC Port (QLLC ポート)」ダイアログ		
SNA port name	最大 8 文字	
X.25 card number	0 から number_of_cards_minus_1	
Initially active	必要な場合に選択します。	
「QLLC Link Station (QLLC リンク・ステーション)」ダイアログ		
リンク・ステーションのフィールド		
Name	最大 8 文字	
SNA port name	最大 8 文字	
Activation	By administrator On node startup On demand	
LU traffic	Any Independent only Dependent only	
独立 LU トラフィック		
Remote node	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字。オプション) リモート・システムが VTAM ホストである場合は、VTAM スタート・リストの NETID パラメーターにネットワーク名 (完全修飾名の最初の 8 文字) があります。後半の 8 文字は、VTAM スタート・リストの SSCPNAME パラメーターに示されています。	
Remote node type	Discover Network node End or LEN node	
従属 LU トラフィック		

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>Remote node role</i>	Host Downstream (SNA Gateway) Downstream (DLUR)	
<i>Local node ID</i>	8 桁の 16 進数 (デフォルトはノード名) VTAM 構成では、最初の 3 桁は PU 定義の <i>IDBLK</i> パラメーターに一致し、最後の 5 桁は <i>IDNUM</i> パラメーターに一致していることが必要です。 AS/400 システムでは、ノード ID は <i>EXCHID</i> パラメーターで構成されています。	
<i>Remote node ID</i>	8 桁の 16 進数 (オプション)	
<i>Downstream PU name</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>Upstream DLUS name</i>	<i>NETNAME.LUNAME</i> (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
連絡先情報		
<i>Remote X.25 address</i>	16 進数字 (SVC の場合のみ)、1-4096 (PVC の場合のみ)	

マルチパス・チャネル

マルチパス・チャネル・リンク・プロトコルを使用する接続をサポートするには、このワークシートに記入します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
「Multipath Channel Port (マルチパス・チャネル・ポート)」ダイアログ		
<i>SNA port name</i>	最大 8 文字	
<i>Port number</i>	マルチパス・チャネル装置の装置番号に一致している必要があります。	
<i>Initially active</i>	必要な場合に選択します。	
「Multipath Channel Link Station (マルチパス・チャネル・リンク・ステーション)」ダイアログ		
リンク・ステーションのフィールド		
<i>Name</i>	最大 8 文字	
<i>SNA port name</i>	最大 8 文字	
<i>Activation</i>	By administrator On node startup On demand	
独立 LU トラフィック		

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>Remote node</i>	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字。オプション) リモート・システムが VTAM ホストである場合は、VTAM スタート・リストの NETID パラメーターにネットワーク名 (完全修飾名の最初の 8 文字) があります。後半の 8 文字は、VTAM スタート・リストの SSCPNAME パラメーターに示されています。	
<i>Remote node type</i>	Discover Network node End or LEN node	

Enterprise Extender (HPR/IP)

Enterprise Extender リンク・プロトコルを使用する接続をサポートするには、このワークシートに記入します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
「Enterprise Extender Port (Enterprise Extender ポート)」ダイアログ		
<i>SNA port name</i>	最大 8 文字	
<i>Initially active</i>	必要な場合に選択します。	
<i>Protocol</i>	このポートのリンク・ステーションで IPv4 または IPv6 アドレスを使用するかどうか。	
<i>Local IP interface</i>	複数の IP ネットワークにアクセスできる場合は、IP リンクに使用するローカル・ネットワーク・アダプター・カードの ID。1 つの IP ネットワークにのみアクセスできる場合は、このフィールドはブランクのままにできます。	
<i>Define on connection network</i>	必要な場合に選択します。	
<i>CN name</i>	NETNAME.CNNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
「Enterprise Extender Link Station (Enterprise Extender リンク・ステーション)」ダイアログ		
リンク・ステーションのフィールド		
<i>Name</i>	最大 8 文字	
<i>SNA port name</i>	最大 8 文字	
<i>Activation</i>	By administrator On node startup On demand	
独立 LU トラフィック		

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
Remote node	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字。オプション) リモート・システムが VTAM ホストである場合は、VTAM スタート・リストの NETID パラメーターにネットワーク名 (完全修飾名の最初の 8 文字) があります。後半の 8 文字は、VTAM スタート・リストの SSCPNAME パラメーターに示されています。	
Remote node type	Discover End or LEN node Network node	

連絡先情報

Remote IP host name	IPv4 ドット 10 進アドレス (193.1.11.100 など)、IPv6 コロン付き 16 進アドレス (2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab など)、名前 (newbox.this.co.uk など)、または別名 (newbox など)。ポートの protocol パラメーターは、アドレスが IPv4 または IPv6 のいずれであるかを決定します。 名前または別名を指定した場合、これを Linux システムが (ローカル TCP/IP 構成を使用するか、ドメイン・ネーム・サーバーを使用して) 完全修飾名に解決できなければなりません。
---------------------	---

パススルー・サービス・ワークシート

次の各項で述べるパススルー・サービスのどれかをローカル・ノードでサポートする必要がある場合は、それぞれ該当のワークシートに記入します。

- [151 ページの『ローカル・ノード上の DLUR』](#)
- [152 ページの『ダウンストリーム・ノード用のパススルー DLUR』](#)
- [152 ページの『SNA ゲートウェイ』](#)
- [153 ページの『TN サーバー』](#)
- [154 ページの『TN Redirector』](#)

ローカル・ノード上の DLUR

ローカル・ノード上の DLUR をサポートするには、このワークシートを完成します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
	ノードの構成: 139 ページの『ノード・ワークシート』 を参照	
	接続の構成: 142 ページの『接続ワークシート』 を参照 ローカル・ノード上の DLUR をサポートするには、APPN ネットワークへの接続を構成します。	
DLUR PU:	.	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>PU name</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>DLUS name</i>	<i>NETNAME.LUNAME</i> (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
<i>Backup DLUS name</i>	このパラメーターはオプションです。 <i>NETNAME.LUNAME</i> (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
<i>PU ID</i>	8 桁の 16 進数 VTAM 構成では、最初の 3 桁は PU 定義の <i>IDBLK</i> パラメーターに一致し、最後の 5 桁は <i>IDNUM</i> 設定に合致している必要があります。 AS/400 システムでは、PU ID は <i>EXCHID</i> パラメーターで構成されています。	
<i>Initially active</i>	必要な場合に選択します。	
<i>Compression supported</i>	必要な場合に選択します。	
<i>Retry contacting DLUS indefinitely</i>	必要な場合に選択します。	

ローカル LU およびアプリケーション構成: [155 ページの『ユーザー・アプリケーション・サポート・ワークシート』](#)を参照 ローカル従属 LU および必要なアプリケーション・サポートを構成する必要があります。

ダウンストリーム・ノード用のパススルー DLUR

ローカル・ノードが APPN ネットワーク・ノードである場合は、ダウンストリーム・ノードのためのパススルー DLUR サービスを提供できます。DLUR をサポートするには、このワークシートを完成します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
	ノードの構成: ノードをネットワーク・ノードとして構成します (139 ページの『APPN ネットワーク・ノード』 を参照)。	
	接続の構成: 142 ページの『接続ワークシート』 を参照 APPN ネットワークへの接続のほか、ダウンストリーム・ノードへの従属トラフィック用の接続も構成します。	

SNA ゲートウェイ

ローカル・ノードが SNA ゲートウェイをサポートする必要がある場合は、このワークシートを完成します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
	ノードの構成: 139 ページの『ノード・ワークシート』 を参照	
	接続の構成: 142 ページの『接続ワークシート』 を参照 ホストへの従属トラフィックのための接続、および各ダウンストリーム・ノードのためのリンクを構成します。	
	ローカル LU およびアプリケーション構成: 155 ページの『ユーザー・アプリケーション・サポート・ワークシート』 を参照	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
「LU Pool (LU プール)」 ダイアログ		
<i>Pool name</i>	1 から 8 個のタイプ AE の EBCDIC 文字	
<i>LU lists</i>	プールに割り当てる LU (タイプ 0 から 3) の名前	
「Downstream LU (ダウンストリーム LU)」 ダイアログ		
<i>Downstream LU name</i>	1 から 8 個のタイプ A の EBCDIC 文字 (一連の LU のベース名の場合は 1 から 5 文字)	
<i>Downstream PU name</i>	タイプ A の EBCDIC ストリング	
<i>LU numbers</i>	1 から 255 (範囲を指定する場合は、最初と最後の番号を入力)	
<i>Upstream LU name</i>	タイプ A の EBCDIC ストリング (LU 名の場合) または タイプ AE の EBCDIC ストリング (LU プール名の場合)	

TN サーバー

ローカル・ノードが TN3270 クライアントをサポートする必要がある場合は、このワークシートを完成します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
ノードの構成: 139 ページ の『ノード・ワークシート』を参照		
接続の構成: 142 ページ の『接続ワークシート』を参照 (ホストへの 従属 LU トラフィック用の構成)。		
ローカル LU およびアプリケーション構成: 155 ページ の『ユーザー・アプリケーション・サポート・ワークシート』を参照		
「LU Pool (LU プール)」 ダイアログ		
<i>Pool name</i>	1 から 8 個のタイプ AE の EBCDIC 文字	
<i>LU lists</i>	プールに割り当てる LU (タイプ 0 から 3) の名前	
「TN Server Access (TN サーバー・アクセス)」 ダイアログ		
<i>TN3270 client address</i>	次のどれかを指定します。 <ul style="list-style-type: none"> • Default record (任意の TN3270 クライアント) • TCP/IP address (クライアントの IP アドレス、IPv4 アドレスまたは IPv6 アドレスのいずれか) • TCP/IP 名または別名 (TCP/IP name or alias) 	
<i>Support TN3270E</i>	TN3270 および TN3287 に加えて TN3270E をサポートする場合に選択します。	
TN3270 ポートおよび LU		
<i>TCP/IP port number</i>	通常は 23	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>Display LU Assigned</i>	LU またはプール名	
<i>Printer LU Assigned</i>	LU またはプール名	
<i>Allow access to specific LU</i>	必要な場合に選択します。	
<i>SSL secure session</i>	必要な場合に選択します。	
<i>Perform client authentication</i>	必要な場合に選択します。	
<i>Encryption strength</i>	次のどれかを指定します。 <ul style="list-style-type: none"> • Authenticate Only • Authenticate Minimum • 40 Bit Minimum • 56 Bit Minimum • 128 Bit Minimum • 168 Bit Minimum 	
「TN Server Association (TN サーバー・アソシエーション)」ダイアログ		
<i>Display LU</i>	LU 名	
<i>Printer LU</i>	LU 名	

