

IBM Communications Server for Linux



NOF プログラマーズ・ガイド

バージョン 6.2.3

IBM Communications Server for Linux



NOF プログラマーズ・ガイド

バージョン 6.2.3

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、857 ページの『付録 C. 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 Linux バージョン 6.2.3、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。

IBM 発行のマニュアルに関する情報のページ

<http://www.ibm.com/jp/manuals/>

こちらから、日本語版および英語版のオンライン・ライブラリーをご利用いただけます。また、マニュアルに関するご意見やご感想を、上記ページよりお送りください。今後の参考にさせていただきます。

(URL は、変更になる場合があります)

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： SC31-6778-02
IBM Communications Server for Linux
Node Operator Facility Programmer's Guide
Version 6.2.3

発 行： 日本アイ・ビー・エム株式会社

担 当： ナショナル・ランゲージ・サポート

目次

表	xvii
図	xix
本書について	xxi
本書の対象読者	xxi
本書の使用法	xxii
本書の構成	xxii
表記上の規則	xxii
グラフィックに関する規則	xxiii
関連資料	xxiii
第 1 章 NOF API について	1
NOF API の目的	1
ノード構成ファイル	2
ドメイン構成ファイル	2
呼び出し可能 TP データ・ファイル	3
Communications Server for Linux のコンポーネント	3
クライアント/サーバー操作	4
マスター・サーバーおよびバックアップ・サーバー	5
AIX または Linux クライアント	6
Windows クライアント	7
特定の Communications Server for Linux 機能を管理する NOF verb	7
NOF verb のターゲット (ノードまたはファイル) の管理	7
始めに	8
3270 通信	9
LUA 通信	10
APPC 通信	11
CPI-C 通信	13
HPR RTP 接続の管理	14
SNA ゲートウェイの管理	14
DLUR の管理	15
TN サーバーの管理	16
TN リダイレクターの管理	17
SNA Management Services 機能の管理	17
ホスト NetView プログラムから Communications Server for Linux システムへのアクセス管理	18
診断設定の管理	18
ディレクトリー・エントリーの管理	20
ネットワーク・トポロジーの照会	21
リモート LU への通信パスの検査	22
Communications Server for Linux LAN 上のサーバーおよびクライアントの管理	22
構成ファイルのヘッダー情報の管理	22
Linux リソース使用状況の管理	23
NOF 指示	23
構成の指示	24
SNA ネットワーク・ファイル指示	24
NOF 状況の指示	24

第 2 章 NOF アプリケーションの作成	27
クライアント/サーバーに関する考慮事項	27
AIX または Linux に関する考慮事項	28
AIX または Linux の NOF API エントリー・ポイント	28
NOF アプリケーションのコンパイルとリンク	34
Windows に関する考慮事項	35
Windows の NOF API エントリー・ポイント	35
NOF アプリケーションのコンパイルとリンク	41
ポータブル・アプリケーションの作成	41
NOF verb のターゲット	42
処理モード	43
NOF verb 相互の順序付けと依存関係	44
ノード構成に基づいた NOF の制約事項	45
APPN エンド・ノードと LEN ノードに関する制約事項	45
複数ドメイン・サポート (MDS) に関する制約事項	46
SNA ゲートウェイと DLUR に関する制約事項	46
QUERY_* verb のリスト・オプション	47
1 つ以上のリソースに関する情報の入手	47
要約情報または詳細情報の入手	48
第 3 章 NOF API verb	49
ACTIVATE_SESSION	50
VCB 構造体	50
指定パラメーター	50
戻りパラメーター: セッション起動の成功時	52
戻りパラメーター: パラメーターの検査	52
戻りパラメーター: セッション起動の失敗時	53
戻りパラメーター: その他の場合	53
ADD_BACKUP	53
VCB 構造体	53
指定パラメーター	53
戻りパラメーター: セッション起動の成功時	54
戻りパラメーター: 状態の検査	54
戻りパラメーター: その他の場合	54
ADD_DLC_TRACE	54
VCB 構造体	55
指定パラメーター	56
戻りパラメーター: セッション起動の成功時	57
戻りパラメーター: パラメーターの検査	58
戻りパラメーター: その他の場合	58
APING	58
VCB 構造体	59
指定パラメーター	59
戻りパラメーター: セッション起動の成功時	61
戻りパラメーター: パラメーターの検査	62
戻りパラメーター: 割り振り失敗時	62
戻りパラメーター: 変換失敗時	63
戻りパラメーター: その他の場合	64

CHANGE_SESSION_LIMIT	64	VCB 構造体	85
VCB 構造体	64	指定パラメーター	86
指定パラメーター	64	戻りパラメーター: セッション起動の成功時	90
戻りパラメーター: セッション起動の成功時	66	戻りパラメーター: パラメーターの検査	90
戻りパラメーター: パラメーターの検査	67	戻りパラメーター: 状態の検査	90
戻りパラメーター: 状態の検査	67	戻りパラメーター: その他の場合	91
戻りパラメーター: セッションの割り振りエラー	68	DEFINE_CPIC_SIDE_INFO	91
パラメーター: CNOS 処理エラー	68	VCB 構造体	91
戻りパラメーター: その他の場合	69	指定パラメーター	92
CLOSE_FILE	69	戻りパラメーター: セッション起動の成功時	93
VCB 構造体	69	戻りパラメーター: パラメーターの検査	93
指定パラメーター	69	戻りパラメーター: その他の場合	94
戻りパラメーター: セッション起動の成功時	69	DEFINE_DEFAULT_PU	94
戻りパラメーター: 状態の検査	69	VCB 構造体	94
戻りパラメーター: その他の場合	70	指定パラメーター	94
CONNECT_NODE	70	戻りパラメーター: セッション起動の成功時	95
VCB 構造体	70	戻りパラメーター: その他の場合	95
指定パラメーター	70	DEFINE_DEFAULTS	95
戻りパラメーター: セッション起動の成功時	71	VCB 構造体	95
戻りパラメーター: パラメーターの検査	71	指定パラメーター	95
戻りパラメーター: 状態の検査	72	戻りパラメーター: セッション起動の成功時	96
戻りパラメーター: その他の場合	72	戻りパラメーター: パラメーターの検査	97
DEACTIVATE_CONV_GROUP	72	戻りパラメーター: その他の場合	97
VCB 構造体	72	DEFINE_DIRECTORY_ENTRY	97
指定パラメーター	73	VCB 構造体	97
戻りパラメーター: セッション起動の成功時	73	指定パラメーター	98
戻りパラメーター: パラメーターの検査	74	戻りパラメーター: セッション起動の成功時	99
戻りパラメーター: その他の場合	74	戻りパラメーター: パラメーターの検査	99
DEACTIVATE_LU_0_TO_3	74	戻りパラメーター: その他の場合	100
VCB 構造体	74	DEFINE_DLC	100
指定パラメーター	75	VCB 構造体	100
戻りパラメーター: セッション起動の成功時	75	指定パラメーター	102
戻りパラメーター: パラメーターの検査	75	戻りパラメーター: セッション起動の成功時	107
戻りパラメーター: その他の場合	75	戻りパラメーター: パラメーターの検査	107
DEACTIVATE_SESSION	75	戻りパラメーター: 状態の検査	107
VCB 構造体	75	戻りパラメーター: その他の場合	107
指定パラメーター	76	DEFINE_DLUR_DEFAULTS	108
戻りパラメーター: セッション起動の成功時	77	VCB 構造体	108
戻りパラメーター: パラメーターの検査	77	指定パラメーター	108
戻りパラメーター: その他の場合	78	戻りパラメーター: セッション起動の成功時	109
DEFINE_ADJACENT_LEN_NODE	78	戻りパラメーター: パラメーターの検査	109
VCB 構造体	78	戻りパラメーター: サポートされていない機能	110
指定パラメーター	78	戻りパラメーター: その他の場合	110
戻りパラメーター: セッション起動の成功時	80	DEFINE_DOMAIN_CONFIG_FILE	110
戻りパラメーター: パラメーターの検査	80	VCB 構造体	110
戻りパラメーター: 状態の検査	80	指定パラメーター	111
戻りパラメーター: その他の場合	80	戻りパラメーター: セッション起動の成功時	111
DEFINE_CN	81	戻りパラメーター: その他の場合	111
VCB 構造体	81	DEFINE_DOWNSTREAM_LU	111
指定パラメーター	81	VCB 構造体	112
戻りパラメーター: セッション起動の成功時	83	指定パラメーター	112
戻りパラメーター: パラメーターの検査	83	戻りパラメーター: セッション起動の成功時	113
戻りパラメーター: 状態の検査	84	戻りパラメーター: パラメーターの検査	113
戻りパラメーター: サポートされていない機能	84	戻りパラメーター: 状態の検査	114
戻りパラメーター: その他の場合	85	戻りパラメーター: サポートされていない機能	115
DEFINE_COS	85	戻りパラメーター: その他の場合	115

DEFINE_DOWNSTREAM_LU_RANGE	115	戻りパラメーター: その他の場合	167
VCB 構造体	116	DEFINE_LU62_TIMEOUT	167
指定パラメーター	116	VCB 構造体	167
戻りパラメーター: セッション起動の成功時	117	指定パラメーター	167
戻りパラメーター: パラメーターの検査	117	戻りパラメーター: セッション起動の成功時	168
戻りパラメーター: 状態の検査	118	戻りパラメーター: パラメーターの検査	168
戻りパラメーター: サポートされていない機能	119	戻りパラメーター: その他の場合	169
戻りパラメーター: その他の場合	119	DEFINE_LU_0_TO_3	169
DEFINE_DSPU_TEMPLATE	119	VCB 構造体	169
VCB 構造体	119	指定パラメーター	169
指定パラメーター	120	戻りパラメーター: セッション起動の成功時	172
戻りパラメーター: セッション起動の成功時	121	戻りパラメーター: パラメーターの検査	172
戻りパラメーター: パラメーターの検査	121	戻りパラメーター: 状態の検査	173
戻りパラメーター: 状態の検査	122	戻りパラメーター: その他の場合	173
戻りパラメーター: サポートされていない機能	123	DEFINE_LU_0_TO_3_RANGE	173
戻りパラメーター: その他の場合	123	VCB 構造体	174
DEFINE_FOCAL_POINT	123	指定パラメーター	174
VCB 構造体	123	戻りパラメーター: セッション起動の成功時	177
指定パラメーター	123	戻りパラメーター: パラメーターの検査	177
戻りパラメーター: セッション起動の成功時	124	戻りパラメーター: 状態の検査	178
戻りパラメーター: パラメーターの検査	124	戻りパラメーター: その他の場合	178
戻りパラメーター: サポートされていない機能	125	DEFINE_LU_LU_PASSWORD	179
戻りパラメーター: 置き換え時	125	VCB 構造体	179
戻りパラメーター: 失敗時	125	指定パラメーター	179
戻りパラメーター: その他の場合	126	戻りパラメーター: セッション起動の成功時	180
DEFINE_INTERNAL_PU	126	戻りパラメーター: パラメーターの検査	180
VCB 構造体	126	戻りパラメーター: その他の場合	181
指定パラメーター	126	DEFINE_LU_POOL	181
戻りパラメーター: セッション起動の成功時	128	VCB 構造体	181
戻りパラメーター: パラメーターの検査	129	指定パラメーター	181
戻りパラメーター: 状態の検査	129	戻りパラメーター: セッション起動の成功時	182
戻りパラメーター: サポートされていない機能	129	戻りパラメーター: パラメーターの検査	182
戻りパラメーター: その他の場合	129	戻りパラメーター: 状態の検査	183
DEFINE_LOCAL_LU	130	戻りパラメーター: その他の場合	183
VCB 構造体	130	DEFINE_MODE	183
指定パラメーター	130	VCB 構造体	183
戻りパラメーター: セッション起動の成功時	133	指定パラメーター	184
戻りパラメーター: パラメーターの検査	133	戻りパラメーター: セッション起動の成功時	187
戻りパラメーター: 状態の検査	134	戻りパラメーター: パラメーターの検査	187
戻りパラメーター: その他の場合	134	戻りパラメーター: 状態の検査	189
デフォルト LU	134	戻りパラメーター: その他の場合	189
DEFINE_LS	135	DEFINE_NODE	189
VCB 構造体	135	VCB 構造体	189
指定パラメーター	139	指定パラメーター	190
戻りパラメーター: セッション起動の成功時	159	戻りパラメーター: セッション起動の成功時	201
戻りパラメーター: パラメーターの検査	159	戻りパラメーター: パラメーターの検査	202
戻りパラメーター: 状態の検査	162	戻りパラメーター: 状態の検査	203
戻りパラメーター: その他の場合	163	戻りパラメーター: その他の場合	203
MAC アドレス内のビット順序	163	DEFINE_PARTNER_LU	203
モデム制御文字	163	VCB 構造体	204
DEFINE_LS_ROUTING	164	指定パラメーター	204
VCB 構造体	165	戻りパラメーター: セッション起動の成功時	205
指定パラメーター	165	戻りパラメーター: パラメーターの検査	205
戻りパラメーター: セッション起動の成功時	166	戻りパラメーター: 状態の検査	206
戻りパラメーター: パラメーターの検査	166	戻りパラメーター: その他の場合	206
戻りパラメーター: 状態の検査	166	DEFINE_PORT	206
		VCB 構造体	206

指定パラメーター	211	戻りパラメーター: セッション起動の成功時	251
戻りパラメーター: セッション起動の成功時	221	戻りパラメーター: パラメーターの検査	251
戻りパラメーター: パラメーターの検査	221	戻りパラメーター: その他の場合	252
戻りパラメーター: 状態の検査	222	DEFINE_TP	252
戻りパラメーター: その他の場合	223	VCB 構造体	252
着呼	223	指定パラメーター	253
DEFINE_RCF_ACCESS	223	戻りパラメーター: セッション起動の成功時	255
VCB 構造体	224	戻りパラメーター: パラメーターの検査	255
指定パラメーター	224	戻りパラメーター: 状態の検査	255
戻りパラメーター: セッション起動の成功時	225	戻りパラメーター: その他の場合	256
戻りパラメーター: パラメーターの検査	225	DEFINE_TP_LOAD_INFO	256
戻りパラメーター: その他の場合	225	VCB 構造体	256
DEFINE_RTP_TUNING	225	指定パラメーター	256
VCB 構造体	225	戻りパラメーター: セッション起動の成功時	258
指定パラメーター	226	戻りパラメーター: パラメーターの検査	258
戻りパラメーター: セッション起動の成功時	227	戻りパラメーター: その他の場合	259
戻りパラメーター: パラメーターの検査	227	DEFINE_USERID_PASSWORD	259
戻りパラメーター: その他の場合	227	VCB 構造体	259
DEFINE_SECURITY_ACCESS_LIST	227	指定パラメーター	259
VCB 構造体	228	戻りパラメーター: セッション起動の成功時	260
指定パラメーター	228	戻りパラメーター: パラメーターの検査	261
戻りパラメーター: セッション起動の成功時	229	戻りパラメーター: その他の場合	261
戻りパラメーター: パラメーターの検査	229	DELETE_ADJACENT_LEN_NODE	261
戻りパラメーター: その他の場合	230	VCB 構造体	262
DEFINE_TN3270_ACCESS	230	指定パラメーター	262
VCB 構造体	230	戻りパラメーター: セッション起動の成功時	263
指定パラメーター	231	戻りパラメーター: パラメーターの検査	263
戻りパラメーター: セッション起動の成功時	237	戻りパラメーター: 状態の検査	263
戻りパラメーター: パラメーターの検査	237	戻りパラメーター: その他の場合	263
戻りパラメーター: その他の場合	237	DELETE_BACKUP	263
DEFINE_TN3270_ASSOCIATION	238	VCB 構造体	264
VCB 構造体	238	指定パラメーター	264
指定パラメーター	238	戻りパラメーター: セッション起動の成功時	264
戻りパラメーター: セッション起動の成功時	239	戻りパラメーター: 状態の検査	264
戻りパラメーター: パラメーターの検査	239	戻りパラメーター: その他の場合	265
戻りパラメーター: その他の場合	239	DELETE_CN	265
DEFINE_TN3270_DEFAULTS	239	VCB 構造体	265
VCB 構造体	239	指定パラメーター	265
指定パラメーター	240	戻りパラメーター: セッション起動の成功時	266
戻りパラメーター: セッション起動の成功時	241	戻りパラメーター: パラメーターの検査	266
戻りパラメーター: パラメーターの検査	241	戻りパラメーター: サポートされていない機能	266
戻りパラメーター: その他の場合	241	戻りパラメーター: その他の場合	267
DEFINE_TN3270_EXPRESS_LOGON	241	DELETE_COS	267
VCB 構造体	242	VCB 構造体	267
指定パラメーター	242	指定パラメーター	267
戻りパラメーター: セッション起動の成功時	242	戻りパラメーター: セッション起動の成功時	267
戻りパラメーター: その他の場合	243	戻りパラメーター: パラメーターの検査	267
戻りパラメーター: その他の場合	243	戻りパラメーター: その他の場合	268
DEFINE_TN3270_SSL_LDAP	243	DELETE_CPIC_SIDE_INFO	268
VCB 構造体	243	VCB 構造体	268
指定パラメーター	243	指定パラメーター	268
戻りパラメーター: セッション起動の成功時	244	戻りパラメーター: セッション起動の成功時	269
戻りパラメーター: パラメーターの検査	245	戻りパラメーター: 状態の検査	269
戻りパラメーター: その他の場合	245	戻りパラメーター: その他の場合	269
DEFINE_TN_REDIRECT	245	DELETE_DIRECTORY_ENTRY	269
VCB 構造体	245	VCB 構造体	269
指定パラメーター	246		

指定パラメーター	270	VCB 構造体	283
戻りパラメーター: セッション起動の成功時	270	指定パラメーター	283
戻りパラメーター: パラメーターの検査	270	戻りパラメーター: セッション起動の成功時	283
戻りパラメーター: 状態の検査	271	戻りパラメーター: パラメーターの検査	283
戻りパラメーター: その他の場合	271	戻りパラメーター: 状態の検査	283
DELETE_DLC	271	戻りパラメーター: その他の場合	284
VCB 構造体	271	DELETE_LS_ROUTING	284
指定パラメーター	271	VCB 構造体	284
戻りパラメーター: セッション起動の成功時	272	指定パラメーター	284
戻りパラメーター: パラメーターの検査	272	戻りパラメーター: セッション起動の成功時	285
戻りパラメーター: 状態の検査	272	戻りパラメーター: パラメーターの検査	285
戻りパラメーター: その他の場合	272	戻りパラメーター: 状態の検査	285
DELETE_DOWNSTREAM_LU	272	戻りパラメーター: その他の場合	286
VCB 構造体	272	DELETE_LU62_TIMEOUT	286
指定パラメーター	273	VCB 構造体	286
戻りパラメーター: セッション起動の成功時	273	指定パラメーター	286
戻りパラメーター: パラメーターの検査	273	戻りパラメーター: セッション起動の成功時	287
戻りパラメーター: 状態の検査	273	戻りパラメーター: パラメーターの検査	287
戻りパラメーター: サポートされていない機能	274	戻りパラメーター: 状態の検査	288
戻りパラメーター: その他の場合	274	戻りパラメーター: その他の場合	288
DELETE_DOWNSTREAM_LU_RANGE	274	DELETE_LU_0_TO_3	288
VCB 構造体	274	VCB 構造体	288
指定パラメーター	274	指定パラメーター	288
戻りパラメーター: セッション起動の成功時	275	戻りパラメーター: セッション起動の成功時	288
戻りパラメーター: パラメーターの検査	275	戻りパラメーター: パラメーターの検査	289
戻りパラメーター: 状態の検査	275	戻りパラメーター: 状態の検査	289
戻りパラメーター: サポートされていない機能	276	戻りパラメーター: その他の場合	289
戻りパラメーター: その他の場合	276	DELETE_LU_0_TO_3_RANGE	289
DELETE_DSPU_TEMPLATE	276	VCB 構造体	289
VCB 構造体	276	指定パラメーター	290
指定パラメーター	276	戻りパラメーター: セッション起動の成功時	290
戻りパラメーター: セッション起動の成功時	277	戻りパラメーター: パラメーターの検査	291
戻りパラメーター: パラメーターの検査	278	戻りパラメーター: 状態の検査	291
戻りパラメーター: その他の場合	278	戻りパラメーター: その他の場合	291
DELETE_FOCAL_POINT	278	DELETE_LU_LU_PASSWORD	291
VCB 構造体	278	VCB 構造体	292
指定パラメーター	278	指定パラメーター	292
戻りパラメーター: セッション起動の成功時	279	戻りパラメーター: セッション起動の成功時	292
戻りパラメーター: パラメーターの検査	279	戻りパラメーター: パラメーターの検査	292
戻りパラメーター: サポートされていない機能	280	戻りパラメーター: その他の場合	293
戻りパラメーター: その他の場合	280	DELETE_LU_POOL	293
DELETE_INTERNAL_PU	280	VCB 構造体	293
VCB 構造体	280	指定パラメーター	293
指定パラメーター	280	戻りパラメーター: セッション起動の成功時	294
戻りパラメーター: セッション起動の成功時	280	戻りパラメーター: パラメーターの検査	294
戻りパラメーター: パラメーターの検査	280	戻りパラメーター: その他の場合	294
戻りパラメーター: 状態の検査	281	DELETE_MODE	294
戻りパラメーター: サポートされていない機能	281	VCB 構造体	295
戻りパラメーター: その他の場合	281	指定パラメーター	295
DELETE_LOCAL_LU	281	戻りパラメーター: セッション起動の成功時	295
VCB 構造体	281	戻りパラメーター: パラメーターの検査	295
指定パラメーター	282	戻りパラメーター: その他の場合	296
戻りパラメーター: セッション起動の成功時	282	DELETE_PARTNER_LU	296
戻りパラメーター: パラメーターの検査	282	VCB 構造体	296
戻りパラメーター: その他の場合	282	指定パラメーター	296
DELETE_LS	283	戻りパラメーター: セッション起動の成功時	296
		戻りパラメーター: パラメーターの検査	296

戻りパラメーター: その他の場合	297	DISCONNECT_NODE	311
DELETE_PORT	297	VCB 構造体	311
VCB 構造体	297	指定パラメーター	311
指定パラメーター	297	戻りパラメーター: セッション起動の成功時	311
戻りパラメーター: セッション起動の成功時	297	戻りパラメーター: 状態の検査	311
戻りパラメーター: パラメーターの検査	297	戻りパラメーター: その他の場合	312
戻りパラメーター: 状態の検査	298	INIT_NODE	312
戻りパラメーター: その他の場合	298	VCB 構造体	312
DELETE_RCF_ACCESS	298	指定パラメーター	312
VCB 構造体	298	戻りパラメーター: セッション起動の成功時	312
指定パラメーター	299	戻りパラメーター: パラメーターの検査	313
戻りパラメーター: セッション起動の成功時	299	戻りパラメーター: 状態の検査	313
戻りパラメーター: その他の場合	299	戻りパラメーター: その他の場合	314
DELETE_SECURITY_ACCESS_LIST	299	INITIALIZE_SESSION_LIMIT	314
VCB 構造体	299	VCB 構造体	314
指定パラメーター	300	指定パラメーター	314
戻りパラメーター: セッション起動の成功時	300	戻りパラメーター: セッション起動の成功時	316
戻りパラメーター: パラメーターの検査	300	戻りパラメーター: パラメーターの検査	316
戻りパラメーター: その他の場合	301	戻りパラメーター: 状態の検査	317
DELETE_TN3270_ACCESS	301	戻りパラメーター: セッションの割り振りエラー	317
VCB 構造体	301	パラメーター: CNOS 処理エラー	318
指定パラメーター	301	戻りパラメーター: その他の場合	318
戻りパラメーター: セッション起動の成功時	303	OPEN_FILE	318
戻りパラメーター: パラメーターの検査	303	VCB 構造体	319
戻りパラメーター: その他の場合	303	指定パラメーター	319
DELETE_TN3270_ASSOCIATION	303	戻りパラメーター: セッション起動の成功時	319
VCB 構造体	303	戻りパラメーター: パラメーターの検査	320
指定パラメーター	304	戻りパラメーター: 状態の検査	320
戻りパラメーター: セッション起動の成功時	304	戻りパラメーター: その他の場合	321
戻りパラメーター: パラメーターの検査	304	PATH_SWITCH	321
戻りパラメーター: 状態の検査	304	VCB 構造体	321
戻りパラメーター: その他の場合	304	指定パラメーター	322
DELETE_TN_REDIRECT	305	戻りパラメーター: セッション起動の成功時	322
VCB 構造体	305	戻りパラメーター: パラメーターの検査	322
指定パラメーター	305	戻りパラメーター: 状態の検査	322
戻りパラメーター: セッション起動の成功時	306	戻りパラメーター: パス・スイッチが使用不可の 場合	322
戻りパラメーター: パラメーターの検査	306	戻りパラメーター: パス・スイッチに障害がある 場合	323
戻りパラメーター: その他の場合	307	戻りパラメーター: ノード検査	323
DELETE_TP	307	戻りパラメーター: その他の場合	323
VCB 構造体	307	QUERY_ACTIVE_TRANSACTION	323
指定パラメーター	307	VCB 構造体	323
戻りパラメーター: セッション起動の成功時	307	指定パラメーター	324
戻りパラメーター: パラメーターの検査	307	戻りパラメーター: セッション起動の成功時	325
戻りパラメーター: その他の場合	308	戻りパラメーター: パラメーターの検査	327
DELETE_TP_LOAD_INFO	308	戻りパラメーター: サポートされていない機能	327
VCB 構造体	308	戻りパラメーター: その他の場合	327
指定パラメーター	308	QUERY_ADJACENT_NN	328
戻りパラメーター: セッション起動の成功時	308	VCB 構造体	328
戻りパラメーター: パラメーターの検査	308	指定パラメーター	328
戻りパラメーター: その他の場合	309	戻りパラメーター: セッション起動の成功時	329
DELETE_USERID_PASSWORD	309	戻りパラメーター: パラメーターの検査	330
VCB 構造体	309	戻りパラメーター: サポートされていない機能	331
指定パラメーター	309	戻りパラメーター: その他の場合	331
戻りパラメーター: セッション起動の成功時	310	QUERY_AVAILABLE_TP	331
戻りパラメーター: パラメーターの検査	310		
戻りパラメーター: その他の場合	311		

VCB 構造体	331	VCB 構造体	360
指定パラメーター	332	指定パラメーター	360
戻りパラメーター: セッション起動の成功時	333	戻りパラメーター: セッション起動の成功時	361
戻りパラメーター: パラメーターの検査	334	戻りパラメーター: パラメーターの検査	365
戻りパラメーター: その他の場合	334	戻りパラメーター: その他の場合	365
QUERY_BUFFER_AVAILABILITY	334	QUERY_CPIC_SIDE_INFO	365
VCB 構造体	335	VCB 構造体	366
指定パラメーター	335	指定パラメーター	366
戻りパラメーター: セッション起動の成功時	335	戻りパラメーター: セッション起動の成功時	367
戻りパラメーター: その他の場合	337	戻りパラメーター: パラメーターの検査	369
QUERY_CENTRAL_LOGGER	337	戻りパラメーター: 状態の検査	370
VCB 構造体	337	戻りパラメーター: その他の場合	370
指定パラメーター	337	QUERY_CS_TRACE	370
戻りパラメーター: セッション起動の成功時	338	VCB 構造体	370
戻りパラメーター: 状態の検査	338	指定パラメーター	371
戻りパラメーター: その他の場合	338	戻りパラメーター: セッション起動の成功時	371
QUERY_CENTRAL_LOGGING	338	戻りパラメーター: パラメーターの検査	372
VCB 構造体	338	戻りパラメーター: その他の場合	372
指定パラメーター	339	QUERY_DEFAULT_PU	373
戻りパラメーター: セッション起動の成功時	339	VCB 構造体	373
戻りパラメーター: パラメーターの検査	339	指定パラメーター	373
状態の検査	339	戻りパラメーター: セッション起動の成功時	373
戻りパラメーター: その他の場合	339	戻りパラメーター: ノードが始動しない場合	374
QUERY_CN	340	戻りパラメーター: その他の場合	374
VCB 構造体	340	QUERY_DEFAULTS	374
指定パラメーター	341	VCB 構造体	374
戻りパラメーター: セッション起動の成功時	341	指定パラメーター	374
戻りパラメーター: パラメーターの検査	344	戻りパラメーター: セッション起動の成功時	374
戻りパラメーター: サポートされていない機能	344	戻りパラメーター: ノードが始動しない場合	375
戻りパラメーター: その他の場合	344	戻りパラメーター: その他の場合	375
QUERY_CN_PORT	344	QUERY_DIRECTORY_ENTRY	376
VCB 構造体	345	VCB 構造体	376
指定パラメーター	345	指定パラメーター	377
戻りパラメーター: セッション起動の成功時	346	戻りパラメーター: セッション起動の成功時	379
戻りパラメーター: パラメーターの検査	347	戻りパラメーター: パラメーターの検査	383
戻りパラメーター: サポートされていない機能	347	戻りパラメーター: その他の場合	384
戻りパラメーター: その他の場合	348	QUERY_DIRECTORY_LU	384
QUERY_CONVERSATION	348	VCB 構造体	384
VCB 構造体	348	指定パラメーター	385
指定パラメーター	348	戻りパラメーター: セッション起動の成功時	386
戻りパラメーター: セッション起動の成功時	350	戻りパラメーター: パラメーターの検査	388
戻りパラメーター: パラメーターの検査	351	戻りパラメーター: その他の場合	389
戻りパラメーター: その他の場合	352	QUERY_DIRECTORY_STATS	389
QUERY_COS	352	VCB 構造体	389
VCB 構造体	352	指定パラメーター	390
指定パラメーター	353	戻りパラメーター: セッション起動の成功時	390
戻りパラメーター: セッション起動の成功時	354	戻りパラメーター: その他の場合	391
戻りパラメーター: パラメーターの検査	355	QUERY_DLC	391
戻りパラメーター: その他の場合	355	VCB 構造体	391
QUERY_COS_NODE_ROW	355	指定パラメーター	392
VCB 構造体	356	戻りパラメーター: セッション起動の成功時	393
指定パラメーター	356	戻りパラメーター: パラメーターの検査	396
戻りパラメーター: セッション起動の成功時	357	戻りパラメーター: その他の場合	396
戻りパラメーター: パラメーターの検査	359	QUERY_DLC_TRACE	396
戻りパラメーター: その他の場合	359	VCB 構造体	397
QUERY_COS_TG_ROW	360	指定パラメーター	397

戻りパラメーター: セッション起動の成功時	400	QUERY_FOCAL_POINT	441
戻りパラメーター: パラメーターの検査	402	VCB 構造体	441
戻りパラメーター: その他の場合	402	指定パラメーター	442
QUERY_DLUR_DEFAULTS	402	戻りパラメーター: セッション起動の成功時	443
VCB 構造体	402	戻りパラメーター: パラメーターの検査	446
指定パラメーター	403	戻りパラメーター: サポートされていない機能	446
戻りパラメーター: セッション起動の成功時	404	戻りパラメーター: その他の場合	446
戻りパラメーター: サポートされていない機能	404	QUERY_GLOBAL_LOG_TYPE	446
戻りパラメーター: その他の場合	404	VCB 構造体	447
QUERY_DLUR_LU	404	指定パラメーター	447
VCB 構造体	404	戻りパラメーター: セッション起動の成功時	447
指定パラメーター	405	戻りパラメーター: パラメーターの検査	448
戻りパラメーター: セッション起動の成功時	406	戻りパラメーター: その他の場合	448
戻りパラメーター: パラメーターの検査	408	QUERY_ISR_SESSION	449
戻りパラメーター: サポートされていない機能	409	VCB 構造体	449
戻りパラメーター: その他の場合	409	指定パラメーター	450
QUERY_DLUR_PU	409	戻りパラメーター: セッション起動の成功時	452
VCB 構造体	409	戻りパラメーター: パラメーターの検査	457
指定パラメーター	410	戻りパラメーター: サポートされていない機能	457
戻りパラメーター: セッション起動の成功時	412	戻りパラメーター: その他の場合	457
戻りパラメーター: パラメーターの検査	415	QUERY_KERNEL_MEMORY_LIMIT	457
戻りパラメーター: サポートされていない機能	416	VCB 構造体	458
戻りパラメーター: その他の場合	416	指定パラメーター	458
QUERY_DLUS	416	戻りパラメーター: セッション起動の成功時	458
VCB 構造体	417	戻りパラメーター: その他の場合	459
指定パラメーター	417	QUERY_LOCAL_LU	459
戻りパラメーター: セッション起動の成功時	418	VCB 構造体	459
戻りパラメーター: パラメーターの検査	420	指定パラメーター	461
戻りパラメーター: サポートされていない機能	421	戻りパラメーター: セッション起動の成功時	462
戻りパラメーター: その他の場合	421	戻りパラメーター: パラメーターの検査	467
QUERY_DOMAIN_CONFIG_FILE	421	戻りパラメーター: その他の場合	467
VCB 構造体	421	QUERY_LOCAL_TOPOLOGY	468
指定パラメーター	422	VCB 構造体	468
戻りパラメーター: セッション起動の成功時	422	指定パラメーター	469
戻りパラメーター: その他の場合	422	戻りパラメーター: セッション起動の成功時	470
QUERY_DOWNSTREAM_LU	422	戻りパラメーター: パラメーターの検査	474
VCB 構造体	422	戻りパラメーター: その他の場合	474
指定パラメーター	424	QUERY_LOG_FILE	474
戻りパラメーター: セッション起動の成功時	425	VCB 構造体	474
戻りパラメーター: パラメーターの検査	430	指定パラメーター	475
戻りパラメーター: 状態の検査	431	戻りパラメーター: セッション起動の成功時	475
戻りパラメーター: サポートされていない機能	431	戻りパラメーター: パラメーターの検査	476
戻りパラメーター: その他の場合	431	戻りパラメーター: その他の場合	476
QUERY_DOWNSTREAM_PU	432	QUERY_LOG_TYPE	476
VCB 構造体	432	VCB 構造体	477
指定パラメーター	433	指定パラメーター	477
戻りパラメーター: セッション起動の成功時	434	戻りパラメーター: セッション起動の成功時	477
戻りパラメーター: パラメーターの検査	436	戻りパラメーター: その他の場合	478
戻りパラメーター: サポートされていない機能	437	QUERY_LS	478
戻りパラメーター: その他の場合	437	VCB 構造体	478
QUERY_DSPU_TEMPLATE	437	指定パラメーター	482
VCB 構造体	437	戻りパラメーター: セッション起動の成功時	483
指定パラメーター	438	戻りパラメーター: パラメーターの検査	505
戻りパラメーター: セッション起動の成功時	439	戻りパラメーター: その他の場合	505
戻りパラメーター: パラメーターの検査	441	QUERY_LS_ROUTING	505
戻りパラメーター: その他の場合	441	VCB 構造体	505

指定パラメーター	506	戻りパラメーター: セッション起動の成功時	556
戻りパラメーター: セッション起動の成功時	507	戻りパラメーター: パラメーターの検査	557
戻りパラメーター: パラメーターの検査	508	戻りパラメーター: その他の場合	558
戻りパラメーター: その他の場合	508	QUERY_NMVT_APPLICATION	558
QUERY_LU_0_TO_3	508	VCB 構造体	558
VCB 構造体	508	指定パラメーター	558
指定パラメーター	511	戻りパラメーター: セッション起動の成功時	559
戻りパラメーター: セッション起動の成功時	513	戻りパラメーター: パラメーターの検査	560
戻りパラメーター: パラメーターの検査	523	戻りパラメーター: その他の場合	561
戻りパラメーター: その他の場合	523	QUERY_NN_TOPOLOGY_NODE	561
QUERY_LU_LU_PASSWORD	524	VCB 構造体	561
VCB 構造体	524	指定パラメーター	562
指定パラメーター	524	戻りパラメーター: セッション起動の成功時	563
戻りパラメーター: セッション起動の成功時	526	戻りパラメーター: パラメーターの検査	566
戻りパラメーター: パラメーターの検査	527	戻りパラメーター: サポートされていない機能	567
戻りパラメーター: その他の場合	527	戻りパラメーター: その他の場合	567
QUERY_LU_POOL	528	QUERY_NN_TOPOLOGY_STATS	567
VCB 構造体	528	VCB 構造体	567
指定パラメーター	528	指定パラメーター	568
戻りパラメーター: セッション起動の成功時	530	戻りパラメーター: セッション起動の成功時	568
戻りパラメーター: パラメーターの検査	532	戻りパラメーター: サポートされていない機能	571
戻りパラメーター: その他の場合	532	戻りパラメーター: その他の場合	571
QUERY_LU62_TIMEOUT	532	QUERY_NN_TOPOLOGY_TG	571
VCB 構造体	533	VCB 構造体	572
指定パラメーター	533	指定パラメーター	573
戻りパラメーター: セッション起動の成功時	535	戻りパラメーター: セッション起動の成功時	575
戻りパラメーター: パラメーターの検査	536	戻りパラメーター: パラメーターの検査	579
戻りパラメーター: その他の場合	536	戻りパラメーター: サポートされていない機能	580
QUERY_MDS_APPLICATION	536	戻りパラメーター: その他の場合	580
VCB 構造体	536	QUERY_NODE	580
指定パラメーター	537	VCB 構造体	580
戻りパラメーター: セッション起動の成功時	538	指定パラメーター	582
戻りパラメーター: パラメーターの検査	539	戻りパラメーター: セッション起動の成功時	582
戻りパラメーター: サポートされていない機能	539	戻りパラメーター: その他の場合	595
戻りパラメーター: その他の場合	539	QUERY_NODE_ALL	595
QUERY_MDS_STATISTICS	539	VCB 構造体	595
VCB 構造体	539	指定パラメーター	596
指定パラメーター	540	戻りパラメーター: セッション起動の成功時	597
戻りパラメーター: セッション起動の成功時	540	戻りパラメーター: パラメーターの検査	598
戻りパラメーター: サポートされていない機能	541	戻りパラメーター: その他の場合	598
戻りパラメーター: その他の場合	541	QUERY_NODE_LIMITS	598
QUERY_MODE	542	VCB 構造体	599
VCB 構造体	542	指定パラメーター	599
指定パラメーター	543	戻りパラメーター: セッション起動の成功時	599
戻りパラメーター: セッション起動の成功時	545	戻りパラメーター: その他の場合	602
戻りパラメーター: パラメーターの検査	548	QUERY_PARTNER_LU	602
戻りパラメーター: その他の場合	549	VCB 構造体	602
QUERY_MODE_DEFINITION	549	指定パラメーター	603
VCB 構造体	549	戻りパラメーター: セッション起動の成功時	605
指定パラメーター	550	戻りパラメーター: パラメーターの検査	609
戻りパラメーター: セッション起動の成功時	551	戻りパラメーター: その他の場合	609
戻りパラメーター: パラメーターの検査	554	QUERY_PARTNER_LU_DEFINITION	610
戻りパラメーター: その他の場合	555	VCB 構造体	610
QUERY_MODE_TO_COS_MAPPING	555	指定パラメーター	611
VCB 構造体	555	戻りパラメーター: セッション起動の成功時	612
指定パラメーター	556	戻りパラメーター: パラメーターの検査	614

戻りパラメーター: その他の場合	615	VCB 構造体	664
QUERY_PORT	615	指定パラメーター	668
VCB 構造体	615	戻りパラメーター: セッション起動の成功時	669
指定パラメーター	617	戻りパラメーター: パラメーターの検査	680
戻りパラメーター: セッション起動の成功時	618	戻りパラメーター: 状態の検査	680
戻りパラメーター: パラメーターの検査	624	戻りパラメーター: サポートされていない機能	681
戻りパラメーター: その他の場合	625	戻りパラメーター: その他の場合	681
QUERY_PU	625	QUERY_TN3270_ACCESS_DEF	681
VCB 構造体	625	VCB 構造体	681
指定パラメーター	626	指定パラメーター	682
戻りパラメーター: セッション起動の成功時	627	戻りパラメーター: セッション起動の成功時	684
戻りパラメーター: パラメーターの検査	630	戻りパラメーター: パラメーターの検査	687
戻りパラメーター: 状態の検査	630	戻りパラメーター: その他の場合	687
戻りパラメーター: その他の場合	631	QUERY_TN3270_ASSOCIATION	687
QUERY_RAPI_CLIENTS	631	VCB 構造体	688
VCB 構造体	631	指定パラメーター	688
指定パラメーター	632	戻りパラメーター: セッション起動の成功時	689
戻りパラメーター: セッション起動の成功時	632	戻りパラメーター: パラメーターの検査	690
戻りパラメーター: パラメーターの検査	635	戻りパラメーター: その他の場合	690
戻りパラメーター: その他の場合	635	QUERY_TN3270_DEFAULTS	690
QUERY_RCF_ACCESS	635	VCB 構造体	691
VCB 構造体	636	指定パラメーター	691
指定パラメーター	636	戻りパラメーター: セッション起動の成功時	691
戻りパラメーター: セッション起動の成功時	636	戻りパラメーター: その他の場合	692
戻りパラメーター: その他の場合	637	QUERY_TN3270_EXPRESS_LOGON	692
QUERY_RTP_CONNECTION	637	VCB 構造体	692
VCB 構造体	637	指定パラメーター	692
指定パラメーター	639	戻りパラメーター: セッション起動の成功時	693
戻りパラメーター: セッション起動の成功時	640	戻りパラメーター: その他の場合	693
戻りパラメーター: パラメーターの検査	645	QUERY_TN3270_SSL_LDAP	693
戻りパラメーター: その他の場合	645	VCB 構造体	693
QUERY_RTP_TUNING	645	指定パラメーター	694
VCB 構造体	646	戻りパラメーター: セッション起動の成功時	694
指定パラメーター	646	戻りパラメーター: その他の場合	695
戻りパラメーター: セッション起動の成功時	646	QUERY_TN_REDIRECT_DEF	695
戻りパラメーター: その他の場合	647	VCB 構造体	695
QUERY_SECURITY_ACCESS_LIST	647	指定パラメーター	696
VCB 構造体	647	戻りパラメーター: セッション起動の成功時	697
指定パラメーター	648	戻りパラメーター: パラメーターの検査	698
戻りパラメーター: セッション起動の成功時	649	戻りパラメーター: その他の場合	698
戻りパラメーター: パラメーターの検査	650	QUERY_TN_SERVER_TRACE	698
戻りパラメーター: その他の場合	651	VCB 構造体	698
QUERY_SESSION	651	指定パラメーター	698
VCB 構造体	651	戻りパラメーター: セッション起動の成功時	699
指定パラメーター	653	戻りパラメーター: その他の場合	699
戻りパラメーター: セッション起動の成功時	655	QUERY_TP	699
戻りパラメーター: パラメーターの検査	660	VCB 構造体	700
戻りパラメーター: その他の場合	660	指定パラメーター	700
QUERY_SNA_NET	660	戻りパラメーター: セッション起動の成功時	701
VCB 構造体	661	戻りパラメーター: パラメーターの検査	702
指定パラメーター	661	戻りパラメーター: その他の場合	703
戻りパラメーター: セッション起動の成功時	662	QUERY_TP_DEFINITION	703
戻りパラメーター: パラメーターの検査	663	VCB 構造体	703
戻りパラメーター: 状態の検査	663	指定パラメーター	704
戻りパラメーター: その他の場合	664	戻りパラメーター: セッション起動の成功時	705
QUERY_STATISTICS	664	戻りパラメーター: パラメーターの検査	708

戻りパラメーター: その他の場合	708	戻りパラメーター: その他の場合	732
QUERY_TP_LOAD_INFO	708	SET_CS_TRACE	732
VCB 構造体	708	VCB 構造体	732
指定パラメーター	709	指定パラメーター	732
戻りパラメーター: セッション起動の成功時	710	戻りパラメーター: セッション起動の成功時	734
戻りパラメーター: パラメーターの検査	711	戻りパラメーター: パラメーターの検査	734
戻りパラメーター: その他の場合	711	戻りパラメーター: その他の場合	734
QUERY_TRACE_FILE	712	SET_GLOBAL_LOG_TYPE	735
VCB 構造体	712	VCB 構造体	735
指定パラメーター	712	指定パラメーター	735
戻りパラメーター: セッション起動の成功時	712	戻りパラメーター: セッション起動の成功時	736
戻りパラメーター: パラメーターの検査	713	戻りパラメーター: パラメーターの検査	737
戻りパラメーター: その他の場合	714	戻りパラメーター: その他の場合	737
QUERY_TRACE_TYPE	714	SET_KERNEL_MEMORY_LIMIT	737
VCB 構造体	714	VCB 構造体	737
指定パラメーター	714	指定パラメーター	737
戻りパラメーター: セッション起動の成功時	714	戻りパラメーター: セッション起動の成功時	738
戻りパラメーター: その他の場合	716	戻りパラメーター: その他の場合	738
QUERY_USERID_PASSWORD	716	SET_LOG_FILE	738
VCB 構造体	716	VCB 構造体	739
指定パラメーター	717	指定パラメーター	739
戻りパラメーター: セッション起動の成功時	717	戻りパラメーター: セッション起動の成功時	741
戻りパラメーター: パラメーターの検査	718	戻りパラメーター: パラメーターの検査	741
戻りパラメーター: その他の場合	719	戻りパラメーター: その他の場合	741
REGISTER_INDICATION_SINK	719	SET_LOG_TYPE	741
VCB 構造体	720	VCB 構造体	742
指定パラメーター	720	指定パラメーター	742
戻りパラメーター: セッション起動の成功時	720	戻りパラメーター: セッション起動の成功時	744
戻りパラメーター: パラメーターの検査	720	戻りパラメーター: パラメーターの検査	744
戻りパラメーター: サポートされていない機能	721	戻りパラメーター: その他の場合	744
戻りパラメーター: その他の場合	721	SET_PROCESSING_MODE	744
REMOVE_DLC_TRACE	721	VCB 構造体	745
VCB 構造体	722	指定パラメーター	745
指定パラメーター	722	戻りパラメーター: セッション起動の成功時	746
戻りパラメーター: セッション起動の成功時	724	戻りパラメーター: パラメーターの検査	746
戻りパラメーター: パラメーターの検査	724	戻りパラメーター: 状態の検査	746
戻りパラメーター: その他の場合	725	戻りパラメーター: その他の場合	747
RESET_SESSION_LIMIT	725	SET_TN_SERVER_TRACE	747
VCB 構造体	725	VCB 構造体	747
指定パラメーター	725	指定パラメーター	747
戻りパラメーター: セッション起動の成功時	727	戻りパラメーター: セッション起動の成功時	748
戻りパラメーター: パラメーターの検査	728	戻りパラメーター: その他の場合	748
戻りパラメーター: 状態の検査	729	SET_TRACE_FILE	748
戻りパラメーター: セッションの割り振りエラー	729	VCB 構造体	749
パラメーター: CNOS 処理エラー	729	指定パラメーター	749
戻りパラメーター: その他の場合	730	戻りパラメーター: セッション起動の成功時	751
SET_BUFFER_AVAILABILITY	730	戻りパラメーター: パラメーターの検査	751
VCB 構造体	730	戻りパラメーター: その他の場合	751
指定パラメーター	730	SET_TRACE_TYPE	751
戻りパラメーター: セッション起動の成功時	730	VCB 構造体	752
戻りパラメーター: その他の場合	731	指定パラメーター	752
SET_CENTRAL_LOGGING	731	戻りパラメーター: セッション起動の成功時	754
VCB 構造体	731	戻りパラメーター: パラメーターの検査	754
指定パラメーター	731	戻りパラメーター: その他の場合	754
戻りパラメーター: セッション起動の成功時	731	トレース・タイプ	754
戻りパラメーター: パラメーターの検査	732	START_DLC	756

VCB 構造体	756
指定パラメーター	756
戻りパラメーター: セッション起動の成功時	756
戻りパラメーター: パラメーターの検査	756
戻りパラメーター: 状態の検査	757
戻りパラメーター: その他の場合	757
START_INTERNAL_PU	757
VCB 構造体	757
指定パラメーター	757
戻りパラメーター: セッション起動の成功時	758
戻りパラメーター: パラメーターの検査	758
戻りパラメーター: 状態の検査	759
戻りパラメーター: 失敗時	759
戻りパラメーター: サポートされていない機能	760
戻りパラメーター: その他の場合	760
START_LS	760
VCB 構造体	760
指定パラメーター	760
戻りパラメーター: セッション起動の成功時	761
戻りパラメーター: パラメーターの検査	761
戻りパラメーター: 状態の検査	761
戻りパラメーター: 失敗時	762
戻りパラメーター: 取り消し時	763
戻りパラメーター: その他の場合	763
START_PORT	763
VCB 構造体	763
指定パラメーター	764
戻りパラメーター: セッション起動の成功時	764
戻りパラメーター: パラメーターの検査	764
戻りパラメーター: 状態の検査	764
戻りパラメーター: 取り消し時	765
戻りパラメーター: その他の場合	765
STOP_DLC	765
VCB 構造体	765
指定パラメーター	765
戻りパラメーター: セッション起動の成功時	766
戻りパラメーター: パラメーターの検査	766
戻りパラメーター: 状態の検査	766
戻りパラメーター: 取り消し時	766
戻りパラメーター: その他の場合	767
STOP_INTERNAL_PU	767
VCB 構造体	767
指定パラメーター	767
戻りパラメーター: セッション起動の成功時	767
戻りパラメーター: パラメーターの検査	768
戻りパラメーター: 状態の検査	768
戻りパラメーター: サポートされていない機能	768
戻りパラメーター: その他の場合	768
STOP_LS	769
VCB 構造体	769
指定パラメーター	769
戻りパラメーター: セッション起動の成功時	770
戻りパラメーター: パラメーターの検査	770
戻りパラメーター: 状態の検査	770
戻りパラメーター: 取り消し時	770
戻りパラメーター: その他の場合	771

