

IBM Communications Server for Data Center Deployment on Linux



管理ガイド

バージョン 7.0

IBM Communications Server for Data Center Deployment on Linux



管理ガイド

バージョン 7.0

お願い

本書および本書で紹介する製品をご使用になる前に、219 ページの『付録 D. 特記事項』に記載されている情報をお読みください。

本書は、IBM Communications Server for Data Center Deployment on AIX or Linux バージョン 7.0 (プログラム番号 5725-H32)、および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。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC31-6771-04
IBM Communications Server for Data Center Deployment
on Linux
Administration Guide
Version 7.0

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第6版第1刷 2012.12

© Copyright IBM Corporation 2000, 2012.

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	xii
本書の編成	xii
表記上の規則	xiii
グラフィックの規則	xiii
本リリースで新しくなった部分	xiv
新しい機能	xiv
廃止された機能	xv
詳細について	xv

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	2
ネットワークのタイプ	2
SNA ノード	2
接続	6
トランザクション・プログラム	6
アプリケーション・プログラミング・インターフェース	6
ネットワーク・アクセス可能単位	7
セッション	9
会話	12
モード	13
経路選択	13
サービス・クラス	13
APPN の基本概念	14
APPN ノード・タイプ	14
APPN 制御点 (APPN Control Point)	17
リソースの検出	18
セッション・ルーティング	22
分岐エクステンダー (Branch Extender)	29
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	31

第 2 章 CS Linux の管理 33

CS Linux の管理の概要	33
管理責任	33
管理ツール	34
管理許可	41
CS Linux 構成の計画	41
計画ワークシート	42
作業シート	42
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	42
CS Linux が使用する環境変数の設定	43
CS Linux プログラムへのパスの指定	43

CS Linux サーバーの使用可能化	43
CS Linux サーバーを使用不可に設定する方法	45
Motif 管理プログラムの使用	46
Motif 管理プログラムの呼び出し	47
リソース・ウィンドウ	47
リソース・ダイアログ	54
状況ダイアログ	56
ヘルプ・ウィンドウ	57
コマンド行管理プログラムの使用	58

第 3 章 基本構成作業 61

クライアント/サーバー機能の構成	61
ノードの構成	62
ノードの構成パラメーター	62
追加構成	63
ロギングの構成	63

第 4 章 接続コンポーネントの定義 67

DLC、ポート、および接続ネットワークの定義	68
DLC、接続ネットワーク、およびポートの構成パラメーター	69
追加構成	73
リンク・ステーションの定義	73
リンク・ステーションの構成パラメーター	74
追加構成	81
DLUR PU の定義	81
DLUR PU の構成パラメーター	82
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	83
追加構成	84

第 5 章 従属 LU の構成 85

LU タイプ 0 から 3 の定義	85
LU タイプ 0 から 3 の構成パラメーター	85
追加構成	87
LU プールの定義	87
LU プールの構成パラメーター	88

第 6 章 APPC 通信の構成 91

ローカル LU の定義	92
ローカル LU の構成パラメーター	93
追加構成	94
リモート・ノードの定義	94
リモート・ノードの構成パラメーター	95
追加構成	96
パートナー LU の定義	96
パートナー LU の構成パラメーター	97
パートナー LU 用のリンク・ステーション・ルーティングの定義	98
追加構成	100
TP の定義	100

サーバー上での TP 呼び出しパラメーター	102
TP 定義パラメーター	106
モードおよびサービス・クラスの定義	107
モードの構成パラメーター	109
追加構成	112
CPI-C サイド情報の定義	112
CPI-C の構成パラメーター	112
追加構成	114
APPC セキュリティーの構成	115
セッション・セキュリティーの構成	115
会話セキュリティーの構成	116
セキュリティー・アクセス・リストの構成	116

第 7 章 ユーザー・アプリケーションの定義 119

第 8 章 パススルー・サービスの構成 121

TN サーバーの構成	121
TN サーバー・アクセス・レコードの構成	122
TN サーバー・アソシエーション・レコードの構成	126
TN リダイレクターの構成	127
TN リダイレクターのアクセス・レコードの構成	127
SNA ゲートウェイの構成	131
ダウンストリーム LU の構成パラメーター	133
追加構成	133
DLUR の構成	134
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	135

第 9 章 NetView からの CS Linux の管理 137

ホスト NetView プログラムの使用	137
NetView 画面の表示	138
コマンド入力域のサイズの変更	138
RCF コマンド構文の概要	139
大文字とエスケープ文字	139
SPCF の使用	140
SPCF で使用する管理コマンドに関する制約事項	140
SPCF コマンドの例	141
UCF の使用	141
UCF コマンド構文	142
使用できるコマンド	142
UCF コマンドの例	143
Linux システム・コマンドからの出力	143
コマンドの取り消し	144
UCF のセキュリティー	144

第 10 章 CS Linux クライアント/サーバー・システムの管理 147

クライアント/サーバー構成の変更	148
クライアントを別のドメインに移動する	149
IP ネットワーキングの要件	149
IPv4 および IPv6 アドレッシング	150
クライアント/サーバー構成におけるホスト名	150

IP ポート番号の設定	150
LAN アクセス・タイムアウト	151
Remote API Client の HTTPS アクセス	153
Windows 上の Remote API Client の管理	153
Windows 上の Remote API Client を使用可能に設定する	154
Windows 上の Remote API Client の状況の表示	155
Windows 上の Remote API Client を使用不可に設定する	155
Windows 上の Remote API Client 構成	156
サーバーのドメイン上のクライアントによって使用される SNA LU リソースの追跡	171
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 別名と、サーバーのドメインの LU 別名のマッピング	172
AIX または Linux 上の Remote API Client の管理	174
AIX または Linux 上の Remote API Client を使用可能および使用不可に設定する方法	174
クライアント・ネットワーク・データ・ファイル (sna_clnt.net)	175
サーバーのドメイン上のクライアントによって使用される SNA LU リソースの追跡	179
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 別名と、サーバーのドメインの LU 別名のマッピング	180
クライアント TP の定義	182

付録 A. 構成計画ワークシート 183

ノード・ワークシート	183
APPN ネットワーク・ノード	183
APPN エンド・ノード	184
APPN 分岐ネットワーク・ノード	185
LEN ノード	185
接続ワークシート	186
SDLC	186
トークンリング	188
イーサネット	190
QLLC (X.25)	192
マルチパス・チャンネル	194
Enterprise Extender (HPR/IP)	195
パススルー・サービス・ワークシート	196
ローカル・ノード上の DLUR	196
ダウンストリーム・ノード用のパススルー DLUR	197
SNA ゲートウェイ	197
TN サーバー	198
TN リダイレクター	199
ユーザー・アプリケーション・サポート・ワークシート	200
APPC	200
CPI-C	203
5250	204
3270	204
LUA	205

付録 B. コマンド行からの呼び出し可能 TP の構成	207
呼び出し可能 TP 定義のファイル形式	208
付録 C. DDDLU 用の TN3270 LU モデ ルの構成.	217
付録 D. 特記事項	219
商標	221
参考文献.	223

CS Linux バージョン 7.0 資料	223
システム・ネットワーク体系 (SNA) 関連資料	225
ホスト構成関連資料	225
z/OS Communications Server 関連資料	225
TCP/IP 関連資料	225
X.25 関連資料	226
APPC 関連資料.	226
プログラミング関連資料.	226
その他の IBM ネットワーキング資料	226
索引	227

表

1. 表記上の規則	xiii	3. RCF コマンド内でのエスケープ文字の使用	140
2. 標準モード名と COS 名	108		



1. SNA サブエリア・ネットワーク	4	10. ノード EN1 から APPN ネットワーク内の各ノ	
2. 複数セッションと並列セッション	11	ードへの直接リンクに必要な定義	27
3. トランザクション・プログラムと論理装置との		11. 仮想ノードを使用する直接リンクに必要な定義	29
通信	12	12. 分岐エクステンダー	30
4. サンプル APPN ネットワークの一部	15	13. CS Linux の「ドメイン (Domain)」ウィンドウ	49
5. LEN ノード・ディレクトリー	20	「Node (ノード)」ウィンドウ	50
6. エンド・ノード・ディレクトリー	20	15. CS Linux ツールバー	53
7. ネットワーク・ノード・ディレクトリー	21	16. サンプル・ダイアログ	55
8. ネットワーク・ノードのネットワーク・トポロ		17. 状況ダイアログの例	56
ジー・データベース	24	18. ヘルプ・ウィンドウの例	57
9. 共有アクセス転送機能を使用する APPN ネット		19. NetView 画面の例	138
ワーク	27		

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、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 が動作する環境は、以下のとおりです。

- Linux が稼働する 32 ビット Intel ワークステーション (i686)
- 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 pSeries® コンピューター (ppc64)

CS Linux for System z®

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

本書では、相違を明示しない限り、「Communications Server for Linux」という名前は両方の変種を示し、「CS Linux コンピューター」という用語は CS Linux が稼働する任意のタイプのコンピューターを示します。

本書は CS Linux のバージョン 7.0 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 が動作するハードウェアと、Linux オペレーティング・システムに精通している必要があります。また、システムが接続しているネットワークについての知識があり、SNA の一般概念も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 の関連資料に関する追加情報については、『参考文献』の節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。

本書の編成

本書は次の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。
- 33 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。
- 61 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。
- 67 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。
- 85 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。
- 91 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。
- 119 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。
- 121 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・システムの間の通信をサポートする機能です。
- 137 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。
- 147 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。
- 183 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。

- 207 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。
- 217 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。tn3270dev.dat 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、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

表記の対象	表記例
強調したい語句	削除する前にファイルをバックアップしてください
資料タイトル	<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	<i>opcode, LU name</i>
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, On 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
コマンドまたは Linux ユーティリティー	define_node, cd
特定タイプのすべてのコマンドを総称的に参照する場合	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
指定値を表す変数	<i>filename; LU_name; user_ID</i>
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、呼び出し、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 pSeries、または Linux for System 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 7.0 は、CS Linux バージョン 6.4 の後継製品です (先行バージョンのサポートは継続されます)。

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。

- サーバーのドメイン上でクライアントによって使用される SNA LU リソースの追跡。

LU のプールにアクセスするクライアントによって使用される SNA LU リソースを追跡できます。この機能は、サーバー上の LU レコードを、各トランザクションのクライアントのシステム名 (DNS 情報) によって動的に更新します。

この機能を使用して、モバイルである (またはメインフレーム・ホストの観点からモバイルに見える) クライアントは、ホストからアクセスすることができます。さらにこの機能により、特定の LU を最後に使用したユーザーをリアルタイムで表示する照会を実行でき、ワークステーションからホスト・アプリケーションへの接続に関する問題判別に役立てることができます。

- クライアント・アプリケーション上でハードコーディングされている LU 別名、サービスのドメインの LU 別名へのマッピング。

一部のアプリケーションは、特定の LU 別名を使用するようにハードコーディングされています。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。これによって SNA のフルスタック実装からマイグレーションした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。さらにサーバーのドメイン内のすべてのサーバー・リソースを使用して、サーバー・ドメイン実装のいくつかの冗長機能を利用できます。

- IPV6 環境での Enterprise Extender および接続ネットワークの拡張サポート。

この機能を使用して、IPV6 の接続ネットワークを定義できます。この接続ネットワークでは、接続は DNS ホスト名を使用するのではなく IP アドレスを使用します。

- TN3270 サーバーと TN リダイレクターの機能拡張は、以下のとおりです。
 - より高度なセキュリティーを提供する、256 ビット SSL 暗号化のサポート。
 - CS Linux for System z のみ: SSL ハンドシェイク中のパフォーマンスを向上させるための、z90Crypto アダプターのサポート。

廃止された機能

このリリースで廃止された機能はありません。

詳細について

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、33 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。

- 『システム・ネットワーク体系』では、SNA の定義を示します。
- 2 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。
- 14 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。
- 31 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) とは、ハードウェアおよびソフトウェアの多種多様なデータ通信製品間で通信するための共通規則を指定した、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・コンポーネント間で交換されるメッセージのレイアウトを定義する形式と、メッセージに応答してネットワーク・コンポーネントがとるアクションを定義するプロトコルの、二種類の定義から成っています。

SNA ネットワークは、リンクで結合される、SNA を使用して通信するコンピューターの集合体です。

本来は、SNA はホスト・コンピューターとの通信のために設計されたものです。各ネットワークまたはサブネットワークは、ホストにより制御されていました。つまり他のコンピューターはホストと直接通信するのみであり、コンピューター相互間で通信することはできませんでした。この旧式の、ホスト制御形式のネットワークを、サブエリア SNA とも呼びます。その後、SNA は、ネットワーク内のコンピューター間の対等通信をホスト抜きでサポートできる形に拡張されました。この新しい対等レベルのネットワークが、APPN です。

多くの SNA ネットワークは、サブエリアと対等ネットワークの両方のエレメントを持っています。ネットワークがサブエリア SNA から APPN へと移行する過程では、APPN 対応のホストは、旧式のシステムを制御し、同時に新式のシステムに対しては対等に機能します。同様に、1 つのコンピューターが対等コンピューター

(APPN ネットワークの) と旧式のホストの両方にアクセスできます。ホストとの通信はそのホストにより制御されますが、他のコンピューターとの通信は対等通信であり、ホストに無関係に行われます。

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していません。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、ノードは、SNA プロトコルを実装し、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 つは持っている Linux システムなどのデバイスと、それらに関連するソフトウェア構成要素です。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 System 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。

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。

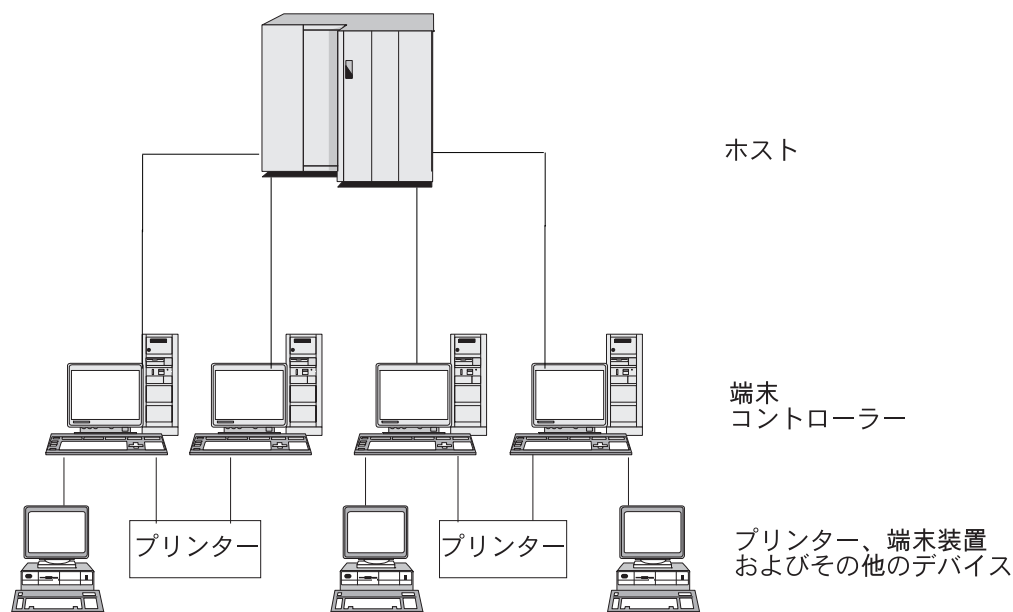


図 1. SNA サブエリア・ネットワーク

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に使わ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認できます。

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、セッションのルーティングを行い、ネットワーク・ノードに直接接続したエンド・ノード (EN) とローエントリー・ネットワークング (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ーを単純化し、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。詳細は、29 ページの『分岐エクステンダー (Branch Extender)』を参照してください。
- ローエントリー・ネットワークング・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いるわけではありません。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、14 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、7 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 (『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 (『物理装置』を参照)
- 論理装置 (『論理装置』を参照)
- 制御点 (9 ページの『制御点 (control point)』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 (9 ページの『制御点 (control point)』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

一定義のデータ・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[®]) で稼働し、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することができます。

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 / 仮想記憶 (IMS/VS) 環境で実行され、IBM 8100 情報システムと通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。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。特定のホスト・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) や一部の小売り業界用および銀行業界用の POS システム (IBM 4680 Store System Operating System 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 プロトコルを使用することを推奨します。

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 (control point)

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、17 ページの『APPN 制御点 (APPN Control Point)』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 (11 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 (17 ページの『APPN 制御点 (APPN Control Point)』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動化を要求します。

- 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 (11 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 リクエスター (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、31 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 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。

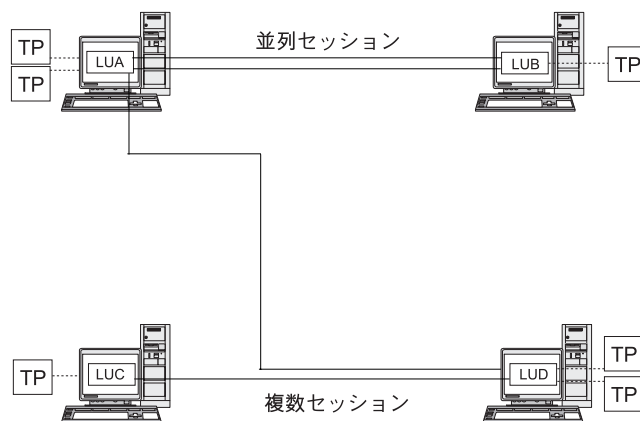


図2. 複数セッションと並列セッション

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。

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。

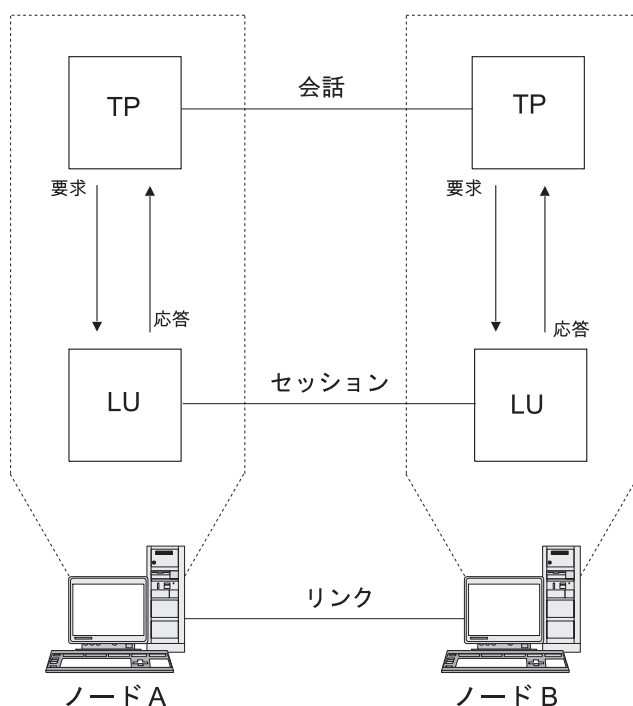


図 3. トランザクション・プログラムと論理装置との通信

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 との間で受け渡す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（『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なります。

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、セキュリティーの許容レベル、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、22 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)、CS Windows 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 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 (『APPN ノード・タイプ』を参照)
- APPN 制御点 (APPN Control Point) 機能 (17 ページの『APPN 制御点 (APPN Control Point)』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 (18 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 (22 ページの『セッション・ルーティング』、および 28 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 (15 ページの『APPN ネットワーク・ノード』を参照)

- エンド・ノード (16 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 (16 ページの『LEN ノード』を参照)。

サンプル APPN ネットワークは、図 4 に示してあります。

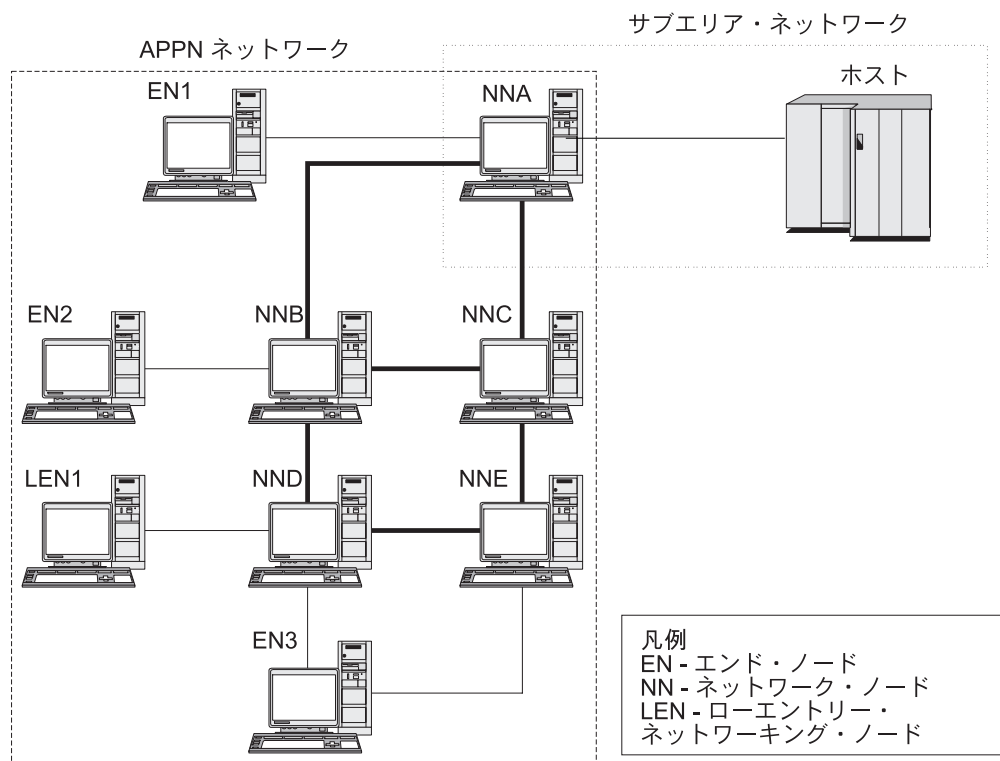


図 4. サンプル APPN ネットワークの一部

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 (26 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 (15 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せんが、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 Control Point)

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・ノードとの間には、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の詳細については、「*IBM Communications Server for Data Center Deployment on Linux 入門*」を参照してください。

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

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 20 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。

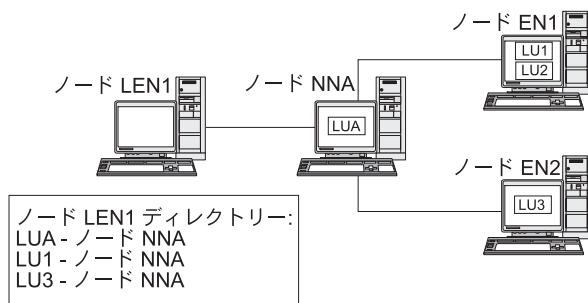


図5. LCN ノード・ディレクトリー

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。

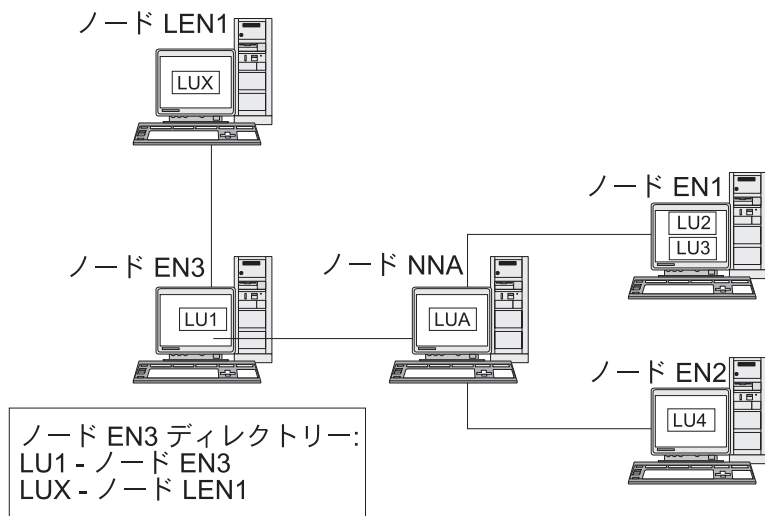


図6. エンド・ノード・ディレクトリー

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 図7 に示します。

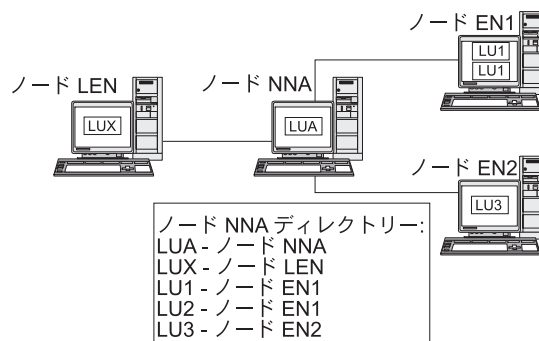


図7. ネットワーク・ノード・ディレクトリー

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必要があります。

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チャー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) の機能が組み込まれています。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、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 (28 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。