TN Redirector

ローカル・ノードが TN リダイレクターを使用して Telnet クライアントをサポートする必要がある場合は、このワークシートを完成します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
ノードの構成: 139 ページ の『ノード・ワークシート』を参照		
「TN Redirector Access (TN リダイレクター・アクセス)」ダイアログ		
<i>Telnet client address</i>	次のどれかを指定します。 <ul style="list-style-type: none"> • Default record (任意の Telnet クライアント) • TCP/IP address (クライアントの IP アドレス、IPv4 アドレスまたは IPv6 アドレスのいずれか) • TCP/IP 名または別名 (TCP/IP name or alias) 	
<i>TCP/IP port number</i>	通常は 23	
<i>SSL secure session</i>	必要な場合に選択します。	
<i>Perform client authentication</i>	必要な場合に選択します。	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>Encryption strength</i>	次のどれかを指定します。 <ul style="list-style-type: none"> • Authenticate Only • Authenticate Minimum • 40 Bit Minimum • 56 Bit Minimum • 128 Bit Minimum • 168 Bit Minimum 	
<i>Host address</i>	次のどれかを指定します。 <ul style="list-style-type: none"> • TCP/IP address (ホストの IP アドレス、IPv4 アドレスまたは IPv6 アドレスのいずれか) • TCP/IP 名または別名 (TCP/IP name or alias) 	
<i>TCP/IP port number</i>		
<i>SSL secure session</i>	必要な場合に選択します。	
<i>Encryption strength</i>	次のどれかを指定します。 <ul style="list-style-type: none"> • Authenticate Only • Authenticate Minimum • 40 Bit Minimum • 56 Bit Minimum • 128 Bit Minimum • 168 Bit Minimum 	

ユーザー・アプリケーション・サポート・ワークシート

ローカル・ノードが次のいずれかのユーザー・レベル・アプリケーションをサポートする必要がある場合は、それぞれ該当のワークシートを完成します。

- [155 ページの『APPC』](#)
- [159 ページの『CPI-C』](#)
- [159 ページの『5250』](#)
- [160 ページの『3270』](#)
- [160 ページの『LUA』](#)

APPC

ローカル・ノードが APPC アプリケーションをサポートする必要がある場合は、このワークシートを完成します。

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
	ノードの構成: 139 ページの『ノード・ワークシート』 を参照	
	接続の構成: 142 ページの『接続ワークシート』 を参照	

Motif フィールド	有効なエントリー/注	ユーザーの設定値
「Local LU (ローカル LU)」 ダイアログ: デフォルトの制御点 LU を使用できる場合は不要		
LU 名	1 から 8 個のタイプ A の EBCDIC 文字	
LU alias	最大 8 文字	
従属 LU パラメーター		
Host LS/DLUR PU	ホストへの従属リンク・ステーションの名前、または DLUR PU (LU の定義の前に定義する必要があります)。	
LU number	1 から 255 この値は、VTAM/NCP LU リソース定義ステートメントの中の LOCADDR パラメーターに一致していなければなりません。	
Member of default pool	必要な場合に選択します (従属 LU の場合のみ)。	
ローカル LU パラメーター		
Support syncpoint	必要な場合に選択します。	
Disable password substitution	必要な場合に選択します。	
Restrict to specific SSCP	必要な場合に選択します (従属 LU の場合のみ)。 SSCP ID は VTAM 起動リストの「SSCPID=」フィールドで定義されます。	
「Remote Node (リモート・ノード)」 ダイアログ: ローカル・ノードが LEN ノードの場合に限り構成		
Node's SNA network name	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
「Partner LU (パートナー LU)」 ダイアログ: LEN ノードとの通信の場合 (パートナー LU 別名を定義するため)、またはローカル・ノードが LEN ノードである場合に限り必要		
Partner LU name	NETNAME.LUNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
Alias	最大 8 文字	
Uninterpreted name	1 から 8 個のタイプ AE の EBCDIC 文字 (ホスト LU 名がローカルで使用されている PLU 名と異なる場合)	
Supports parallel sessions	サポートされている場合に選択します。	
Location	NETNAME.CPNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
「LS Routing (LS ルーティング)」 ダイアログ: リンク・ステーションがパートナー LU を見つける場合に限り必要		
LU 名	1 から 8 個のタイプ A の EBCDIC 文字	
LS name	最大 8 文字	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>Partner LU name</i>	NETNAME.LUNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
<i>Use partner LU name as a wildcard</i>	必要な場合に選択します。	
「Mode (モード)」 ダイアログ: 標準外モードを使用している場合に限り必要		
<i>Name</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>COS name</i>	1 から 8 個のタイプ A の EBCDIC 文字	
セッション限度		
<i>Initial session limit</i>	最大セッション限度まで。推奨値は 8	
<i>Maximum session limit</i>	最大 32767	
<i>Minimum contention winner sessions</i>	最大セッション限度まで。推奨値は 0	
<i>Minimum contention loser sessions</i>	推奨値は 0	
<i>Auto-activated sessions</i>	0 から <i>minimum_contention_winners</i>	
受信ペーシング・ウィンドウ		
<i>Initial window size</i>	推奨値は 4	
<i>Maximum window size</i>	オプション	
<i>Session timeout</i>		
<i>Maximum RU size</i>	推奨上限値は 1024	
Compression supported		
<i>Max inbound compression</i>	None	
	RLE	
	LZ9	
	LZ10	
<i>Max outbound compression</i>	None	
	RLE	
	LZ9	
	LZ10	
「Session Security (セッション・セキュリティ)」 ダイアログ: 特定の ローカル LU とパートナー LU の間のセッションにセッション・セキュリティ が必要な場合に限り必要		
<i>Local LU</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>Partner LU</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>Password</i>	16 桁の 16 進数値	
「TP Invocation (TP 呼び出し)」 ダイアログ: リモート・システムから の要求に応じてローカル TP を開始する必要がある場合に限り必要		

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>TP name</i>	ユーザー・アプリケーション: 最大 64 個の ASCII 文字 サービス TP: 最大 8 桁の 16 進数	
<i>Restrict to specific LU</i>	必要な場合に選択します。	
<i>LU alias</i>	最大 8 文字	
<i>Multiple instances supported</i>	非キューの TP を選択します。これを選択しなかった場合、TP が既に実行中のときには着呼割り振り要求はキューに入ります。	
<i>Route incoming Allocates to running TP</i>	ブロードキャスト待機 TP の場合に選択します。	
<i>Full path to TP executable</i>	実行可能ファイルのパスおよびファイル名 (デフォルトは「 <i>TP name</i> 」)	
<i>Arguments</i>	実行可能ファイルに渡す有効な引数	
<i>User ID</i>	最大 64 文字	
<i>Group ID</i>	最大 64 文字	
「TP Definition (TP 定義)」 ダイアログ: APPC の特性を定義します。		
<i>TP name</i>	ユーザー・アプリケーション: 最大 64 個の ASCII 文字 サービス TP: 最大 8 桁の 16 進数	
<i>Conversation level security required</i>	割り振り要求時に有効なユーザー名およびパスワードが必要であることを指定するために選択します。	
<i>Restrict access</i>	セキュリティ・アクセス・リストにユーザー名が含まれていなければならないことを指定するために選択します。	
<i>Security access list</i>	セキュリティ・アクセス・リストの名前	
<i>Conversation type</i>	Basic Mapped Either	
<i>Sync level</i>	None Confirm Sync-point None または Confirm None、Confirm、または Sync-point	
<i>PIP allowed</i>	必要な場合に選択します。	
会話セキュリティ・ダイアログ: リモート・システムからの要求に応じて開始する必要があるローカル TP について、会話セキュリティが必要な場合に限り必要		
<i>User ID</i>	最大 10 文字	

Motif フィールド	有効なエントリー/注	ユーザーの設定値
Password	最大 10 文字	

CPI-C

ローカル・ノードが CPI-C アプリケーションをサポートする必要がある場合は、このワークシートを完成します。

Motif フィールド	有効なエントリー/注	ユーザーの設定値
	ノードの構成: 139 ページ の『ノード・ワークシート』を参照	
	接続の構成: 142 ページ の『接続ワークシート』を参照	
	APPC の構成: 155 ページ の『APPC』を参照	
CPI-C 宛先ダイアログ		
Symbolic destination name	1 から 8 文字	
Local LU	別名 (最大 8 文字) または完全修飾名 (NETNAME.LUNAME、それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
Partner LU	別名 (最大 8 文字) または完全修飾名 (NETNAME.LUNAME、それぞれ 1 から 8 個のタイプ A の EBCDIC 文字)	
Mode	タイプ A の EBCDIC スtring	
パートナー TP 名	ユーザー・アプリケーション: 最大 64 文字 サービス TP: 最大 8 桁の 16 進数 (例: 06F3F0F1) (0x は使用しないでください)	
Security	None Same Program	
User ID	security level が Same または Program の場合のみ (ユーザー・ログイン ID に関連していないもの)	
Password	security level が Program の場合のみ (ユーザー・ログイン・パスワードに関連していないもの)	

5250

ローカル・ノードが 5250 通信をサポートする必要がある場合は、このワークシートを完成します。

Motif フィールド	有効なエントリー/注	ユーザーの設定値
	ノードの構成: 139 ページ の『ノード・ワークシート』を参照	
	接続の構成: 142 ページ の『接続ワークシート』を参照 (独立トラフィック用の構成)	

Motif フィールド	有効なエントリー/注	ユーザーの設定値
APPC の構成: 155 ページの『APPC』 を参照		

3270

ローカル・ノードが 3270 通信をサポートする必要がある場合は、このワークシートを完成します。

Motif フィールド	有効なエントリー/注	ユーザーの設定値
ノードの構成: 139 ページの『ノード・ワークシート』 を参照		

接続の構成: [142 ページの『接続ワークシート』](#) を参照 (従属トラフィック用の構成)

「LU Type 0-3 (LU タイプ 0 から 3)」ダイアログ

LU 名	1 から 8 個のタイプ A の EBCDIC 文字 (または、LU 範囲のベース名の場合は 1 から 5 文字)
Host LS/DLUR PU	ホストへの従属リンク・ステーションの名前、または DLUR PU (LU の定義の前に定義する必要があります)。
LU numbers	1 から 255 (範囲を指定する場合は、最初と最後の番号を入力) この値は、VTAM/NCP LU リソース定義ステートメントの中の LOCADDR パラメーターに一致していなければなりません。
LU type	3270 model 2 (80x24) ディスプレイ 3270 model 3 (80x32) ディスプレイ 3270 model 4 (80x43) ディスプレイ 3270 model 5 (132x27) ディスプレイ 3270 Printer (3270 プリンター) SCS Printer (SCS プリンター)
LU in pool	使用したいオプションを選択します (ディスプレイおよび非制限 LU の場合のみ)
Pool name	1 から 8 個のタイプ AE の EBCDIC 文字