STOP_PORT	771
VCB 構造体	771
指定パラメーター	771
戻りパラメーター: セッション起動の成功時	771
戻りパラメーター: パラメーターの検査	772
戻りパラメーター: 状態の検査	772
戻りパラメーター: 取り消し時	772
戻りパラメーター: その他の場合	772
TERM_NODE	773
VCB 構造体	773
指定パラメーター	773
戻りパラメーター: セッション起動の成功時	774
戻りパラメーター: その他の場合	774
UNREGISTER_INDICATION_SINK	774
VCB 構造体	774
指定パラメーター	774
戻りパラメーター: セッション起動の成功時	775
戻りパラメーター: パラメーターの検査	775
戻りパラメーター: サポートされていない機能	775
戻りパラメーター: その他の場合	775

第 4 章 NOF 指示 777

CONFIG_INDICATION	777
VCB 構造体	777
DIRECTORY_INDICATION	778
VCB 構造体	778
パラメーター	779
DLC_INDICATION	782
VCB 構造体	782
パラメーター	782
DLUR_LU_INDICATION	783
VCB 構造体	783
パラメーター	783
DLUR_PU_INDICATION	784
VCB 構造体	784
パラメーター	784
DLUS_INDICATION	786
VCB 構造体	786
パラメーター	787
DOWNSTREAM_LU_INDICATION	788
VCB 構造体	788
パラメーター	789
DOWNSTREAM_PU_INDICATION	792
VCB 構造体	792
パラメーター	792
FOCAL_POINT_INDICATION	794
VCB 構造体	794
パラメーター	795
ISR_INDICATION	796
VCB 構造体	796
パラメーター	797
LOCAL_LU_INDICATION	800
VCB 構造体	800
パラメーター	800
LOCAL_TOPOLOGY_INDICATION	803
VCB 構造体	803

パラメーター	803
LS_INDICATION	805
VCB 構造体	805
パラメーター	806
LU_0_TO_3_INDICATION	809
VCB 構造体	809
パラメーター	810
MODE_INDICATION	813
VCB 構造体	813
パラメーター	813
NN_TOPOLOGY_NODE_INDICATION	814
VCB 構造体	814
パラメーター	814
NN_TOPOLOGY_TG_INDICATION	815
VCB 構造体	815
パラメーター	816
NOF_STATUS_INDICATION	817
VCB 構造体	817
パラメーター	818
PLU_INDICATION	818
VCB 構造体	819
パラメーター	819
PORT_INDICATION	820
VCB 構造体	820
パラメーター	820
PU_INDICATION	821
VCB 構造体	821
パラメーター	822
RAPI_CLIENT_INDICATION	824
VCB 構造体	824
パラメーター	824
REGISTRATION_FAILURE	826
VCB 構造体	827
パラメーター	827
RTP_INDICATION	828
VCB 構造体	828
パラメーター	829
SERVER_INDICATION	832
VCB 構造体	833

パラメーター	833
SESSION_INDICATION	834
VCB 構造体	834
パラメーター	835
SNA_NET_INDICATION	838
VCB 構造体	838
TN_REDIRECTION_INDICATION	839
VCB 構造体	839
パラメーター	839

付録 A. 戻りコード値 843

1 次戻りコード	843
2 次戻りコード	844

付録 B. 共通戻りコード 851

通信サブシステムが活動状態でない場合	851
指示	852
無効な機能	852
無効な verb セグメント	853
パラメーターの検査	853
状態の検査	854
システム・エラー	855

付録 C. 特記事項 857

商標	859
----	-----

参考文献 861

Communications Server for Linux バージョン 6.2.3	
関連資料	861
システム・ネットワーク体系 (SNA) 関連資料	862
ホスト構成関連資料	863
z/OS Communications Server 関連資料	863
TCP/IP 関連資料	863
X.25 関連資料	863
APPC 関連資料	864
プログラミング関連資料	864
その他の IBM ネットワーキング関連資料	864

索引 865

表

1. 表記上の規則	xxii	2. モデム制御文字のエスケープ・シーケンス	164
---------------------	------	------------------------	-----



1. Communications Server for Linux のコンポーネント 4
2. Communications Server for Linux の全体構造 755

本書について

「*Communications Server for Linux NOF プログラマーズ・ガイド*」には、Node Operator Facility (NOF) API を使用して IBM® Communications Server for Linux® リソースを管理する C 言語アプリケーション・プログラムを開発するために必要な情報を記載しています。IBM Communications Server for Linux は Linux を実行しているコンピューターが SNA ネットワーク上の別のノードと情報を交換できるようにする IBM ソフトウェア製品です。

IBM Communications Server for Linux には、作動するハードウェアに応じて、2 つのインストール・バリエーションがあります。

Communications Server for Linux

Communications Server for Linux (プログラム・プロダクト番号 5724-i33) は、以下のハードウェアで作動します。

- Linux (i686) を稼働する 32 ビット Intel® ワークステーション
- Linux (x86_64) を稼働する 64 ビット AMD64/Intel EM64T ワークステーション
- Linux (ppc64) を稼働する IBM pSeries® コンピューター

Communications Server for Linux on System z™

Communications Server for Linux on System z (プログラム・プロダクト番号 5724-i34) は、Linux for System z (s390 または s390x) を稼働する System z メインフレーム上で作動します。

本書では、相違が明示的に記述されていない限り、Communications Server for Linux という名称はこれらの 2 つのバリエーションのいずれかを示すために使用され、

「Communications Server for Linux コンピューター」という用語は、Communications Server for Linux が稼働しているコンピューターのタイプを示す場合に使用されます。

本書は Communications Server for Linux バージョン 6.2.3 に適用されます。

本書の対象読者

本書は Communications Server for Linux を備えたシステム用のシステム・ネットワーク体系 (SNA) トランザクション・プログラムを作成した経験のある C プログラマーを対象としています。

本書は Communications Server for Linux を使用するシステム管理者およびアプリケーション・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 Communications Server for 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、システムを保守します。システム管理者は Communications Server for Linux が稼働するハードウェアおよび Linux オペレーティング・システムについて理解する必要があります。

また、システム管理者はシステムを接続するネットワークに関する知識があり、SNA の一般的な概念を理解していなければなりません。

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、Communications Server for 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・プログラムを設計し、コーディングします。アプリケーション・プログラマーは SNA だけでなく、トランザクション・プログラムまたはアプリケーション・プログラムの通信先となるリモート・プログラム、および AIX® または 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を十分に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ついての詳細は、個々の API の資料に説明があります。

本書の使用法

この節では、本書の構成と表記について説明します。

本書の構成

本書は次の各章で構成されています。

- 1 ページの『第 1 章 NOF API について』では、Communications Server for Linux NOF API の概要と、その機能について説明します。
- 27 ページの『第 2 章 NOF アプリケーションの作成』では、プログラマーが NOF アプリケーションを作成するときに必要な一般的な情報と、アプリケーションのコンパイルとリンクに関する情報を記載します。
- 49 ページの『第 3 章 NOF API verb』では、NOF の各 verb を、パラメーターと戻りコードも含めて詳しく説明します。
- 777 ページの『第 4 章 NOF 指示』では、NOF アプリケーションが受信できるように登録することができる指示について、個々に詳しく説明します。
- 843 ページの『付録 A. 戻りコード値』では、NOF インターフェースの可能性のある戻りコードを番号順にリストして、その意味を説明します。
- 851 ページの『付録 B. 共通戻りコード』では、すべての NOF verb に共通する戻りコードについて説明します。

表記上の規則

表 1 は、本書で使用する表記上の規則を示しています。

表 1. 表記上の規則

内容	表記例
資料名	<i>Communications Server for Linux NOF プログラマーズ・ガイド</i>
ファイル名またはパス名	<code>sna.err</code>
ディレクトリー名	<code>/var/opt/ibm/sna</code>
ヘッダー・ファイル	<code>nof_c.h</code>
プログラムまたはアプリケーション	<code>snaadmin</code>
コマンド	<code>define_local_lu、cd</code>

表 1. 表記上の規則 (続き)

内容	表記例
特定のタイプのすべての verb への一般的な参照	DEFINE_* (リソースを定義するすべての NOF API verb を示す)
オプションまたはフラグ	-I
パラメーター	<i>opcode</i>
リテラル値またはユーザーが入力できる選択項目 (デフォルト値を含む)	255
定数	AP_MODE_READ_ONLY
戻り値	AP_INVALID_FORMAT、0
指定する値を表す変数	<i>a.b.c.d</i>
環境変数	LD_RUN_PATH
プログラミング verb	CONNECT_NODE
ユーザー入力	snaadmin
関数、コール、またはエントリー・ポイント	status_dependent_lu,pu_name=ETH0
データ構造	ioctl
16 進値	NOF_CALLBACK 0x20

グラフィックに関する規則

UNIX

このシンボルは、AIX または Linux オペレーティング・システムにのみ適用されるテキストのセクションの開始を示すために使用されます。これは Linux サーバーおよび AIX、Linux、Linux for pSeries または Linux for System z 上で稼動する IBM Remote API Client に適用されます。

WINDOWS

このシンボルは、Windows® で稼動する IBM Remote API Client に適用されるテキストのセクション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・システム固有テキストのセクションの終了を示します。このシンボルに続く情報は、オペレーティング・システムに関係なく適用されます。

関連資料

SNA、APPN®, または LU 6.2 アーキテクチャーについて詳しくは、以下の IBM 資料を参照してください。

- システム・ネットワーク体系:
 - *APPN Architecture Reference* (SC30-3422)
 - *Format and Protocol Reference Manual: Architectural Logic* (SC30-3112)

関連資料

- *Formats* (GA27-3136)
- *LU6.2 Reference: Peer Protocols* (SC31-6808)
- *Management Services Reference* (SC30-3346)
- *Sessions between Logical Units* (GC20-1868)
- *Technical Overview* (GC30-3073)
- *System/390 Principles of Operation* (SA22-7201)
- *z/Architect Principles of Operation* (SA22-7832)

第 1 章 NOF API について

この章では、Communications Server for Linux NOF API の概要について説明します。ここでは、次の情報を記載しています。

- NOF API の目的
- クライアント/サーバー操作
- NOF verb と指示

NOF API がアクセスする Communications Server for Linux のコンポーネントおよびリソースについては、「*Communications Server for Linux 入門*」を参照してください。

NOF API の目的

Communications Server for Linux NOF API を使用すると、NOF verb と呼ばれるコマンドの標準セットへアクセスできます。NOF verb は、アプリケーション・プログラム内から Communications Server for Linux システムを管理する場合に使用できます。これらの verb を使用すると、リソースの定義と削除、診断レベルやファイル名などの Communications Server for Linux パラメーターを指定すること、定義したリソースを始動および停止すること、リソースの定義または現在の状況を照会することができます。そして、マスター構成ファイル・サーバーを使用できない場合に、Communications Server for Linux LAN 上のどのサーバーをバックアップ・マスターとして機能させるかについて管理することができます。

クライアント/サーバー・システムでは、サーバーで実行中のアプリケーションで、いずれの NOF verb も使用できます。Remote API Client で実行中のアプリケーションは、NOF verb を使用して構成または状況情報を照会できますが、その他の verb を使用して構成を変更したり、リソースを始動および停止することはできません。

NOF verb はコマンド行管理プログラム **snaadmin** へ発行したコマンド、または Linux 構成ファイル内のレコードと同じ機能を提供します。たとえば、**DEFINE_LOCAL_LU** という NOF verb は、**snaadmin** プログラムへ発行した **define_local_lu** コマンドと構成ファイル内の **define_local_lu** レコードの両方と同じ機能を持ちます。これらの 3 種類はいずれも、Communications Server for Linux ローカル APPC LU のパラメーターを指定します。

Motif 管理プログラム **xsnaadmin** を使用すると、NOF verb または管理コマンドと同じ機能（たとえば、ローカル APPC LU の定義）を実行できます。ただし、このプログラムでは、いくつかの NOF verb に含まれるパラメーターの一部にアクセスできない場合があります。Motif 管理プログラムの使用方法について詳しくは、「*Communications Server for Linux 管理ガイド*」を参照してください。

NOF verb は、次のターゲットに対して発行できます。

- 稼働中の Communications Server for Linux ノード — そのノードのリソースを管理したり、そのノードの動作を監視する。

- 実行中でないサーバー — 保管されている構成情報を照会したり、次にそのノードが始動されたときに使用される構成情報を変更する。
- Communications Server for Linux ドメイン全体 — ドメイン・リソース (特定のユーザー・プログラム (たとえば CPI-C サイド情報エントリーなど) をサポートするために使用され、特定のノードへ関連付けられていないリソース) の構成を定義、変更、照会する。
- Communications Server for Linux 呼び出し可能 TP データ・ファイル — 呼び出し可能 (ターゲット) TP を開始するために Communications Server for Linux が必要とする情報を定義する。または TP に関連する他の情報 (TP へアクセスするために必要なセキュリティー・レベルなど) を定義する。

NOF API を使用すると、次のことができます。

- Communications Server for Linux システムを管理する独自のアプリケーション・プログラムを開発する。
- 独自のリソースも管理できるように他の Communications Server for Linux API を使用するアプリケーション・プログラムを開発する (たとえば、ある APPC アプリケーションで、会話の割り振りを行う前にパートナー TP への通信リンクが活動状態であるかどうかを検査したり、パートナー TP が配置されているリモート LU を定義したりできます)。

ノード構成ファイル

Communications Server for Linux ノードの構成情報はノードごとに、そのノードが実行されるコンピューターのテキスト・ファイルに保持されています。このファイルには、ノードのリソースに関する情報が入っており、Communications Server for Linux を始動したときに、どのリソースを活動状態にするかを指定します。ノードの始動時に、使用可能なリソースの初期定義がこのファイルによって提供されます。要件が変更されたときには、NOF API または Communications Server for Linux 管理ツールを使用して、実行中のノードのリソースを変更できます。

複数の構成ファイルをセットアップし、異なる時点で使用するさまざまな Communications Server for Linux の構成を保管しておき、Linux ソフトウェアの起動時に、使用するファイルをその中から選択することができます。

APPN ネットワーク内の構成は動的に処理されます。ユーザーは Communications Server for 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、変更することができます。構成ファイルにより使用可能なリソースの初期定義が提供されます。また、現行の定義を保管して、ノードを再始動する必要があるときに、その定義を再使用できます。しかし、Communications Server for Linux ソフトウェアを起動する前に構成全体を定義する必要はありません。

ドメイン構成ファイル

Communications Server for Linux ドメイン・リソースに関する構成情報は、マスター・サーバーの 1 つのテキスト・ファイルに保持されています。複数のドメイン構成ファイルをセットアップし、異なる時点で使用するさまざまな Communications Server for Linux の構成を保管しておき、マスター・サーバー上で Communications Server for Linux ソフトウェアの起動時に、使用するファイルをその中から選択することができます。

APPN ネットワーク内の構成は動的に処理されます。ユーザーはシステムの実行中に、必要に応じてリソースを追加、削除、変更することができます。ドメイン構成ファイルにより使用可能なドメイン・リソースの初期定義を提供します。また、現行の定義を保管して、システムを再始動する必要があるときに、その定義を再使用できます。しかし、Communications Server for Linux ソフトウェアを起動する前にドメイン構成全体を定義したり、変更を加えたときに、このソフトウェアを再起動する必要はありません。

呼び出し可能 TP データ・ファイル

呼び出し可能 (ターゲット) TP を開始するために Communications Server for Linux が必要とする情報は、ファイル `/etc/sna/sna_tps` (AIX) または `/etc/opt/ibm/sna/sna_tps` (Linux) にあります。このファイルには他の情報 (TP ヘアアクセスするために必要なセキュリティー・レベルなど) も含まれます。呼び出し可能 TP データ・ファイルは、TP が実行されているコンピューターにあります。

Communications Server for Linux のコンポーネント

Communications Server for Linux は他のノードと通信するため APPN ノードを SNA ネットワークにインプリメントします。これにより、3270 通信および LUA 通信に論理装置 (LU) 0、1、2、3 をサポートするのみでなく、APPC 機能、CPI-C 機能に LU 6.2 をサポートします。

Communications Server for Linux は、構成に応じて LEN ノード、エンド・ノード、ネットワーク・ノード、または分岐ネットワーク・ノードのいずれの APPN ノード・タイプとしても動作可能です。一部の機能は、APPN アーキテクチャーによって定義されたとおり、特定のノード・タイプでのみサポートされます。本書では、必要に応じてこれらの相違点を示します。相違点が示されていないければ、すべてのノード・タイプに情報を適用できます。

4 ページの図 1 は Communications Server for Linux のコンポーネントと、それらが連携して機能する様子を示しています。

Communications Server for Linux のコンポーネント

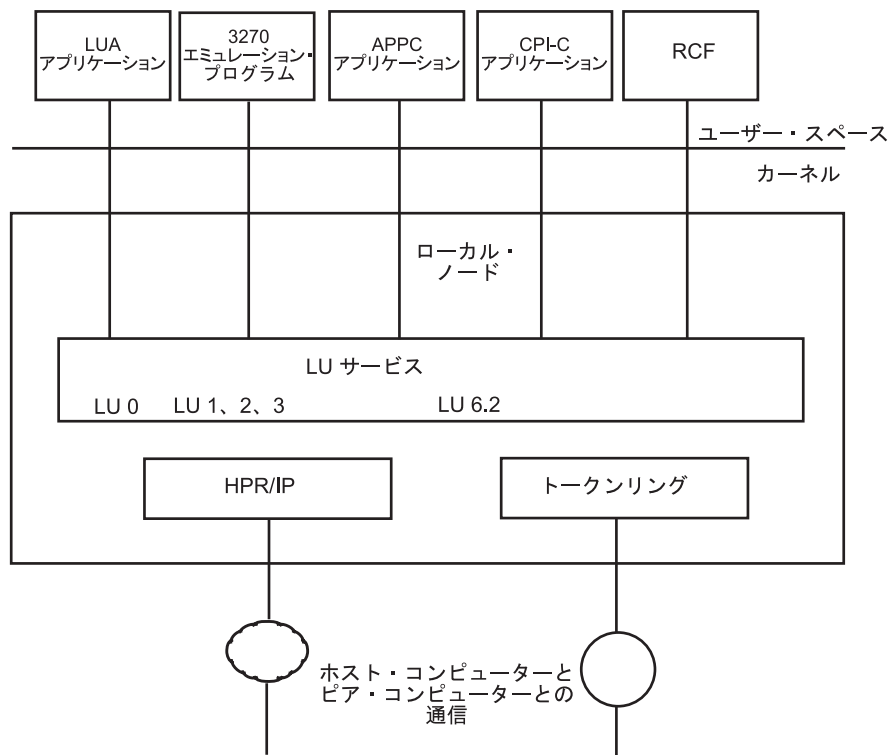


図1. Communications Server for Linux のコンポーネント

ローカル・ノードは、それに関連した接続リソース (DLC、ポート、LS) も含め、Communications Server for Linux のコンポーネントとして Linux システムのカーネル内にインプリメントされます。

APPC トランザクション・プログラム、CPI-C アプリケーション、LUA アプリケーション、およびリモート・コマンド機能 (RCF) は、ユーザー・スペース・プログラムです。Communications Server for Linux は、同時に実行されている複数の APPC TP、CPI-C アプリケーション、および LUA アプリケーションをサポートします。

クライアント/サーバー操作

Communications Server for Linux 上のコンピューターには、サーバーとクライアントの 2 つのタイプがあります。サーバーには Communications Server for Linux ノードおよびその関連接続コンポーネントが含まれます。クライアントには、このような接続コンポーネントは含まれませんが、ネットワークによってサーバー上の接続コンポーネントにアクセスします。サーバーは Linux コンピューターです。クライアントは AIX、Linux、または Windows で稼動することができます。(Linux コンピューターはサーバーまたはクライアントのいずれかで稼動できますが、サーバーとクライアントの両方を稼動することはできません。すなわち、同じコンピューターにサーバーとクライアントの両方をインストールすることはできません。) サ

サーバーとクライアントはバークレー・ソフトウェア・ディストリビューション (BSD) ソケットを使用し、ネットワークを介して通信します。

ドメインとして参照されるそれぞれの Communications Server for Linux ネットワークは、ドメイン・ネームによって識別されます。この名前はそれぞれの Communications Server for Linux コンピューター (サーバーまたはクライアント) のインストール時に指定されるため、単一の Communications Server for Linux ネットワーク内のすべてのコンピューターは同じドメイン・ネームを持ちます。同一の物理ネットワークに 2 つの Communications Server for Linux ドメインを別々にインストールするには、各コンピューターが属するドメインを識別する 2 つの異なるドメイン・ネームを単に使用します。単一の Communications Server for Linux ドメインは、TCP/IP サブネットに対応するか、TCP/IP サブネットの一部となるか (同じサブネットに複数の Communications Server for Linux ドメインを存在させる)、または複数のサブネットにわたることができません。

各サーバーでは、ノード構成に関する情報がノード構成ファイルに保管されます。Communications Server for Linux 管理ツールまたは NOF API を使用すると、ノードの構成を確認できます。この確認は、SNA ソフトウェアが実行されている場合は (ノードが始動されているかどうかに関わらず)、このサーバーまたはドメイン内のその他のコンピューターから行うことができます。また、このサーバーまたは他のサーバーで Communications Server for Linux 管理ツールまたは NOF API を使用して、ノードの構成を変更したり、ノードのリソースを始動および停止することができます。

完全な Communications Server for Linux ネットワークのドメイン・リソースの構成に関する情報は、ドメイン構成ファイルに保持されています。ネットワーク上に複数のサーバーがある場合、Communications Server for Linux はすべてのサーバーにわたって、この情報の整合性を確実に保ちます。

マスター・サーバーおよびバックアップ・サーバー

Communications Server for Linux とすべてのプログラムを 1 台のコンピューターか、サーバーが 1 つだけあるネットワークで使用している場合は、この節を読む必要はありません。

マスター・サーバーと呼ばれるネットワーク上の 1 つのサーバーは、常に、Communications Server for Linux ドメイン構成ファイルのマスター・コピーを保持しています。ネットワーク上の他のサーバーをバックアップ・サーバーとして定義することができます。(バックアップ・サーバーの始動、またはマスター・コピーの変更のいずれかの場合) ドメイン構成ファイルがバックアップ・サーバーにコピーされるため、すべてのバックアップ・サーバーは最新の情報のコピーを保持します。

マスター・サーバーで障害が発生した場合、またはそのコンピューター上の SNA ソフトウェアが停止した場合に、バックアップ・サーバーがマスター・サーバーの動作を引き継ぎます。このサーバー上のドメイン構成ファイルがマスター・コピーとして使用され、必要に応じて他のサーバーにコピーされます。マスター・サーバーを再始動すると、マスター・サーバーは、現在マスターとして機能しているバックアップ・サーバーからドメイン構成ファイルのコピーを受け取り、マスターとしての機能を引き継ぎます。

クライアント/サーバー操作

通常、マスター・サーバーの他に少なくとも 1 つのバックアップ・サーバーを定義します。残りのサーバーは、追加のバックアップ・マスター・サーバーとして定義するか、またはピア・サーバーとして残しておくことができます。ピア・サーバーは、必要に応じてマスター・サーバーから構成情報を取得しますが、バックアップ・サーバーとして動作することはできません。

マスター・サーバーとすべてのバックアップ・サーバーがアクティブではなくなった場合でも、ピア・サーバー上のノードは動作し続けることができるため、ノードの構成を変更することができます。ただし、ドメイン構成ファイルにアクセスすることはできないため、ノード・リソースの場合とは異なり、ドメイン・リソースの構成にアクセスすることはできません。つまり、構成ファイルで定義されているシンボリック宛先名を使用して CPI-C 会話を割り振ることができなくなります。

Communications Server for Linux がネットワーク全体のドメイン・リソースの構成の整合性を保持することができない状況があります。このような場合、構成を保持するのはユーザーの責任です。この状況は、各ドメインに 1 つ以上のバックアップ・サーバーが存在する場合に、該当するネットワークがネットワーク障害によって 2 つのドメインに分断され、相互のドメイン間通信を行うことができないときに発生します。この場合、各ドメインに代行マスター・サーバーがありますが、これらのサーバーは、おのおののドメインのドメイン構成ファイルに対して行われた変更は保持していますが、他のドメインで行われた変更については認識していません。ネットワーク接続が再度確立されると、元のマスター・サーバー（または、マスターがこの時点で活動状態にない場合は、2 つのうちいずれかのドメインで使用可能な最上位バックアップ・サーバー）のドメイン構成ファイルがネットワーク上でのドメイン構成ファイルとなります。この構成ファイルは、ネットワークが分断されていた間に他のドメインでドメイン構成ファイルに対して行われた変更を上書きします。このため、ネットワーク接続が分断されている間に 2 つのドメインのどちらにおいてもドメイン構成ファイルを変更しないようにしてください。変更は、個々のノードの構成に対して行うことができます。

Communications Server for Linux は、マスター・サーバーとバックアップ・サーバーに関する情報を SNA ネットワーク・データ・ファイルと呼ばれる **sna.net** ファイルに保管します。このファイルのマスター・コピーは、マスター・サーバーに保管されます。このファイルに対して行われた変更は、ドメイン構成ファイルに対して行われた変更がバックアップ・サーバーにコピーされるのと同様に、自動的に他のすべてのサーバーにコピーされます。このファイルの内容を直接編集することはできません。その代わりに、Communications Server for Linux は、ファイルにアクセスするための NOF verb を提供しています。

SNA ネットワーク・データ・ファイルについて詳しくは、「*Communications Server for Linux* 管理コマンド解説書」を参照してください。

AIX または Linux クライアント

クライアント・コンピューターは、構成ファイルまたは SNA ネットワーク・データ・ファイルを保持していません。クライアント・コンピューターは、Communications Server for Linux ネットワーク上のサーバーにアクセスするために必要な情報だけ保持しており、必要な構成情報の提供についてはサーバーに依存しています。

必要な SNA ネットワーク情報はファイル `/etc/sna/sna_clnt.net` (AIX) または `/etc/opt/ibm/sna/sna_clnt.net` (Linux) に保持されています。このファイルについて詳しくは、「*Communications Server for Linux 管理コマンド解説書*」を参照してください。

クライアントで、NOF API を使用して、構成の照会、セッションの初期化または活性化、およびローカル・ロギングとトレース・オプションの管理を行うことができます。ノード上では、ノード構成の変更またはリソースの停止/始動を行うことはできません。

Windows クライアント

Communications Server for Linux Windows クライアント・ソフトウェアは、Microsoft® Windows 2000、Windows XP、Windows Server 2003、または Windows Vista を稼働するマシンにインストールできます。Windows クライアントに必要な構成情報は、Windows レジストリーを介して管理されます。

Windows レジストリー、および Windows クライアントの管理について詳しくは、「*Communications Server for Linux 管理ガイド*」を参照してください。

クライアントで、NOF API を使用して、構成の照会、セッションの初期化または活性化、およびローカル・ロギングとトレース・オプションの管理を行うことができます。ノード上では、ノード構成の変更またはリソースの停止/始動を行うことはできません。

特定の Communications Server for Linux 機能を管理する NOF verb

以下の各項では、特定の Communications Server for Linux 機能に関連する NOF verb のリストを示します。各 verb について詳しくは、49 ページの『第 3 章 NOF API verb』を参照してください。

NOF verb のターゲット (ノードまたはファイル) の管理

NOF verb は、ノード、ドメイン構成ファイル、SNA ネットワーク・データ・ファイルのいずれかに対して発行できます。ターゲットのノードまたはファイルにアクセスするには、次のいずれかの verb を使用します。

- OPEN_FILE
- CONNECT_NODE

ターゲットにアクセスするために上記の verb を発行すると、初期設定では構成を照会する verb の発行のみに制限され、構成を変更する verb は発行できません。サーバー上 (クライアント上ではない) で NOF アプリケーションを実行している場合は、構成を変更する verb を発行できるように、ターゲットのノードまたはファイルへの書き込みアクセスを取得することができます。次の verb を使用します。

- SET_PROCESSING_MODE

ターゲットの構成が変更されたときにはその旨の通知が出されるように登録するには、次の verb を使用します。

- REGISTER_INDICATION_SINK

特定の Communications Server for Linux 機能を管理する NOF verb

通知が不要になったときに登録を解除するには、次の verb を使用します。

- UNREGISTER_INDICATION_SINK

NOF verb の発行を終了したときに、ターゲットのノードまたはファイルを解放するには、次のいずれかの verb を使用します。

- DISCONNECT_NODE、CLOSE_FILE

クライアント上で実行中のアプリケーションおよびサーバー上で実行中のアプリケーションから、OPEN_FILE、CONNECT_NODE、DISCONNECT_NODE、および CLOSE_FILE verb、さらに NOF QUERY verb を発行することができます。クライアントからは、その他の NOF verb のどれも発行することができません。

始めに

最初の手順では、各コンピューター上で稼働する Linux ノードと、そのノードから他のコンピューターへの通信リンクを定義します。それらのコンポーネントを定義するには、次の verb を使用します。

- DEFINE_NODE
- DEFINE_DLC、DEFINE_PORT、DEFINE_LS

これらのコンポーネントを定義したあと、リモート・システムへのリンクを設定するため、それらのコンポーネントを活動化します。(前記の DEFINE_* verb を使用して、DLC、ポート、および LS を「初期設定で活動状態」に定義すると、ノードの始動時に自動的に始動できます。その場合、それらのコンポーネントを手動で始動する必要はありません。) コンポーネントを活動化するには、次の verb を使用します。

- INIT_NODE
- START_DLC、START_PORT、START_LS

これらのコンポーネントは、それぞれ 1 つ前のコンポーネントに依存しているため、ここで示した順序で始動する必要があります。

リモート・システムへのアクセスが不要になったときに、これらのコンポーネントを停止するには、次の verb を使用します。

- STOP_LS、STOP_PORT、STOP_DLC

これらのコンポーネントの構成または現在の状況に関する情報を入手するには、次の verb を使用します。

- QUERY_NODE
- QUERY_DLC、QUERY_PORT、QUERY_LS

LS またはポートの使用状況に関する情報を入手するには、次の verb を使用します。

- QUERY_STATISTICS

接続コンポーネントが不要になったときに、それらのコンポーネントを削除するには、次の verb を使用します。

- DELETE_DLC、DELETE_PORT、DELETE_LS

特定の Communications Server for Linux 機能を管理する NOF verb

同じ共用アクセス・トランスポート機能 (SATF) 上にある多数のノードと通信している場合、それぞれのノードに対して LS を明示的に定義する必要なく、それらのノードを代表する 1 つの接続ネットワーク (CN) を設定できます。ローカル・ノードが LEN ノードである場合は、CN を使用できません。

CN を設定するには、まず SATF 上の各ノードにアクセスするために DLC およびポートを定義します。

そのあと、それらのポートをすべて含んだ CN を定義します。LS を定義する必要はありません。必要に応じて CN への動的 LS が設定されるためです。CN を定義するか、既存の CN にポートを追加するには、次の verb を使用します。

- DEFINE_CN

定義した CN について、またはある CN 上のポートに関する情報を入手するには、次の verb を使用します。

- QUERY_CN、QUERY_CN_PORT

不要になった CN を削除するか、CN を削除しないでその CN からポートを除去するには、次の verb を使用します。

- DELETE_CN

ノードを停止することによって、そのノードに関連したすべてのリソースを非活動化するには、次の verb を使用します。

- TERM_NODE

ノードが使用するデフォルト・パラメーターを定義するか、それらのパラメーターの定義を照会するには、次の verb を使用します。

- DEFINE_DEFAULTS、QUERY_DEFAULTS

そのノードについての Communications Server for Linux ライセンスで許可されているオプションと制限を照会するには、次の verb を使用します。

- QUERY_NODE_LIMITS

3270 通信

Communications Server for Linux ユーザーが 3270 エミュレーションを使用してホスト・システムと通信する場合は、ホストへの通信リンクを定義する必要があります。詳しくは、8 ページの『始めに』を参照してください。ホストへの LS の定義には、3270 エミュレーションに必要な LU を所有するためにローカル PU の名前を入れ、なおかつ *solicit_sscp_sessions* パラメーターを AP_YES に設定する必要があります。

次に、3270 エミュレーションに使用できる LU を定義する必要があります。そのためには、次の verb を使用します。

- DEFINE_LU_0_TO_3、DEFINE_LU_0_TO_3_RANGE

LU の構成または現在の状況に関する情報を入手するには、次の verb を使用します。

- QUERY_LU_0_TO_3

特定の Communications Server for Linux 機能を管理する NOF verb

LU を所有している PU に関する情報を入手するには、次の verb を使用します。

- QUERY_PU

不要になった LU を削除するには、次の verb を使用します。

- DELETE_LU_0_TO_3、DELETE_LU_0_TO_3_RANGE

LU プール (各ユーザー・セッションに LU を明示的に定義するのではなく、必要に応じてユーザー・セッションへ割り当てることができる LU のグループ) を提供したい場合は、次の verb を使用してプールを定義するか、定義に関する情報を入手するか、不要になったプールを削除したり、不要になった LU をプールから除去したりします。

- DEFINE_LU_POOL、QUERY_LU_POOL、DELETE_LU_POOL

LUA 通信

Communications Server for Linux 上で実行されるアプリケーションが LUA を使用してホスト・プログラムと通信する場合は、ホストへの通信リンクを定義する必要があります。詳しくは、8 ページの『はじめに』を参照してください。ホストへの LS の定義には、LU を所有するためにローカル PU の名前を入れ、なおかつ *solicit_sscp_sessions* パラメーターを AP_YES に設定する必要があります。

次に、LUA に使用できる LU を定義する必要があります。LU を定義するには、次の verb を使用します。

- DEFINE_LU_0_TO_3 (LU を 1 つずつ定義する)、または
DEFINE_LU_0_TO_3_RANGE (1 つの verb で複数の LU を定義する)

不要になった LU を削除するには、次の verb を使用します。

- DELETE_LU_0_TO_3 (LU を 1 つずつ削除する)、または
DELETE_LU_0_TO_3_RANGE (1 つの verb で複数の LU を削除する)

LU の構成または現在の状況に関する情報を入手するには、次の verb を使用します。

- QUERY_LU_0_TO_3

LU を所有している PU に関する情報を入手するには、次の verb を使用します。

- QUERY_PU

LU プール (各アプリケーションに LU を明示的に定義するのではなく、必要に応じてアプリケーションへ割り当てることができる LU のグループ) を提供したい場合は、次の verb を使用してプールを定義するか、定義に関する情報を入手するか、不要になったプールを削除したり、不要になった LU をプールから除去したりします。

- DEFINE_LU_POOL、QUERY_LU_POOL、DELETE_LU_POOL

Communications Server for Linux 上で実行されるアプリケーションが LUA を使用してホスト・プログラムと通信する場合は、ダウンストリーム・コンピューター上で LU を定義して、これらを Communications Server for Linux ノード上の LU にマップする必要があります。ダウンストリーム LU を定義するには、次の verb を使用します。

特定の Communications Server for Linux 機能を管理する NOF verb

- DEFINE_DOWNSTREAM_LU、 DEFINE_DOWNSTREAM_LU_RANGE、
DEFINE_DSPU_TEMPLATE

ダウンストリーム LU の構成または現在の状況に関する情報、またはそれらの LU にサービスするダウンストリーム PU に関する情報を入手するには、次の verb を使用します。

- QUERY_DOWNSTREAM_LU、 QUERY_DOWNSTREAM_PU 、
QUERY_DSPU_TEMPLATE

不要になったダウンストリーム LU を削除するには、次の verb を使用します。

- DELETE_DOWNSTREAM_LU、 DELETE_DOWNSTREAM_LU_RANGE、
DELETE_DSPU_TEMPLATE

APPC 通信

Communications Server for Linux 上で実行中のアプリケーションが APPC を使用して、ホスト・コンピューターまたは対等コンピューター上で実行中のアプリケーションと通信する場合は、APPC で使用できる LU を定義する必要があります。

APPN ネットワークでの APPC の構成は、APPN 以前の SNA ネットワークでの構成よりもかなり簡単です。必要なコンポーネントと、それらのコンポーネント間の対話は、セッションの開始時に動的に定義したり決定したりすることができるため、初期設定で明示的に指定する必要はありません。

各ノードには、デフォルトの APPC ローカル LU (制御点 LU) が 1 つずつ含まれています。APPC アプリケーションはこの LU を使用できますが、異なるアプリケーションが別々の LU を使用できるように追加 LU を定義することもできます。LU を定義するには、次の verb を使用します。

- DEFINE_LOCAL_LU

制御点 LU も含め、LU の構成または現在の状況に関する情報を入手するには、次の verb を使用します。

- QUERY_LOCAL_LU

APPN は、ローカル・アプリケーションがパートナー LU へのセッションを開始する必要があるときに、そのパートナー LU の位置を動的に見つけることができるため、通常、パートナー LU を定義する必要はありません。ただし、会話セキュリティーなどの特定の APPC 機能を強制的に使用させる必要がある場合は、パートナー LU を定義しなければならない場合もあります。パートナー LU を定義するには、次の verb を使用します。

- DEFINE_PARTNER_LU

パートナー LU が明示的に定義された場合、そのパートナー LU の現在の状況またはその定義に関する情報を入手するには、次の verb を使用します。

- QUERY_PARTNER_LU、 QUERY_PARTNER_LU_DEFINITION

ローカル・アプリケーションが、標準 SNA 定義モードの 1 つを使用してパートナーと通信する場合、モードを定義する必要はありません。しかし、標準モードでサポートされない特定の要件があるアプリケーションに追加モードを定義することもできます。モードを定義するには、次の verb を使用します。

特定の Communications Server for Linux 機能を管理する NOF verb

- DEFINE_MODE

認識されないモード名に使用するパラメーターを指定するデフォルト・モードを定義または照会するには、次の verb を使用します。

- DEFINE_DEFAULTS、QUERY_DEFAULTS

モードに使用するサービス・クラス (COS) は通常、標準 SNA 定義サービス・クラスの 1 つです。ただし、特定の COS へ 1 つのモードをマップできるようにノードを構成することもできます (DEFINE_NODE verb の *mode_to_cos_map_supp* パラメーター)。その場合、標準 COS でサポートされない特定の要件があるアプリケーションに COS の追加を定義することもできます。COS を定義するには、次の verb を使用します。

- DEFINE_COS

認識されないモードのマップ先にするデフォルトの COS を指定するには、次の verb を使用します。

- DEFINE_MODE

モードの定義または現在の使用状況に関する情報、あるモードが使用する COS に関する情報、COS の定義に関する情報を入手するには、次の verb を使用します。

- QUERY_MODE_DEFINITION、QUERY_MODE、
QUERY_MODE_TO_COS_MAPPING
- QUERY_COS、QUERY_COS_NODE_ROW、QUERY_COS_TG_ROW

ローカル LU およびパートナー LU がセッション・レベル・セキュリティーを使用する場合は、ローカル LU とパートナー LU の間にセッションを確立するため、パスワードを定義する必要があります。パスワードを定義するか、現行の定義を検査するか、不要になったパスワードを削除するには、次の verb を使用します。

- DEFINE_LU_LU_PASSWORD、QUERY_LU_LU_PASSWORD、
DELETE_LU_LU_PASSWORD

不要のなくなったローカル LU、パートナー LU、モード、COS を削除するには、次の verb を使用します。

- DELETE_LOCAL_LU、DELETE_PARTNER_LU
- DELETE_MODE、DELETE_COS

Communications Server for Linux はセッションを確立するときに、自動的にパートナー LU とセッション限度を折衝します。ローカル LU とそのパートナー LU の間のセッション限度を明示的に管理する必要がある場合は、次の verb を使用します。

- INITIALIZE_SESSION_LIMIT、CHANGE_SESSION_LIMIT、
RESET_SESSION_LIMIT

個々のセッションおよび会話を管理するには、次の verb を使用します。

- QUERY_SESSION、QUERY_ISR_SESSION、QUERY_CONVERSATION
- ACTIVATE_SESSION、DEACTIVATE_SESSION、DEACTIVATE_CONV_GROUP

特定の Communications Server for Linux 機能を管理する NOF verb

Communications Server for Linux の呼び出し可能 TP がオペレーターによって起動される場合、通常は、それらの TP を定義する必要はありません。リモート TP から TP に会話が割り振られたときに、Communications Server for Linux によって TP が自動的に起動される場合、TP がオペレーターによって起動され、ブロードキャスト待機 TP となる必要がある場合 (つまり、着呼会話要求を実行中の任意の TP に動的に経路指定できる場合)、または TP がオペレーターによって起動され、特定の Receive_Allocate タイムアウト値を必要とする場合には、その TP を Communications Server for Linux の呼び出し可能 TP データ・ファイル内に指定する必要があります。このファイルについて詳しくは、「*Communications Server for Linux* 管理コマンド解説」を参照してください。

さらに、TP (オペレーターによって起動されるか自動的に起動される) を特定の会話セキュリティー、確認同期、会話タイプ (マップ式または基本) の値のみに制限する必要がある場合、または一度に実行できるその TP のインスタンス数を制限する必要がある場合は、その TP を定義する必要があります。次の verb を使用します。

- DEFINE_TP

ある TP の定義に関する情報、その TP の現在の使用状況に関する情報、あるいは現在活動状態である呼び出し可能 TP に関する情報を入手するには、次の verb を使用します。

- QUERY_TP_DEFINITION、QUERY_TP、QUERY_AVAILABLE_TP

不要になった定義済み TP を削除するには、次の verb を使用します。

- DELETE_TP

呼び出し可能 TP に会話レベル・セキュリティーが必要な場合は、リモート TP が Communications Server for Linux TP へアクセスするために使用できるユーザー ID とパスワードを定義する必要があります。ユーザー ID とパスワードを定義するか、現在の定義を検査するか、不要になったユーザー ID とパスワードを削除するには、次の verb を使用します。

- DEFINE_USERID_PASSWORD、QUERY_USERID_PASSWORD、DELETE_USERID_PASSWORD

TP の使用を特定のリストにある許可ユーザー ID に制限するか、許可ユーザー ID の現行リストをチェックするか、または、不要になったユーザー ID のリストを削除するには、次の verb を使用します。

- DEFINE_SECURITY_ACCESS_LIST、QUERY_SECURITY_ACCESS_LIST、DELETE_SECURITY_ACCESS_LIST

CPI-C 通信

CPI-C アプリケーションは、APPC アプリケーションと同じリソースを使用します。11 ページの『APPC 通信』の説明は、APPC のみでなく CPI-C にも適用されます。

さらに、CPI-C アプリケーションで使用するサイド情報エントリーをセットアップできます。それぞれのエントリーは、特定のパートナー・アプリケーションとそれにアクセスするために必要な情報を定義します。そのあと、ローカル CPI-C アプリ

ケーションは、パートナー・アプリケーションをサイド情報エントリーの名前で容易に識別でき、パートナー LU 名と TP 名、モード名、会話セキュリティー要件を明示的に指定せずに済みます。サイド情報エントリーを定義するか、現在の定義を検査するか、不要になったエントリーを削除するには、次の verb を使用します。

- DEFINE_CPIC_SIDE_INFO、QUERY_CPIC_SIDE_INFO、DELETE_CPIC_SIDE_INFO

HPR RTP 接続の管理

RTP 接続をセットアップするときに使用されるパラメーターを調整するには、次の verb を使用します。

- DEFINE_RTP_TUNING

RTP 接続をセットアップするときに使用するための現在定義済みの調整パラメーターを確認する場合、または現在アクティブな RTP 接続の詳細を確認する場合は、次の verb を使用します。

- QUERY_RTP_TUNING、QUERY_RTP_CONNECTION

SNA ゲートウェイの管理

ノードが SNA ゲートウェイ (DEFINE_NODE verb の *pu_conc_support* パラメーター) をサポートする場合、ダウンストリーム・コンピューター上のタイプ 0-3 の LU が、Communications Server for Linux ノード上に定義された LU を使用してホスト・システムと通信できるように、まず、以下のものを定義しなければなりません。

- Communications Server for Linux からダウンストリーム・コンピューターへの DLC、ポート、および LS。これらのコンポーネントの定義方法については、8 ページの『はじめに』を参照してください。LS は、次のパラメーターを使用して定義しなければなりません。

```
solicit_sscp_sessions = NO  
dspu_services         = PU_CONCENTRATION
```

```
dspu_name = ダウンストリーム・コンピューター上で LU にサービス提供する PU の名前  
pu_name   = すべてゼロ
```

- ホストと通信するための Communications Server for Linux ノード上にあるタイプ 0-3 の 1 つ以上の LU (およびオプションでこれらの LU が入っている LU プール)。LU および LU プールの定義方法については、9 ページの『3270 通信』を参照してください。

次に、ダウンストリーム・コンピューター上に LU を定義し、それらの LU を Communications Server for Linux ノード上の LU へマップします。ダウンストリーム LU を定義するには、次の verb を使用します。

- DEFINE_DOWNSTREAM_LU、DEFINE_DOWNSTREAM_LU_RANGE

ダウンストリーム LU の構成または現在の状況に関する情報、またはそれらの LU にサービスするダウンストリーム PU に関する情報を入手するには、次の verb を使用します。

- QUERY_DOWNSTREAM_LU、QUERY_DOWNSTREAM_PU

不要になったダウンストリーム LU を削除するには、次の verb を使用します。

- DELETE_DOWNSTREAM_LU、DELETE_DOWNSTREAM_LU_RANGE

DLUR の管理

ノードが DLUR (DEFINE_NODE verb の *dlur_support* パラメーター) をサポートしていて、Communications Server for Linux ノード上の LU が DLUR を使用してホスト・システムと通信する場合は、それらの LU を所有するローカル Communications Server for Linux ノード上に PU を定義する必要があります。これは、ホストと直接通信する LU に PU を定義する (DEFINE_LS verb を使用して行う) のとは違います。

PU を定義するには、次の verb を使用します。

- DEFINE_INTERNAL_PU

PU に関する情報を入手するには、次の verb を使用します。

- QUERY_PU

この PU に関連した LU を定義して管理するには、前記の 9 ページの『3270 通信』または 10 ページの『LUA 通信』を参照してください。

LU を使用するために PU を始動する (ホストから ACTPU を要求する) か、アプリケーションがそれらの LU を使用しなくなったときに、その PU を停止するには、次の verb を使用します。

- START_INTERNAL_PU、STOP_INTERNAL_PU

不要になった PU を削除するには、次の verb を使用します。

- DELETE_INTERNAL_PU

ローカル・ノードがネットワーク・ノードであり、ダウンストリーム PU 上の LU が DLUR を使用してホスト・システムと通信する場合は、8 ページの『はじめに』で説明しているように、ダウンストリーム PU への通信リンクを定義する必要があります。LS 定義では、ローカル・ノードがダウンストリーム PU への DLUR サービスを提供するように指定しなければなりません。

ダウンストリーム PU を定義する必要はありません。Communications Server for Linux は通信リンクが確立されたときに、必要な情報を動的に入手します。現在 DLUR を使用している LU とダウンストリーム PU に関する情報を入手するには、次の verb を使用します。

- QUERY_DOWNSTREAM_PU、QUERY_DOWNSTREAM_LU

DLUR の構成を簡単にして、他の DLUR verb で必要な情報を減らすためにデフォルト値をセットアップするには、次の verb を使用します。

- DEFINE_DLUR_DEFAULTS

現在 DLUR を (ローカル・ノード上かダウンストリーム PU 上で) 使用している LU と PU に関する情報、または、それらが使用している DLUR ノードに関する情報を入手するには、次の verb を使用します。

- QUERY_DLUR_PU、QUERY_DLUR_LU、QUERY_DLUR

TN サーバーの管理

TN3270 ユーザーが Communications Server for Linux ノード上の TN サーバー機能を使用してホスト・システムと通信する場合は、ホストへの通信リンクを定義する必要があります。詳しくは、8 ページの『始めに』を参照してください。ホストへの LS の定義には、3270 LU を所有するためにローカル PU の名前を入れ、なおかつ *solicit_sscp_sessions* パラメーターを AP_YES に設定する必要があります。

次に、3270 エミュレーションに使用できる LU を定義して、オプションでそれらの LU を LU プールにグループ化します。LU およびプールの定義方法については、9 ページの『3270 通信』を参照してください。

すべての TN サーバー・ユーザーに適用するパラメーターを定義するには、次の verb を使用します。

- DEFINE_TN3270_DEFAULTS

セキュア・ソケット層 (SSL) クライアント認証を使用していて、外部 LDAP サーバー上の認証取り消しリストでクライアントを検査する場合、このサーバーへのアクセス方法の詳細を構成する必要があります。また、クライアント・ユーザーが TN3270 高速ログオン機能を使用することができ、これらのユーザーのセキュリティ証明書許可によって、TN3270 セキュリティーに通常使用される標準のユーザー ID とパスワードが置き換えられる場合は、この機能の管理に使用されるホストの Digital Certificate Access Server (DCAS) を構成する必要があります。次の verb を使用します。

- DEFINE_TN3270_SSL_LDAP
- DEFINE_TN3270_EXPRESS_LOGON

TN サーバーにアクセスできる TN3270 ユーザーを定義し、それらのユーザーを Communications Server for Linux 3270 LU に割り当てるには、次の verb を使用します。

- DEFINE_TN3270_ACCESS

TN3270 ディスプレイ LU とプリンター LU の関連を定義して、TN3270E クライアントがプリンター LU の名前を意識することなく、ディスプレイ LU に関連したプリンター LU に接続できるようにするには、次の verb を使用します。

- DEFINE_TN3270_ASSOCIATION

TN サーバーおよび TN3270 ユーザーの構成に関する情報を入手するには、次の verb を使用します。

- QUERY_TN3270_ACCESS_DEF、QUERY_TN3270_ASSOCIATION、
QUERY_TN3270_DEFAULTS、QUERY_TN3270_SSL_LDAP、
QUERY_TN3270_EXPRESS_LOGON

TN3270 ユーザーを削除し、それらのユーザーが TN サーバーを 3270 エミュレーションで使用できないようにするか、LU 関連情報を削除するには、次の verb を使用します。

- DELETE_TN3270_ACCESS、DELETE_TN3270_ASSOCIATION

TN リダイレクターの管理

Telnet ユーザーが **Communications Server for Linux** ノード上の TN リダイレクター機能を使用してホスト・システムと通信する場合は、これらのユーザーとそのホストへのアクセス方法を定義する必要があります。