24 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。

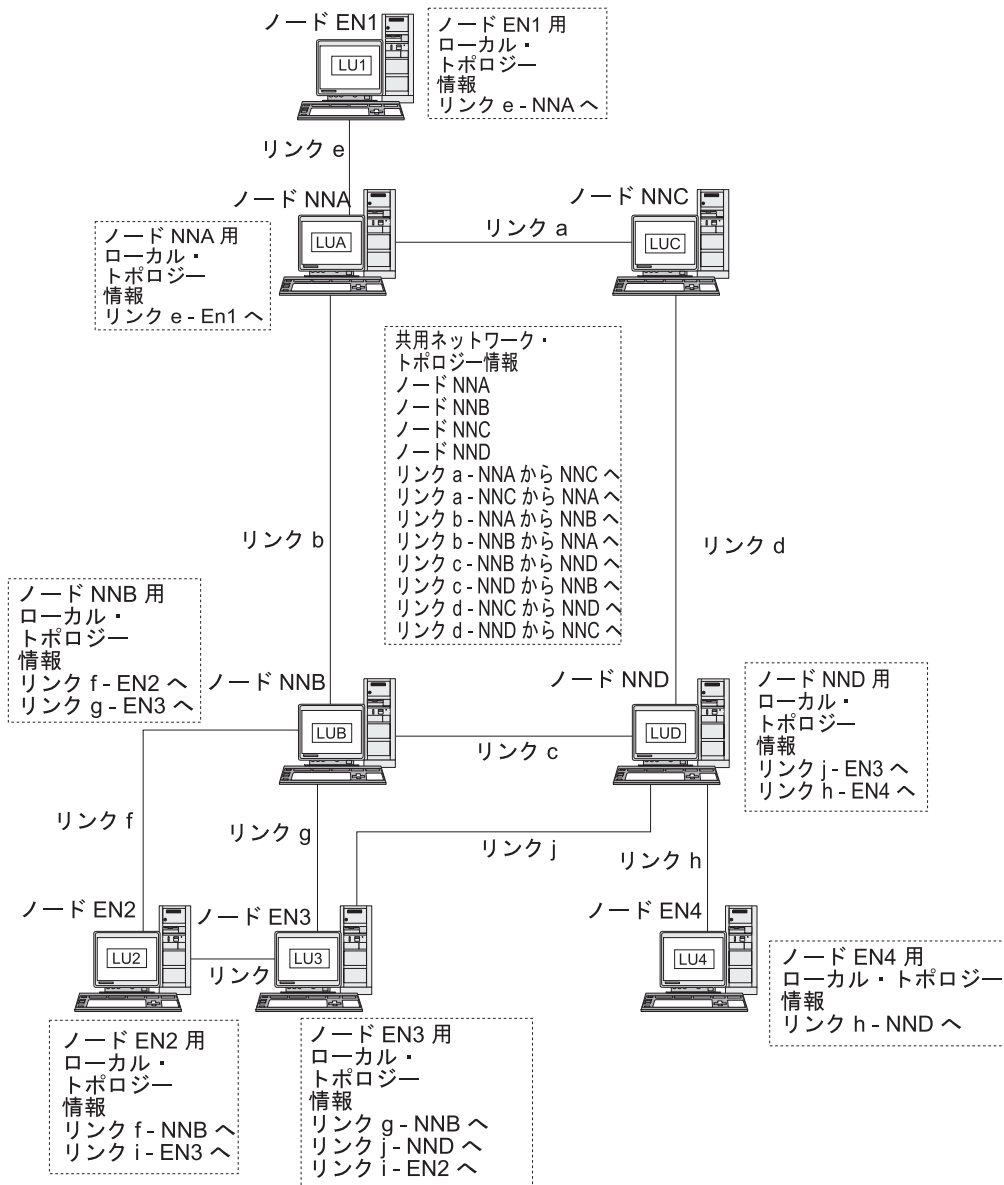


図8. ネットワーク・ノードのネットワーク・トポロジー・データベース

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) が活動化または非活動化

するとき、または既存のリソースの特性を変更するときに、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、27 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 図9 に示します。

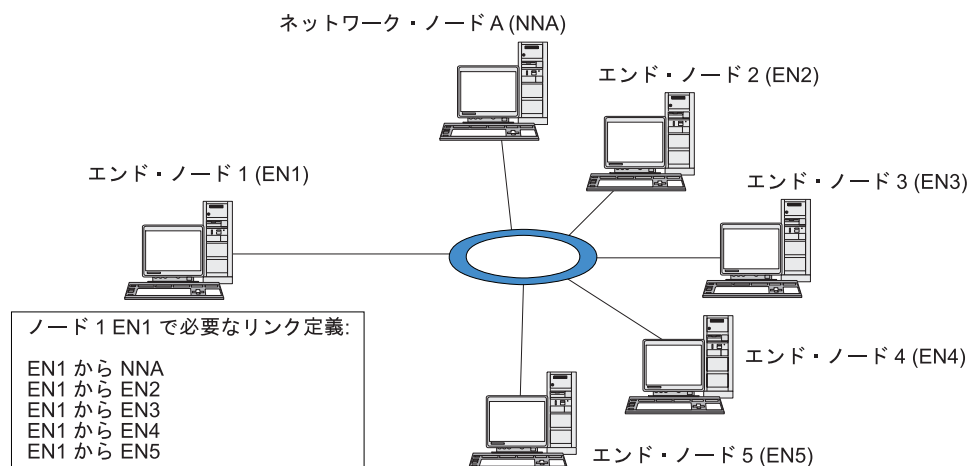


図9. 共有アクセス転送機能を使用する APPN ネットワーク

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、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

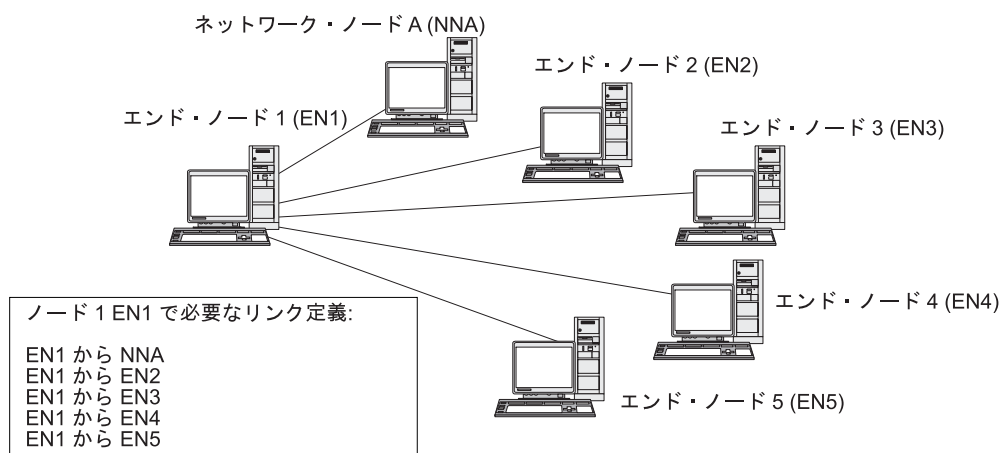


図10. ノード EN1 から APPN ネットワーク内の各ノードへの直接リンクに必要な定義

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 ネットワークについては、APPN 接続ネットワークを使用すると、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が大幅に減少します。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 29 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。

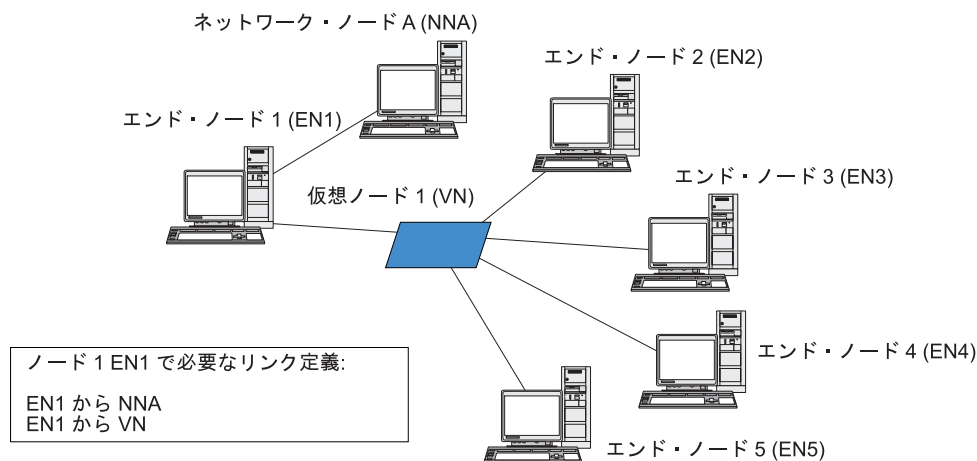


図 11. 仮想ノードを使用する直接リンクに必要な定義

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 (27 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 (SAP) アドレスおよびメディア・アクセス制御 (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 (Branch Extender)

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

APPN の基本概念

必要があります。ネットワークのサイズが拡大するにつれて、保管されている情報量やトポロジーに関するネットワーク・トラフィックの量が大規模になり、管理が難しくなることがあります。

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。

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。

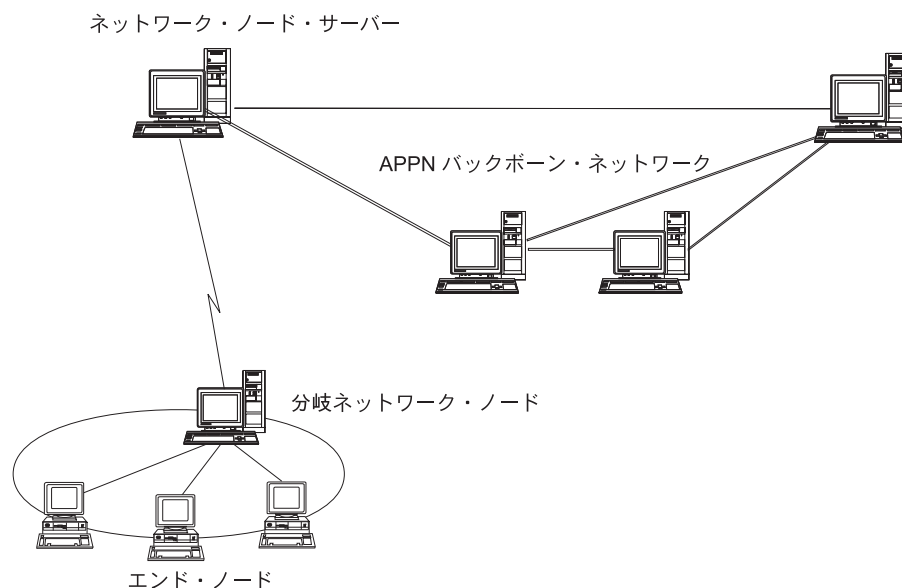


図 12. 分岐エクステンダー

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、『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。41 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 42 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 (46 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。

この代わりに、58 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を使用することもできます。

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、34 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 の管理を開始する前に、CS Linux 製品の主要な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・プログラムで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、137 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に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、41 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 (『Motif 管理プログラム』を参照)
- Web 管理プログラム (36 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 (37 ページの『コマンド行管理プログラム』を参照。または「*IBM Communications Server for Data Center Deployment on Linux* 管理コマンド解説書」を参照)
- サービス・ポイント・コマンド機能 (38 ページの『リモート・コマンド機能』を参照)
- 構成ファイル (38 ページの『構成ファイル』を参照)
- 診断ツール (39 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。詳細は、39 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ための最も簡単な方法は、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 のリソースを表示し管理する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 の運用をリアルタイムにモニターするために使用できますが、性能に多少影響を与えます。Linux for System z の LPAR には、直接接続されている端末がないため、**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・トラフィックおよびシステムに対する CPU 使用率がさらに増加します。より大規模なネットワーク構成の場合、コマンド行管理ツールまたは Web 管理パッケージの使用をお勧めします。これらのツールは、照会により更新を行い、性能に影響を与えかねない動的トラフィックを生成しません。

直接接続の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、さほど性能に影響を与えません。しかし、システム性能へのいかなる影響も、表示する内容および表示の更新頻度によって左右されます。

次の管理操作ができます。

- 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とアクティブ・リソースの現在の状況の照会
- リソースの削除

Motif 管理プログラムは、ノード・リソース (CS Linux ソフトウェアがサーバー上で実行されていれば、LAN 上の任意のサーバーが対象になります) とドメイン・リソースの両方を管理するために使用できます。それぞれの通信タイプ (3270 または APPC など) について、必要なリソースを構成するための説明がプログラムから示されます。

注: Motif 管理プログラムのウィンドウやダイアログは、特定のダイアログの選択内容によって決まるため、本書に示すものとは違っている場合があります。

Motif 管理プログラムには、ヘルプ画面があります。この画面では、SNA および CS Linux の概要、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説明が表示されます。

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 (詳しくは、33 ページの『第 2 章 CS Linux の管理』を参照)。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 &

開始済みの CS Linux サーバーすべてがメインスクリーンに表示されます。ノードが構成済みの場合は、プログラムからノードを選択すると、選択したノードの構成が表示されます。そうでなければ、プログラムはノードを選択するためのプロンプトを出して、ノードを定義するために必要な手順を案内します。

Motif 管理プログラムを使用して CS Linux リソースを定義し管理する方法の詳細については、47 ページの『Motif 管理プログラムの呼び出し』を参照するか、プログラムから提供されるヘルプ画面を参照してください。

注: Motif 管理プログラムでは、標準 CS Linux 構成に必要なすべてのパラメーターを設定できます。拡張パラメーターについては、Motif 管理プログラムはデフォルト値を提供します。指定する必要があるのは最も重要な構成情報のみであるため、SNA 通信を素早く簡単に確立できます。

その他の CS Linux 管理ツールにはコマンド行構成機能および NOF アプリケーション・プログラムも組み込まれていて、Motif 管理プログラムで表示されるものより広範囲にわたる構成パラメーターおよびオプションにアクセスできます。しかし、Motif 管理プログラムでは、構成のために必要なキー・フィールドが表示され、大半のユーザーが変更してはならないフィールドは隠されているので、ほとんどの場合、必要な構成はすべて Motif 管理プログラムから実行できます。コマンド行構成で指定されるデフォルト値は、Motif 管理プログラムのデフォルト値と異なる場合があります。これは、Motif プログラムの方が、値の選択方法が高度なためです。Motif 管理プログラムでは、構成作業の実際の状況に基づいて値を選択することが可能です。

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行い、そのあとで他の管理ツールを使用して追加機能を指定できます。後日 Motif 管理プログラムを使用して変更済みの構成を管理する場合、構成した追加機能は Motif プログラムには表示されませんが、他のツールを使用して加えた変更はそのプログラムに保存されています。

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 <http://www.ibm.com/software/network/commserver/support> 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。同一のパッケージが、CS Linux がサポートする Intel、System p[®]、および System z のプラットフォームで稼働します。

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 (37 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。

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コマンドを発行できます。**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・スクリプト内部から使用することもできます。

コマンドは、ノードのリソースを管理するために特定の CS Linux ノードに対して発行したり、マスターとバックアップのサーバーを管理するために SNA ネットワーク・データ・ファイルに対して発行したり、ドメイン・リソースを管理するためにドメイン構成ファイルに対して発行したりできます。

管理コマンドはすべてサーバー上で実行できます。ただし、IBM Remote API Client で実行できるコマンドについては制限があります。

- Windows クライアントには **snaadmin** プログラムがないため、コマンドは実行できません。
- AIX および Linux クライアントでは、**query** コマンドまたは **status** コマンドをすべて実行できます。「IBM Communications Server for Data Center Deployment on Linux 管理コマンド解説書」に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 から実行できることが明示されています。明示されていないコマンドは、サーバーからのみ実行できます。

コマンド行管理のヘルプを参照するには、次のいずれかのコマンドを使用します。

- **snaadmin -h** を使用すると、コマンド行管理の基本ヘルプと、コマンド行ヘルプの使い方の説明が表示されます。
- **snaadmin -h -d** を使用すると、**snaadmin** プログラムに指定できるコマンドのリストが表示されます。
- **snaadmin -h command** を使用すると、指定した *command* についてのヘルプが表示されます。
- **snaadmin -h -d command** を使用すると、指定の *command* についての詳細なヘルプが、そのコマンドに指定できる構成パラメーターのリストと共に表示されます。

詳しくは、「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。

リモート・コマンド機能

リモート・コマンド機能 (RCF) は、ホスト上で 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) により、ホスト NetView コンソールのオペレーターは、NetView から CS Linux 管理コマンドを実行して 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 Linux コマンドを発行できます。

RCF の詳細は、137 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、次のテキスト・ファイルに保管されます。

ノード構成ファイル

`/etc/opt/ibm/sna/sna_node.cfg` ファイルには、特定のノードの 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、ノードを稼働しているコンピューターに存在しています。このファイルには、ノードのリソースに関する情報が含まれていて、ノードで CS Linux が開始されたときにどのリソースをアクティブにするかが指定されています。

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。管理者は、条件に変更があった場合は、必要に応じて、他の管理ツールを使用して実行中のノード・リソースを変更できます。加えた変更は、すべてこのファイルに自動的に保管されるので、ノードを停止して再始動した場合も、変更済み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・ノードに関連しないリソース) に関する情報が含まれています。このファイルのマスター・コピーは、マスター・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 を開始するために必要な情報が入っているほか、その他の情報 (TP にアクセスするために必要なセキュリティ・レベルなど) も提供できます。このファイルは、TP を実行しているコンピューターに組み込まれています。

このファイルの詳細は、100 ページの『TP の定義』を参照してください。

構成は、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 を使用して変更できます。これらのツールはすべて、状況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、必要な変更を加えます。構成情報は非暗号化テキストとして保管されているため、標準的な ASCII テキスト・エディター (`vi` など) を使用してファイルを直接変更したり、Linux ユーティリティ (`awk` または `sed` など) を使用してシェル・スクリプトから変更することもできます。テキスト・エディターを使用して構成ファイルを変更するのは、CS Linux を開始する前

でなければなりません。CS Linux 構成ファイルの形式については、「*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。

注: CS Linux の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、構成全体をすべて定義する必要はありません。構成ファイルからは使用可能なリソースの初期定義が指定されますが、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、または変更できます。CS Linux は、現行の定義を保管して、システムの再始動時にそれを再び使用できるようにします。

次のファイルには、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あります。

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、どのサーバーがマスターか、どのサーバーがバックアップ・サーバーとして使用できるかについての情報があります。このバイナリー・ファイルは、マスター・サーバー上にあります。このファイルの内容は、管理プログラムまたは NOF API を使用して変更できます。

このファイルと、その変更方法の詳細は、61 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、CS Linux サーバーにアクセスするための情報があります。これは IBM Remote API Client が必要とする情報です。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、その変更方法の詳細は、175 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 クライアントにおける同等の情報の構成に関する情報は、147 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が書き込まれます。

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 (メッセージごとにログ記録の原因の詳細と、必要な処置がログ・ファイルに記載される)、または要約ログ (ログ記録のソースの要約とメッセージ・テキストのみが記載される) 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では、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『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、183 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もあります。

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 が使用する環境変数の設定

CS Linux は、その操作を制御するために多数の環境変数を使用します。これらはサーバー上で CS Linux ソフトウェアを使用可能にする前に設定しておく必要があります。これらを管理する最も簡単な方法は、それを CS Linux が始動時に読み取るテキスト・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、**environment** という名前のテキスト・ファイルを作成し、それぞれの環境変数を別々の行に設定します。例えば、次のように入力します。

```
export LANG=en_US
export PATH="$PATH:/opt/ibm/sna/bin"
export LD_LIBRARY_PATH=/usr/lib:/opt/ibm/sna/lib
export LD_RUN_PATH=/usr/lib:/opt/ibm/sna/lib
```

CS Linux プログラムへのパスの指定

CS Linux の実行可能プログラムは、CS Linux 固有のディレクトリーに格納されています。これらのプログラムを実行する場合は、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、初めてプログラムを実行する前に、このディレクトリーを `PATH` 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組み込みます。

Motif 管理プログラムは、ディレクトリー `/opt/ibm/sna/bin/X11` に格納され、その他のプログラムはディレクトリー `/opt/ibm/sna/bin` に格納されています。`.login` ファイルまたは `.profile` ファイルの `PATH` 環境変数の定義にこれらのディレクトリーを追加した場合、CS Linux はプログラムを自動的に見つけます。この代わりに、次の例に示すように、プログラムの実行時にディレクトリー名を指定することもできます。

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

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

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

本書に示すサンプル・コマンド行では、ディレクトリーを `PATH` 環境変数に追加していることが前提になっているため、ディレクトリー名は含まれていません。

CS Linux サーバーの使用可能化

このセクションでは、サーバーとしてインストールされたコンピューター（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コンピューター）上で CS Linux を使用可能にする方法を説明します。クライアント上で CS Linux を使用可能に設定する場合は、174 ページの『AIX または Linux 上の Remote API Client を使用可能および使用不可に設定する方法』を参照してください。

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

ローカル・ノードを (ローカルに、またはリモート CS Linux ノードから) 構成または管理するには、その前にローカル・システムで 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・プロンプトに次のコマンドを入力します。

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

注: **sna start** コマンドを実行する場合、CS Linux ソフトウェアでは、コマンドが発行されたディレクトリーが現行作業ディレクトリーとして使用され、1 つ以上のオープン・ファイル記述子がこのディレクトリーに維持されます。したがって、CS Linux ソフトウェアの実行中は、このディレクトリーが含まれているファイル・システムをアンマウントできないこととなります。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ーから CS Linux ソフトウェアを実行する必要があります。例えば、**sna start** コマンドを実行する前に `cd /` を使用して、ルート・ディレクトリー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、スタートアップ・ファイル `/etc/rc.d/init.d/snastart` を自動的に更新して **sna start** コマンドを組み込みます。これにより、CS Linux はシステムの起動時に自動的に起動されます。CS Linux が自動的に始動しないようにしたい場合は、この行を除去するかコメント化し、次に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してください。

sna start コマンドのパラメーターおよびオプションは次のとおりです。

-s CS Linux でメッセージをシステム・コンソールに書き込まないように指定します。このオプションを使用しないと、CS Linux は、終了時にメッセージをコンソールに書き込み、ある種のエラー・ログ・メッセージのテキストをログ・ファイルとコンソールの両方に書き込みます。

-m *kernel_memory_limit*

CS Linux が任意の時点で使用できるカーネル・メモリーの最大量を K バイト単位で指定します。(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 の構成要素がカーネル・メモリーを割り振ろうとして、CS Linux 構成要素に現在割り振られているメモリー量の合計がこの限度を超える場合は、その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かった場合は、カーネル・メモリーの使用量は限定されません。

-t カーネル・コンポーネント間のインターフェースすべてのトレースを活動化し、クライアント/サーバー・トレースも活動化します。(このオプションでは、DLC トレースはオンになりません。) トレースを使用すると、始動時に発生した問題を診断できます。このオプションを使用しなかった場合は、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。この場合、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でトレースを開始できます。

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

すべてのインターフェースのトレースをオンにすると、CS Linux コンポーネントのパフォーマンスが低下します。ソフトウェアを使用可能にしたあとで、コマンド行管理プログラム **snaadmin** を使用して、任意のインターフェースについて不要なトレースを個別に停止できます。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。

CS Linux は、初期化中であること、および初期化が正常に完了したことを示すメッセージを標準エラー出力 (通常は使用中の端末の画面) に書き込みます。

初期化が失敗した場合、エラーの原因についての情報のほか、(該当する場合は) Linux オペレーティング・システム・エラー・メッセージなどの追加情報もメッセージに含まれています。標準エラー出力に書き込まれるテキストには、エラー・ログ・ファイルにさらに詳細な情報が記録されていることを示すメッセージが含まれていることもあります。エラーの場合、**sna start** コマンドは、エラーの性質を示すゼロ以外の終了コードを戻して終了します。

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

SNA start コマンドの拡張機能オプション

特に新しい CS Linux 構成をテストしている場合など、時には、(ファイル `/etc/opt/ibm/sna/sna_node.cfg` および `/etc/opt/ibm/sna/sna_domn.cfg` の中の標準の構成ではなく、) 一時ファイルに保管した構成で Linux を開始したいことがあります。このために、**sna start** コマンドに、次の付加的なオプションを使用することができます。

-n *node_config_file*
-d *domain_config_file*

node_config_file は、ノード構成を保管したファイルの絶対パス名で (`/etc/opt/ibm/sna/sna_node.cfg` の代わり)、*domain_config_file* は、ドメイン構成を保管したファイルの絶対パス名です (`/etc/opt/ibm/sna/sna_domn.cfg` の代わり)。

注: これらのオプションは、一般の使用を意図したものではありません。特に必要のある場合を除いて、使用しないでください。

これらのオプションが指定されて Linux が実行されている場合、**snagetpd** コマンドは、常に標準の構成ファイルから情報を収集するため、正しく作動しません。**snagetpd** を使用する前に、これらのオプションを付けずに CS Linux を開始し、標準の構成ファイルを使用して実行するようになっていることを確認してください。

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。

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

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

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 58 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 43 ページの『CS Linux サーバーの使用可能化』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 ウィンドウを表示するために、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 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。

「ドメイン (Domain)」ウィンドウ

「ドメイン (Domain)」ウィンドウは、使用しているシステムの CS Linux ドメイン内にあるアクティブな SNA ノードがそれぞれ表示されます。(CS Linux がノード上で稼働していない場合は、「ドメイン (Domain)」ウィンドウにノードは表示されません。)各ノードの識別には、システムの名前が使用されます。「ドメイン (Domain)」ウィンドウには、ドメイン内の各ノードの現行状況も表示されます。

注: 「ドメイン (Domain)」ウィンドウのノードのリストにあるべきサーバーが欠落している場合は、サーバーのスイッチが入っていることと、CS Linux ソフトウェアがサーバー上で実行されていることを確認してください。必要に応じて、**sna start** コマンドを使用して CS Linux ソフトウェアをそのノード上で開始します (43 ページの『CS Linux サーバーの使用可能化』を参照)。

ドメイン内のノードの 1 つが、常にそのドメインの構成サーバーとして指定されます。「ドメイン (Domain)」ウィンドウ内で、そのノードの隣に「Master (マスター)」という語が表示されます。マスター構成サーバーは、ドメイン・リソースの構成情報を常時保管しています。バックアップ構成サーバーは、このウィンドウで「Backup (バックアップ)」という語によって識別されます。バックアップ構成サーバーは、ドメイン・リソースの構成情報のコピーを保管しています。

「ドメイン (Domain)」ウィンドウの例を 49 ページの図 13 に示します。

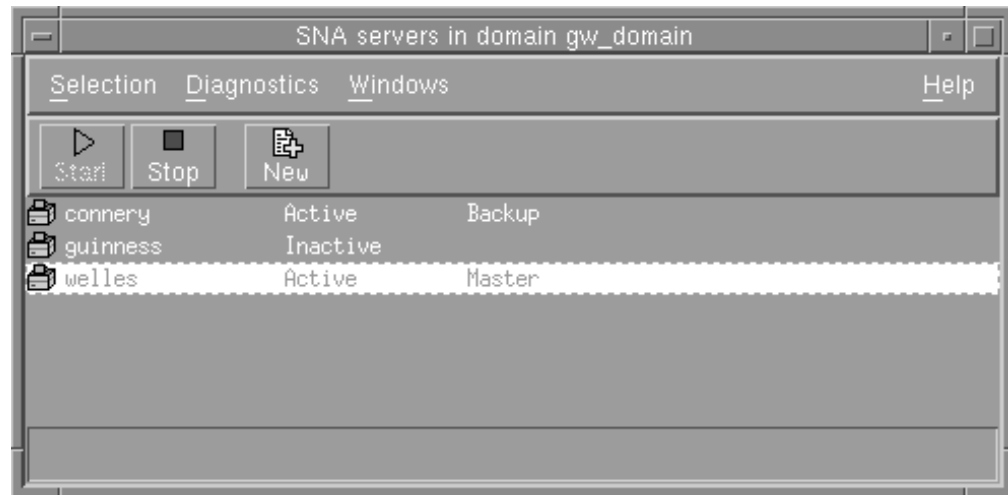


図 13. CS Linux の「ドメイン (Domain)」ウィンドウ

ドメイン内でアクティブ・ノード (CS Linux が実行されているノード) が構成されていない場合は、そのノードを構成するように CS Linux からプロンプトが出されます。

注: 「ドメイン (Domain)」ウィンドウには IBM Remote API Client はリストされません。クライアントは、CS Linux サーバー (SNA ノード) のリソースを使用して SNA リソースにアクセスします。