「LU Pool (LU プール)」ダイアログ

Pool name	1 から 8 個のタイプ AE の EBCDIC 文字
LU lists	プールに割り当てる LU (タイプ 0 から 3) の名前

LUA

ローカル・ノードが LUA アプリケーションをサポートする必要がある場合は、このワークシートを完成します。

Motif フィールド	有効なエントリー/注	ユーザーの設定値
ノードの構成: 139 ページの『ノード・ワークシート』 を参照		
接続の構成: 142 ページの『接続ワークシート』 を参照 (従属トラフィック用の 構成)		
「LU Type 0-3 (LU タイプ 0 から 3)」ダイアログ		
LU 名	1 から 8 個のタイプ A の EBCDIC 文字 (または、LU 範囲のベース名の場合は 1 から 5 文字)	
Host LS/DLUR PU	ホストへの従属リンク・ステーションの名前、または DLUR PU (LU の定義の前に定義する必要があります)。	
LU numbers	1 から 255 (範囲を指定する場合は、最初と最後の番号を入力) この値は、VTAM/NCP LU リソース定義ステートメントの中の LOCADDR パラメーターに一致していなければなりません。	
LU type	Unrestricted	
LU in pool	使用したいオプションを選択します (ディスプレイおよび非制限 LU の場合のみ)	
Pool name	1 から 8 個のタイプ AE の EBCDIC 文字	
「LU Pool (LU プール)」ダイアログ		
Pool name	1 から 8 個のタイプ AE の EBCDIC 文字	
LU lists	プールに割り当てる LU (タイプ 0 から 3) の名前	

付録 B コマンド行からの呼び出し可能 TP の構成

CS Linux には、TP インストール・プログラムのユーザーまたはライターが、呼び出し可能 TP を定義できるようにするコマンド行ユーティリティーが組み込まれています。このユーティリティーは、サーバーまたはクライアント上で実行できます。

コマンドの構文は、TP 定義を定義、除去、または照会するかどうかによって異なります。また、Windows 上の Remote API Client では異なります。

WINDOWS

tpinst32 コマンドは、クライアント・ソフトウェアがインストールされているディレクトリーに変更します。を使用する前に

tpinst32 コマンドは、通常、クライアント・ソフトウェアのインストール時に選択した言語で情報を表示します。この情報を別の言語で表示したい場合は、コマンドを実行する前に、希望する言語に対応するサブディレクトリーに移動してください。例えば、フランス語で情報が表示されるようにするには、クライアント・ソフトウェアがインストールされているディレクトリーの下の fr_FR サブディレクトリーに移動します。

呼び出し可能 TP を定義します。

UNIX

snatpinstall -a ファイル名

WINDOWS

tpinst32 -a ファイル名

注: tpinst32 コマンドは、32 ビット版と x64 バージョンの Windows の両方に適用されます。

このコマンドは、指定したファイル名から 1 つ以上の TP 定義を追加します。ファイルに指定されている TP がすでに定義されている場合は、ファイル内の情報によって既存の定義が置き換えられます。必要なファイル・フォーマットについては、[164 ページの『呼び出し可能 TP 定義のファイル形式』](#)を参照してください。

呼び出し可能 TP 定義を除去します。

UNIX

snatpinstall -r -t 名前変更 [- -l LU_alias]

このコマンドは、指定された TP 名を持つ項目を除去し、(複数の APPC TP が同じ TP 名で定義されている場合は)、指定された LU 別名を除去します。CPI-C アプリケーション用の項目がある場合、または指定した TP 名で定義された APPC TP が 1 つしかない場合は、オプション - -l LU_alias を取り除いてください。

WINDOWS

tpinst32 -r -t 名前変更

このコマンドは、指定された TP 名を持つエントリーを除去します。

呼び出し可能 TP 定義の照会：

UNIX

```
snatpinstall -q[-t 名前変更][[- -l LU_alias]
```

このコマンドは、指定された TP 名を持つエントリーを照会し、(同じ TP 名を使用して複数の APPC TP が定義されている場合は) 指定された LU 別名を照会します。CPI-C アプリケーション用の項目がある場合、または指定した TP 名で定義された APPC TP が 1 つしかない場合は、オプション `-l LU_alias` を取り除いてください。 `-t` 名前変更オプションを指定しないと、コマンドは呼び出し可能なすべての TP 定義を照会します。

WINDOWS

```
tpinst32 -q -t 名前変更
```

このコマンドは、指定された TP 名を持つエントリーを照会します。 `-t` 名前変更オプションを指定しないと、コマンドは呼び出し可能なすべての TP 定義を照会します。

呼び出し可能 TP 定義のファイル形式

呼び出し可能 TP に関する構成情報を提供するファイルは、任意の標準 テキスト・エディターで変更できる ASCII テキスト・ファイルです。このファイルの中の各エントリーの形式は次のとおりです。

```
[TPname]
PATH          = full_pathname_of_executable_file
ARGUMENTS    = command-line_arguments_separated_by_spaces
TYPE         = QUEUED | QUEUED-BROADCAST | NON-QUEUED
TIMEOUT      = nnn
```

UNIX

```
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
```

パラメーターの意味は以下のとおりです。オペレーターが開始する TP の場合は、使用されるパラメーターは、TP 名、TP タイプ、およびタイムアウト値、および (AIX または Linux 上の APPC TP に対する) LU 別名のみです。その他のパラメーターは自動開始の TP のみに適用されます。

UNIX

AIX または Linux では、CS Linux は無効なパラメーターが入力されると、エラー・メッセージを戻します。

WINDOWS

Windows マシンでは、CS Linux は無効なパラメーターを無視します。

TPname

TP の名前 (1 から 64 文字で、途中にスペースを含まないもの)。Receive_Allocate または着信割り振り要求で指定されている TP 名が、この名前と突き合わされます。自動開始 TP の場合、その開始時に Receive_Allocate にこの TP 名が指定されていることが必要です。これにより、CS Linux は着呼接続を正しい TP に経路指定することができます。

この名前は大括弧で囲む必要があります。この名前は、二重引用符で囲んだ ASCII スtring として指定できます (例えば、["TPNAME1"])。また、TP 名を EBCDIC 文字で表した 16 進数配列 (例えば [<35304E414D45F1>])、または 2 つの組み合わせ (例えば [<3f>"TP1"]) として指定することができます。この例では、先頭文字は印刷不能文字 0x3f で、そのあとに続く文字は "TP1" です。

CS Linux は、提供された ASCII スtring を EBCDIC に変換しますが、16 進数スtring については (既に EBCDIC であるとみなされるため) 変換を行いません。さらに、CS/AIX は、右側に EBCDIC のスペースを埋め込み (総文字数が 64 になるまで)、そのうえで指定されている TP 名と突き合わせます。

PATH

この TP 用の実行可能ファイルのパスとファイル名。

この行はオプションです。この行を含めなかった場合、CS Linux は実行可能ファイル名が TP 名と同じであるとみなします。パスなしでファイル名を指定した場合、AIX または Linux システムでは、デフォルト・パスは /etc/opt/ibm/sna になります。Windows クライアントの場合、CS Linux は、Windows の通常の機構を使用して実行可能ファイルを探します。

ARGUMENTS

TP に渡すコマンド行引数を、スペースで区切って指定します。これらの引数は、コマンド行上での順序と同じ順序で TP に渡されます。

この行はオプションです。これを省略した場合は、コマンド行引数なしで TP が呼び出されます。

TYPE

値は次のいずれかです。

QUEUED

この TP は待機 TP です。この TP の実行中に到着した着呼割り振り要求は、TP が新しい Receive_Allocate を発行するか、または TP の実行が終わって再始動が可能になるまで、待機状態になります。着信割り振り要求がこの TP に送付されるのは、着信割り振り要求をこのコンピューターに送付するように構成されている LU が、その要求を受信した場合だけです。

QUEUED-BROADCAST

この TP は、ブロードキャスト待機 TP です。この TP の実行中に到着した着呼割り振り要求は、TP が新しい Receive_Allocate を発行するか、または TP の実行が終わって再始動が可能になるまで、待機状態になります。この TP が開始されると、この TP に関する情報が LAN 上のすべてのサーバーにブロードキャストされます。別のコンピューター上の LU が着信割り振り要求を受信した場合に、ルーティング情報が構成されていないときは、その LU は動的にこの TP を見つけて、それに対して割り振り要求を送付します。

QUEUED の代わりに QUEUED-BROADCAST を使用すれば、LU に関する明示的なルーティング情報を構成する必要はなく、しかも、同じ TP の複数のコピーを別々のコンピューターで実行することで負荷が均衡化されます。ただし、LAN トラフィックを減らすために情報のブロードキャストを避けたい場合、または、特定の LU に到着した着信割り振り要求が常に TP の同じコピーに送付されるようにしたい場合は、QUEUED を使用してください。

NON-QUEUED

この TP は非待機 TP です。CS Linux は、この TP を宛先とする着呼割り振り要求が到着するたびに、この TP の新しいコピーを開始します。非待機 TP の場合は、TIMEOUT パラメーターは指定しないでください。

非待機と定義された TP は、オペレーターが開始することはできません。常に CS Linux により自動的に開始されます。オペレーターが開始する TP の場合は、NON-QUEUED を指定しないでください。ユーザーが非待機 TP を開始しようとした場合、CS Linux は、Receive_Allocate を待っている着信割り振り要求がないため、Receive_Allocate をリジェクトします。

非待機の TP が会話を終了したあとで、TP を終了させるか、または別の `Receive_Allocate` を発行することができます。頻繁に使用されるプログラムについては、各会話でプログラムの新しいインスタンスを開始するという効率のオーバーヘッドを回避できます。非待機の自動開始済み TP に対する接続を受信するたびに、CS Linux は、この TP のインスタンスから出された未解決の `Receive_Allocate` が既に存在するかどうか検査します。未解決がある場合、この TP は着呼の会話に使用されます。そうでない場合は、CS Linux はプログラムの新しいインスタンスを開始します。

`NON-QUEUED` を使用した場合は、TP の複数のコピーを同時に実行できます。TP がファイルに書き込みをする場合、TP のさまざまなコピーが、互いに相手のファイルへ上書きしないように注意する必要があります。そのためには、次のどちらかの方法を使用してください。

- TP が、ファイルを新規に作成しないで、既存のファイルにデータが追記するようにする（つまり、TP のすべてのコピーが同じファイルにデータを追加されるようにする）。
- TP が、実行時に、TP の実行に使用するプロセス ID に基づいてファイル名を生成するように設計する（つまり、TP の各コピーがそれぞれ異なるファイルに書き込まれるようにする）。

この行はオプションです。これを省略した場合、または無効な値を指定した場合は、デフォルトとして `QUEUED` が使用されます。

TIMEOUT

保留状態の着信割り振り要求がない場合に、TP が発行した `Receive_Allocate` 呼び出しを停止する最大時間（秒数）。この時間内に着信割り振り要求が受信されなかった場合、呼び出しは失敗し、「`State check - Allocate not pending`（状態チェック - 保留中の割り振り要求がない）」を示す戻りコードが戻されます。

タイムアウト値として `0` を指定した場合は、呼び出しを発行した時点で既に保留状態になっている着信割り振りがない限り、呼び出しは失敗します。`-1` のタイムアウト値は、呼び出しが着信割り振りを無期限に待機し、タイムアウトしないことを示します。