TN リダイレクターにアクセスできる TN3270 ユーザーを定義するには、次の verb を使用します。

- **DEFINE_TN_REDIRECT**

セキュア・ソケット層 (SSL) クライアント認証を使用していて、外部 LDAP サーバー上の認証取り消しリストでクライアントを検査する場合、このサーバーへのアクセス方法の詳細を構成する必要があります。また、クライアント・ユーザーが TN3270 高速ログオン機能を使用することができ、これらのユーザーのセキュリティー証明書許可によって、TN3270 セキュリティーに通常使用される標準のユーザー ID とパスワードが置き換えられる場合は、この機能の管理に使用されるホストの Digital Certificate Access Server (DCAS) を構成する必要があります。次の verb を使用します。

- **DEFINE_TN3270_SSL_LDAP**

TN リダイレクターおよび TN リダイレクター・ユーザーの構成に関する情報を入手するには、次の verb を使用します。

- **QUERY_TN_REDIRECT_DEF**、**QUERY_TN3270_SSL_LDAP**

TN リダイレクター・ユーザーを削除し、それらのユーザーが TN リダイレクターを使用してホストにアクセスできないようにするには、次の verb を使用します。

- **DELETE_TN_REDIRECT**

SNA Management Services 機能の管理

Communications Server for Linux 上で実行中のアプリケーションが MS API を使用してリモート MS アプリケーションと通信する場合は、ノードが必要に応じて適切なリモート・アプリケーションを見つけるため、そのリソースを明示的に定義する必要はありません。ただし、使用する特定のリモート・アプリケーションを指定したい場合には、そのリソースを明示的に定義できます。

NMVT レベルのアプリケーションで使用するデフォルト PU を指定する (NMVT レベルのアプリケーションが特定のホストの NetView® プログラムにアクセスできるようにする) には、次の verb を使用します。

- **DEFINE_DEFAULT_PU**

MDS レベルのアプリケーションで使用するフォーカル・ポイント・アプリケーションを指定する (リモート・フォーカル・ポイント・アプリケーションに、そのアプリケーションが管理するノードを判別できるようにするのではなく) には、次の verb を使用します。

- **DEFINE_FOCAL_POINT**

現在使用中のフォーカル・ポイントに関する情報を入手するか、以前定義したフォーカル・ポイントを削除するには、次の verb を使用します。

- **QUERY_FOCAL_POINT**、**DELETE_FOCAL_POINT**

特定の Communications Server for Linux 機能を管理する NOF verb

MS 機能を使用して活動アプリケーション (NMVT レベルまたは MDS レベル) に関する情報を入手するには、次の verb を使用します。

- QUERY_NMVT_APPLICATION、QUERY_MDS_APPLICATION

MDS レベル・アプリケーションからの未処理の要求に関する情報を入手するか、直前の要求に関する統計情報を入手するには、次の verb を使用します。

- QUERY_ACTIVE_TRANSACTION、QUERY_MDS_STATISTICS

ホスト NetView プログラムから Communications Server for Linux システムへのアクセス管理

ホストの NetView コンソールでオペレーターがサービス・ポイント・コマンド機能 (SPCF) または Communications Server for Linux コマンド機能 (UCF) のいずれかを使用して、UNIX[®] コンピューターに対してコマンドを発行できるようにしたい場合は、オペレーターのアクセス許可を定義する必要があります。

このアクセス許可を定義し、NetView オペレーターが SPCF か UCF のどちらか一方またはその両方にアクセスできるようにするには、次の verb を使用します。

- DEFINE_RCF_ACCESS

現在定義されている許可を検査するには、次の verb を使用します。

- QUERY_RCF_ACCESS

オペレーターが SPCF または UCF を使用できないようにするには、次の verb を使用します。

- DELETE_RCF_ACCESS

一方の機能へのアクセスを除去し、他方の機能へのアクセスを残すには、次の verb を使用します。

- DEFINE_RCF_ACCESS

診断設定の管理

ログ・メッセージについての Communications Server for Linux のデフォルト設定では、問題メッセージと例外メッセージはログに記録されますが、監査メッセージは記録されず、中央ロギング (マスター・サーバー上の中央ログ・ファイルに送信された、すべてのサーバーからのメッセージ) が使用されます。succinct ログが使用されます (つまり、ヘッダー・パラメーターとメッセージ・テキストがログに記録されますが、各メッセージの原因と処置の完全な詳細は記録されません)。問題メッセージと例外メッセージに使用されるエラー・ログ・ファイルは

/var/opt/ibm/sna/sna.err で、監査メッセージに使用される監査ログ・ファイル (使用可能な場合は **/var/opt/ibm/sna/sna.aud** です。これらのファイルはそれぞれのファイル・サイズが 1 メガバイトになった時点でバックアップが取られ、リセットされます。succinct ログ、例外および監査のログ、ファイル名、ファイル・サイズのデフォルトの設定は、次のようにすべて NOF verb を使用して上書きできます。

中央ロギング・オプションとグローバル・ロギング・オプションを管理する verb は、クライアントおよびサーバーに適用されます。ただし、Windows クライアント

特定の Communications Server for Linux 機能を管理する NOF verb

での他の診断設定は、Windows レジストリーのオプションによって制御され、NOF verb では制御されません。詳しくは、「*Communications Server for Linux 管理コマンド解説*」を参照してください。

Communications Server for Linux は使用量のログ・ファイル `/var/opt/ibm/sna/sna.usage` も保守します。このファイルは Communications Server for Linux リソースの現在の使用量およびピーク時の使用量に関する情報を記録するために使用されます。このファイルはエラー・ログ・ファイルおよび監査ログ・ファイルと同様に、バックアップが取られてリセットされます。ファイル名およびファイル・サイズも同様に指定できます。

中央ロギングを使用可能にするかどうかを指定するには、次の verb を使用します。

- SET_CENTRAL_LOGGING

例外メッセージと監査メッセージのいずれか一方または両方をログに記録するように指定したり、succinct ログとフル・ログのいずれを使用するかを指定したり、すべてのサーバーのグローバル・デフォルト設定を確立するか、特定のサーバーのデフォルトを上書きするかを指定したりするには、次の verb を使用します。

- SET_GLOBAL_LOG_TYPE、SET_LOG_TYPE

ログ・メッセージに使用するファイル名またはディレクトリを変更するか、ファイルをバックアップしてリセットするサイズを変更するには、次の verb を使用します。

- SET_LOG_FILE

中央ロガーとして現在定義されているサーバーを検査したり、中央ロギングが使用可能になっているかどうかを検査するには、次の verb を使用します。

- QUERY_CENTRAL_LOGGER、QUERY_CENTRAL_LOGGING

記録しているメッセージのタイプを検査したり、succinct ロギングとフル・ロギングのいずれが使用されているかを検査したり、グローバルに使用されているか特定のサーバーで使用されているかを検査するには、次の verb を使用します。

- QUERY_GLOBAL_LOG_TYPE、QUERY_LOG_TYPE

特定のログ・タイプに使用されているファイル、ファイル・サイズ、またはディレクトリを検査するには、次の verb を使用します。

- QUERY_LOG_FILE

特定の Communications Server for Linux ノード上の接続コンポーネントに関する問題を診断するためにトレースを活動化したい場合、または必要なデータを収集したあとにトレースを非活動化したい場合は、次の verb を使用します。

- ADD_DLC_TRACE、REMOVE_DLC_TRACE

その他の Communications Server for Linux カーネル・コンポーネントに関する問題を診断するためにトレースを活動化したい場合、または必要なデータを収集したあとにトレースを非活動化したい場合は、次の verb を使用します。

- SET_TRACE_TYPE

特定の Communications Server for Linux 機能を管理する NOF verb

Communications Server for Linux LAN 全体のクライアントとサーバー間の通信における問題を診断するためにトレースを活動化したり、必要なデータを収集した後にトレースを非活動状態にするには、次の verb を使用します。

- SET_CS_TRACE

Communications Server for Linux TN サーバー機能に関する問題を診断するためにトレースを活動化したり、必要なデータを収集したあとにトレースを非活動状態にしたりするには、次の verb を使用します。

- SET_TN_SERVER_TRACE

トレース・データに使用されるデフォルト・ファイルは次のとおりです。

- /var/opt/ibm/sna/sna1.trc および /var/opt/ibm/sna/sna2.trc (特定のコンピューターに対するトレース用)
- /var/opt/ibm/sna/snacs1.trc および /var/opt/ibm/sna/snacs2.trc (LAN トレース用)
- /var/opt/ibm/sna/snatsv1.trc および /var/opt/ibm/sna/snatsv2.trc (TN サーバー・トレース用)

これらのトレース・タイプのいずれかに別のファイルまたはディレクトリーを使用したい場合、または特定のタイプのすべてのトレースを 2 つのファイルではなく 1 つのファイルへ送信したい場合は、次の verb を使用します。

- SET_TRACE_FILE

特定のトレース・タイプに関する現在の設定を検査するか、特定のトレース・タイプに使用されているファイルを検査するには、次の verb を使用します。

- QUERY_DLC_TRACE、QUERY_TRACE_TYPE、QUERY_CS_TRACE、
QUERY_TN_SERVER_TRACE、QUERY_TRACE_FILE

ディレクトリー・エントリーの管理

ローカル・ノードが LEN ノードである場合、Communications Server for Linux の通信先となる隣接ノードと、それらのノードに関連した LU を識別するため、ローカル・ノードのディレクトリー内にエントリーをセットアップする必要があります。特定のノード内に複数の名前が類似した LU が含まれている場合は、ディレクトリー内にワイルドカード・エントリーをセットアップして、その範囲内にあるすべての LU が指定したノード上にあることを示すことができます。

あるノードと、そのノードに関連した LU を定義するには、次の verb を使用します。

- DEFINE_ADJACENT_LEN_NODE

データベース内にある特定のノードまたは LU エントリーに関する情報を入手するには、次の verb を使用します (ただし、この verb はワイルドカード・エントリーに関する情報を戻すためには使用できません)。

- QUERY_DIRECTORY_ENTRY

データベース内にある特定の LU エントリーまたはワイルドカード・エントリーに関する情報を入手するには、次の verb を使用します。

- QUERY_DIRECTORY_LU

特定の Communications Server for Linux 機能を管理する NOF verb

ディレクトリー・エントリーに関する統計情報を入手するには、次の verb を使用します。

- QUERY_DIRECTORY_STATS

あるノードと、そのノードに関連した LU を削除するか、あるノード・エントリーから LU を削除するには、次の verb を使用します。

- DELETE_ADJACENT_LEN_NODE

ローカル・ノードが LEN ノードと通信するネットワーク・ノードかエンド・ノードである場合、または LEN ノードにサービスを提供するネットワーク・ノードである場合は、上記の各コマンドを使用して LEN ノードとその LU のディレクトリー・エントリーをセットアップする必要があります。その他のノード・タイプとの通信では、ノードが必要に応じて動的にディレクトリー・エントリーを見つけるため (そして、再び使用できるようにディレクターに追加する)、そのセットアップは必要ありません。

ただし、特定のノードまたは LU のエントリーをセットアップし、ローカル・ノードが探索しなくてもそれらのリソースと通信できるようにすることもできます。特定のノードまたは LU 用にエントリーをセットアップすると、通常の APPN のリソース探索プロセスが上書きされるため、定義が正しくない場合には、そのノードかネットワーク内の別のノードで問題が起こる可能性があります。別のノードにあるリソースについて明示的なエントリーを定義する場合には、必ず正しく定義してください。

個々のノード、LU、またはある範囲の LU のワイルドカード・エントリーを定義するには、次の verb を使用します。

- DEFINE_DIRECTORY_ENTRY

個々のノード、LU、またはワイルドカード・エントリーをディレクトリーから削除するには、次の verb を使用します。

- DELETE_DIRECTORY_ENTRY

上記の各 verb を使用して明示的に定義したディレクトリー・エントリー (これらのエントリーは QUERY_DIRECTORY_ENTRY verb では HOME というエントリー・タイプを戻します) のみを削除してください。キャッシュに書き込まれたエントリー (ネットワーク検索の結果として動的にセットアップされたエントリー) を削除するために、この verb を使用しないでください。

ネットワーク・トポロジーの照会

隣接ネットワーク・ノードに関する (あるネットワーク・ノード上) 情報を入手するには、次の verb を使用します。

- QUERY_ADJACENT_NN

隣接ネットワーク・ノードへの TG に関する情報を入手するには、次の verb を使用します。

- QUERY_LOCAL_TOPOLOGY

特定の Communications Server for Linux 機能を管理する NOF verb

ネットワーク・ノードとネットワーク内の仮想経路指定ノード (VRN) に関する (あるネットワーク・ノード上) 情報、またはそれらのノードへの TG に関する情報を入手するには、次の verb を使用します。

- QUERY_NN_TOPOLOGY_NODE、QUERY_NN_TOPOLOGY_TG

ローカル・ノードのトポロジー・データベース内にあるエントリの使用状況に関する (あるネットワーク・ノード上) 統計情報を入手するには、次の verb を使用します。

- QUERY_NN_TOPOLOGY_STATS

リモート LU への通信パスの検査

特定のターゲット LU にアクセスできるかどうか (その LU を所有しているノードが活動状態で、その LU への通信パスが存在するかどうか) を検査するには、次の verb を使用します。

- APING

Communications Server for Linux LAN 上のサーバーおよびクライアントの管理

Communications Server for Linux LAN 上のサーバー (ノード) のリストを取得するには、次の verb を使用します。

- QUERY_NODE_ALL

特定のノードに関する詳細情報を取得するには、次の verb を使用します。

- QUERY_NODE

マスター構成ファイル・サーバーおよびバックアップ・マスター・サーバーとして動作するサーバーを確認するには、次の verb を使用します。

- QUERY_SNA_NET

リストに新規バックアップ・マスター・サーバーを追加したり、リストから既存のサーバーを除去してこれらのサーバーがマスター・サーバーとして動作しないようにするには、次の verb を使用します。

- ADD_BACKUP、DELETE_BACKUP

Communications Server for Linux LAN 上の特定のサーバーを使用して Remote API Client (AIX、Linux または Windows 上) のリストを取得するには、次の verb を使用します。

- QUERY_RAPI_CLIENTS

構成ファイルのヘッダー情報の管理

ドメイン構成ファイルに説明のコメント・ストリングを追加するには、次の verb を使用します。

- DEFINE_DOMAIN_CONFIG_FILE

ドメイン構成ファイルの作成対象となった Communications Server for Linux バージョン番号に関する情報、またはそのファイルに保管されているコメント・ストリングに関する情報を入手するには、次の verb を使用します。

- QUERY_DOMAIN_CONFIG_FILE

ノード構成ファイル内のヘッダー情報は Communications Server for Linux の内部でのみ使用するため、これらに対応するノード構成ファイルの verb はありません。ノード構成ファイルを変更しないでください。

Linux リソース使用状況の管理

Communications Server for Linux が内部データ構造に使用できるカーネル・メモリー容量に限度を設定するか、STREAMS バッファに使用できるメモリーの最大容量を指定するには、次の verb を使用します。

- SET_KERNEL_MEMORY_LIMIT、SET_BUFFER_AVAILABILITY

現在の限度と使用状況に関する情報を入手するには、次の verb を使用します。

- QUERY_KERNEL_MEMORY_LIMIT、QUERY_BUFFER_AVAILABILITY

NOF 指示

NOF アプリケーションでは、REGISTER_INDICATION_SINK verb を使用して、Communications Server for Linux の構成またはそのリソース状況の変更内容についての情報を要求できます。その場合、Communications Server for Linux は、変更されるたびに指示メッセージをアプリケーションへ送信します。

アプリケーションが要求できる指示について詳しくは、777 ページの『第 4 章 NOF 指示』を参照してください。

CONFIG_INDICATION、NOF_STATUS_INDICATION、および SNA_NET_INDICATION を除いて、指定したタイプのリソースの状況が変更されるたびに指示が戻されます。たとえば、DLC 指示を受け取るようにアプリケーションを登録した場合、Communications Server for Linux は DLC が活動状態または非活動状態になるたびに、アプリケーションへ DLC_INDICATION メッセージを送信します。

指示は、発生した変更に関する要約情報を戻します。必要に応じて、アプリケーションは適切な QUERY_* verb を発行して、さらに詳しい情報を入手できます。

ローカル・ノードはリソースが不足している場合、指示を一時的に抑止してアプリケーションへ送信しません。ローカル・ノードは、リソース不足の状態を解決したあと、前に抑止したタイプの指示を生成したときに、指示に関するパラメーターを設定して前に生成したそのタイプの 1 つ以上の指示が失われたことをアプリケーションに通知します。そのあとで、アプリケーションは該当するリソース・タイプについて QUERY_* verb を発行し、リソースの現在の状態を判別する必要があります。

指示を受信するための登録方法について詳しくは、719 ページの

『REGISTER_INDICATION_SINK』を参照してください。個々の指示について詳しくは、777 ページの『第 4 章 NOF 指示』を参照してください。

構成の指示

アプリケーションは、特定のターゲット（ドメイン構成ファイル、実行中のノード、非活動ノードのいずれか）の構成に加えられた変更内容に関する情報を受信する登録ができます。これにより、アプリケーションは他の NOF アプリケーションまたは管理プログラムの変更内容を追跡できます。追跡するには、アプリケーションは他の指示の場合と同様に、要求する指示タイプとして `CONFIG_INDICATION` を指定して登録します。

この指示タイプへ関連付けられている特定の VCB 構造体はありません。その代わりに、構成が変更されると、Communications Server for Linux はこの変更をアプリケーションに示すため、変更を行った NOF verb から VCB 全体のコピーを送信します。

構成ディレクティブについて詳しくは、777 ページの『`CONFIG_INDICATION`』を参照してください。

SNA ネットワーク・ファイル指示

アプリケーションは、マスター・サーバー上の SNA ネットワーク・ファイル `sna.net` に対する変更についての情報を受信するための登録を行うことができます。これにより、アプリケーションは他の NOF アプリケーションまたはコマンド行管理プログラムによるこのファイルへの変更内容を追跡できます。追跡するには、アプリケーションは他の指示の場合と同様に、要求する指示タイプとして `SNA_NET_INDICATION` を指定して登録します。

この指示タイプには、次の 2 つの VCB 構造が関連付けられます。

- `ADD_BACKUP` (バックアップ・サーバーがファイルの終わりに追加されたことを示します)
- `DELETE_BACKUP` (未使用のバックアップ・サーバーがファイルから除去されたことを示します)

`SNA_NET_INDICATION` のタイプで登録した場合、バックアップ・サーバーが追加されると `ADD_BACKUP` 指示が戻され、サーバーが削除されると `DELETE_BACKUP` 指示が戻されます。アプリケーションは、それぞれの指示ごとに個別に登録を行う必要がありません。いずれの場合も、指示のフォーマットは、変更を行った NOF verb から完了した VCB のコピーです。

SNA ネットワーク・ファイル指示について詳しくは、838 ページの『`SNA_NET_INDICATION`』を参照してください。

NOF 状況の指示

Communications Server for Linux は、アプリケーションがターゲットのノードまたはファイルへアクセスできなくなったときに（ターゲット・コンピューター上の Communications Server for Linux ソフトウェアが停止したか、そのコンピューターへの通信パスが失われたため）、登録された NOF アプリケーションへ NOF 状況の指示を送信します。アプリケーションがマスター構成ファイルから指示を受信するように登録されている場合、別のサーバーがマスターの動作を引き継いだときにもこの指示が戻されます（このため、ターゲット・ファイルは、マスター構成ファイルではなくなります）。

アプリケーションは、この指示を受信するために明示的に登録する必要はありません。Communications Server for Linux は、該当するターゲットに関する任意のタイプの NOF 指示について登録したすべてのアプリケーションへこの指示を戻します。指示は、アプリケーションが REGISTER_INDICATION_SINK verb (アプリケーションが複数回この verb を発行した場合は、最初の REGISTER_INDICATION_SINK verb) へ提供したコールバック・ルーチンへ戻されます。

ターゲットに障害が起きたという指示をアプリケーションが受信したあと、関連するターゲット・ハンドルを使用する、それ以後のすべての verb は拒否されます。しかし、ターゲット・ハンドルを解放する DISCONNECT_NODE と CLOSE_FILE は例外です。さらに、そのターゲット・ハンドル上の指示についての登録は失われます。ターゲットが使用可能になったときに、アプリケーションが指示の受信を続行するためには、再度ターゲットに接続し、必要な指示について再度登録する必要があります。

NOF 状況の指示について詳しくは、777 ページの『第 4 章 NOF 指示』を参照してください。

第 2 章 NOF アプリケーションの作成

この章では、次の項目について説明します。

- クライアント/サーバーに関する考慮事項

UNIX

- AIX または Linux に関する考慮事項
 - Linux の NOF API エントリー・ポイント
 - NOF アプリケーションのコンパイルとリンク

WINDOWS

- Windows に関する考慮事項
 - Windows の NOF API エントリー・ポイント
 - NOF アプリケーションのコンパイルとリンク

- ポータブル・アプリケーションの作成
- NOF verb のターゲット (ノードまたはファイル) と、それらの verb とターゲットとの対話方法
- NOF verb 相互の順序付けと依存関係
- ノード構成に基づいた NOF の制約事項
- QUERY_* verb を使用した 1 つ以上のデータ・エントリーを要求する方法

クライアント/サーバーに関する考慮事項

クライアント/サーバー・システムでは、サーバーで実行中のアプリケーションで、いずれの NOF verb も使用できます。Remote API Client で実行中のアプリケーションは、以下のように NOF verb を使用できます。

- アプリケーションで構成または状況情報を照会する場合は、QUERY_* verb を使用します。
- アプリケーションでセッションを活動化したり、セッション限度を初期化したりする場合、またはロギングおよびクライアント/サーバーのロギングを管理する場合には、verb を使用します。NOF アプリケーションには、これらのコマンドを使用するために、ユーザー ID root を使用するか、または sys グループ (AIX) または sna グループ (Linux) のメンバーであるユーザー ID を使用して実行しなければなりません。
- アプリケーションでは、その他の verb を使用して構成を変更したり、リソースを始動および停止することはできません。NOF アプリケーションで構成を変更

したり、リソースを始動および停止する必要がある場合は、このアプリケーションを Linux 用に作成して、サーバーで実行しなければなりません。

AIX または Linux に関する考慮事項

UNIX

この節では、AIX または Linux 環境で使用する NOF アプリケーションを作成するときに考慮する必要がある、オペレーティング・システムの問題について説明します。

AIX または Linux の NOF API エントリー・ポイント

アプリケーションは、次のエントリー・ポイント関数コールを使用して NOF API にアクセスします。

nof NOF verb を同期形式で発行します。 Communications Server for Linux は、 verb の処理が終了するまでアプリケーションに制御を戻しません。 REGISTER_INDICATION_SINK と UNREGISTER_INDICATION_SINK を除いて、すべての NOF verb はこのエントリー・ポイントを介して発行できます。

アプリケーションがこのエントリー・ポイントを使用できるのは、 Communications Server for Linux が verb の処理を完了するのを待機しているときに、そのアプリケーションが一時的に中断可能な場合のみです。

nof_async

NOF verb を非同期形式で発行します。 Communications Server for Linux はすぐにアプリケーションに制御を戻し、 verb 処理がまだ進行中であるか、正常に終了したかを戻り値で示します。戻り値で verb の処理がまだ進行中であることが示された場合、 Communications Server for Linux はアプリケーションが提供するコールバック・ルーチンを使用して verb の処理の結果を戻します。 Communications Server for Linux が要求の処理を完了できる場合、コールバック・ルーチンは呼び出されません。

すべての NOF verb は、このエントリー・ポイントを介して発行できます。 REGISTER_INDICATION_SINK verb と UNREGISTER_INDICATION_SINK verb は、必ずこのエントリー・ポイントを介して発行しなければなりません。

次のいずれかの条件が当てはまる場合、アプリケーションは必ずこのエントリー・ポイントを使用しなければなりません。

- アプリケーションが NOF 指示を受信する必要がある場合。
- Communications Server for Linux が verb の処理を完了するのを待つ間、アプリケーションを中断できません。

nof_async コールバック・ルーチン

非同期 NOF API エントリー・ポイントを使用する場合、アプリケーションはコールバック・ルーチンへのポインターを提供しなければなりません。

Communications Server for Linux は、このコールバック・ルーチンを、verb を完了させるため、および NOF のデータと状況の指示も戻すため、の両方の目的で使用します。

nof および nof_async エントリー・ポイントは、NOF ヘッダー・ファイル **nof_c.h** に定義されています。これらのエントリー・ポイントおよび NOF VCB で使用される AP_UINT32 などのパラメーター・タイプは、NOF ヘッダー・ファイル **nof_c.h** の中にある共通ヘッダー・ファイル **values_c.h** に定義されています。これらのファイルは両方とも **/usr/include/sna** (AIX) または **/opt/ibm/sna/include** (Linux) に保管されます。

同期エントリー・ポイント: nof

アプリケーションは、nof エントリー・ポイントを使用して NOF verb を同期形式で発行します。Communications Server for Linux は、verb の処理が終了するまでアプリケーションに制御を戻しません。

関数コール:

```
void nof (
    AP_UINT32      target_handle,
    void *         nofvcb
);
```

指定パラメーター: アプリケーションは、nof エントリー・ポイントを使用するときに次のパラメーターを提供します。

target_handle

アプリケーションでターゲットの Communications Server for Linux ノードまたはファイルを識別するために使用する ID。このパラメーターは、次のいずれかの方法で指定します。

- 次の verb の場合は、このパラメーターは提供されないため、0 (ゼロ) に設定してください。verb が正常に終了した場合、Communications Server for Linux はターゲット・ハンドルを VCB パラメーターの 1 つとして戻します。アプリケーションは、そのターゲット・ハンドルをあとに続く verb に使用します。
 - CONNECT_NODE (実行中のノードにアクセスするため、または Communications Server for Linux ソフトウェアが起動されたがノードがまだ始動されていないサーバー上のノードにアクセスするため)
 - OPEN_FILE (ドメイン構成ファイルまたは SNA ネットワーク・データ・ファイルにアクセスするため)
- 次の verb の場合は、アプリケーションはヌル値を提供します。
 - QUERY_NODE_ALL (実行中のノードのリストを取得するため)
 - QUERY_CENTRAL_LOGGER
- その他のすべての NOF verb の場合、アプリケーションは CONNECT_NODE verb または OPEN_FILE verb で戻された値を提供します。

nofvcb 発行する verb のパラメーターが入っている verb 制御ブロック (VCB) を指すポインター。各 verb の VCB 構造体については、49 ページの『第 3 章 NOF API verb』で説明しています。これらの構造体は、NOF API ヘッ

ダー・ファイル `/usr/include/sna/nof_c.h` (AIX) または `/opt/ibm/sna/include/nof_c.h` (Linux) に定義されています。

注: NOF VCB には、「予約済み」とマークされたパラメーターが多数あります。一部のパラメーターは Communications Server for Linux ソフトウェアで内部的に使用されます。その他のパラメーターはこのバージョンでは使用されませんが、今後のバージョンで使用される可能性があります。アプリケーションでは、これらの予約済みパラメーターにアクセスしないようにしなければなりません。逆に、`verb` で使用する他のパラメーターを設定する前に、VCB の内容をすべてゼロに設定して、これらのパラメーターすべてが確実にゼロになるようにしなければなりません。このようにすることで、Communications Server for Linux が内部使用するパラメーターを誤って解釈することがなくなり、また、アプリケーション側でも、これらのパラメーターを使用した新規機能が提供される可能性のある今後のバージョンの Communications Server for Linux を継続して使用できるようになります。

VCB の内容をゼロに設定するには、`memset` を使用します。

`memset(nofvcb, 0, sizeof(nofvcb));`

戻り値: `nof` エントリー・ポイントには戻り値がありません。コールが戻ったときに、アプリケーションは VCB 内の戻りコードを調べて `verb` が正常に終了したかどうかを判別し、あとに続く `verb` に必要なパラメーターを決定する必要があります。特に、`CONNECT_NODE verb` または `OPEN_FILE verb` が正常に終了した場合、VCB には、あとに続く `verb` を発行するときにアプリケーションが使用しなければならない `target_handle` が入っています。

同期エントリー・ポイントの使用: 1 つのターゲット・ハンドルで、ある時点で未処理の状態にできる同期 `verb` は 1 つのみです。同期 `verb` は、同じターゲット・ハンドルに別の同期 `verb` が進行中であると、1 次戻りコード `AP_STATE_CHECK` と 2 次戻りコード `AP_SYNC_PENDING` で異常終了します。

非同期エントリー・ポイント: `nof_async`

アプリケーションは、`nof_async` を使用して NOF `verb` を非同期形式で発行します。アプリケーションは、コールバック・ルーチンへのポインターも提供します。Communications Server for Linux はすぐにアプリケーションに制御を戻し、`verb` 処理がまだ進行中であるか、正常に終了したかを戻り値で示します。ほとんどの場合、制御がアプリケーションに戻った時点では、`verb` の処理が続いています。この場合、Communications Server for Linux はアプリケーションが提供したコールバック・ルーチンを使用して、`verb` の処理の結果をあとで戻します。一部のケースでは、Communications Server for Linux がアプリケーションに制御を戻し、Communications Server for Linux がアプリケーションのコールバック・ルーチンを使用しなくなった時点で、`verb` 処理は完了です。

関数コール:

```
AP_UINT16 nof_async(  
    AP_UINT32      target_handle,  
    void *         nofvcb,  
    NOF_CALLBACK   (*comp_proc),  
    AP_CORR        corr
```



```

);

typedef void (*NOF_CALLBACK) (
    AP_UINT32      target_handle,
    void *         nofvcb,
    AP_CORR        corr,
    AP_UINT32      indic_length
);

typedef union ap_corr {
    void *         corr_p;
    AP_UINT32      corr_l;
    AP_INT32       corr_i;
} AP_CORR;

```

NOF_CALLBACK 構造体のパラメーターについては、33 ページの『nof_async エントリー・ポイントで指定したコールバック・ルーチン』を参照してください。

指定パラメーター: アプリケーションは、nof_async エントリー・ポイントを使用するときに 次のパラメーターを提供します。

target_handle

このパラメーターは、次のいずれかの方法で指定します。

- 次の *verb* の場合、このパラメーターは使用されないので、0 (ゼロ) に設定してください。 *verb* が正常に終了した場合、Communications Server for Linux はターゲット・ハンドルを VCB パラメーターの 1 つとして戻します。アプリケーションは、そのターゲット・ハンドルをあとに続く *verb* に使用します。
 - CONNECT_NODE (実行中のノードにアクセスするため、または Communications Server for Linux ソフトウェアが起動されたがノードがまだ始動されていないサーバー上のノードにアクセスするため)
 - OPEN_FILE (ドメイン構成ファイルまたは SNA ネットワーク・データ・ファイルにアクセスするため)
- 次の *verb* の場合は、アプリケーションはヌル値を提供します。
 - QUERY_NODE_ALL (実行中のノードのリストを取得するため)
 - QUERY_CENTRAL_LOGGER
- その他のすべての NOF *verb* の場合、アプリケーションは CONNECT_NODE *verb* または OPEN_FILE *verb* で戻された値を提供します。

nofvcb 発行する *verb* のパラメーターが入っている *verb* 制御ブロック (VCB) を指すポインター。各 *verb* の VCB 構造体については、49 ページの『第 3 章 NOF API *verb*』で説明しています。これらの構造体は、NOF API ヘッダー・ファイル `/usr/include/sna/nof_c.h` (AIX) または `/opt/ibm/sna/include/nof_c.h` (Linux) に定義されています。

注: NOF VCB には、「予約済み」とマークされたパラメーターが多数あります。一部のパラメーターは Communications Server for Linux ソフトウェアで内部的に使用されます。その他のパラメーターはこのバージョンでは使用されませんが、今後のバージョンで使用される可能性があります。アプリケーションでは、これらの予約済みパラメーターにアクセスしないようにしなければなりません。逆に、*verb* で使用する他のパラメーターを設定する前に、VCB の内容をすべてゼロに設定して、これ

らのパラメーターすべてが確実にゼロになるようにしなければなりません。このようにすることで、Communications Server for Linux がその内部使用するパラメーターを誤って解釈することがなくなり、また、アプリケーション側でも、これらのパラメーターを使用した新規機能が提供される可能性のある今後のバージョンの Communications Server for Linux を継続して使用できるようになります。

VCB の内容をゼロに設定するには、memset を使用します。

```
memset(nofvcb, 0, sizeof(nofvcb));
```

comp_proc

verb の完了時に Communications Server for Linux が呼び出すコールバック・ルーチン。コールバック・ルーチンの要件については、33 ページの『nof_async エントリー・ポイントで指定したコールバック・ルーチン』を参照してください。

corr アプリケーションで使用するオプションの相関関係子。このパラメーターは C の共用体として定義され、アプリケーションで 3 つのパラメーター型 (ポインター型、32 ビット整数型、16 ビット整数型) のいずれかを指定できます。

Communications Server for Linux はこの値を使用しませんが、*verb* の完了時にこの値をパラメーターとしてコールバック・ルーチンに渡します。この値をアプリケーションで使用すると、戻された情報をアプリケーションの別の処理へ関連付けることができます。

戻り値: この非同期エントリー・ポイントは、次のいずれかの値を戻します。

AP_COMPLETED

verb は既に完了しています。アプリケーションは VCB 内のパラメーターを検査して、*verb* が正常に完了したかどうかを判別できます。Communications Server for Linux は、この *verb* 用に提供されたコールバック・ルーチンを呼び出しません。

AP_IN_PROGRESS

verb はまだ完了していません。アプリケーションは他の処理を続行でき、現行 *verb* が完了しなくても別の NOF *verb* を発行することもできます。ただし、この *verb* へ提供された VCB 内のパラメーターを検査したり、変更したりしないでください。

Communications Server for Linux は提供されたコールバック・ルーチンを呼び出して、*verb* 処理が完了した時期を示します。そのあと、アプリケーションは VCB パラメーターを検査できます。

非同期エントリー・ポイントの使用: 非同期エントリー・ポイントを使用する場合は、次の点に注意してください。

- アプリケーションで *comp_proc* パラメーターにヌル・ポインターを指定した場合、*verb* は同期形式で完了します (アプリケーションで同期エントリー・ポイントを使用して *verb* を発行した場合と同じ)。
- *nof_async* のコールをアプリケーション・コールバック内から行う場合、*comp_proc* パラメーターにヌル・ポインターを指定することはできません。その

場合、Communications Server for Linux は 1 次戻りコード値 AP_PARAMETER_CHECK および 2 次戻りコード値 AP_SYNC_NOT_ALLOWED で verb をリジェクトします。

- アプリケーションは、コールバック・ルーチンが呼び出されるまで、VCB 内のパラメーターを使用したり、変更しないでください。
- 複数の verb は、必ずしも発行した順序で完了しません。特に、アプリケーションから非同期 verb を発行したあとに同期 verb を発行した場合、同期 verb の完了は非同期 verb が既に完了していることを保証するものではありません。

nof_async エントリー・ポイントで指定したコールバック・ルーチン

非同期 NOF API エントリー・ポイントを使用する場合、アプリケーションはコールバック・ルーチンへのポインタを提供しなければなりません。Communications Server for Linux は、この両方のコールバック・ルーチンを verb を完了させるために使用するだけでなく、NOF 指示を戻すためにも使用します。

(REGISTER_INDICATION_SINK verb もコールバック・ルーチンを指定する非同期 verb として発行され、そのコールバックは指示を受信するたびに呼び出されます。その他の NOF verb の場合、verb が完了したときに指示を受信します。) アプリケーションは、VCB 内の opcode パラメーターを検査して、コールバック・ルーチン内にどのイベントが含まれているかを判別しなければなりません。

この節では、Communications Server for Linux がコールバック・ルーチンを使用する方法と、コールバック・ルーチンが実行しなければならない機能について説明します。

コールバック関数:

```
NOF_CALLBACK (*comp_proc);
typedef void (*NOF_CALLBACK) (
    AP_UINT32      target_handle,
    void *         nofvcb,
    AP_CORR        corr,
    AP_UINT32      indic_length
);
typedef union ap_corr {
    void *         corr_p;
    AP_UINT32     corr_l;
    AP_INT32      corr_i;
} AP_CORR;
```

指定パラメーター: Communications Server for Linux は、以下のパラメーターを使用してコールバック・ルーチンを呼び出します。

target_handle

NOF 指示として、Communications Server for Linux は REGISTER_INDICATION_SINK verb で提供されたターゲット・ハンドルを渡します。verb を完了する場合、このパラメーターは未定義です。

nofvcb 次のいずれかです。

- NOF 指示としては、Communications Server for Linux が提供した VCB へのポインタ。
- verb の完了では、アプリケーションが提供した VCB へのポインタ。VCB には、Communications Server for Linux によって設定された戻りパラメーターが含まれます。

corr アプリケーションが提供した相関関係子の値。この値をアプリケーションで使用すると、戻された情報をアプリケーションの別の処理へ関連付けることができます。

コールバック・ルーチンでは、上記のすべてのパラメーターを使用する必要はありません（『指示用のコールバック・ルーチンの使用』に記載している場合は除く）。コールバック・ルーチンでは、戻されたパラメーターに必要な処理をすべて実行できます。また、*verb* が完了したことを NOF アプリケーションに通知するために変数を設定するのみでも構いません。

戻り値: コールバック関数は値を戻しません。

指示用のコールバック・ルーチンの使用: アプリケーションは NOF *verb* 用に VCB を割り振りますが、Communications Server for Linux は指示用に VCB を割り振ります。したがって、アプリケーションはコールバック・ルーチン内からのみ VCB 情報にアクセスでき、Communications Server for Linux がコールバック・ルーチンに提供した VCB ポインターは、コールバック・ルーチンの外部では無効です。アプリケーションは、必要なすべての処理をコールバック・ルーチン内から実行するか、そのルーチンの外部で使用するために必要な VCB データのコピーを作成しなければなりません。

ターゲット・ハンドルの有効範囲

NOF を使用する必要がある各アプリケーションは、CONNECT_NODE *verb* を発行して、固有のハンドルを取得しなければなりません。2 つの NOF アプリケーションで同じ NOF ターゲット・ハンドルを使用することはできません。

特に、CONNECT_NODE を発行したアプリケーションが、そのあと、fork して子プロセスを作成した場合、子プロセスは、親プロセスが取得したターゲット・ハンドルを使用する NOF *verb* を発行できません。ただし、子プロセスは、別の CONNECT_NODE を発行することにより、その子プロセス自身のターゲット・ハンドルを取得できます。

NOF アプリケーションのコンパイルとリンク

AIX アプリケーション

32 ビット・アプリケーションをコンパイルするには、以下のオプションを使用します。

```
-bimport:/usr/lib/sna/nof_r.exp -I  
/usr/include/sna
```

64 ビットのアプリケーションをコンパイルおよびリンクする場合は、以下のオプションを使用します。

```
-bimport:/usr/lib/sna/nof_r64_5.exp -I  
/usr/include/sna
```

Linux アプリケーション

NOF アプリケーションのコンパイルおよびリンクを実行する前に、アプリケーションが実行時に共有ライブラリーを検出できるように、共用ライブラリー

の保管ディレクトリーを指定します。これを行うには、環境変数 LD_RUN_PATH を /opt/ibm/sna/lib に設定し、64 ビット・アプリケーションをコンパイルする場合は /opt/ibm/sna/lib64 に設定します。

32 ビット・アプリケーションをコンパイルするには、以下のオプションを使用します。

```
-I /opt/ibm/sna/include -L  
/opt/ibm/sna/lib -lnof -lsna_r -lpthread -lpLiS
```

64 ビットのアプリケーションをコンパイルおよびリンクする場合は、以下のオプションを使用します。

```
-I /opt/ibm/sna/include -L  
/opt/ibm/sna/lib64 -lnof -lsna_r -lpthread -lpLiS
```

オプション -lpLiS は、Communications Server for Linux サーバー上でアプリケーションを実行する場合のみ必要です。アプリケーションを IBM Remote API Client 上でビルドして、それをクライアントでのみ実行する場合は、このオプションを使用する必要はありません。このオプションを使用する代わりに、アプリケーションをコンパイルおよびリンクする前に、環境変数の LD_PRELOAD を /usr/lib/libpLiS.so に設定することができます。

Windows に関する考慮事項

WINDOWS

この節では、Windows クライアントで使用する NOF アプリケーションを作成する場合に考慮する必要があるオペレーティング・システム問題について説明します。

Windows の Remote API Client で実行中のアプリケーションは、NOF QUERY_* verb を使用して構成または状況情報を照会できますが、その他の verb を使用して構成を変更したり、リソースを始動および停止することはできません。NOF アプリケーションで構成を変更したり、リソースを始動および停止する必要がある場合は、このアプリケーションを Linux 用に作成して、サーバーで実行しなければなりません。

Windows の NOF API エントリー・ポイント

Windows NOF アプリケーションは、次のエントリー・ポイント関数呼び出しを使用して NOF API にアクセスします。

nof NOF verb を同期形式で発行します。Remote API は、verb の処理が終了するまでアプリケーションに制御を戻しません。

アプリケーションがこのエントリー・ポイントを使用できるのは、Remote API が verb の処理を完了するのを待機するときに、そのアプリケーションが一時的に中断可能な場合のみです。

nof_async

NOF verb を非同期形式で発行します。Remote API はすぐにアプリケーシ

ョンに制御を戻し、verb 処理がまだ進行中であるか、正常に終了したかを戻り値で示します。戻り値で verb の処理がまだ進行中であることが示された場合、verb は後ほど非同期に完了します。Remote API はアプリケーションが提供するイベント・ハンドルをシグナル通知することによって完了を示します。Remote API が要求の処理を完了できる場合、イベント・ハンドルはシグナル通知されません。

Remote API が verb の処理を完了するまで待機する間、アプリケーションを中断できない場合、そのアプリケーションはこのエントリー・ポイントを使用する必要があります。

nof および `nof_async` エントリー・ポイントは、NOF ヘッダー・ファイルの `winnof.h` に定義されています。このファイルは、Windows クライアント・ソフトウェアをインストールしたディレクトリーのサブディレクトリー `¥sdk` (32 ビット・アプリケーションの場合)、または `¥sdk64` (64 ビット・アプリケーションの場合) にインストールされています。これらのエントリー・ポイントおよび NOF VCB で使用される、`AP_UINT32` などのパラメーター・タイプは、共通ヘッダー・ファイル `values.c.h` で定義されています。このファイルは、同じディレクトリーにインストールされており、NOF ヘッダー・ファイル `winnof.h` に含まれています。

同期エントリー・ポイント: `nof`

アプリケーションは、`nof` エントリー・ポイントを使用して NOF verb を同期形式で発行します。Remote API は、verb の処理が終了するまでアプリケーションに制御を戻しません。

関数コール:

```
void WINAPI nof (
    AP_UINT32    target_handle,
    void *       nofvcb
);
```

指定パラメーター: アプリケーションは、`nof` エントリー・ポイントを使用するときに次のパラメーターを提供します。

target_handle

アプリケーションでターゲットの Communications Server for Linux ノードまたはファイルを識別するために使用する ID。このパラメーターは、次のいずれかの方法で指定します。

- 次の verb の場合は、このパラメーターは提供されないので、0 (ゼロ) に設定してください。verb が正常に終了すると、Remote API はターゲット・ハンドルを VCB パラメーターの 1 つとして戻します。アプリケーションは、そのターゲット・ハンドルをあとに続く verb に使用します。
 - `CONNECT_NODE` (実行中のノードにアクセスするため、または Communications Server for Linux ソフトウェアが起動されたがノードがまだ始動されていないサーバー上のノードにアクセスするため)
 - `OPEN_FILE` (ドメイン構成ファイルまたは SNA ネットワーク・データ・ファイルにアクセスするため)
- 次の verb の場合は、アプリケーションはヌル値を提供します。
 - `QUERY_NODE_ALL` (実行中のノードのリストを取得するため)
 - `QUERY_CENTRAL_LOGGER`

- その他のすべての NOF verb の場合、アプリケーションは CONNECT_NODE verb または OPEN_FILE verb で戻された値を提供します。

nofvcb 発行する verb のパラメーターが入っている verb 制御ブロック (VCB) を指すポインター。各 verb の VCB 構造体については、49 ページの『第 3 章 NOF API verb』で説明しています。これらの構造体は、NOF API ヘッダー・ファイル **nof_c.h** に定義されています。

注: NOF VCB には、「予約済み」とマークされたパラメーターが多数あります。一部のパラメーターは Communications Server for Linux ソフトウェアで内部的に使用されます。その他のパラメーターはこのバージョンでは使用されませんが、今後のバージョンで使用される可能性があります。アプリケーションでは、これらの予約済みパラメーターにアクセスしないようにしなければなりません。逆に、verb で使用する他のパラメーターを設定する前に、VCB の内容をすべてゼロに設定して、これらのパラメーターすべてが確実にゼロになるようにしなければなりません。このようにすることで、Communications Server for Linux がその内部使用するパラメーターを誤って解釈することがなくなり、また、アプリケーション側でも、これらのパラメーターを使用した新規機能が提供される可能性のある今後のバージョンの Communications Server for Linux を継続して使用できるようになります。

VCB の内容をゼロに設定するには、memset を使用します。

```
memset(nofvcb, 0, sizeof(nofvcb));
```

戻り値: nof エントリー・ポイントには戻り値がありません。コールが戻ったときに、アプリケーションは VCB 内の戻りコードを調べて verb が正常に終了したかどうかを判別し、あとに続く verb に必要なパラメーターを決定する必要があります。特に、CONNECT_NODE verb または OPEN_FILE verb が正常に終了した場合、VCB には、あとに続く verb を発行するときにアプリケーションが使用しなければならない *target_handle* が入っています。

同期エントリー・ポイントの使用: 1 つのターゲット・ハンドルで、ある時点で未処理の状態にできる同期 verb は 1 つのみです。同期 verb は、同じターゲット・ハンドルに別の同期 verb が進行中であると、1 次戻りコード AP_STATE_CHECK と 2 次戻りコード AP_SYNC_PENDING で異常終了します。

非同期エントリー・ポイント: nof_async

アプリケーションは、nof_async を使用して NOF verb を非同期形式で発行します。アプリケーションは、コールバック・ルーチンへのポインターも提供します。Remote API はすぐにアプリケーションに制御を戻し、verb の処理がまだ進行中であるか、既に完了したかを戻り値で示します。ほとんどの場合、制御がアプリケーションに戻った時点では、verb の処理が続いています。その場合、Remote API はアプリケーションが提供したコールバック・ルーチンを使用して、verb の処理の結果をあとで戻します。一部のケースでは、Remote API がアプリケーションに制御を戻し、Remote API がアプリケーションのコールバック・ルーチンを使用しなくなった時点で、verb 処理は完了です。

関数コール:

Windows に関する考慮事項

```
AP_UINT16 WINAPI nof_async(  
    AP_UINT32    target_handle,  
    void *       nofvcb,  
    NOF_CALLBACK (*comp_proc),  
    AP_CORR      corr  
);  
  
typedef void (*NOF_CALLBACK) (  
    AP_UINT32    target_handle,  
    void *       nofvcb,  
    AP_CORR      corr  
    AP_UINT32    indic_length  
);  
  
typedef union ap_corr {  
    void *       corr_p;  
    AP_UINT32    corr_l;  
    AP_INT32     corr_i;  
} AP_CORR;
```

NOF_CALLBACK 構造体のパラメーターについては、40 ページの『nof_async エントリー・ポイントで指定したコールバック・ルーチン』を参照してください。

指定パラメーター: アプリケーションは、nof_async エントリー・ポイントを使用するときに 次のパラメーターを提供します。

target_handle

このパラメーターは、次のいずれかの方法で指定します。

- 次の verb の場合、このパラメーターは使用されないため、0 (ゼロ) に設定してください。 verb が正常に終了すると、Remote API はターゲット・ハンドルを VCB パラメーターの 1 つとして戻します。アプリケーションは、そのターゲット・ハンドルをあとに続く verb に使用します。
 - CONNECT_NODE (実行中のノードにアクセスするため、または Communications Server for Linux ソフトウェアが起動されたがノードがまだ始動されていないサーバー上のノードにアクセスするため)
 - OPEN_FILE (ドメイン構成ファイルまたは SNA ネットワーク・データ・ファイルにアクセスするため)
- 次の verb の場合は、アプリケーションはヌル値を提供します。
 - QUERY_NODE_ALL (実行中のノードのリストを取得するため)
 - QUERY_CENTRAL_LOGGER
- その他のすべての NOF verb の場合、アプリケーションは CONNECT_NODE verb または OPEN_FILE verb で戻された値を提供します。

nofvcb 発行する verb のパラメーターが入っている verb 制御ブロック (VCB) を指すポインター。各 verb の VCB 構造体については、49 ページの『第 3 章 NOF API verb』で説明しています。これらの構造体は、NOF API ヘッダー・ファイル **nof_c.h** に定義されています。

注: NOF VCB には、「予約済み」とマークされたパラメーターが多数あります。一部のパラメーターは Communications Server for Linux ソフトウェアで内部的に使用されます。その他のパラメーターはこのバージョンでは使用されませんが、今後のバージョンで使用される可能性があります。アプリケーションでは、これらの予約済みパラメーターにアクセ

スしないようにしなければなりません。逆に、verb で使用する他のパラメーターを設定する前に、VCB の内容をすべてゼロに設定して、これらのパラメーターすべてが確実にゼロになるようにしなければなりません。このようにすることで、Communications Server for Linux がその内部使用するパラメーターを誤って解釈することがなくなり、また、アプリケーション側でも、これらのパラメーターを使用した新規機能が提供される可能性のある今後のバージョンの Communications Server for Linux を継続して使用できるようになります。

VCB の内容をゼロに設定するには、memset を使用します。

```
memset(nofvcb, 0, sizeof(nofvcb));
```

comp_proc

verb の完了時に Remote API が呼び出すコールバック・ルーチン。コールバック・ルーチンの要件について詳しくは、40 ページの『nof_async エントリー・ポイントで指定したコールバック・ルーチン』を参照してください。

corr

アプリケーションで使用するオプションの相関関係子。このパラメーターは C の共用体として定義され、アプリケーションで 3 つのパラメーター型 (ポインター型、32 ビット整数型、16 ビット整数型) のいずれかを指定できます。