「ドメイン (Domain)」ウィンドウから、次の管理作業を実行できます。

ドメイン内のノードを開始または停止する

ノードの行を選択し、このウィンドウの「**Start (開始)**」ボタンまたは「**Stop (停止)**」ボタンをクリックします。(代わりに、ノードの行をクリックして、「**Selection (選択)**」メニューから「**Start Node (ノードの開始)**」または「**Stop Node (ノードの停止)**」を選択できます。)

特定のノードを管理する

「ドメイン (Domain)」ウィンドウで、そのノードの行をダブルクリックします。(代わりに、ノードの行をクリックして、「**Selection (選択)**」メニューから「**Properties (プロパティ)**」を選択できます。また、「**Windows (ウィンドウ)**」メニューからノードのウィンドウを選択することもできます。)

管理するノードを選択すると、CS Linux は 50 ページの図 14 に示すような「**Node (ノード)**」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、ドメインにはノードが 1 つしかないので、CS Linux は「ドメイン (Domain)」ウィンドウを表示しません。その代わりに、管理プログラムを開始すると CS Linux は「**Node (Node)**」ウィンドウを即時に表示します)

ドメインのサーバーのリストにノードを追加する

ノードの行をクリックして、「**Selection (選択)**」メニューから「**Make configuration server (構成サーバーの作成)**」を選択します。

ドメインのサーバーのリストからノードを除去する

ノードの行をクリックして、「**Selection (選択)**」メニューから「**Remove configuration server (構成サーバーの除去)**」を選択します。

ドメイン内のノードすべてのロギングを構成する

「Diagnostics (診断)」メニューから「Logging (ロギング)」を選択します。

特定のノードのトレースをオン/オフにする

ノードの行をクリックして、「Diagnostics (診断)」メニューから「Tracing on selected node (選択されたノードのトレース・オン)」を選択します。

ドメイン・リソースに関する情報を入手する

「Windows (ウィンドウ)」メニューのオプションを選択します。共有ドメイン・リソースのほかに、「Windows (ウィンドウ)」メニューは、ドメイン内の各「Node (ノード)」ウィンドウもリストします。

「Node (ノード)」ウィンドウ

サンプルの「Node (ノード)」ウィンドウを図 14 に示します。タイトル・バーにシステムの名前が示されています。

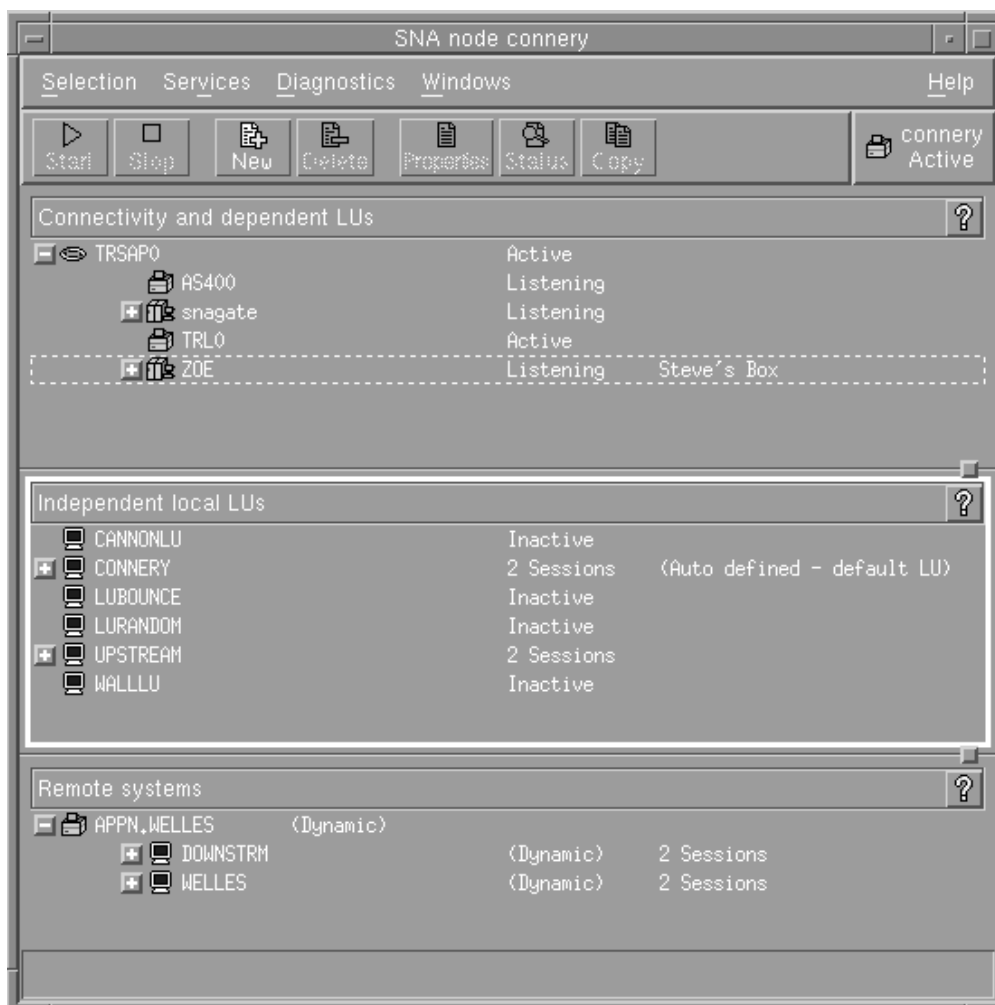


図 14. 「Node (ノード)」ウィンドウ

「ノード (Node)」ウィンドウから、その CS Linux ノードのすべてのリソースと構成要素を追加、削除、変更、および管理できます。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ています。どのリソースを表示するかを制御できます。

「Node (ノード)」ウィンドウの右上の隅の「Node」ボックスには、ノードが「Active (アクティブ)」であるか「Inactive (非アクティブ)」であるかが表示されます。

ノードに定義されているすべてのポート、ローカル LU、およびリモート・ノードが常に表示されます。「Node (ノード)」ウィンドウには、親ポートの下に各リンク・ステーションが表示され、親リンク・ステーションの下に各従属 LU が表示されます。また、ローカル LU およびリモート・ノードの下にはパートナー LU も表示されます。

「Node (ノード)」ウィンドウのボディは、次のように、ノードのリソースのタイプ別のペインに分割されています。

「Connectivity (接続)」ペイン

「Node (ノード)」ウィンドウの上段のペインには、ポート、各ポートのリンク・ステーションまたは PU、および個別のリンク・ステーションまたは PU の従属 LU を含む、ノードの接続リソースが表示されます。このウィンドウには、それぞれのリソースごとに現行の状況情報が表示されます。

「Independent Local LUs (独立ローカル LU)」ペイン

中段のペインには、ノードの独立 LU が表示されます。このウィンドウには、それぞれの LU ごとに、LU を使用中のセッションに関する情報も表示されます。

「Remote Systems (リモート・システム)」ペイン

下段のペインには、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す。また、各リモート・ノードまたはパートナー LU について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、ペイン間の境界線上でクリックしてドラッグします。

ペインを選択するには、ペイン内部でクリックします。リソースの行をクリックすることによって、ペイン内の特定のリソースを選択することもできます。ある項目について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。)

リストで表示されている各項目には、その項目に関する情報の中に、項目に所属するリソースがネストされています。例えば、リンク・ステーションは、その所属先のポートの下にグループ化されています。リソースが現在表示されていない場合は、項目の隣にある「**拡張 (Expand)**」ボタン



をクリックすると、その項目のリソースが表示されます。項目のリソースを隠すには、「**縮小 (Contract)**」ボタン



をクリックします。

「Node (ノード)」ウィンドウからは、次の管理作業を実行できます。

リソースの開始または停止

リソースを選択し、「**Start (開始)**」ボタンまたは「**Stop (停止)**」ボタンをクリックします。(代わりに、「**Selection (選択)**」メニューから「**Start item (項目の開始)**」または「**Stop item (項目の停止)**」を選択することもできます。)

項目へのリソースの追加

項目を選択し、「**New (新規)**」ボタンをクリックします (または、「**Selection (選択)**」メニューから「**New (新規)**」を選択します)。例えば、ポートにリンク・ステーションを追加するには、そのポートを選択して、「**New (新規)**」ボタンをクリックします。

リソースの削減

リソースを選択し、「**Delete (削除)**」ボタンをクリックします (または、「**Selection (選択)**」メニューから「**Delete**」を選択します)。

リソースの構成の表示または変更

リソースを選択し、「**Properties (属性)**」ボタンをクリックします (または、「**Selection (選択)**」メニューから「**Selection (選択)**」を選択します)。

リソースからの状況情報の取得

リソースを選択し、「**Status (状況)**」ボタンをクリックします (または、「**Selection (選択)**」メニューから「**Status (状況)**」を選択します)。

リソースの構成のコピー

リソースを選択し、「**Copy (コピー)**」ボタンをクリックします (または、「**Selection (選択)**」メニューから「**Copy (コピー)**」を選択します)。

このほか、「**Services (サービス)**」メニューからノードの特定の構成作業を選択する、「**Diagnostics (診断)**」メニューから (ドメインの) ログギングおよび (ノードの) トレースを制御する、「**Windows (ウィンドウ)**」メニューから項目の 1 つを選択することによってドメイン・リソースを表示または変更する、などの操作ができます。

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示しています。

項目に、それに関連した 1 つ以上の子項目がある場合は、その項目の横に「**Expand (拡張)**」ボタンまたは「**Contract (縮小)**」ボタンが表示されます。「**Expand (拡張)**」ボタンは、関連した子項目が隠れていることを示します。「**Expand (拡張)**」ボタンをクリックすると、子項目を表示できます。「**Contract (縮小)**」ボタンは、子項目が表示されていることを示します。「**Contract (縮小)**」ボタンをクリックすると、子項目を隠すことができます。項目の横にどちらのボタンも表示されていないならば、その項目に関連する子リソースはありません。

例えば、リンク・ステーションは特定のポートに関連付けられます。「Node (ノード)」ウィンドウの「**Connectivity (接続)**」ペインでは、このリンク・ステーションは、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・ステーションと共に表示されます。ポートは常に表示されますが、関連のリンク・ステ

ーションのリストを表示するか隠すかは選択できます。同様に、関連した LU のリストがあるリンク・ステーションは、拡張して LU を表示することも、縮小して LU を隠すこともできます。

親リソースは、常に子リソースより前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、その子リソースも削除されます。

ツールバー・ボタン

リソース・ウィンドウには、共通機能の実行に便利なツールバー・ボタンがあります。CS Linux 用のツールバーを図 15 に示します。



図 15. CS Linux ツールバー

各リソース・ウィンドウのツールバーにすべてのボタンが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では無効な場合（またはボタンを操作するには項目を選択する必要があるが、どの項目も選択されていない場合）は、そのようなボタンの輪郭がグレー表示され、機能を選択できません（ボタンを押すことができません）。リソース・ウィンドウに表示されるボタンは次のとおりです。



選択された項目を開始します。



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)」ウィンドウでは、リソースは選択されたペインに追加されます。）



選択された項目を削除します。



Motif 管理プログラムの使用

選択された項目の構成を表示または変更するために、その項目用のダイアログをオープンします。



選択され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を複製したフィールドが表示されたダイアログがオープンします。新規リソースを追加するには、ダイアログのフィールドを（新規項目の名前を入力して）完成します。



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、多くのリソースは、アクティブになっている間は変更できません。しかし、リソースを選択し「**Properties (属性)**」ボタンを押してダイアログをオープンすることによってアクティブ・リソースのパラメーターを表示したり、「**Status (状況)**」ボタンをクリックしてそのリソースの詳細な状況情報を表示することは可能です。

リソース・ダイアログ

リソース・ダイアログには、リソースの現行の構成情報が表示されます。タイプ 0 から 3 の LU のサンプル・ダイアログを 55 ページの図 16 に示します。

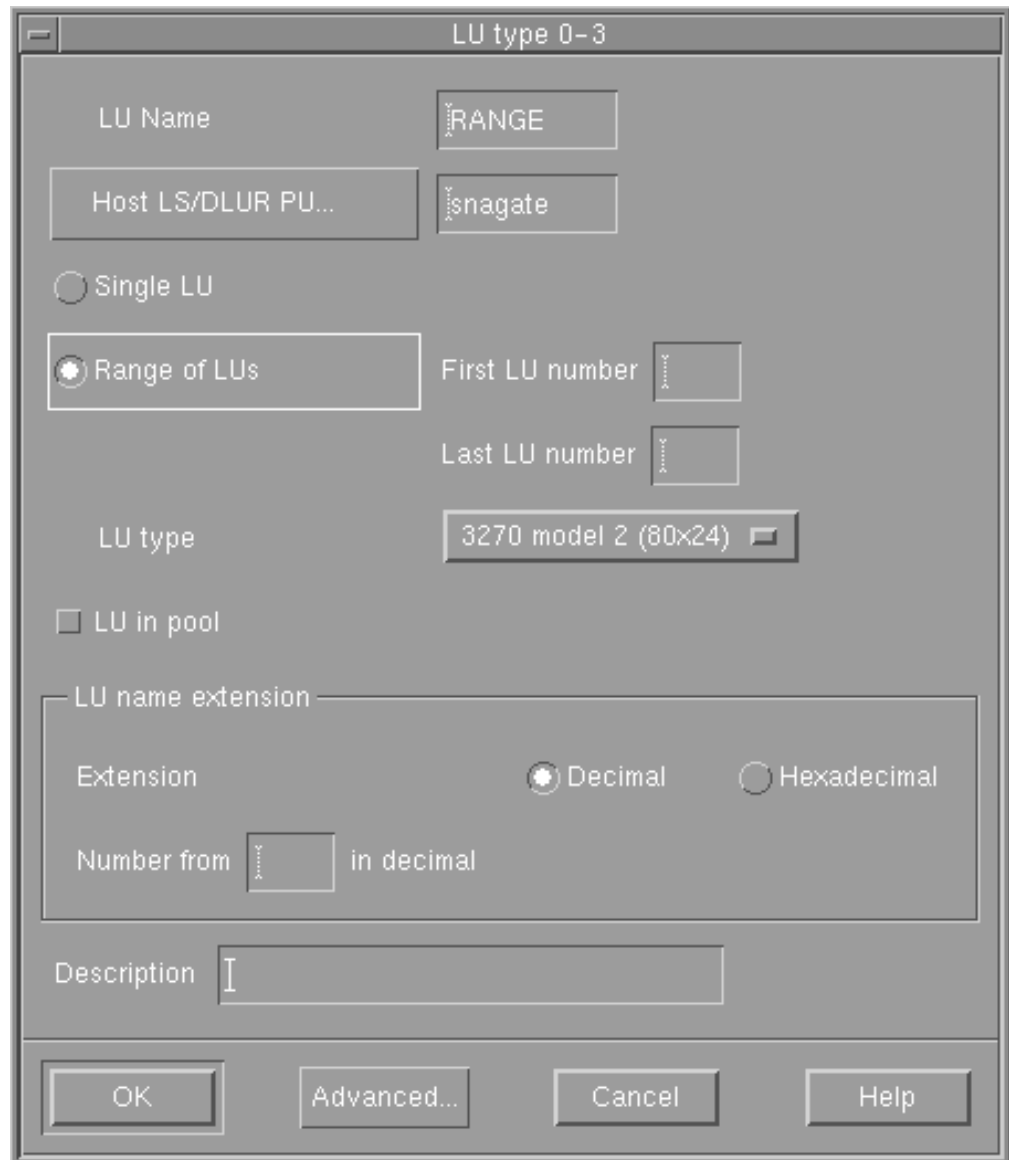


図 16. サンプル・ダイアログ

リソース・ダイアログでは、手順に沿って構成プロセスを進めることができ、デフォルト値がある場合はそれが表示されます。例えば、従属 LU を追加する場合、Motif 管理プログラムは、指定したリンク・ステーションで使用可能な LU 番号を「LU number (LU 番号)」フィールドに自動的に表示します。必須値を入力しなかった場合は、指定が必要な情報を示すメッセージ・ポップアップがプログラムから表示されます。

ほとんどのダイアログには、「Description (説明)」フィールドがあります。ここに入力した情報は、リソースが表示されるウィンドウに表示されます。

リソース・ダイアログの情報を変更する許可が与えられている場合 (新規の項目を追加したり、または既存の項目を変更する場合)、ダイアログには、「OK」ボタンおよび「Cancel (キャンセル)」ボタンがあります。「OK」ボタンを押して作業を完了するか、「Cancel (キャンセル)」ボタンを押して、リソースの構成を変更しないで終了します。

Motif 管理プログラムの使用

リソース・ダイアログの情報を変更できない場合 (例えば、リソースの構成がアクティブなときは変更できない場合)、ダイアログには、「OK」ボタンの代わりに、「Close (閉じる)」ボタンが含まれます。ダイアログの情報の表示を完了したときに、このボタンをクリックします。

ダイアログのコンテキスト・ヘルプを表示するには、「ヘルプ (Help)」ボタンをクリックします。

注: 基本 Motif ダイアログには、重要な構成フィールドだけが表示されます。CS Linux は、拡張フィールド用のデフォルト値を提供しています。拡張構成パラメーターにアクセスするには、「Advanced (拡張)」ボタンをクリックします。拡張ダイアログは基本パラメーターに入力した値に応じて変わるので、拡張パラメーターを調整する場合は、拡張ダイアログをオープンする前に基本ダイアログを完成する必要があります。拡張構成フィールドの詳細は、Motif 管理プログラムのオンライン・ヘルプを参照してください。

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、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。

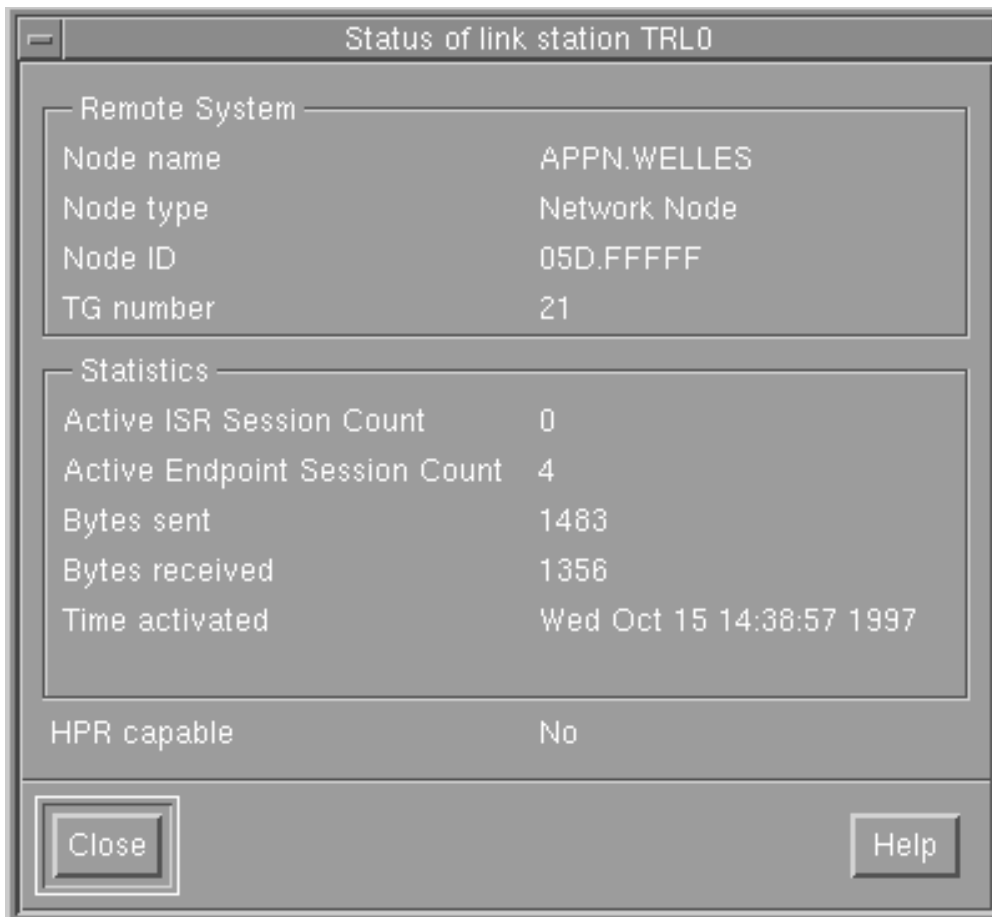


図 17. 状況ダイアログの例

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。

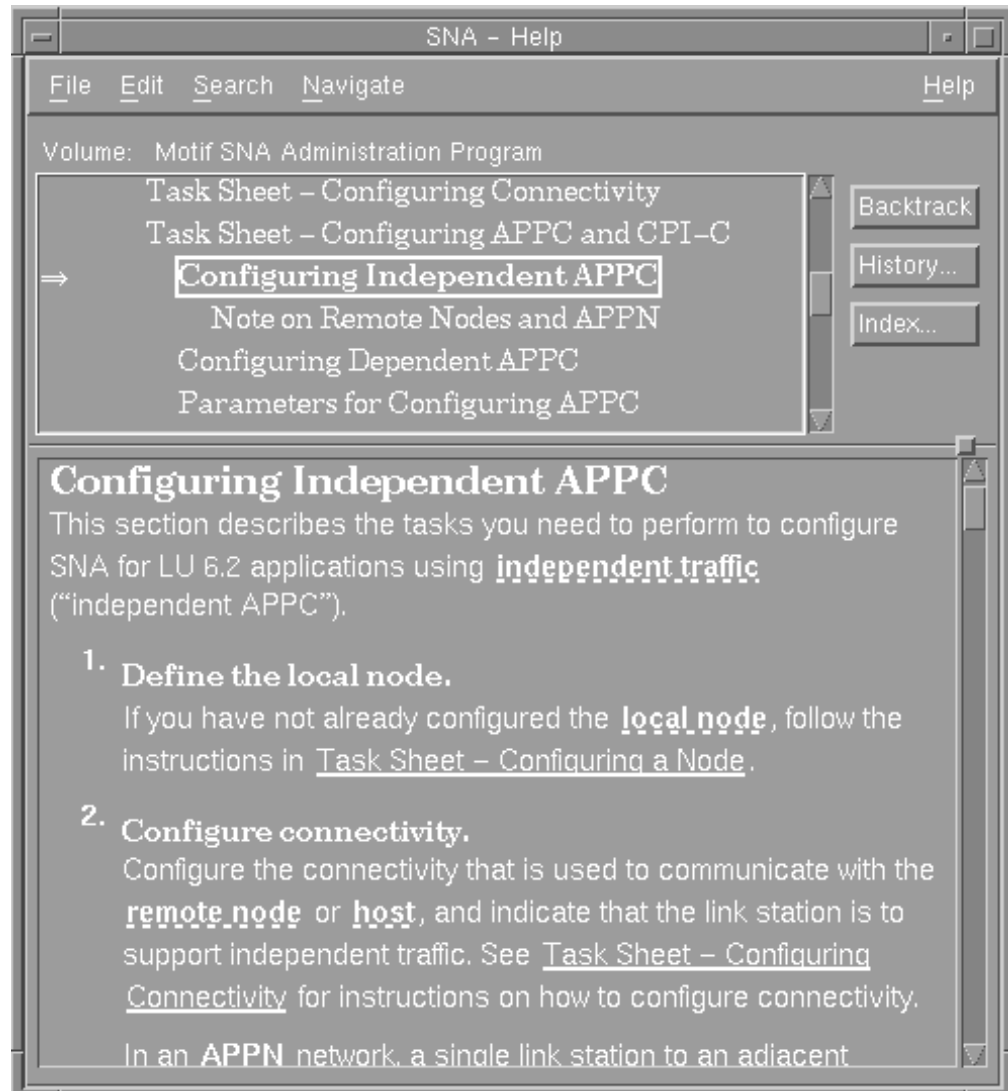


図 18. ヘルプ・ウィンドウの例

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

コマンド行構成では、すべての CS Linux 構成パラメーターを変更できます。これを使用すると、Motif 管理プログラムから使用可能ななどのリソースでも構成でき、Motif プログラムでは表示されない構成パラメーターも設定または変更できます。ただし、この管理方式では、一般に、Motif 管理の場合より提供しなければならない情報が多くなります。さらに、提供した情報が有効で既存のリソース定義との整合性があることを、確認する必要があります。(Motif 管理プログラムでは、入力したデータの整合性をプログラムが自動的に確認するので、このプログラムの使用をお勧めします。さらにこのプログラムでは、メニューやダイアログの選択項目に基づいて多くの構成値が推定され、明示された定義に基づく推奨値が表示されます。)

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** コマンドは次の形式で発行することができます。

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

snaadmin コマンド行管理のヘルプを表示するには、次のコマンドのいずれかを使用します。

- **snaadmin -h** を使用すると、コマンド行管理の基本ヘルプと、コマンド行ヘルプの使い方の説明が表示されます。
- **snaadmin -h -d** を使用すると、**snaadmin** プログラムに指定できるコマンドのリストが表示されます。
- **snaadmin -h *command*** を使用すると、指定した *command* についてのヘルプが表示されます。
- **snaadmin -h -d *command*** を使用すると、指定の *command* についての詳細なヘルプが、そのコマンドに指定できる構成パラメーターのリストと共に表示されます。

一部のコマンドは、コマンドに **-n** オプションを付けてサーバー名を指定すれば、IBM Remote API Client から実行できます。このコマンドは、指定したサーバーから実行した場合と同じ効果があります。

以下に、リソース・タイプ別の管理コマンドを要約して示します。リストされているコマンドのタイプの一部は次のとおりです。

status_*

リソースのタイプについての要約情報を提供します。

define_*

新しい **define_*** レコードを構成ファイル内に作成するか、または同じリソースの既存のレコードを新規定義で置換します。

delete_*

ファイルから対応する **define_*** レコードを除去します。

query_*

該当のコンポーネントの構成ファイルから情報を戻しますが、ファイルを変更することはありません。

set_* トレース・パラメーターやロギング・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 Deployment 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、マスター・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、ドメインのマスター・サーバーのみに保管され、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マスタ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マスター・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、マスター・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、マスターまたは第 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) がマスター・サーバーになります。現行のマスタ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、またはマスタ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マスターになります。

CS Linux がマスタ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: マスター・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。マスター・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。

クライアント/サーバー機能の構成

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、マスター・サーバーを削除する (リストの 2 番目にあるサーバーがマスタ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 (そのサーバーがマスタ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、マスター・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、使用可能なマスター・サーバーが少なくとも 1 つ必要です。

147 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

CS Linux をシステムに構成する場合の最初のステップは、ローカル・ノードの構成です。ノード構成は、ノードが APPN ネットワークに参加するために必要な基本情報を提供するものです。ノードの接続性または他のリソースを定義するには、まずノードを構成する必要があります。

ノードが既に構成済みの場合は、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、次のどちらかの方式を使用します。

Motif 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「**Configure node parameters (ノード・パラメーターの構成)**」を選択します。