この行はオプションです。これを省略した場合、または無効な値（非数値）を指定した場合、デフォルトは `-1`（無期限）になります。

`TYPE` パラメーターを `NON-QUEUED` に設定した場合は、このパラメーターは指定しないでください。TP は常に着信割り振り要求への応答として開始されるものであり、したがって必ず `1` が保留状態になっているので、CS Linux は非待機 TP についてタイムアウト値 `0` を使用します。

UNIX

USERID

TP を開始するために CS Linux が使用するユーザー ID を指定します。TP は、このユーザー ID に関連したホーム・ディレクトリーの中で開始されます。このホーム・ディレクトリーは、トレース・ファイル、および TP がアクセスするその他のファイルのデフォルトのパスでもあります（アプリケーションで、絶対パスの指定により上書きされた場合を除きます）。アプリケーションで、パスなしのファイル名が指定された場合は、CS Linux はこのホーム・ディレクトリーの中でそのファイルを検索します。アプリケーションで、ファイル名と相対パスが指定されている場合は、CS Linux は、このホーム・ディレクトリーを基準として相対的に指定されたディレクトリーの中で、そのファイルを検索します。

この行は必須であり、指定する必要があります。この ID は、CS Linux コンピューター上の有効なログイン ID でなければなりません。AIX または Linux 構成によりユーザー名の文字数が制限されている場合を除き、この ID には最大 64 文字を使用できます。

ここで指定するユーザーには、この TP 用の実行ファイル（`PATH` パラメーターに指定したもの）の実行許可が与えられていることが必要です。さらに、`USERID` を `root` に設定する場合は、ファイルは `root` が所有しているものでなければならず、また、CS Linux がこのファイルを自動開始できるようにするには、`setuid` 許可および `setgid` 許可が設定されていることも必要です。

GROUP

TP を開始するために CS Linux が使用するグループ ID を指定します。この ID は、CS Linux コンピューター上の有効な AIX グループ ID でなければなりません。AIX または Linux 構成によりグループ名の文字数が制限されている場合を除き、この ID には最大 64 文字を使用できます。

この行はオプションで、これが含まれていない場合のデフォルトは `other` です。

LUALIAS

TP がどのローカル LU からの着呼接続を受け入れるかを LU 別名で指定します。

注: このパラメーターは、TP が APPC TP である場合にのみ使用できます。TP が CPI-C アプリケーションである場合は、このパラメーターを指定しないでください。CPI-C は、特定のローカル LU からの着呼 Attach 要求の受け入れをサポートしていません。CPI-C アプリケーションに LU の別名を (LU の別名がブランクの場合でも) 指定すると、着呼 Attach 要求を TP に経路指定する際のエラーの原因となります。

これは 8 文字の名前で、CS Linux のローカル APPC LU の名前に一致していなければなりません。

TP がどのローカル LU からの着呼接続でも受け付けるよう指定するには、このパラメーターを、ブランクの LU 別名を表す 2 個の二重引用符 ("") に設定します。呼び出し可能 TP データ・ファイルに、同じ TP 名について複数のエントリーが含まれている場合は、ブランクの LU 別名を指定できるのはそれらのエントリーのうちの 1 つのみで、その他のエントリーには、それぞれ異なる明示的な LU 別名を指定する必要があります。CS Linux は、可能であれば、この TP 名への着呼接続を、該当する LU 別名を指定する TP と突き合わせます。一致する LU 別名がない場合は、ブランク LU 別名を指定する TP と突き合わせます。

ファイルに非ブランクの LU 別名が指定されている場合、TP は APPC RECEIVE_ALLOCATE verb の拡張フォームを使用し、verb へのパラメーターとしてこの LU 別名を指定する必要があります。これにより、CS Linux は、着信接続を正しい TP に送ることができるようになります。さまざまな形式の RECEIVE_ALLOCATE について詳しくは、「*IBM Communications Server for Data Center Deployment on AIX or Linux APPC プログラマーズ・ガイド*」を参照してください。アプリケーションに LU 別名を事前に組み込まないで、TP が実行時に正しい LU 別名を判別できるようにする場合は、該当の LU 別名が入る環境変数を設定し (ENV パラメーターを使用)、RECEIVE_ALLOCATE の発行方法を決定するための環境変数をアプリケーションが読み込むように設計します。

この行はオプションで、これが含まれていない場合のデフォルトでは、どのローカル LU からの着信接続も受け入れ、TP はどちらの形式の APPC RECEIVE_ALLOCATE verb も使用できます。

ENV

TP が必要とする環境変数を指定します。各変数は、それぞれ別個の ENV 行に、`environment_variable=value` の形式で指定します。最大 64 個の ENV 行を含めることができます。変数は、ここで入力したのと同じ順序で設定されます。

`environment_variable=value` のストリングの中で、= 文字の前後にスペースまたはタブ文字を使用しないでください。

WINDOWS**SHOW**

このパラメーターは、アプリケーションが GUI アプリケーションの場合にのみ設定できます。アプリケーションがコンソール・アプリケーションであれば、このパラメーターは無視されます。アプリケーションの開始時の表示方法を指定します。このパラメーターはアプリケーションに渡され、CS Linux によっては処理されません。このパラメーターの解釈と処理は、アプリケーションが引き受けます。次のいずれかの値を入力できます。

MAXIMIZED

アプリケーションを最大化します。

MINIMIZED

アプリケーションを最小化します。

HIDDEN

アプリケーションを画面に表示しません。

NORMAL

アプリケーションを通常のサイズと位置で表示します。

NOACTIVATE

アプリケーションを通常のサイズと位置で表示し、フォーカスは直前のアクティブ・ウィンドウに残します。アプリケーションのウィンドウは、アクティブ・ウィンドウになりません。

MINNOACTIVATE

アプリケーションを最小化し、フォーカスは直前のアクティブ・ウィンドウに残します。

このパラメーターはオプションです。これが含まれていない場合のデフォルトは NORMAL です。

SECURITY_TYPE

TP 実行可能ファイルのセキュリティー・タイプを指定します。

APPLICATION

TP 実行可能ファイルは、CreateProcess システム呼び出しを使用してアプリケーションとして開始されます。

SERVICE

TP 実行可能ファイルは、StartService システム呼び出しを使用してサービスとして開始されます。この場合は、SERVICE_NAME パラメーターに指定された名前を使用して、サービス・コントロール・マネージャーにサービスが事前にインストールされている必要があります。

この値は、Windows サービスとして実行される TP を参照します (16 進数で指定された 4 文字の名前をもつ SNA サービス TP でなく)。Windows 上では、特定のサービスを一度に 1 つだけ実行できるので、TYPE パラメーターを NON-QUEUED に設定することはできません。この値を指定した場合は、値 QUEUED-BROADCAST が代わりに使用されます。

SERVICE_NAME

サービス・コントロール・マネージャーにインストール済みのサービスの名前。このパラメーターは、SECURITY_TYPE が SERVICE である場合のみ使用されます。

USERID

SECURITY_TYPE が APPLICATION である場合に、クライアントが TP の開始に使用するドメインとユーザー ID を指定します。このパラメーターの形式は、Windows Client コンピューターがドメインの一部である場合は、*domain_name*¥*user_ID* で、Windows Client コンピューターがドメインの一部でない場合は、*computer_name*¥*user_ID* (ドメイン・ネームではなく、Windows Client 専用のコンピューター名を示す) です。

クライアントは、指定されたユーザーのログオン・セッション内で TP の開始を試行します。USERID がブランクまたは指定されていない場合、TP はコンソール・セッション内で開始されます。指定されたユーザーがログオンしていない場合、またはコンソールでログオンしているユーザーがない場合は、TP は開始されず、CS Linux サーバーに障害が通知されます。



上記のエントリーの形式については、次の点に注意してください。

- 行の先頭文字として # を含めると、その行をコメント行にすることができます。CS Linux はこの行を無視します。CS Linux は完全なブランク行も無視します。
- どの *parameter=value* エントリーも、それぞれ単独の行に入力する必要があります。途中に行区切り文字を含めることはできません。1 行の最大長は 255 文字で、それを超えた文字は無視されます。
- 行の先頭または末尾、または = 文字の前またはあとにあるホワイト・スペース (スペース文字またはタブ文字) は無視されます (ただし、ENV パラメーターの *environment_variable=value* スtring の場合を除きます)。
- 各 TP 定義は、TP 名を識別する行で始まり、ファイルの終わりまたは次の TP 名で終わります。
- ENV 行 (64 回指定できる) を除き、同じ TP について同じパラメーターを 2 回以上指定しないでください。同じパラメーターを複数回指定した場合は、各キーワードの最後のインスタンスのみが使用されます。

付録 C DDDLU 用の TN3270 LU モデルの構成

TN3270 クライアントの場合 DDDLU を使用して CS Linux TN サーバーを介してホストに接続する場合、CS Linux は、クライアントが必要とする LU モデルに関する情報をホストに送信する必要があります。通常、これは、クライアントによって指定された端末タイプ (装置タイプ) からの標準マッピングを使用して、LU モデルを判別します。

TN3270 デバイス・タイプと LU モデル間のマッピングを変更する必要がある場合は、`tn3270dev.dat` (装置) ファイルを使用してこれを行うことができます。このファイルのサンプル・バージョンは、`/opt/ibm/sna/samples` に提供されています。このファイルを `/etc/opt/ibm/sna` にコピーし、`vi` のような標準の ASCII テキスト・エディターを使用して変更を行います。この変更は、次回 CS Linux を再始動するとき有効になります。

ファイル内の各行は、このデバイスのホストに送信する必要がある、TN3270 デバイスと LU モデル・ストリングとの間のマッピングを表します。スペースで区切られた、以下の項目から構成されます。

- 最初の項目は、このマッピングが TN3270E クライアント (TN3270 拡張機能を使用) の場合は単一文字 Y、または標準の TN3270 クライアント用の場合は N です。
- 2 番目の項目は、クライアントによって指定される端末タイプ (装置タイプ) です。これは、最大 40 文字のテキスト・ストリングです。有効な文字は、A から Z の文字、0 から 9、/、および - の文字です。
- 3 番目の項目は、CS Linux がホストに送信して、このクライアントの正しい LU モデルを識別するために送信する、7 文字の LU モデル・ストリングです。

CS Linux で提供されるサンプル・ファイルには、TN3270 バージョンと TN3270E バージョンの両方で 18 個の標準デバイス・タイプのマッピングが含まれています (すべてのバージョンの 36 項目)

- これらの標準装置の 1 つまたは複数に別の LU モデル・ストリングを指定する必要がある場合には、ファイルの該当する行の 3 番目の項目を変更して、必要な 7 文字のストリングを指定してください。
- サンプル・ファイルに含まれていない追加のデバイス・タイプをサポートする必要がある場合は、上記のフォーマットを使用して、デバイス・タイプごとにファイルに新しい行を追加します。