Remote API はこの値を使用しませんが、verb の完了時にこの値をパラメーターとしてコールバック・ルーチンに渡します。この値をアプリケーションで使用すると、戻された情報をアプリケーションの別の処理へ関連付けることができます。

戻り値: この非同期エントリー・ポイントは、次のいずれかの値を戻します。

AP_COMPLETED

verb は既に完了しています。アプリケーションは VCB 内のパラメーターを検査して、verb が正常に完了したかどうかを判別できます。Remote API は、この verb 用に提供されたコールバック・ルーチンを呼び出しません。

AP_IN_PROGRESS

verb はまだ完了していません。アプリケーションは他の処理を続行でき、現行 verb が完了しなくても別の NOF verb を発行することもできます。ただし、この verb へ提供された VCB 内のパラメーターを検査したり、変更したりしないでください。

Remote API は提供されたコールバック・ルーチンを呼び出して、verb 処理が完了した時期を示します。そのあと、アプリケーションは VCB パラメーターを検査できます。

非同期エントリー・ポイントの使用: 非同期エントリー・ポイントを使用する場合は、次の点に注意してください。

- アプリケーションで *comp_proc* パラメーターにヌル・ポインターを指定した場合、verb は同期形式で完了します (アプリケーションで同期エントリー・ポイントを使用して verb を発行した場合と同じ)。
- *nof_async* のコールをアプリケーション・コールバック内から行う場合、*comp_proc* パラメーターにヌル・ポインターを指定することはできません。その

場合、Remote API は 1 次戻りコード値 AP_PARAMETER_CHECK および 2 次戻りコード値 AP_SYNC_NOT_ALLOWED で verb をリジェクトします。

- アプリケーションは、コールバック・ルーチンが呼び出されるまで、VCB 内のパラメーターを使用したり、変更しないでください。
- 複数の verb は、必ずしも発行した順序で完了しません。特に、アプリケーションから非同期 verb を発行したあとに同期 verb を発行した場合、同期 verb の完了は非同期 verb が既に完了していることを保証するものではありません。

nof_async エントリー・ポイントで指定したコールバック・ルーチン

非同期 NOF API エントリー・ポイントを使用する場合、アプリケーションはコールバック・ルーチンへのポインターを提供しなければなりません。Remote API では、このコールバック・ルーチンを使用して verb の完了を示します。この節では、Remote API がコールバック・ルーチンを使用する方法と、コールバック・ルーチンが実行しなければならない機能について説明します。

コールバック関数:

```
NOF_CALLBACK (*comp_proc);
typedef void (*NOF_CALLBACK) (
    AP_UINT32      target_handle,
    void *         nofvcb,
    AP_CORR        corr,
    AP_UINT32      indic_length
);
typedef union ap_corr {
    void *         corr_p;
    AP_UINT32      corr_l;
    AP_INT32       corr_i;
} AP_CORR;
```

指定パラメーター: Remote API は、以下のパラメーターを使用してコールバック・ルーチンを呼び出します。

target_handle

このパラメーターは未定義です。

nofvcb アプリケーションが提供した VCB へのポインター。VCB には、Remote API によって設定された戻りパラメーターが含まれます。

corr アプリケーションが提供した相関関係子の値。この値をアプリケーションで使用すると、戻された情報をアプリケーションの別の処理へ関連付けることができます。

コールバック・ルーチンでは、上記のすべてのパラメーターを使用する必要はありません。コールバック・ルーチンでは、戻されたパラメーターに必要な処理をすべて実行できます。また、verb が完了したことを NOF アプリケーションに通知するために変数を設定するのみでも構いません。

戻り値: コールバック関数は値を戻しません。

ターゲット・ハンドルの有効範囲

NOF を使用する必要がある各アプリケーションは、CONNECT_NODE verb を発行して、固有のハンドルを取得しなければなりません。2 つの NOF アプリケーションで同じ NOF ターゲット・ハンドルを使用することはできません。

NOF アプリケーションのコンパイルとリンク

この節では、Windows での NOF アプリケーションのコンパイルおよびリンク方法について説明します。

構造体パッキングのためのコンパイラー・オプション

NOF verb の VCB 構造体はパックされません。このパッキング方式を変更するコンパイラー・オプションを使用しないでください。

DWORD パラメーターは *DWORD* 境界、*WORD* パラメーターはワード境界、および *BYTE* パラメーターはバイト境界にあります。

ヘッダー・ファイル

Windows NOF アプリケーションに組み込む NOF ヘッダー・ファイルは、**nof_c.h** という名前です。このファイルは、Windows ソフトウェアに Remote API Client をインストールしたディレクトリーのサブディレクトリー **¥sdk** (32 ビット・アプリケーションの場合)、または **¥sdk64** (64 ビット・アプリケーションの場合) にインストールされています。

ロード時リンク

ロード時にアプリケーションを NOF にリンクするには、TP を API ライブラリー・ファイル **¥sdk¥winnof32.lib** (32 ビット・アプリケーションの場合)、または **¥sdk64¥winnof32.lib** (64 ビット・アプリケーションの場合) にリンクします。

実行時リンク

実行時にアプリケーションを NOF にリンクするには、次の関数を TP に組み込みます。

- LoadLibrary (NOF ダイナミック・リンク・ライブラリー **winnof32.dll** をロードします)
- GetProcAddress (nof か nof_async の、必要な各 NOF エントリー・ポイントを指定します)
- FreeLibrary (ライブラリーが不要の場合)



ポータブル・アプリケーションの作成

以下に、他の環境に移植可能な Communications Server for Linux NOF アプリケーションを作成するためのガイドラインを示します。

- NOF ヘッダー・ファイルをパス名プレフィックスなしで組み込みます。これにより、異なるファイルシステム的环境でアプリケーションを使用できるようになります。コンパイラーの組み込みオプションを使用して、ファイルを検索します (34 ページの『NOF アプリケーションのコンパイルとリンク』または『NOF アプリケーションのコンパイルとリンク』を参照)。
- ヘッダー・ファイルに示される数値ではなく、パラメーター値のシンボリック定数名と戻りコードを使用します。これにより、これらの値がメモリーに格納される方法に関係なく、正しい値が使用されます。

ポータブル・アプリケーションの作成

- 現在のオペレーティング・システムに適用できる戻りコード以外の戻りコードに対する検査を組み込み (switch ステートメントに「デフォルトの」 case を使用するなど)、適切な診断を行います。
- 予約済みとして示されるパラメーターを 0 (ゼロ) に設定します。

NOF verb のターゲット

NOF verb は、次のいずれかのターゲットに対して発行できます。

- 実行中のノード (そのノードのリソースを管理するため)
- **Communications Server for Linux** ソフトウェアを実行しているサーバー上にあり、まだ始動されていないノード (そのノードを始動するか、そのノードの保管された構成情報に照会するか、構成を変更してノードの再始動時にその変更を有効にするため)
- **ドメイン構成ファイル** (ドメイン・リソースを管理するため)
- **sna.net** ファイル (マスター・サーバーを使用できなくなった場合に、バックアップ・マスターとして動作することができる **Communications Server for Linux** サーバーを管理するため)

特定の NOF verb のターゲットは、NOF コールで使用する *target_handle* パラメーターによって識別されます。次のように、アプリケーションは、ターゲットによって異なる NOF verb を使用してターゲット・ハンドルを取得します。

実行中のノードまたは実行中のサーバー上のノード

アプリケーションから、必要なノードの名前とヌルのターゲット・ハンドルを指定して **CONNECT_NODE** を発行します。 **Communications Server for Linux** はファイルのターゲット・ハンドルを **CONNECT_NODE** の **VCB** パラメーターの 1 つとして戻します。

ドメイン構成ファイル

アプリケーションでヌルのターゲット・ハンドルを指定して **OPEN_FILE** を発行します。 **Communications Server for Linux** はファイルのターゲット・ハンドルを **OPEN_FILE** の **VCB** パラメーターの 1 つとして戻します。

sna.net ファイル

アプリケーションでヌルのターゲット・ハンドルを指定して **OPEN_FILE** を発行します。 **Communications Server for Linux** はファイルのターゲット・ハンドルを **OPEN_FILE** の **VCB** パラメーターの 1 つとして戻します。

一部の NOF verb は、次のように特定のターゲット・タイプに対してのみ発行できます。

- **DEFINE_NODE** は、実行中のノードに対して発行することはできません。これは、ノードが実行中でないサーバーに対して発行しなければなりません。
- ノード・リソースに関連した verb (たとえば **DEFINE_LOCAL_LU** など) は、1 つのノードに対して発行しなければなりません。
- ノード・リソースの始動と停止を行う **START_* verb** および **STOP_* verb** は、実行中のノードに対して発行しなければなりません。

- ドメイン・リソースに関連した verb は、ドメイン構成ファイルに対して発行しなければなりません。
- さまざまな QUERY_* verb は、リソースの定義かリソースの現在の状況、またはリソースの定義と状況の両方に関する情報を戻します。状況情報は、実行中のノードのみから入手できます。状況情報のみを戻す verb を非活動ノードに対して発行することはできず、定義と状況の両方を戻す verb を非活動ノードに対して発行すると、定義情報のみが戻されます。たとえば、`QUERY_PARTNER_LU_DEFINITION` は、非活動ノードに対して (保管されている構成情報を判別するために) 発行でき、実行中のノードに対しても (現在の定義を判別するために) 発行できます。ただし、`QUERY_PARTNER_LU` (LU の現在のセッションに関する情報を戻す) は、実行中のノードに対してのみ発行できます。`QUERY_LS` (LS の定義と現在の状況の両方を戻す) は、非活動ノードと実行中のノードのどちらに対しても発行できますが、非活動ノードに対して発行した場合、状況情報は戻されません。49 ページの『第 3 章 NOF API verb』に示す各 QUERY_* verb の説明には、その verb に有効なターゲット・タイプに関する情報も記載しています。
- 管理バックアップ・マスター・サーバーと関連付けられている verb (SNA ネットワーク・ファイル指示の `ADD_BACKUP`、`DELETE_BACKUP`、`QUERY_SNA_NET`、および `REGISTER_INDICATION_SINK` または `UNREGISTER_INDICATION_SINK`) を `sna.net` ファイルに発行する必要があります。

処理モード

アプリケーションで使用される各ターゲット・ハンドルには処理モードが 1 つずつ関連付けられ、その処理モードは NOF verb の `SET_PROCESSING_MODE` を使用して変更できます。モードでは、アプリケーションのファイル・ロックとアクセス許可を制御します。

クライアント上で実行中の NOF アプリケーションの場合、使用可能なモードは読み取り専用モードのみです。このモードでは、QUERY_* verb のみが使用できます。その他のすべての verb (リソースの構成情報または状況を変更するもの) は、拒否されます。これはアプリケーションはこのモードを使用してリソースの構成情報または状況を検査できますが、それらを変更することはできません。

サーバー上で実行中の NOF アプリケーションの場合、次のモードが使用可能です。

AP_MODE_READ_ONLY

このモードでは、QUERY_* verb のみが使用できます。その他のすべての verb (リソースの構成情報または状況を変更するもの) は、拒否されます。

これはターゲット・ハンドルが最初に割り当てられたときのデフォルト・モードです。アプリケーションはこのモードを使用してリソースの構成情報または状況を検査できますが、それらを変更することはできません。

AP_MODE_READ_WRITE

このモードでは、リソースの構成または状況を変更する verb を含めたすべての NOF verb が使用できます。

AP_MODE_COMMIT

このモードは、ターゲット・ハンドルがドメイン構成ファイルを識別する場合（ノードに対して verb を発行する場合ではなく）のみ使用できます。このモードはファイルに対するロックを取得し、そのアプリケーションのみがファイルにアクセスできるようにします。ファイル・ロックによって、そのアプリケーションが一連の verb を発行している間、他のプロセスによってそのファイルが変更されないようにします。また、ファイル・ロックは、発行された一連の verb が完了するまで（アプリケーションが AP_MODE_COMMIT モードから、その他のいずれかのモードに変更されるまで）、そのファイルが変更されないようにします。

このモードは他のプログラムがそのファイルにアクセスできないようにするため、必要なときのみ使用するようになっています。アプリケーションはただちにファイルを変更するために必要な verb をすべて発行し、他のいずれかのモードへ移行しなければなりません。

ファイル・ロックを（たとえば、別のプログラムが現在そのファイルを変更中であるなどの理由で）取得できなかった場合、SET_PROCESSING_MODE verb は失敗します。

注：ファイルへの読み取り/書き込みアクセス権限またはコミット・アクセス権限を取得するには、SNA 管理者グループ sna のメンバーであるユーザー ID を使用して（または root 使用して）、NOF アプリケーションを実行している必要があります。ユーザー ID がこのグループのメンバーでないか、または root でない場合、有効な処理モードは AP_MODE_READ_ONLY のみです。

NOF verb 相互の順序付けと依存関係

NOF verb の順序に関する主な制約事項は、特定のリソースへの最初の参照がそのリソースについての DEFINE_* verb 内に存在しなければならないことです。この制約事項により、次の依存関係が成立します。

- ノード構成ファイルを新たに作成する場合、最初に発行する verb は DEFINE_NODE でなければなりません。
- DLC は、その DLC を参照するポートを定義する前に定義しなければなりません。
- ポートは、そのポートを参照する LS または CN を定義する前に定義しなければなりません。
- COS は、その COS を参照するモードを定義する前に定義しなければなりません。
- PU 名は、その PU を参照する従属 LU を定義する前に定義しなければなりません (LS 定義の一部として)。
- LU は、その LU を含む LU プールを定義する前に定義しなければなりません。
- ダウンストリーム PU 名 (LS 定義の一部として) とホスト LU は、それらを参照するダウンストリーム LU を定義する前に定義しなければなりません。
- リソース名は、そのリソース名を START_* verb が参照する前に定義し、STOP_* verb がそのリソース名を参照する前に開始しなければなりません。

さらに、実行中のノードを変更する場合、`DEFINE_* verb` を 2 回 (前の定義を変更するために) 使用することは、必ずしも有効ではありません。それらの `verb` には、2 回目の定義がまったく有効でない (リソースを削除してから再度定義しなければなりません) ものと、リソースが現在活動状態でない場合に限り 2 回目の定義が有効なものがあります。49 ページの『第 3 章 NOF API verb』に示す各 `DEFINE_* verb` の説明では、2 回目の定義が有効であるかどうかに関する情報も提供します。ドメイン構成ファイルを変更する場合は、以前の定義を変更するために、2 回目の `DEFINE_* verb` を常に使用できます。

ノード構成ファイルを新たに作成する場合、最初に発行する `verb` は `DEFINE_NODE` でなければなりません。そのあとに、そのノードに関連したすべてのリソースについての `DEFINE_* verb` と `SET_* verb` を発行する必要があります。

ドメイン構成ファイルでは、ドメイン・リソース・レコードの順序に関する制約事項はありません。

ノード構成に基づいた NOF の制約事項

`DEFINE_NODE verb` には、ノードがサポートする機能の範囲を定義するパラメーターがあります。いくつかの `NOF verb` は、ノードがサポートできるかサポートできないかを選択するオプション機能に関連しています。したがって、それらの `verb` は、関連する機能をサポートするノードに対して発行した場合に限り有効です。

この節では、どの `NOF verb` を使用できるかに影響を及ぼすオプション機能について要約します。これらの機能について詳しくは、189 ページの『`DEFINE_NODE`』を参照してください。

APPN エンド・ノードと LEN ノードに関する制約事項

Communications Server for Linux に可能なローカル・ノードは、APPN ネットワーク・ノード、APPN 分岐ネットワーク・ノード、APPN エンド・ノード、LEN ノードのいずれかです。

次の `NOF verb` は、ネットワーク・ノード、分岐ネットワーク・ノード、またはエンド・ノードでのみ有効です。これらの `verb` を LEN ノードで発行すると、1 次戻りコード `AP_FUNCTION_NOT_SUPPORTED` が戻されます。

- `DEFINE_CN`
- `DELETE_CN`
- `QUERY_CN`
- `QUERY_CN_PORT`

次の `NOF verb` は、ネットワーク・ノードまたは分岐ネットワーク・ノードでのみ有効です。これらの `verb` をエンド・ノードまたは LEN ノードで発行すると、1 次戻りコード `AP_FUNCTION_NOT_SUPPORTED` が戻されます。

- `QUERY_ADJACENT_NN`
- `QUERY_ISR_SESSION`
- `QUERY_NN_TOPOLOGY_NODE`

ノード構成に基づいた NOF の制約事項

- QUERY_NN_TOPOLOGY_STATS
- QUERY_NN_TOPOLOGY_TG
- 次のいずれかの指示についての REGISTER_INDICATION_SINK
 - ISR_INDICATION
 - NN_TOPOLOGY_NODE_INDICATION
 - NN_TOPOLOGY_TG_INDICATION

複数ドメイン・サポート (MDS) に関する制約事項

ローカル・ノードは、複数ドメイン・サポート (MDS) の有無にかかわらず実行できます。次の NOF verb は、MDS を使って実行されているノードでのみ有効です。これらの verb を MDS がないノードで発行しようとすると、1 次戻りコード AP_FUNCTION_NOT_SUPPORTED が戻されます。

- QUERY_ACTIVE_TRANSACTION
- QUERY_MDS_APPLICATION
- QUERY_MDS_STATISTICS

SNA ゲートウェイと DLUR に関する制約事項

ローカル・ノードは、SNA ゲートウェイまたは DLUR、あるいはその両方についてのサポートの有無にかかわらず実行できます。

次の NOF verb は、SNA ゲートウェイを使用可能にして実行されているノードでのみ有効です。これらの verb を SNA ゲートウェイがないノードで発行すると、1 次戻りコード AP_FUNCTION_NOT_SUPPORTED が戻されます。

- DEFINE_DOWNSTREAM_LU、DEFINE_DOWNSTREAM_LU_RANGE
- DELETE_DOWNSTREAM_LU、DELETE_DOWNSTREAM_LU_RANGE

次の NOF verb は、ノードが DLUR を使用可能にして実行されている場合に限り有効です。これらの verb を DLUR がないノードで発行すると、1 次戻りコード AP_FUNCTION_NOT_SUPPORTED が戻されます。

- DEFINE_DLUR_DEFAULTS
- DEFINE_INTERNAL_PU、DELETE_INTERNAL_PU
- START_INTERNAL_PU、STOP_INTERNAL_PU
- QUERY_DLUR_LU、QUERY_DLUR_PU、QUERY_DLUS

次の NOF verb は、ノードが SNA ゲートウェイまたは DLUR、あるいはその両方を使用可能にして実行されている場合に限り有効です。これらの verb を、その 2 つの機能のどちらもサポートしないノードで発行すると、1 次戻りコード AP_FUNCTION_NOT_SUPPORTED が戻されます。

- QUERY_DOWNSTREAM_LU、QUERY_DOWNSTREAM_PU

QUERY_* verb のリスト・オプション

NOF アプリケーションは、該当するリソース・タイプに対して QUERY_* verb を発行することにより、特定の Communications Server for Linux リソースに関する情報を入手できます。たとえば、QUERY_LS を発行することにより、LS の構成に関する情報を入手できます。これらの verb は、使用するオプションに応じて、特定のリソースに関する情報 (たとえば、特定の LS の構成) か、同じタイプの多数のリソースに関する情報 (たとえば、構成されている LS すべての要約) を戻すことができます。また、一部の QUERY_* verb には、指定したリソースに関する要約情報を戻すか、詳細情報を戻すかを選択するオプションがあります。この節では、それらのオプションの使用方法について説明します。

1 つ以上のリソースに関する情報の入手

QUERY_* verb から戻された情報は、リソース名の順序で並んだリスト形式で保管することができます。たとえば、QUERY_LS が戻す情報は、LS 名の順になっています。リストの通常の順序は、次のとおりです。

- 名前の長さ順 (最も短い名前が最初)
- 同じ長さの名前については ASCII の字句解釈配列

リストの順序がこれと異なる場合 (たとえば、リストが数値の順になっている場合) は、49 ページの『第 3 章 NOF API verb』の各 verb に関する説明でその相違点を示します。

したがって、アプリケーションで完全なリストまたはその指定した一部を要求することにより、複数のリソースに関する情報を入手できます。次の QUERY_* verb のパラメーターは、リストからどのエントリーを戻すかを決定します。

buf_size

戻された情報を受け取るためにアプリケーションが提供するデータ・バッファのサイズ。

num_entries

リソース (それらのリソースに関する情報が戻されます) の最大数。アプリケーションは、ある範囲でなく特定のエントリーを要求するには 1、ある範囲を要求するには 1 より大きい値、できるだけ多数のエントリーを要求するには 0 (ゼロ) を指定できます。

list_options

最初に要求するエントリーのリスト内の位置は次のとおりです。

- リスト内の最初のエントリー
- 指定した名前が付いたエントリーから始まる複数のエントリー
- 指定した名前が付いたエントリーのあとにある最初のエントリーから始まる複数のエントリー。(指定した名前はリストの順序に従って、開始位置が決められ、リスト内になくても構いません。たとえば、リストに NODEA、NODEB、NODED、NODEF などのエントリーがあり、アプリケーションが NODEC のあとにある最初のエントリーから始まるエントリーを要求した場合、最初に戻されるエントリーは NODED です。)

QUERY_* verb のリスト・オプション

さらに、*list_options* パラメーターで最初のエントリーから始めることを要求しなかった場合、リスト内にある特定のエントリーの名前を使用して、求めるエントリーの開始位置を示します。

戻されるエントリー数は、次の値の最小値です。

- *num_entries* パラメーター (ゼロ以外の場合)
- 指定したデータ・バッファーに格納できる最大エントリー数
- 指定した開始位置からリストの末尾までの間のエントリー数

さらにこの verb は、使用可能なエントリーの合計数に関する情報と、すべてのエントリーを一度に戻すために必要なバッファーのサイズに関する情報を戻します。アプリケーションは、求めるすべての情報をまだ受け取っていない場合には、さらに verb を発行して、残りの情報を入手することができます。

これらのオプションをアプリケーションで使用すると、受け取る情報を次のように管理できます。

- 特定のエントリーを入手するには、索引値をそのエントリーの名前に設定し、*list_options* で「指定したエントリーから開始する」ように設定し、*buf_size* をエントリー 1 つ分のサイズ以上に設定し、さらに *num_entries* を 1 に設定します。
- 一度に少数のエントリーずつ、完全なリストを入手するには、最初に *list_options* を「リストの先頭から開始する」ように設定し、戻される情報の量を制限するため、*buf_size* と *num_entries* のどちらかを使用します。戻り値がさらに入手可能な情報があることが示す場合、アプリケーションは *list_options* で「次のエントリーから開始する」ように設定し、最後に受け取ったエントリーの名前を索引値に設定して、もう 1 つの verb を発行します。この 2 番目の verb は、リストの次のセクションを戻します。アプリケーションは、求めるすべてのエントリーを受け取るまで、このプロセスを繰り返します。

要約情報または詳細情報の入手

一部の QUERY_* verb には、指定したリソースに関する要約情報を戻すか、詳細情報を戻すかを選択するオプションがあります。たとえば、QUERY_LOCAL_LU は、LU 名と LU 別名のみ (要約情報) を戻すのみでなく、LU アドレスとセッション限度などの情報 (詳細情報) も戻すことができます。49 ページの『第 3 章 NOF API verb』の各 QUERY_* verb の説明では、その verb が要約情報と詳細情報のどちらかを戻すオプションがあるかどうかを示します。

このオプションを提供する verb では、*list_options* パラメーターが、リスト内の開始位置を示すのみでなく、要約と詳細のどちらの情報が必要であるかを示すためにも使用されます。それらのオプションを指定するには、論理 OR 演算を使用して 2 つの値 (1 つの値はリスト内の開始位置を指定し、もう 1 つの値は要約情報と詳細情報のどちらが必要であるかを示す) を結合し、*list_options* パラメーターをそれら 2 つの値を組み合わせた値に設定します。このオプションをサポートしていない verb の場合は、*list_options* のみをリスト内の開始位置を示す 1 つの値に設定します。

第 3 章 NOF API verb

この章では、各 NOF API verb について次の情報を記載しています。

- NOF API verb の目的と使用方法
- verb をアクティブ・ノード、非アクティブ・ノード、ドメイン構成ファイル、または SNA ネットワーク・データ・ファイルに対して発行できるかどうか (特に断りがない限り、verb はアクティブ・ノードと非アクティブ・ノードのどちらにも発行できます)
- NOF API ヘッダー・ファイル **nof_c.h** に定義された verb 制御ブロック (VCB) 構造体
- アプリケーションが verb に提供するパラメーター
- アプリケーションへ戻されるパラメーター
- 正常に実行されなかった場合のエラー戻りコード

NOF インターフェースに提供したり、戻されたりするパラメーターの多くは 16 進値です。コーディングを単純化するため、値は、NOF ヘッダー・ファイル **nof_c.h** 内にあるヘッダー・ファイル **values_c.h** に定義されている分かりやすい記号定数で表されます。たとえば、ACTIVATE_SESSION verb の *opcode* パラメーターは、記号定数 AP_ACTIVATE_SESSION で表される 16 進値です。ファイル **values_c.h** には、NOF VCB で使用される AP_UINT16 などのパラメーター・タイプの定義も含まれています。

指定パラメーターに値を設定する際、または戻りパラメーターの値をテストする際に、16 進値ではなく記号定数を使用することが重要です。これは、別の Linux システムにおいては、これらの値が異なる方法でメモリーに保管されるため、示される値がシステムで認識できるフォーマットであるとは限らないためです。

この章で説明するエラー戻りコードは、各 verb に固有のもので、その他、すべての NOF API verb に共通する戻りコードについては、851 ページの『付録 B. 共通戻りコード』で説明しています。

さらに、アプリケーションで REGISTER_INDICATION_SINK verb を使用して登録することによって受信できる NOF API 指示は、777 ページの『第 4 章 NOF 指示』で説明しています。

注: NOF VCB には、「予約済み」とマークされたパラメーターが多数あります。一部のパラメーターは Communications Server for Linux ソフトウェアで内部的に使用されます。その他のパラメーターはこのバージョンでは使用されませんが、今後のバージョンで使用される可能性があります。アプリケーションでは、これらの予約済みパラメーターにアクセスしないようにしなければなりません。逆に、verb で使用する他のパラメーターを設定する前に、VCB の内容をすべてゼロに設定して、これらのパラメーターすべてが確実にゼロになるようにしなければなりません。このようにすることで、Communications Server for Linux がその内部使用するパラメーターを誤って解釈することがなくなり、ま

た、アプリケーション側でも、これらのパラメーターを使用した新規機能が提供される可能性のある今後のバージョンの Communications Server for Linux を継続して使用できるようになります。

VCB の内容をゼロに設定するには、`memset` を使用します。

```
memset(nofvcb, 0, sizeof(nofvcb));
```

ACTIVATE_SESSION

`ACTIVATE_SESSION verb` は、ローカル LU と指定したパートナー LU 間のセッションを、指定したモードを使用して活動化するよう Communications Server for Linux に要求します。`cnos_permitted` が `AP_YES` に設定されている場合を除き、`ACTIVATE_SESSION verb` を発行する前に、`INITIALIZE_SESSION_LIMIT verb` を発行しなければなりません。

この verb は、実行中のノードに対して発行しなければなりません。

この verb はクライアント上で実行中の NOF アプリケーションから発行することができます。この verb を AIX または Linux クライアント上で実行するには、NOF アプリケーションは `root` のユーザー ID か、あるいは `sys` グループ (AIX) または `sna` グループ (Linux) のメンバーのユーザー ID を使用して実行しなければなりません。

VCB 構造体

```
typedef struct activate_session
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code         */
    AP_UINT32      secondary_rc;    /* secondary return code       */
    unsigned char  lu_name[8];     /* local LU name               */
    unsigned char  lu_alias[8];    /* local LU alias              */
    unsigned char  plu_alias[8];   /* partner LU alias            */
    unsigned char  mode_name[8];   /* mode name                   */
    unsigned char  fqplu_name[17]; /* fully qualified partner LU name */
    unsigned char  polarity;       /* requested session polarity  */
    unsigned char  session_id[8];  /* session ID                  */
    unsigned char  cnos_permitted; /* is implicit CNOS permitted? */
    unsigned char  reserv4[15];    /* reserved                     */
} ACTIVATE_SESSION;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode `AP_ACTIVATE_SESSION`

lu_name

Communications Server for Linux に対して定義されているローカル LU の LU 名。これは 8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。LU が LU 名でなく LU 別名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

lu_alias

Communications Server for Linux に対して定義されているローカル LU の LU 別名。これは 8 バイトからなる ASCII スtringで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。このパラメーターは、*lu_name* をすべてゼロに設定した場合のみ使用します。

LU 名と LU 別名の両方をすべてゼロに設定した場合、*verb* は CP (デフォルトの LU) に関連した LU へ転送されます。

plu_alias

パートナー LU の LU 別名。これは 8 バイトからなる ASCII スtringで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。パートナー LU が LU 別名でなく完全修飾 LU 名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

mode_name

LU が使用するモードの名前。これは 8 バイトの英数字からなる (英文字で始まる) タイプ A の EBCDIC スtringで、名前が 8 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。

fqplu_name

Communications Server for Linux に対して定義されているパートナー LU の完全修飾 LU 名。このパラメーターは、*plu_alias* フィールドをゼロに設定した場合のみ使用し、*plu_alias* を指定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringです。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

polarity

セッションの極性。値は次のいずれかです。

```
AP_POL_EITHER
AP_POL_FIRST_SPEAKER
AP_POL_BIDDER
```

AP_POL_EITHER を設定した場合、*ACTIVATE_SESSION* は最初のスピーカー・セッションが使用可能であれば、それを活動化し、使用可能でなければ、ビッダー・セッションが活動化されます。*AP_POL_FIRST_SPEAKER* または *AP_POL_BIDDER* を設定した場合、*ACTIVATE_SESSION* は要求した極性のセッションが使用可能である場合のみ成功します。

cnos_permitted

CNOS 処理が許可されていることを示します。値は次のいずれかです。

AP_YES CNOS 処理が許可されています。

AP_NO CNOS 処理が許可されていません。

指定したモードのセッション限度がリセットされたために新規セッションを活動化できず、このパラメーターが *AP_YES* に設定されている場合、暗黙の CNOS 処理によってセッション限度が初期化されます。CNOS 処理が活動状態である間、このコマンドの実行は中断されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
値は次のいずれかです。

AP_AS_NEGOTIATED

セッションは正常に活動化されました。そのモードについて定義されたセッション限度は、活動化プロセスの間に折衝されました。

AP_AS_SPECIFIED

セッションは正常に活動化されました。セッション限度は変更されませんでした。

session_id
8 バイトからなる活動化されたセッションの ID。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_EXCEEDS_MAX_ALLOWED

セッションを活動化すると、この LU-LU モードの組み合わせについての現行セッション限度を超えるため、セッションは活動化されません。

AP_INVALID_LU_ALIAS

lu_alias パラメーターが、定義されているどのローカル LU 別名にも一致しませんでした。

AP_INVALID_LU_NAME

lu_name パラメーターが、定義されているどのローカル LU 名にも一致しませんでした。

AP_INVALID_PLU_NAME

fqplu_name パラメーターが定義されているどのパートナー LU 名にも一致しなかったか、*plu_alias* パラメーターが定義されているどのパートナー LU 名にも一致しませんでした。

AP_INVALID_CNOS_PERMITTED

cnos_permitted パラメーターに指定した値が、有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: セッション起動の失敗時

その他のエラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターのいずれかを戻します。

primary_rc

値は次のいずれかです。

AP_ACTIVATION_FAIL_NO_RETRY

処置が必要な状態 (たとえば、構成の不一致またはセッション・プロトコル・エラーなど) のために、セッションを活動化できませんでした。エラー状態について Communications Server for Linux ログ・ファイルを確認し、エラー状態を訂正してから、この `verb` を再試行してください。

AP_ACTIVATION_FAIL_RETRY

一時的な状態 (たとえばリンクの障害など) のために、セッションを活動化できませんでした。 `verb` を再度実行してください。ただし、エラー状態を解決するためタイムアウトのあとに実行するようにしてください。エラー状態について Communications Server for Linux ログ・ファイルを確認してください。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

ADD_BACKUP

アプリケーションはこの `verb` を使用して、**sna.net** ファイルのバックアップ・マスター・サーバーのリストにサーバーを追加します。これにより、現在のマスターがアクティブでなくなったときに、このサーバーがマスター構成ファイル・サーバーとして動作できるようになります。新規サーバーはリストの最後に追加されます。これにより、このサーバーは、ファイルにリストされているその他すべてのサーバーがアクティブでなくなった場合にのみマスター・サーバーとなります。

この `verb` は、**sna.net** ファイルに対して発行する必要があります。

VCB 構造体

```
typedef struct add_backup
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  backup_name[128]; /* name of backup server to add */
    unsigned char  reserv3[4];     /* reserved                     */
} ADD_BACKUP;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_ADD_BACKUP

ADD_BACKUP

backup_name

バックアップ・サーバーのリストに追加するサーバーの名前。

サーバー名に . (ピリオド) 文字が含まれている場合、Communications Server for Linux はこれが完全修飾名であると想定します。ピリオドが含まれていない場合は、DNS ルックアップを実行してサーバー名を判別します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

戻りパラメーター: 状態の検査

状態エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_DUPLICATE_RECORD

指定したサーバー名は、既にファイルにリストされています。

AP_INVALID_TARGET

NOF API 呼び出しのターゲット・ハンドルで構成ファイルまたはノードが指定されています。この *verb* は、**sna.net** ファイルに対して発行する必要があります。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

ADD_DLC_TRACE

この *verb* は、DLC 上で送信された SNA メッセージについてのトレースを指定します。この *verb* を使用すると、特定の DLC、ポート、LS、HPR RTP 接続、または指定した LS 上の特定のセッションについて、トレースを活動化でき、どのタイプのメッセージをトレースするかを指定できます。また、すべての DLC、ポート、リンク・ステーション、HPR RTP 接続について、トレースを活動化することもできます。Communications Server for Linux トレースの使用方法について詳しくは、「*Communications Server for Linux 管理者用ガイド*」を参照してください。

同じリソースに関連した複数の ADD_DLC_TRACE verb を発行した場合、メッセージは、現在活動状態である verb のいずれかに一致すれば、トレースされます。次に例を示します。

- あるポートとその LS についてのすべてのメッセージをトレースするために verb を発行し、さらに、そのポートが所有する LS の 1 つについて指定された LFSID を持つメッセージのみをトレースするために再度 verb を発行した場合、その LS についてのすべてのメッセージが (それらのメッセージは最初の verb に一致するため) 引き続きトレースされます。そこで、REMOVE_DLC_TRACE を使用してそのポートのトレースを除去した場合、指定した LFSID を持つ LS についてのメッセージは、(まだ活動状態である 2 回目の verb に一致するため) 引き続きトレースされますが、その LS に関するその他のメッセージはトレースされなくなります。
- すべてのリソースについて XID メッセージをトレースするために verb を発行し、さらに、特定の LS に関する SC メッセージと DFC メッセージをトレースするために再度 verb を発行した場合、その LS について 3 つのメッセージ・タイプのすべてがトレースされます。

SDLC 回線をトレースしていて、さらに詳細情報を入手したい場合は、回線トレースだけでなく SDLC の内部トレースを行って、これ入手できます。追加の詳細情報は回線トレースの出力の一部としてフォーマット設定されていますので、SDLC トレースのすべてを 1 つのファイルで見ることができます。詳しくは、751 ページの『SET_TRACE_TYPE』を参照してください。

注: SET_TRACE_TYPE verb にはトレース・ファイルの各エントリを指定の長さに切り捨てるためのオプションがあります。このオプションは、SET_TRACE_TYPE で指定されるカーネル・コンポーネント・トレースと同様に、DLC トレースに適用されます。

VCB 構造体

```
typedef struct add_dlc_trace
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
    DLC_TRACE_FILTER filter;        /* resource to be traced    */
} ADD_DLC_TRACE;

typedef struct dlc_trace_filter
{
    unsigned char  resource_type;    /* type of resource         */
    unsigned char  resource_name[8]; /* name of resource         */
    SNA_LFSID      lfsid;           /* session identifier       */
    unsigned char  message_type;    /* type of messages        */
} DLC_TRACE_FILTER;

typedef struct sna_lfsid
{
    union
    {
        AP_UINT16      session_id;
        struct
        {
            unsigned char sidh;
        };
    };
};
```

```

        unsigned char  sidl;
    } s;
} uu;
AP_UINT16          odai;
} SNA_LFSID;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_ADD_DLC_TRACE

filter.resource_type

トレースするリソースを指定し、オプションでそのリソースについてトレースする具体的なメッセージ・タイプを指定します。値は次のいずれかです。

AP_ALL_RESOURCES

すべての DLC、ポート、リンク・ステーション、HPR RTP 接続のトレース・オプションをセットアップします。

AP_DLC *resource_name* で名前を指定した DLC について、またはその DLC を使用するすべてのポートと LS について、トレースをセットアップします。

AP_PORT

resource_name で名前を指定したポートについて、またはそのポートを使用するすべての LS について、トレースをセットアップします。

AP_LS *resource_name* で名前を指定した LS のトレース・オプションをセットアップします。

AP_RTP_RESOURCE_TYPE

resource_name で名前を指定した RTP 接続の、トレース・オプションを指定します。

AP_PORT_DEFINED_LS

resource_name で名前を指定したポート、またはそのポートを使用する、定義されたすべての LS (暗黙 LS でなく) のトレース・オプションをセットアップします。

AP_PORT_IMPLICIT_LS

resource_name で名前を指定したポート、またはそのポートを使用するすべての暗黙 LS (定義済みの LS でなく) のトレース・オプションをセットアップします。

filter.resource_name

トレースを活動化する DLC、ポート、LS、または RTP 接続の名前。このパラメーターは、*resource_type* が AP_ALL_RESOURCES に設定されている場合は予約済みです。

resource_type が AP_RTP_RESOURCE_TYPE に設定されている場合、特定の RTP 接続の名前 (この名前は @ 文字で始まります) を指定したり、このパラメーターをすべてゼロに設定してすべての RTP トラフィックをトレースするように指示したりできます。

filter.lfsid

指定した LS 上のセッションのローカル・フォーム・セッション ID。これ

は *resource_type* AP_LS の場合のみ有効で、このセッションについてのメッセージのみをトレースすることを示します。この構造体には次の 3 つの値が入っており、これらの値は QUERY_SESSION verb の SESSION_STATS セクションに戻されます。

filter.lfsid.uu.s.sidh

セッション ID の上位バイト。

filter.lfsid.uu.s.sidl

セッション ID の下位バイト。

filter.lfsid.odai

起点宛先アサイナー・インディケータ。

filter.message_type

指定したリソースまたはセッションをトレースするメッセージのタイプ。このパラメーターは、すべてのメッセージをトレースするために AP_TRACE_ALL に設定するか、次の値の 1 つ以上を (論理 OR で結合して) 指定します。

AP_TRACE_XID

XID メッセージ

AP_TRACE_SC

セッション制御 RU

AP_TRACE_DFC

データ・フロー制御 RU

AP_TRACE_FMD

FMD メッセージ

AP_TRACE_SEGS

RH を含まない BBIU 以外のセグメント

AP_TRACE_CTL

MU と XID 以外のメッセージ

AP_TRACE_NLP

ネットワーク層プロトコルのトレース・メッセージ

AP_TRACE_NC

ネットワーク制御のトレース・メッセージ

RTP 接続に対するトレースでは、値 AP_TRACE_XID、AP_TRACE_NLP、AP_TRACE_CTL は無視されます。RTP トレースでは、リストされた値以外の値を最低 1 つ指定しなければなりません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_RESOURCE_TYPE

resource_type パラメーターに、無効な値が指定されました。

AP_INVALID_MESSAGE_TYPE

message_type パラメーターに、無効な値が指定されました。

INVALID_RTP_CONNECTION

resource_name パラメーターがいずれの RTP 接続とも一致しません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

APING

APING は「ping」ユーティリティの APPN バージョンです。これにより、管理アプリケーションは APING を使用すると、ローカル LU からネットワーク内のリモート LU への通信パスを検査できます。

Communications Server for Linux APING は、内部定義された APPC TP を使用してインプリメントされます。この TP はパートナー LU ヘータを送信し、オプションでパートナー LU からデータを受信します。TP が正常に完了した場合、APING `verb` はパートナー LU への会話の割り振りとデータの送受信に要した時間に関する情報を戻します。

アプリケーションは、基本的な APING VCB 構造体のみでなく、要求されたサイズのパートナー TP 検証ストリングを入れるのに十分な VCB を提供しなければなりません。戻されたデータには、このストリングがあり、基本構造体の末尾に付加されています。

この `verb` の目的は、リモート・ノード上の LU へのパスを検査することです。APING を使用してローカル・ノード上のパートナー LU との通信を検査すると、ローカル・コンピューター上の別のプログラムのパフォーマンスに影響を及ぼすため、お勧めできません。

この `verb` は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct aping
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  lu_name[8];     /* local LU name                */
    unsigned char  lu_alias[8];    /* local LU alias               */
    AP_UINT32      sense_data;     /* sense data                   */
    unsigned char  plu_alias[8];   /* partner LU alias             */
    unsigned char  mode_name[8];   /* mode name                    */
    unsigned char  tp_name[64];    /* destination TP name         */
    unsigned char  security;       /* security level               */
    unsigned char  reserv3a[3];    /* reserved                     */
    unsigned char  pwd[10];        /* password                     */
    unsigned char  user_id[10];    /* user ID                      */
    AP_UINT16      dlen;           /* length of data to send      */
    AP_UINT16      consec;         /* number of consecutive sends  */
    unsigned char  fqplu_name[17]; /* fully qualified partner LU name */
    unsigned char  echo;          /* data echo flag               */
    AP_UINT16      iterations;     /* number of iterations        */
    AP_UINT32      alloc_time;     /* time taken for ALLOCATE     */
    AP_UINT32      min_time;       /* minimum send/receive time   */
    AP_UINT32      avg_time;       /* average send/receive time   */
    AP_UINT32      max_time;       /* maximum send/receive time   */
    AP_UINT16      partner_ver_len; /* size of string to receive   */
} APING;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_APING

lu_name

ローカル LU の LU 名。この名前は、8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。LU が LU 名でなく LU 別名で識別されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定し、次に示すパラメーターに LU 別名を指定します。

lu_alias

ローカル LU の LU 別名。このパラメーターは、*lu_name* フィールドを 8 桁の 2 進ゼロに設定した場合のみ使用され、それ以外の場合は無視されます。この名前は 8 バイトからなる ASCII スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。デフォルト LU (CP へ関連付けられている LU) を使用するには、*lu_name* と *lu_alias* の両方のパラメーターを 8 桁の 2 進ゼロに設定します。

plu_alias

パートナー LU 別名。これは、リモート・ノード上にある LU の別名でなければなりません。ローカル・ノード上のパートナー LU に APING を使用することはお勧めできません。

この名前は 8 バイトからなる ASCII スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。LU が別名でなく完全修飾名で識

別されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定し、*fqplu_name* パラメーターに LU 名を指定します。

mode_name

LU の対が使用するモードの名前。これは 8 バイトからなる (英文字で始まる) タイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

tp_name

呼び出し先 TP の名前 (一般に、APINGD に設定されます)。これは、右側にスペースを入れた 64 バイトのストリングです。

security

TP を始動するために会話セキュリティ情報が必要かどうかを指定します。値は次のいずれかです。

AP_NONE

セキュリティ情報は必要ありません。

AP_SAME

セキュリティ情報は、第 3 の TP の代わりにこの TP を呼び出した TP によって検証される場合があります。

AP_PGM TP を起動するためにユーザー ID とパスワードが必要です。

AP_PGM_STRONG

TP を起動するためにはパスワードとユーザー ID が必要ですが、パスワードを平文で送信してはいけません。セッションでパスワードの代用がサポートされていない場合、**aping** は失敗します。それ以外の場合、パスワードは暗号化されて送信されます。

pwd パートナー TP にアクセスするために必要なパスワード。このパラメーターは、セキュリティ・パラメーターを AP_PGM に設定した場合のみ必要です。これは 10 バイトからなるタイプ AE の EBCDIC 文字ストリングで、このパスワードが 10 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。

user_id パートナー TP にアクセスするために必要なユーザ ID。このパラメーターは、セキュリティ・パラメーターを AP_SAME または AP_PGM に設定した場合のみ必要です。これは 10 バイトからなるタイプ AE の EBCDIC 文字ストリングで、このユーザー ID が 10 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。

dlen パートナー LU へ送信するデータ・ストリングの長さ。(NOF API アプリケーションからデータ・ストリングを提供する必要はありません。APING TP は、ゼロからなる指定した長さのストリングを送信するのみです。)

consec 反復するたびにパートナー LU へ送信する連続したデータ・ストリングの数。APING TP は、このデータ・ストリングの数を送信し、それぞれのデータ・ストリングには、*dlen* パラメーターで指定したバイト数が入っています。そのあと、APING TP は、*echo* パラメーターの設定に従って、パートナー TP にデータまたは確認メッセージを要求します。

fqplu_name

パートナー LU の完全修飾ネットワーク名。このパラメーターは、*plu_alias* フィールドを 8 桁の 2 進ゼロに設定した場合のみ使用され、それ以外の場合

合は無視されます。これは、リモート・ノード上にある LU の名前でない限りなりません。ローカル・ノード上のパートナー LU に APING を使用することはお勧めできません。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringです。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

echo APING TP がパートナー LU ヘデータを送信したあと、パートナー LU からのデータを要求するかどうかを指定します。値は次のいずれかです。

AP_YES 指定した数のデータ・Stringを送信したあと、APING はパートナー LU からデータを受信するのを待ちます。

AP_NO 指定した数のデータ・Stringを送信したあと、APING はパートナー LU からの確認のみを要求し、データを受信しません。

iterations

パートナー LU ヘデータを送信してデータまたは確認を要求するシーケンスを APING TP が実行する回数。

partner_ver_len

NOF API アプリケーションが受信できるパートナー TP 検証データ・Stringの最大長。アプリケーションは、基本的な APING VCB 構造体のみでなく、このStringを入れるのに十分な VCB を提供しなければなりません。このStringが戻される VCB に付加されるためです。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、APING は以下のパラメーターを戻します。

primary_rc

AP_OK

alloc_time

会話をパートナーに割り振るミリ秒単位の時間 (APING TP から発行された MC_ALLOCATE *verb* が完了までに要した時間)。

min_time

データ送信の反復に要したミリ秒単位の最小時間 (データを送信してデータまたは確認を受信する 1 回の反復に要した最短の実測時間)。反復をゼロに設定した場合、このパラメーターは使用されません。

avg_time

データ送信の反復に要したミリ秒単位の平均時間 (データを送信してデータまたは確認を受信する 1 回の反復の平均時間)。反復をゼロに設定した場合、このパラメーターは使用されません。

max_time

データ送信の反復に要したミリ秒単位の最大時間 (データを送信してデータまたは確認を受信する 1 回の反復に要した最長の実測時間)。反復をゼロに設定した場合、このパラメーターは使用されません。

partner_ver_len

パートナー TP が戻した検証Stringの長さ。

これらの戻りパラメーターのほかに、パートナー TP が戻す検証ストリングが APING VCB の最後に付加されます。このストリングの長さは *partner_ver_len* で指定します。 *partner_ver_len* がゼロの場合、このストリングは戻されません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_ALIAS

lu_alias パラメーターが、定義されているどの LU 別名とも一致しませんでした。

AP_INVALID_LU_NAME

lu_name パラメーターが、定義されているどの LU 名にも一致しませんでした。

AP_BAD_SECURITY

security パラメーターが、有効な値に設定されていませんでした。

AP_UNKNOWN_PARTNER_MODE

plu_alias、*fqplu_name*、*mode_name* のいずれかに指定した値が、定義されているどのパートナー LU またはモードとも一致しませんでした。

AP_BAD_PARTNER_LU_ALIAS

plu_alias に指定した値が、定義されているどのパートナー LU とも一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 割り振り失敗時

Communications Server for Linux が APPC 会話を割り振ることができなかったために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_ALLOCATION_ERROR

secondary_rc

値は次のいずれかです。

AP_ALLOCATION_FAILURE_NO_RETRY

構成エラーまたはセッション・プロトコル・エラーなどの永続的な状態のために会話を割り振ることができません。詳しくは、*sense_data* パラメーターとエラー・ログ・ファイルを検査してください。エラーの訂正が終了するまで、APING *verb* を再度実行しないでください。

AP_ALLOCATION_FAILURE_RETRY

リンクの障害などの一時的な状態のために会話を割り振ることができませんでした。詳しくは、エラー・ログ・ファイルを検査してください。APING verb を再度実行してください。ただし、エラー状態を解決するため、タイムアウトのあとに実行するようにしてください。

AP_SECURITY_NOT_VALID

指定したユーザー ID またはパスワードが、パートナー LU によって受け入れられませんでした。

AP_TP_NAME_NOT_RECOGNIZED

パートナー LU が、指定した TP 名を認識しませんでした。

AP_TRANS_PGM_NOT_AVAIL_NO_RETRY

リモート LU は、要求されたパートナー TP を起動できなかったため、割り振り要求を拒否しました。この状態は永続的なものです。エラーの原因は、リモート・ノード上のログに記録されている可能性があります。エラーの原因の訂正が終了するまで、APING verb を再度実行しないでください。

AP_TRANS_PGM_NOT_AVAIL_RETRY

リモート LU は、要求されたパートナー TP を起動できなかったため、割り振り要求を拒否しました。この状態は、タイムアウトなどの一時的なものです。エラーの原因は、リモート・ノード上のログに記録されている可能性があります。APING verb を再度実行してください。ただし、エラー状態を解決するため、タイムアウトのあとに実行するようにしてください。

sense_data

secondary_rc パラメーターが AP_ALLOCATION_FAILURE_NO_RETRY である場合、このパラメーターにはエラーに関連した SNA センス・データが入っています。他のすべての *secondary_rc* 値の場合、このパラメーターは予約済みです。

戻りパラメーター：変換失敗時

パートナー TP との APPC 会話が失敗したために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

*primary_rc***AP_CONV_FAILURE_NO_RETRY**

セッション・プロトコル・エラーなどの永続的な状態のために会話が終了しました。エラー・ログ・ファイルを調べて、エラーの原因を判別してください。エラーの訂正が終了するまで、APING verb を再度実行しないでください。

*primary_rc***AP_CONV_FAILURE_RETRY**

一時エラーのために会話が終了しました。APING verb を再度実行してください。問題が再び起こる場合は、エラー・ログ・ファイルを調べてエラーの原因を判別してください。

primary_rc

AP_DEALLOC_ABEND

パートナー TP が、エラー状態のために会話の割り振りを解除しました。エラーの原因は、リモート・ノード上のログに記録されている可能性があります。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

CHANGE_SESSION_LIMIT

CHANGE_SESSION_LIMIT verb は、特定の LU-LU モードの組み合わせについて、セッション限度を変更するように Communications Server for Linux に要求します。この verb を処理した結果、セッションが活動化されるか非活動化される場合があります。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct change_session_limit
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;        /* secondary return code */
    unsigned char  lu_name[8];          /* local LU name */
    unsigned char  lu_alias[8];         /* local LU alias */
    unsigned char  plu_alias[8];        /* partner LU alias */
    unsigned char  fqp_lu_name[17];     /* fully qualified partner
                                        /* LU name */
    unsigned char  reserv3;             /* reserved */
    unsigned char  mode_name[8];        /* mode name */
    unsigned char  reserv3a;            /* reserved */
    unsigned char  set_negotiable;      /* set max negotiable limit? */
    AP_UINT16      plu_mode_session_limit; /* session limit */
    AP_UINT16      min_conwinners_source; /* minimum source contention
                                        /* winner sessions */
    AP_UINT16      min_conwinners_target; /* minimum target contention
                                        /* winner sessions */
    AP_UINT16      auto_act;            /* auto activation limit */
    unsigned char  responsible;         /* who is responsible for
                                        /* deactivating */
    unsigned char  reserv4[3];          /* reserved */
    AP_UINT32      sense_data;         /* sense data */
} CHANGE_SESSION_LIMIT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_CHANGE_SESSION_LIMIT

lu_name

Communications Server for Linux に対して定義されているローカル LU の LU 名。これは 8 バイトからなるタイプ A の EBCDIC スtringで、名

前が 8 バイトに満たない場合は、右側にスペースを入れます。LU が LU 名でなく LU 別名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

lu_alias

Communications Server for Linux に対して定義されているローカル LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。これは、*lu_name* をゼロに設定した場合にのみ使用します。

CP へ関連付けられている LU (デフォルト LU) を示すには、*lu_name* と *lu_alias* の両方を 8 桁の 2 進ゼロに設定します。

plu_alias

パートナー LU の LU 別名。

これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。パートナー LU が LU 別名でなく完全修飾 LU 名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

fqplu_name

Communications Server for Linux に対して定義されているパートナー LU の完全修飾 LU 名。このパラメーターは、*plu_alias* フィールドをゼロに設定した場合のみ使用し、*plu_alias* を指定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

mode_name

LU が使用するモードの名前。

これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。

set_negotiable

このモードの折衝可能な最大セッション限度を変更するかどうかを指定します。値は次のいずれかです。

AP_YES *plu_mode_session_limit* に指定した値を、この LU-LU モードの組み合わせの折衝可能な最大セッション限度として使用します。

AP_NO 折衝可能な最大セッション限度を、そのモードに指定した値のままにしておきます。

plu_mode_session_limit

この LU-LU モードの組み合わせについて要求する合計セッション限度。つまり、このモードを使用するこれら 2 つの LU の間で許可される並列セッションの最大数。1 ~ 32,767 の範囲で値を指定します (値は、ローカル

CHANGE_SESSION_LIMIT

LU に対して DEFINE_LOCAL_LU verb で指定したセッション限度を超えてはなりません)。この値は、パートナー LU と折衝される場合があります。

min_conwinners_source

ローカル LU がコンテンション勝者である、このモードを使用するセッションの最小数。0 ~ 32,767 の範囲の値を指定してください。

min_conwinners_source パラメーターと *min_conwinners_target* パラメーターの合計は、*plu_mode_session_limit* パラメーターを超えてはなりません。

min_conwinners_target

パートナー LU がコンテンション勝者である、このモードを使用するセッションの最小数。0 ~ 32,767 の範囲の値を指定してください。

min_conwinners_source パラメーターと *min_conwinners_target* パラメーターの合計は、*plu_mode_session_limit* パラメーターを超えてはなりません。

auto_act

セッション限度を変更したあとに、自動的に活動化されるセッションの数。0 ~ 32,767 の範囲で値を指定します (値は、*plu_mode_session_limit* パラメーター、あるいはローカル LU に対して DEFINE_LOCAL_LU verb で指定したセッション限度を超えてはなりません)。自動的に活動化されるセッションの実際数は、この値と、ローカル LU のコンテンション勝者セッションの折衝された最小数のどちらか小さい方です。セッションが通常の方法 (AP_DEACT_NORMAL を指定) でこの限度以下で非活動化された場合は、この限度まで新規セッションが活動化されます。

responsible

セッション限度が変更されたあと、ローカルとパートナーのどちらの LU がセッションの非活動化を行うかを示します。値は次のいずれかです。

AP_SOURCE

ローカル LU が行います。

AP_TARGET

パートナー LU が行います。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

値は次のいずれかです。

AP_AS_NEGOTIATED

セッション限度が変更されましたが、パートナー LU によって 1 つ以上の値が折衝されました。

AP_AS_SPECIFIED

セッション限度は要求どおりに変更され、パートナー LU による折衝は行われませんでした。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_EXCEEDS_MAX_ALLOWED

plu_mode_session_limit、*min_conwinners_source*、*min_conwinners_target*、*auto_act* のいずれかのパラメーターが、有効範囲外の値に設定されました。

AP_CANT_CHANGE_TO_ZERO

この `verb` を使用して *plu_mode_session_limit* パラメーターをゼロに設定することはできません。代わりに RESET_SESSION_LIMIT を使用してください。

AP_INVALID_LU_ALIAS

lu_alias パラメーターが、定義されているどのローカル LU 別名にも一致しませんでした。

AP_INVALID_LU_NAME

lu_name パラメーターが、定義されているどのローカル LU 名にも一致しませんでした。

AP_INVALID_MODE_NAME

mode_name パラメーターが、定義されているどのモード名にも一致しませんでした。

AP_INVALID_PLU_NAME

fqplu_name パラメーターが、定義されているどのパートナー LU 名にも一致しませんでした。

AP_INVALID_RESPONSIBLE

responsible パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_SET_NEGOTIABLE

set_negotiable パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

CHANGE_SESSION_LIMIT

AP_MODE_RESET

この LU-LU モードの組み合わせについて、現在活動状態であるセッションはありません。CHANGE_SESSION_LIMIT ではなく INITIALIZE_SESSION_LIMIT を使用して限度を指定してください。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: セッションの割り振りエラー

セッション割り振りエラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_ALLOCATION_ERROR

secondary_rc

AP_ALLOCATION_FAILURE_NO_RETRY

訂正処置が必要な状態のために、セッションを割り振ることができませんでした。sense_data パラメーターとログに記録されたメッセージを検査して障害の理由を判別し、必要な処置を実行してください。エラー状態を訂正してからこの verb を再試行してください。

sense_data

割り振りの障害に関連した SNA センス・データ。

パラメーター: CNOS 処理エラー

エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_CONV_FAILURE_NO_RETRY

処置を必要とする状態 (構成の不一致またはセッション・プロトコル・エラーなど) のために、セッション限度を変更できませんでした。エラー状態について Communications Server for Linux ログ・ファイルを確認し、エラー状態を訂正してから、この verb を再試行してください。

primary_rc

AP_CNOS_PARTNER_LU_REJECT

secondary_rc

AP_CNOS_COMMAND_RACE_REJECT

セッションを活動化または非活動化したり、セッション限度を処理するために別の管理プログラム (または内部的に Communications Server for Linux ソフトウェア) が指定したモードでアクセスしていたため、verb が失敗しました。アプリケーションで verb を再試行してください。ただし、競争状態が解決されるように、タイムアウトのあとで再試行するようにしてください。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

CLOSE_FILE

アプリケーションは、ドメイン構成ファイルへの NOF verb の発行が終了したときに、この verb を使用してドメイン構成ファイルまたは **sna.net** ファイルのハンドルを解放します。アプリケーションでクローズが指定されているファイルは、コールでの *target_handle* パラメーターによって識別されます。

アプリケーションは、終了する前にオープンしているファイル・ハンドルについて必ず CLOSE_FILE を発行しなければなりません。この verb が正常終了したあと、ファイルを識別するターゲット・ハンドルは有効でなくなります。

この verb は、ドメイン構成ファイルまたは **sna.net** ファイルに対して発行する必要があります。

VCB 構造体