コマンド行管理プログラム

define_node コマンドを発行します。

ノード構成用の拡張パラメーターにより、未定義のパートナー LU とのセッション、セキュリティー障害の報告、および限定リソース・タイムアウトを制御できます。

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。

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。

- 67 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。
- 91 ページの『第 6 章 APPC 通信の構成』または 119 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。
- 119 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 (および必要な場合は正常なイベント) 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場合は、まず最初にログ・ファイルを調べてください。ログ・メッセージには、問題の原因に関する情報とその処置が示されています。

CS Linux は、次のカテゴリーのイベントについてのメッセージをログに記録します。

問題 ユーザーが気付くような形でシステムを劣化させる異常なイベント (セッションの異常終了など)。

例外 システムを劣化させるがユーザーがすぐには気付かないような異常なイベント (リソース不足など)、またはシステムを劣化はさせないがあとで発生する例外または問題の原因を示す可能性のあるイベント (リモート・システムからの予期しないメッセージの受信など)。

監査 正常なイベント (セッションの開始など)。

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。

正常な状態に関するログとエラー状態に関するログを区別するために、カテゴリーの異なるメッセージは異なるログ・ファイルに記録されます。問題メッセージと例外メッセージは、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。このフィルターは、監査ログ、例外ログ、および問題ログのすべてのタイプのログに適用されます。ログのフィルター処理の詳細については、「*IBM Communications Server for Data Center Deployment on Linux 診断ガイド*」を参照してください。

CS Linux にも、ログ・ファイルが大きくなりすぎてディスク・リソースを消費してしまうのを防止するためのバックアップ手段があります。ログ・ファイルが許容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たあと、ログ・ファイルをクリアします。

デフォルトでは、CS Linux は次のログ・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)

使用量ログ・ファイル

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

`/var/opt/ibm/sna/bak.usage` (バックアップ)

ログ・ファイルの表示は、テキスト・エディター、またはその他の Linux システム・ユーティリティを使用します。

- vi** テキスト・エディターにファイルを表示します。これにより、ファイルの前方や後方に移動でき、特定のエントリーを検索できます。
- pg** ファイルを一度に 1 ページずつ表示します。このユーティリティーは、簡潔で使いやすいですが、小さなログ・ファイル向けです。
- tail** ファイルのテール (終わりの部分) を表示します。ファイルの終わりは、最新のログ・メッセージがあるところです。システムの実行中にログ・ファイルを監視するには、このユーティリティーに **-f** オプションを指定して使用します。

詳細ログでなく要約ログを選択した場合は、**snahelp** コマンドを使用して、特定のメッセージ番号についての原因および処置の情報を判別できます。

ほとんどの場合は、デフォルト設定のロギングで十分ですが、次のような変更を加えることができます。

- ログに記録するメッセージのカテゴリーを指示する。

問題メッセージは常にログに記録されます。問題メッセージを使用不可能にすることはできません。その他の 2 つのメッセージ・カテゴリーについては、ロギングは通常は使用できないようになっていますが、必要に応じて使用可能にすることができます。

- メッセージをログに記録する場合の詳細レベルを指定する。
- ドメインのセントラル・ロギングか、ノードごとのローカル・ロギングかを指定する。
- ログ・ファイルの名前とサイズを変更する。

ロギングを構成するには、次のどちらかの方式を使用します。

Motif 管理プログラム

「Node (ノード)」ウィンドウまたは「Domain (ドメイン)」ウィンドウの「**Diagnostics (診断)**」メニューから「**Logging (ロギング)**」を選択します。

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。

- **set_central_logging**
- **set_global_log_type**
- **set_log_type**
- **set_log_file**

Motif 管理プログラムの中の「Logging (ロギング)」ダイアログは、ドメイン全体のログ設定に影響します。コマンド行を使用して、特定のマシンに対するローカル・ログ設定値を構成することにより、ドメイン設定値を指定変更できます。

Motif 管理プログラムは、ロギングを制御できるほか、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を使用すると、ロギングとトレースについては、コマンド行インターフェースの方がきめ細かい制御ができます。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 Deployment 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、68 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、68 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)、73 ページの『リンク・ステーションの定義』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 を使用してホストと通信する場合、81 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 APPC を使用する場合

CPI-C または APPC 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

接続コンポーネントの定義

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 を使用した 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ので、ユーザーが構成する必要はありません。ポートを定義するときに、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 APPC 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 DLUS 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、通信リンクのローカル側の終端をネットワーク内の固有のアクセス・ポイントとして表したものです。各ポートは、特定のリンク・プロトコルに関連付けられます。リンク・プロトコルは次のどれでも構いません。

- SDLC
- 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・チャネル (MPC)(CS Linux on System z のみ)
- Enterprise Extender (HPR/IP)

特定のリンク・プロトコルを使用するポートを 2 つ以上構成できます。一般に、1 つのポートはアダプター・カードなどのような 1 つの物理アクセス・ポイントに対応していますが、一部のリンク・プロトコル (トークンリングなど) では、1 つのアダプターについて複数のポートを定義できる場合があります。この場合、個々のポートはアドレス (SAP 番号など) により区別されます。

Motif 管理プログラムを使用して特定リンク・プロトコル用のポートを定義すると、CS Linux は、そのポートに必要な DLC がまだ定義されていないタイプのものであれば、そのポート用の DLC を自動的に定義します。コマンド行構成の場合は、ポートと DLC をそれぞれ異なるコマンドを使用して定義しなければなりません。

トークンリング・リンク・プロトコルを使用している APPN ネットワークでは、「SAP Configuration (SAP 構成)」ダイアログを使用して、ポートを接続ネットワークの一部にすることを指示できます。

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す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 をサポートすることができます。ポートがアクティブになっている間に暗黙 PU フィールド

ドが変更された場合、その変更は、変更後に生成されるすべての暗黙リンク・ステーションのインスタンスに影響を与えます。

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

Motif 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「Connectivity (接続)」および「New port (新規ポート)」を選択します。

コマンド行管理プログラム

DLC を構成するには、次のコマンドを使用します。

define_type_dlc

ポートを構成するには、次のコマンドを使用します。

define_type_port

これらのコマンドにおいて、*type* はリンク・プロトコルのタイプ (sd1c、tr、ethernet、qllc、mpc、ip) を示します。

接続ネットワークを構成するには、次のコマンドを使用します。

define_cn

拡張ポート構成パラメーターを使用すると、BTU サイズ、許容アクティブ・リンク数、暗黙ダウストリーム LU の生成、および動的リンク・ステーションの設定を制御できます。

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。)

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。

DLC、ポート、および接続ネットワークの定義

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。

Secondary

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・

DLC、ポート、および接続ネットワークの定義

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。

Primary

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。

Negotiable

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お問い合わせください。

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 を使用する必要がある場合のみにしてください。OSA-Express アダプターの場合、ローカル SAP 番号は、この Linux イメージの ethX インターフェースに対応する入出力装置の OSA/SF に定義された番号と一致する必要があります。

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お問い合わせください。

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。

DLC、ポート、および接続ネットワークの定義

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

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

Local IP interface

これはオプション・フィールドです。複数の IP ネットワークにアクセスできる場合は、ポートに使用するローカル IP ネットワークをこのフィールドに指定できます。1 つの IP ネットワークにのみアクセスできる場合は、このフィールドはブランクのままにできます。

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 または en0 など)。
- 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)。

インターフェース ID を識別するためには、カードが取り付けられたサーバー上で、**ipconfig -a** のコマンドを実行します。これにより、インターフェース ID および関連する IP アドレスがリストされます。

Protocol

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

Define on connection network

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。ポート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。

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 (131 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。

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、『リンク・ステーションの定義』を参照してください。
- DLUR PU を定義するには、81 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、91 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・ステーションの定義

SNA ネットワーク内の他のノードと通信するには、SNA ネットワーク内の隣接ノードまでのリンク・ステーション (LS) の特性を構成する必要があります。リンク・ステーションを定義する前に、使用中のアダプター (およびリンク・プロトコル) のためのポートを定義しておく必要があります。リンク・ステーションの構成に必要な情報のほとんどは、使用中のプロトコルに関係なく同じです。

リンク・ステーションは、CS Linux ローカル・ノードとリモート・コンピューターの間を 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・コンピューターは次のいずれでも構いません。

- ホスト・コンピューター。この場合、CS Linux は、3270 通信または LUA 通信を使用してホスト・プログラムにアクセスします (または、プログラム間通信用の 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・コンピューターは対等のパートナーとして通信します (APPN ネットワークの代表的な配置)。

リンク・ステーションの定義

- ・ ダウンストリーム・コンピューター。この場合、ホストへアクセスするために、CS Linux SNA ゲートウェイ機能または DLUR 機能を使用します。

リンク・ステーションは特定のポートに関連付けられます。各ポートには 1 つまたは複数のリンク・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、関連した物理装置 (physical unit: PU) があります。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 PU を分離したリソースとして取り扱いません。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環として開始および停止されます。リンク・ステーションは、「Node (ノード)」ウィンドウの接続セクションに表示されます。つまり、PU は任意のウィンドウには表示されません。

注: ほとんどの状況では、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、ダウンストリーム SNA ゲートウェイまたは APPC トラフィックに対してのみ動的に作成されるリンク・ステーションを使用する場合、リンクが常にリモート・ノードからアクティブになるため、リンク・ステーションを明示的に構成する必要はありません。

リモート・ノードがローカル・ノードに接続しようとしても、着呼に指定されている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かった場合、ローカル・ノードに適切なポートが定義されていれば、CS Linux が暗黙にリンク・ステーションを定義します。この動的に作成されるリンク・ステーションは、接続が維持されている期間中は、「Node (ノード)」ウィンドウの接続セクションに表示されています。

リンク・ステーションを構成するには、次のどちらかの方式を使用します。

Motif 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「**Connectivity (接続)**」および「**New link station (新規リンク・ステーション)**」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_type_ls
```

これらのコマンドにおいて、*type* はリンク・プロトコルのタイプ (sdlc、tr、ethernet、qllc、mpc、ip) を示します。

リンク・ステーション用の拡張パラメーターを使用すると、伝送特性、XID 交換、オプションのリンク機能、リンクを使用する LU 0 から 3 セッションの圧縮、および再活動化プロシーチャーをより高度に制御できます。

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、『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、78 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、79 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、それ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

リンク・ステーションの定義

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問い合わせてください。

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、78 ページの『独立 LU トラフィック用のパラメーター』および 79 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、78 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、79 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ければなりません。

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 は 1 から最大 1024 の数です。パーマネント・バーチャル・サーキットが 1 つだけの場合は、そのチャンネル ID はおそらく 1 になります。

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です。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 (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)。
- 別名 (newbox など)。

名前または別名を指定する場合は、Linux システムは、(ローカル TCP/IP の構成またはドメイン・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、それを指定の方が賢明です。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。

リンク・ステーションの定義

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、『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、『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、121 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、119 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、91 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、別のノードを経由する間接接続のみしかない場合があります。このような間接接続ノードの LU からは、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。

ここでは、ホスト・コンピューターへの接続を提供する DLUR PU の構成方法を説明します。DLUR PU を構成することによって、ローカル・ノードが DLUR サービスを提供できるようになります。

APPN ノード (CS Linux を実行中のノードなど) の DLUR は、ホストの従属 LU サーバー (DLUS) と協力して働いて、DLUR ノードの従属 LU から、APPN ネット

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。LU のロケーションになるノードでは DLUR を使用可能にし、ホスト・ノードでは DLUS を使用可能にしておく必要がありますが、セッション経路の中間ノードには DLUR は不要です。

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。そのような 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。

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・ノードに関連した 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。

DLUR PU を構成するには、次のどちらかの方式を使用します。

Motif 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「**Connectivity (接続)**」および「**New DLUR PU (新規 DLUR PU)**」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_internal_pu
```

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、ホスト上で構成された PU 名に一致する必要があります。(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ます。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。

DLUR PU の定義

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、134 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、121 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、119 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、91 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。

- 62 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。
- 67 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。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。詳細は、131 ページの『SNA ゲートウェイの構成』および 81 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 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 タイプ 0 から 3 の定義

LU name

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問い合わせてください。

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。

指定した値に応じて、CS Linux は、標準 VTAM テーブルで使用される値に一致するように、以下のストリングの 1 つを DDDLU NMVT 内のホストに送信します。

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、217 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、『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、121 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 の場合は、LU プールを定義することによって、ユーザーが行う構成作業を単純化し、ホスト・セッションを確立する際の柔軟性を高めることができます。

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

- ユーザーのセッションをプール内にある特定の 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

LU プールの定義

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。

LU プールが複数の CS Linux サーバーにわたっていても構いません。このためには、単にそれぞれのサーバー上で同一の名前を使用して LU プールを定義するだけです。サーバーに障害が起こった場合、またはサービス休止状態になった場合は、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の構成を単純化でき、別のサーバーを追加したり、既存のサーバーに LU を追加したりして容量を簡単に増やすことができますようになります。

「LU Pools (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 プールを表示できます。このウィンドウには、システム内に構成されている LU プールが表示され、LU プールに追加する LU を選択することができます。LU プールの下には、その LU プール内の個々の LU が表示されます。

LU は次のように識別されます。

- 3270 display LU (3270 ディスプレイ LU)
- Unrestricted LU (非制限 LU)
- SCS Printer (SCS プリンター)
- 3270 Printer (3270 プリンター)

タイプの異なる LU を同じプール内に混在させてはなりません。(例えば、ディスプレイ LU とプリンター LU を同じプールに入れてはなりません。) TN3270E クライアントをサポートする場合以外は、プリンター LU のプールが必要になることはほとんどありません。

LU プールを構成するには、次のどちらかの方式を使用します。

Motif 管理プログラム

「Node (ノード)」ウィンドウの「**Windows (ウィンドウ)**」メニューから「**LU Pools (LU プール)**」を選択し、「**New (新規)**」を選択してプールを追加します。

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_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。

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。既存のプールの名前を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。

LU プールの定義

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 アプリケーション用として定義するリソースは、APPC アプリケーションの場合と同じです。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、112 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。

- 62 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。
- 67 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。

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. 92 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. 94 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. 96 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. 100 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. 107 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. 112 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. 115 ページの『APPC セキュリティーの構成』の説明に従って、APPC セキュリティーを定義します。
8. 5250 通信を構成するには、119 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。詳しくは、Remote API Client on AIX or Linux については 180 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ディングされた LU 別名と、サーバーのドメインの LU 別名のマッピング』を、Windows クライアントについては 172 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ディングされた LU 別名と、サーバーのドメインの LU 別名のマッピング』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 として、そのノードを構成したときに自動的に定義されている制御点 LU を使用できます。これはデフォルトの LU で、アプリケーションは、特定の LU が指定されていない場合にこの LU を使用できます。アプリケーションがデフォルトの 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。これについては、使用する APPC アプリケーションの資料を調べるか、またはアプリケーション・プログラマーに問い合わせてください。

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 を構成する場合は、その LU をデフォルト・プールのメンバーとして定義しておくとな便利な場合があります。このようにしておく、特定のローカル LU が指定されていないアプリケーションには、デフォルト LU として定義されている LU のプールから、未使用の LU のいずれかが割り当てられます。

従属 LU 6.2 をデフォルト LU として定義できます (また、複数のノード上でデフォルト LU を定義できます)。デフォルト LU を要求するアプリケーションは、使用可能になっているこれらの LU のどれにでも割り当てることができます。LU がアプリケーションと同じコンピューター上になくても構いません。ただし、アプリケーションのパートナー LU を定義する場合は、デフォルト LU が定義されているすべてのノード上でパートナー LU を定義する必要があります。これにより、アプリケーションはいずれかのノード上で定義されたデフォルト・ローカル LU を使用して、適切なパートナー LU と通信できます。

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、それぞれ 1 つのローカル LU と 1 つのパートナー LU が関与します。ローカル LU には、ノード制御点に関連付けられている定義済みのデフォルト LU を使用することもでき、また新しいローカル LU を構成することもできます。パートナー LU は、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・ノードである場合は、構成する必要はありません。APPN は、パートナー LU を動的に検出できます。ただし、ネットワークが APPN ネットワークでないか、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。その場合は、まずパートナー LU があるリモート・ノードを構成し、そのリモート・ノード上にパートナー LU を定義します。(パートナー LU がリモート・ノード上のデフォルトの LU である場合は、パートナー LU はリモート・ノードを定義した時点で自動的に追加されるので、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、次のどちらかの方式を使用します。

Motif 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「**APPC**」を選択し、次に「**New independent local LU (新規独立ローカル LU)**」または「**New dependent local LU (新規従属ローカル LU)**」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_local_lu

拡張ダイアログを使用すれば、同期点サポート、接続ルーティング特性、パスワード置換を使用するかどうか、SSCP アクセスに関する制約事項、LU に関連したシステム名、およびセキュリティーを指定できます。

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 name

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お問い合わせください。

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。)

ローカル LU の定義

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。

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、『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、96 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、100 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、107 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、112 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、115 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、119 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

次の場合は、リモート・ノード (そしてその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 を使用した通信の相手方となるすべてのパートナー LU を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・ノードを定義することで、パートナー LU の検索ができるようになります。
- リモート・ノードが LEN ノードであり、ローカル・ノードが、その LEN ノードのネットワーク・ノード・サーバーとして働くネットワーク・ノードである場合は、その LEN ノード (およびそのノードのパートナー LU) を、ネットワーク・ノード・サーバー上でリモート・ノードとして定義する必要があります。この定義により、APPN ネットワーク内の他のノードが、LEN ノード上の LU を見つけることができるようになります。
- リモート・ノードが他の APPN ネットワーク内にある場合は、動的に見つけることができないため、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・ステーションの定義時にその定義をしていない場合は、そのリンクを介して APPC 通信を使用する前に、その定義をしておく必要があります。

リモート・ノード定義を追加すると、そのリモート・ノードと同じ名前のパートナー LU が自動的に追加されます。これが、そのリモート・ノードの制御点 LU になります。使用中のアプリケーションがこのパートナー LU を使用する場合は、新しくパートナー LU を追加する必要はありませんが、そのパートナー LU の LU 別名を追加しておくると便利な場合があります。別名を追加するには、「Partner LU Configuration (パートナー LU 構成)」ダイアログで、該当のパートナー LU をダブルクリックし、別名を入力します。

ローカル・ノードとリモート・ノードが両方ともエンド・ノードまたはネットワーク・ノードであり、どちらも同じ APPN ネットワークに所属している場合は、パートナー LU は必要に応じて動的に探索されます。この場合は、LU 探索のためのリモート・ノードを定義しないでください。このノードを定義すると、LU を動的に探索するための APPN 内のプロトコルが誤動作することがあります。

この誤動作を防ぐために、CS Linux では、アクティブな CP-CP セッションの相手方になるリモート・ノード (または過去にアクティブな CP-CP セッションの相手方になったことのあるリモート・ノード) を、ユーザーが定義できないようになっています。さらに、ユーザーがあるリモート・ノードを既に定義してあり、CS Linux がそのリモート・ノードとの CP-CP セッションを確立する場合は、既存のエントリーが一時的に動的なエントリーに変換されます。この障害を修正するために、このリモート・ノードが非アクティブ状態にあるときに、リモート・ノード定義を削除するようにしてください。

リモート・ノードを構成するには、次のどちらかの方式を使用します。

Motif 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、「APPC」および「New remote node (新規リモート・ノード)」を選択します。

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、『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、100 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、107 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、112 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、115 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、119 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・ノードの場合、アプリケーションがパートナー LU を参照する LU 名を使用しているときは、パートナー LU を定義する必要はありません。パートナー LU は、APPN を使用して動的に見つけられます。ただし、アプリケーションがパートナー LU を参照する LU 別名を使用している場合は、パートナー LU 別名の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・ノードのどちらかが LEN ノードである場合は、パートナー LU をリモート・ノードの子として定義する必要があります。LEN ノードは LU のダイナミック検索に加わることはできないからです。使用中のアプリケーションが、リモート・ノードの制御点 LU をパートナー LU として使用する場合は、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ています。

同じリモート・ノードに所属していて、同じ文字群で始まる名前を持つ複数のパートナー LU を構成するには、ワイルドカードを使用できます。ワイルドカードを使用すれば、各パートナー LU を個別に構成する必要はなくなります。

パートナー LU を構成するには、次のどちらかの方式を使用します。

Motif 管理プログラム

Motif 管理プログラムでは、パートナー LU 別名を追加したり、特定のリモート・ノードにパートナー LU の定義を追加したり、ワイルドカードを使用して複数のパートナー LU を定義したりできます。「Node (ノード)」ウィンドウの「Services (サービス)」メニューから、「APPC」、「New partner LUs (新規パートナー LU)」、および次のいずれかを選択します。

- 「Partner LU alias (パートナー LU 別名)」
- 「Partner LU on remote node (リモート・ノード上のパートナー LU)」
- 「Wildcard partner LU on remote node (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、次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・エンタリーは、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・エンタリーは、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。

パートナー LU の定義

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 用のリンク・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すると、パートナー LU 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 の位置を識別できます。

注:

1. リソースを動的に配置できる APPN ネットワーク内では、リンク・ステーション・ルーティングは必要ありません。通常の APPN ルーティング・メカニズムがバイパスされるため、APPN ネットワーク内ではリンク・ステーション・ルーティングの使用はお勧めしません。
2. Enterprise Extender (HPR/IP) リンク・ステーションに対して、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・タイプのトラフィックはすべて RTP 接続を経由する必要があるからです。この接続は特定のリンク・ステーションに固定されず、別のパスに切り替わる可能性があります。

パートナー LU 用のリンク・ステーション・ルーティングを構成するには、次のどちらかの方式を使用します。

Motif 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、「**APPC**」、「**New partner LUs (新規パートナー LU)**」、および「**Partner LU on link station (リンク・ステーション上のパートナー LU)**」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_ls_routing
```

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 name

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

パートナー LU の定義

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 名をワイルドカードとして使用)」

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、『TP の定義』を参照してください。
- モードを定義するには、107 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、112 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、115 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、119 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

ここでは、APPC TP を定義する方法を説明します。

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。ただし、次の場合は、TP 定義を構成する必要があります。

APPC 特性

CS Linux コンピューター上の TP が呼び出し側 TP (ソース TP、つまり APPC 会話を開始する TP) である場合は、TP へのアクセスを制限する必要はなく、TP を定義する必要はありません。ただし、106 ページの『TP 定義パラメーター』の説明に従って APPC TP を定義することにより、次の特性を指定できます。

- TP の会話セキュリティを定義する。
- TP が基本会話またはマップ式会話のどちらを使用するかを指示する。
- 同期点処理を指定する。
- PIP データの取り扱いを指定する。

呼び出し可能 TP

着信割り振り要求にこたえて自動的に TP を開始できるようにするには、102 ページの『サーバー上での TP 呼び出しパラメーター』の説明に従って、その TP を呼び出し可能 TP として定義します。

呼び出し可能 TP (またはターゲット TP) は、着信割り振り要求に応じて開始される TP です。呼び出し可能 TP については、TP 定義を作成する必要

があります。呼び出し可能 TP になることができるのは、RECEIVE_ALLOCATE を発行する APPC TP、または Accept_Conversation または Accept_Incoming を発行する CPI-C アプリケーションです。

注: 本書で使用する「Receive_Allocate」という語句は、これら 3 つの API 呼び出しのいずれかを示します。

実行中の TP に着信割り振り要求を送るための呼び出し可能 TP を定義することもできます。

呼び出し可能 TP については、タイムアウト値を指定して、割り振り要求を待つ時間を制限することもできます。(このオプションを構成するには、コマンド行管理プログラムを使用する必要があります。)

CS Linux は、呼び出し可能 TP 定義を次の目的に使用します。

- TP が Receive_Allocate を発行すると、CS Linux は、該当の TP 名を持つ呼び出し可能 TP 定義を検索します。その定義が存在し、それに Receive_Allocate タイムアウトの値が含まれている場合は、CS Linux は、Receive_Allocate を処理するときその値を使用します。タイムアウト値がない場合は、CS/AIX は、デフォルト (タイムアウトなし、つまり TP は無期限に待機する) を使用します。
- ターゲット・システムに着呼割り振り要求が到着したときに、要求された TP が、まだ、未処理の Receive_Allocate を持った状態で実行されていない場合は、CS Linux は、着呼割り振り要求に指定されている TP 名を持つ TP 定義を検索します。その定義が存在する場合は、CS Linux はその定義の中の情報を使用して、その TP を開始するか (複数のインスタンスが許容されているか、またはその TP がまだ実行されていない場合)、または、着呼割り振りをキューに入れるべきであると判断します (TP が既に実行中で、しかも複数のインスタンスが許容されていない場合)。

必要なら、同じ TP について両方のタイプの定義を構成することもできます (例えば、呼び出し可能 TP について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、次のどちらかの方式を使用します。

APPC 特性を定義する場合

次のいずれかの方式を使用します。

Motif 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、「**APPC**」および「**Transaction Programs (トランザクション・プログラム)**」を選択します。CS Linux が TP ウィンドウを表示したら、下部のペインを選択して「**New (新規)**」ボタンをクリックするか、既存の TP 定義を選択して、「**Properties (プロパティ)**」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` コマンドを発行します。

呼び出し可能 TP を定義する場合

サーバーとクライアントの構成方式は異なります。

- サーバーの場合は、次のどちらかの方式を使用します。

Motif 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、「**APPC**」および「**Transaction Programs (トランザクション・プログラム)**」を選択します。CS Linux が TP ウィンドウを表示したら、上部のペインを選択して「**New (新規)**」ボタンをクリックするか、既存の呼び出し可能 TP 定義を選択して、「**Properties (プロパティ)**」ボタンをクリックします。

コマンド行管理

snatpinstall コマンドを発行します。

•

UNIX

AIX または Linux 上の IBM Remote API Client で、**snatpinstall** コマンドを発行します。

•

WINDOWS

Windows クライアントの場合は、クライアント・ソフトウェアがインストールされたディレクトリーに移動して、**tpinst32** コマンドを実行します。(このコマンドは 32 ビット・バージョンと x64 バージョンの両方の Windows に適用されます。)

snatpinstall または **tpinst32** コマンドの使用に関する情報は、207 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。

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、207 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おい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・ディレクトリの中で開始されます。このホーム・ディレクトリは、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・ディレクトリを基準として相対的に指定されたディレクトリの中で、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、「ユーザー ID (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が必要です。

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 `APPCLU` 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。確認同期については詳しくは、「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

モードは、ローカル LU (LU タイプ 6.2) がパートナー LU との通信に使用する一連の特性を指定します。これらの特性には、2 つの LU 間でのデータの伝送方法 (最大 RU 長やペーシング・ウィンドウ・サイズなど) に関する情報、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組み込まれています。

さらに、LU 間の通信パスに関する要件も指定する必要があります。これには、特定レベルのネットワーク・セキュリティーの適用、伝送時間の最小化、または、高費用の通信リンクの使用の回避などがあります。これらの要件を定義するには、サービスのクラス名 (COS) を使用できます。サービスのクラス名は、伝送時間、伝送コスト、およびネットワーク・セキュリティーなどの特性に対する、最大許容値および最小許容値を指定します。さらに、COS は、これらの値の各範囲に対応した重み付けも指定します。これによって、ノードは、同じリモート LU に到達するために使用可能な経路が 2 つ以上ある場合に、ネットワークでの最適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定義である場合は、各モードの定義に、そのモードに必要な COS の名前が含まれています。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、モードに COS を関連付ける必要はありません。この場合は、COS 名は動的に決定されます。

SNA では、ほとんどのシステムの要件を満たす多数の標準モードおよび関連の COS が定義されています。通常は、追加のモードおよび COS を定義する必要はあ

モードおよびサービス・クラスの定義

りません。モードの定義が必要になるのは、定義済みの標準モード（「Mode (モード)」ウィンドウに表示されます) のいずれにも該当しないモードが必要な場合のみです。

着呼会話に示されているモード名が認識不能のものである場合は、デフォルトのモードが使用されます。デフォルト・モードを指定しなかった場合は、デフォルト・モードはブランクのモード名となります。

表 2 に、標準のモード名とそれぞれに関連した COS 名を示します。これらの標準名に関連したパラメーターの詳細は、IBM SNA 関連資料の「LU 6.2 Reference—Peer Protocols (モードについて)」、および「APPN Architecture Reference」(COS について) を参照してください。

表 2. 標準モード名と COS 名

モード名	関連の COS 名	目的
(ブランク)	#CONNECT	モード名の指定のないセッション (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	バッチ処理アプリケーションで、最低レベルのルーティング・セキュリティーにより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し、最低レベルのルーティング・セキュリティーを使用するセッション
#INTER	#INTER	対話式アプリケーションで使用されるセッション
#INTERSC	#INTERSC	対話型アプリケーションで、最低レベルのルーティング・セキュリティーにより使用されるセッション
#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し、最低レベルのルーティング・セキュリティーを使用するセッション
SNASVCMG	SNASVCMG	CNOS (セッション数変更) 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 リクエスト (DLUR) 用に使用される CP-CP セッション
QPCSUPP	#CONNECT	5250 エミュレーション用に使用されるセッション

構成済みのモードは、ローカル LU とパートナー LU の間のセッションを開始するために、任意の APPC または CPI-C アプリケーションで使用できます。APPC アプリケーションでは、使用するモードを指定する必要がありますが、CPI-C アプリケーションでは、CPI-C サイド情報 (これにモード名が含まれています) を使用できます。CPI-C サイド情報の構成方法の詳細は、112 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、次のどちらかの方式を使用します。

Motif 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「**APPC**」および「**Modes (モード)**」を選択し、次に「**Mode (モード)**」ウィンドウで「**New (新規)**」を選択します。

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問い合わせてください。

Session limits

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、一対の 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

コンテンツ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問い合わせてください。

コンテンツン勝者セッションの最小数とコンテンツ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

コンテンツ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問い合わせてください。

コンテンツン勝者セッションの最小数とコンテンツ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。

Receive pacing window

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) の初期設定数です。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必要があります。

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。

モードおよびサービス・クラスの定義

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、『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、115 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、119 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 シンボリック宛先名を使用する 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、パートナー TP、パートナー LU、モード、および会話のセキュリティに関する情報に、シンボリック宛先名を関連付けるためのものです。

CPI-C のシンボリック宛先名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)。

CPI-C サイド情報を構成するには、次のどちらかの方式を使用します。

Motif 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、「**APPC**」および「**CPI-C**」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_cplic_side_info
```