付録 D IBM へのご意見の送付方法

本資料に関するご意見をお待ちしております。情報の明確性、正確性、および完全性に関するご意見およびその他の情報をご自由にお寄せください。

以下のいずれかの方法でご意見をお送りください。

1. Knowledge Center の下部にあるフィードバック・リンクを使用してください。
2. 以下のフィードバック・テンプレートを 사용하여 "mhvrcfs@us.ibm.com" に E メールを送信してください。
3. 次の住所にご意見をお送りください。

IBM Corporation
Attention: MHVRCFS Reader's Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
US

E メール・フィードバック・テンプレート

以下のテンプレートをメールにカット・アンド・ペーストしてください。その後、必要な情報を入力します。

- お名前:
- 企業、大学、機関名:
- ご意見の対象となるトピックまたは Web ページの URL:
- ご意見の本文

ご意見を口頭でいただける場合は、お客様の電話番号とご都合の良い時間も記入していただけます。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、適切な方法で、使用もしくは配布することができるものとします。

IBM または他の組織は、お客様からいただいた個人情報、問題解決のためのお客様への連絡以外の目的で使用することはありません。

技術的な問題がある場合

リストされたご意見の送付方法は使用しないでください。代わりに、以下のいずれかのアクションを実行してください。

- IBM サービス担当員にお問い合わせください。
- IBM 技術サポートに連絡してください。
- IBM サポート・ポータル (<https://www.ibm.com/support/home/>) にアクセスしてください。

付録 E 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

- 〒 106-8711
- 東京都港区六本木 3-2-12
- 日本アイ・ビー・エム株式会社
- 法務・知的財産
- 知的財産権ライセンス 渉外

- 〒 106-8711
- 東京都港区六本木 3-2-12
- 日本アイ・ビー・エム株式会社
- 法務・知的財産
- 知的財産権ライセンス 渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

- IBM Corporation
- P.O. Box 12195
- 3039 Cornwallis Road
- Research Triangle Park, NC 27709-2195
- U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾: 本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物にも、次のように、著作権表示を入れていただく必要があります。© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。© Copyright IBM Corp. 2000, 2005, 2006, 2007, 2008, 2021. All rights reserved.

商標

以下は、世界の多くの国で登録された International Business Machines Corporation の商標です。

- ACF/VTAM
- Advanced Peer-to-Peer Networking
- AIX
- Application System/400
- APPN
- AS/400
- CICS
- DATABASE 2
- DB2
- Enterprise System/3090
- Enterprise System/4381
- Enterprise System/9000
- ES/3090
- ES/9000
- eServer
- IBM
- IBMLink
- IMS
- MVS
- MVS/ESA
- Operating System/2
- Operating System/400
- OS/2
- OS/400
- PowerPC
- PowerPC Architecture
- S/390
- System/390
- System p5
- System z
- System z9
- VSE/ESA
- VTAM
- WebSphere

以下は、各々の会社の商標または登録商標です。

Java およびすべての Java 関連の商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Intel および EM64T は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

AMD64 は Advanced Micro Devices, Inc の米国およびその他の国における商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft、Windows、Windows 10、Windows Server 2012、Windows Server 2016、Windows Server 2019 および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

参考文献

以下の IBM 資料では、本書で説明しているトピックについての情報を記載しています。資料は、以下のトピック別に大きく分けてあります。

- CS Linux バージョン 7.1
- システム・ネットワーク体系 (SNA)
- ホスト構成
- z/OS Communications Server
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- X.25
- 拡張プログラム間通信機能 (APPC)
- プログラミング
- その他の IBM ネットワーキング・トピック

CS Linux ライブラリーの資料については、その要旨が説明されています。その他の資料については、タイトルおよび資料番号のみをここに記しています。

CS Linux バージョン 7.1 の資料

CS Linux ライブラリーは、以下の資料により構成されています。なお、これらの資料のソフトコピー版が CD-ROM で提供されています。CD-ROM のソフトコピー・ファイルへのアクセス方法については、「*IBM Communications Server for Data Center Deployment on Linux 入門*」を参照してください。これらのソフトコピー・ブックをシステムにインストールするには、9～15MB のハード・ディスク・スペースが必要になります (このスペースは、インストールする各国語バージョンによって異なります)。

- *IBM Communications Server for Data Center Deployment on Linux 入門* (GC88-9996 および GC88-9997)
この資料は CS Linux の概要を示すもので、サポートされているネットワークの特性、インストール、構成、および操作について説明しています。この資料には、以下の 2 つのバージョンがあります。
 - IBM CS Linux 入門 (GC88-9996)
 - IBM CS Linux for IBM Z 入門 (GC88-9997)
- *IBM Communications Server for Data Center Deployment on Linux 管理ガイド* (SC88-9999)
この資料には、SNA および CS Linux の概要、および CS Linux の構成および操作に関する情報が記載されています。
- *IBM Communications Server for Data Center Deployment on Linux 管理コマンド解説書* (SC88-9998)
この資料には、SNA および CS Linux のコマンドに関する情報が記載されています。
- *IBM Communications Server for Data Center Deployment on AIX or Linux CPI-C プログラマーズ・ガイド* (SC88-5826)
この資料では、「C」または Java™ の経験があるプログラマーを対象として、CS Linux CPI 通信 API を使用する SNA トランザクション・プログラムの作成に関する情報を提供します。
- *IBM Communications Server for Data Center Deployment on AIX or Linux APPC プログラマーズ・ガイド* (SC88-5825)
この資料では、拡張プログラム間通信機能 (APPC) を使用する アプリケーション・プログラムを作成するために必要な情報を記載しています。
- *IBM Communications Server for Data Center Deployment on AIX or Linux LUA プログラマーズ・ガイド* (SC88-5827)
この資料では、従来型 LU アプリケーション・プログラミング・インターフェース (LUA) を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。

- *IBM Communications Server for Data Center Deployment on AIX or Linux 共通サービス Verb プログラマーズ・ガイド* (SC88-5824)
この資料では、Common Service Verb (CSV) アプリケーション・プログラミング・インターフェース (API) を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。
- *IBM Communications Server for Data Center Deployment on AIX or Linux MS プログラマーズ・ガイド* (SC88-5829)
この資料では、Management Services (MS) API を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。
- *IBM Communications Server for Data Center Deployment on Linux NOF プログラマーズ・ガイド* (SC88-8591)
この資料では、Node Operator Facility (NOF) API を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。
- *IBM Communications Server for Data Center Deployment on Linux 診断ガイド* (GC88-8601)
この資料では、SNA ネットワークの問題解決について説明しています。
- *IBM Communications Server for Data Center Deployment on AIX or Linux APPC アプリケーション・スイート ユーザーズ・ガイド* (SC88-5828)
この資料には、CS Linux で使用される APPC アプリケーションに関する情報が記載されています。
- *IBM Communications Server for Linux 用語集* (GC88-8602)
この資料は、CS Linux ライブラリーで頻繁に使用される用語とその定義を包括的に収録しています。

システム・ネットワーク体系 (SNA) 関連資料

以下の資料では、SNA ネットワークについての情報を記載しています。

- *Systems Network Architecture; Format and Protocol Reference Manual - Architecture Logic for LU Type 6.2* (SC30-3269)
- *Systems Network Architecture; Formats* (GA27-3136)
- *Systems Network Architecture; Guide to SNA Publications* (GC30-3438)
- *Systems Network Architecture; Network Product Formats* (LY43-0081)
- *Systems Network Architecture; Technical Overview* (GC30-3073)
- *Systems Network Architecture; APPN Architecture Reference* (SC30-3422)
- *Systems Network Architecture; Sessions between Logical Units* (GC20-1868)
- *Systems Network Architecture; LU 6.2 Reference - Peer Protocols* (SC31-6808)
- *Systems Network Architecture; Transaction Programmer's Reference Manual for LU Type 6.2* (GC30-3084)
- *Systems Network Architecture; 3270 Datastream Programmer's Reference* (GA23-0059)
- *Networking Blueprint Executive Overview* (GC31-7057)
- *Systems Network Architecture; Management Services Reference* (SC30-3346)

ホスト構成関連資料

以下の資料では、ホスト構成についての情報を記載しています。

- *ES-9000, ES-3090 IOCP ユーザーズ・ガイド・ボリューム A04* (GC38-0097)
- *3174 Establishment Controller インストール・ガイド* (GG24-3061)
- *3270 Information Display System 3174 Establishment Controller; 計画ガイド* (GA27-3918)
- *OS/390 ハードウェア構成定義 (HCD) ユーザーズ・ガイド* (SC88-6630)

z/OS Communications Server 関連資料

以下の資料には、z/OS Communications Server に関する情報について記載されています。

- *z/OS V1R7 Communications Server: SNA ネットワーク・インプリメンテーション・ガイド* (SC88-8928)
- *z/OS V1R7 Communications Server: SNA 診断* (Vol 1: GC31-6850、Vol 2: GC31-6851)
- *z/OS V1R6 Communications Server: リソース定義解説書* (SC88-8929)

TCP/IP 関連資料

以下の資料には、Transmission Control Protocol/Internet Protocol (TCP/IP) ネットワーク・プロトコルに関する情報について記載されています。

- *z/OS V1R7 Communications Server: IP 構成ガイド* (SC88-8926)
- *z/OS V1R7 Communications Server: IP 構成解説書* (SC88-8927)
- *z/VM V5R1 TCP/IP 計画およびカスタマイズ* (SD88-6453)

X.25 関連資料

以下の資料には、X.25 ネットワーク・プロトコルに関する情報について記載されています。

- *Communications Server for OS/2 Version 4 X.25 Programming* (SC31-8150)

APPC 関連資料

以下の資料では、拡張プログラム間通信機能 (APPC) についての情報を記載しています。

- *APPC Application Suite V1 ユーザーズ・ガイド* (SC31-6532)
- *APPC Application Suite V1 管理* (SC31-6533)
- *APPC Application Suite V1 プログラミング* (SC31-6534)
- *APPC Application Suite V1 オンライン製品ライブラリー* (SK2T-2680)
- *APPC Application Suite ライセンス・プログラムの資料* (GC31-6535)
- *z/OS V1R2.0 Communications Server: APPC Application Suite User's Guide* (SC31-8809)

プログラミング関連資料

次の資料では、プログラミングについての情報を記載しています。

- *共通プログラミング・インターフェース コミュニケーション (CPI-C) 解説書* (SC26-4399)
- *Communications Server for OS/2 Version 4 Application Programming Guide* (SC31-8152)

その他の IBM ネットワーキング関連資料

次の資料では、CS Linux に関連する その他のトピックについての情報を記載しています。

- *SDLC Concepts* (GA27-3093)
- *Local Area Network Concepts and Products; LAN アーキテクチャー* (SG24-4753)
- *Local Area Network Concepts and Products; LAN アダプター、ハブおよび ATM* (SG24-4754)
- *Local Area Network Concepts and Products; ルーターおよびゲートウェイ* (SG24-4755)
- *Local Area Network Concepts and Products; LAN オペレーティング・システムおよび管理* (SG24-4756)
- *IBM ネットワーク制御プログラム リソース定義ガイド* (SC30-3349)