```
typedef struct close_file
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
} CLOSE_FILE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_CLOSE_FILE

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_STATE_CHECK

secondary_rc

AP_VERB_IN_PROGRESS

指定したファイルは、前にそのターゲット・ハンドルに対して発行した verb が未処理なので、解放できません。ターゲット・ファイルについてのすべての verb は、そのファイルをクローズする前に完了していなければなりません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

CONNECT_NODE

アプリケーションは communications with a Communications Server for Linux ノード (活動状態または非活動状態) との通信を確立するためにこの verb を使用します。この verb はノードを識別するターゲット・ハンドルを戻し、アプリケーションはそのハンドルを他の NOF verb で使用して、verb のターゲットを示すことができます。

VCB 構造体

```
typedef struct connect_node
{
    AP_UINT16      opcode;          /* verb operation code          */
    unsigned char  reserv2;        /* reserved                      */
    unsigned char  format;        /* reserved                      */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  node_type;     /* which node to connect to    */
    unsigned char  node_name[128]; /* name of Node                 */
    AP_UINT32      target_handle;  /* handle for subsequent verbs  */
    unsigned char  node_status;   /* node status                  */
    unsigned char  reserv3[12];   /* reserved                      */
} CONNECT_NODE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_CONNECT_NODE

node_type

ノードの構成を管理するために特定のノードへ接続するには、このパラメーターを AP_SPECIFIED_NODE に設定します。

中央ロガーとして現在機能しているノードに接続するには、このパラメーターを AP_CENTRAL_LOGGER に設定します。この値は、アプリケーションから次の verb を発行する場合には必須です。

- SET_CENTRAL_LOGGING、QUERY_CENTRAL_LOGGING
- SET_GLOBAL_LOG_TYPE、QUERY_GLOBAL_LOG_TYPE
- SET_LOG_FILE、QUERY_LOG_FILE (中央ロガーが使用中の場合)

node_name

接続先の Communications Server for Linux ノードの名前。このパラメーターは、*node_type* が AP_CENTRAL_LOGGER に設定されている場合は予約済みです。

ノード名に . (ピリオド) 文字が含まれている場合、Communications Server for Linux はこれが完全修飾名であると想定します。ピリオドが含まれていない場合は、DNS ルックアップを実行してノード名を判別します。

Communications Server for Linux が単一のコンピューター上にあるすべてのコンポーネントを使用して実行されている場合は、このパラメーターをすべて 2 進ゼロに設定することができます。ノード名については、指定する必要がありません。これ以外の場合に、このパラメーターを 2 進ゼロに設定すると、デフォルトのローカル・ノード (アプリケーションと同じ Communications Server for Linux サーバー上) が指定されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

target_handle

あとに続く *verb* で使用する戻り値。

node_status

ノードの状況を指定します。値は次のいずれかです。

AP_NDE_STARTING

ノードは、活動化するプロセスを行っています。

AP_NDE_STARTED

ノードは、活動状態です。

AP_NDE_STOPPING

ノードは、非活動化するプロセスを行っています。

AP_NDE_STOPPED

ノードは、活動状態ではありません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

*secondary_rc***AP_INVALID_NODE_NAME**

node_name パラメーターに指定した値が無効です。

CONNECT_NODE

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_CONNECTION_NOT_MADE

ノードへの接続でエラーが起きました。

AP_INVALID_VERSION

アプリケーションが実行されているコンピューターと、ターゲット・ノードが定義されているコンピューター上の Communications Server for Linux ソフトウェアのバージョンが一致しないため、アプリケーションはノードに接続できません。ネットワークのアップグレード処理を行い、異なるコンピューターが異なるレベルの Communications Server for Linux ソフトウェアを実行している場合、バックレベル・ソフトウェアで実行中のノードを管理することができるのは、バックレベル・ソフトウェアで実行中のアプリケーションのみです。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEACTIVATE_CONV_GROUP

DEACTIVATE_CONV_GROUP verb は、指定した会話グループに対応するセッションの非活動化を要求します。この verb は NOF API の一部ですが、主に APPC API を使用した TP を作成するアプリケーション・プログラマーによって使用されます。会話グループ ID は、APPC verb の [MC_]ALLOCATE、[MC_]GET_ATTRIBUTES、および RECEIVE_ALLOCATE によって戻されます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct deactivate_conv_group
{
    AP_UINT16      opcode;          /* verb operation code */
    unsigned char  reserv2;        /* reserved */
    unsigned char  format;         /* reserved */
    AP_UINT16      primary_rc;     /* primary return code */
}
```

```

AP_UINT32      secondary_rc; /* secondary return code */
unsigned char  lu_name[8];   /* local LU name */
unsigned char  lu_alias[8]; /* local LU alias */
AP_UINT32      conv_group_id; /* conversation group identifier */
unsigned char  type;        /* deactivation type */
unsigned char  reserv3[3];  /* reserved */
AP_UINT32      sense_data;  /* deactivation sense data */
} DEACTIVATE_CONV_GROUP;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEACTIVATE_CONV_GROUP

lu_name

Communications Server for Linux に対して定義されているローカル LU の LU 名。これは 8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。LU が LU 名でなく LU 別名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

lu_alias

Communications Server for Linux に対して定義されているローカル LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。これは、*lu_name* をゼロに設定した場合にのみ使用します。

CP へ関連付けられている LU (デフォルト LU) を示すには、*lu_name* と *lu_alias* の両方を 8 桁の 2 進ゼロに設定します。

conv_group_id

非活動化するセッションの会話グループ ID。

type 非活動化のタイプ。値は次のいずれかです。

AP_DEACT_CLEANUP

セッションが終了するのを待たず、すぐにセッションを非活動化します。

AP_DEACT_NORMAL

そのセッションを使用するすべての会話が終了してからセッションを非活動化します。

sense_data

type を AP_DEACT_CLEANUP に設定した場合、このパラメーターはセッションを非活動化するとき使用するセンス・データを指定します。それ以外の場合、このパラメーターは使用しません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_PARAMETER_CHECK`

secondary_rc

値は次のいずれかです。

AP_DEACT_CG_INVALID_CGID

conv_group_id パラメーターが、どの有効な会話グループ ID とも一致しませんでした。

AP_INVALID_CLEANUP_TYPE

type パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LU_ALIAS

lu_alias パラメーターが、定義されているどの LU 別名とも一致しませんでした。

AP_INVALID_LU_NAME

lu_name パラメーターが、定義されているどの LU 名にも一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEACTIVATE_LU_0_TO_3

`DEACTIVATE_LU_0_TO_3 verb` は、3270 エミュレーションまたは LUA に使用する特定の LU (タイプ 0、1、2、3 の LU) のセッションを非活動化するように Communications Server for Linux に要求します。Communications Server for Linux は PLU-SLU セッションについて、`TERM_SELF` メッセージをホストへ送信することによって、セッションを非活動化します。

この `verb` は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct deactivate_lu_0_to_3
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;
    unsigned char  format;
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  lu_name[8];      /* LU Name                       */
} DEACTIVATE_LU_0_TO_3;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEACTIVATE_LU_0_TO_3

lu_name

Communications Server for Linux に対して定義されている LU の LU 名。これは 8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。この戻りコードは、指定 LU に活動セッションがないこと (そのセッションが既に非活動化されていることを意味する) も示す場合があります。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

lu_name パラメーターが、定義されているどの LU 名にも一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEACTIVATE_SESSION

DEACTIVATE_SESSION *verb* は、特定のセッション、または特定のモードのすべてのセッションを非活動化するよう Communications Server for Linux に要求します。

この *verb* は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct deactivate_session
{
    AP_UINT16      opcode;           /* verb operation code */
    unsigned char  reserv2;         /* reserved */
    unsigned char  format;         /* reserved */
}
```

DEACTIVATE_SESSION

```
AP_UINT16      primary_rc;      /* primary return code      */
AP_UINT32      secondary_rc;    /* secondary return code    */
unsigned char   lu_name[8];      /* local LU name            */
unsigned char   lu_alias[8];    /* local LU alias           */
unsigned char   session_id[8];  /* session identifier       */
unsigned char   plu_alias[8];   /* partner LU alias         */
unsigned char   mode_name[8];   /* mode name                */
unsigned char   type;           /* deactivation type        */
unsigned char   reserv3[3];     /* reserved                 */
AP_UINT32      sense_data;     /* deactivation sense data  */
unsigned char   fqplu_name[17]; /* fully qualified partner  */
/* LU name                */
unsigned char   reserv4[20];    /* reserved                 */
} DEACTIVATE_SESSION;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEACTIVATE_SESSION

lu_name

Communications Server for Linux に対して定義されているローカル LU の LU 名。これは 8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。LU が LU 名でなく LU 別名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

lu_alias

Communications Server for Linux に対して定義されているローカル LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。これは、*lu_name* をゼロに設定した場合にのみ使用します。

CP へ関連付けられている LU (デフォルト LU) を示すには、*lu_name* と *lu_alias* の両方を 8 桁の 2 進ゼロに設定します。

session_id

非活動化するセッションを示す 8 バイトの ID。このフィールドを 8 桁の 2 進ゼロに設定した場合、Communications Server for Linux はパートナーの LU とモードのためのすべてのセッションを非活動化します。

plu_alias

パートナー LU の LU 別名。

これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。パートナー LU が LU 別名でなく完全修飾 LU 名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

mode_name

LU が使用するモードの名前。

これは 8 バイトの英数字からなる (英文字で始まる) タイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。

type 非活動化のタイプ。値は次のいずれかです。

AP_DEACT_CLEANUP

セッションが終了するのを待たず、すぐにセッションを非活動化します。

AP_DEACT_NORMAL

そのセッションを使用するすべての会話が終了してからセッションを非活動化します。

sense_data

type を **AP_DEACT_CLEANUP** に設定した場合、このパラメーターはセッションを非活動化するときに使用するセンス・データを指定します。それ以外の場合、このパラメーターは使用しません。

fqplu_name

Communications Server for Linux に対して定義されているパートナー LU の完全修飾 LU 名。このパラメーターは、*plu_alias* フィールドをゼロに設定した場合のみ使用し、*plu_alias* を指定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A String 文字からなるネットワーク名で構成されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。この戻りコードは、セッション ID が活動セッションのセッション ID に一致しなかったこと (そのセッションが既に非活動化されていることを意味する) も示す場合があります。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_CLEANUP_TYPE

type パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LU_ALIAS

lu_alias パラメーターが、定義されているどの LU 別名とも一致しませんでした。

AP_INVALID_LU_NAME

lu_name パラメーターが、定義されているどの LU 名にも一致しませんでした。

DEACTIVATE_SESSION

AP_INVALID_MODE_NAME

mode_name パラメーターが、定義されているどのモード名にも一致しませんでした。

AP_INVALID_PLU_NAME

fqplu_name パラメーターが、定義されているどのパートナー LU 名にも一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_ADJACENT_LEN_NODE

DEFINE_ADJACENT_LEN_NODE は、隣接 LEN ノードとそれに関連した LU のエントリーをノード・ディレクトリー・データベースに追加するか、以前定義した LEN ノードの追加 LU エントリーを追加します。

この verb は、LEN ノードとそれに関連した LU についての一連の DEFINE_DIRECTORY_ENTRY verb と同じ機能で、1 つの verb で LEN ノードの構成を迅速に定義できます。この verb によって作成されたディレクトリー・エントリーを照会するには、QUERY_DIRECTORY_ENTRY を使用します。

この verb を LEN ノード用のサーバーとして機能しているネットワーク・ノードに対して発行した場合、LEN ノードのリソースがネットワーク・ノードのディレクトリー・データベースに追加されます。したがって、ネットワーク・ノードがそれらのリソースを求めるネットワーク検索に回答し、ネットワーク全体からアクセスできるようになります。この verb をエンド・ノードに対して発行した場合、LEN ノードのリソースは、そのエンド・ノードのみからアクセスできます。

VCB 構造体

```
typedef struct define_adjacent_len_node
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  cp_name[17];    /* CP name                   */
    unsigned char  description[32]; /* resource description     */
    unsigned char  reserv1[16];    /* reserved                  */
    unsigned char  num_of_lus;     /* number of LUs            */
    unsigned char  wildcard_lus;  /* wildcard LUs             */
    unsigned char  reserv3[8];     /* reserved                  */
    unsigned char  lu_names[10][8]; /* LU names                  */
} DEFINE_ADJACENT_LEN_NODE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_ADJACENT_LEN_NODE

cp_name

隣接 LEN ノード内の CP の完全修飾名。これは、LEN ノードが XID (サポートしている場合) 上で送信した名前に一致しなければならず、LEN ノードへのリンクの DEFINE_LS 上で指定した隣接 CP 名にも一致しなければなりません。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringです。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

description

隣接 LEN ノードを記述したヌルで終了するテキスト・スString (0 ~ 31 文字のあとにヌル文字を続けたスString)。このスStringは情報専用です。このスStringはノードの構成ファイル内に保管され、`QUERY_DIRECTORY_ENTRY verb` で戻されますが、`Communications Server for Linux` では他の用途に使用されません。

num_of_lus

0 ~ 10 の範囲で定義する LU の数。10 LU を超える隣接ノードを定義するには、同じ CP 名について複数の `DEFINE_ADJACENT_LEN_NODE verb` を使用します。

wildcard_lus

指定する LU 名がワイルドカード・エントリーであるか、それとも明示的な LU 名であるかを示します。値は次のいずれかです。

AP_YES 指定する LU 名はワイルドカード・エントリーです。

AP_NO 指定する LU 名は明示的なエントリーです。

lu_names

LEN ノード上に定義する LU の名前。それぞれの名前は、右側に EBCDIC のスペースを入れた 8 バイトのタイプ A の EBCDIC 文字スStringで、完全修飾 LU 名の 2 番目の部分に対応します (完全修飾名の最初の部分は、上記の *cp_name* パラメーターによって定義されます)。

LEN ノードの制御点に関連した LU (CP LU またはデフォルト LU) を定義するには、ノードの完全修飾 CP 名を *cp_name* パラメーターに指定し、その名前の「ネットワーク名」部分 (EBCDIC のドットのあとの 8 文字) を LU 名の 1 つとして組み込みます。

名前の最初の部分の文字のみを指定することにより、複数の LU 名に一致する「ワイルドカード」LU 名を指定できます。たとえば、ワイルドカード LU 名の「LU」は、「LUNAME」または「LU 01」と一致します (しかし、「NAMELU」とは一致しません)。ただし、1 つの `verb` で指定するすべての LU 名は、*wildcard_lus* パラメーターで定義するように、同じタイプ (ワイルドカードか明示的) の LU 名でなければなりません。同じ LEN ノードについて両方のタイプの LU 名を追加するには、複数の `DEFINE_ADJACENT_LEN_NODE verb` を使用します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_CP_NAME
cp_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_LU_NAME
 指定した LU 名の 1 つ以上に、有効でない文字が入っていました。

AP_INVALID_NUM_LUS
num_of_lus パラメーターが、有効な範囲内にありませんでした。

AP_INVALID_WILDCARD_NAME
wildcard_lus パラメーターを AP_YES に設定しましたが、指定した LU 名の 1 つ以上が、別の親ノード上に既に定義されていました。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_CP_NAME
 指定した CP 名は、既にディレクトリー・エントリー内に定義されており、LEN ノードではありません。

AP_INVALID_LU_NAME
 指定した LU 名の 1 つ以上が、既に別の親ノード上に定義されていました。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_CN

DEFINE_CN は接続ネットワーク (仮想経路指定ノード (Virtual Routing Node: VRN) と呼ばれる) を定義します。この verb は接続ネットワークのネットワーク修飾名と、その伝送グループ (TG) 特性を提供します。その接続ネットワークへアクセスできるローカル・ポートの名前のリストも提供します。

DEFINE_CN は、既存の接続ネットワークを再定義するためにも使用できます。特に、再度 DEFINE_CN を発行すると、接続ネットワークへアクセスするポートのリストへ新しいポートを追加できます (同様に、DELETE_CN verb を発行すると、ポートを除去できます)。

この verb はネットワーク・ノードまたはエンド・ノードでのみ有効であり、LEN ノードでは無効です。

VCB 構造体

```
typedef struct define_cn
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  fqcn_name[17];  /* name of connection network */
    CN_DEF_DATA    def_data;       /* CN defined data          */
    unsigned char  port_name[8][8]; /* port names               */
} DEFINE_CN;

typedef struct cn_def_data
{
    unsigned char  description[32]; /* resource description     */
    unsigned char  reserve0[16];   /* reserved                 */
    unsigned char  num_ports;      /* number of ports on CN   */
    unsigned char  cn_type;        /* reserved                 */
    unsigned char  reserve1[15];   /* reserved                 */
    TG_DEFINED_CHARS tg_chars;     /* TG characteristics      */
} CN_DEF_DATA;

typedef struct tg_defined_chars
{
    unsigned char  effect_cap;     /* effective capacity       */
    unsigned char  reserve1[5];    /* reserved                 */
    unsigned char  connect_cost;  /* connection cost         */
    unsigned char  byte_cost;     /* byte cost               */
    unsigned char  reserve2;      /* reserved                 */
    unsigned char  security;      /* security                 */
    unsigned char  prop_delay;    /* propagation delay       */
    unsigned char  modem_class;   /* reserved                 */
    unsigned char  user_def_parm_1; /* user-defined parameter 1 */
    unsigned char  user_def_parm_2; /* user-defined parameter 2 */
    unsigned char  user_def_parm_3; /* user-defined parameter 3 */
} TG_DEFINED_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_CN

fqcn_name

接続ネットワークの完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文

DEFINE_CN

字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

def_data.description

接続ネットワークを記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、`QUERY_CN verb` で戻されますが、Communications Server for Linux では他の用途に使用されません。

def_data.num_ports

この `verb` で含まれるポートの数。1 つの `DEFINE_CN verb` で最大 8 ポートを指定できます。8 ポートを超えるポートがある CN を定義するには、同じ CN 名に対して複数の `DEFINE_CN verb` を発行します。1 つの CN 上の最大合計ポート数は 239 です。

def_data.tg_chars.effect_cap

実際のビット/秒率 (回線速度)。値は 1 バイトの浮動小数点数としてエンコードされます。このときの公式は、 $0.1 \text{ mmm} * 2^{\text{eeee}}$ で、バイトのビット表現は `b'eeeeemmm'` です。300 ビット/秒の倍数で指定してください。

def_data.tg_chars.connect_cost

接続時間当たりのコスト。有効な値は 0 ~ 255 の範囲の整数値で、0 が接続時間当たりの最低コスト、255 が最高コストです。

def_data.tg_chars.byte_cost

バイト当たりのコスト。有効な値は 0 ~ 255 の範囲の整数値で、0 がバイト当たりの最低コスト、255 が最高コストです。

def_data.tg_chars.security

ネットワークのセキュリティー・レベル。値は次のいずれかです。

AP_SEC_NONSECURE

セキュリティーなし。

AP_SEC_PUBLIC_SWITCHED_NETWORK

公衆交換網を介してデータを伝送します。

AP_SEC_UNDERGROUND_CABLE

保護された地下ケーブルを介してデータを伝送します。

AP_SEC_SECURE_CONDUIT

ガードされていない保護されたコンジット (配管) 内の回線を介してデータを伝送します。

AP_SEC_GUARDED_CONDUIT

物理接続による盗聴から保護されたコンジット内の回線を介してデータを伝送します。

AP_SEC_ENCRYPTED

回線を介して伝送する前にデータを暗号化します。

AP_SEC_GUARDED_RADIATION

物理接続による盗聴および電波盗聴から保護された回線を介してデータを伝送します。

def_data.tg_chars.prop_delay

伝搬遅延: 信号をリンクの長さのみ伝送するのに要する時間。リンクのタイプに応じて、次のいずれかの値を指定します。

AP_PROP_DELAY_MINIMUM

最小の伝搬遅延。

AP_PROP_DELAY_LAN

遅延は 480 マイクロ秒未満です (LAN の通常値)。

AP_PROP_DELAY_TELEPHONE

遅延は 480 ~ 49,512 マイクロ秒の範囲です (電話網の通常値)。

AP_PROP_DELAY_PKT_SWITCHED_NET

遅延は 49,512 ~ 245,760 マイクロ秒の範囲です (パケット交換ネットワークの通常値)。

AP_PROP_DELAY_SATELLITE

遅延は 245,760 マイクロ秒を超えます (サテライト・リンクの通常値)。

AP_PROP_DELAY_MAXIMUM

最大の伝搬遅延。

def_data.tg_chars.user_def_parm_1 ~ def_data.tg_chars.user_def_parm_3

ユーザー定義パラメーター。上記のパラメーターで対応できないその他の TG 特性を組み込むために使用できます。これらのパラメーターは、それぞれ 0 ~ 255 の範囲の値で設定しなければなりません。

port_name

接続ネットワーク上で定義される最大 8 つまでのポート名からなる配列。各ポート名は 8 バイトの ASCII スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。また、DEFINE_PORT verb によって既に定義されていなければなりません。ポート・タイプは接続ネットワークをサポートするネットワーク・タイプ (イーサネット、トークンリング、Enterprise Extender) でなければなりません。接続ネットワーク上に追加ポートを定義するには、新しいポート名を指定して別の DEFINE_CN を発行します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_DEF_LINK_INVALID_SECURITY

security パラメーターが、有効な値のいずれかに設定されていませんでした。

AP_EXCEEDS_MAX_ALLOWED

指定した数のポートを追加すると、1 つの CN 上の最大合計ポート数を超えます。

AP_INVALID_CN_NAME

fqcn_name パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

AP_INVALID_NUM_PORTS_SPECIFIED

num_ports パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_PORT_NAME

指定したポート名の 1 つ以上が、定義されているポートの名前に一致しませんでした。

AP_INVALID_PORT_TYPE

指定した 1 つ以上のポートは、DLC タイプがネットワーク・タイプではなくポイントツーポイント・タイプ (SDLC など) であるため、CN 上に存在できません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_PORT_ACTIVE

指定したポートは現在活動状態であるため、変更できません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードが LEN ノードであるために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは LEN ノードです。この verb は、ネットワーク・ノードまたはエンド・ノードでのみ有効です。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_COS

DEFINE_COS は、サービス・クラス (COS) の定義を追加するか、以前定義した COS を変更します。この定義は TG の「行」とノードの「行」を指定し、経路計算に使用される重みを付けてノードの範囲と TG 特性を関連付けます。重みが低いほど、適切な経路になります。

VCB 構造体

DEFINE_COS verb は可変数の `cos_tg_row` 構造体と `cos_node_row` 構造体を含んでおり、それぞれの番号は `num_of_node_rows` パラメーターと `num_of_tg_rows` パラメーターによって指定されます。TG の行はメインの DEFINE_COS 構造体の末尾に重みの昇順で組み込まれ、そのあとにノードの行も重みの昇順で続きます。

```
typedef struct define_cos
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;                /* reserved                  */
    AP_UINT16      primary_rc;            /* primary return code      */
    AP_UINT32      secondary_rc;          /* secondary return code    */
    unsigned char  cos_name[8];           /* class of service name    */
    unsigned char  description[32];       /* resource description      */
    unsigned char  reserv1[16];           /* reserved                  */
    unsigned char  transmission_priority; /* transmission priority    */
    unsigned char  reserv3[9];            /* reserved                  */
    unsigned char  num_of_node_rows;      /* number of node rows      */
    unsigned char  num_of_tg_rows;        /* number of TG rows        */
} DEFINE_COS;

typedef struct cos_tg_row
{
    TG_DEFINED_CHARS minimum;             /* minimum                   */
    TG_DEFINED_CHARS maximum;             /* maximum                   */
    unsigned char  weight;                 /* weight                    */
    unsigned char  reserv1;                 /* reserved                   */
} COS_TG_ROW;

typedef struct tg_defined_chars
{
    unsigned char  effect_cap;             /* effective capacity        */
    unsigned char  reserv1[5];             /* reserved                   */
    unsigned char  connect_cost;           /* cost per connect time    */
    unsigned char  byte_cost;              /* cost per byte             */
    unsigned char  reserve2;               /* reserved                   */
    unsigned char  security;               /* security                  */
    unsigned char  prop_delay;              /* propagation delay         */
    unsigned char  modem_class;            /* reserved                   */
    unsigned char  user_def_parm_1;        /* user defined parameter 1  */
    unsigned char  user_def_parm_2;        /* user defined parameter 2  */
    unsigned char  user_def_parm_3;        /* user defined parameter 3  */
} TG_DEFINED_CHARS;
```

DEFINE_COS

```
typedef struct cos_node_row
{
    COS_NODE_STATUS    minimum;           /* minimum          */
    COS_NODE_STATUS    maximum;          /* maximum          */
    unsigned char      weight;           /* weight           */
    unsigned char      reserv1;          /* reserved         */
} COS_NODE_ROW;

typedef struct cos_node_status
{
    unsigned char      rar;               /* route additional resistance*/
    unsigned char      status;           /* node status      */
    unsigned char      reserv1[2];       /* reserved         */
} COS_NODE_STATUS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_COS

cos_name

サービス・クラス名。これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れません。

description

COS を記述したヌルで終了するテキスト・String (0 ~ 31 文字のあとにヌル文字を続けたString)。このStringは情報専用です。このStringはノードの構成ファイル内に保管され、QUERY_COS verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

transmission_priority

伝送優先順位。値は次のいずれかです。

AP_LOW

AP_MEDIUM

AP_HIGH

AP_NETWORK

num_of_node_rows

DEFINE_COS VCB に続く (TG 行のあとの) ノード行の数。最大値は 8 です。

num_of_tg_rows

DEFINE_COS VCB に続く TG 行の数。最大値は 8 です。

各 TG 行には、最小 TG 特性セット、最大 TG 特性セット、および重みが含まれます。ある TG の重みを計算する場合、その TG の特性が各 TG 行に定義された最小と最大の特性に突き合わせて検査されます。そのあと、TG に、すべての TG の特性を指定した限度内に制限する、最初の TG 行の重みが割り当てられます。TG の特性が、リストされたどの TG 行にも満たない場合、その TG はこの COS には適さないとみなされ、無限大の重みが割り当てられます。TG 行は、重みの昇順で連結しなければなりません。

cos_tg_row.minimum.effect_cap

実際のビット/秒率の下限 (回線速度)。値は 1 バイトの浮動小数点数としてエンコードされます。このときの公式は、 $0.1 \text{ mmm} * 2^{\text{eeee}}$ で、バイトのビット表現は `b'eeeeemmm'` です。300 ビット/秒の倍数で指定してください。

cos_tg_row.minimum.connect_cost

接続時間当たりのコストの下限。有効な値は 0 ~ 255 の範囲の整数値で、0 が接続時間当たりの最低コスト、255 が最高コストです。

cos_tg_row.minimum.byte_cost

バイト当たりのコストの下限。有効な値は 0 ~ 255 の範囲の整数値で、0 がバイト当たりの最低コスト、255 が最高コストです。

cos_tg_row.minimum.security

最小セキュリティー・レベル。値は次のいずれかです。

AP_SEC_NONSECURE

セキュリティーなし。

AP_SEC_PUBLIC_SWITCHED_NETWORK

公衆交換網を介してデータを伝送します。

AP_SEC_UNDERGROUND_CABLE

保護された地下ケーブルを介してデータを伝送します。

AP_SEC_SECURE_CONDUIT

ガードされていない保護されたコンジット (配管) 内の回線を介してデータを伝送します。

AP_SEC_GUARDED_CONDUIT

物理接続による盗聴から保護されたコンジット内の回線を介してデータを伝送します。

AP_SEC_ENCRYPTED

回線を介して伝送する前にデータを暗号化します。

AP_SEC_GUARDED_RADIATION

物理接続による盗聴および電波盗聴から保護された回線を介してデータを伝送します。

cos_tg_row.minimum.prop_delay

伝搬遅延の下限: 信号をリンクの長さのみ伝送するのに要する時間。リンクのタイプに応じて、次のいずれかの値を指定します。

AP_PROP_DELAY_MINIMUM

最小の伝搬遅延。

AP_PROP_DELAY_LAN

遅延は 480 マイクロ秒未満です (LAN の通常値)。

AP_PROP_DELAY_TELEPHONE

遅延は 480 ~ 49,512 マイクロ秒の範囲です (電話網の通常値)。

AP_PROP_DELAY_PKT_SWITCHED_NET

遅延は 49,512 ~ 245,760 マイクロ秒の範囲です (パケット交換ネットワークの通常値)。

AP_PROP_DELAY_SATELLITE

遅延は 245,760 マイクロ秒を超えます (サテライト・リンクの通常値)。

AP_PROP_DELAY_MAXIMUM

最大の伝搬遅延。

cos_tg_row.minimum.user_def_parm_1 ~ cos_tg_row.user_def_parm_3

ユーザー定義パラメーターの最小値。上記のパラメーターで対応できないその他の TG 特性を組み込むために使用できます。これらのパラメーターは、それぞれ 0 ~ 255 の範囲の値で設定しなければなりません。

cos_tg_row.maximum.effect_cap

実際のビット/秒率の上限 (回線速度)。値は 1 バイトの浮動小数点数としてエンコードされます。このときの公式は、 $0.1 \text{ mmm} * 2^{\text{eeee}}$ で、バイトのビット表現は `b'eeeeemmm'` です。300 ビット/秒の倍数で指定してください。

cos_tg_row.maximum.connect_cost

接続時間当たりのコストの上限。有効な値は 0 ~ 255 の範囲の整数値で、0 が接続時間当たりの最低コスト、255 が最高コストです。

cos_tg_row.maximum.byte_cost

バイト当たりのコストの上限。有効な値は 0 ~ 255 の範囲の整数値で、0 がバイト当たりの最低コスト、255 が最高コストです。

cos_tg_row.maximum.security

最大セキュリティー・レベル。値は次のいずれかです。

AP_SEC_NONSECURE

セキュリティーなし。

AP_SEC_PUBLIC_SWITCHED_NETWORK

公衆交換網を介してデータを伝送します。

AP_SEC_UNDERGROUND_CABLE

保護された地下ケーブルを介してデータを伝送します。

AP_SEC_SECURE_CONDUIT

ガードされていない保護されたコンジット (配管) 内の回線を介してデータを伝送します。

AP_SEC_GUARDED_CONDUIT

物理接続による盗聴から保護されたコンジット内の回線を介してデータを伝送します。

AP_SEC_ENCRYPTED

回線を介して伝送する前にデータを暗号化します。

AP_SEC_GUARDED_RADIATION

物理接続による盗聴および電波盗聴から保護された回線を介してデータを伝送します。

cos_tg_row.maximum.prop_delay

伝搬遅延の上限: 信号をリンクの長さのみ伝送するのに要する時間。リンクのタイプに応じて、次のいずれかの値を指定します。

AP_PROP_DELAY_MINIMUM

最小の伝搬遅延。

AP_PROP_DELAY_LAN

遅延は 480 マイクロ秒未満です (LAN の通常値)。

AP_PROP_DELAY_TELEPHONE

遅延は 480 ~ 49,512 マイクロ秒の範囲です (電話網の通常値)。

AP_PROP_DELAY_PKT_SWITCHED_NET

遅延は 49,512 ~ 245,760 マイクロ秒の範囲です (パケット交換ネットワークの通常値)。

AP_PROP_DELAY_SATELLITE

遅延は 245,760 マイクロ秒を超えます (サテライト・リンクの通常値)。

AP_PROP_DELAY_MAXIMUM

最大の伝搬遅延。

cos_tg_row.maximum.user_def_parm_1 ~ cos_tg_row.maximum.user_def_parm_3

ユーザー定義パラメーターの最大値。上記のパラメーターで対応できないその他の TG 特性を組み込むために使用できます。これらのパラメーターは、それぞれ 0 ~ 255 の範囲の値で設定しなければなりません。

cos_tg_row.weight

この TG 行に関連した重み。

各ノード行には、最小ノード特性セットと最大ノード特性セット、および重みが含まれます。あるノードの重みを計算する場合、そのノードの特性が各ノード行に定義された最小と最大の特性に突き合わせて検査されます。そのあと、ノードに、すべてのノードの特性を指定した限度内に制限する、最初のノード行の重みが割り当てられます。ノードの特性が、リストされたどのノード行にも満たない場合、そのノードはこの COS には適さないとみなされ、無限大の重みが割り当てられます。ノード行は、重みの昇順で並べなければなりません。

cos_node_row.minimum.rar

経路追加抵抗の最小値。値は 0 ~ 255 の範囲でなければなりません。

cos_node_row.minimum.status

ノードの最小輻輳状況を指定します。値は次のいずれかです。

AP_UNCONGESTED

ISR セッションの数は、ノードの構成で *isr_sessions_upper_threshold* の値未満です。

AP_CONGESTED

ISR セッションの数は、しきい値を超えています。

cos_node_row.maximum.rar

経路追加抵抗の最大値。値は 0 ~ 255 の範囲でなければなりません。

cos_node_row.maximum.status

ノードの最大輻輳状況を指定します。値は次のいずれかです。

AP_UNCONGESTED

ISR セッションの数は、ノードの構成で *isr_sessions_upper_threshold* の値未満です。

AP_CONGESTED

ISR セッションの数は、しきい値を超えています。

cos_node_row.weight

このノード行に関連した重み。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_COS_NAME

cos_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_NUMBER_OF_NODE_ROWS

num_of_node_rows パラメーターが、有効な範囲内にありませんでした。

AP_INVALID_NUMBER_OF_TG_ROWS

num_of_tg_rows パラメーターが、有効な範囲内にありませんでした。

AP_NODE_ROW_WGT_LESS_THAN_LAST

ノード行が重みの昇順で並べられていませんでした。

AP_TG_ROW_WGT_LESS_THAN_LAST

TG 行が重みの昇順で並べられていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

AP_COS_TABLE_FULL

新しい COS を定義できません。このノードに許可された COS 定義の最大数 (DEFINE_NODE の *cos_cache_size* パラメーターで指定した値) を超えるためです。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_CPIC_SIDE_INFO

この verb では、サイド情報エントリーを追加するか置換します。CPI-C サイド情報エントリーは、一連の会話特性を記号宛先名に関連付けます。この verb で指定したのと同じ記号宛先名が付いたサイド情報エントリーが既にある場合、そのエントリーは、このコールで指定したデータによって上書きされます。

この verb と CPI-C 関数 *Set_CPIC_Side_Information* の違いに注意してください。この verb はドメイン構成ファイルを変更するため、すべての Communications Server for Linux CPI-C アプリケーションに影響を及ぼします。CPI-C 関数はサイド情報テーブルのメモリーにあるアプリケーション固有のコピーを変更し、他の CPI-C アプリケーションには影響を及ぼしません。

この verb は、ドメイン構成ファイルに対して発行しなければなりません。

VCB 構造体

```
typedef struct define_cplic_side_info
{
    AP_UINT16          opcode;           /* verb operation code */
    unsigned char     reserv2;          /* reserved */
    unsigned char     format;           /* reserved */
    AP_UINT16         primary_rc;       /* primary return code */
    AP_UINT32         secondary_rc;     /* secondary return code */
    unsigned char     reserv2a[8];      /* reserved */
    unsigned char     sym_dest_name[8]; /* Symbolic destination name */
    CPIC_SIDE_INFO_DEF_DATA def_data;
} DEFINE_CPIC_SIDE_INFO;

typedef struct cpic_side_info_def_data
{
    unsigned char     description[32];   /* resource description */
    unsigned char     reserv1[16];      /* reserved */
    CPIC_SIDE_INFO    side_info;        /* CPIC side info */
    unsigned char     user_data[24];    /* reserved */
} CPIC_SIDE_INFO_DEF_DATA;

typedef struct cpic_side_info
{
    unsigned char     partner_lu_name[17]; /* Fully qualified */
                                           /* partner LU name */
    unsigned char     reserved[3];        /* Reserved */
    AP_UINT32         tp_name_type;      /* TP name type */
    unsigned char     tp_name[64];       /* TP name */
    unsigned char     mode_name[8];      /* Mode name */
    AP_UINT32         conversation_security_type; /* Conversation security */
}
```

DEFINE_CPIC_SIDE_INFO

```

                                /* type          */
unsigned char security_user_id[10]; /* User ID      */
unsigned char security_password[10]; /* Password     */
unsigned char lu_alias[8];          /* LU alias     */
} CPIC_SIDE_INFO;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_CPIC_SIDE_INFO

sym_dest_name

サイド情報エントリーを識別する記号宛先名。これは 8 バイトの ASCII ストリングで、必要に応じて右側にスペースを入れます。この名前には、表示可能な文字を含めることができます。

def_data.description

サイド情報エントリーを記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングは構成ファイル内に保管され、**QUERY_CPIC_SIDE_INFO verb** で戻されますが、**Communications Server for Linux** では他の用途に使用されません。

def_data.side_info.partner_lu_name

パートナー LU の完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

def_data.side_info.tp_name_type

ターゲット TP のタイプ (TP 名に有効な文字は、TP タイプによって決まります)。値は次のいずれかです。

XC_APPLICATION_TP

アプリケーション TP。TP 名のすべての文字は、有効な ASCII 文字でなければなりません。

XC_SNA_SERVICE_TP

サービス TP。TP 名は、4 文字の 16 進数字を表す 8 文字の ASCII ストリングで指定しなければなりません。たとえば、名前の 16 進表記が 0x21F0F0F8 の場合、*def_data.side_info.tp_name* パラメーターは、8 文字のストリング「21F0F0F8」に設定します。

先頭文字 (2 バイトで表す) は、0x0 ~ 0x3F の範囲の 16 進値でなければなりません。ただし、0x0E と 0x0F は除きます。残りの文字 (それぞれ 2 バイトで表す) は有効な EBCDIC 文字でなければなりません。

def_data.side_info.tp_name

ターゲット TP の TP 名。これは、右側に ASCII のスペースを入れた 64 バイトの ASCII 文字ストリングです。

def_data.side_info.mode_name

ターゲット TP にアクセスするために使用するモードの名前。これは 8 バイトの ASCII 文字ストリングで、右側にスペースを入れます。

def_data.side_info.conversation_security_type

ターゲット TP が会話セキュリティを使用するかどうかを指定します。値は次のいずれかです。

XC_SECURITY_NONE

ターゲット TP は、会話セキュリティを使用しません。

XC_SECURITY_PROGRAM

ターゲット TP は、会話セキュリティを使用します。ターゲット TP へアクセスするために、以下に指定した *security_user_id* パラメーターと *security_password* パラメーターを使用します。

XC_SECURITY_PROGRAM_STRONG

ローカル・ノードがパスワードを平文でネットワーク上に送信してはならない点を除けば、XC_SECURITY_PROGRAM と同じです。この値を使用できるのは、リモート・システムがパスワード置換をサポートしている場合のみです。

XC_SECURITY_SAME

ターゲット TP は会話セキュリティを使用し、ローカル TP から「検査済み」インディケータを受け入れることができます。(これは、ローカル TP 自体が別の TP から呼び出され、その TP が提供したセキュリティ・ユーザー ID とパスワードをローカル TP が検証済みであることを示しています。) ターゲット TP にアクセスするために、以下に指定した *security_user_id* パラメーターが使用されます。したがって、パスワードは必要ありません。

def_data.side_info.security_user_id

パートナー TP にアクセスするために使用するユーザー ID。このパラメーターは、*conversation_security_type* パラメーターを XC_SECURITY_NONE に設定した場合は必要ありません。

def_data.side_info.security_password

パートナー TP にアクセスするために使用するパスワード。このパラメーターは、*conversation_security_type* パラメーターを XC_SECURITY_PROGRAM または XC_SECURITY_PROGRAM_STRONG に設定した場合のみ必要です。

def_data.side_info.lu_alias

ターゲット TP との通信に使用するローカル LU の別名。この別名は、ローカル・システム上で表示可能な文字を使用した文字ストリングです。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

DEFINE_CPIC_SIDE_INFO

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
```

AP_INVALID_SYM_DEST_NAME

sym_dest_name パラメーターに、有効でない文字が入っていました。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_DEFAULT_PU

DEFINE_DEFAULT_PU は、Communications Server for Linux 管理サービス・データを処理するデフォルト PU を指定します。別の PU 名に対して再度 DEFINE_DEFAULT_PU verb を発行すると、以前の定義は上書きされます。

DEFINE_DEFAULT_PU を使用すると、デフォルト PU のすべてのフィールドの定義、再定義、変更ができます。また、この verb を使用してヌルの PU 名を指定することにより、デフォルト PU を削除することもできます。

アプリケーションで PU 名を指定せずに MS API verb の TRANSFER_MS_DATA を発行した場合、データはローカル・ノードに定義されたデフォルト PU へ転送され、その PU のホスト SSCP とのセッション上へ送信されます。

TRANSFER_MS_DATA について詳しくは、「*Communications Server for Linux MS Programmer's Guide*」を参照してください。

VCB 構造体

```
typedef struct define_default_pu
{
    AP_UINT16      opcode;          /* verb operation code */
    unsigned char  reserv2;        /* reserved */
    unsigned char  format;        /* reserved */
    AP_UINT16      primary_rc;     /* primary return code */
    AP_UINT32      secondary_rc;   /* secondary return code */
    unsigned char  pu_name[8];     /* PU name */
    unsigned char  description[32]; /* resource description */
    unsigned char  reserv1[16];    /* reserved */
    unsigned char  reserv3[16];    /* reserved */
} DEFINE_DEFAULT_PU;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