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、『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、119 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・セキュリティの構成 (『セッション・セキュリティの構成』)
- 会話セキュリティの構成 (116 ページの『会話セキュリティの構成』)
- セキュリティー・アクセス・リストの構成 (116 ページの『セキュリティ・アクセス・リストの構成』)

セッション・セキュリティの構成

セッション・レベル・セキュリティは、LU-LU セッションの妥当性検査のために使用します。各定義は、ローカル LU 名、パートナー LU 名、およびパスワードから成っています。

CS Linux は、パスワードを使用して、ローカル LU とパートナー LU 間のセッションの妥当性を検査します。(このパスワードは Linux ログオン・パスワードには関係ありません。)

セッション・セキュリティを構成するには、次のどちらかの方式を使用します。

Motif 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、「**APPC**」、「**Security (セキュリティ)**」、および「**Session-level security (セッション・レベル・セキュリティ)**」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_lu_lu_password
```

セッション・セキュリティ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、『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、119 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、着呼会話の妥当性検査のために使用します。各定義は、ユーザー ID とパスワードから成っています。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。

会話セキュリティを構成するには、次のどちらかの方式を使用します。

Motif 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、「APPC」、「**Security (セキュリティ)**」、および「**Conversation-level security (会話レベル・セキュリティ)**」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_userid_password
```

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、119 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・アクセス・リストを定義することにより、LU または TP (またはその両方) へのアクセスを制御できます。このリストは、APPC ローカル LU または TP 用に定義により参照できます。

セキュリティ・アクセス・リストを構成するには、次のどちらかの方式を使用します。

Motif 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、「APPC」、「**Security (セキュリティ)**」、および「**Conversation-level**

security (会話レベル・セキュリティー)」を選択し、次に「Security Access Lists (セキュリティー・アクセス・リスト)」ペイン、続いて「**New (新規)**」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_security_access_list
```

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 (100 ページの『TP の定義』)

APPC セキュリティーの構成

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 of 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。

- 62 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。
- 67 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。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。詳細は、131 ページの『SNA ゲートウェイの構成』および 81 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、85 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、87 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)、92 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。

ユーザー・アプリケーションの定義

- b. ローカル・ノードが LEN ノードである場合は、94 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. 85 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、87 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 を使用するように割り当てることができます。詳しくは、Remote API Client on AIX or Linux については 180 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ディングされた LU 別名と、サーバーのドメインの LU 別名のマッピング』を、Windows クライアントについては 172 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ディングされた LU 別名と、サーバーのドメインの LU 別名のマッピング』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、『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、127 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 (131 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 (68 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での従属 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、134 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。

- 62 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。
- 67 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・ノードとホストの間で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、85 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、87 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 サーバーの構成

- 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 (『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せん (126 ページの『TN サーバー・アソシエーション・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必要があります。

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、135 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ー情報』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、どの LU を使用すべきかを示します。各アクセス・レコードは、TN サーバーへのアクセスが許可されている 1 つの TN3270 クライアント、そのクライアントが接続する TCP/IP ポート、そしてそのクライアントが使用する LU または LU プールを特定します。

任意の TN3270 クライアントが (すべてのクライアントが同じ LU または LU プールを使用して) アクセスできるデフォルト・レコードも定義できます。

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のときにのみ 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、次のどちらかの方式を使用します。

Motif 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「**TN server (TN サーバー)**」を選択し、次にサブメニューから「**TN server (TN サーバー)**」を選択します。その次に表示されるウィンドウで、「TN Server Client Access Permissions (TN サーバー・クライアント・アクセス許可)」ペインを選択し、「**New (新規)**」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_tn3270_access

注: コマンド行管理プログラム **snaadmin** または NOF アプリケーションで TN サーバー・アクセス・レコードを定義する場合は、*listen_local_address* パラメーターを使用して、TN3270 クライアントが接続されるローカル TN サーバー・コンピュータ上のアドレスを指定できます。このパラメーターを使用した場合、アクセス・レコードは Motif 管理プログラムでは表示されないため、アクセス・レコードの表示または管理にはこのプログラムは使用できません。しかし、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管理を行うことはできます。

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

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では、**hostname** コマンドを使用すると、そのコンピュータの 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。アドレスは次のいずれかです。

- 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)。

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。

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、Telnet クライアントは、ノードがアクティブであるときに **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。

証明書が有効であることの検査に加えて、TN サーバーは証明書を外部 LDAP サーバー上の証明書取り消しリストと突き合わせて、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 のセキュリティー・レベルを指定します。セッションは、クライアントとサーバーの双方がサポートできる最高のセキュリティー・レベルを使用します。クライアントが、要求されたレベル以上のセキュリティー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、暗号化は使用されません。このオプションは通常、クライアントがセキュア・イントラネットへ接続する際に、暗号化のオーバーヘッドを回避するために使用されます。

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す)。

40 Bit Minimum

クライアントは少なくとも 40 ビット暗号化をサポートしている必要があります。

56 Bit Minimum

クライアントは少なくとも 56 ビット暗号化をサポートしている必要があります。

128 Bit Minimum

クライアントは少なくとも 128 ビット暗号化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 ビット暗号化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 ビット暗号化をサポートしている必要があります。

注: 暗号化を使用するには、CS Linux に追加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。ユーザーの場所によっては、暗号化レベルのサポートに必要なソフトウェアがユーザーの国では使用できないために、リストされているすべての暗号化レベルを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。

- 『TN サーバー・アソシエーション・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・アソシエーション・レコードの構成

TN サーバー・アソシエーション・レコードは、プリンター LU とディスプレイ LU の間のアソシエーションを定義して、TN3270E または TN3287 プロトコルによってその 2 つを接続できるようにします。クライアント用のアクセス・レコードにより特定の LU の選択が可能になる場合は、クライアントは、関連のディスプレイ LU を指定することにより特定のプリンターを選択できます。

TN サーバー・アソシエーション・レコードを構成するには、次のどちらかの方式を使用します。

Motif 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「**TN Server (TN サーバー)**」を選択し、次に「TN Server (TN サーバー)」ウィンドウ上の「Association Records (アソシエーション・レコード)」ペインを選択し、「**New (新規)**」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_tn3270_association
```

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（『TN リダイレクターのアクセス・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、135 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。ここでは一部の 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 リダイレクターのアクセス・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 リンクを経由して TN リダイレクターにアクセスできる Telnet クライアントを指示します。各アクセス・レコードは、TN リダイレクターへのアクセスが許可されている 1 つの Telnet クライアント、そのクライアントが CS Linux への接続に使用する TCP/IP ポート、CS Linux がホストへの接続に使用する TCP/IP ポート、および SSL セキュリティー設定を特定します。また、どのクライアントからもアクセス可能にするためのデフォルトのレコードも定義できます。

任意のクライアントが TN リダイレクターを使用し、すべてのクライアントが同じホスト・アクセスの構成を使用するようにしたい場合は、デフォルト・レコードを構成することができます。

Telnet クライアントは、ノードがアクティブであるときのみ TN リダイレクターを使用することができます。

TN リダイレクターのアクセス・レコードを構成するには、次のどちらかの方式を使用します。

Motif 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「**TN server (TN サーバー)**」を選択し、次にサブメニューから「**TN server (TN サーバー)**」を選択します。その次に表示されるウィンドウで、「**TN Redirector Client Access Permissions (TN リダイレクター・クライアント・アクセス許可)**」ペインを選択し、「**New (新規)**」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_redirect
```

TN リダイレクターの構成

注: コマンド行管理プログラム **snaadmin** または NOF アプリケーションで TN リダイレクター・アクセス・レコードを定義する場合は、*listen_local_address* パラメーターを使用して、TN3270 クライアントが接続されるローカル TN サーバー・コンピューター上のアドレスを指定できます。このパラメーターを使用した場合、アクセス・レコードは Motif 管理プログラムでは表示されないため、アクセス・レコードの表示または管理にはこのプログラムは使用できません。しかし、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管理を行うことはできます。

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

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。アドレスは次のいずれかです。

- 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)。

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、セキュリティー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。

証明書が有効であることの検査に加えて、TN リダイレクターは証明書を外部 LDAP サーバー上の証明書取り消しリストと突き合わせて、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ることがあります。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 のセキュリティー・レベルを指定します。セッションは、クライアントとサーバーの双方がサポートできる最高のセキュリティー・レベルを使用します。クライアントが、要求されたレベル以上のセキュリティー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、暗号化は使用されません。このオプションは通常、クライアントがセキュア・イントラネットへ接続する際に、暗号化のオーバーヘッドを回避するために使用されます。

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す)。

40 Bit Minimum

クライアントは少なくとも 40 ビット暗号化をサポートしている必要があります。

56 Bit Minimum

クライアントは少なくとも 56 ビット暗号化をサポートしている必要があります。

128 Bit Minimum

クライアントは少なくとも 128 ビット暗号化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 ビット暗号化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 ビット暗号化をサポートしている必要があります。

注: 暗号化を使用するには、CS Linux に追加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。ユーザーの場所によっては、暗号化レベルのサポートに必要なソフトウェアがユーザーの国では使用できないために、リストされているすべての暗号化レベルを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

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。アドレスは次のいずれかです。

- 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)。

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、セキュリティー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 のセキュリティー・レベルを指定しま

す。セッションは、クライアントとサーバーがサポートできる最高のセキュリティ・レベルを使用します。ホストが要求されているセキュリティ・レベル、またはそれ以上のレベルをサポートできない場合は、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、暗号化は使用されません。このオプションは通常、ホストがセキュア・イントラネットを経由して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されます。

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せん)。

40 Bit Minimum

ホストは少なくとも 40 ビット暗号化をサポートしている必要があります。

56 Bit Minimum

ホストは少なくとも 56 ビット暗号化をサポートしている必要があります。

128 Bit Minimum

ホストは少なくとも 128 ビット暗号化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 ビット暗号化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 ビット暗号化をサポートしている必要があります。

注: 暗号化を使用するには、CS Linux に追加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。ユーザーの場所によっては、暗号化レベルのサポートに必要なソフトウェアがユーザーの国では使用できないために、リストされているすべての暗号化レベルを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。しかし、CS Linux を実行しホストへの直接通信リンクを持つノードは、ダウンストリーム・コンピューター上の LU に SNA ゲートウェイの機能も提供できます。これにより、ダウンストリーム・コンピューターの LU は、CS Linux ノ

SNA ゲートウェイの構成

ードからの通信リンクを介してホストにアクセスできます。 ダウンストリーム・コンピューターには、ホストとの従属通信をサポートするための SNA PU タイプ 2.0 または 2.1 が組み込まれていなければなりません。 ダウンストリーム・コンピューターになるのは、スタンドアロン構成で CS Linux を実行する他のコンピューターなどです。

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・ノードを経由します。したがって、ダウンストリーム・コンピューターは、CS Linux コンピューターまたは他のダウンストリーム・コンピューターとホスト接続を共用でき、直接リンクは必要ありません。例えば、ローカル・トークンリング・ネットワークを介して CS Linux に接続する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、それらのコンピューターがすべて、CS Linux からホストまでの同じ長距離 SDLC 専用回線にアクセスするようにできます。

SNA ゲートウェイを使用すれば、ホストでの構成も簡素化されます。ホストの構成に組み込む必要があるのは、CS Linux コンピューターとそのホスト通信リンクのみです。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ません。

SNA ゲートウェイを構成するには、次の構成作業を行う必要があります。

- 62 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。
- 67 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。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のポートとリンク・ステーションも構成してください。ダウンストリーム・リンクについては、暗黙ダウンストリーム LU (ローカル・ノードで明示的に定義されていない LU) をサポートするために、ポート上のテンプレートを構成できます。
- ホストとの通信に使用するローカル・ノード上の LU (アップストリーム LU) を定義します。アップストリーム LU (従属 LU 6.2 LU など) を定義するときは、「LU タイプ 0 から 3 構成 (LU Type 0-3 Configuration)」ダイアログで、LU タイプとして「無制限 (不明) (Unrestricted (Unknown))」を指定します。LU を追加する方法については、85 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、87 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、ローカル・ノード上の LU を、ダウンストリーム・ワークステーションの間のセッションをサポートするように構成する必要があります。(暗黙ダウンストリーム LU をサポートするためのポート上の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。) ローカル・ノード上で定義されている LU を『downstream LU』といいます。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 番号と、ホスト LU の名前が必要です。(ダウンストリーム・ノード上で定義される LU は、どの従属 LU タイプのものでも構いません。)

ダウンストリーム LU を構成するには、次のどちらかの方式を使用します。

Motif 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、「**SNA gateway (SNA ゲートウェイ)**」および「**New downstream LU (新規ダウストリーム LU)**」を選択します。

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_downstream_lu
```

```
define_downstream_lu_range
```

ダウストリーム LU の構成パラメーター

次のパラメーターは、ダウストリーム LU 構成用の必須パラメーターです。

Downstream LU name (ダウストリーム LU 名)

各ダウストリーム 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、119 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る場合があります。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. 62 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 (67 ページの『第 4 章 接続コンポーネントの定義』を参照)。

3. 81 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、119 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・ノードとして定義します (62 ページの『ノードの構成』を参照)。
2. ダウンストリーム・ノードへの接続を構成します。67 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ます。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ー情報

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 セキュリティーを保守するためのプロトコル変更に関する追加情報を記載しています。その変更は一部の 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ります。

CS Linux には、RFC 5746 (<http://tools.ietf.org/html/rfc5746>) (バージョン 8 の GSKIT パッケージで実装) で説明されている機能が組み込まれています。これにより、SSL で保護されている TN サーバーおよび TN リダイレクター接続のハンドシェイク再ネゴシエーションのセキュリティーぜい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しておらず、再ネゴシエーションの使用を試行する場合は、クライアントからのセキュア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する場合は、このプロトコル変更の前に、(再ネゴシエーションを許可して) CS Linux の動作を旧操作に戻すことができます。これを実行するには、CS Linux を開始する前に、環境変数 GSKIT_RENEGOTIATION をシェルに設定します。

- 再ネゴシエーションを許可するには、クライアントの要件に応じて **GSKIT_RENEGOTIATION=FULL** または **GSKIT_RENEGOTIATION=ABBREVIATED** を設定します。
- 再ネゴシエーションをしないようにするには、**GSKIT_RENEGOTIATION=NONE** を設定します。

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ー情報

- 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つもりであれば、<http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-3555> を参照し、そこに記述されている起きる可能性があるセキュリティーの問題を認識していることを確認してください。

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、『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、140 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、141 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。

ホスト NetView プログラムの使用

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 画面の表示

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 によって異なります。代表的なレイアウトを、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。

```
NCCF          N E T V I E W          [SCAN DDAC12  07/18/95  13:52:24 A]
              RUNCMD SP=ADCDPU01,APL=NODE,START_DLCL,DLC_NAME=TKKR01
              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 コマンドも UCF コマンドも、RCF コマンド構文を使用します。

runcmd *sp=spname, appl=component, commandtext*

NetView は **runcmd** ユーティリティを使用して、リモート・システムにコマンド・ストリングを送ります。このコマンドには次のパラメーターがあります。

sp=spname

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。この情報は、ホスト NetView の担当者から入手できます。

appl=component

NetView がコマンドを送る CS Linux 構成要素の名前を、次のように指定します。

node サービス・ポイント名 *spname* に関連付けられている CS Linux ノード (SPCF コマンドの場合)。

unix サービス・ポイント名 *spname* に関連付けられている CS Linux コンピューターで実行されている UCF デーモン・プログラム。

commandtext

発行するコマンドのテキストを入力します。SPCF の場合は、これは 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す。UCF の場合は、これは Linux オペレーティング・システム用のコマンドです。使用できるコマンドの詳細は、140 ページの『SPCF で使用する管理コマンドに関する制約事項』または 142 ページの『使用できるコマンド』を参照してください。

大文字とエスケープ文字

Linux では英字の大文字と小文字は区別されますが、NetView プログラムでは区別されない場合があります。NetView **netvaxis** コマンドを使用して **runcmd** に大/小文字混合データを入力できますが、CS Linux RCF には **netvaxis** が使用されているかどうかを判別する手段がありません。RCF はホストから受け取った英字がもともと大文字か小文字か判別できないため、受け取った文字を小文字として仮定します。また、一部のコマンドでは大括弧の [および] が必要になりますが、ホスト文字セットではこれらの文字がサポートされないことがあります。

RCF は、次のように円記号 ¥ を使用することにより、大文字および大括弧をサポートできるようになります。

- コマンド・ストリングに大文字を含めるには、その文字の前に円記号 ¥ を付けます。前に円記号 ¥ が付いていない英字は、すべて小文字として解釈されます。
- 大括弧 [および] を含めるには、それぞれ ¥(および ¥) の文字列を使用します。
- 円記号 ¥ 自体を含めるには、この文字を 2 つ続けて入力します。

1 つの ¥ のあとにその他の英字以外の文字を続けた場合は、円記号は無視され、その文字は未変更のまま残されます。

表3 に例を示します。

表3. RCF コマンド内でのエスケープ文字の使用

生成する文字	入力
ABcd	\Abcd
[]	\(\)
¥a	\a
\	\\

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、RCF 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。また、SPCF を使用して管理コマンドを発行するときは、LIST_FROM_NEXT などの定数名には大文字小文字の区別はありません。このような文字は、エスケープを使用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** を使用して実行できるコマンドと同じです（「*IBM Communications Server for Data Center Deployment on Linux 管理コマンド解説書*」で説明されています）。

SPCF コマンドの構文については、139 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 で使用す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。コマンドは、すべて NetView コンソールで直接入力する必要があります。

query_* コマンドでは、コマンド行オプション `-a` (全エントリーを戻す) および `-d` (詳細情報を戻す) を、Linux コマンド行にコマンドを入力するときと同じ方法で使用できます。

セキュリティを確保するためには、SPCF から特定タイプのコマンドしか発行できないように、CS Linux 構成情報を設定することができます。例えば、リモート・ユーザーに、**query_*** コマンドの発行は許可し、CS Linux の構成要素の活動化または非活動化は許可しないことが可能です。次に示すそれぞれのコマンド・グループについて、別々にアクセスを制御できます。

- **define_***、**set_***、**delete_***、**add_***、および **remove_*** コマンド、さらに **init_node**
- **query_*** コマンド
- 「アクション」コマンド: **start_***、**stop_***、**activate_***、**deactivate_***、および、**aping**、**initialize_session_limit**、**change_session_limit**、および **reset_session_limit**

SPCF のセキュリティー・オプションの設定の詳細については、「*IBM Communications Server for Data Center Deployment on Linux 管理コマンド解説書*」の **define_rcf_access** コマンドの説明を参照してください。

SPCF コマンドの例

次の例は、SPCF を使用して **define_lu_0_to_3** コマンドを発行する方法を示しています。この例では、円記号を使用して、2 つの文字ストリング LU\$01 および PU2 の中の大文字を指示しています。定数名 **3270_display_model_2** の中の文字を大文字にする必要はありません。**snaadmin** プログラムは、このストリングを小文字のまま受け付けます。

```
runcmd sp=mypname, appl=node, define_lu_0_to_3, lu_name=¥¥u$01,
nau_address=1, pu_name=¥p¥u2, lu_model=3270_display_model_2
```

次の例は、SPCF を使用して **query_lu_0_to_3** 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「全エントリを戻す」を指示しているので、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「詳細情報を戻す」を指示しているので、**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役割をします。

```
runcmd sp=mypname, appl=node, -a -d query_lu_0_to_3
```

UCF の使用

UCF は、NetView オペレーターが NetView コンソールでコマンド・テキストを入力することにより、CS Linux を実行するコンピューターで Linux コマンドを発行し、そのコマンドからの出力を表示できるようにします。この機能は CS Linux に関連したコマンドだけに限定されるわけではありません。142 ページの『使用できるコマンド』に示す制約の範囲内で、どのようなタイプも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 コンピューターでのアクティビティを監視し、問題を診断し、必要な場合には訂正処置をとります。

CS Linux が UCF をサポートするかどうかは、**define_rcf_access** コマンドを使用して指定できます（「*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）。構成に UCF のサポートが指定されている場合、CS Linux は、ノードの開始時に、UCF デーモン・プログラムを開始します。UCF デーモンは、UCF から投入された Linux コマンドを処理すると

きに、各コマンドごとに新しい Linux シェルを起動し、そのシェルの中でコマンドを実行します。UCF サポートが組み込まれていない場合は、CS Linux はこのプログラムを開始しません。

構成情報には、UCF ユーザーの名前を指定します。この名前は CS Linux コンピューター上の有効なログイン名でなければなりません。UCF シェルは、そのユーザー用として指定されているシェル・プログラム、ログイン ID、許可、および **.login** または **.profile** を使用して開始されます。(シェル・プログラムが指定されていない場合は、**/bin/sh** を使用します。)これは、通常の Linux システム・セキュリティ機能を使用して、ファイルおよびコマンドへの UCF ユーザーのアクセスを制限でき、したがって UCF から使用できるコマンドの範囲を限定できることを意味します。

UCF 構成情報の設定方法の詳細については、「*IBM Communications Server for Data Center Deployment on Linux 管理コマンド解説書*」にある **define_rcf_access** コマンド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が異なります (139 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 `cat filename` 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 `mv filename1 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せんが、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 `vi filename` や、ユーザーが停止するまで完了しない `tail -f filename` 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、144 ページの『UCF のセキュリティー』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=mypsname, appl=unix, grep \temp \ab\*.c >\tle\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 画面に次のメッセージが表示されます。

```
=== EXECUTING UNIX COMMAND ===
(any output from the command, including error messages)
=== UNIX COMMAND COMPLETED ===
```

これらのメッセージは、NetView 画面に同時には表示されないことがあります。UCF デーモン・プログラムがコマンドを受け取り、NetView オペレーターに制御を戻した時点で、**EXECUTING UNIX COMMAND** メッセージが表示されます。コマンドから生成された出力はすべて NetView に送られ、一連の独立したメッセージとして表示されます。**UNIX COMMAND COMPLETED** メッセージは、Linux コマンドが完了し、シェルが終了した時点で表示されます。

Linux コマンドからの出力にタブ文字が含まれている場合は、CS Linux は、出力を NetView に送る前に、各タブをスペース文字に変換します。タブ文字がない場合は、出力は無変更のまま送られます。

前のコマンドの実行中に (つまり **UNIX COMMAND COMPLETED** メッセージを受信する前に) 別のコマンドを発行した場合は、次のメッセージが表示されます。

```
=== COMMAND QUEUED ===
```

新しい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 には、実行中のコマンドを取り消す方法があります。これを使用すると、現行
コマンドの実行を停止したり、**vi filename** などのように追加入力がないと完了でき
ないような対話型コマンドを取り消したりできます。これは、**Ctrl + C** などの割
り込みシーケンスを使用して端末でのプロセスの実行を停止したり、Linux **kill** コマ
ンドを使用してプロセスを停止したりする場合と同じです。

現在実行中のコマンドを取り消すのみでなく、CS Linux は、そのコマンド以降にキ
ューに入れられたコマンドも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
にストリング **ux-cancel** を使用します。例えば、次のように入力します。

```
runcmd sp=mypname, appl=unix, ux-cancel
```

未処理の各コマンド (現在実行中のコマンドとキューに入っているコマンド) には、
次のメッセージが表示されます。