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。
なお, 濁音と半濁音は清音と同等に扱われています。

[ア行]

アタン
記述 [1](#)
アプリケーション
プログラム [4](#)
ワークシート [139](#)
アプリケーション・プログラミング・インターフェース (API) [5](#)
イーサネット
ポート構成 [52](#)
ワークシート [146](#)
ウィンドウ
説明 [37](#)
ツールバー・ボタン [42](#)
ドメイン [37](#)
ノード [37](#), [39](#)
メニュー [37](#)
リソース [37](#)
リソース項目 [41](#)
CPI-C Destination Names (CPI-C 宛先名) [38](#)
LU Pools (LU プール) [38](#)
ウンチ
許可 [106](#)
コマンド構文 [104](#)
コマンドの取り消し [108](#)
出力 [108](#)
使用 [106](#)
担保 [106](#)
デーモン・プログラム [106](#)
利用者 [106](#)
エスケープ文字、RCF [105](#)
エラー・ログ・ファイル [32](#)
エンタープライズ・エクステンダー (HPR/IP)
ポート構成 [52](#)
エンド・ノード
サンプル APPN ネットワーク内の [12](#)
説明 [4](#)
ディレクトリー [15](#), [16](#)
APPN [13](#)

[カ行]

カーネル・コンポーネント、トレース [35](#)
カーネルコンポーネントのトレース [35](#)
カーネル・メモリー制限 [35](#)
開始コマンド [35](#)
会話
説明 [9](#)
担保 [87](#)
会話セキュリティ
構成メソッド [87](#)
パラメーター [87](#)
拡張対等通信ネットワーク機能 (APPN) [1](#)

仮想記憶通信アクセス方式 (VTAM) [11](#)
仮想ルーティング・ノード (VRN) [22](#)
環境
データ・ファイル [31](#)
監査ログ・ファイル [32](#)
完全修飾 LU 名 [14](#)
管理
責任 [27](#)
ツール [28](#)
管理サービス (MS) [13](#)
管理サービス (MS) API [5](#)
技術的な問題
解決方法 [171](#)
基本会話 [10](#)
境界ノード [2](#)
共用アクセス転送機能 (shared-access transport facility: SATF) [21](#)
クライアント
管理 [111](#)
ネットワークングの要件 [113](#)
ネットワークデータファイル [31](#)
呼び出し可能 TP 構成 [163](#)
ARGUMENTS パラメーター [165](#)
PATH パラメーター [165](#)
SECURITY_TYPE パラメーター [168](#)
SERVICE_NAME パラメーター [168](#)
SHOW パラメーター [167](#)
TIMEOUT パラメーター [166](#)
TP の定義 [138](#)
TPname パラメーター [165](#)
TYPE パラメーター [165](#)
クライアント / サーバー
トレース [133](#)
クライアント / サーバー
構成 [47](#)
トレース [126](#)
クライアント構成ユーティリティー、Windows [118](#)
クラスター・コントローラー [3](#)
クルック
ポート構成 [52](#)
計画ワークシート [33](#)
経路 (route) [10](#)
経路選択 [10](#), [17](#), [20](#)
検出、リソースの [14](#)
構成
作業 [47](#)
従属 LU [65](#)
セキュリティアクセスリスト [87](#)
接続 [51](#)
接続ネットワーク [52](#)
ドゥルツ [52](#)
ノード [48](#)
パススルー・サービス [91](#)
ファイル [30](#)
ポート [52](#)
APPC セキュリティー [86](#)
APPC 通信 [69](#)

構成 (続き)

- CPI-C サイド情報 [84](#)
- DLUR [99](#)
- SNA ゲートウェイ [98](#)
- TN サーバー・アクセス・レコード [92](#)
- TN サーバー関連付けレコード [94](#)
- TN リダイレクター・レコード [95](#)
- TP [75](#)

構成サーバー

- 除去 [47](#)
- 追加 [47](#)

高性能ルーティング (HPR) [11, 17](#)

高速トランスポート・プロトコル (RTP) [11, 17](#)

コース

- タイプ [80](#)
- 目的 [80](#)

顧客情報管理システム (CICS) [6](#)

コマンド

- 構成サーバーの変更 [47](#)

コマンド行管理プログラム

- 記述 [30](#)
- クライアントから [46](#)
- コマンド・タイプ [46](#)
- 使用 [46](#)
- ヘルプ [46](#)

混合ネットワーク [2, 24](#)

コントローラー・サーバー の指定 [112](#)

[サ行]

サーバー

- クライアントとの関係 [111](#)
- 使用可能化 [35](#)
- 使用不可に設定する [36](#)
- 除去 [47](#)
- 追加 [47](#)

サービス・アクセス・ポイント (SAP) [23](#)

サービス・クラス (COS) [11](#)

サービス・ポイント [103](#)

サービス・ポイント・コマンド機能 (SPCF) [30, 103](#)

作業シート [34](#)

サブエリア SNA [1](#)

サブエリア・ネットワーク

- 経路選択 [10](#)
- 説明 [2](#)
- ノード・タイプ [2](#)
- 例 [3](#)

サブエリア・ノード [2](#)

システム・サービス制御点 (SSCP) [7](#)

自動ネットワーク・ルーティング (automatic network routing: ANR) [11, 17](#)

従属 LU

- 構成 [65](#)
- 説明 [8](#)

従属 LU サーバー (DLUS) [25](#)

従属ノード [2](#)

周辺ノード [2](#)

シュブクフ

- コマンド構文 [104](#)

シュルツ

- ポート構成 [52](#)

使用可能化、CS Linux ソフトウェアの

- Windows 上の Remote API Client [116](#)

状況

Windows 上のリモート API クライアント [117](#)

詳細ログ [32](#)

使用不可化、CS Linux ソフトウェアの

Windows 上の Remote API Client [117](#)

使用不可に設定する、ソフトウェアを [36](#)

情報管理システム / 仮想記憶 (Information Management System/Virtual Storage) [6](#)

資料に関するコメント

- フィードバックの送信 [171](#)

診断ツール [32](#)

スナ

- 記述 [1](#)
- サブエリア [1](#)
- ネットワーク [1](#)
- ネットワークデータファイル [31, 133](#)

制御データ [8](#)

制御点 (CP) [7](#)

セキュリティアクセスリスト

- 構成メソッド [87](#)

- 目的 [87](#)

セキュリティ

- APPC [86](#)

- UCF [109](#)

セキュリティ・アクセス・リスト

- パラメーター [88](#)
- 必要な追加構成 [88](#)

セッション

- 説明 [7](#)
- タイプ [7](#)
- ルーティング [17](#)
- セッション・セキュリティ
- 構成メソッド [86](#)
- パラメーター [86](#)
- 必要な追加構成 [87](#)

接続

- 構成 [51](#)
- 説明 [4](#)
- 直接 [21](#)
- ワークシート [139, 142](#)

接続ネットワーク

- 構成 [52](#)
- 構成メソッド [52](#)
- 説明 [11](#)
- トポロジー情報 [18](#)
- 必要な追加構成 [56](#)
- APPN [22](#)
- IPv6 Address Only [55](#)

接続ネットワーク・パラメーターで定義 [55](#)

ソース TP [9, 75](#)

[タ行]

ターゲット TP [9, 76](#)

ダイアログ [37](#)

対等通信 [1](#)

対等ネットワーク

- 経路選択 [11](#)
- ノード・タイプ [4](#)

タイプ 2.0 ノード [2](#)

タイプ 2.1 ノード (type 2.1 node) [2](#)

タイプ 4 ノード [2](#)

タイプ 5 ノード [2](#)

ダウンストリーム LU、SNA ゲートウェイの

- ダウンストリーム LU、SNA ゲートウェイの (続き)
 - 構成メソッド [99](#)
- ダウンストリーム LU、SNA ゲートウェイ用の
 - パラメーター [99](#)
 - 必要な追加構成 [99](#)
- 端末コントローラー [3](#)
- 端末装置 [3](#)
- 担保
 - ウンチ [106](#)
 - 会話 [87](#)
 - セッション [86](#)
- 中央ロギング [32](#)
- 中間セッション・ルーティング (ISR) [17](#), [20](#)
- 中間ルーティング [20](#)
- 直接接続 [21](#)
- 通信コントローラー [3](#)
- 通信コントローラー・ノード [2](#)
- 通信リンク [3](#)
- ディレクトリー
 - エンド・ノード [15](#), [16](#)
 - ネットワーク・ノード [15](#), [16](#)
 - CS Linux 実行可能プログラム用 [34](#)
 - LEN ノード [15](#)
- データ・ファイル
 - 環境 [31](#)
 - クライアント・ネットワーク [31](#)
 - ドメイン構成 [31](#)
 - ノード構成 [30](#)
 - 呼び出し可能 TP [31](#)
 - ログ・フィルター [31](#)
 - SNA ネットワーク [31](#)
 - TP 定義 [31](#)
- データ・リンク制御 (DLC) [51](#)
- 伝送グループ [18](#)
- ドゥルツ
 - 構成メソッド [52](#)
- ドゥルル
 - TN3270 の LU モデル [169](#)
- トークンリング
 - ポート構成 [52](#)
 - ワークシート [144](#)
- 読者のコメント
 - フィードバックの送信方法 [171](#)
- 独立 LU
 - 構成 [69](#)
 - 説明 [8](#)
- トポロジー/ルーティング・サービス (TRS) [18](#)
- トポロジー情報
 - 接続ネットワーク [18](#)
 - ローカル [18](#)
- トポロジー・データベース更新 (TDU) [19](#)
- ドメイン
 - 構成ファイル [31](#)
 - 説明 [2](#)
- ドメイン・ネーム
 - 変更 [112](#)
- ドメイン・リソース [47](#)
- トラブルシューティング [32](#)
- トランザクション・プログラム (TP) [4](#)
- トランスポート・ネットワーク [11](#)
- ドループ
 - 構成メソッド [62](#)
- トレース
 - クライアント / サーバー [133](#)

- トレース (続き)
 - ラン [133](#)
- トレース・ファイル [32](#)

[ナ行]

- 内部トレース
 - Windows 上の Remote API Client [128](#)
- ネットワーク
 - 管理 [103](#)
 - 混合 [24](#)
 - タイプ [2](#)
 - トポロジー・データベース [18](#)
- ネットワーク・アクセス可能単位 (NAU) [5](#)
- ネットワーク・アクセス・プロセス (NAP) [116](#)
- ネットワーク・アドレス可能単位 [5](#)
- ネットワーク管理データ [8](#)
- ネットワークデータファイル
 - 記述 [31](#)
 - AIX または Linux 上のリモート API クライアント [133](#)
- ネットワーク・ノード
 - サンプル構成 [12](#)
 - ディレクトリー [15](#), [16](#)
- ネットワーク・ノード・サーバー [4](#), [12](#)
- ノード
 - 構成ファイル [30](#)
 - 構成メソッド [48](#)
 - サブエリア [2](#)
 - 周辺 [2](#)
 - 対等 [2](#)
 - タイプ [2](#), [4](#)
 - パラメーター [48](#), [49](#)
 - 必要な追加構成 [49](#)
 - 目的 [48](#)
 - ワークシート [139](#)
 - SNA [2](#)
 - ノード・ウィンドウ [39](#)
 - ノード・オペレーター機能 (NOF) API [5](#)
 - ノード・リソース [47](#)