```
opcode AP_DEFINE_DEFAULT_PU
```

```
pu_name
```

デフォルト PU の名前。これは、直前の DEFINE_LS verb で定義した PU

名でなければなりません。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC ストリングで、必要に応じて右側に EBCDIC のスペースを入れます。

デフォルトの PU を削除するには、すべてゼロを指定します。

description

PU を記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、QUERY_DEFAULT_PU verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_DEFAULTS

DEFINE_DEFAULTS は、ノードが使用するデフォルト・パラメーターを指定します。

VCB 構造体

```
typedef struct define_defaults
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    DEFAULT_CHARS  default_chars;  /* default parameters      */
} DEFINE_DEFAULTS;

typedef struct default_chars
{
    unsigned char  description[32]; /* resource description     */
    unsigned char  reserv2[16];    /* reserved                  */
    unsigned char  mode_name[8];   /* default mode name       */
    unsigned char  implicit_plu_forbidden; /* disallow implicit PLUs? */
    unsigned char  specific_security_codes; /* generic security sense  */
                                     /* codes?                   */
    AP_UINT16      limited_timeout; /* timeout for limited sessions */
    unsigned char  reserv[244];    /* reserved                  */
} DEFAULT_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

```
opcode AP_DEFINE_DEFAULTS
```

DEFINE_DEFAULTS

default_chars.description

デフォルト・パラメーターを記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、`QUERY_DEFAULTS verb` で戻されますが、`Communications Server for Linux` では他の用途に使用されません。

default_chars.mode_name

デフォルト・モードの名前。セッションを開始しようとするときに、認識されていないモード名をアプリケーションで指定した場合、認識されていないモードのデフォルト定義として、このモードからのパラメーターが使用されます。

これは、以前の `DEFINE_MODE verb` によって定義したモードか、1 ページの『NOF API の目的』のリストにある、いずれかの SNA 定義モードでなければなりません。名前は 8 バイトからなる (英字で始まる) タイプ A の EBCDIC ストリングで、必要に応じて右側に EBCDIC のスペースを入れます。

default_chars.implicit_plu_forbidden

`Communications Server for Linux` で不明なパートナー LU の代わりに暗黙の定義を使用するかどうかを指定します。値は次のいずれかです。

AP_YES `Communications Server for Linux` で不明なパートナー LU の代わりに暗黙の定義を使用しません。すべてのパートナーは明示的に定義されなければなりません。

AP_NO `Communications Server for Linux` で不明なパートナー LU の代わりに暗黙の定義を使用します。

default_chars.specific_security_codes

セキュリティーの認証または許可が失敗したときに、`Communications Server for Linux` では特定のセンス・コードを使用するかどうかを指定します。特定のセンス・コードは、セッション上でそれらのセンス・コードのサポートを報告したパートナー LU にのみ戻されます。値は次のいずれかです。

AP_YES `Communications Server for Linux` では特定のセンス・コードを使用します。

AP_NO `Communications Server for Linux` では特定のセンス・コードを使用しません。

default_chars.limited_timeout

使用されていない限定リソース競合勝者セッションを非活動化するまでのタイムアウトを指定します。0 ~ 65,535 秒の範囲で値を指定します。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、`Communications Server for Linux` は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_PARAMETER_CHECK`

secondary_rc

`AP_INVALID_MODE_NAME`

mode_name パラメーターが、定義されているどのモード名にも一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_DIRECTORY_ENTRY

`DEFINE_DIRECTORY_ENTRY` は、ノード・ディレクトリー・データベース内に新しいエントリーを定義します。この `verb` を使用して既存のエントリーを変更することはできません。この `verb` は、ネットワーク修飾リソース名とリソース・タイプ (ネットワーク・ノード、エンド・ノード、LU またはワイルドカード) を提供します。

隣接ノードとその LU を定義する場合は、`DEFINE_DIRECTORY_ENTRY` でなく `DEFINE_ADJACENT_LEN_NODE` を使用するようにしてください。これにより、ノードとその LU を 1 つの `verb` で定義できます。(`DEFINE_DIRECTORY_ENTRY` はエントリーを 1 つのみ定義します。したがって、複数の `verb` を使用して隣接ノードとその LU のエントリーを定義する必要があります。)

データベースは階層状になっているため、各エントリーには親リソースの名前が含まれています。したがって、LU の場合、親リソースは所有側の制御点で、エンド・ノードまたは LEN ノードの場合、親リソースはネットワーク・ノード・サーバーです。ただし、`DEFINE_DIRECTORY_ENTRY` をエンド・ノードまたは LEN ノード上で使用して、直接の通信先となる隣接 LEN ノード・リソースを定義する場合、エントリーに親リソース名は含まれません。

名前の最初の部分の文字のみを指定することにより、複数の LU 名に一致する「ワイルドカード」LU 名を指定できます。たとえば、ワイルドカード LU 名の `APPN.LU` は、`APPN.LUNAME` または `APPN.LU01` と一致します (ただし、`APPN.NAME.LU` とは一致しません)。

VCB 構造体

```
typedef struct define_directory_entry
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
};
```

DEFINE_DIRECTORY_ENTRY

```
unsigned char    format;           /* reserved */
AP_UINT16       primary_rc;       /* primary return code */
AP_UINT32       secondary_rc;    /* secondary return code */
unsigned char    resource_name[17]; /* network qualified resource name */
unsigned char    reserv1a;        /* reserved */
AP_UINT16       resource_type;    /* resource type */
unsigned char    description[32]; /* resource description */
unsigned char    reserv3[16];     /* reserved */
unsigned char    parent_name[17]; /* fully qualified parent name */
unsigned char    reserv1b;        /* reserved */
AP_UINT16       parent_type;     /* parent's resource type */
unsigned char    reserv4[8];     /* reserved */
} DEFINE_DIRECTORY_ENTRY;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_DIRECTORY_ENTRY

resource_name

登録するリソースの完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

resource_type

定義するリソースのタイプを指定します。値は次のいずれかです。

AP_ENCP_RESOURCE

エンド・ノードまたは LEN ノード

AP_NNCP_RESOURCE

ネットワーク・ノード

AP_LU_RESOURCE

LU

AP_WILDCARD_LU_RESOURCE

ワイルドカード LU 名

LU またはワイルドカード LU の場合、親リソース (所有側 CP) のディレクトリー・エントリーが既に定義されていなければなりません。

description

ディレクトリー・エントリーを記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、QUERY_DIRECTORY_ENTRY verb および QUERY_DIRECTORY_LU verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

parent_name

親リソースの完全修飾名。LU の場合、親リソースは所有側の制御点であり、エンド・ノードまたは LEN ノードの場合、親リソースはネットワーク・ノード・サーバーです。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の

A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A String文字からなるネットワーク名で構成されます。

次の場合には、このパラメーターをすべて 2 進ゼロに設定してください。

- ネットワーク・ノード CP を登録する場合
- ローカル・ノードの直接の通信先となる隣接 LEN ノード CP を定義するために、エンド・ノードまたは LEN ノードに対してこの verb を発行する場合

parent_type

定義するリソースの親タイプを指定します。値は次のいずれかです。

AP_ENCP_RESOURCE

エンド・ノード (エンド・ノードによって所有された LU リソースの場合)

AP_NNCP_RESOURCE

ネットワーク・ノード (ネットワーク・ノードによって所有された LU リソースの場合、または EN リソースまたは LEN リソースの場合)

親リソース名を指定しなかった場合は、このパラメーターをゼロに設定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_FQ_OWNING_CP_NAME

parent_name パラメーターが、定義されているリソースの名前に一致しませんでした。

AP_INVALID_LU_NAME

resource_name パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

AP_INVALID_RESOURCE_TYPE

resource_type パラメーターが、有効な値に設定されていませんでした。

DEFINE_DIRECTORY_ENTRY

AP_INVALID_WILDCARD_NAME

resource_type パラメーターが AP_WILDCARD_LU_RESOURCE に設定されていましたが、*resource_name* パラメーターに有効なワイルドカード・エントリーが入っていませんでした。

AP_DUPLICATE

resource_name パラメーターに既に定義されているワイルドカード・エントリーが入っていました。

AP_INVALID_RESOURCE_NAME

resource_name パラメーターに指定したノード名が、*verb* の発行先であるノードの名前と矛盾します。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_DLC

DEFINE_DLC は、新しい DLC を定義します。また、これは DLC が現在活動状態ではなく、その他のパラメーター (DLC タイプ、折衝可能リンク・サポート、および有効なポート・タイプなど) を変更することができない場合に、既存の DLC の DLC 固有パラメーターを変更するために使用することもできます。

VCB 構造体

```
typedef struct define_dlc
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  dlc_name[8];    /* name of DLC                 */
    DLC_DEF_DATA  def_data;       /* DLC defined data            */
} DEFINE_DLC;

typedef struct dlc_def_data
{
    unsigned char  description[32]; /* resource description         */
    unsigned char  initially_active; /* is the DLC initially active? */
    unsigned char  reserv1[15];    /* reserved                     */
    unsigned char  dlc_type;       /* DLC type                    */
    unsigned char  neg_ls_supp;    /* negotiable link station support */
    unsigned char  port_types;     /* port types supported by DLC type */
    unsigned char  hpr_only;      /* only support HPR?          */
    unsigned char  reserv3;       /* reserved                     */
    unsigned char  retry_flags;    /* reserved                     */
    AP_UINT16      max_activation_attempts; /* reserved                 */
    AP_UINT16      activation_delay_timer; /* reserved                 */
    unsigned char  reserv4[4];     /* reserved                     */
    AP_UINT16      dlc_spec_data_len; /* Length of DLC specific data */
} DLC_DEF_DATA;
```

マルチパス・チャンネル (MPC) の DLC 固有データ、System z 上の Communications Server for Linux のみ:

```
typedef struct chnl_dlc_spec_data
{
    V0_MUX_INFO    mux_info;           /* streams information          */
    AP_UINT16      mu_credit;          /* reserved                     */
    unsigned char  stats_support;     /* reserved                     */
    unsigned char  reserve1[31];      /* pad and future expansion     */
} CHNL_DLC_SPEC_DATA;
```

Enterprise Extender (HPR/IP) の DLC 固有データ:

```
typedef struct ipdlc_dlc_spec_data
{
    V0_MUX_INFO    mux_info;           /* streams information          */
    AP_UINT16      udp_port[5];       /* UDP port numbers for traffic */
                                           /* priorities LLC, Network, High, */
                                           /* Medium, Low                 */
    unsigned char  ip_precedence[5];  /* IP precedence 0-7 for traffic */
                                           /* priorities                   */
    unsigned char  no_dns_lookup;     /* are all remote hosts specified by */
                                           /* numeric IP address?         */
} IPDLC_DLC_SPEC_DATA;
```

SDLC の DLC 固有データ:

```
typedef struct sdl_spec_data
{
    V0_MUX_INFO    mux_info;           /* Streams config info         */
    AP_UINT16      mu_credit;          /* amount of credit to allow PC to send*/
    unsigned char  stats_support;     /* activate statistics gathering? */
    unsigned char  reserve1;          /* reserved                     */
    AP_UINT16      sdh_parms_len;     /* Length of HMOD stub create_parms */
    SDH_CREATE_PARMS sdh_parms;      /* HMOD stub create_parms structure */
} SDL_SPEC_DATA;
typedef struct sdh_create_parms
{
    AP_UINT16      length;             /* Length of HMOD stub create_parms */
    AP_UINT16      num_ports;          /* max number of ports DLC can support */
    AP_UINT32      creators_pid;      /* process ID of DLC             */
    V0_MUX_INFO    mux_info;          /* reserved                       */
} SDH_CREATE_PARMS;
```

QLLC の DLC 固有データ:

```
typedef struct vql_dlc_spec_data
{
    V0_MUX_INFO    mux_info;           /* streams config info         */
} VQL_DLC_SPEC_DATA;
```

トークンリング、イーサネットの DLC 固有データ:

```
typedef struct vmc_dlc_cfg
{
    V0_MUX_INFO    mux_info;           /* Streams config info         */
    AP_UINT16      lan_type;           /* type of LAN                 */
    AP_UINT16      min_rcv_dsf;       /* reserved                     */
} VMC_DLC_CFG;
```

すべての DLC タイプの場合:

```
typedef struct v0_mux_info
{
    AP_UINT16      dlc_type;           /* DLC implementation type     */
    unsigned char  need_vrfy_fixup;   /* reserved                     */
    unsigned char  num_mux_ids;       /* reserved                     */
}
```

```

AP_UINT32      card_type;           /* type of adapter card      */
AP_UINT32      adapter_number;     /* DLC adapter number        */
AP_UINT32      oem_data_length;    /* reserved                   */
AP_INT32       mux_ids[5];         /* reserved                   */
} V0_MUX_INFO;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_DLC

dlc_name

DLC の名前。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

def_data.description

DLC を記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、**QUERY_DLC verb** で戻されますが、**Communications Server for Linux** では他の用途に使用されません。

def_data.initially_active

ノードの始動時に DLC を自動的に起動するかどうかを指定します。値は次のいずれかです。

AP_YES DLC はノードの始動時に自動的に起動します。

AP_NO DLC は、その DLC を使用するポートまたは LS が初期設定で活動状態になるように定義されている場合にのみ自動的に起動します。それ以外の場合は、手動で起動する必要があります。

def_data.dlc_type

DLC のタイプ。既存の DLC に対するこのパラメーターは変更できません。このパラメーターは、新規 DLC の作成時にのみ指定できます。値は次のいずれかです。

AP_SDLC

SDLC

AP_X25 QLLC

AP_TR トークンリング

AP_ETHERNET

イーサネット

AP_MPC マルチパス・チャネル (MPC)、System z 上の **Communications Server for Linux** のみ

AP_IP Enterprise Extender (HPR/IP)

def_data.neg_ls_supp

DLC が折衝可能リンク・ステーションをサポートするかどうかを指定します。既存の DLC に対するこのパラメーターは変更できません。このパラメーターは、新規 DLC の作成時にのみ指定できます。 *dlc_type* を **AP_QLLC**

に設定した場合は、これを AP_YES に設定しなければなりません。 *dlc_type* を AP_MPC に設定した場合は、これを AP_YES に設定しなければなりません。

値は次のいずれかです。

AP_YES この DLC を使用するリンク・ステーションは、折衝可能にできます。

AP_NO この DLC を使用するリンク・ステーションは、1 次リンク・ステーションまたは 2 次リンク・ステーションのいずれかとして定義する必要があります。折衝可能リンク・ステーションはサポートされません。

def_data.port_types

dlc_type を AP_TR、AP_ETHERNET、AP_IP のいずれかに設定した場合は、このパラメーターを AP_PORT_SATF に設定します。*dlc_type* を AP_MPC に設定した場合は、このパラメーターは AP_PORT_SWITCHED に設定します。他のすべての DLC タイプの場合、このパラメーターは予約済みです。

def_data.hpr_only

DLC が HPR トラフィックのみをサポートするかどうかを指定します。

dlc_type を AP_IP に設定した場合は、これを AP_YES に設定しなければなりません。*dlc_type* を AP_MPC に設定した場合は、これを AP_NO に設定する必要があります。値は次のいずれかです。

AP_YES この DLC を Enterprise Extender のリンクに使用し、HPR トラフィックのみをサポートします。

AP_NO この DLC は Enterprise Extender 以外のリンク・タイプで使用され、非 HPR トラフィックをサポートします。HPR トラフィックもサポートできます。

def_data.dlc_spec_data_len

その DLC のタイプに固有なデータの長さ (バイト単位)。DLC 固有のデータ構造体は、基本 VCB 構造体の末尾になければなりません。

Enterprise Extender (HPR/IP) の DLC 固有データ:

ipdlc_dlc_spec_data.mux_info.dlc_type

DLC のタイプ。これは AP_IP に設定します。

ipdlc_dlc_spec_data.mux_info.card_type

アダプター・カードのタイプ。これは AP_CARD_IP に設定します。

ipdlc_dlc_spec_data.mux_info.adapter_number

このパラメーターは予約済みです (ゼロに設定します)。

ipdlc_dlc_spec_data.udp_port

異なるトラフィック優先順位において DLC が使用する 5 つの UDP ポート番号の配列。これらは通常、12000 ~ 12004 に設定されます。

udp_port[0]

LLC コマンドで使用する UDP ポート。

udp_port[1]

ネットワーク優先順位トラフィックで使用する UDP ポート。

udp_port[2]

高優先順位トラフィックで使用する UDP ポート。

udp_port[3]

中優先順位トラフィックで使用する UDP ポート。

udp_port[4]

低優先順位トラフィックで使用する UDP ポート。

ipdlc_dlc_spec_data.ip_precedence

異なるトラフィック優先順位において DLC が使用する 5 つの IP 優先順位値の配列。この配列における各項目は、0 (最小) ~ 7 (最大) の範囲の値です。

ip_precedence[0]

LLC コマンドで使用する IP 優先順位。これは通常、6 に設定します。

ip_precedence[1]

ネットワーク優先順位トラフィックで使用する IP 優先順位。これは通常、6 に設定します。

ip_precedence[2]

高優先順位トラフィックで使用する IP 優先順位。これは通常、4 に設定します。

ip_precedence[3]

中優先順位トラフィックで使用する IP 優先順位。これは通常、2 に設定します。

ip_precedence[4]

低優先順位トラフィックで使用する IP 優先順位。これは通常、1 に設定します。

ipdlc_dlc_spec_data.no_dns_lookup

リモート・ホストの IP アドレスについて、ドメイン・ネーム・サーバーでルックアップすることが必要であるかどうかを指定します。値は次のいずれかです。

AP_YES 着信 IP 接続を受信したときに、リモート IP アドレスからホスト名のルックアップを試行しません。

リモート IP アドレスを解決できない場合は、このオプションを使用します。このケースでは、着信接続は LS がホスト名ではなく明示的 IP アドレス (IPv4 または IPv6 のいずれか) を使用するよう構成されている場合のみ、構成済みの LS と一致する可能性があります。

AP_NO この DLC に定義されている各リンク・ステーションのリモート・ホストの IP アドレスは、数値アドレス (IPv4 または IPv6 のいずれか)、名前 (newbox.this.co.uk など)、または別名 (newbox など) で指定できます。ノードはドメイン・ネーム・サーバー・ルックアップを実行して、必要などころのすべての着呼におけるリモート・ホスト名を判別します。

MPC の DLC 固有データ、System z 上の Communications Server for Linux のみ:

chnl_dlc_spec_data.mux_info.dlc_type

DLC のタイプ。これは AP_IMPL_MPC_GDLC に設定してください。

chnl_dlc_spec_data.mux_info.card_type

アダプター・カードのタイプ。これは AP_CARD_IBM_ESCON に設定してください。

chnl_dlc_spec_data.mux_info.adapter_number

このパラメーターは予約済みです (ゼロに設定します)。

SDLC の DLC 固有データ:

sdl_spec_data.mux_info.dlc_type

DLC のタイプ。これは AP_IMPL_SDLC_SL に設定します。

sdl_spec_data.mux_info.card_type

アダプター・カードのタイプ。

値は次のいずれかです。

AP_CARD_IBM_SDLC

AP_CARD_IBM_MPCA

AP_CARD_IBM_MPA

AP_CARD_DCA_ISCA

sdl_spec_data.mux_info.adapter_number

DLC が使用するアダプター番号。サーバーに複数の SDLC アダプター・カードがある場合、最初のカードには 0 (ゼロ)、2 番目のカードには 1 というように指定します。それ以外の場合、このパラメーターは 0 (ゼロ) に設定してください。

sdl_spec_data.mu_credit

DLC_MU をリンク・コンポーネントへ送信するためのクレジット値を指定します。このパラメーターには 4 を設定します。

sdl_spec_data.stats_support

DLC により、リンク統計情報が収集されるかどうかを指定します。値は次のいずれかです。

AP_YES DLC により、QUERY_STATISTICS を使用して調べることができるリンク統計情報が収集されます。

AP_NO DLC により、リンク統計情報は収集されません。

sdl_spec_data.sdh_parms_len

sdh_parms 構造体の長さ (バイト)。これは sizeof(SDH_CREATE_PARMS) に設定します。

sdl_spec_data.sdh_parms.length

sdh_parms 構造体の長さ (バイト)。これは sizeof(SDH_CREATE_PARMS) に設定します。

sdl_spec_data.sdh_parms.num_ports

この DLC が同時にサポートする必要のあるポートの最大数。

sdl_spec_data.sdh_parms.creators_pid

このパラメーターは予約済みです (ゼロに設定します)。

QLLC の DLC 固有データ:

DEFINE_DLC

vql_dlc_spec_data.mux_info.dlc_type

DLC のタイプ。これは AP_IMPL_NLI_QLLC に設定します。

vql_dlc_spec_data.mux_info.card_type

アダプター・カードのタイプ。これは AP_CARD_QLLC_NLI に設定します。

vql_dlc_spec_data.mux_info.adapter_number

DLC が使用するアダプター番号。サーバーに、複数の X.25 アダプター・カードがある場合、最初のカードにはゼロ、2 番目のカードには 1 というように指定します。それ以外の場合、このパラメーターはゼロに設定してください。

トークンリング、イーサネットの DLC 固有データ:

vmc_dlc_cfg.mux_info.dlc_type

DLC のタイプ。値は次のいずれかです。

AP_IMPL_TR_SNAP_LLC2

トークンリング

AP_IMPL_ETHER_SNAP_LLC2

イーサネット

vmc_dlc_cfg.mux_info.card_type

アダプター・カードのタイプ。

値は次のいずれかです。

AP_CARD_TOKEN_RING_LLI

トークンリング

AP_CARD_ETHERNET_LLI

イーサネット

vmc_dlc_cfg.mux_info.adapter_number

DLC が使用するアダプター番号。

サーバーに、この DLC タイプの複数のアダプター・カードがある場合、最初のカードにはゼロ、2 番目のカードには 1 というように指定します。それ以外の場合、このパラメーターはゼロに設定してください。

vmc_dlc_cfg.lan_type

DLC がアクセスするネットワークのタイプ。値は次のいずれかです。

LLC_DIX

DIX

LLC2_802_3

802.3

LLC2_802_3_DIX

まだ決定されていません (802.3 または DIX)。Communications Server for Linux は、隣接ステーションがこれらのフォーマットのいずれかのフレームに最初に応答する際に、正しいタイプ (上記の 2 つの値のうちいずれか) を検出します。

LLC2_TOKEN_RING

トークンリング

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_DLC_NAME
指定した *dlc_name* パラメーターに、有効でない文字が入っていました。

AP_INVALID_DLC_TYPE
指定した *dlc_type* パラメーターが、許可された値の 1 つではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_DLC_ACTIVE
指定した DLC は現在活動状態であるため、変更できません。

AP_INVALID_DLC_TYPE
既存の DLC の DLC タイプ、折衝可能リンク・サポート、またはサポートされるポート・タイプを変更することはできません。これらは、新規 DLC の作成時にのみ指定できます。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_DLUR_DEFAULTS

DEFINE_DLUR_DEFAULTS は、デフォルトの従属 LU サーバー (DLUS) とバックアップ用のデフォルト DLUS を定義します。デフォルト DLUS またはバックアップ・デフォルト DLUS が既に定義されている場合、この verb は既存の定義を上書きします。デフォルト DLUS 名は、関連する DLUS が明示的に指定されていない PU に対して DLUR が SSCP-PU の活動化を開始するときに、DLUR により使用されます。(PU とそれに関連した DLUS を定義する場合、ローカル PU には DEFINE_INTERNAL_PU、ダウンストリーム PU には DEFINE_LS を使用してください。)

この verb は、デフォルト DLUS またはバックアップ・デフォルト DLUS を取り消して定義をなくすためにも使用できます。

VCB 構造体

```
typedef struct define_dlur_defaults
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;                /* reserved                  */
    AP_UINT16      primary_rc;            /* primary return code      */
    AP_UINT32      secondary_rc;         /* secondary return code    */
    unsigned char  description[32];      /* resource description     */
    unsigned char  reserv1[16];          /* reserved                  */
    unsigned char  dlus_name[17];        /* DLUS name                */
    unsigned char  bkup_dlus_name[17];   /* Backup DLUS name        */
    unsigned char  reserv3;               /* reserved                  */
    unsigned char  dlus_retry_timeout;   /* retry timeout            */
    unsigned char  dlus_retry_limit;     /* retry limit              */
    unsigned char  prefer_active_dlus;   /* retry using active DLUS */
    unsigned char  persistent_pipe_support; /* reserved                */
    unsigned char  reserv4[14];          /* reserved                  */
} DEFINE_DLUR_DEFAULTS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_DLUR_DEFAULTS

description

DLUR デフォルトを記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管されますが、Communications Server for Linux では他の用途に使用されません

dlus_name

デフォルトとして機能する DLUS ノードの名前。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

現在のデフォルト DLUS を取り消してデフォルト DLUS が定義されていない状態にするには、このパラメーターを 17 桁の 2 進ゼロに設定します。

bkup_dlus_name

バックアップ・デフォルトとして機能する DLUS ノードの名前。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

現在のバックアップ・デフォルト DLUS を取り消してデフォルト DLUS が定義されていない状態にするには、このパラメーターを 17 桁の 2 進ゼロに設定します。

dlus_retry_timeout

DLUS に接続するための再活動化タイマー。このパラメーターには、Communications Server for Linux が DLUS との接続に失敗した場合に再試行する時間間隔 (秒単位) を指定します。0x0001 ~ 0xFFFF の範囲で値を指定します。

dlus_retry_limit

DLUS に接続するための再試行カウント。このパラメーターを使用して、Communications Server for Linux が DLUS への最初の接続の試みに失敗したときに再試行する回数を指定します。

0x0001 ~ 0xFFFE の範囲で値を指定するか、Communications Server for Linux が DLUS に接続するまで無限に再試行することを示す 0xFFFF を指定します。

prefer_active_dlus

Communications Server for Linux が、DLUS から否定 RSP(REQACTPU) を受信した場合の動作、または失敗した DLUR PU の再活動化を試行する場合の動作を指定します。値は次のいずれかです。

AP_YES デフォルトのプライマリー DLUS またはデフォルトのバックアップ DLUS が活動状態である場合、Communications Server for Linux は活動状態である DLUS のみを使用して PU を活動化または再活動化しようとします。

AP_NO Communications Server for Linux は、標準再試行ロジックを使用して PU を活動化または再活動化しようとします。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

DEFINE_DLUR_DEFAULTS

secondary_rc

値は次のいずれかです。

AP_INVALID_DLUS_NAME

指定した *dlus_name* パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

AP_INVALID_BKUP_DLUS_NAME

指定した *dlus_name* パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードが DLUR をサポートしていません。これは、DEFINE_NODE verb の *dlur_support* パラメーターで定義されています。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_DOMAIN_CONFIG_FILE

DEFINE_DOMAIN_CONFIG_FILE は、ドメイン構成ファイルのヘッダーに組み込むコメント・ストリングを指定するか、既存のコメント・ストリングを変更します。

ノード構成ファイルには、この verb に相当する verb はありません。これは、ノード構成ファイルのヘッダーにはコメント・ストリングが含まれていないためです。ノード構成ファイルにコメント情報を組み込むには、DEFINE_NODE verb で説明のパラメーターを使用してください。

この verb は、ドメイン構成ファイルに対して発行しなければなりません。

VCB 構造体

```
typedef struct define_domain_config_file
{
    AP_UINT16          opcode;           /* verb operation code */
    unsigned char     reserv2;          /* reserved */
    unsigned char     format;           /* reserved */
    AP_UINT16         primary_rc;       /* primary return code */
    AP_UINT32         secondary_rc;     /* secondary return code */
    unsigned char     reserv3[8];       /* Reserved */
    CONFIG_FILE_HEADER hdr;             /* defined data */
} DEFINE_DOMAIN_CONFIG_FILE;
```



```
typedef struct config_file_header
{
    AP_UINT16      major_version;      /* reserved */
    AP_UINT16      minor_version;      /* reserved */
    AP_UINT16      update_release;     /* reserved */
    AP_UINT32      revision_level;     /* reserved */
    unsigned char  comment[100];       /* optional comment string */
    unsigned char  updating;           /* reserved */
} CONFIG_FILE_HEADER;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_DOMAIN_CONFIG_FILE

hdr.comment

ファイルに関する情報を格納するオプションのコメント・ストリング。これは 0 ~ 99 文字の ASCII ストリングで、最後にヌル文字が続きます。このパラメーターは情報専用です。Communications Server for Linux は QUERY_DOMAIN_CONFIG_FILE verb でこの情報を戻しますが、他の用途には使用されません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_DOWNSTREAM_LU

DEFINE_DOWNSTREAM_LU は新しいダウンストリーム LU を定義し、それをアップストリーム・ホスト LU または LU プール (DEFINE_LU_0_TO_3 または DEFINE_LU_POOL を使用して定義した) にマップします。これにより、ダウンストリーム LU は Communications Server for Linux の SNA ゲートウェイ機能を使用してホスト・コンピューターにアクセスできます。この verb を使用して既存のダウンストリーム LU を変更することはできません。

この verb は既に定義されているダウンストリーム LU を活動化する場合にも使用できます (たとえば、ダウンストリーム・ワークステーションを活動化した直後など)。これを行うには、その LU に対して DEFINE_DOWNSTREAM_LU verb を再発行します。定義を変更することはできないため、すべてのパラメーターがオリジナルの定義と同じでなければならないことに注意してください。

DEFINE_DOWNSTREAM_LU は、Communications Server for Linux プライマリー RUI アプリケーションと通信するアプリケーションによって使用されるダウンスト

DEFINE_DOWNSTREAM_LU

リーム LU を定義する場合にも使用できます。プライマリー RUI について詳しくは、「Communications Server for Linux LUA Programmer's Guide」を参照してください。

VCB 構造体

```
typedef struct define_downstream_lu
{
    AP_UINT16          opcode;          /* verb operation code */
    unsigned char     reserv2;         /* reserved */
    unsigned char     format;         /* reserved */
    AP_UINT16          primary_rc;     /* primary return code */
    AP_UINT32          secondary_rc;   /* secondary return code */
    unsigned char     dslu_name[8];   /* Downstream LU name */
    DOWNSTREAM_LU_DEF_DATA def_data;  /* Defined data */
} DEFINE_DOWNSTREAM_LU;

typedef struct downstream_lu_def_data
{
    unsigned char     description[32]; /* resource description */
    unsigned char     reserv1[16];    /* reserved */
    unsigned char     nau_address;     /* downstream LU nau address */
    unsigned char     dspu_name[8];   /* Downstream PU name */
    unsigned char     host_lu_name[8]; /* Host LU or Pool name */
    unsigned char     allow_timeout;  /* Allow timeout of host LU? */
    unsigned char     delayed_logon;  /* Allow delayed logon to */
    /* host LU */
    unsigned char     reserv2[6];     /* reserved */
} DOWNSTREAM_LU_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_DOWNSTREAM_LU

dslu_name

定義するダウンストリーム LU の名前。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

def_data.description

ダウンストリーム LU を記述したヌルで終了するテキスト・String (0 ~ 31 文字のあとにヌル文字を続けたString)。このStringは情報専用です。このStringはノードの構成ファイル内に保管され、QUERY_DOWNSTREAM_LU verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

def_data.nau_address

ダウンストリーム LU のネットワーク・アクセス可能単位アドレス。これは、1 ~ 255 の範囲でなければなりません。

def_data.dspu_name

この LU に関連したダウンストリーム PU の名前 (DEFINE_LS で指定した名前)。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC Stringで、右側に EBCDIC のスペースを入れます。

def_data.host_lu_name

このダウンストリーム LU のマップ先となるホスト LU またはホスト LU プールの名前。これは 8 バイトからなるタイプ A の EBCDIC Stringで、右側に EBCDIC のスペースを入れます。

SNA ゲートウェイの場合、ホスト LU を従属 LU タイプ 6.2 にすることはできません。ただし、ダウンストリーム LU が LU タイプ 6.2 である場合は、ホスト LU を LU タイプ 0-3 として構成し、ホスト LU のモデル・タイプを不明として指定できます。

ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は、このフィールドを EBCDIC でストリング #PRIRUI# と設定します。

def_data.allow_timeout

ダウンストリーム LU とアップストリーム LU 間のセッションが、アップストリーム LU の定義上で指定されたタイムアウト期間に、活動状態でないまま放置された場合、そのセッションをタイムアウトにするかどうかを指定します。値は次のいずれかです。

AP_YES このダウンストリーム LU がアップストリーム LU との間に設定したセッションをタイムアウトにします。

AP_NO このダウンストリーム LU がアップストリーム LU との間に設定したセッションをタイムアウトにしません。

このフィールドは、ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は無視されます。

def_data.delayed_logon

このダウンストリーム LU で遅延ログオンを使用するかどうかを指定します (アップストリーム LU はユーザーが要求するまで活動化しません)。値は次のいずれかです。

AP_YES このダウンストリーム LU に遅延ログオンを使用し、アップストリーム LU はユーザーが要求するまで活動化しません。

AP_NO このダウンストリーム LU には遅延ログオンを使用しません。

このフィールドは、ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は無視されます。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

DEFINE_DOWNSTREAM_LU

AP_INVALID_DNST_LU_NAME

指定した *dslu_name* パラメーターに、有効でない文字が入っていました。

AP_INVALID_NAU_ADDRESS

指定した NAU アドレスが、有効な範囲内にありませんでした。

AP_INVALID_ALLOW_TIMEOUT

指定した *allow_timeout* パラメーターの値が有効ではありませんでした。

AP_INVALID_DELAYED_LOGON

指定した *delayed_logon* パラメーターの値が有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PU_NAME

指定した *dspu_name* パラメーターが、有効ではありませんでした。

AP_PU_NOT_DEFINED

指定した *dspu_name* パラメーターが、定義されているどの PU 名にも一致しませんでした。

AP_INVALID_PU_TYPE

dspu_name パラメーターで指定した PU は、SNA ゲートウェイをサポートしたダウンストリーム PU ではありません。

AP_LU_ALREADY_DEFINED

指定した名前の LU は、既に定義されており、この verb を使用して変更することはできません。

AP_DSLU_ACTIVE

LU は既に活動状態になっています。

AP_LU_NAU_ADDR_ALREADY_DEFD

指定した NAU アドレスの LU は、既に定義されています。

AP_INVALID_HOST_LU_NAME

指定したホスト LU 名が有効ではありませんでした。

AP_LU_NAME_POOL_NAME_CLASH

指定した LU 名は、既存の LU プールの名前と矛盾しています。

AP_PU_NOT_ACTIVE

dspu_name パラメーターで指定した PU は、現在活動状態ではありません。

AP_LU_ALREADY_ACTIVATING

dslu_name パラメーターに指定した名前の LU は、現在活動化中です。

AP_LU_DEACTIVATING

dslu_name パラメーターに指定した名前の LU は、非活動化処理中です。

AP_LU_ALREADY_ACTIVE

dslu_name パラメーターに指定した名前の LU は、既に活動状態になっています。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻しません。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは SNA ゲートウェイをサポートしていません。これは、DEFINE_NODE verb の *pu_conc_support* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_DOWNSTREAM_LU_RANGE

DEFINE_DOWNSTREAM_LU_RANGE は新しいダウンストリーム LU の範囲を定義し、それらの LU をアップストリーム・ホスト LU または LU プール (DEFINE_LU_0_TO_3 または DEFINE_LU_POOL を使用して定義した) にマップします。これにより、ダウンストリーム LU は Communications Server for Linux の SNA ゲートウェイ機能を使用してホスト・コンピューターにアクセスできます。この verb を使用して既存のダウンストリーム LU を変更することはできません。

この verb に指定するパラメーターには、新しい LU のベース名と NAU アドレスの範囲が含まれます。新しい LU の名前は、そのベース名と NAU アドレスを結合して生成されます。たとえば、LUNME というベース名が 11 ~ 14 の NAU 範囲と結合して、LUNME011、LUNME012、LUNME013、LUNME014 という LU が定義されます。

DEFINE_DOWNSTREAM_LU_RANGE

DEFINE_DOWNSTREAM_LU_RANGE は、Communications Server for Linux プライマリー RUI アプリケーションと通信するアプリケーションによって使用されるダウンストリーム LU を定義する場合にも使用できます。プライマリー RUI については、「Communications Server for Linux LUA Programmer's Guide」を参照してください。

VCB 構造体

```
typedef struct define_downstream_lu_range
{
    AP_UINT16      opcode;          /* verb operation code          */
    unsigned char  reserv2;        /* reserved                     */
    unsigned char  format;        /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  dslu_base_name[5]; /* Downstream LU base name    */
    unsigned char  reserv3;        /* reserved                     */
    unsigned char  description[32]; /* resource description        */
    unsigned char  reserv1[16];    /* reserved                     */
    unsigned char  min_nau;        /* Minimum NAU address in range */
    unsigned char  max_nau;        /* Maximum NAU address in range */
    unsigned char  dspu_name[8];   /* Downstream PU name          */
    unsigned char  host_lu_name[8]; /* Host LU or Pool name        */
    unsigned char  allow_timeout;  /* Allow timeout of host LU?    */
    unsigned char  delayed_logon;  /* Allow delayed logon to host LU */
    unsigned char  reserv4[6];     /* reserved                     */
} DEFINE_DOWNSTREAM_LU_RANGE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_DOWNSTREAM_LU_RANGE

dslu_base_name

新しい LU の名前のベース名。これは 5 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、ベース名が 5 文字に満たない場合、右側に EBCDIC のスペースを入れます。Communications Server for Linux は、この名前に 3 桁の 10 進値からなる NAU アドレスを付加し、それぞれの LU の LU 名を生成します。

description

ダウンストリーム LU を記述したヌルで終了するテキスト・String (0 ~ 31 文字のあとにヌル文字を続けたString。範囲内の各 LU に同じStringが使用されます)。このStringは情報専用です。このStringはノードの構成ファイル内に保管され、QUERY_DOWNSTREAM_LU verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

min_nau

最初の LU の NAU アドレス (範囲は 1 ~ 255)。

max_nau

最後の LU の NAU アドレス (範囲は 1 ~ 255)。

dspu_name

この範囲のダウンストリーム LU が使用するダウンストリーム PU の名前

(DEFINE_LS verb で指定した名前)。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、必要に応じて右側に EBCDIC のスペースを入れます。

host_lu_name

指定した範囲のダウンストリーム LU のマップ先となるホスト LU またはホスト LU プールの名前。これは 8 バイトからなるタイプ A の EBCDIC スtringで、必要に応じて右側に EBCDIC スペースを入れます。

ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は、このフィールドを EBCDIC でスtring #PRIRUI# と設定します。

allow_timeout

この範囲のダウンストリーム LU とアップストリーム LU の間のセッションが、アップストリーム LU の定義上で指定されたタイムアウト期間に活動状態でないまま放置された場合、そのセッションをタイムアウトにするかどうかを指定します。値は次のいずれかです。

AP_YES この範囲のダウンストリーム LU とアップストリーム LU の間のセッションをタイムアウトにします。

AP_NO この範囲のダウンストリーム LU とアップストリーム LU の間のセッションをタイムアウトにしません。

このフィールドは、ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は無視されます。

delayed_logon

この範囲のダウンストリーム LU に遅延ログオンを使用するかどうかを指定します (アップストリーム LU は、ユーザーが要求するまで活動化しません)。値は次のいずれかです。

AP_YES この範囲のダウンストリーム LU に遅延ログオンを使用し、アップストリーム LU はユーザーが要求するまで活動化しません。

AP_NO この範囲のダウンストリーム LU には遅延ログオンを使用しません。

このフィールドは、ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

DEFINE_DOWNSTREAM_LU_RANGE

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_DNST_LU_NAME

指定した *dslu_base_name* パラメーターに、有効でない文字が入っていました。

AP_INVALID_NAU_ADDRESS

指定した NAU アドレスの 1 つ以上が、有効な範囲内にありませんでした。

AP_INVALID_ALLOW_TIMEOUT

指定した *allow_timeout* パラメーターの値が有効ではありませんでした。

AP_INVALID_DELAYED_LOGON

指定した *delayed_logon* パラメーターの値が有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PU_NAME

指定した *dspu_name* パラメーターが、有効ではありませんでした。

AP_PU_NOT_DEFINED

指定した *dspu_name* パラメーターが、定義されているどの PU 名にも一致しませんでした。

AP_INVALID_PU_TYPE

dspu_name パラメーターで指定した PU は、SNA ゲートウェイをサポートしたダウンストリーム PU ではありません。

AP_LU_ALREADY_DEFINED

この範囲内の名前の 1 つに一致する名前が付いた LU が、既に定義されています。この verb を使用して既存の LU を変更することはできません。

AP_DSLU_ACTIVE

この範囲内の名前の 1 つに一致する名前が付いた LU が、既に活動状態です。この verb を使用して既存の LU を変更することはできません。

AP_LU_NAU_ADDR_ALREADY_DEFD

この範囲内のアドレスの 1 つに一致する NAU アドレスを備えた LU が、既に定義されています。

AP_INVALID_HOST_LU_NAME

指定したホスト LU 名が有効ではありませんでした。

AP_LU_NAME_POOL_NAME_CLASH

この範囲内の LU 名の 1 つが、既存の LU プールの名前と矛盾しています。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻しません。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは SNA ゲートウェイをサポートしていません。これは、DEFINE_NODE verb の *pu_conc_support* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_DSPU_TEMPLATE

DEFINE_DSPU_TEMPLATE verb は、Linux SNA ゲートウェイ機能を使用するダウンストリーム LU のテンプレートを定義します。このテンプレートは、ワークステーションが暗黙リンク (事前に定義されていないリンク) を介して接続する場合に、ダウンストリーム・ワークステーションのグループ上にダウンストリーム LU を定義するために使用されます。

DEFINE_DSPU_TEMPLATE は、Communications Server for Linux ノード上のプライマリー RUI アプリケーションと通信するアプリケーションをサポートするダウンストリーム LU を定義する場合にも使用できます。プライマリー RUI について詳しくは、「Communications Server for Linux LUA Programmer's Guide」を参照してください。

VCB 構造体

```
typedef struct define_dspu_template
{
    AP_UINT16          opcode;          /* verb operation code */
    unsigned char     reserv3;         /* reserved */
    unsigned char     format;         /* reserved */
    AP_UINT16         primary_rc;     /* primary return code */
    AP_UINT32         secondary_rc;   /* secondary return code */
}
```

DEFINE_DSPU_TEMPLATE

```
    unsigned char    template_name[8]; /* Name of template */
    unsigned char    description[32]; /* resource description */
    unsigned char    reserv2[16]; /* reserved */
    unsigned char    modify_template; /* Modify existing template? */
    unsigned char    reserv1[11]; /* reserved */
    AP_UINT16        max_instance; /* Max active template */
                    /* instances */
    AP_UINT16        num_of_dslu_templates; /* number of DSLU templates*/
} DEFINE_DSPU_TEMPLATE;

typedef struct dslu_template
{
    unsigned char    min_nau; /* Minimum NAU address in range*/
    unsigned char    max_nau; /* Maximum NAU address in range*/
    unsigned char    allow_timeout; /* Allow timeout of host LU? */
    unsigned char    delayed_logon; /* Allow delayed logon to host */
                    /* LU */
    unsigned char    reserv1[8]; /* reserved */
    unsigned char    host_lu[8]; /* Host LU or Pool name */
} DSLU_TEMPLATE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_DSPU_TEMPLATE

template_name

ダウンストリーム・ワークステーションのグループ上に存在するダウンストリーム LU のテンプレート名。

description

QUERY_DSPU_TEMPLATE verb が戻す、リソースの記述。

modify_template

この verb で既存の DSPU テンプレートに追加 DSLU テンプレートを追加するのか、既存の DSPU テンプレートを置換するのかを指定します。値は次のいずれかです。

AP_MODIFY_DSPU_TEMPLATE

指定した DSPU テンプレートがなければ、そのテンプレートを作成します。指定した DSPU テンプレートが存在する場合、付加した DSLU テンプレートが既存の DSPU テンプレートに追加されます。

AP_REPLACE_DSPU_TEMPLATE

既存の定義を上書きして、新しいテンプレートを作成します。

max_instance

同時に活動状態にできるテンプレートのインスタンスの最大数。この限度に達した場合、新しいインスタンスは作成されません。0 ~ 65,535 の範囲で値を指定します。0 (ゼロ) は制限がないことを意味します。

num_of_dslu_templates

この verb によって定義されるダウンストリーム LU (DSL)U テンプレートの数。

サブレコード *dslu_template* は、次のパラメーターを含んでいます。

min_nau

最初のダウンストリーム PU の NAU アドレス (範囲は 1 ~ 255)。

max_nau

最後のダウンストリーム PU の NAU アドレス (範囲は 1 ~ 255)。

allow_timeout

ホスト LU 定義で指定されたタイムアウト期間に、セッションが活動状態でないまま放置された場合、ダウンストリーム LU が使用しているホスト LU をタイムアウトにするかどうかを指定します。値は次のいずれかです。

AP_YES Communications Server for Linux は、このダウンストリーム LU が使用しているホスト LU をタイムアウトにすることができます。

AP_NO Communications Server for Linux は、このダウンストリーム LU が使用しているホスト LU をタイムアウトにすることができません。

このフィールドは、ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は無視されます。

delayed_logon

ダウンストリーム LU とホスト LU との接続を、ダウンストリーム LU から最初のデータを受信するまで遅らせるかどうかを指定します。値は次のいずれかです。

AP_YES Communications Server for Linux はダウンストリーム LU のホスト LU への接続を遅らせます。シミュレートされたログオン画面がダウンストリーム LU へ送信されます。

AP_NO Communications Server for Linux はダウンストリーム LU のホスト LU への接続を遅らせません。

このフィールドは、ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は無視されます。

host_lu ダウンストリーム LU が使用するホスト LU またはホスト LU プールの名前。この名前は 8 バイトからなるタイプ A の文字ストリングです。

ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は、このフィールドを EBCDIC でストリング #PRIRUI# と設定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

DEFINE_DSPU_TEMPLATE

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_TEMPLATE_NAME

template_name パラメーターに指定した名前が有効ではありませんでした。

AP_INVALID_NAU_ADDRESS

min_nau パラメーターまたは *max_nau* パラメーターが、有効ではありませんでした。

AP_INVALID_NAU_RANGE

min_nau パラメーターまたは *max_nau* パラメーターで指定したアドレスが有効な範囲内にありませんでした。

AP_CLASHING_NAU_RANGE

dslu_template サブレコード内の *min_nau* パラメーターと *max_nau* パラメーターで指定したアドレス範囲が、*template_name* パラメーターによって指定したテンプレート内の別の *dslu_template* サブレコードが指定する範囲と矛盾します。

AP_INVALID_NUM_DSPU_TEMPLATES

num_of_dslu_templates パラメーターに指定した値が、有効な範囲内にありませんでした。

AP_INVALID_ALLOW_TIMEOUT

allow_timeout パラメーターに指定した値が、有効ではありませんでした。

AP_INVALID_DELAYED_LOGON

delayed_logon パラメーターに指定した値が、有効ではありませんでした。

AP_INVALID_MODIFY_TEMPLATE

modify_template パラメーターに指定した値が、有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_HOST_LU_NAME

指定した *host_lu_name* パラメーター値が有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは SNA ゲートウェイをサポートしていません。これは、DEFINE_NODE verb の *pu_conc_support* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_FOCAL_POINT

DEFINE_FOCAL_POINT verb は、特定の Management Services カテゴリのフォーカル・ポイントを指定します。新しいフォーカル・ポイントを指定した場合、Communications Server for Linux は MS_CAPABILITIES 要求を送信することにより、指定されたフォーカル・ポイントと暗黙の 1 次フォーカル・ポイントの関係を確立しようとします。

VCB 構造体

```
typedef struct define_focal_point
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code         */
    AP_UINT32      secondary_rc;    /* secondary return code       */
    unsigned char  reserved;       /* reserved                     */
    unsigned char  ms_category[8]; /* management services category */
    unsigned char  fp_fqcp_name[17]; /* Fully qualified focal
                                     /* point cp name                */
    unsigned char  ms_appl_name[8]; /* Focal point application name */
    unsigned char  description[32]; /* resource description         */
    unsigned char  reserv1[16];    /* reserved                     */
    unsigned char  backup;         /* is focal point a backup     */
    unsigned char  reserv3[16];    /* reserved                     */
} DEFINE_FOCAL_POINT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_FOCAL_POINT

ms_category

Management Services カテゴリ。これは、「Systems Network Architecture: Management Services Reference」(xxiii ページの『関連資料』を参照)の

DEFINE_FOCAL_POINT

『MS Discipline-Specific Application Programs』の表に指定されたカテゴリ一名の 1 つ (EBCDIC スペース (0x40) で埋め込まれている) か、またはユーザー定義のカテゴリです。ユーザー定義カテゴリ名は、8 バイトからなるタイプ 1134 の EBCDIC スtringで、必要に応じて EBCDIC のスペース (0x40) を入れます。

fp_fqcp_name

フォーカル・ポイントの完全修飾制御点名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC Stringです。この名前は、最大 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A String文字からなるネットワーク名で構成されます。

ms_appl_name

フォーカル・ポイント・アプリケーション名。これは、通常、タイプ 1134 の文字を使用した EBCDIC Stringです。または、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照) で指定された『MS Discipline-Specific Application Program』の 1 つである場合もあります。このStringは、長さが 8 文字でなければならず、必要に応じて EBCDIC のスペース文字 (0x40) を右側に埋め込みます。

description

フォーカル・ポイントを記述したヌルで終了するテキスト・String (0 ~ 31 文字のあとにヌル文字を続けたString)。このStringは情報専用です。このStringはノードの構成ファイル内に保管され、`QUERY_FOCAL_POINT verb` で戻されますが、`Communications Server for Linux` では他の用途に使用されません。

backup 指定したアプリケーションが、このカテゴリのメイン・フォーカル・ポイントであるか、バックアップ・フォーカル・ポイントであるかを示します。値は次のいずれかです。

AP_YES バックアップ・フォーカル・ポイント (メイン・フォーカル・ポイントが使用できない場合にのみ使用されます)。

AP_NO メイン・フォーカル・ポイント。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、`Communications Server for Linux` は、以下のパラメーターを戻します。

primary_rc

AP_OK フォーカル・ポイントは要求どおりに定義されました。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、`Communications Server for Linux` は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_CATEGORY_NAME