```
=== UNIX COMMAND CANCELLED ===
```

このメッセージは、コマンドを実行していた Linux シェルが停止したことを示しま
す。以降も、必要に応じて Linux コマンドを発行できます。

コマンドによって Linux コンピューターでデーモン・プロセスが開始された場合、
このプロセスは **ux-cancel** によって停止されないことがあります。その場合は、
Linux **kill** コマンドを使用して (端末で発行するかまたは UCF を使用)、プロセス
を明示的に停止する必要があります。

ux-cancel を使用したときに実行中の UCF コマンドが 1 つもなかった場合は、
UCF は次のメッセージを表示します。

```
NO OUTSTANDING COMMANDS
```

この場合は、**ux-cancel** コマンドは無視されます。アクションは不要です。前のコマ
ンドが終了していても、まだ **UNIX COMMAND COMPLETED** メッセージが受信
されていないうちに **ux-cancel** コマンドを発行すると、このメッセージが表示され
ることがあります。

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、セキュリティー
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 システムにすべての SNA コンポーネントとアプリケーションを備えたスタンドアロン・システムとして実行することができ、またクライアント/サーバー・ドメインの一部として実行することもできます。クライアント/サーバー・ドメインには、サーバー (SNA ノード) と IBM Remote API Client (サーバーを介して SNA 接続にアクセスできる) の両方が含まれます。

複数の CS Linux サーバーがあるドメインでは、1 つのサーバーが CS Linux ドメイン構成ファイルのマスター・コピーを保持します。このサーバーをマスター・サーバーと呼びます。ドメイン内のその他のサーバーは、バックアップ・サーバーとして定義できます。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できるように、バックアップ・サーバーが開始したとき、またはマスター・コピーが変更されたときに、ドメイン構成ファイルはバックアップ・サーバーにコピーされます。

Remote API Client は、AIX、Linux、Linux for pSeries、Linux for System z、または Microsoft Windows を実行するコンピューターとすることができます。

サーバーとクライアントは、TCP/IP を使用して CS Linux ドメイン上で通信します。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、1 つ以上のサーバーに同時にアクセスでき、必要なときにいつでも同時アプリケーションを実行できます。クライアント/サーバー構成のためのネットワーキング要件については、149 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間で使用される TCP/IP 接続は、物理 LAN、WAN、または VM の下で実行されるサーバー間の仮想パスを経由することができます。CS Linux の資料では、これらすべてを LAN という用語で記述しています。

UNIX

AIX または Linux 上の Remote API Client に対しては、CS Linux ネットワークとサーバーに関する情報を提供する必要があります。この機能、およびクライアント上で CS Linux ソフトウェアを使用可能または使用不可にする手順については、174 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。

管理コマンドはすべてサーバー上で実行できます。ただし、AIX および Linux クライアントで実行できるコマンドについては制限があります。

- AIX または Linux クライアントでは、**query** コマンドまたは **status** コマンドをすべて実行できます。
- 「IBM Communications Server for Data Center Deployment on Linux 管理コマンド解説書」に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 から実行できることが明示されています。明示されていないコマンドは、サーバーからのみ実行できます。

WINDOWS

Windows クライアントに対しては、クライアント・ソフトウェアを使用可能に設定するために CS Linux によって使用される情報を提供する必要があります。

Windows クライアント上で呼び出し可能 TP を使用する場合は、TP に関する情報も提供する必要があります。これらの機能の概要、および Windows クライアント上で CS Linux ソフトウェアを使用可能および使用不可に設定する手順については、153 ページの『Windows 上の Remote API Client の管理』を参照してください。

「IBM Communications Server for Data Center Deployment on Linux 管理コマンド解説書」で定義されている管理コマンドは、Windows クライアントから実行することはできません。



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、初期時にはスタンドアロン・モード (すべてのコンポーネントが単一の Linux コンピューターに置かれる) でインストールされます。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 つのサーバーをマスター・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。(マスタ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 (マスター・サーバーから始めます)。

```
sna stop  
snanetutil master_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。

master_name

サーバーが属することになるドメインのマスター・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マスター・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。

マスタ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、マスター・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 ソフトウェアはマスター・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。

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

クライアントを別のドメインに移動する

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。詳細は、156 ページの『Windows 上の Remote API Client 構成』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。詳しくは、175 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、153 ページの『Remote API Client 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、150 ページの『IP ポート番号の設定』で説明するように競合を解決する必要があります。

また、151 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。

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 アドレッシングのセットアップ方法および使用方法の詳細については、「*IBM Communications Server for Data Center Deployment on Linux 入門*」を参照してください。

クライアント/サーバー構成におけるホスト名

CS Linux では、サーバーとクライアント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・システムでは、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。それが不可能な場合は、構成内でホスト名が必要な箇所には必ず完全修飾名 (newbox.this.co.uk など) を使用し、別名 (newbox など) は使わないようにします。

各コンピューターのローカル・サーバー名は、`/etc/hosts` ファイルから取られます。このファイルのエントリーには、まず IP アドレスを指定し、次に完全修飾名を指定し、最後に別名を指定しなければなりません。例えば、次のようにします。

```
9.42.108.28 newbox.this.co.uk newbox
```

マルチホーム・サーバーの場合 (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あって、それぞれが異なる IP アドレスを持つ場合) は、`/etc/hosts` ファイル内のエントリーですべてのアドレスに同じ IP 名を指定する必要があります。このようにすれば、すべてのネットワーク・インターフェースについて名前が正しく解決されます。例えば、次のように入力します。

```
9.42.108.28 newbox.this.co.uk newbox
9.42.80.127 newbox.this.co.uk newbox
```

また、サーバーがマルチホームであることを示すために、`/etc/host.conf` ファイルに `multi on` という行を含める必要があります。

IP ポート番号の設定

CS Linux は、TCP/IP と UDP/IP の両方の通信を使用して、クライアント/サーバー・データを LAN 経由で送信します。デフォルトでは、両タイプの通信に、ポート番号 1553 を使用します。ほとんどのシステムの場合は、このポート番号が適しており、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設定する際に問題が生じた場合は、エラー・ログ・ファイルを検査して、CS Linux に使用されているポート番号が、別のプログ

ラムに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がないかどうか調べてください。このようなメッセージが見つかった場合は、次の手順を行います。

1. エラーが発生したコンピューター上の `/etc/services` ファイルを見て、別のプログラムが TCP/IP 通信または UDP/IP 通信用にポート番号 1553 を使用していることがリストに示されているかどうか確認します。これが当てはまる場合は、まず他のプログラムを変更して別のポートを使用することを試みてください。
2. この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、このファイルにリストされているどのプログラムにも使用されていない、別のポート番号を見つけます。同じドメイン内にある他の CS Linux コンピューター上の `/etc/services` ファイルを見て、その番号が別のコンピューターに使用されていないことを確認します。
3. ドメイン内にある各コンピューターの `/etc/services` ファイルに、次の形式で 2 つの行を追加します。

```
sna-cs      nnnn/tcp
sna-cs      nnnn/udp
```

`nnnn` エントリーは、新しいポート番号です。この番号は、CS Linux ドメイン内にあるすべてのコンピューター上で同じ値に設定する必要があります。

- 4.

WINDOWS

CS Linux ドメインに Windows クライアントが含まれる場合は、各 Windows コンピューター上の `services` ファイルに同じ 2 行を追加します。`services` ファイルは Linux ファイルと同じ形式で、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ています。詳細情報が必要な場合は、ご使用の Windows TCP/IP の資料を参照してください。

5. CS Linux サーバーと Remote API Client ソフトウェアを再び使用可能にします。

注: 無許可アクセスを防止するために、ファイアウォールを使用して、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された新規ポート番号) を保護することをお勧めします。他の CS Linux サーバーおよび Remote API Clients との間では TCP トラフィックと UDP トラフィックの両方を許可する必要がありますが、それ以外のコンピューターには上記ポートへのアクセスを許可しないようにします。

LAN アクセス・タイムアウト

接続料金が課金されるネットワークを経由してクライアントが CS Linux サーバーと通信する場合は、クライアント上のアプリケーションが CS Linux リソースの使用を停止した後で、クライアントからの TCP/IP 接続を自動的にドロップするように設定できます。クライアント上の SNA ソフトウェアは自動的に使用不可に設定

IP ネットワーキングの要件

されません。SNA ソフトウェアはアクティブの状態を保ち、アプリケーションがあとでサーバーに接続する必要がある場合は、サーバーへの接続の再確立を試行します。

lan_access_timeout パラメーター (AIX または Linux 上の Remote API Client の場合は **sna_clnt.net** ファイル内、Windows 上の Remote API Client の場合はレジストリー内) を使用して、クライアント上の SNA ソフトウェアを使用不可に設定できます。指定した時間内にクライアント上で次に示すイベントが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 会話がアクティブになった (または会話の開始を試行した)
- LUA セッションがアクティブになった
- Windows クライアントからの CSV TRANSFER_MS_DATA verbs
- MS verbs (Linux クライアントのみ)
- NOF verbs (**query_central_logger** または **query_node_all** verbs を除く)
- 管理コマンド (クライアントが接続を再開する理由にならない、次に示すイベントを除く)
 - クライアントによってエラーまたは監査メッセージがログに記録された (これらは、中央ロギングが使用されていても、クライアント上でローカルに記録されます)
 - 管理コマンドの **query_central_logger** または **query_node_all** (これらは、TCP/IP 接続がドロップされる前に入手可能だった情報を戻すため、LAN の現在の状況とは一致しない可能性があります)
 - NOF verbs **query_central_logger** または **query_node_all** (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設定したにもかかわらず、指定したタイムアウト期間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かった場合に、TCP/IP 接続がドロップされます。

TCP/IP 接続が停止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 175 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)』または 160 ページの『Servers』の * および *servername* パラメーターで説明するように、サーバーへの接続の試行を再開します。

TCP/IP 接続が停止している間は、このクライアント上の呼び出された TP に対する接続の着信を受け入れることはできません。ターゲット・システムがアクティブでない場合と同じように、接続はリジェクトされます。つまり、クライアント上で自動的に開始された TP は、クライアント上で他のアプリケーションが実行されない場合に TCP/IP 接続がタイムアウトになると、使用不可になります。ただし、クライアント上でオペレーターが開始した TP は、いつでも使用できます。これは、TP によって実行された Receive_Allocate verb が TCP/IP 接続を再確立するからです。

Remote API Client の HTTPS アクセス

Remote API Client が HTTPS を使用して CS Linux サーバーに接続するようなクライアント/サーバー・システムを実行する予定の場合、そ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 が必要になります。このサーバーのインストールと構成の方法については、「*IBM Communications Server for Data Center Deployment on Linux 入門*」を参照してください。

CS Linux ドメインに新しいサーバーを追加し、Remote API Client がこれらのサーバーに HTTPS を使用してアクセスするように設定する場合は、WebSphere サーバー構成ファイルを更新して、これらのサーバーが含まれるように設定する必要があります。このファイルは `snahttpsrv.cfg` という名前です。USER_INSTALL_ROOT 環境変数で指定された、WebSphere サーバー上のディレクトリに保管されています。それがどこにあるか分からない場合は、次の手順を行います。

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、「Environment (環境)」、「Manage WebSphere Variables (WebSphere 変数の管理)」を選択します。
3. このリストの中で **USER_INSTALL_ROOT** 変数を見つけ、その値 (これは、WebSphere サーバー上のディレクトリのパスです) をメモします。環境変数のリストは、2 ページ以上になる場合もあり、「次へ」ボタンを使用して、リストをスクロールする必要がある場合もあります。

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 Remote API Client でアクセスできるすべての CS Linux サーバーのリストを含めます。各サーバーは次の形式で、別々の行に指定する必要があります。

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

新しいサーバーにアクセスする各 Remote API Client 上のクライアント・ネットワーク・データ・ファイル (または Windows Client の場合は Windows レジストリー) で、サーバーのリストに新しいサーバー名を追加する必要があります。この章のあとの方の該当するクライアント・タイプの節を参照してください。

Windows 上の Remote API Client の管理

WINDOWS

CS Linux では、Microsoft Windows が稼働するマシンを CS Linux ドメインのクライアントとして使用することができます。CS Linux クライアント・ソフトウェアに組み込まれている API ライブラリーは、Microsoft Host Integration Server、Windows Open Systems Architecture (WOSA)、および IBM Personal Communications and Communications Server for Windows 提供のインターフェースと互換性があります。これによって、これらのインプリメンテーションに適合するように書かれたアプリケーションは、変更を加えずに Windows 上の Remote API Client 上で実行できます。

Windows 上の Remote API Client は、次の WOSA API をサポートします。

Windows 上の Remote API Client の管理

- Windows APPC
- Windows CPI-C
- Windows LUA
- Windows CSV

Windows SNA API の詳細は、Microsoft Host Integration Server に付属の資料を参照してください。

SNA ネットワーク情報など、Windows 上の Remote API Client が必要とする情報は、Windows レジストリーに保管されています。

CS Linux アプリケーション、またはエミュレーション・プログラムをクライアント上で使用するには、事前にクライアントを使用可能に設定する必要があります。詳細は、『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。クライアントを使用可能に設定すると、クライアントは CS Linux 機能にアクセスするために、TCP/IP ネットワーク経由で CS Linux を実行するサーバーに接続します。

クライアントの操作は、Windows レジストリーの情報によっても制御されます。Windows レジストリーには、次の情報もあります。

- Windows 上の Remote API Client 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るアプリケーションのロギングとトレースのオプション
- クライアントで実行される CPI-C および CSV アプリケーションの追加のオプション
- 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)

通常使用されるパラメーターも、クライアント構成ユーティリティを使用して変更できます。クライアント構成ユーティリティは、パラメーターを変更する場合に最もよく使用される方法です。詳しくは、156 ページの『Windows 上の Remote API Client 構成』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、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、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。詳細は、156 ページの『Windows 上の Remote API Client 構成』を参照してください。

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、156 ページの『Windows 上の Remote API Client 構成』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 上の Remote API Client の状況の表示

クライアント・モニターは、システム・トレイにアイコンを表示します。マウス・ポインターをこのアイコンに重ねると、クライアントの状況が表示されます。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により手動で実行することもできます。

- スタート・メニューで、Windows プログラム・グループの Remote API Client からクライアント・モニターを選択します。
- コマンド・プロンプトで、クライアント・ソフトウェアがインストールされた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。この情報を異なる言語で表示したい場合、次のようにコマンド・プロンプトからクライアント・モニターを開始することで可能になります。クライアント・ソフトウェアがインストールされたディレクトリーに移動して、さらに希望する言語に対応したサブディレクトリーに移動後、`sxclappl` コマンドを実行します。例えば、情報をフランス語で表示する場合は、クライアント・ソフトウェアがインストールされたディレクトリー下の `fr_FR` サブディレクトリーから `sxclappl` コマンドを実行します。

クライアント・モニターは、以下のいずれかの状況を表示します。

Not Active

クライアントは開始されていません。

Not Connected

クライアントは開始済みですが、サーバーに接続していません (または、接続を失いました)。

Server_Name

クライアントは、示されたサーバーに接続しています。

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。

Windows 上の Remote API Client の管理

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 上の Remote API Client 構成

Windows 上の Remote API Client 上で、構成情報は Windows レジストリーに保管されています。レジストリーには、SNA ネットワーク情報が保管されています (AIX または Linux 上の Remote API Client のクライアント・ネットワーク・データ・ファイルに保管されている情報と同様)。レジストリーには、Windows 上の Remote API Client に固有の追加構成情報も含まれます。

注: CPI-C アプリケーションの構成情報 (ローカル TP 名とローカル LU 別名) は、環境変数、またはレジストリーの中に指定できます。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があります。詳細は、170 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は、一般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ます。クライアント構成ユーティリティーは、パラメーターを変更する場合の推奨方法です。このプログラムは、次のいずれかの方法で開始できます。

- スタート・メニューで、Windows プログラム・グループの Remote API Client から構成ユーティリティーを選択します。
- コマンド・プロンプトで、クライアント・ソフトウェアがインストールされたディレクトリーに移動して、`sxclconf` コマンドを実行します。

構成ユーティリティーは通常、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。この情報を異なる言語で表示したい場合、次のようにコマンド・プロンプトから構成ユーティリティーを開始することで可能になります。クライアント・ソフトウェアがインストールされたディレクトリーに移動して、さらに希望する言語に対応したサブディレクトリーに移動後、`sxclconf` コマンドを実行します。例えば、情報をフランス語で表示する場合は、クライアント・ソフトウェアがインストールされたディレクトリー下の `fr_FR` サブディレクトリーから `sxclconf` コマンドを実行します。

このプログラムは、最初のインストール・プロセスで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の変更方法については、「IBM Communications Server for Data Center Deployment on Linux 入門」の Windows での Remote API Client のインストールに関する章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あと、クライアントを停止して再起動し、変更を有効にする必要があります。これを行う方法の詳細は、155 ページの『Windows 上の Remote API Client を使用不可に設定する』と 154 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。

レジストリーの中で、情報は、次に示すキーのサブキーの下に構成される値として保管されます。

```
\\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次のとおりです。

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_timeout = nn
broadcast_attempt_count = nn
server_lost_timeout = nn
client_start_timeout = nn
```

Servers

```
Server1 = * | [ webservername : [ portnumber : ] ]servername1
Server2 = [ webservername : [ portnumber : ] ]servername2
.
.
.
Server9 = [ webservername : [ portnumber : ] ]servername9
```

Logging

```
exception_logging_enabled = YES | NO
audit_logging_enabled = YES | NO
log_directory = directory
error_file = error_filename
backup_error_file = backup_error_filename
error_file_wrap_size = error_file_size
audit_file = audit_filename
backup_audit_file = backup_audit_filename
audit_file_wrap_size = audit_file_size
succinct_errors = YES | NO
succinct_audits = YES | NO
```

API_tracing

```
file1 = trace_filename_1
file2 = trace_filename_2
flip_size = filesize
truncation_length = length
all_api = YES | NO
appc = YES | NO
cpic = YES | NO
csv = YES | NO
rui = YES | NO
nof = YES | NO
```

CS_tracing

Windows 上の Remote API Client の管理

```
file1 = cs_trace_filename_1
file2 = cs_trace_filename_2
flip_size = filesize
admin_msg = YES | NO
datagram = YES | NO
data = YES | NO
send = YES | NO
receive = YES | NO

Internal_tracing
file1 = internal_trace_filename_1
file2 = internal_trace_filename_2
flip_size = filesize
trace_level = nn
trace_flushing = YES | NO

Appl_Name
APPCTPN = tp_name
APPCLLU = lu_name

CSV_data
CSVTBLG = table_G_filename
```

注: domain = domain_name 値は、レジストリー内で必要な唯一の値です。

次の節では、構成パラメーターについて説明します。パラメーターが YES または NO の値をとる場合、Y または y で始まるストリングは YES として解釈され、N または n で始まるストリングは NO として解釈されます。

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・インストール時に指定された CS Linux LAN のドメイン名を示します。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。

次のいずれかの値を指定します。

- 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、100 ページの『TP の定義』または 207 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。詳細は、151 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、アクティビティーがないため、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・メッセージ間の時間は、1 から 1440 (1 分から 1 日) の範囲で、分単位で指定します。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、アクティビティー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 (『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 (『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 (『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できる CS Linux 構成サーバーに関する情報が入っています。このリストには、クライアントと同じドメイン内のマスタ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。マスター・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、61 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ことについて詳しくは、「*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 から 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。

Logging

Logging 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すると、ドメイン全体に対して指定されたロギング・オプションを指定変更するクライアント・ロギング設定値を指定できます。ドメインのロギング・オプション指定の詳細は、63 ページの『ロギングの構成』を参照してください。

中央ロギングが使用可能に設定されている場合は、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、ここで指定された *exception_logging_enabled* と *audit_logging_enabled* パラメーターのみ使用されます。あとのパラメーターは無視されます。

ロギング・オプションは、次のように指定します。

exception_logging_enabled

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、次のいずれかの値に設定します。

YES 例外メッセージを記録します。

NO 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。指定しない場合、クライアントは、グローバル・ドメイン設定値を使用して、例外メッセージを記録するかどうか決定します。(初期のデフォルトでは、例外メッセージは記録されます)

audit_logging_enabled

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、次のいずれかの値に設定します。

YES 監査メッセージを記録します。

NO 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。指定しない場合、クライアントはグローバル・ドメイン設定値を使用して、監査メッセージを記録するかどうか決定します。(初期のデフォルトでは、監査メッセージは記録されます)

log_directory

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。ログ・ファイルとバックアップ・ログ・ファイル (後続のパラメーターに指定される) は、すべてこのディレクトリーに保管されます。「*IBM Communications Server for Data Center Deployment on Linux 診断ガイド*」に説明されているフィルター機能を使用する場合は、(この機能を制御する) ファイル **logfilter.txt** 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。指定しない場合、ファイルは Windows 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 REG_SZ です。

エラー・メッセージを書き込む先のファイルの名前。このパラメーターはオプションです。指定しない場合、デフォルトは **sna.err** です。

エラー・メッセージと監査メッセージのログを単一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。

backup_error_file

この値のレジストリー・データ・タイプは REG_SZ です。

バックアップのエラー・ログ・ファイルの名前。エラー・ログ・ファイルが *error_file_wrap_size* に指定されたサイズに達すると、CS Linux はログ・ファイルの内容をバックアップ・ファイルにコピーしてから (既存ファイルは上書きされる)、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。指定しない場合、デフォルトは **bak.err** です。

エラー・メッセージと監査メッセージのログを単一のファイルに記録するには、このパラメーターと *backup_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。

error_file_wrap_size

この値のレジストリー・データ・タイプは REG_DWORD です。

error_file に指定されたログ・ファイルの最大サイズ。書き込まれたメッセージが原因でファイル・サイズがこの限度を超えた場合、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。このため、エラー・ログ・ファイルによって占有されるディスク・スペースの最大量は、*error_file_wrap_size* パラメーターの値の約 2 倍です。

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 1000000 (バイト) です。エラー・メッセージと監査メッセージのログを同じ

ファイルに記録する場合、このパラメーターは *audit_file_wrap_size* 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 REG_SZ です。

監査メッセージを書き込む先のファイルの名前。このパラメーターはオプションです。指定しない場合、デフォルトは **sna.aud** です。

エラー・メッセージと監査メッセージのログを単一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。

backup_audit_file

この値のレジストリー・データ・タイプは REG_SZ 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 *audit_file_wrap_size* に指定されたサイズに達すると、CS Linux はログ・ファイルの内容をバックアップ・ファイルにコピーしてから (既存ファイルは上書きされる)、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。指定しない場合、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のログを単一のファイルに記録するには、このパラメーターと *backup_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_wrap_size

この値のレジストリー・データ・タイプは REG_DWORD です。

audit_file に指定されたログ・ファイルの最大サイズ。書き込まれたメッセージが原因でファイル・サイズがこの限度を超えた場合、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。このため、監査ログ・ファイルによって占有されるディスク・スペースの最大量は、*audit_file_wrap_size* パラメーターの値の約 2 倍です。

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 1000000 (バイト) です。エラー・メッセージと監査メッセージのログを同じファイルに記録する場合、このパラメーターは *error_file_wrap_size* パラメーターと同じ値に設定する必要があります。

succinct_errors

この値のレジストリー・データ・タイプは REG_SZ です。

エラー・ログ・ファイル内で要約ログを使用するか、詳細ロ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。次のどちらかの値を指定できます。

YES 要約ログを使用します。ログ・ファイル内の各メッセージには、メッセージ・ヘッダー情報の要約 (メッセージ番号とログ・タイプなど) と、メッセージ・テキスト・ストリングとパラメーターが示されます。ログが記録された原因、および必要な処置についての詳細情報を入手するには、Linux を実行するコンピューター上で **snahelp** ユーティリティーを使用できます。

NO 詳細ログを使用します。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、およびログが記録された原因と必要な処置に関する追加情報が示されます。

このパラメーターはオプションです。指定しない場合、デフォルトはマスター・サーバーに対して前に実行された **set_global_log_type** コマンドから取り込まれます (または、Motif 管理プログラムを使用して設定される)。

set_global_log_type コマンドが実行される前の初期のデフォルトでは、要約ログを使用します。

中央ロギングを使用する場合、すべてのコンピューターからのメッセージに要約または詳細のどちらのロギングを使用するかを選択は、中央ロガーとして機能するサーバー上でのこのパラメーターの設定値によって決まります。この設定値は、**set_global_log_type** コマンド、またはデフォルトを指定変更するためにそのサーバーに対して実行された **set_log_type** コマンドによって設定されます。

succinct_audits

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・ファイル内で要約ログを使用するか、詳細ログを使用するかを指定します。許される値とその意味は、*succinct_errors* 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。

トレース・ファイルに書き込む各メッセージの情報の最大長 (バイト)。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* を超えるデータを破棄します。これにより、長いメッセージでファイルがいっぱいに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。指定する場合、デフォルトは NO です。*all_api* パラメーターを YES に設定した場合は、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* パラメーターを YES に設定した場合は、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。指定する場合、デフォルトは NO です。*all_api* パラメーターを YES に設定した場合は、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。

ru 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* パラメーターを YES に設定した場合は、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。それ以外の場合は 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は使用されませんが、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で使用されます。

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* パラメーターを YES に設定した場合は、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。

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 はまず環境変数を確認し、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は、環境変数を使用する必要があります。

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。

サーバーのドメイン上のクライアントによって使用される SNA LU リソースの追跡

APPC アプリケーションは、タスクを完了させるためにそのパートナー・アプリケーションと 2 回以上の会話が必要になる場合があります。パートナー・アプリケーションがその処理を即時に完了できない場合は特にそのようになります。例えば、アプリケーション A はデータをアプリケーション B に処理のために送信し、この処理は完了にいくらか時間がかかり、その結果を待たずに終了する場合などです。後で、処理を完了したときに、アプリケーション B は結果を返すためにアプリケーション A と新規の会話を開始します。

このように動作するアプリケーションが 2 つ以上のクライアントで実行する場合、CS Linux によって、要求を行った元のクライアントに、新規会話が確実に戻されるようにする必要があります。特に、クライアント・アプリケーションが LU プール内の LU を使用する場合、元の要求に使用される LU との関連付けを維持し、新規会話がこの LU から正しいクライアント・アプリケーションにルーティングできるようにする必要があります。

この機能を使用するために、LU プールを所有するノードに対して、`define_node` コマンドの `ptf_flags` パラメーターで `CLI_OVERWRITE_SYS_NAME` オプションを設定する必要があります。それから標準テキスト・エディターを使用して、クライアント・ソフトウェアがインストールされているディレクトリーに、`tpnamehost` 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。

`tpnamehost` ファイルがクライアントに存在している場合は、以下のようになります。

- CS Linux は、この IP ホスト名に一致する、アプリケーションによって使用される LU の `sys_name` を変更します。これにより、パートナー・アプリケーション

がこの LU を使用して新規会話を要求する場合、CS Linux は必ずそれを、指定されたクライアント上で実行しているアプリケーション (この LU を使用して元の要求を行ったアプリケーション) と接続します。