[ハ行]

- バージョン、IP アドレス [113](#)
- バージョン番号、NetView [103](#)
- パートナー LU
 - 構成メソッド [73](#)
 - パラメーター [73](#), [74](#)
 - 必要な追加構成 [75](#)
 - 複数、ワイルドカードによる定義 [73](#)
 - 別名、定義 [73](#)
 - リモート・ノード、定義 [73](#)
- パススルー DLUR [63](#)
- パススルー・サービス
 - 構成 [91](#)
 - ワークシート [139](#), [151](#)
- バックアップ・サーバー [47](#), [111](#)
- 汎用データ・ストリーム (GDS) [6](#)
- フィードバック
 - 読者のコメントの送信 [171](#)
 - E メール・テンプレート [171](#)
- プー
 - DLUR の場合 [62](#)
- フォーマット [1](#)

複数セッション [8](#)
フット [115](#)
物理装置 (PU) [5](#)
物理装置制御点 (PUCP) [7](#)
プリンター [3](#)
ブロードキャスト検索 [17](#)
プロトコル [1](#)
プロトコル・パラメーター [55](#)
フロントエンド・プロセッサ (front-end processor: FEP) [3](#)
分岐エクステンダー [23](#)
分岐ネットワーク・ノード [4, 23](#)
文書コンテンツ・アーキテクチャー (DCA) [6](#)
並列セッション [8](#)
別名、パートナー LU [73](#)
ヘルプ
 コマンド行管理プログラム [46](#)
 Motif 管理プログラム [45](#)
ポート
 構成 [52](#)
 パラメーター [52-55](#)
 必要な追加構成 [56](#)
ホスト [3](#)
ホスト・ノード [2](#)

[マ行]

マップ式会話 [10](#)
マルチパス・チャンネル
 ワークシート [149](#)
マルチパス・チャンネル (MPC)
 ポート構成 [52](#)
メディア・アクセス制御 (MAC) [23](#)
メプツ
 ポート構成 [52](#)
モード
 構成 [81, 82](#)
 説明 [10](#)
 パラメーター [82-84](#)
 必要な追加構成 [84](#)
 標準 [80](#)
文字、RCF コマンドの [105](#)
問題判別援助機能
 ログイン [49](#)
問題判別補助プログラム
 概要 [32](#)

[ヤ行]

有向検索 [17](#)
ユーザー・アプリケーション・サポート・ワークシート [155](#)
要求単位 (RU) [83](#)
要約ログ [32](#)
呼び出し可能 TP
 データ・ファイル [31](#)
 CS Linux への定義 [76](#)
 snatpinstall の使用 [163](#)
呼び出し側 TP [9](#)

[ラ行]

リソースの検出 [14](#)
リソース名 [14](#)
リモート・コマンド機能 (RCF) [30](#)

リモート・ジョブ入力 (RJE) [6](#)
リモート・ノード
 構成メソッド [72](#)
 定義 [71](#)
 パートナー LU [73](#)
 必要な追加構成 [72](#)
 LU [7](#)
 Node's SNA network name パラメーター [72](#)
リンク・ステーション
 構成 [56](#)
 説明 [4](#)
 パラメーター [57-61](#)
 必要な追加構成 [61](#)
リンク・ステーション経路指定
 パラメーター [74, 75](#)
隣接ノード [11](#)
ルア
 プール構成 [67](#)
ローエントリー・ネットワーク (LEN) ノード [4](#)
ローカル IP インターフェース・パラメーター [55](#)
ローカル LU
 構成メソッド [70](#)
 説明 [7](#)
 定義 [70](#)
 パラメーター [70, 71](#)
 必要な追加構成 [71](#)
ローカル・トポロジー・データベース [18](#)
ローカル・ノード
 LU [7](#)
ロギング
 Windows 上のリモート API クライアント [122](#)
ログ・ファイル
 構成 [49](#)
 タイプ [49](#)
ログ・フィルター
 データ・ファイル [31](#)
ログ・メッセージ [32](#)
論理装置 (LU) [6](#)
論理レコード [10](#)

[ワ行]

ワークシート [33](#)
ワイルドカード [73](#)

[数字]

1 次 LU [8](#)
2 次 LU [8](#)
3270
 プール構成 [67](#)
 ワークシート [160](#)
 LU 構成 [65](#)
5250
 ワークシート [159](#)

A

Activation パラメーター [57](#)
Adapter card number パラメーター [53](#)
AIX または Linux 上の Remote API Client
 管理 [132](#)
AIX または Linux 上のリモート API クライアント

AIX または Linux 上のリモート API クライアント (続き)

- サーバー・キャッシュ・タイムアウト [135](#)
- サーバー名 [135](#)
- ブロードキャスト試行回数 (t_count) [134](#)
- 呼び出し側 tps [134](#)
- l_access_timeout [134](#)
- LU 別名、LUA LU 名、または TP 名のオーバーライド [136](#)
- poll_ タイマー [134](#)

Alias パラメーター [73](#)

Allow access to specific LU パラメーター [93](#)

Allow timeout パラメーター [99](#)

ANR

- 説明 [11](#), [17](#)
- 動的再ルーティング [21](#)

API

- 説明 [5](#)
- プロプラエタリー [5](#)
- CS Linux に組み込まれた [5](#)

API トレース

Windows 上の Remote API Client [125](#)

APPC

- 構成 [69](#)
- セキュリティー [86](#)
- ワークシート [155](#)

APPCLLU

Windows 上の Remote API Client [129](#)

APPCTPN

Windows 上の Remote API Client [129](#)

Application System/400 (AS/400) [11](#)

APPN

- エンド・ノード [4](#), [13](#), [140](#)
- 機能 [11](#)
- 経路選択 [20](#)
- 制御点 (Control Point) [13](#)
- 接続ネットワーク [22](#)
- 説明 [11](#)
- ネットワーク [11](#), [21](#)
- ネットワーク・ノード [4](#), [12](#), [139](#)
- ネットワークの例 [12](#)
- ノード・タイプ [11](#)
- 分岐ネットワーク・ノード [4](#), [140](#)

APPN support パラメーター [48](#)

Arguments パラメーター [78](#)

AS/400 (Application System/400) [11](#)

Assigned LUs パラメーター [68](#)

Auto-activated sessions パラメーター [83](#)

B

Backup DLUS Name パラメーター [62](#)

BIND 要求 [8](#)

Branch link type パラメーター [60](#)

C

CICS (顧客情報管理システム) [6](#)

CN (接続ネットワーク) [11](#)

CN name パラメーター [54](#)

CN 名パラメーター [55](#)

compression supported パラメーター [63](#), [83](#)

Configure downstream LUs for implicit PU access パラメーター [55](#)

Control point alias パラメーター [48](#)

Control point name パラメーター [48](#)

Conversation level security required パラメーター [80](#)

Conversation type パラメーター [80](#)

COS

説明 [11](#)

COS name パラメーター [82](#)

CP (制御点) [7](#)

CP の完全修飾名 [14](#)

CP-CP セッション [8](#)

CPI-C (通信用共通プログラミング・インターフェース) サイド情報 [84](#)

CPI-C (共通プログラミング・インターフェース・コミュニケーション)

ワークシート [159](#)

CPI-C サイド情報

構成メソッド [84](#)

パラメーター [84-86](#)

必要な追加構成 [86](#)

CS Linux 実行可能プログラムのパス [34](#)

CS Linux ソフトウェアの使用可能化

サーバー上の [35](#)

初期設定時の [36](#)

CSVTBLG

Windows 上の Remote API Client [129](#)

D

DCA (文書コンテンツ・アーキテクチャー) [6](#)

Define on connection network パラメーター [54](#)

Delayed logon パラメーター [99](#)

Destination host address パラメーター [97](#)

Device name [55](#)

Display LU assigned パラメーター [93](#)

Display LU パラメーター [94](#)

DLC

構成 [51](#), [52](#)

必要な追加構成 [56](#)

DLUR

構成 [99](#)

説明 [25](#)

必要な追加構成 [63](#)

ワークシート [151](#)

DLUR ダウンストリーム・ノード [63](#)

DLUS

説明 [25](#)

DLUS Name パラメーター [62](#), [63](#)

Domain window [38](#)

Downstream LU name パラメーター [99](#)

Downstream PU Name パラメーター [61](#), [63](#)

E

EN (エンド・ノード) [4](#)

Enterprise Extender (HPR/IP)

ワークシート [150](#)

ENV パラメーター [167](#)

Environment パラメーター [79](#)

Ethernet type パラメーター [54](#)

F

FEP (フロントエンド・プロセッサ) [3](#)

Full path to TP executable パラメーター [78](#)

G

GDS (汎用データ・ストリーム) [6](#)

Group ID パラメーター [79](#)

GROUP パラメーター [166](#)

H

Host LS/DLUR PU パラメーター [65](#), [70](#)

HPR

説明 [11](#), [17](#)

HPR supported on implicit links パラメーター [55](#)

I

IMS/VS (情報管理システム / 仮想記憶) [6](#)

Initial session limit パラメーター [82](#)

Initial window size パラメーター [83](#)

Initially active パラメーター [53](#), [63](#)

IP アドレス・フォーマット [113](#)

IP ポート番号 [114](#)

IPv4 アドレス [113](#)

IPv6 Address Only [55](#)

IPv6 アドレス [113](#)

ISR [17](#), [20](#)

L

LAN アクセス・タイムアウト [114](#)

LAN トレース

クライアント上 [133](#)

LEN ノード

機能 [12](#)

説明 [4](#), [13](#)

ディレクトリー [15](#)

ワークシート [141](#)

Line details パラメーター [53](#)

Link level error recovery on implicit links パラメーター [55](#)

Link station name パラメーター [74](#)

Linux クライアント

最大 process_count [133](#)

最大エレメント数の最大値 [134](#)

最大ヘッド・カウント [134](#)

ドメイン名 [133](#)

Linux コマンド [103](#)

Local LU alias パラメーター [84](#)

Local LU name パラメーター [74](#)

Local LU パラメーター [84](#), [86](#)

Local node ID パラメーター [60](#)

Local SAP number パラメーター [54](#)

Location パラメーター [74](#)

LS (リンク・ステーション) [56](#)

LU

説明 [6](#)

タイプ [6](#)

LU 0

説明 [6](#)

LU 1 [6](#)

LU 2 [6](#)