指定したカテゴリ名に、有効でない文字が入っていました。

AP_INVALID_FP_NAME

完全修飾名またはアプリケーション名が有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは MS ネットワーク管理機能をサポートしていません。これは、DEFINE_NODE verb の *mds_supported* パラメーターによって定義されます。

戻りパラメーター: 置き換え時

この verb の直後に別のフォーカル・ポイントを指定する別の verb が続いていたために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_REPLACED

この verb が未処理の間に、同じノードに対して再度 DEFINE_FOCAL_POINT が発行され、同じ MS カテゴリに異なるフォーカル・ポイントが指定されました。この verb は中止されました。ノードは 2 回目の verb で指定されたフォーカル・ポイントに接続を試みます。

戻りパラメーター: 失敗時

フォーカル・ポイントの関係を確立できないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_UNSUCCESSFUL

secondary_rc

値は次のいずれかです。

AP_IMPLICIT_REQUEST_REJECTED

指定したフォーカル・ポイントが要求を拒否しました。

AP_IMPLICIT_REQUEST_FAILED

ノードが、指定したフォーカル・ポイントへ要求を送信できません

でした。原因としては、指定した制御点またはアプリケーションが見つからなかった可能性があります。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_INTERNAL_PU

DEFINE_INTERNAL_PU verb は、ローカル・ノード上に DLUR からサービスを受ける PU を定義します。(DLUR または SNA ゲートウェイからサービスを受けるダウンストリーム PU を定義する場合、またはホストへ直接接続するローカル PU を定義する場合は、DEFINE_INTERNAL_PU でなく DEFINE_LS を使用してください。)

VCB 構造体

```
typedef struct define_internal_pu
{
    AP_UINT16          opcode;           /* verb operation code */
    unsigned char     reserv2;          /* reserved */
    unsigned char     format;           /* reserved */
    AP_UINT16         primary_rc;       /* primary return code */
    AP_UINT32         secondary_rc;     /* secondary return code */
    unsigned char     pu_name[8];       /* internal PU name */
    INTERNAL_PU_DEF_DATA def_data;      /* defined data */
} DEFINE_INTERNAL_PU;

typedef struct internal_pu_def_data
{
    unsigned char     description[32];   /* resource description */
    unsigned char     initially_active;  /* is PU initially active? */
    unsigned char     reserv1[15];      /* reserved */
    unsigned char     dlus_name[17];    /* DLUS name */
    unsigned char     bkup_dlus_name[17]; /* backup DLUS name */
    unsigned char     pu_id[4];         /* PU identifier */
    AP_UINT16         dlus_retry_timeout; /* DLUS retry timeout */
    AP_UINT16         dlus_retry_limit; /* DLUS retry limit */
    unsigned char     conventional_lu_compression; /* compression for LU 0-3? */
    unsigned char     conventional_lu_cryptography; /* reserved */
    unsigned char     pu_can_send_dddlu_offline; /* does the PU send NMVT */
                                                         /* (power off) to host? */
    unsigned char     reserv2[1];       /* reserved */
} INTERNAL_PU_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_INTERNAL_PU

pu_name

定義する内部 PU の名前。これは (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

この名前は、ホストで構成されている PU 名と一致している必要があります (Communications Server for Linux は PU を識別するために PU 名およ

び PU ID の両方をホストへ送信します。ホストは、通常、PU 名によって PU を識別し、一致する PU 名が見つからない場合は、PU ID によって PU を識別します)。

def_data.description

内部 PU を記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、QUERY_DLUR_PU および QUERY_PU verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

def_data.initially_active

この内部 PU をノードの起動時に自動的に始動するかどうかを指定します。値は次のいずれかです。

AP_YES PU はノードの始動時に自動的に起動します。

AP_NO PU は自動的に起動しません。手動で起動させる必要があります。

def_data.dlus_name

DLUR が SSCP-PU の活動化を開始するときに使用する DLUS ノードの名前。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

DLUR がグローバル・デフォルト DLUS を使用することを示すには、このパラメーターの 17 桁の 2 進ゼロに設定します。その場合は、DEFINE_DLUR_DEFAULTS を使用してグローバル・デフォルト DLUS も定義しなければなりません。

def_data.bkup_dlus_name

この PU のバックアップ DLUS として機能する DLUS ノードの名前。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

DLUR がグローバル・バックアップ・デフォルト DLUS を使用することを示すには、このパラメーターの 17 桁の 2 進ゼロに設定します。その場合は、DEFINE_DLUR_DEFAULTS を使用してグローバル・バックアップ・デフォルト DLUS も定義しなければなりません。

def_data.pu_id

PU ID。これは 4 バイトの 16 進数ストリングで、ブロック番号 (3 桁の 16 進数字) とノード番号 (5 桁の 16 進数字) で構成されます。この PU ID は、ホスト側で構成された *pu_id* に一致する必要があります (Communications Server for Linux は PU を識別するために PU 名および PU ID の両方をホストへ送信します。ホストは、通常、PU 名によって PU を識別し、一致する PU 名が見つからない場合は、PU ID によって PU を識別します)。

def_data.dlus_retry_timeout

DLUS に接続するための再活動化タイマー。このパラメーターには、Communications Server for Linux が DLUS との接続に失敗した場合に再試行する時間間隔 (秒単位) を指定します。1 回目の試行と 2 回目の試行の間隔は常に 1 秒です。

0x0001 ~ 0xFFFF の範囲で値を指定します。ゼロを指定した場合は、DEFINE_DLUR_DEFAULTS verb で指定したデフォルト値が使用されます。

def_data.dlus_retry_limit

DLUS に接続するための再試行カウント。このパラメーターを使用して、Communications Server for Linux が DLUS への最初の接続の試みに失敗したときに再試行する回数を指定します。

0x0001 ~ 0xFFFF の範囲で値を指定するか、Communications Server for Linux が DLUS に接続するまで無限に再試行することを示す 0xFFFF を指定します。

def_data.conventional_lu_compression

この PU を使用して、LU 0 から 3 のセッションにデータ圧縮を要求するかどうかを指定します。

値は次のいずれかです。

AP_YES ホストが要求した場合、データ圧縮は、この PU を使用して LU 0 から 3 のセッションに対して使用する必要があります。

AP_NO データ圧縮は、この PU を使用して LU 0 から 3 のセッションに対して使用しないでください。

def_data.pu_can_send_dddlu_offline

ローカル PU がホストに NMVT (電源オフ) メッセージを送信するかどうかを指定します。ホスト・システムが DDDLU (従属 LU の動的定義) をサポートしている場合、Communications Server for Linux は、動的に定義された LU を使用し終わったときに、NMVT (電源オフ) をホストへ送信します。これによって、ホストは、定義が不要になった際にその定義を除去することによって、リソースを節約することができます。

値は次のいずれかです。

AP_YES ローカル PU は、NMVT (電源オフ) メッセージをホストに送信します。

AP_NO ローカル PU は、NMVT (電源オフ) メッセージをホストに送信しません。

ホストが DDDLU はサポートしていても NMVT (電源オフ) メッセージをサポートしていない場合、このパラメーターは AP_NO に設定しなければなりません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_PARAMETER_CHECK`

secondary_rc

値は次のいずれかです。

AP_INVALID_PU_NAME

pu_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_PU_ID

pu_id パラメーターに、有効でない文字が入っていました。

AP_INVALID_DLUS_NAME

dlus_name パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

AP_INVALID_BKUP_DLUS_NAME

bkup_dlus_name パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_STATE_CHECK`

secondary_rc

AP_PU_ALREADY_DEFINED

指定した名前の PU は、既に定義されています。

戻りパラメーター: サポートされていない機能

ノードの構成がこの `verb` をサポートしていないために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ノードは DLUR をサポートしていません。これは、`DEFINE_NODE` `verb` の *dlur_support* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_LOCAL_LU

DEFINE_LOCAL_LU verb は、新しいローカル LU を定義します。この verb は、接続経路指定データ、*disable* パラメーター、既存 LU (またはローカル・ノードの制御点に関連したデフォルト LU) の説明を変更するために使用することもできますが、それ以外のパラメーターを変更することはできません。既存の LU を変更する場合、他のパラメーターは現在定義されている値に設定しなければなりません。

VCB 構造体

```
typedef struct define_local_lu
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;          /* reserved                      */
    unsigned char  format;           /* reserved                      */
    AP_UINT16      primary_rc;       /* primary return code          */
    AP_UINT32      secondary_rc;     /* secondary return code        */
    unsigned char  lu_name[8];       /* local LU name                 */
    LOCAL_LU_DEF_DATA def_data;     /* defined data                  */
} DEFINE_LOCAL_LU;

typedef struct local_lu_def_data
{
    unsigned char  description[32];  /* resource description          */
    unsigned char  reserv1;          /* reserved                      */
    unsigned char  security_list_name[14]; /* security access list name    */
    unsigned char  reserv3;          /* reserved                      */
    unsigned char  lu_alias[8];      /* local LU alias                */
    unsigned char  nau_address;      /* NAU address                   */
    unsigned char  syncpt_support;   /* is Syncpoint supported?      */
    AP_UINT16      lu_session_limit; /* LU session limit             */
    unsigned char  default_pool;     /* is LU in the pool of default */
                                     /* LUs?                          */
    unsigned char  reserv2;          /* reserved                      */
    unsigned char  pu_name[8];       /* PU name                       */
    unsigned char  lu_attributes;    /* LU attributes                 */
    unsigned char  sscp_id[6];       /* SSCP ID                       */
    unsigned char  disable;          /* disable or enable local LU   */
    ROUTING_DATA  attach_routing_data; /* routing data for incoming    */
                                     /* attaches                       */
    unsigned char  reserv6;          /* reserved                      */
    unsigned char  reserv4[7];       /* reserved                      */
    unsigned char  reserv5[16];      /* reserved                      */
} LOCAL_LU_DEF_DATA;

typedef struct routing_data
{
    unsigned char  sys_name[128];     /* Name of target system for TP  */
    AP_INT32      timeout;           /* timeout value in seconds      */
    unsigned char  back_level;       /* reserved                      */
    unsigned char  reserved[59];     /* reserved                      */
} ROUTING_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_LOCAL_LU

lu_name

ローカル LU の名前。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

ローカル・ノードの制御点に関連したデフォルト LU の接続経路指定データまたは記述を変更するには、このパラメーターを 8 桁の 2 進ゼロに設定します。

def_data.description

ローカル LU を記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、`QUERY_LOCAL_LU verb` で戻されますが、`Communications Server for Linux` では他の用途に使用されません。

def_data.security_list_name

このローカル LU が使用するセキュリティー・アクセス・リスト (`DEFINE_SECURITY_ACCESS_LIST verb` を使用して定義されたもの) の名前。このパラメーターは、指定リストに指定されているユーザーのみが LU を使用できるように制限します。すべてのユーザーが LU を使用できるように指定するには、このパラメーターを 14 桁の 2 進ゼロに設定します。

def_data.lu_alias

ローカル LU の別名。これは 8 バイトの ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、必要に応じて 8 バイトになるように右側にスペースを入れます。

def_data.nau_address

LU のネットワーク・アクセス可能単位アドレス。LU が独立 LU であればゼロを指定し、従属 LU であれば 1 ~ 255 の範囲でアドレスを指定します。

def_data.syncpt_support

LU が同期点機能をサポートするかどうかを指定します。これは標準 `Communications Server for Linux` 製品のほかに同期点マネージャー (SPM) と会話保護リソース・マネージャー (C-PRM) が組み込まれている場合にのみ `AP_YES` に設定してください。値は次のいずれかです。

AP_YES 同期点はサポートされます。

AP_NO 同期点はサポートされません。

def_data.lu_session_limit

LU がサポートするセッションの (すべてのモードでの) 最大合計数。

従属 LU の場合、これは 1 に設定しなければなりません。独立 LU の場合は、無制限を示すゼロを指定するか、1 ~ 65,535 の範囲で値を指定します。明示的に限度を指定する場合は、次の点に注意してください。

- LU が並列セッション・リモート LU と通信する場合、セッション限度には CNOS 折衝に十分なセッション数が含まれていなければなりません。安全な最小値は 3 で、パートナー LU ごとに 2 つのセッションを追加します。
- LU のセッション限度は、その LU が使用するすべてのモードのセッション限度を合計した値に等しいか、それ以上でなければなりません。
- LU が全二重 APPC 会話で使用される場合、全二重会話ごとに 2 つのセッションが必要となります。

DEFINE_LOCAL_LU

def_data.default_pool

LU がデフォルトの従属 LU のプールにあるかどうかを指定します。詳しくは、134 ページの『デフォルト LU』を参照してください。値は次のいずれかです。

AP_YES LU はデフォルト LU のプールに入り、LU 名を指定しないアプリケーションから使用できます。

AP_NO LU はプールに入りません。

LU が独立 LU の場合、このパラメーターは予約済みです。

def_data.pu_name

この LU が使用する PU の、DEFINE_LS verb で指定した名前。このフィールドは従属 LU のみが使用し、独立 LU の場合は 8 桁の 2 進ゼロに設定します。名前は 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtring で、必要に応じて右側に EBCDIC のスペースを入れます。

def_data.lu_attributes

LU に関する追加情報を識別します。値は次のいずれかです。

AP_NONE

追加情報は識別されません。

AP_DISABLE_PWSUB

ローカル LU のパスワード置換サポートを使用不可にします。パスワード置換とは、パスワードを平文で送信せずに、ローカル LU とリモート LU の間での伝送を行う前にパスワードを暗号化することを言います。通常、Communications Server for Linux は、リモート・システムがパスワード置換をサポートしている場合、パスワード置換を使用します。

この値は、パスワード置換を正しくインプリメントしていないリモート・システムと通信するための予備手段として提供されています。このオプションを使用する場合、平文でのパスワードの送受信 (セキュリティ上のリスクがある場合があります) が行われることに注意しなければなりません。このオプションは、リモート・システムにおけるパスワード置換のインプリメンテーションに関して問題がある場合以外は設定しないでください。

def_data.sscp_id

この LU の活動化が許可される SSCP の ID を指定します。この ID は 6 バイトの 2 進 String です。このパラメーターは従属 LU のみが使用し、LU が独立 LU の場合、または LU がすべての SSCP から活動化できる場合には、すべて 2 進ゼロに設定します。

def_data.disable

ローカル LU を使用不可にするか、使用可能にするかを指定します。値は次のいずれかです。

AP_YES ローカル LU を使用不可にします。

AP_NO ローカル LU を使用可能にします。

def_data.attach_routing_data.sys_name

このローカル LU に到着する着信割り振り要求 (APPC または CPI-C 会話を開始するためのパートナー TP からの要求) に対応するターゲット・コンピューターのシステム名。

ターゲット TP がブロードキャスト待機 TP である場合 (つまり、ターゲット TP への着信割り振り要求を経路指定できるように、サーバーに開始時点の位置が通知される)、またはこの LU を所有するノードと同じ

Communications Server for Linux サーバー上で実行されている場合は、このパラメーターを 2 進ゼロに設定します。それ以外の場合は、このパラメーターを TP が実行されているコンピューターの名前に設定してください。

この名前は別名か、または完全修飾名のいずれかにする必要があります。IP アドレスは指定できません。システム名に . (ピリオド) 文字が含まれている場合、Communications Server for Linux はこれが完全修飾名であると想定します。ピリオドが含まれていない場合は、DNS ルックアップを実行してシステム名を判別します。

def_data.attach_routing_data.timeout

ダイナミック・ロード要求のタイムアウト値。呼び出された TP がこの時間内に Receive_Allocate verb (APPC)、Accept_Conversation または Accept_Incoming (CPI-C) を発行しなかった場合、要求はタイムアウトになります。秒単位でタイムアウト値を指定するか、タイムアウトがない (ダイナミック・ロード要求を無限に待つ) ことを示す -1 を指定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_DISABLE
disable パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LU_NAME
指定した LU 名に、有効でない文字が入っていました。

AP_INVALID_NAU_ADDRESS
指定した NAU アドレスが、有効な範囲内にありませんでした。

AP_INVALID_SESSION_LIMIT
指定したセッション限度が、有効な範囲内にありませんでした。

AP_INVALID_TIMEOUT

指定したタイムアウト値が、有効な範囲内にありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

lu_name パラメーターまたは *lu_alias* パラメーターに、有効でない文字が入っていました。

AP_LU_ALREADY_DEFINED

この名前が付いた LU は、既に定義されています。接続経路指定データを除き、この verb を使用して既存の LU のどのパラメーターも変更することはできません。

AP_PU_NOT_DEFINED

pu_name パラメーターが、定義されているどの PU 名にも一致しませんでした。

AP_SECURITY_LIST_NOT_DEFINED

security_list_name パラメーターが、定義されているどのセキュリティ・アクセス・リスト名にも一致しませんでした。

AP_LU_ALIAS_ALREADY_USED

この別名が付いた LU は、既に定義されています。接続経路指定データを除き、この verb を使用して既存の LU のどのパラメーターも変更することはできません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

デフォルト LU

アプリケーションが会話の開始時に LU 名を明示的に指定しなくても、ローカル LU の構成をセットアップすることができます。ノードは、アプリケーションの使用に適したデフォルト LU を選択します。これを実行する方法は、次のようにアプリケーションが従属 LU と独立 LU のどちらを必要とするかによって異なります。この機能を従属 LU と独立 LU の両方に提供することはできません。

- アプリケーションが従属 LU を必要とする場合は、1 つ以上の従属 LU についての DEFINE_LOCAL_LU で *default_pool* パラメーターを使用し、それらの LU をデフォルト LU として使用できることを指定します。アプリケーションでローカル LU 名を指定せずに会話を開始しようとした場合、Communications Server for Linux はデフォルト LU として定義された LU のプールから未使用の LU を選択します。
- 複数のノード上の LU をデフォルト LU として定義できます。デフォルト LU を要求するアプリケーションを、それらの LU のうち、使用可能な LU に割り当てることができます。その LU がアプリケーションと同じコンピューター上になくとも構いません。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、それらの LU を、デフォルト LU が定義されたすべてのノード上で定義しなければなりません (これにより、アプリケーションはデフォルトのローカル LU を使用して正しいパートナー LU に接続できます)。
- アプリケーションが独立 LU を必要とする場合は、*default_pool* パラメーターを使用してローカル LU をデフォルト LU として使用しないでください。この場合、デフォルト LU を要求するアプリケーションは、ローカル・ノードの CP に関連付けられている LU (Communications Server for Linux によって各ノードに対して自動的に定義された独立 LU) へ割り当てられます。

DEFINE_LS

DEFINE_LS は、新しいリンク・ステーション (LS) を定義するか、既存のリンク・ステーションを変更するために使用します。この verb を発行する前に、DEFINE_PORT verb を発行して、この LS が使用するポートを定義しておく必要があります。リンク固有のデータは、基本構造体に連結されます。

マルチパス・チャンネル (MPC) LS を定義する場合、ホストを識別するアドレッシング情報がポート構成に含まれているため、各 MPC ポート (マルチパス・チャンネル・デバイス */dev/mpcn* を定義) に関連付けることができるアクティブ LS は 1 つだけです。同時に複数のアクティブ MPC LS を使用するには、複数のポートを定義して、それぞれのポートに 1 つの LS を定義します (マルチパス・チャンネルは、System z 上の Communications Server for Linux でのみ使用可能です)。

DEFINE_LS を使用して、既存の LS が使用するポートを変更することはできません。この verb で指定する *port_name* は、LS の以前の定義に一致しなければなりません。LS は、始動されていない場合にのみ変更できます。

VCB 構造体

```
typedef struct define_ls
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;               /* reserved                      */
    unsigned char  format;                /* reserved                      */
    AP_UINT16      primary_rc;            /* primary return code          */
    AP_UINT32      secondary_rc;         /* secondary return code        */
    unsigned char  ls_name[8];           /* name of link station         */
    LS_DEF_DATA    def_data;             /* LS defined data              */
} DEFINE_LS;

typedef struct ls_def_data
{
    unsigned char  description[32];       /* resource description          */
}
```

DEFINE_LS

```

unsigned char    initially_active;    /* is this LS initially active? */
AP_UINT16       reserv2;             /* reserved */
AP_UINT16       react_timer;         /* timer for retrying failed LS */
AP_UINT16       react_timer_retry;   /* retry count for failed LS */
AP_UINT16       activation_count;     /* reserved */
unsigned char    restart_on_normal_deact; /* restart the link on any
                                       /* failure */

unsigned char    reserv3[7];          /* reserved */
unsigned char    port_name[8];        /* name of associated port */
unsigned char    adj_cp_name[17];     /* adjacent CP name */
unsigned char    adj_cp_type;         /* adjacent node type */
LINK_ADDRESS     dest_address;        /* destination address */
unsigned char    auto_act_supp;       /* auto-activate supported */
unsigned char    tg_number;           /* pre-assigned TG number */
unsigned char    limited_resource;    /* limited resource */
unsigned char    solicit_sscp_sessions; /* solicit SSCP sessions */
unsigned char    pu_name[8];          /* Local PU name (reserved if
                                       /* solicit_sscp_sessions is set
                                       /* to AP_NO)

unsigned char    disable_remote_act;  /* disable remote activation */
unsigned char    dspu_services;       /* Services provided for
                                       /* downstream PU

unsigned char    dspu_name[8];         /* Downstream PU name (reserved
                                       /* if dspu_services is AP_NONE)
unsigned char    dlus_name[17];       /* DLUS name if dspu_services
                                       /* set to AP_DLUR
unsigned char    bkup_dlus_name[17];  /* Backup DLUS name if
                                       /* dspu_services set to AP_DLUR

unsigned char    hpr_supported;        /* does the link support HPR? */
unsigned char    hpr_link_lvl_error;  /* does the link use link-level
                                       /* error recovery for HPR frms?

AP_UINT16       link_deact_timer;     /* link deactivation timer */
unsigned char    reserv1;             /* reserved */
unsigned char    default_nn_server;   /* default LS to NN server? */
unsigned char    ls_attributes[4];    /* LS attributes */
unsigned char    adj_node_id[4];      /* adjacent node ID */
unsigned char    local_node_id[4];    /* local node ID */
unsigned char    cp_cp_sess_support;  /* CP-CP session support */
unsigned char    use_default_tg_chars; /* Use the default tg_chars */
TG_DEFINED_CHARS tg_chars;            /* TG characteristics */
AP_UINT16       target_pacing_count;  /* target pacing count */
AP_UINT16       max_send_btu_size;    /* maximum send BTU size */
unsigned char    ls_role;              /* link station role */
unsigned char    max_ifrm_rcvd;        /* no. before acknowledgement */
AP_UINT16       dlus_retry_timeout;   /* seconds to recontact a DLUS */
AP_UINT16       dlus_retry_limit;     /* attempts to recontact a DLUS */
unsigned char    conventional_lu_compression; /* compression for LU 0-3? */
unsigned char    conventional_lu_cryptography; /* reserved */
unsigned char    reserv3a;             /* reserved */
unsigned char    retry_flags;          /* reserved */
AP_UINT16       max_activation_attempts; /* reserved */
AP_UINT16       activation_delay_timer; /* reserved */
unsigned char    branch_link_type;     /* is link an up or down link */
unsigned char    adj_brnn_cp_support;  /* adj CP allowed to be BrNN? */
unsigned char    mltg_pacing_algorithm; /* reserved */
unsigned char    reserv5;              /* reserved */
AP_UINT16       max_rcv_btu_size;     /* reserved */
unsigned char    tg_sharing_prohibited; /* reserved */
unsigned char    link_spec_data_format; /* reserved */
unsigned char    pu_can_send_dddlu_offline; /* does the PU send NMVT
                                       /* (power off) to host?

unsigned char    reserv4[13];          /* reserved */
AP_UINT16       link_spec_data_len;    /* length of link specific data */
} LS_DEF_DATA;

typedef struct tg_defined_chars
{
    unsigned char    effect_cap;        /* effective capacity */

```

```

    unsigned char  reserve1[5];          /* reserved */
    unsigned char  connect_cost;        /* connection cost */
    unsigned char  byte_cost;           /* byte cost */
    unsigned char  reserve2;           /* reserved */
    unsigned char  security;           /* security */
    unsigned char  prop_delay;         /* propagation delay */
    unsigned char  modem_class;        /* reserved */
    unsigned char  user_def_parm_1;    /* user-defined parameter 1 */
    unsigned char  user_def_parm_2;    /* user-defined parameter 2 */
    unsigned char  user_def_parm_3;    /* user-defined parameter 3 */
} TG_DEFINED_CHARS;
typedef struct link_address
{
    unsigned char  format;              /* type of link address */
    unsigned char  reserve1;           /* reserved */
    AP_UINT16     length;              /* length */
    unsigned char  address[135];       /* address */
} LINK_ADDRESS;

```

SDLC の DLC 固有データ:

```

typedef struct sdl_link_spec_data
{
    V0_MUX_INFO   mux_info;            /* Streams config info */
    AP_UINT16     reserve8;            /* reserved */
    AP_UINT16     reserve9;            /* reserved */
    AP_UINT32     contact_timer;       /* reserved */
    AP_UINT16     contact_timer_retry; /* reserved */
    AP_UINT16     reserve1;            /* reserved */
    AP_UINT32     contact_timer2;      /* reserved */
    AP_UINT16     contact_timer_retry2; /* reserved */
    AP_UINT16     reserve2;            /* reserved */
    AP_UINT32     disc_timer;          /* reserved */
    AP_UINT16     disc_timer_retry;    /* reserved */
    AP_UINT16     reserve3;            /* reserved */
    AP_UINT32     nve_poll_timer;      /* reserved */
    AP_UINT16     nve_poll_timer_retry; /* reserved */
    AP_UINT16     reserve4;            /* reserved */
    AP_UINT32     nve_poll_timer2;     /* reserved */
    AP_UINT16     nve_poll_timer_retry2; /* reserved */
    AP_UINT16     reserve5;            /* reserved */
    AP_UINT32     no_resp_timer;        /* reserved */
    AP_UINT16     no_resp_timer_retry; /* reserved */
    AP_UINT16     reserve6;            /* reserved */
    AP_UINT32     rem_busy_timer;       /* reserved */
    AP_UINT16     rem_busy_timer_retry; /* reserved */
    unsigned char re_tx_threshold;     /* reserved */
    unsigned char repoll_threshold;    /* reserved */
    AP_UINT32     rr_timer;            /* reserved */
    unsigned char group_address;        /* reserved */
    unsigned char poll_frame;          /* Poll frame to use when Primary */
                                           /* and contact polling secondary */
                                           /* XID, SNRM */
    AP_UINT16     poll_on_iframe;      /* reserved */
    AP_UINT16     stub_spec_data_len;  /* reserved */
    STUB_SPEC_DATA stub_spec_data;     /* reserved */
} SDL_LINK_SPEC_DATA;

```

QLLC の DLC 固有データ:

```

typedef struct vql_ls_spec_data
{
    V0_MUX_INFO   mux_info;            /* streams config info */
    AP_UINT16     reserve1;            /* reserved */
    AP_UINT16     reserve2;            /* reserved */
    unsigned char vc_type;             /* Virtual Circuit type */
    unsigned char req_rev_charge;      /* reserved */
}

```

DEFINE_LS

```
unsigned char loc_packet; /* reserved */
unsigned char rem_packet; /* reserved */
unsigned char loc_wsize; /* reserved */
unsigned char rem_wsize; /* reserved */
AP_UINT16 fac_len; /* X.25 facilities length */
unsigned char fac[128]; /* X.25 facilities */
AP_UINT16 retry_limit; /* reserved */
AP_UINT16 retry_timeout; /* reserved */
AP_UINT16 idle_timeout; /* reserved */
AP_UINT16 pvc_id; /* PVC logical channel identifier */
AP_UINT16 sn_id_len; /* reserved */
unsigned char sn_id[4]; /* reserved */
AP_UINT16 cud_len; /* length of any call user data
/* to send */
unsigned char cud[128]; /* actual call user data */
AP_UINT32 xtras; /* reserved */
AP_UINT32 xtra_len; /* reserved */
unsigned char rx_thruput_class; /* reserved */
unsigned char tx_thruput_class; /* reserved */
unsigned char cugo; /* reserved */
unsigned char cug; /* reserved */
AP_UINT16 cug_index; /* reserved */
AP_UINT16 nuid_length; /* reserved */
unsigned char nuid_data[109]; /* reserved */
unsigned char reserve3[2]; /* reserved */
unsigned char rpoa_count; /* reserved */
AP_UINT16 rpoa_ids[30]; /* reserved */
} VQL_LS_SPEC_DATA;
```

トークンリング、イーサネットの DLC 固有データ:

```
typedef struct llc_link_spec_data
{
    V0_MUX_INFO mux_info; /* Streams config info */
    AP_UINT16 reserve1; /* reserved */
    AP_UINT16 reserve2; /* reserved */
    AP_UINT16 length; /* reserved */
    AP_UINT16 xid_timer; /* XID timeout value in seconds */
    AP_UINT16 xid_timer_retry; /* XID retry limit */
    AP_UINT16 test_timer; /* TEST timeout value in seconds */
    AP_UINT16 test_timer_retry; /* TEST retry limit */
    AP_UINT16 ack_timeout; /* acknowledgment timeout in ms */
    AP_UINT16 p_bit_timeout; /* POLL response timeout in ms */
    AP_UINT16 t2_timeout; /* acknowledgment delay in ms */
    AP_UINT16 rej_timeout; /* REJ response timeout in seconds */
    AP_UINT16 busy_state_timeout; /* remote busy timeout in seconds */
    AP_UINT16 idle_timeout; /* idle RR interval in seconds */
    AP_UINT16 max_retry; /* retry limit for any response */
} LLC_LINK_SPEC_DATA;
```

マルチパス・チャンネル (MPC) の DLC 固有データ、System z 上の Communications Server for Linux のみ:

```
typedef struct chnl_link_spec_data
{
    V0_MUX_INFO mux_info; /* streams information */
    AP_UINT16 device_end; /* BlkMux protocol flag */
    unsigned char escd_port; /* reserved */
    unsigned char cuadd; /* reserved */
    unsigned char local_name[8]; /* reserved */
    unsigned char remote_name[8]; /* reserved */
    unsigned char reserve1[32]; /* pad and future expansion */
} CHNL_LINK_SPEC_DATA;
```

Enterprise Extender (HPR/IP) の DLC 固有データ:

```
typedef struct ipdlc_link_spec_data
{
    V0_MUX_INFO      mux_info;           /* streams information      */
    AP_UINT16        ack_timeout;        /* ACK timer for command frames */
    AP_UINT16        max_retry;          /* Retry limit for command frames */
    AP_UINT16        liveness_timeout;   /* Liveness timer          */
    unsigned char    short_hold_mode;    /* Run in short-hold mode    */
    unsigned char    remote_hostname[255]; /* Name of remote host to contact */
} IPDLC_LINK_SPEC_DATA;
```

すべての DLC タイプのデータ:

```
typedef struct v0_mux_info
{
    AP_UINT16        dlc_type;           /* DLC implementation type    */
    unsigned char    need_vrfy_fixup;    /* reserved                   */
    unsigned char    num_mux_ids;        /* reserved                   */
    AP_UINT32        card_type;          /* type of adapter card      */
    AP_UINT32        adapter_number;     /* DLC adapter number        */
    AP_UINT32        oem_data_length;    /* reserved                   */
    AP_INT32         mux_ids[5];         /* reserved                   */
} V0_MUX_INFO;
```

トークンリングまたはイーサネットの場合、link_address 構造体にある address パラメータは次のように置換されます。

```
typedef struct tr_address
{
    unsigned char    mac_address[6];     /* MAC address                */
    unsigned char    lsap_address;       /* local SAP address          */
} TR_ADDRESS;
```

Enterprise Extender (HPR/IP) の場合、link_address 構造体にある address パラメータは次のように置換されます。

```
typedef struct ip_address_info
{
    unsigned char    lsap;               /* Local Service Access Point addr */
    unsigned char    version;            /* IPv4 or IPv6                */
    unsigned char    address[272];       /* reserved                     */
} IP_ADDRESS_INFO;
```

MPC の場合、link_address 構造体は予約済みです。

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_LS

ls_name

リンク・ステーションの名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

def_data.description

LS を記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、`QUERY_LS` verb、`QUERY_PU` verb、および `QUERY_DOWNSTREAM_PU` verb で戻されますが、`Communications Server for Linux` では他の用途に使用されません。

def_data.initially_active

この LS をノードの始動時に自動的に起動するかどうかを指定します。値は次のいずれかです。

AP_YES LS はノードの起動時に自動的に始動します。

AP_NO LS は自動的に始動しません。手動で始動させる必要があります。

LS が専用 SDLC リンク または QLLC PVC リンクの場合は、このパラメーターを AP_YES に設定して、必ず、このリンクを常時使用可能にしておくことをお勧めします。

def_data.react_timer

障害が起きた LS を再活動化するための再活動化タイマー。下の *react_timer_retry* パラメーターがゼロ以外の場合、このパラメーターは、障害が起きた LS の活動化を Communications Server for Linux が再試行するように指定するため、再試行の時間間隔を秒単位で指定します。LS で障害が起きた場合、または障害が起きた LS の再活動化の試行が失敗した場合、Communications Server for Linux は、指定した時間まで待機してから活動化を再試行します。*react_timer_retry* がゼロの場合は、このパラメーターは無視されます。

def_data.react_timer_retry

障害が起きた LS を再活動化するための再試行カウント。このパラメーターは、LS の使用中に障害が起きた (または、LS 開始の試行が失敗した) ときに、Communications Server for Linux が LS の再活動化を試行するかどうかを指定する場合に使用されます。

Communications Server for Linux に LS の再活動化を試行させないことを示すためにゼロを指定するか、実行する再試行の回数を指定します。65,535 の値は、LS が活動化されるまで Communications Server for Linux が無制限に再試行することを示します。

Communications Server for Linux は、再試行から次の再試行までの間、上記の *react_timer* パラメーターで指定した時間を待機します。LS が正常に再活動化されないで再試行カウントに達した場合、または Communications Server for Linux が活動化を再試行しているときに STOP_LS が発行された場合、再試行はそれ以上実行されず、LS に対して START_LS が発行されない限り、LS は非活動状態のままになります。

auto_act_supp パラメーターを AP_YES に設定した場合、再活動化タイマーの各フィールドは無視されます。リンクに障害が起きた場合、Communications Server for Linux は、そのセッションを使用していたユーザー・アプリケーションがセッションの再起動を試行するまで、そのリンクの再活動化を試行しません。

LS が専用 SDLC リンクまたは QLLC PVC リンクの場合は、このパラメーターをゼロ以外の値に設定して、必ず、このリンクを常時使用可能にしておくことをお勧めします。

def_data.restart_on_normal_deact

LS がリモート・システムで正常に非活動化された場合に、Communications Server for Linux に LS の再活動化を試行させるかどうかを指定します。値は次のいずれかです。

AP_YES リモート・システムが LS を正常に非活動化した場合、Communications Server for Linux は、障害のあった LS を再活動化する場合と同じ再試行タイマー値およびカウンタ値 (上記の *react_timer* および *react_timer_retry* パラメーター) を使用して、LS の再活動化を試行します。

AP_NO リモート・システムが LS を正常に非活動化した場合、Communications Server for Linux は LS の再活動化を試行しません。

LS が (*def_data.adj_cp_type* パラメーターで指定した) ホスト・リンクである場合、あるいはノードの始動時に自動的に開始された (*initially_active* パラメーターを **AP_YES** に設定) 場合、このパラメーターは無視されます。リモート・システムが通常どおり LS を非活動化すると、Communications Server for Linux は常に LS の再活動化を試みます (*react_timer_retry* がゼロでない場合)。

def_data.port_name

このリンク・ステーションに関連したポートの名前。これは 8 バイトの ASCII スtringで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。この名前は、定義されているポートの名前に一致しなければなりません。

def_data.adj_cp_name

この LS の隣接 CP の完全修飾名。

次の *adj_cp_type* パラメーターを **AP_NETWORK_NODE** または **AP_END_NODE** に設定し、事前に割り当てられた TG 番号を使用する場合は、このパラメーターを隣接ノードで定義された CP 名に設定します。隣接ノードが XID 交換のときに CP 名を送信する場合、その CP 名はこの値と突き合わせて検査されます。

adj_cp_type を **AP_BACK_LEVEL_LEN_NODE** に設定した場合、Communications Server for Linux は、この値を ID としてのみ使用します。この値には、このノードで定義されている他の CP 名と一致しない (以下に説明するフォーマットの) スtringに設定してください。

adj_cp_type をその他の値に設定した場合、または事前に割り当てられた TG 番号を使用しない場合、このパラメーターを指定する必要はありません。Communications Server for Linux は CP 名が指定された場合にのみ CP 名を検査します。

この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A スtring文字からなるネットワーク名で構成されます。

def_data.adj_cp_type

隣接ノード・タイプ。

隣接ノードが APPN ノードであり、事前に割り当てられた TG 番号を使用しない場合、通常、TG 番号はノード・タイプが不明であることを示す AP_APPN_NODE に設定されます。 Communications Server for Linux は、XID 交換のときにタイプを判別します。

事前に割り当てられた TG 番号を使用する場合は、ノード・タイプを明示的に指定しなければなりません。事前に割り当てられた TG 番号を使用しない場合でも、追加のセキュリティー検査としてノード・タイプを指定できます。この場合、Communications Server for Linux は、ここで隣接ノードのノード・タイプが指定したタイプに一致しなければ、隣接ノードからの接続の試みをリジェクトします。次のいずれかの値を使用します。

AP_APPN_NODE

ノード・タイプは不明です。 Communications Server for Linux は、XID 交換のときにタイプを判別します。

AP_END_NODE

エンド・ノード、ローカル・ノードの全体像のエンド・ノードとして機能する分岐ネットワーク・ノード、または上位レベル LEN ノード (XID3 にネットワーク名 CV が含まれているノード)。

AP_NETWORK_NODE

ネットワーク・ノード、またはローカル・ノードの全体像のネットワーク・ノードとして機能する分岐ネットワーク・ノード。

隣接ノードが APPN ノードでない場合は、次のいずれかの値を使用します。これらの値は Enterprise Extender、MPC リンクには無効であり、APPN ノードに使用するものです。

AP_BACK_LEVEL_LEN_NODE

バックレベル LEN ノード (XID3 にネットワーク名 CV が含まれていないノード)。

AP_HOST_XID3

ホスト・ノード。 Communications Server for Linux はノードからのポーリング XID にフォーマット 3 XID で応答します。

AP_HOST_XID0

ホスト・ノード。 Communications Server for Linux はノードからのポーリング XID にフォーマット 0 XID で応答します。

AP_DSPU_XID

ダウンストリーム PU。 Communications Server for Linux はリンクの活動化に XID 交換を含めます。 *dspu_name* フィールドと *dspu_services* フィールドも設定する必要があります。

AP_DSPU_NOXID

ダウンストリーム PU。 Communications Server for Linux はリンクの活動化に XID 交換を含めません。 *dspu_name* フィールドと *dspu_services* フィールドも設定する必要があります。

この LS 上で独立 LU 6.2 トラフィックを稼働させたい場合は、*adj_cp_type* パラメーターを AP_APPN_NODE、AP_END_NODE、AP_NETWORK_NODE、AP_BACK_LEVEL_LEN_NODE のいずれかに設定しなければなりません。

def_data.dest_address.format

MPC の場合、このパラメーターは予約済みです。

指定したリンク・アドレスのタイプ。値は次のいずれかです。

AP_IP_ADDRESS_INFO

IP アドレス。Enterprise Extender (HPR/IP) リンクでこの値を指定します。

AP_UNSPECIFIED

未指定のアドレス・フォーマット。Enterprise Extender (HPR/IP) 以外のすべてのリンク・タイプでこの値を指定します。

def_data.dest_address.length

MPC の場合、このパラメーターは予約済みです。

次のパラメーターで指定した宛先アドレス・フィールドの長さ。

Enterprise Extender (HPR/IP) の場合、このパラメーターと *dest_address.address* は予約済みです。代わりに、リンク固有データの *remote_hostname* パラメーターを使用してアドレスを指定します。

SDLC の場合:

def_data.dest_address.address

この LS 上の 2 次ステーションのアドレス

- この LS を所有するポートが着呼だけに使用される場合 (DEFINE_PORT の *out_link_act_lim* がゼロ)、このパラメーターは予約済みです。
- この LS を所有するポートが交換用の 1 次であり、発呼用に使用される場合 (DEFINE_PORT 上の *port_type* は AP_SWITCHED、*ls_role* は AP_LS_PRI、および *out_link_act_lim* はゼロ以外の値)、このパラメーターには、2 次ステーションにどのようなアドレスが構成されていても受け入れるようにするための 0xFF を設定するか、または 2 次ステーションに構成されている値と一致しなければならない 0x01 ~ 0xFE の範囲の 1 バイト値に設定します。
- それ以外は、リンク・ステーションを識別するための 0x01 ~ 0xFE の範囲の 1 バイト値に設定します。ポートが 1 次のマルチドロップの場合 (DEFINE_PORT 上の *ls_role* が AP_LS_PRI および *tot_link_act_lim* が 1 より大きい)、このアドレスはポート上の各アドレスごとに異ならなければなりません。

QLLC の場合:

def_data.dest_address.address

この LS の宛先ノードのアドレス。このパラメーターは SVC 発呼だけに使用されます (リンク固有データの *vc_type* パラメーター、および DEFINE_PORT 上のリンク活動化限度パラメーターによって定義される)。着呼または PVC に対しては無視されます。

アドレスは 1 ~ 14 文字のストリングです。アドレスは X.25 (1980) フォーマットであり、それ以降のアドレス・フォーマットはサポートされていません。

トークンリング、イーサネットの場合:

def_data.dest_address.mac_address

隣接ノードの MAC アドレス。

非選択 listen LS (着呼専用を使用するが、従属 LU トラフィックをサポートする定義済み LU を持つもの) を定義する必要がある場合、このパラメーターをヌル・ストリングに設定します。この LS はすべてのリンク・セッションからの着呼の受信に使用できますが、発呼には使用できません。独立 LU トラフィックのみを使用している場合には、非選択 listen LS を定義する必要はありません。これは、独立 LU トラフィックの LS は、必要に応じて動的にセットアップすることができるからです。

ローカル・ノードと隣接ノードが、ブリッジで接続した異なるタイプ (一方がトークンリングで、他方がイーサネット) の LAN 上にあるときは、ほとんどの場合、MAC アドレス内のバイトのビット順序を逆にする必要があります。詳しくは、163 ページの『MAC アドレス内のビット順序』を参照してください。2 つのノードが同じ LAN にある場合、またはブリッジで接続された同じタイプの複数の LAN にある場合は、変更する必要はありません。

def_data.dest_address.lsap_address

隣接ノードのローカル SAP アドレス。これは、0x04 の倍数でなくてはなりません。

Enterprise Extender (HPR/IP) の場合:

def_data.dest_address.ip_address_info.lsap

Enterprise Extender の場合: ポートのローカル SAP アドレス。0x04 ~ 0xEC の範囲で 0x04 の倍数を指定してください。通常値は 0x04 ですが、VTAM® は一部の環境では 0x08 を使用できます。

同じ TCP/IP インターフェースで異なる LSAP アドレスを指定して複数のポートを使用する必要がある場合は、複数の Enterprise Extender DLC を作成する必要があり、そのために、それぞれの DLC には同じ *if_name* (ただし、異なる LSAP アドレス) を指定して、個別の Enterprise Extender ポートを作成します。

def_data.dest_address.ip_address_info.version

Enterprise Extender の場合: 以下のフィールドが IPv4 または IPv6 アドレスを表すかどうかを指定します。値は次のいずれかです。

IP_VERSION_4_HOSTNAME

address フィールドには、IPv4 アドレス、あるいは IPv4 アドレスに解決するホスト名または別名を指定します。

IP_VERSION_6_HOSTNAME

address フィールドには、IPv6 アドレス、あるいは IPv6 アドレスに解決するホスト名または別名を指定します。

すべてのリンク・タイプの場合:

def_data.auto_act_supp

セッションが必要なときに、リンクを自動的に活動化できるかどうかを指定します。値は次のいずれかです。

AP_YES リンクは自動的に活動化できます。

AP_NO リンクは自動的に活動化できません。

このパラメーターを AP_YES に設定した場合:

- 再活動化タイマーの各フィールドは無視されます。LS に障害が起きた場合、Communications Server for Linux は、そのセッションを使用していた従属 LU アプリケーションがセッションの再起動を試みるまで、その LS の再活動化を試みません。独立 LU が使用していた LS は Communications Server for Linux によって再活動化されないため、手動で再始動しなければなりません。
- APPN ノードへのリンクである場合、LS には事前に割り当てられた TG 番号が定義されていなければならず (以下のパラメーターを参照)、*cp_cp_sess_support* を AP_NO に設定する必要があります。
- ローカル・ノードまたは隣接ノードのどちらかがエンド・ノードの場合、LS は隣接ノードで自動活動化が可能として定義されている必要もありません。

def_data.tg_number

事前に割り当てられた TG 番号。このパラメーターは、隣接ノードが APPN ノードの場合 (*adj_cp_type* が AP_NETWORK_NODE または AP_END_NODE の場合) のみ使用され、それ以外の場合は無視されます。

この TG 番号は、リンクを活動化するとき、そのリンクを表すために使用されます。ノードは、このリンクの活動化のとき、隣接ノードからこれ以外の番号を受け入れません。隣接ノードが、事前に割り当てられた TG 番号を使用している場合は、同じ TG 番号が隣接リンク・ステーション上の隣接ノードによって定義されていなければなりません。

ローカル・ノードが LEN ノードである場合、または隣接ノードが LEN ノードでリンクを自動的に活動可能にしたい場合は、TG 番号を 1 に設定します。それ以外の場合は、1 ~ 20 の範囲の数値を指定するか、リンクの活動化のときに TG 番号が事前に割り当てられず、折衝されることを示すゼロを指定します。

事前に割り当てられた TG 番号を定義する場合は、*adj_cp_name* パラメーターも定義しなければならず、*adj_cp_type* パラメーターを AP_END_NODE、AP_NETWORK_NODE のどちらかに設定しなければなりません。

def_data.limited_resource

リンクを使用するセッションがないとき、このリンク・ステーションを非活動化するかどうかを指定します。非交換ポート上のリンク・ステーションは、限定リソースとして構成することはできません。値は次のいずれかです。

AP_NO リンクは限定リソースでないため、自動的に非活動化されません。

AP_NO_SESSIONS

リンクは限定リソースであるため、それを使用する活動状態であるセッションがなくなると、自動的に非活動化されます。

AP_INACTIVITY

リンクは限定リソースであり、このリンクを使用する活動セッショ

ンがなくなったとき、または *link_deact_timer* フィールドに指定した期間に、このリンク上にデータが流れなかった場合、このリンクは自動的に非活動化されます。

- リンク中に活動状態である SSCP-PU セッションがなくなると、ノードはリンクを即時に非活動化します。
- SSCP-PU セッションが活動状態であっても、指定の時間枠内にトラフィックがない場合、ノードはホストに REQDISCONT(normal) を送信します。その後、ホストがすべての LU および PU の非活動化を担当し、同時にローカル・ノードがリンクを非活動化します。しかし、ホストは活動状態である PLU-SLU セッションを持つ LU を非活動化しないことがあります。この場合、これらのセッションがすべて非活動化される (たとえば、ユーザーのログアウト) まで、リンクは活動状態のままです。この振る舞いは、START_NODE verb の *ptf* フィールドのオプションを使用して変更することができます。

このフィールドを AP_NO_SESSIONS に設定し、*cp_cp_sess_support* を AP_YES に設定することにより、限定リソース・リンク・ステーションを CP-CP セッション・サポート用に構成することもできます。その場合、このリンクを介して CP-CP セッションを起動すると、Communications Server for Linux は、このリンクを限定リソースとして処理しません (したがって、このリンクを非活動化しません)。

def_data.solicit_sscp_sessions

Enterprise Extender (HPR/IP) または MPC ポートの場合、このパラメーターは予約済みです。

隣接ノードに、SSCP とローカル CP および従属 LU 間のセッションを開始するよう要求するかどうかを指定します。このパラメーターは、隣接ノードが APPN ノードの場合 (*adj_cp_type* が AP_NETWORK_NODE または AP_END_NODE の場合) のみ使用され、それ以外の場合は無視されます。隣接ノードがホストである場合 (*adj_cp_type* が AP_HOST_XID3 または AP_HOST_XID0 のいずれかである場合)、Communications Server for Linux は常にホストに SSCP セッションを開始するよう要求します。

値は次のいずれかです。

AP_YES 隣接ノードに SSCP セッションの開始を要求します。

AP_NO 隣接ノードに SSCP セッションの開始を要求しません。

隣接ノードが APPN ノードで、*dspu_services* を AP_NONE 以外の値に設定した場合、このパラメーターは AP_NO に設定しなければなりません。

def_data.pu_name

Enterprise Extender (HPR/IP) または MPC ポートの場合、このパラメーターは予約済みです。

このリンクを使用するローカル PU の名前。このパラメーターは、*adj_cp_type* を AP_HOST_XID3 または AP_HOST_XID0 に設定した場合、または *solicit_sscp_sessions* を AP_YES に設定した場合にのみ使用され、それ以

外の場合は無視されます。これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

def_data.disable_remote_act

リモート・ノードによる LS の活動化をできないようにするかどうかを指定します。値は次のいずれかです。

AP_YES LS は、ローカル・ノードのみから活動化できます。LS をリモート・ノードが活動化しようとした場合、Communications Server for Linux はその活動化をリジェクトします。

AP_NO LS は、リモート・ノードから活動化できます。

def_data.dspu_services

Enterprise Extender (HPR/IP) または MPC ポートの場合、このパラメーターは予約済みです。

ローカル・ノードがこのリンクを介してダウンストリーム PU に提供するサービスを指定します。このパラメーターは、隣接ノードがダウンストリーム PU であるか、*solicit_sscp_sessions* を AP_NO に設定した APPN ノードである場合のみ使用されます。それ以外の場合は予約済みです。値は次のいずれかです。

AP_PU_CONCENTRATION

ローカル・ノードは、ダウンストリーム PU に SNA ゲートウェイを提供します。ローカル・ノードは、SNA ゲートウェイをサポートするように定義されていなければなりません。