- さらに 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、% 文字と、それに続くクライアント・コンピューターの IP ホスト名を付加します。この変更された TP 名ストリングは、このクライアントで実行するアプリケーションのロギングとトレースで使用されます。これにより、異なるクライアント上の複数のアプリケーションのコピーを区別することができます。

注: `tpnamehost` ファイルをクライアント上で作成する場合は、`tpnameipv4` という名前のファイルが同じロケーションに存在していないことを確認します。(このファイルの使用については、173 ページの『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』で説明しています。) その両方のファイルが存在している場合は、`tpnameipv4` ファイルが優先します。CS Linux は IP ホスト名ではなく IP アドレスを付加し、クライアント・アプリケーションとその LU との間のアソシエーションは作成しません。

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 別名と、サーバーのドメインの LU 別名のマッピング

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。(Remote API Client on AIX or Linux における同等機能についての情報は、180 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ディングされた LU 別名と、サーバーのドメインの LU 別名のマッピング』を参照してください。)

Remote API Client on Windows の `lualiasmap.txt` ファイルは、このクライアント上で実行す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 LU との間の変換を定義します。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、それらが異なる LU を使用するように強制することができます。

- 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、これは `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た `lua_luname` パラメーターと、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タイプ 0 から 3 の LU とをマップします。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は、標準テキスト・エディターを使用して、**lualiasmap.txt** という名前の ASCII テキスト・ファイルを、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作成します。

このファイルの各行には、シングル・スペースで分離された 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 で開始する必要があり (最初の 8 文字ストリングに 1 つのスペースが続いた後の位置)、このクライアント上で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で 64 行まで含めることができます。

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。
- LU 名 LUALU3 または LUALU4 を指定するこのクライアント上で稼働する 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 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。

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

Remote API Client on Windows 上の **tpnameipv4** ファイルは、CS Linux に、クライアント上で実行する APPC または CPI-C アプリケーションによって指定された TP 名を変更するように指示します。これによって各クライアントは (その IPv4 アドレスによって区別される) 固有の TP 名を使用することになります。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 構成内での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は、標準テキスト・エディターを使用して、**tpnameipv4** という名前のファイルを、クライアント・ソフトウェアがインストール

Windows 上の Remote API Client の管理

されているディレクトリーに作成します。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存在している場合、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、それに続くクライアント・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、このクライアントで実行するアプリケーションのロギングとトレースで使用されます。これにより、異なるクライアント上の複数のアプリケーションのコピーを区別することができます。



AIX または Linux 上の Remote API Client の管理

UNIX

Remote API Client は、AIX、Linux、Linux for pSeries、または Linux for System 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している必要があります。

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、175 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 上の Remote API Client を使用可能および使用不可に設定する方法

AIX または Linux 上の Remote API Client ソフトウェアを使用可能に設定するには、コマンド・プロンプトに次のコマンドを入力します。

sna start [-t]

クライアントをインストールするときに、インストール・ユーティリティーが自動的にスタートアップ・ファイル **/etc/rc.sna** (AIX) または **/etc/rc.d/init.d/snastart** (Linux) を更新し、**sna start** コマンドを組み込みます。これにより、クライアントがシステム起動時に自動的に開始するようになります。クライアントを自動的に開始しない場合は、この行を除去またはコメント化し、ここで説明する手順のとおりに行ってソフトウェアを手動で使用可能に設定します。

オプションは次の 1 つだけです。

-t クライアント/サーバー・トレースを活動化します。このトレースを使用す

ると、クライアントがサーバーへの接続を試行しているときに発生した問題を診断できます。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。この場合は、コマンド行管理プログラム **snaadmin** を使用して、必要に応じてトレースを活動化できます。

このオプションは、Motif 管理プログラムで「すべてのトレースをオンに設定 (Set all tracing on)」フィールドを選択した場合に相当します (DLC トレースが使用可能に設定されないことを除く)。

トレースをオンにすると、CS Linux コンポーネントのパフォーマンスが低下します。ソフトウェアを使用可能にしたあとで、トレースが必要なくなったときは、コマンド行管理プログラム **snaadmin** を使用してトレースを停止できます。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。

Remote API Client を使用不可にするには、コマンド・プロンプトに次のコマンドを入力します。

```
sna stop
```

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 Remote API Client で使用可能な CS Linux 機能、および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 クライアントにおける同等のファイルについての情報は、147 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。)

さらに、このファイルには、CS Linux がクライアント/サーバー通信に使用する IP ポート番号のセットアップに関する情報が入っています。ほとんどのケースではデフォルト・ポート番号が適切であり、この情報の参照が必要になるのは、同じコンピュータ上の別のプログラムとポート番号の競合が起こったことを示すエラー・メッセージが CS Linux によってログに記録された場合だけです。

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せず、CS Linux LAN 上のサーバーへのアクセスに必要な情報のみを保持します。必要な構成情報の提供は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** に保管されています。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** にあります。このファイルは、クライアントのインストール処理時にセットアップされます。このファイルは ASCII テキスト・ファイルで、あとで必要に応じて標準のテキスト・エディターを使用して変更できます。

注: このファイルの中のパラメーターを変更したあと、クライアントを停止して再始動し、変更を有効にする必要があります。これを行う方法の詳細は、174 ページの『AIX または Linux 上の Remote API Client を使用可能および使用不可に設定する方法』を参照してください。

このファイルの内容は次のとおりです。

```
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
.
.
.
```

次に、ファイルの各行のパラメーターについて説明します。

domain domain_name パラメーターの値は、CS Linux LAN のドメイン名を示します。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。この行は必要です。

maximum_process_count

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 4096 です。

maximum_header_count、*maximum_element_count*

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 1250 と 1800 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

呼び出された TP (RECEIVE_ALLOCATE verb を実行する APPC TP、または Accept_Conversation verb か Accept_Incoming verb を実行する CPI-C アプリケーション) を実行するためにこのクライアントを使用する場合は、`invoked_tps = YES` を指定します。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、100 ページの『TP の定義』を参照してください。

呼び出された TP を実行するためにこのクライアントを使用しない場合は、`invoked_tps = NO` を指定します。

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は NO です。

lan_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 接続をアクティブに保つ時間を秒数で指定します。詳細は、151 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 秒に丸められます)。さらに速く接続を停止するには、クライアント上の CS Linux ソフトウェアを使用不可にします。

タイムアウトを指定せずに、クライアント上で CS Linux ソフトウェアが実行されている限り接続をアクティブにしておくためには、このパラメーターを指定しません。

このパラメーターはオプションで、指定しない場合のデフォルトはタイムアウトなしです。

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。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、アクティビティーがないため、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・メッセージ間の時間は、1 から 1440 (1 分から 1 日) の範囲で、分単位で指定します。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、アクティビティー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

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 (最初のサーバー名の代わりに * を指定)、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。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。

このパラメーターはオプションで、指定しない場合のデフォルトは 5 です。

server_lost_timeout

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場合、またはブロードキャストか指定されたサーバー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の試行を開始または再開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、指定しない場合のデフォルトは 200 秒です。

server names

このクライアントが使用するリソースを含む、1 つ以上の CS Linux サーバーの名前を指定します。Linux servers このリストには、クライアントと同じドメイン内のマスタ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。マスター・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、61 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。またリストには、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含める必要があります。クライアントは、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。

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ことについて詳しくは、「*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、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 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ない場合、このクライアントがアクセスする必要が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。

リストされたサーバーすべての試行に失敗した場合、クライアントは前述の `server_lost_timeout` に指定された時間だけ待ってから、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)。

`sna_clnt.net` のほかに、`server.current` という追加ファイルが、同じディレクトリー (AIX の場合は `/var/sna` Linux の場合は `/var/opt/ibm/sna`) に保管されます。これはテキスト・ファイルで、現在クライアントが接続しているサーバーがある場合はそのサーバーの名前が入っています。このファイルを調べて、クライアントがドメインに接続するための接続点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。

サーバーのドメイン上のクライアントによって使用される SNA LU リソースの追跡

APPC アプリケーションは、タスクを完了させるためにそのパートナー・アプリケーションと 2 回以上の会話が必要になる場合があります。パートナー・アプリケーションがその処理を即時に完了できない場合は特にそのようになります。例えば、アプリケーション A はデータをアプリケーション B に処理のために送信し、この処理は完了にいくらか時間がかかり、その結果を待たずに終了する場合などです。後で、処理を完了したときに、アプリケーション B は結果を返すためにアプリケーション A と新規の会話を開始します。

このように動作するアプリケーションが 2 つ以上のクライアントで実行する場合、CS Linux によって、要求を行った元のクライアントに、新規会話が確実に戻されるようにする必要があります。特に、クライアント・アプリケーションが LU プール内の LU を使用する場合、元の要求に使用される LU との関連付けを維持し、新規会話がこの LU から正しいクライアント・アプリケーションにルーティングできるようにする必要があります。

この機能を使用するために、LU プールを所有するノードに対して、`define_node` コマンドの `ptf_flags` パラメーターで `CLI_OVERWRITE_SYS_NAME` オプションを設定する必要があります。次いで標準テキスト・エディターを使用して、`tpnamehost` という名前のファイルを、AIX 上では `/etc/sna` ディレクトリー、Linux 上では `/etc/opt/ibm/sna` ディレクトリーに作成します。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。

`tpnamehost` ファイルがクライアントに存在している場合は、以下のようになります。

- CS Linux は、この IP ホスト名に一致する、アプリケーションによって使用される LU の `sys_name` を変更します。これにより、パートナー・アプリケーションがこの LU を使用して新規会話を要求する場合、CS Linux は必ずそれを、指定されたクライアント上で実行しているアプリケーション (この LU を使用して元の要求を行ったアプリケーション) と接続します。
- さらに 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、% (パーセント記号) 文字と、それに続くクライアント・コンピューターの IP ホスト名を付加します。この変更された TP 名ストリングは、このクライアントで実行するアプリケーションのロギングとトレースで使用されます。これにより、異なるクライアント上の複数のアプリケーションのコピーを区別することができます。

注: `tpnamehost` ファイルをクライアント上で作成する場合は、`tpnameipv4` という名前のファイルが同じロケーションに存在していないことを確認します。(このファイルの使用については、181 ページの『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』で説明しています。) その両方のファイルが存在している場合は、`tpnameipv4` ファイルが優先します。CS Linux は IP ホスト名ではなく IP アドレスを付加し、クライアント・アプリケーションとその LU との間のアソシエーションは作成しません。

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 別名と、サーバーのドメインの LU 別名のマッピング

2 つ以上の Remote API Client on AIX or Linux 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。(Windows クライアントにおける同等の機能についての情報は、172 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ディングされた LU 別名と、サーバーのドメインの LU 別名のマッピング』を参照してください。)

Remote API Client on AIX or Linux の `lualiasmap.txt` ファイルは、このクライアント上で実行す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 との間をマッピングを定義します。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、それらが異なる LU を使用するように強制することができます。

- 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、これは `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た `lua_luname` パラメーターと、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タイプ 0 から 3 の LU

とをマップします。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は、標準テキスト・エディターを使用して、**lualiasmap.txt** という名前の ASCII テキスト・ファイルを、AIX 上では **/etc/sna** ディレクトリー、Linux 上では **/etc/opt/ibm/sna** ディレクトリーに作成します。

このファイルの各行には、シングル・スペースで分離された 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 で開始する必要があり (最初の 8 文字ストリングに 1 つのスペースが続いた後の位置)、このクライアント上で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で 64 行まで含めることができます。

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。
- LU 名 LUALU3 または LUALU4 を指定するこのクライアント上で稼働する 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 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。

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

Remote API Client on AIX or Linux 上の **tpnameipv4** ファイルは、CS Linux に、クライアント上で実行する APPC または CPI-C アプリケーションによって指定された TP 名を変更するように指示します。これによって各クライアントは (その IPv4 アドレスによって区別される) 固有の TP 名を使用することになります。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。

AIX または Linux 上の Remote API Client の管理

この機能を使用する必要がある場合は、標準テキスト・エディターを使用して、**tpnameipv4** という名前のファイルを、AIX 上では **/etc/sna** ディレクトリー、Linux 上では **/etc/opt/ibm/sna** ディレクトリーに作成します。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存在している場合、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、それに続くクライアント・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、このクライアントで実行するアプリケーションのロギングとトレースで使用されます。これにより、異なるクライアント上の複数のアプリケーションのコピーを区別することができます。



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、100 ページの『TP の定義』または 207 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。これらのワークシートは、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詳しくは、本書の本文中の該当するセクション、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。

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、『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、186 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、196 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、200 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。

- 『APPN ネットワーク・ノード』
- 184 ページの『APPN エンド・ノード』
- 185 ページの『APPN 分岐ネットワーク・ノード』
- 185 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

APPN support	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: 186 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: 200 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		
<i>APPN support</i>	End 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	
接続の構成: 186 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: 200 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		
<i>APPN support</i>	Branch 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	
接続の構成: 186 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: 200 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: 186 ページの『接続ワークシート』を参照		

ノード・ワークシート

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
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: 200 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。

- 『SDLC』
- 188 ページの『トークンリング』
- 190 ページの『イーサネット』
- 192 ページの『QLLC (X.25)』
- 194 ページの『マルチパス・チャネル』
- 195 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>	Negotiable Primary Primary multi-drop Secondary Secondary multi-PU	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
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	
<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>	<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) 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。	
<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)	

接続ワークシート

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<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必要があります。	
	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>Poll address</i>	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。	
	2 桁の 16 進数:	
	<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 	
	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。	

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		
<i>SNA port name</i>	最大 8 文字	
<i>Token ring card number</i>	0 から <i>number_of_cards_minus_1</i>	
<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。	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<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)	
「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		
<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		
<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)	

接続ワークシート

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<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必要があります。	
	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>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 <i>MACADDR=</i> 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 <i>ADPTADR</i> パラメーターです。	
<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	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>Ethernet card number</i>	0 から <i>number_of_cards_minus_1</i>	
<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		
<i>Remote node</i>	<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)	
	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。	
<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)	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<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必要があります。	
	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>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 <i>MACADDR=</i> 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 <i>ADPTADR</i> パラメーターです。	
<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。	

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	
<i>SNA port 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>X.25 card number</i>	0 から <i>number_of_cards_minus_1</i>	
<i>Initially active</i>	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		
<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		
<i>Remote node</i>	<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)	
	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。	
<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 <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ます。	
<i>Remote node ID</i>	8 桁の 16 進数 (オプション)	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<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>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		
<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		
<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	
連絡先情報		

接続ワークシート

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 IP host name</i>	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 <i>protocol</i> 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する場合は、Linux システムは、(ローカル TCP/IP の構成またはドメイン・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。

- 『ローカル・ノード上の DLUR』
- 197 ページの『ダウンストリーム・ノード用のパススルー DLUR』
- 197 ページの『SNA ゲートウェイ』
- 198 ページの『TN サーバー』
- 199 ページの『TN リダイレクター』

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: 183 ページの『ノード・ワークシート』を参照	
	接続の構成: 186 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: .	
<i>PU name</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>DLUS name</i>	NETNAME.LUNAME (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。 NETNAME.LUNAME (それぞれ 1 から 8 個のタイプ A の EBCDIC 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>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: 200 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 (183 ページの『APPN ネットワーク・ノード』を参照)。	
	接続の構成: 186 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: 183 ページの『ノード・ワークシート』を参照	
	接続の構成: 186 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: 200 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	
「Downstream LU (ダウンストリーム LU)」ダイアログ		

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>Downstream LU name</i> (ダウンストリーム LU 名)	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: 183 ページの『ノード・ワークシート』を参照		
接続の構成: 186 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: 200 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	
<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。	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<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 リダイレクター

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: 183 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。	
<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 	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<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。

- 『APPC』
- 203 ページの 『CPI-C』
- 204 ページの 『5250』
- 204 ページの 『3270』
- 205 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: 183 ページの 『ノード・ワークシート』 を参照		
接続の構成: 186 ページの 『接続ワークシート』 を参照		
「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		
<i>LU name</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>LU alias</i>	最大 8 文字	
従属 LU パラメーター		
<i>Host LS/DLUR PU</i>	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)。	

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>LU number</i>	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 <i>LOCADDR</i> パラメーターに一致していなければなりません。	
<i>Member of default pool</i>	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		
<i>Support syncpoint</i>	必要な場合に選択します。	
<i>Disable password substitution</i>	必要な場合に選択します。	
<i>Restrict to specific SSCP</i>	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		
<i>Node's SNA network name</i>	<i>NETNAME.CPNAME</i> (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		
<i>Partner LU 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>Alias</i>	最大 8 文字	
<i>Uninterpreted name</i>	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)	
<i>Supports parallel sessions</i>	サポートされている場合に選択します。	
<i>Location</i>	<i>NETNAME.CPNAME</i> (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		
<i>LU name</i>	1 から 8 個のタイプ A の EBCDIC 文字	
<i>LS name</i>	最大 8 文字	
<i>Partner LU 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>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	

ユーザー・アプリケーション・サポート・ワークシート

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<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		
<i>TP name</i>	ユーザー・アプリケーション: 最大 64 個の ASCII 文字	
<i>Restrict to specific LU</i>	サービス TP: 最大 8 桁の 16 進数 必要な場合に選択します。	
<i>LU alias</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>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 の場合を選択します。	
<i>Full path to TP executable Arguments</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「TP name」)	
<i>User ID</i>	実行可能ファイルに渡す有効な引数	
<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	
<i>Conversation level security required</i>	サービス TP: 最大 8 桁の 16 進数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>	セキュリティー・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>	セキュリティー・アクセス・リストの名前	
<i>Conversation type</i>	Basic	
	Mapped	
<i>Sync level</i>	Either	
	None	
	Confirm	
	Sync-point	
	None または Confirm	
<i>PIP allowed</i>	None、Confirm、または Sync-point 必要な場合に選択します。	
会話セキュリティー・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、会話セキュリティーが必要な場合に限り必要		
<i>User ID</i>	最大 10 文字	
<i>Password</i>	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。

ユーザー・アプリケーション・サポート・ワークシート

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: 183 ページの『ノード・ワークシート』を参照		
接続の構成: 186 ページの『接続ワークシート』を参照		
APPC の構成: 200 ページの『APPC』を参照		
CPI-C 宛先ダイアログ		
<i>Symbolic destination name</i>	1 から 8 文字	
<i>Local LU</i>	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)	
<i>Partner LU</i>	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)	
<i>Mode</i>	タイプ A の EBCDIC スtring	
<i>Partner TP name</i>	ユーザー・アプリケーション: 最大 64 文字	
<i>Security</i>	サービス TP: 最大 8 桁の 16 進数 None Same	
<i>User ID</i>	Program 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)	
<i>Password</i>	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: 183 ページの『ノード・ワークシート』を参照		
接続の構成: 186 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)		
APPC の構成: 200 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: 183 ページの『ノード・ワークシート』を参照		
接続の構成: 186 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)		

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
「LU Type 0-3 (LU タイプ 0 から 3)」ダイアログ		
<i>LU name</i>	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)	
<i>Host LS/DLUR PU</i>	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)。	
<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>LU type</i>	この値は、VTAM/NCP LU リソース定義ステートメントの中の <i>LOCADDR</i> パラメーターに一致していなければなりません。 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)	
<i>LU in pool</i>	SCS Printer (SCS プリンター) 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)	
<i>Pool name</i>	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		
<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	

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: 183 ページの『ノード・ワークシート』を参照		
接続の構成: 186 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		
<i>LU name</i>	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)	

ユーザー・アプリケーション・サポート・ワークシート

Motif フィールド	有効なエントリー / 注	ユーザーの設定値
<i>Host LS/DLUR PU</i>	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)。	
<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)	
	この値は、VTAM/NCP LU リソース定義ステートメントの中の <i>LOCADDR</i> パラメーターに一致していなければなりません。	
<i>LU type</i>	Unrestricted	
<i>LU in pool</i>	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)	
<i>Pool name</i>	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		
<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	

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作成者が、呼び出し可能 TP を定義するために使用でき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、サーバー上でもクライアント上でも実行できます。

コマンドの構文は、TP 定義を定義するのか、除去するのか、照会するのかによって異なり、また Windows 上の Remote API Client でも異なります。

WINDOWS

tpinst32 コマンドを使用する前に、クライアント・ソフトウェアがインストールされたディレクトリーに移動します。

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。この情報を異なる言語で表示したい場合、希望する言語に対応したサブディレクトリーに移動後、コマンドを実行します。例えば、情報をフランス語で表示する場合は、クライアント・ソフトウェアがインストールされたディレクトリー下の **fr_FR** サブディレクトリーに移動します。

呼び出し可能 TP の定義

UNIX

snatpinstall -a file_name

WINDOWS

tpinst32 -a file_name

注: **tpinst32** コマンドは、32 ビット・バージョンと x64 バージョンの両方の Windows に適用されます。

このコマンドは、指定した *file_name* から 1 つ以上の TP 定義を追加します。指定したファイル内に名前が含まれている TP が既に存在している場合は、既存の定義はこのファイル内の情報に置き換えられます。必要なファイル形式については、208 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。

コマンド行からの呼び出し可能 TP の構成

呼び出し可能 TP 定義の除去

UNIX

```
snatpinstall -r -t TP_name [ -l LU_alias ]
```

このコマンドは、指定された TP 名、および指定された LU 別名を持つエントリー (複数の APPC TP が同じ TP 名で定義されている場合は) を除去します。エントリーが CPI-C アプリケーション用である場合、または指定された TP 名に APPC TP が 1 つしか定義されていない場合は、オプション **-l LU_alias** を省略します。

WINDOWS

```
tpinst32 -r -t TP_name
```

このコマンドは、指定した 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 TP_name ] [ -l LU_alias ]
```

このコマンドは、指定された TP 名、および指定された LU 別名を持つエントリー (複数の APPC TP が同じ TP 名で定義されている場合は) を照会します。エントリーが CPI-C アプリケーション用である場合、または指定された TP 名に APPC TP が 1 つしか定義されていない場合は、オプション **-l LU_alias** を省略します。オプション **-t TP_name** を指定しない場合、コマンドはすべての呼び出し可能 TP 定義を照会します。

WINDOWS