LU 3 [6](#)

LU 6.2

LU 6.2 (続き)

構成 [69](#)

説明 [6](#)

LU alias パラメーター [70](#), [77](#)

LU in pool パラメーター [66](#)

LU name パラメーター [65](#), [70](#)

LU number パラメーター [66](#), [71](#), [99](#)

LU traffic パラメーター [57](#)

LU type パラメーター [66](#)

LU タイプ 0 から 3

構成方式 [65](#)

パラメーター [65-67](#)

必要な追加構成 [67](#)

LU プール

構成メソッド [67](#)

定義 [67](#)

パラメーター [68](#)

表示 [67](#)

LU 別名、オーバーライド [130](#)

LU 別名のオーバーライド [136](#)

LU-LU セッション [7](#)

LUA

構成 [65](#)

ワークシート [160](#)

LUA LU 名、オーバーライド [130](#), [136](#)

LUALIAS パラメーター [167](#)

lualiasmap.txt [130](#)

M

MAC (メディア・アクセス制御) [23](#)

MAC address パラメーター [58](#)

Maximum active template instances パラメーター [55](#)

Maximum RU size パラメーター [83](#)

Maximum session limit パラメーター [82](#)

Maximum window size パラメーター [83](#)

Member of default pool パラメーター [71](#)

Minimum contention loser sessions パラメーター [82](#)

Minimum contention winner sessions パラメーター [82](#)

Mode パラメーター [85](#)

Motif administration program

dialog [43](#)

Domain window [38](#)

Motif 管理プログラム

記述 [28](#)

使用 [37](#)

ダイアログ [44](#)

ツールバー・ボタン [42](#)

ノード・ウィンドウ [39](#)

ヘルプ [45](#)

呼び出し [37](#)

リソース・ウィンドウ [37](#)

リソース項目 [41](#)

MPC group パラメーター [59](#)

MS (管理サービス) [13](#)

Multiple instances supported パラメーター [78](#)

N

Name パラメーター

セキュリティー・アクセス・リスト [88](#)

モード [82](#)

リンク・ステーション [57](#)

Name パラメーター (続き)
CPI-C シンボリック宛先 [84](#)
LU プール [68](#)
NAP (ネットワーク・アクセス・プロセス) [116](#)
NAU (ネットワーク・アクセス可能単位) [5](#)
NetView
画面表示 [104](#)
コマンド [103](#)
コマンド入力域のサイズの変更 [104](#)
サービス・ポイント [103](#)
説明 [103](#)
バージョン番号 [103](#)
プログラム [103](#)
NN (ネットワーク・ノード) [4](#)
Node ID パラメーター [49](#)
Node's SNA network name パラメーター [72](#)
NOF (ノード・オペレーター機能) API [32](#)

P

Parameters are for invocation on any LU パラメーター [77](#)
Partner LU name パラメーター [73](#), [75](#)
Partner LU パラメーター [85](#), [86](#)
Partner TP パラメーター [85](#)
Password パラメーター [86](#), [87](#)
PIP allowed パラメーター [80](#)
Poll address パラメーター [58](#)
Pool name パラメーター [67](#)
Port number パラメーター [53](#)
Printer LU assigned パラメーター [93](#)
Printer LU パラメーター [95](#)
PU
説明 [5](#)
PU ID パラメーター [63](#)
PU Name パラメーター [62](#)
PUCP (物理装置制御点) [7](#)

Q

QLLC
ワークシート [148](#)

R

RCF
機能 [30](#)
コマンド構文 [104](#)
有効な文字 [105](#)
Receive pacing window パラメーター [83](#)
Remote API Client on Windows
LU 別名、LUA LU 名、または TP 名のオーバーライド [130](#)
Remote node ID パラメーター [60](#)
Remote node name パラメーター [59](#)
Remote node role パラメーター [61](#)
Remote node type パラメーター [60](#)
Remote X.25 address パラメーター [59](#)
Reset to SNA defined values パラメーター [84](#)
Restrict access パラメーター [80](#)
Restrict maximum RU size パラメーター [83](#)
Retry contacting DLUS indefinitely パラメーター [63](#)
RJE (リモート・ジョブ入力) [6](#)
Route incoming Allocates to running TP パラメーター [78](#)

RTP
エンドポイント [21](#)
説明 [11](#), [17](#)
RU (要求単位) [83](#)

S

SAP (サービス・アクセス・ポイント) [23](#)
SAP number パラメーター [59](#)
SATF
直接接続 [21](#)
APPN ネットワークの [22](#)
SDLC
ワークシート [142](#)
Secure Sockets Layer (SSL)
クライアント認証 [93](#), [96](#)
サーバー認証 [94](#), [96](#), [98](#)
データ暗号化 [93](#), [96](#)
Security access list パラメーター [80](#)
Security パラメーター [85](#)
SEND 関数 [10](#)
Session timeout パラメーター [83](#)
SNA
階層構造 [1](#)
基本概念 [1](#)
層 [1](#)
ネットワークのタイプ [2](#)
APPN の概念 [11](#)
SNA port name パラメーター [52](#), [57](#)
SNA ゲートウェイ
目的 [98](#)
ワークシート [152](#)
SNA ソフトウェアの使用可能化
AIX または Linux 上のリモート API クライアント [132](#)
SNA ネットワーク情報
Windows 上のリモート API クライアント [118](#)
sna_clnt.net ファイル [133](#)
snaadmin プログラム [30](#)
snanetutil プログラム [112](#)
SPCF
コマンド [105](#)
説明 [30](#), [103](#)
Specify timeout パラメーター [83](#)
SSCP (システム・サービス制御点) [7](#)
SSCP 従属 LU [8](#)
SSCP-LU セッション [7](#)
SSCP-PU セッション [8](#)
Standard error パラメーター [79](#)
Standard input パラメーター [79](#)
Standard output パラメーター [79](#)
stop コマンド [36](#)
Support TN3270E パラメーター [92](#)
Supports parallel sessions パラメーター [74](#)
Sync level パラメーター [80](#)
System name [71](#)

T

TCP/IP port number パラメーター [93](#), [96](#), [97](#)
TDU (トポロジー・データベース更新) [19](#)
Telnet client address パラメーター [95](#)
TN サーバー
アクセス・レコード [92](#), [94](#)

TN サーバー (続き)

アクセス・レコード・パラメーター [92, 93](#)
アソシエーション・レコード・パラメーター [94, 95](#)
関連付けレコード [94](#)
ワークシート [153](#)

TN リダイレクター

アクセス・レコード [95](#)

TN3270 client address パラメーター [92](#)

TN3270 クライアント

ドゥルル [169](#)
DDDLU の LU モデル [169](#)

TP

クライアント [138](#)
構成 [75](#)
構成メソッド [76](#)
説明 [4](#)
ソース [9, 75](#)
ターゲット [9](#)
呼び出し [9](#)
呼出し [75](#)
呼び出し可能 [9, 76](#)
呼び出しパラメーター [77-79](#)
標的 [76](#)
APPC 定義パラメーター [79, 80](#)

TP name パラメーター [77, 79](#)

TP 構成パラメーター

ENV [167](#)
GROUP [166](#)
LUALIAS [167](#)
USERID、AIX または Linux [166](#)
USERID、Windows [168](#)

TP の呼び出し [75](#)

TP 名、オーバーライド [130, 136](#)

TRS (トポロジー/ルーティング・サービス) [18](#)

U

UCF

コマンド構文 [107](#)
コマンドの例 [107](#)
使用できるコマンド [107](#)
セキュリティー [109](#)
説明 [30, 103](#)
ファイルへのアクセス [109](#)
有効なコマンド [107](#)
ユーザー名 [109](#)

UDP/IP 通信 [114](#)

Uninterpreted Name パラメーター [74](#)

Upstream DLUS name パラメーター [61](#)

Upstream LU name パラメーター [99](#)

Use default LU パラメーター [84](#)

User ID パラメーター [78, 85, 87](#)

USERID パラメーター

AIX または Linux [166](#)
Windows [168](#)

Users in access list パラメーター [88](#)

ux-cancel コマンド [108](#)

V

VRN

説明 [22](#)

VTAM (仮想記憶通信アクセス方式) [11](#)

W

Web 管理パッケージ
説明 [29](#)

WebSphere アプリケーション・サーバー [115](#)

Wildcard partner LU name パラメーター [73](#)

window

Domain [38](#)

Windows 上の Remote API Client

クライアント/サーバーのトレース情報 [126](#)

構成情報 [119](#)

サーバー情報 [121](#)

使用可能に設定 [116](#)

使用不可に設定する [117](#)

内部トレース情報 [128](#)

admin_msg [127](#)

all_api [125](#)

API トレース情報 [125](#)

appc [126](#)

APPCLLU [129](#)

APPCTPN [129](#)

broadcast_attempt_count [120](#)

client_start_timeout [121](#)

CPI-C アプリケーション・データ [129](#)

cpic [126](#)

csv [126](#)

CSV アプリケーション・データ [129](#)

CSVTBLG [129](#)

data [127](#)

datagram [127](#)

domain [119](#)

file1 [125](#)

file1 (CS_tracing) [126](#)

file1 (Internal_tracing) [128](#)

file2 [125](#)

file2 (CS_tracing) [127](#)

file2 (Internal_tracing) [128](#)

flip_size [125](#)

flip_size (CS_tracing) [127](#)

flip_size (Internal_tracing) [128](#)

invoked TPs [120](#)

lan_access_timeout [120](#)

maximum_element_count [120](#)

maximum_header_count [120](#)

maximum_process_count [119](#)

nof [126](#)

poll_timer [120](#)

receive [127](#)

rui [126](#)

send [127](#)

server_lost_timeout [121](#)

Server1 [121](#)

Server2-Server9 [122](#)

trace_flushing [128](#)

trace_level [128](#)

truncation_length [125](#)

Windows 上のリモート API クライアント

エラー・ファイル・ラップ・サイズ [123](#)

監査ファイル・ラップ・サイズ [124](#)

クライアント構成ユーティリティー [118](#)

構成 [118](#)

状況 [117](#)

正常性エラー [124](#)

正常に監査されず [125](#)

Windows 上のリモート API クライアント (続き)

バックアップ・エラー・ファイル [123](#)

例外ログ・ロギングが使用可能 [122](#)

ロギング情報 [122](#)

ログ・ディレクトリ [123](#)

audit_file [124](#)

audit_logging_enabled [123](#)

error_file [123](#)

Windows オープン・システム・アーキテクチャー (WOSA)
[116](#)

Windows クライアント

ネットワーク・アクセス・プロセス (NAP) [116](#)

WOSA (Windows オープン・システム・アーキテクチャー)
[116](#)

X

xsnaadmin プログラム [28](#)

[特殊文字]

TN Redirector

アクセス・レコード・パラメーター [95-97](#)

ワークシート [154](#)

UNIX コマンド機能 (UCF) [30](#)

DLUR PU

パラメーター [62, 63](#)



SC88-9999-05