```
tpinst32 -q -t TP_name
```

このコマンドは、指定した TP 名を持つエントリーを照会します。オプション **-t TP_name** を指定しない場合、コマンドはすべての呼び出し可能 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。

呼び出し可能 TP 定義のファイル形式

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 ストリング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 [<53504E414D45F1>])、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 ストリングを EBCDIC に変換しますが、16 進数ストリングについては (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 次のいずれかの値を指定します。

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無い限り、呼び出しは失敗します。タイムアウト値として -1 を指定した場合は、タイムアウトは発生せず、呼び出しは着呼割り振り要求が受信されるまで無期限の待機状態になります。

呼び出し可能 TP 定義のファイル形式

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) を指定した場合は、デフォルトとして -1 (無期限) が使用されます。

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 つ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 に設定した場合、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 *setuid* および *setgid* 許可が設定されていて CS Linux による自動開始が可能になっていなければなりません。

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、このパラメーターは使用しないでください。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 LU 別名 (ブランクの LU 別名でも) 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。

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。

呼び出し可能 TP 定義のファイル形式

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いない場合は、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、それを越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。

呼び出し可能 TP 定義のファイル形式

付録 C. DDDLU 用の TN3270 LU モデルの構成

TN3270 クライアントが 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、クライアントに必要な 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。Communications Server for Linux は、通常、標準マッピングを使用して、クライアントが指定した端末型式 (装置タイプ) から LU モデルを確定します。

TN3270 装置タイプと LU モデル間のマッピングを変更する必要がある場合、**tn3270dev.dat** ファイルを使用して変更が可能です。このファイルのサンプル版は、`/opt/ibm/sna/samples` にあります。このファイルを `/etc/opt/ibm/sna` にコピー後、ここで **vi** など標準の ASCII テキスト・エディターを使用して変更します。変更は、次の CS Linux 再始動時に反映されます。

ファイルの各行は、この装置用にホストに送信される、TN3270 装置と LU モデル・ストリング間のマッピングを示します。各行はスペースで区切られ、以下の項目で構成されます。

- 最初の項目は、このマッピングが TN3270 拡張機能を使用する TN3270E クライアント用の場合は Y 1 文字となり、標準の TN3270 クライアント用の場合は N 1 文字となります。
- 2 番目の項目は、クライアントが指定した端末型式 (装置タイプ) です。この項目は最大 40 文字のテキスト・ストリングで、有効な文字は A から Z、数値の 0 から 9、`/`、および から です。
- 3 番目の項目は、7 文字の LU モデルのストリングで、CS Linux がホストに送信して、このクライアント用の正確な LU モデルが識別されます。

CS Linux で提供されるサンプル・ファイルには、TN3270 および TN3270E の両方のバージョンに 18 の標準装置タイプ用のマッピング (合計で 36 エントリー) が含まれます。

- これらの 1 つ以上の標準装置に、異なる LU モデルのストリングの指定が必要な場合は、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ます。
- サンプル・ファイルに含まれないその他の装置タイプをサポートする必要がある場合は、上記で説明した形式を使用して、各装置タイプ用の新しい行をファイルに追加します。

DDDLU 用に TN3270 LU モデルの構成

付録 D. 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あり、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。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ピー・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。

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 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証できません。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。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

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。

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めかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

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は、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. _年を入れる_.

商標

IBM、IBM ロゴおよび ibm.com[®] は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Java[™] およびすべての Java 関連の商標およびロゴは Oracle やその関連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。

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。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、および Pentium 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。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT 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。

他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。

参考文献

以下の IBM 資料には、このライブラリーで説明しているトピックに関する情報が記載されています。資料は、次のトピック別に大きく分けてあります。

- CS Linux バージョン 7.0
- システム・ネットワーク体系 (SNA)
- ホスト構成
- z/OS® Communications Server
- 伝送制御プロトコル / インターネット・プロトコル (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記しています。

CS Linux バージョン 7.0 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 に収録されているソフトコピー・ファイルへのアクセスについては、「*IBM Communications Server for Data Center Deployment on Linux 入門*」を参照してください。システムにこれらのソフトコピー・ブックをインストールするには、9 から 15 MB のハード・ディスク・スペースが必要です (インストールする各国語バージョンにより異なります)。

- *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 (GC88-9996-03)

GC88-9997 は、CS Linux for System z 用です。

- *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)

この資料には、熟練した “C” または Java プログラマー向けに、CS Linux CPI コミュニケーション API を使用した SNA トランザクション・プログラムの作成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 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、管理サービス (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、ノード・オペレーター機能 (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* (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 Data Center Deployment on 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

次の資料では、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

次の資料では、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 User's Guide Volume A04* (GC38-0097)
- *3174 Establishment Controller Installation Guide* (GG24-3061)
- *3270 Information Display System 3174 Establishment Controller: Planning Guide* (GA27-3918)
- *OS/390 ハードウェア構成定義 ユーザーズ・ガイド* (SC88-6630)

z/OS Communications Server 関連資料

次の資料では、z/OS Communications Server についての情報を記載しています。

- *z/OS 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 Communications Server: SNA Diagnostics* (Vol 1: GC31-6850, Vol 2: GC31-6851)
- *z/OS VIR6 Communications Server SNA リソース定義解説書* (SC88-8929)

TCP/IP 関連資料

次の資料では、伝送制御プロトコル / インターネット・プロトコル (TCP/IP) ネットワーク・プロトコルについての情報を記載しています。

- *z/OS Communications Server: IP 構成ガイド* (SC88-8926)
- *z/OS 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

次の資料では、X.25 ネットワーク・プロトコルについての情報を記載しています。

- *Communications Server for OS/2 Version 4 X.25 Programming* (SC31-8150)

APPC 関連資料

次の資料では、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 User's Guide* (SC31-6532)
- *APPC Application Suite V1 Administration* (SC31-6533)
- *APPC Application Suite V1 Programming* (SC31-6534)
- *APPC Application Suite V1 Online Product Library* (SK2T-2680)
- *APPC Application Suite Licensed Program Specifications* (GC31-6535)
- *z/OS V1R2.0 Communications Server: APPC Application Suite User's Guide* (SC31-8809)

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。

- *Common Programming Interface Communications CPI-C Reference* (SC26-4399)
- *Communications Server for OS/2 Warp 日本語版 32 ビット アプリケーション・プログラミングの手引き バージョン 6* (SC88-5585)

その他の IBM ネットワーキング資料

次の資料には、CS Linux に関連するその他のトピックについての情報が記載されています。

- *SDLC Concepts* (GA27-3093)
- *Local Area Network Concepts and Products: LAN Architecture* (SG24-4753)
- *Local Area Network Concepts and Products: LAN Adapters, Hubs and ATM* (SG24-4754)
- *Local Area Network Concepts and Products: Routers and Gateways* (SG24-4755)
- *Local Area Network Concepts and Products: LAN Operating Systems and Management* (SG24-4756)
- *IBM Network Control Program Resource Definition Guide* (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]

アプリケーション
プログラム 6
ワークシート 183
アプリケーション・プログラミング・インターフェース (API) 6
イーサネット
ポート構成 68
ワークシート 190
ウィンドウ
説明 47
ツールバー・ボタン 53
ドメイン 47, 48
ノード 47, 50
メニュー 47
リソース 47
リソース項目 52
CPI-C Destination Names (CPI-C 宛先名) 48
LU Pools (LU プール) 48
エスケープ文字, RCF 139
エラー・ログ・ファイル 40
エンド・ノード
サンプル APPN ネットワーク内の 15
説明 5
ディレクトリー 19, 20
APPN 16

[カ行]

カーネル・コンポーネントのトレース 44
カーネル・メモリー限度 44
会話
セキュリティ 116
説明 12
会話セキュリティ
構成方式 116
パラメーター 116
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) 14
仮想ルーティング・ノード (VRN) 28
監査ログ・ファイル 40

完全修飾 LU 名 18
管理
責任 33
ツール 34
管理サービス (MS) 16
管理サービス (MS) API 7
基本会話 12
境界ノード 3
共有アクセス転送機能 (SATF) 26
クライアント
管理 147
ネットワークの要件 149
ネットワーク・データ・ファイル 39
呼び出し可能 TP の構成 207
ARGUMENTS パラメーター 210
PATH パラメーター 210
SECURITY_TYPE パラメーター 214
SERVICE_NAME パラメーター 214
SHOW パラメーター 213
TIMEOUT パラメーター 211
TP の定義 182
TPname パラメーター 209
TYPE パラメーター 210
クライアント構成ユーティリティー、Windows 156
クライアント/サーバー
構成 61
トレース 167, 175
クラスター・コントローラー 4
計画ワークシート 42
形式 1
経路 (route) 13
経路選択 13, 22, 25
検出, リソースの 18
構成
作業 61
従属 LU 85
セキュリティ・アクセス・リスト 116
接続 67
接続ネットワーク 68
ノード 62
パススルー・サービス 121
ファイル 38
ポート 68
APPC セキュリティー 115
APPC 通信 91
CPI-C サイド情報 112
DLC 68
DLUR 134
SNA ゲートウェイ 132

構成 (続き)
TN サーバー・アクセス・レコード 122
TN サーバー・アソシエーション・レコード 126
TN リダイレクター・アクセス・レコード 127
TP 100
構成サーバー 61
除去 62
追加 62
高性能ルーティング (HPR) 13, 22
高速トランスポート・プロトコル (RTP) 13, 22
顧客情報管理システム (CICS) 8
コマンド
構成サーバーの変更 62
コマンド行管理プログラム
クライアントから 58
コマンドのタイプ 58
使用 58
説明 37
ヘルプ 58
混合ネットワーク 2, 31

[サ行]

サーバー
クライアントとの関係 147
使用可能に設定 44
使用不可に設定する 45
除去 62
追加 62
サービス・アクセス・ポイント (SAP) 29
サービス・クラス (COS) 13
サービス・ポイント 137
サービス・ポイント・コマンド機能 (SPCF) 38, 137
作業シート 42
サブエリア SNA 1
サブエリア・ネットワーク
経路選択 13
説明 2
ノード・タイプ 2
例 4
サブエリア・ノード 3
システム・サービス制御点 (SSCP) 9
自動ネットワーク・ルーティング (ANR) 22
自動ネットワーク・ルーティング (automatic network routing: ANR) 13

- 従属 LU
 - 構成 85
 - 説明 11
- 従属 LU サーバー (DLUS) 31
- 従属ノード 3
- 周辺ノード 3
- 使用可能化、CS Linux ソフトウェアの
 - サーバー上の 44
 - 初期化中の問題 45
 - Windows 上の Remote API Client 154
- 使用可能にする、SNA ソフトウェアを
 - AIX または Linux 上の Remote API Client 174
- 状況
 - Windows 上の Remote API Client 155
- 詳細ログ 40
- 使用不可化、CS Linux ソフトウェアの
 - Windows 上の Remote API Client 155
- 使用不可に設定する、ソフトウェアを 46
- 情報管理システム / 仮想記憶 (Information Management System/Virtual Storage) 8
- 診断ツール 39
- 制御データ 10
- 制御点 (CP) 9
- セキュリティー
 - 会話 116
 - セッション 115
 - APPC 115
 - UCF 142, 145
- セキュリティー・アクセス・リスト
 - 構成方式 116
 - パラメーター 117
 - 必要な追加構成 117
 - 目的 116
- セッション
 - 説明 9
 - タイプ 10
 - ルーティング 22
- セッション・セキュリティー
 - 構成方式 115
 - パラメーター 115
 - 必要な追加構成 116
- 接続
 - 構成 67
 - 説明 6
 - 直接 26
 - ワークシート 183, 186
- 接続ネットワーク
 - 構成 68, 69
 - 構成方式 69
 - 説明 14
 - トポロジー情報 23
 - 必要な追加構成 73
 - APPN 28
- ソース TP 12, 100

[夕行]

- ターゲット TP 12, 100
- ダイアログ 47
- 対等通信 1
- 対等ネットワーク 2
 - 経路選択 13
 - ノード・タイプ 5
- タイプ 2.0 ノード 3
- タイプ 2.1 ノード (type 2.1 node) 3
- タイプ 4 ノード 3
- タイプ 5 ノード 3
- ダウンストリーム LU、SNA ゲートウェイ用の
 - 構成方式 132
 - パラメーター 133
 - 必要な追加構成 133
- 端末コントローラー 4
- 端末装置 4
- 中央ログ 40
- 中間セッション・ルーティング (ISR) 22, 26
- 中間ルーティング 26
- 直接接続 26
- 通信コントローラー 3
- 通信コントローラー・ノード 3
- 通信リンク 3
- データ・ファイル
 - クライアント・ネットワーク 39
 - ドメイン構成 38
 - ノード構成 38
 - 呼び出し可能 TP 38
 - SNA ネットワーク 39
 - TP 定義 38
- データ・リンク制御 (DLC) 67
- ディレクトリー
 - エンド・ノード 19, 20
 - ネットワーク・ノード 19, 21
 - CS Linux 実行可能プログラムの 43
 - LEN ノード 18, 19
- 伝送グループ 22
- トークンリング
 - ポート構成 68
 - ワークシート 188
- 独立 LU
 - 構成 91
 - 説明 11
- トポロジー情報 10
 - 接続ネットワーク 23
 - ローカル 23
- トポロジー・ルーティング・サービス (TRS) 22
- トポロジー・データベース更新 (TDU) 24
- ドメイン
 - 構成ファイル 38

- ドメイン (続き)
 - 説明 3
 - 「ドメイン (Domain)」ウィンドウ 48
- ドメイン・ネーム
 - 変更 148
- ドメイン・リソース 61
- トラブルシューティング 39
- トランザクション・プログラム (TP) 6
- トランスポート・ネットワーク 13
- トレース
 - クライアント/サーバー 175
 - LAN 175
- トレース・ファイル 40

[ナ行]

- 内部トレース
 - Windows 上の Remote API Client 169
- ネットワーク
 - 管理 137
 - 混合 31
 - タイプ 2
 - トポロジー・データベース 22
 - ネットワーク管理データ 10
 - ネットワーク・アクセス可能単位 (NAU) 7
 - ネットワーク・アクセス処理 (NAP) 153
 - ネットワーク・アドレス可能単位 7
 - ネットワーク・データ・ファイル
 - 説明 39
 - AIX または Linux 上の Remote API Client 175
 - ネットワーク・ノード
 - サンプル構成 15
 - ディレクトリー 19, 21
 - ネットワーク・ノード・サーバー 5, 15
 - ノード
 - 構成ファイル 38
 - 構成方式 62
 - サブエリア 2
 - 周辺 2
 - 対等 2
 - タイプ 2, 5
 - パラメーター 63
 - 必要な追加構成 63
 - 目的 62
 - ワークシート 183
 - SNA 2
 - ノード・オペレーター機能 (NOF) API 7
 - ノード・リソース 61

[ハ行]

- バージョン、IP アドレス 150
- バージョン番号、NetView 137

パートナー LU 9
 構成方式 96
 パラメーター 97, 98
 必要な追加構成 100
 複数の定義、ワイルドカードによる 96
 別名の定義 96
 リモート・ノードの定義 96
 パス、CS Linux 実行可能プログラムの 43
 パススルー DLUR 83
 パススルー・サービス
 構成 121
 ワークシート 183, 196
 バックアップ・サーバー 147
 バックアップ・マスター・サーバー 61
 汎用データ・ストリーム (GDS) 7
 複数セッション 11
 物理装置 (PU) 7
 物理装置制御点 (PUCP) 9
 プリンター 4
 ブロードキャスト検索 21
 プロトコル 1
 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 29
 分岐ネットワーク・ノード 5, 29
 文書コンテンツ・アーキテクチャー (DCA) 8
 並列セッション 11
 別名、パートナー LU 96
 ヘルプ
 コマンド行管理プログラム 58
 Motif 管理プログラム 57
 ポート
 構成 68, 69
 パラメーター 69, 70, 71, 72, 73
 必要な追加構成 73
 ホスト 3
 ホスト・ノード 3

[マ行]

マスター・サーバー 61, 147
 の指定 148
 マップ式会話 13
 マルチバス・チャンネル
 ワークシート 194
 マルチバス・チャンネル (MPC)
 ポート構成 68
 メディア・アクセス制御 (MAC) 29
 モード 107
 構成 109
 説明 13
 パラメーター 109, 110, 111, 112
 必要な追加構成 112

モード (続き)
 標準 108
 文字、RCF コマンドの中の 139
 問題判別補助プログラム
 概要 39
 ロギング 63

[ヤ行]

ユーザー・アプリケーション・サポート・
 ワークシート 200
 有向検索 21
 要求単位 (RU) 110, 111
 要約ログ 40
 呼び出し可能 TP 12
 データ・ファイル 38
 CS Linux への定義 100
 snatpinstall の使用 207
 呼び出し側 TP 12, 100

[ラ行]

リソースの検出 18
 リソース名 18
 リモート・コマンド機能 (RCF) 38
 リモート・ジョブ入力 (RJE) 8
 リモート・ノード
 構成方式 95
 定義 94
 パートナー LU 96
 必要な追加構成 96
 LU 9
 Node's SNA network name パラメータ
 ー 95
 リンク・ステーション
 構成 73, 74
 説明 6
 パラメーター 75, 76, 77, 78, 79, 80,
 81
 必要な追加構成 81
 リンク・ステーション・ルーティング
 パラメーター 99
 隣接ノード 14
 ローエントリー・ネットワーク
 (LN) ノード 5
 ローカル LU
 構成方式 93
 説明 9
 定義 92
 パラメーター 93, 94
 必要な追加構成 94
 ローカル・トポロジー・データベース 22
 ローカル・ノード
 LU 9
 ロギング 65

ロギング (続き)
 Windows 上の Remote API Client 162
 ログ・ファイル
 構成 63
 タイプ 64
 ログ・メッセージ 40
 論理装置 (LU) 7
 論理レコード 12

[ワ行]

ワークシート 42
 ワイルドカード 96

[数字]

1 次 LU 10
 2 次 LU 10
 3270
 プールの構成 87
 ワークシート 204
 LU 構成 85
 5250
 ワークシート 204

A

Activation パラメーター 75
 Adapter card number パラメーター 69
 AIX または Linux 上の Remote API
 Client
 管理 174
 サーバー名 178
 broadcast_attempt_count 177
 invoked_tps 176
 lan_access_timeout 176
 LU 別名、LUA LU 名、または TP 名
 のオーバーライド 180
 poll_timer 177
 server_lost_timeout 177
 Alias パラメーター 97
 Allow access to specific LU パラメーター
 124
 Allow timeout パラメーター 133
 ANR
 説明 13, 22
 動的再ルーティング 26
 API
 説明 6
 プロプラエタリー 7
 CS Linux に組み込まれた 6
 API トレース
 Windows 上の Remote API Client 165
 APPC
 構成 91

APPC (続き)
 セキュリティー 115
 ワークシート 200

APPCLU
 Windows 上の Remote API Client 170

APPCTPN
 Windows 上の Remote API Client 170

Application System/400 (AS/400) 14

APPN
 エンド・ノード 5, 16, 184
 機能 14
 経路選択 25
 制御点 (Control Point) 17
 接続ネットワーク 28
 説明 1, 14
 ネットワーク 13, 14, 27
 ネットワークの例 15
 ネットワーク・ノード 5, 15, 183
 ノード・タイプ 14
 分岐ネットワーク・ノード 5, 185

APPN support パラメーター 63

Arguments パラメーター 105

Assigned LUs パラメーター 89

AS/400 (Application System/400) 14

Auto-activated sessions パラメーター 110

B

Backup DLUS Name パラメーター 82

BIND 要求 10

Branch link type パラメーター 79

C

CICS (顧客情報管理システム) 8

CN name パラメーター 72

CN (接続ネットワーク) 14

compression supported パラメーター 83, 112

Configure downstream LUs for implicit PU access パラメーター 73

Control point alias パラメーター 63

Control point name パラメーター 63

Conversation level security required パラメーター 106

Conversation type パラメーター 107

COS
 説明 13
 タイプ 108
 目的 107

COS name パラメーター 109

CP (制御点) 9

CP の完全修飾名 18

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)
 サイド情報 112
 ワークシート 203

CPI-C サイド情報
 構成方式 112
 パラメーター 113, 114
 必要な追加構成 114

CP-CP セッション 10

CSVTLBG
 Windows 上の Remote API Client 171

D

DCA (文書コンテンツ・アーキテクチャ) 8

DDDLU
 TN3270 用 LU モデル 217

Define on connection network パラメーター 71, 72

Delayed logon パラメーター 133

Destination host address パラメーター 130

Display LU assigned パラメーター 124

Display LU パラメーター 126

DLC
 構成 67, 69
 構成方式 69
 必要な追加構成 73

DLUR
 構成 134
 説明 31
 必要な追加構成 84
 ワークシート 196

DLUR PU
 構成方式 82
 パラメーター 82, 83

DLUR ダウンストリーム・ノード 83

DLUS
 説明 31

DLUS Name パラメーター 82, 84

Downstream LU name パラメーター 133

Downstream PU Name パラメーター 81, 83

E

EN (エンド・ノード) 5

Enterprise Extender (HPR/IP)
 ポート構成 68
 ワークシート 195

ENV パラメーター 213

Environment パラメーター 106

Ethernet type パラメーター 72

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 104

G

GDS (汎用データ・ストリーム) 7

Group ID パラメーター 105

GROUP パラメーター 212

H

Host LS/DLUR PU パラメーター 86, 93

HPR
 説明 13, 22

HPR supported on implicit links パラメーター 73

HTTPS 153

I

IMS/VS (情報管理システム / 仮想記憶) 8

Initial session limit パラメーター 109

Initial window size パラメーター 110

Initially active パラメーター 70, 83

IP アドレス・フォーマット 150

IP ポート番号 150

IPv4 アドレス 150

IPv6 アドレス 150

ISR 22, 26

L

LAN アクセス・タイムアウト 152

LAN トレース
 クライアント上の 175

LEN ノード
 機能 15
 説明 5, 16
 ディレクトリ 18, 19
 ワークシート 185

Line details パラメーター 70

Link level error recovery on implicit links パラメーター 73

Link station name パラメーター 99

Linux クライアント
 ドメイン・ネーム 176
 maximum_element_count 176
 maximum_header_count 176
 maximum_process_count 176

Linux コマンド 137

Local IP interface パラメーター 72

Local LU alias パラメーター 113
 Local LU name パラメーター 99
 Local LU パラメーター 113, 115
 Local node ID パラメーター 79
 Local SAP number パラメーター 71
 Location パラメーター 98
 LS 73
 LU
 説明 7
 タイプ 7
 LU 0
 説明 8
 LU 1 8
 LU 2 8
 LU 3 8
 LU 6.2
 構成 91
 説明 7
 LU alias パラメーター 93, 103
 LU in pool パラメーター 87
 LU name パラメーター 86, 93
 LU number パラメーター 86, 94, 133
 LU traffic パラメーター 76
 LU type パラメーター 86
 LU タイプ 0 から 3
 構成方式 85
 パラメーター 86, 87
 必要な追加構成 87
 LU プール
 構成方式 88
 定義 87
 パラメーター 88, 89
 表示 88
 LU 別名、オーバーライド 172, 180
 LUA
 構成 85
 プールの構成 87
 ワークシート 205
 LUA LU 名、オーバーライド 172, 180
 LUALIAS パラメーター 213
 LU-LU セッション 10

M

MAC address パラメーター 77
 MAC (メディア・アクセス制御) 29
 Maximum active template instances パラメーター 73
 Maximum RU size パラメーター 111
 Maximum session limit パラメーター 110
 Maximum window size パラメーター 111
 Member of default pool パラメーター 94
 Minimum contention loser sessions パラメーター 110
 Minimum contention winner sessions パラメーター 110

Mode パラメーター 113
 Motif 管理プログラム
 使用 46
 説明 34
 ダイアログ 54, 56
 ツールバー・ボタン 53
 「ドメイン (Domain)」ウィンドウ 48
 ヘルプ 57
 呼び出し 47
 リソース項目 52
 リソース・ウィンドウ 47
 「Node (ノード)」ウィンドウ 50
 MPC
 ポート構成 68
 MPC group パラメーター 78
 MS (管理サービス) 16
 Multiple instances supported パラメーター 103

N

Name パラメーター
 セキュリティ・アクセス・リスト 117
 モード 109
 リンク・ステーション 75
 CPI-C シンボリック宛先 113
 LU プール 88
 NAP (network access process、ネットワーク・アクセス処理) 153
 NAU (ネットワーク・アクセス可能単位) 7
 NetView
 画面表示 138
 コマンド 137
 コマンド入力域のサイズの変更 138
 サービス・ポイント 137
 説明 137
 バージョン番号 137
 プログラム 137
 NN (ネットワーク・ノード) 5
 Node ID パラメーター 63
 「Node (ノード)」ウィンドウ 50
 Node's SNA network name パラメーター 95
 NOF (ノード・オペレーター機能) API 39

P

Parameters are for invocation on any LU パラメーター 103
 Partner LU name パラメーター 97, 99
 Partner LU パラメーター 113, 115
 Partner TP パラメーター 113

Password パラメーター 114, 115, 116
 PIP allowed パラメーター 107
 Poll address パラメーター 76
 Pool name パラメーター 87
 Port number パラメーター 69
 Printer LU assigned パラメーター 124
 Printer LU パラメーター 126
 Protocol パラメーター 72
 PU
 説明 7
 DLUR 用の 81
 PU ID パラメーター 83
 PU Name パラメーター 82
 PUCP (物理装置制御点) 9

Q

QLLC
 ポート構成 68
 ワークシート 192

R

RCF
 機能 38
 コマンド構文 139
 有効な文字 139
 Receive pacing window パラメーター 110
 Remote API Client on Windows
 LU 別名、LUA LU 名、または TP 名のオーバーライド 172
 Remote node ID パラメーター 80
 Remote node name パラメーター 78
 Remote node role パラメーター 80
 Remote node type パラメーター 79
 Remote X.25 address パラメーター 78
 Reset to SNA defined values パラメーター 112
 Restrict access パラメーター 106
 Restrict maximum RU size パラメーター 111
 Retry contacting DLUS indefinitely パラメーター 83
 RJE (リモート・ジョブ入力) 8
 Route incoming Allocates to running TP パラメーター 104
 RTP
 エンドポイント 26
 説明 13, 22
 RU (要求単位) 110, 111

S

SAP number パラメーター 77
SAP (サービス・アクセス・ポイント) 29
SATF
直接接続 26
APPN ネットワークの 28
SDLC
ポート構成 68
ワークシート 186
Secure Sockets Layer (SSL) 124, 128
クライアント認証 125, 129
サーバー認証 125, 129, 131
データ暗号化 125, 129
Security access list パラメーター 107
Security パラメーター 113
SEND 関数 12, 13
Session timeout パラメーター 111
SNA
階層構造 2
基本概念 2
サブエリア 1
説明 1
層 2
ネットワーク 1
ネットワークのタイプ 2
ネットワーク・データ・ファイル 39, 175
APPN の概念 14
SNA port name パラメーター 69, 75
SNA ゲートウェイ
目的 132
ワークシート 197
SNA ネットワーク情報
Windows 上の Remote API Client 156
snaadmin プログラム 37
snanetutil プログラム 148
sna_clnt.net ファイル 175
SPCF
コマンド 140
コマンド構文 139
説明 38, 137
Specify timeout パラメーター 111
SSCP (システム・サービス制御点) 9
SSCP 従属 LU 11
SSCP-LU セッション 10
SSCP-PU セッション 10
Standard error パラメーター 106
Standard input パラメーター 105
Standard output パラメーター 105
start コマンド 44
stop コマンド 46
Support TN3270E パラメーター 123
Supports parallel sessions パラメーター 98
Sync level パラメーター 107

T

TCP/IP port number パラメーター 124, 128, 130
TDU (トポロジー・データベース更新) 24
Telnet client address パラメーター 128
TN サーバー
アクセス・レコード 122, 126
アクセス・レコード・パラメーター 123, 124
アソシエーション・レコード 126
アソシエーション・レコード・パラメーター 126
ワークシート 198
TN リダイレクター
アクセス・レコード 127
アクセス・レコード・パラメーター 128, 130
ワークシート 199
TN3270 client address パラメーター 123
TN3270 クライアント
DDDLU 217
DDDLU 用の LU モデル 217
TP
クライアント 182
構成 100
構成方式 101
説明 6
ソース 12, 100
ターゲット 12, 100
呼び出し 12, 100
呼び出し可能 12, 100
呼び出しパラメーター 102, 103, 104, 105, 106
APPC 定義パラメーター 106, 107
TP name パラメーター 102, 106
TP 構成パラメーター
ENV 213
GROUP 212
LUALIAS 213
USERID, AIX または Linux 212
USERID, Windows 214
TP 名、オーバーライド 172, 180
TRS (トポロジー/レーティング・サービス) 22

U

UCF
許可 142
コマンド構文 139, 142
コマンドの取り消し 144
コマンドの例 143
出力 143
使用 141

UCF (続き)

使用できるコマンド 143
セキュリティー 142, 145
説明 38, 137
デーモン・プログラム 141
ファイルへのアクセス 145
ユーザー 142
ユーザー名 145
有効なコマンド 143
UDP/IP 通信 150
Uninterpreted Name パラメーター 98
UNIX コマンド機能 (UCF) 38
Upstream DLUS name パラメーター 81
Upstream LU name パラメーター 133
Use default LU パラメーター 113
User ID パラメーター 105, 114, 116
USERID パラメーター
AIX または Linux 212
Windows 214
Users in access list パラメーター 117
ux-cancel command 144

V

VRN
説明 28
VTAM (仮想記憶通信アクセス方式) 14

W

Web 管理パッケージ
説明 36
WebSphere Application Server 153
Wildcard partner LU name パラメーター 97
Windows Open Systems Architecture (WOSA) 153
Windows 上の Remote API Client
クライアント構成ユーティリティ 156
クライアント/サーバーのトレース情報 167
構成 156
構成情報 158
サーバー情報 160
使用可能に設定 154
状況 155
使用不可に設定する 155
ドメイン 158
内部トレース情報 169
ロギング情報 162
admin_msg 168
all_api 166
API トレース情報 165
appc 166

Windows 上の Remote API Client (続き)

X

APPCLLU 170
APPCTPN 170
audit_file 164
audit_file_wrap_size 164
audit_logging_enabled 162
backup_audit_file 164
backup_error_file 163
broadcast_attempt_count 159
client_start_timeout 160
cpic 166
CPI-C アプリケーション・データ
170
csv 166
CSV アプリケーション・データ 171
CSVTBLG 171
data 168
datagram 168
error_file 163
error_file_wrap_size 163
exception_logging_enabled 162
file1 165
file1 (CS_tracing) 167
file1 (Internal_tracing) 169
file2 165
file2 (CS_tracing) 167
file2 (Internal_tracing) 169
flip_size 166
flip_size (CS_tracing) 168
flip_size (Internal_tracing) 169
invoked TPs 159
lan_access_timeout 159
log_directory 163
maximum_element_count 158
maximum_header_count 158
maximum_process_count 158
nof 167
poll_timer 159
receive 168
rui 167
send 168
Server1 161
Server2-Server9 162
server_lost_timeout 160
succinct_audits 165
succinct_errors 164
trace_flushing 169
trace_level 169
truncation_length 166

Windows クライアント

ネットワーク・アクセス処理
(NAP) 153

WOSA (Windows Open Systems
Architecture) 153

xsnaadmin プログラム 34



プログラム番号: 5725-H32

Printed in Japan

SC88-9999-04



日本アイ・ビー・エム株式会社

〒103-8510 東京都中央区日本橋箱崎町19-21