AP_DLUR

ローカル・ノードは、ダウンストリーム PU に DLUR サービスを提供します。ローカル・ノードは、DLUR をサポートするように定義されていなければなりません。(エンド・ノードではサポートされていません。)

AP_NONE

ローカル・ノードは、このダウンストリーム PU にサービスを提供しません。

def_data.dspu_name

Enterprise Extender (HPR/IP) または MPC ポートの場合、このパラメーターは予約済みです。

ダウンストリーム PU の名前。名前は 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

このパラメーターは、次の両方の条件に当てはまる場合には必須ですが、それ以外の場合は予約済みです。

- *solicit_sscp_sessions* パラメーターが AP_NO に設定されていること
- *dspu_services* パラメーターが AP_PU_CONCENTRATION または AP_DLUR に設定されていること

DLUR に対してダウンストリーム PU が使用されている場合、この名前はホストで構成されている PU 名と一致している必要があります

(Communications Server for Linux は PU を識別するために PU 名および PU ID の両方をホストへ送信します。ホストは、通常、PU 名によって PU を識別し、一致する PU 名が見つからない場合は、PU ID によって PU を識別します)。

def_data.dlus_name

Enterprise Extender (HPR/IP) または MPC ポートの場合、このパラメーターは予約済みです。

ダウンストリーム・ノードに対してリンクを活動化するとき、DLUR が SSCP サービスを請求する請求元の DLUS ノードの名前。 *dspu_services* が AP_DLUR に設定されていない場合、このフィールドは予約済みです。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringです。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

DEFINE_DLUR_DEFAULTS verb を使用して定義したグローバル・デフォルト DLUS を指定するには、このパラメーターを 17 桁の 2 進ゼロに設定します。このパラメーターをゼロに設定したときにグローバル・デフォルト DLUS がないと、DLUR はリンクを活動化するとき SSCP 接続を開始しません。

def_data.bkup_dlus_name

Enterprise Extender (HPR/IP) または MPC ポートの場合、このパラメーターは予約済みです。

dlus_name で指定したノードが活動状態でない場合に、DLUR が SSCP サービスを請求する送信元となるバックアップ DLUS ノードの名前。 *dspu_services* が AP_DLUR に設定されていない場合、このフィールドは予約済みです。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringです。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

DEFINE_DLUR_DEFAULTS verb を使用して定義したグローバル・バックアップ・デフォルト DLUS を指定するには、このパラメーターを 17 桁の 2 進ゼロに設定します。

def_data.hpr_supported

このリンクで HPR がサポートされているかどうかを指定します。リンクが Enterprise Extender (HPR/IP) リンクである場合、このパラメーターを AP_YES に設定しなければなりません。リンクが MPC リンクの場合、このパラメーターは AP_NO に設定する必要があります。そのほかのリンクの場合、*adj_cp_type* パラメーターがリンクを APPN ノードに接続するよう指示していない限り、AP_NO に設定しなければなりません。値は次のいずれかです。

AP_YES このリンクでは HPR がサポートされています。

AP_NO このリンクでは HPR はサポートされていません。

def_data.hpr_link_lvl_error

このリンク上でリンク・レベル・エラー回復を使用して HPR トラフィックを送信するかどうかを指定します。このパラメーターは、*hpr_supported* を AP_YES に設定した場合以外は無視されます。

このパラメーターは SDLC/チャンネル/ MPC+/Enterprise Extender (HPR/IP) の各リンク用に予約済みです。

値は次のいずれかです。

AP_YES このリンク上でリンク・レベル・エラー回復を使用して HPR トラフィックを送信します。

AP_NO このリンク上でリンク・レベル・エラー回復を使用して HPR トラフィックを送信しません。

def_data.link_deact_timer

限定リソース・リンクの非活動化タイマー (秒単位)。限定リソース・リンクは、このパラメーターで指定した時間にリンク上でデータ・フローがなければ、自動的に非活動化されます。このパラメーターは、*limited_resource* を INACTIVITY 以外の値に設定した場合には、使用されません。

最小値は 5 で、1 ~ 4 の範囲の値は 5 とみなされます。

値 0 (ゼロ) は、次のいずれかを示します。

- *hpr_supported* パラメーターを AP_YES に設定した場合は、デフォルト非活動化タイマーの値として 30 が使用されます。
- *hpr_supported* パラメーターを AP_NO に設定した場合は、タイムアウトが使用されません (*limited_resource* を AP_NO に設定した場合と同様に、リンクは非活動化されません)。

def_data.default_nn_server

エンド・ノード。ローカル・ノードのネットワーク・ノード・サーバーとして機能できる、ネットワーク・ノードとの CP-CP セッションをサポートするリンクであるかどうかを指定します。ローカル・ノードにネットワーク・ノード・サーバーとの CP-CP セッションがなく、それを確立する必要がある場合は、ローカル・ノードはその定義済み LS でこのパラメーターを検査し、活動化に適した LS を検索します。これにより、どの隣接 NN が NN サーバーとして機能するのに適しているかを指定できます (たとえば、コストのかかるリンクまたは低速のリンクを介してアクセスする NN を使用しないようにします)。

値は次のいずれかです。

AP_YES このリンクは、ローカル・ノードの NN サーバーとして機能できるネットワーク・ノードへの CP-CP セッションをサポートします。ローカル・ノードは、NN サーバーに接続する必要がある場合には、このリンクを自動的に活動化できます。 *cp_cp_sess_support* パラメーターを AP_YES に設定しなければなりません。

AP_NO ネットワーク・ノード・サーバーと接続しようとしても、このリンクは自動的に活動化しません。

ローカル・ノードがエンド・ノードでない場合、このパラメーターは無視されます。

def_data.ls_attributes

この配列には、次のパラメーターで記述する隣接ノードに関する詳しい情報が入ります。

def_data.ls_attributes[0]

ホスト・タイプ。これは、AP_SNA に設定してください。ただし、次のリストにないタイプのホストと通信する場合は除きます。値は次のいずれかです。

AP_SNA 標準 SNA ホスト。

AP_FNA 富士通 Network Architecture (VTAM-F) ホスト。

AP_HNA 日立 Network Architecture ホスト。

def_data.ls_attributes[1]

バックレベル LEN ノードとのリンクのネットワーク名 CV 抑止。

adj_cp_type を AP_BACK_LEVEL_LEN_NODE または AP_HOST_XID3 に設定した場合は、次のいずれかの値を使用して、LEN ノードへ送信するフォーマット 3 XID フォーマットにネットワーク名 CV を組み込まないようにするかどうかを指定します。

AP_NO XID にネットワーク名 CV を組み込みます。

AP_SUPPRESS_CP_NAME

ネットワーク名 CV を組み込みません。

adj_cp_type をその他の値に設定した場合、このパラメーターは無視されません。

def_data.adj_node_id

MPC リンクの場合、このパラメーターは予約済みです。

隣接ノードのノード ID。これは 4 バイトの 16 進数ストリングで、ブロック番号 (3 桁の 16 進数字) とノード番号 (5 桁の 16 進数字) で構成されます。ノード ID 検査を使用不可にするには、これをゼロに設定します。このリンク・ステーションが交換ポート上に定義されている場合、*node_id* は固有でなければならず、1 つの交換ポート上に存在できるヌルの *node_id* は 1 つのみです。

def_data.local_node_id

MPC リンクの場合、このパラメーターは予約済みです。

この LS の XID で送信されるノード ID。これは 4 バイトの 16 進数ストリングで、ブロック番号 (3 桁の 16 進数字) とノード番号 (5 桁の 16 進数字) で構成されます。DEFINE_NODE verb で指定したノード ID を使用するには、このパラメーターをゼロに設定します。

def_data.cp_cp_sess_support

CP-CP セッションがサポートされるかどうかを指定します。このパラメーターは、隣接ノードがエンド・ノードかネットワーク・ノードである場合 (*adj_cp_type* が AP_NETWORK_NODE、AP_END_NODE、AP_APPN_NODE のいずれかである場合) にのみ有効で、それ以外の場合は無視されます。ローカル・ノードと隣接ノードの両方がネットワーク・ノードである場合、これらのノード間で APPN 機能を使用するには、このパラメーターを AP_YES に設定してください。

値は次のいずれかです。

AP_YES CP-CP セッションがサポートされます。MPC または MPC+ LS の場合、*solicit_sscp_sessions* パラメーターを AP_NO に設定しなければなりません。

AP_NO CP-CP セッションはサポートされません。

def_data.use_default_tg_chars

DEFINE_PORT verb で指定したデフォルト TG 特性を使用するかどうかを指定します。TG 特性は APPN ノードに対するリンクである場合にのみ適用されます。それ以外の場合、このパラメーターと *tg_chars* 構造体のパラメーターは無視されます。値は次のいずれかです。

AP_YES デフォルト TG 特性を使用します。この verb の *tg_chars* 構造体は無視されます。

AP_NO この verb の *tg_chars* 構造体を使用します。

def_data.tg_chars.effect_cap

実際のビット/秒率 (回線速度)。値は 1 バイトの浮動小数点数としてエンコードされます。このときの公式は、 $0.1 \text{ mmm} * 2^{\text{eeee}}$ で、バイトのビット表現は *b'eeeeemmm'* です。300 ビット/秒の倍数で指定してください。

イーサネットまたは Enterprise Extender (HPR/IP) リンクの場合、このパラメーターには、パスのステップダウンまたはボトルネックを含めて、リンクで使用されるアダプターの理論的なキャパシティーではなく、必ずリンクの真の「有効キャパシティー」を設定します。たとえば、GigE アダプターは 1 ギガビットを処理できますが、リンクがイーサネットを使用する場合は、100MBps 以下を指定する必要がある FastEthernet を使用するターゲット・ボックスに切り替えます。

def_data.tg_chars.connect_cost

接続時間当たりのコスト。有効な値は 0 ~ 255 の範囲の整数値で、0 が接続時間当たりの最低コスト、255 が最高コストです。

def_data.tg_chars.byte_cost

バイト当たりのコスト。有効な値は 0 ~ 255 の範囲の整数値で、0 がバイト当たりの最低コスト、255 が最高コストです。

def_data.tg_chars.security

ネットワークのセキュリティー・レベル。値は次のいずれかです。

AP_SEC_NONSECURE

セキュリティーなし。

AP_SEC_PUBLIC_SWITCHED_NETWORK

公衆交換網を介してデータを伝送します。

AP_SEC_UNDERGROUND_CABLE

保護された地下ケーブルを介してデータを伝送します。

AP_SEC_SECURE_CONDUIT

ガードされていない保護されたコンジット (配管) 内の回線を介してデータを伝送します。

AP_SEC_GUARDED_CONDUIT

物理接続による盗聴から保護されたコンジット内の回線を介してデータを伝送します。

AP_SEC_ENCRYPTED

回線を介して伝送する前にデータを暗号化します。

AP_SEC_GUARDED_RADIATION

物理接続による盗聴および電波盗聴から保護された回線を介してデータを伝送します。

def_data.tg_chars.prop_delay

伝搬遅延: 信号をリンクの長さのみ伝送するのに要する時間。リンクのタイプに応じて、次のいずれかの値を指定します。

AP_PROP_DELAY_MINIMUM

最小の伝搬遅延。

AP_PROP_DELAY_LAN

遅延は 480 マイクロ秒未満です (LAN の通常値)。

AP_PROP_DELAY_TELEPHONE

遅延は 480 ~ 49,512 マイクロ秒の範囲です (電話網の通常値)。

AP_PROP_DELAY_PKT_SWITCHED_NET

遅延は 49,512 ~ 245,760 マイクロ秒の範囲です (パケット交換ネットワークの通常値)。

AP_PROP_DELAY_SATELLITE

遅延は 245,760 マイクロ秒を超えます (サテライト・リンクの通常値)。

AP_PROP_DELAY_MAXIMUM

最大の伝搬遅延。

def_data.tg_chars.user_def_parm_1 ~ def_data.tg_chars.user_def_parm_3

ユーザー定義パラメーター。上記のパラメーターで対応できないその他の TG 特性を組み込むために使用できます。これらのパラメーターは、それぞれ 1 ~ 255 の範囲で値を設定しなければなりません。

def_data.target_pacing_count

求めるペーシング・ウィンドウ・サイズを示す 1 以上 32,767 以下の数値。(現行バージョンの Communications Server for Linux では、この値は使用されません。)

def_data.max_send_btu_size

このリンク・ステーションから送信できる最大 BTU サイズ。この値は、1 対のリンク・ステーションが相互の通信に使用できる最大 BTU サイズを折衝するときに使用されます。この値には、RU のみでなく TH と RH の長さ (合計 9 バイト) も含まれます。265 ~ 65535 (SDLC の場合は 265 ~ 4105) の範囲で値を指定します。

def_data.ls_role

リンク・ステーションの役割。このパラメーターは通常、LS を所有している

るポートの定義からこの LS の役割を取ることを指定する
 AP_USE_PORT_DEFAULTS に設定します。 MPC リンクの場合は、有効な値は
 この値のみです。

個別の LS についてポートの LS の役割を上書きする必要がある場合は、
 次のいずれかの値を指定します。

AP_LS_PRI

1 次

AP_LS_SEC

2 次

AP_LS_NEG

折衝可能

Enterprise Extender (HPR/IP) ポートの場合、AP_USE_PORT_DEFAULTS を使用
 しなければなりません。ポートの LS 役割を変更することはできません。

def_data.max_ifrm_rcvd

このリンク・ステーションが、肯定応答を送信する間に受信できる I フレームの最大数。 0 ~ 127 の範囲で値を指定します。 0 を指定した場合は、ポート定義からの値が使用されます。

def_data.dlus_retry_timeout

Enterprise Extender (HPR/IP) または MPC ポートの場合、このパラメーターは予約済みです。

DLUS に接続するための再活動化タイマー。このパラメーターには、Communications Server for Linux が DLUS との接続に失敗した場合に再試行する時間間隔 (秒単位) を指定します。

0x0001 ~ 0xFFFF の範囲で値を指定します。

def_data.dlus_retry_limit

Enterprise Extender (HPR/IP) または MPC ポートの場合、このパラメーターは予約済みです。

dlus_name パラメーターと *bkup_dlus_name* パラメーターで指定した DLUS との接続を 2 回目以降に試みる場合の間隔 (秒単位)。 0x0001 ~ 0xFFFF の範囲で値を指定するか、Communications Server for Linux が DLUS に接続するまで無限に再試行することを示す 0xFFFF を指定します。 1 回目の試行と 2 回目の試行の間隔は常に 1 秒です。ゼロを指定した場合は、**DEFINE_DLUR_DEFAULTS verb** で指定したデフォルト値が使用されます。このパラメーターは、*dspu_services* パラメーターが AP_DLUR に設定されていない場合は、無視されます。

def_data.conventional_lu_compression

MPC リンクの場合、このパラメーターは予約済みです。

このリンク上の LU 0 から 3 のセッションでデータ圧縮が要求されるかどうかを指定します。このパラメーターは、このリンクが LU 0 から 3 のトラフィックを伝送する場合にのみ使用されます。LU 6.2 セッションには適用されません。

値は次のいずれかです。

AP_YES データ圧縮は、ホストが要求した場合にこのリンク上の LU 0 から 3 のセッションに対して使用されます。

AP_NO データ圧縮は、このリンク上の LU 0 から 3 のセッションに対して使用されません。

def_data.branch_link_type

このパラメーターが適用されるのは、ローカル・ノードが分岐ネットワーク・ノードである場合のみです。ローカル・ノードが他のタイプの場合は予約済みです。

パラメーター *def_data.adj_cp_type* が AP_NETWORK_NODE、AP_END_NODE、AP_APPN_NODE、または AP_BACK_LEVEL_LEN_NODE のいずれかに設定されている場合は、このパラメーターは、リンクがアップリンクであるかまたはダウンリンクであるかを定義します。値は次のいずれかです。

AP_UPLINK

リンクはアップリンクです。

AP_DOWNLINK

リンクはダウンリンクです。

def_data.adj_cp_type が AP_NETWORK_NODE に設定されている場合、このパラメーターは AP_UPLINK に設定しなければなりません。

def_data.adj_brnn_cp_support

このパラメーターが適用されるのは、ローカル・ノードが分岐ネットワーク・ノードで、隣接ノードがネットワーク・ノードの場合 (パラメーター *def_data.adj_cp_type* が AP_NETWORK_NODE に設定されているか、あるいは AP_APPN_NODE に設定され、XID 交換時に発見されたノード・タイプがネットワーク・ノードの場合) のみです。ローカル・ノードとリモート・ノードが他のタイプの場合は、予約済みです。

このパラメーターは、隣接ノードを、ローカル・ノードから見てネットワーク・ノードとして機能している分岐ネットワーク・ノードにすることができるかどうかを定義します。値は次のいずれかです。

AP_BRNN_ALLOWED

隣接ノードを分岐ネットワーク・ノードにすることができます (ただし、必須ではありません)。

AP_BRNN_REQUIRED

隣接ノードは分岐ネットワーク・ノードでなければなりません。

AP_BRNN_PROHIBITED

隣接ノードを分岐ネットワーク・ノードにすることはできません。

def_data.adj_cp_type が AP_NETWORK_NODE に設定されていて、*auto_act_supp* が AP_YES に設定されている場合、このパラメーターは AP_BRNN_REQUIRED または AP_BRNN_PROHIBITED に設定しなければなりません。

def_data.pu_can_send_dddlu_offline

MPC リンクの場合、このパラメーターは予約済みです。

ローカル PU がホストに NMVT (電源オフ) メッセージを送信するかどうかを指定します。ホスト・システムが DDDL (従属 LU の動的定義) をサポートしている場合、Communications Server for Linux は、動的に定義され

た LU を使用し終わったときに、NMVT (電源オフ) をホストへ送信します。これによって、ホストは、定義が不要になった際にその定義を除去することによって、リソースを節約することができます。

このパラメーターを使用するのは、これがホストへのリンクである (*solicit_sscp_sessions* が AP_YES に設定され、*dspu_services* が AP_NONE に設定されていない) 場合のみです。

値は次のいずれかです。

AP_YES ローカル PU は、NMVT (電源オフ) メッセージをホストに送信します。

AP_NO ローカル PU は、NMVT (電源オフ) メッセージをホストに送信しません。

ホストが DDDLU はサポートしていても NMVT (電源オフ) メッセージをサポートしていない場合、このパラメーターは AP_NO に設定しなければなりません。

def_data.link_spec_data_len

リンク固有データの長さ。このデータは、基本構造体に連結されます。

SDLC の リンク固有データ:

mux_info.dlc_type

DLC のタイプ。これは AP_IMPL_SDLC_SL に設定します。

poll_frame

事前活動化ポーリングに使用するフレーム。これは通常 XID であり、ポーリングが DLC ユーザーの制御内にあることを示します。ただし、Communications Server for Linux は以前の 2 次インプリメンテーションと 1 次対話をしているときに、その他のフレームを使用してポーリングすることが必要になる場合があります。可能な値は XID、SNRM です。

QLLC の リンク固有データ:

mux_info.dlc_type

DLC のタイプ。これは AP_IMPL_NLI_QLLC に設定します。

vc_type

LS のバーチャル・サーキット・タイプ。値は次のいずれかです。

VQL_SVC

スイッチド・バーチャル・サーキット

VQL_PVC

パーマネント・バーチャル・サーキット

同じローカル・ノードとリモート・ノード間の SVC および PVC LS の両方を定義すると、SVC LS が初めて開始される場合に予測不能の結果になることがあります (着呼が正しい LS と一致しないことがあるため)。これらの問題を回避するために、同じノード・ペアの間の SVC LS より前に PVC LS を必ず活動化します。

fac_len 後に続く追加の X.25 機能データ (*fac* パラメーターで指定) の長さ。追加データが必要でない場合はゼロを指定します。X.25 ネットワークが機能

ゴシエーションをサポートしていない場合は、ゼロを指定して詳細情報については *fac* パラメーターを確認します。

fac リモート・システムに送信されるコール・パケットに必要な機能データを指定します。使用する X.25 ネットワークの管理者、またはリモート・システムの管理者と相談して、このパラメーターに指定する内容を決めてください。

pvc_id PVC ID。この LS に使用される PVC を識別するための 10 進数を (使用している X.25 プロバイダー・ソフトウェア用に定義された PVC の範囲から) 設定します。上記の *vc_type* が VQL_SVC に設定されている場合、このフィールドは予約済みです。

cud_len 後に続く呼び出し利用者データ (*cud* パラメーターで指定) の長さ。

cud 呼び出し利用者データ: このパラメーターは基礎となる X.25 バーチャル・サーキットで使用されるプロトコルを識別し、*vc_type* パラメーターが VQL_SVC に設定される場合のみ使用されます。たいていのインプリメンテーションでは、これは単一の 16 進バイトで設定する必要があり、呼び出されるノードが 1980 QLLC レベルのサポートを要求するための 0xC3 に設定するか、または 1984 QLLC レベルのサポートを要求するための 0xCB に設定します。一部のリモート・システムでは、追加のバイトが必要になる場合があります。リモート・システムのシステム管理者に確認してください。

トークンリング、イーサネットの DLC 固有データ:

mux_info.dlc_type DLC のタイプ。

値は次のいずれかです。

AP_IMPL_TR_SNAP_LLC2
トークンリング

AP_IMPL_ETHER_SNAP_LLC2
イーサネット

xid_timer リモート・ステーションに接続を試みるときに、XID を再送するまでに必要なタイムアウト。このタイマーは、秒単位で指定します。リモート・ステーションがブリッジで接続した別のトークンリング上にあると、値を大きくしなければならない場合があります。

xid_timer_retry XID の伝送と再伝送が許可される回数。このカウントには、最初の伝送は含まれません。つまり、値 1 は、「1 回伝送した後に 1 回再試行する」ことを示します。リモート・ステーションがブリッジで接続した別のトークンリング上にあると、値を大きくしなければならない場合があります。

test_timer リモート・ステーションへの接続を試行するときに TEST フレームが再送されるまでに必要なタイムアウト。このタイマーは、秒単位で指定します。リモート・ステーションがブリッジで接続した別のトークンリング上にあると、値を大きくしなければならない場合があります。

test_timer_retry

TEST フレームの伝送および再送の許可されている回数。このカウントには、最初の伝送は含まれません。つまり、値 1 は、「1 回伝送した後に 1 回再試行する」ことを示します。リモート・ステーションがブリッジで接続した別のトークンリング上にあると、値を大きくしなければならない場合があります。

ack_timeout

通知タイムアウト: その時間内に、隣接リンク・ステーションに送信された I フレームに対する応答が受信される必要があるミリ秒単位の時間。

p_bit_timeout

ポーリング・ビット・タイムアウト: その時間内に、隣接リンク・ステーションにポーリング・ビット・セットとともに送信されたフレームに対する応答が受信される必要があるミリ秒単位の時間。

t2_timeout

ローカル・ステーションが、受信した I フレームに対する応答を送信するまでに待機できる最大時間 (ミリ秒)。タイムアウトを長くすると、ローカル・ステーションが複数の I フレームに単一の RR で応答できるため、通知トラフィックを削減できます。

rej_timeout

リジェクト・タイムアウト: その時間内に、隣接リンク・ステーションに送信された REJ フレームに対する応答が受信される必要がある秒単位の時間。

busy_state_timeout

ローカル・ステーションが、ビジー状態 (RNR) がクリアされたことを示す隣接リンク・ステーションからの通知を待機する時間 (秒)。

idle_timeout

アイドル・タイムアウト: 完全に非アクティブ・ラインを検出するために使用されるタイマー。この時間内に何も受信しなかった場合、その回線はアイドル状態とみなされます。このタイマーは、秒単位で指定します。

max_retry

応答やビジー状態のクリアを待機しているときのローカル・ステーションの再試行の最大回数。

マルチパス・チャンネル (MPC) のリンク固有データ、System z 上の Communications Server for Linux のみ:

chnl_link_spec_data.mux_info.dlc_type

DLC のタイプ。これは AP_IMPL_MPC_GDLC に設定する必要があります。

Enterprise Extender (HPR/IP) のリンク固有データ:

ipdlc_link_spec_data.mux_info.dlc_type

DLC のタイプ。これは AP_IP に設定します。

ipdlc_link_spec_data.ack_timeout

肯定応答タイマー (T1 タイマーとも呼ぶ) の期間: 隣接リンク・ステーショ

ンに送信したコマンド・フレームの応答を受信するまでの制限時間 (ミリ秒単位)。この時間内に応答を受信しなかった場合、重複フレームが送信されます。

このパラメーターの値を小さくすると、脱落パケットは素早く検出されますが、ネットワーク・トラフィックは増加することになります。

0 ~ 65,535 の範囲で値を指定します。このパラメーターは、予想されるネットワーク待ち時間の 2 倍より大きい値に設定してください。通常の場合は 2000 ms (ミリ秒) です。

ipdlc_link_spec_data.max_retry

ローカル・ステーションがコマンド・フレームの送信を再試行する最大回数。この再試行カウントを超えても応答を受信していない場合、リンクは失敗したと見なされます。

このパラメーターの値を小さくすると、リンク障害は素早く検出されますが、いくつかのパケットが脱落した際に、不要なリンク障害を報告する原因になることがあります。

0 ~ 255 の範囲で値を指定します。通常の場合は 10 回です。

ipdlc_link_spec_data.liveness_timeout

活性タイマー (TL タイマーとも呼ばれる) の期間: リモート・ステーションが現在も活動状態であるという形跡がなくなったとき、リンクを活動状態のままにしておく時間 (ミリ秒単位)。

このパラメーターの値を小さくすると、リンク障害は素早く検出されますが、アイドル活動リンクのネットワーク・トラフィックが増加することがあります。

1 ~ 65535 ミリ秒の範囲で値を指定します。通常の場合は 10000 (10 秒) です。

ipdlc_link_spec_data.short_hold_mode

最後にデータが伝送されてから、リモート・システムが現在も活動状態であるという形跡がなくなった場合にのみ、活性プロトコルを実行するかどうか (AP_YES または AP_NO) を指定します。

このパラメーターを AP_YES に設定すると、リンクを活動状態のままにし、不要なデータ・トラフィックがなくてもアイドル状態にすることができますが、ローカル・ステーションがデータの送信を試みるまで、リンク障害は検出されません。通常、このパラメーターは、AP_NO に設定します。

ipdlc_link_spec_data.remote_hostname

このリンクの宛先ノードのリモート・ホスト名。指定できるのは、以下のいずれかです。

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

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

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_CANT_MODIFY_PORT_NAME

`ls_name` パラメーターが既存の LS の名前に一致しましたが、`port_name` パラメーターが既存の定義に一致しませんでした。既存の LS の定義を変更するときに、ポート名を変更することはできません。

AP_DEF_LINK_INVALID_SECURITY

`tg_chars.security` パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_AUTO_ACT_SUPP

`auto_act_supp` パラメーターが、有効な値に設定されていなかったか、`cp_cp_sess_support` を `AP_YES` に設定しているときに `AP_YES` に設定されていました。

AP_INVALID_CP_NAME

`adj_cp_name` パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

AP_INVALID_LIMITED_RESOURCE

`limited_resource` パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LINK_NAME

`ls_name` パラメーターに、有効でない文字が入っていました。

AP_INVALID_LS_ROLE

`ls_role` パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_NODE_TYPE

`adj_cp_type` パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_PORT_NAME

port_name パラメーターが、定義されているどのポート名にも一致しませんでした。

AP_INVALID_PU_NAME

pu_name パラメーターが定義済みの PU の名前と一致しないか、または定義済み LS で新規の値に設定されています。

AP_INVALID_DSPU_NAME

dspu_name パラメーターが定義済みの PU の名前と一致しないか、または定義済み LS で新規の値に設定されています。

AP_INVALID_DSPU_SERVICES

dspu_services パラメーターが、有効な値に設定されていなかったか、予期されていないときに設定されていました。

AP_INVALID_SOLICIT_SSCP_SESS

solicit_sscp_sess パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_TARGET_PACING_CNT

target_pacing_count パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_DLUS_NAME

dlus_name パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

AP_INVALID_BKUP_DLUS_NAME

bkup_dlus_name パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

AP_INVALID_TG_NUMBER

指定した TG 番号が、有効な範囲内にありませんでした。

AP_MISSING_CP_NAME

TG 番号は定義されましたが、CP 名が指定されませんでした。

AP_MISSING_CP_TYPE

TG 番号は定義されましたが、CP タイプが指定されませんでした。

AP_MISSING_TG_NUMBER

リンクは自動的に活動化されるよう定義されましたが、TG 番号が指定されませんでした。

AP_PARALLEL_TGS_NOT_SUPPORTED

このノードは、同じ隣接ノードとの間に定義された複数の LS をサポートできません。

AP_INVALID_DLUS_RETRY_LIMIT

dlus_retry_limit に指定した値が、有効ではありませんでした。

AP_INVALID_DLUS_RETRY_TIMEOUT

dlus_retry_timeout に指定した値が、有効ではありませんでした。

AP_INVALID_LS_ROLE

ls_role パラメーターに指定した値が、有効ではありませんでした。

AP_INVALID_NODE_TYPE_FOR_HPR

adj_cp_type パラメーターに指定したノード・タイプは、HPR をサポートしていません。

AP_INVALID_BTU_SIZE

max_send_btu_size パラメーターに指定した値が、有効ではありませんでした。

AP_INVALID_MAX_IFRM_RCVD

max_ifrm_rcvd パラメーターに指定した値が、有効ではありませんでした。

AP_UNKNOWN_IP_HOST

この値は Enterprise Extender (HPR/IP) リンクだけに適用されます。
remote_hostname パラメーターに指定したストリングが、有効な IP アドレスに解決できませんでした。

AP_INVALID_IP_VERSION

この値は Enterprise Extender (HPR/IP) リンクだけに適用されます。
ip_version パラメーターに指定した値が所有している IP ポートに指定された値と一致しませんでした。

AP_INVALID_BRANCH_LINK_TYPE

branch_link_type パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_BRNN_SUPPORT

adj_brnn_cp_support パラメーターが、有効な値に設定されていませんでした。

AP_BRNN_SUPPORT_MISSING

adj_brnn_cp_support パラメーターが AP_BRNN_ALLOWED に設定されています。この値は、隣接ノードがネットワーク・ノードであり、*auto_act_supp* が AP_YES に設定されているため無効です。

AP_INVALID_UPLINK

branch_link_type パラメーターが AP_UPLINK に設定されていますが、ローカル・ノードと隣接ノードの間の既存 LS の定義はこれをダウンリンクと指定しています。分岐リンク・タイプは、同じ 2 つのノード間におけるすべての LS で同じでなくてはなりません。

AP_INVALID_DOWNLINK

branch_link_type パラメーターが AP_DOWNLINK に設定されていますが、ローカル・ノードと隣接ノードの間の既存 LS の定義はこれをアップリンクと指定しています。分岐リンク・タイプは、同じ 2 つのノード間におけるすべての LS で同じでなくてはなりません。

AP_INVALID_LINK_SPEC_FORMAT

予約済みパラメーターが、ゼロ以外の値に設定されました。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_STATE_CHECK`

secondary_rc

値は次のいずれかです。

AP_DUPLICATE_CP_NAME

adj_cp_name パラメーターで指定した CP 名へのリンクが、既に定義されていました。

AP_DUPLICATE_DEST_ADDR

address パラメーターで指定した宛先アドレスへのリンクが、既に定義されていました。

LLC2 リンク・タイプの場合: *mac_address* パラメーターと *lsap_address* パラメーターを組み合わせで指定した宛先アドレスへのリンクが、既に定義されていました。

AP_DUPLICATE_ADJ_NODE_ID

adj_node_id (隣接ノードのノード ID) が、別のリンク・ステーション内で既に定義されています。

AP_INVALID_LINK_NAME

ls_name パラメーターで指定したリンク・ステーション値が有効ではありませんでした。

AP_INVALID_NUM_LS_SPECIFIED

指定したリンク・ステーションの数が有効ではありませんでした。

AP_LOCAL_CP_NAME

adj_cp_name パラメーターに指定した名前が、ローカル CP 名と同じでした。

AP_LS_ACTIVE

ls_name パラメーターで指定したリンク・ステーションは、現在活動状態です。

AP_PU_ALREADY_DEFINED

pu_name パラメーターで指定した PU は、既に定義されています。

AP_DSPU_ALREADY_DEFINED

dspu_name パラメーターで指定したダウンストリーム PU は、既に定義されています。

AP_DSPU_SERVICES_NOT_SUPPORTED

dspu_services パラメーターで `AP_PU_CONCENTRATION` または `AP_DLUR` を指定しましたが、ノードではそれをサポートしていません。

AP_DUPLICATE_TG_NUMBER

tg_number パラメーターで指定した TG 番号は、既に定義されています。

AP_TG_NUMBER_IN_USE

tg_number パラメーターに指定した TG 番号は、既に別の LS によって使用されています。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

MAC アドレス内のビット順序

イーサネットの LAN では、トークンリングの場合と異なる MAC アドレス表記が使用されます。イーサネットでは、アドレスの各バイトのビット順序が、トークンリングの順序の逆になっています。通常の場合、ローカル・ノードとリモート・ノードは同じ LAN 上にあるか、ブリッジで接続した同じタイプの LAN 上にあります。その場合、どちらのノードも同じ MAC アドレス表記を使用するため、変換する必要はありません。

2 つのノードが、ブリッジで接続した異なるタイプ (一方がイーサネットで、他方がトークンリング) の LAN 上にある場合は、通常、リモート MAC アドレスを指定するときに、アドレスの各バイトのビット順序を逆にする必要があります。そのためには、次のステップを実行します。

1. MAC アドレスを 6 バイトとしてリスト出力し、各バイトを 2 桁の 16 進数で表記します。
2. MAC アドレスを 6 バイトとしてリスト出力し、各バイトを 2 桁の 16 進数で表記します。
3. それぞれの桁を次に示すように変換します。

0 → 0	8 → 1
1 → 8	9 → 9
2 → 4	A → 5
3 → C	B → D
4 → 2	C → 3
5 → A	D → B
6 → 6	E → 7
7 → E	F → F

Example of Bit Ordering in a MAC Address

元のアドレス	1A	2B	3C	4D	5E	6F
桁のスワップ	A1	B2	C3	D4	E5	F6
桁の変換 (元のアドレスのビット順序を逆にした形式)	58	D4	3C	B2	7A	F6

モデム制御文字

SDLC の場合、*hmod_data* パラメーターに 1 つ以上の印刷不可能な制御文字を使用する場合は、164 ページの表 2 のリストのように 16 進値で制御文字を指定しま

DEFINE_LS

す。

表2. モデム制御文字のエスケープ・シーケンス

エスケープ・シーケンス	10 進値	16 進値
NUL	0	0x00
SOH	1	0x01
STX	2	0x02
ETX	3	0x03
EOT	4	0x04
ENQ	5	0x05
ACK	6	0x06
BEL	7	0x07
BS	8	0x08
HT	9	0x09
LF	10	0x0A
VT	11	0x0B
FF	12	0x0C
CR	13	0x0D
SO	14	0x0E
SI	15	0x0F
DLE	16	0x10
DC1	17	0x11
DC2	18	0x12
DC3	19	0x13
DC4	20	0x14
NAK	21	0x15
SYN	22	0x16
ETB	23	0x17
CAN	24	0x18
EM	25	0x19
SUB	26	0x1A
ESC	27	0x1B
FS	28	0x1C
GS	29	0x1D
RS	30	0x1E
US	31	0x1F
SP	32	0x20
DEL	127	0x7F

DEFINE_LS_ROUTING

DEFINE_LS_ROUTING verb は、リンク・ステーションを使用するパートナー LU の位置を定義します。

注: Enterprise Extender (HPR/IP) リンク・ステーションの場合、
DEFINE_LS_ROUTING を使用することはできません。これは、このリンク・タイプ上のトラフィックが RTP (高速トランスポート・プロトコル) 接続を介しフローしなければならないという理由によります。RTP 接続は、特定のリンクに固定しておらず、異なるパスに切り替えが可能です。

VCB 構造体

```
typedef struct define_ls_routing
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  lu_name[8];     /* LU Name                    */
    unsigned char  lu_alias[8];   /* reserved                    */
    unsigned char  fq_partner_lu[17]; /* partner lu name            */
    unsigned char  wildcard_fqplu; /* wildcard partner LU flag    */
    unsigned char  ls_name[8];    /* link to use                 */
    unsigned char  reserv3[2];    /* reserved                     */
} DEFINE_LS_ROUTING;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_LS_ROUTING

lu_name

ls_name パラメーターで指定したリンクを介して (*fq_partner_lu* パラメーターで指定した) パートナー LU と通信するローカル LU の名前。これは 8 バイトからなるタイプ A の文字ストリングです。

fq_partner_lu

ls_name パラメーターで指定したリンクを介して (*lu_name* パラメーターで指定した) ローカル LU が通信するパートナー LU の完全修飾名。1 ~ 8 文字のネットワーク名、1 つのピリオド、1 ~ 8 文字のパートナー LU 名の順序で 3 ~ 17 文字を指定します。

名前の一部のみを指定して、*wildcard_fqplu* パラメーターを AP_YES に設定することにより、部分的または完全なワイルドカードを使用したパートナー LU 名を指定できます。次に例を示します。

- APPN.NEW は APPN.NEW1、APPN.NEWLU などと一致します。
- APPN. は、その LU 名とは無関係に、APPN というネットワーク名を持つすべて LU と一致します。
- APPN は、APPN で始まるネットワーク名を持つすべての LU (APPN.NEW1、APPNNEW.LUTWO など) と一致します。

すべてのパートナー LU に同じリンクを使用してアクセスするように完全ワイルドカード・エンタリーを指定するには、*wildcard_fqplu* を AP_YES に設定し、*fq_partner_lu* をヌル・ストリングに設定します。

wildcard_fqplu

fq_partner_lu パラメーターに、完全ワイルドカードまたは部分ワイルドカードが含まれているかどうかを示すワイルドカード・パートナー LU フラグ。値は次のいずれかです。

AP_YES *fq_partner_lu* パラメーターには、ワイルドカード・エンタリーが含まれています。

AP_NO *fq_partner_lu* パラメーターには、ワイルドカード・エンタリーは含まれていません。

ls_name

(*lu_name* パラメーターで指定した) ローカル LU と (*fq_partner_lu* パラメーターで指定した) パートナー LU の間の通信に使用するリンク・ステーションの名前。 1 ~ 8 文字のローカル・システム上で表示可能な文字を指定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_LU_NAME
lu_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_PLU_NAME
fq_partner_lu パラメーターに有効でない文字が入っていたか、名前が完全修飾名ではありませんでした。

AP_INVALID_WILDCARD_NAME
wildcard_fqplu パラメーターを指定しましたが、*fq_partner_lu* パラメーターが有効なワイルドカード名ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_STATE_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_LU_NAME
lu_name パラメーターで識別したローカル LU がありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_LU62_TIMEOUT

DEFINE_LU62_TIMEOUT verb は、未使用の LU 6.2 セッションのタイムアウト期間を定義します。それぞれのタイムアウトは、指定したリソース・タイプとリソース名についてのタイムアウトです。既に定義されているリソース・タイプと名前の対に DEFINE_* verb を発行した場合、そのコマンドは以前の定義を上書きします。新しいタイムアウト期間は、定義が変更されたあとに活動化されたセッションについてのみ使用されます。

あるセッションに関連する複数のタイムアウト期間を定義した場合は、その最短期間が適用されます。

VCB 構造体

```
typedef struct define_lu62_timeout
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;        /* secondary return code */
    unsigned char  resource_type;        /* resource type */
    unsigned char  resource_name[17];    /* resource name */
    AP_UINT16      timeout;              /* timeout */
} DEFINE_LU62_TIMEOUT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_LU62_TIMEOUT

resource_type

定義するタイムアウトのタイプを指定します。値は次のいずれかです。

AP_GLOBAL_TIMEOUT

ローカル・ノードのすべての LU 6.2 セッションにタイムアウトが適用されます。*resource_name* パラメーターは、すべてゼロに設定してください。

AP_LOCAL_LU_TIMEOUT

resource_name パラメーターで指定したローカル LU のすべての LU 6.2 セッションにタイムアウトが適用されます。

AP_PARTNER_LU_TIMEOUT

resource_name パラメーターで指定したパートナー LU のすべての LU 6.2 セッションにタイムアウトが適用されます。

AP_MODE_TIMEOUT

resource_name パラメーターで指定したモード上のすべての LU 6.2 セッションにタイムアウトを適用します。

resource_name

照会するリソースの名前。この値は次のいずれかです。

- *resource_type* を AP_GLOBAL_TIMEOUT に設定した場合は、このパラメーターを指定しないでください。
- *resource_type* を AP_LOCAL_LU_TIMEOUT に設定した場合は、ローカル LU 名として、1 ~ 8 文字のローカル・システム上で表示可能なタイプ A の文字を指定してください。
- *resource_type* を AP_PARTNER_LU_TIMEOUT に設定した場合は、パートナー LU の完全修飾名を指定するため、1 ~ 8 文字のネットワーク名、1 つのピリオド、1 ~ 8 文字のパートナー LU 名の順序で、17 文字からなるローカルに表示可能なタイプ A の文字を指定してください。
- *resource_type* を AP_MODE_TIMEOUT に設定した場合は、モード名として、1 ~ 8 文字のローカル・システム上で表示可能なタイプ A の文字を指定してください。

timeout タイムアウト期間 (秒単位)。値 0 (ゼロ) は、セッションをすぐに解放することを示します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_RESOURCE_TYPE
定義したタイムアウトのタイプが有効ではありませんでした。

AP_INVALID_LU_NAME
resource_type パラメーターに、有効でない LU 名を指定しました。

AP_INVALID_PARTNER_LU
resource_type パラメーターに、有効でないパートナー LU 名を指定しました。

AP_INVALID_MODE_NAME
resource_type パラメーターに、有効でないモード名を指定しました。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_LU_0_TO_3

DEFINE_LU_0_TO_3 verb は、3270 エミュレーションまたは LUA に使用する LU (タイプ 0、1、2、3 のいずれか) を定義し、オプションでその LU を LU プールに割り当てます。

この verb を使用して既存の LU を変更する場合は、*description*、*priority*、*lu_model* の各パラメーターのみを変更できます。その他のすべてのパラメーターは、既存の値に設定しなければなりません。

VCB 構造体

```
typedef struct define_lu_0_to_3
{
    AP_UINT16          opcode;                /* verb operation code */
    unsigned char      reserv2;              /* reserved */
    unsigned char      format;              /* reserved */
    AP_UINT16          primary_rc;          /* primary return code */
    AP_UINT32          secondary_rc;        /* secondary return code */
    unsigned char      lu_name[8];         /* LU name */
    LU_0_TO_3_DEF_DATA def_data;          /* defined data */
} DEFINE_LU_0_TO_3;

typedef struct lu_0_to_3_def_data
{
    unsigned char      description[32];      /* resource description */
    unsigned char      reserv1[16];        /* reserved */
    unsigned char      nau_address;        /* LU NAU address */
    unsigned char      pool_name[8];       /* LU Pool name */
    unsigned char      pu_name[8];        /* PU name */
    unsigned char      priority;          /* LU priority */
    unsigned char      lu_model;          /* LU model (type) */
    unsigned char      sscp_id[6];        /* SSCP ID */
    AP_UINT16          timeout;           /* Timeout */
    unsigned char      app_spec_def_data[16]; /* reserved */
    unsigned char      model_name[7];     /* reserved */
    unsigned char      term_method;       /* session termination type */
    unsigned char      disconnect_on_unbind; /* disconnect on UNBIND flag */
    unsigned char      reserv3[15];      /* reserved */
} LU_0_TO_3_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_LU_0_TO_3

lu_name

ローカル LU の名前。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtring で、右側に EBCDIC のスペースを入れます。

def_data.description

LU を記述したヌルで終了するテキスト・String (0 ~ 31 文字のあとにヌル文字を続けたString)。このStringは情報専用です。このス

トリングはノードの構成ファイル内に保管され、`QUERY_LU_0_TO_3 verb` で戻されますが、`Communications Server for Linux` では他の用途に使用されません。

def_data.nau_address

LU のネットワーク・アクセス可能単位アドレス。これは、1 ~ 255 の範囲の数値です。

def_data.pool_name

この LU が所属するプールの名前。これはタイプ A の EBCDIC スtring で、名前が 8 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。指定した名前のプールがまだ定義されていない場合、

`Communications Server for Linux` はこの名前が付いた新しいプールを追加し、LU をそのプールに割り当てます。

LU がプールに所属しない場合は、このフィールドを 8 桁の 2 進ゼロに設定します。

def_data.pu_name

この LU が使用する PU の名前 (`DEFINE_LS verb` で指定したもの)。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtring で、名前が 8 バイト未満の場合は、右側に EBCDIC のスペースを入れます。

def_data.priority

ホストへ送信する場合の LU 優先順位。値は次のいずれかです。

AP_NETWORK

AP_HIGH

AP_MEDIUM

AP_LOW

def_data.lu_model

LU のタイプ。値は次のいずれかです。

AP_3270_DISPLAY_MODEL_2

AP_3270_DISPLAY_MODEL_3

AP_3270_DISPLAY_MODEL_4

AP_3270_DISPLAY_MODEL_5

AP_PRINTER

AP_SCS_PRINTER

AP_RJE_WKSTN

AP_UNKNOWN (LU タイプは、ホストへのセッションが確立されたときに決定されます。)

3270 エミュレーションに LU を使用していない場合は、明示的な LU タイプを指定する必要はなく、このパラメーターは AP_UNKNOWN に設定します。

ホスト・システムが DDDL (従属 LU の動的定義) をサポートしている場合、`Communications Server for Linux` は、ホストへの通信リンクが確立されるたびに、そのホストで LU を動的に定義します。定義される LU モデ

ル・タイプをダウンストリーム TN3270 クライアントの要求するタイプと一致させたい場合は、このパラメーターを AP_UNKNOWN に設定します。ユーザーが指定する値に応じて、Communications Server for Linux は、標準の VTAM テーブルで使用される値と一致する、以下のストリングのいずれかを DDDL U NMVT のホストへ送信します。

AP_3270_DISPLAY_MODEL_2 の場合は 3270002

AP_3270_DISPLAY_MODEL_3 の場合は 3270003

AP_3270_DISPLAY_MODEL_4 の場合は 3270004

AP_3270_DISPLAY_MODEL_5 の場合は 3270005

AP_PRINTER の場合は 3270DSC

AP_SCS_PRINTER の場合は 3270SCS

AP_RJE_WKSTN の場合は 3270000

TN3270 クライアントで AP_UNKNOWN の場合は 327000n、ここで n はクライアントが提供する型式番号 (2 ~ 5) です。

LUA クライアントで AP_UNKNOWN の場合は 3270000

ホストが DDDL U をサポートしていない場合、この LU はホスト構成に含まれていなければなりません。

def_data.sscp_id

この LU の活動化が許可される SSCP の ID を指定します。すべての SSCP がこの LU を活動化できる場合は、このパラメーターのすべてのビットを 0 (ゼロ) に設定します。特定の SSCP のみがこの LU を活動化できる場合は、このパラメーターの最初の 4 バイトを 0x05000000 に設定し、最後の 2 バイトを、この LU の活動化を許可された SSCP の SSCP ID に設定します。

def_data.timeout

指定した LU のタイムアウト (秒単位)。このタイムアウトをゼロ以外の値に設定し、この LU のユーザーがセッション非活動タイムアウトをサポートする場合、LU は、PLU-SLU セッションが指定した期間に非活動状態で、次のいずれかの条件が当てはまるときに非活動化されます。

- セッションが、限定リソース・リンクを通る。
- セッションが再度使用される前に、別のアプリケーションがその LU の使用を要求した。

このタイムアウトを 0 (ゼロ) に設定した場合、LU は非活動化されません。

セッション非活動タイムアウトのサポートは、LU (3270 エミュレーション・プログラムなど) を使用しているアプリケーションによって決まります。SNA ゲートウェイが LU を使用している場合、セッション非活動タイムアウトがサポートされるのは DEFINE_DOWNSTREAM_LU verb で *allow_timeout* が指定されている場合のみです。

def_data.term_method

このパラメーターは、この LU からホストへの PLU-SLU セッションを Communications Server for Linux が終了させる方法を指定します。値は次のいずれかです。

AP_USE_NODE_DEFAULT

DEFINE_NODE の *send_term_self* パラメーターで指定された、ノードのデフォルト終了方法を使用します。

AP_SEND_UNBIND

UNBIND を送信してセッションを終了します。

AP_SEND_TERM_SELF

TERM_SELF を送信してセッションを終了します。

def_data.disconnect_on_unbind

このパラメーターは、この LU が TN3270 クライアントによって使用される場合のみ適用されます。これは、ホストが VTAM MSG10 を表示するか、またはホスト・セッション・マネージャーに戻る代わりに UNBIND を送信するときに、セッションを終了させるかどうかを指定します。値は次のいずれかです。

AP_YES ホストがタイプ 2 (BIND が用意されている) でない UNBIND を送信する場合は、セッションを終了させます。

AP_NO ホストが UNBIND を送信してもセッションを終了させません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_LU_NAME

lu_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_POOL_NAME

pool_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_NAU_ADDRESS

nau_address パラメーターが、許可された範囲内にありませんでした。

AP_INVALID_PRIORITY

priority パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_TERM_METHOD

term_method パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PU_NAME

pu_name パラメーターが、有効ではありませんでした。

AP_PU_NOT_DEFINED

pu_name パラメーターが、定義されているどの PU 名にも一致しませんでした。

AP_INVALID_PU_TYPE

pu_name パラメーターで指定した PU は、ホスト PU ではありません。

AP_LU_NAME_POOL_NAME_CLASH

LU 名が LU プールの名前と矛盾しています。

AP_LU_ALREADY_DEFINED

指定した名前の LU は、既に定義されています。

AP_LU_NAU_ADDR_ALREADY_DEFD

指定した NAU アドレスの LU は、既に定義されています。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_LU_0_TO_3_RANGE

DEFINE_LU_0_TO_3 verb は、3270 エミュレーションまたは LUA に使用する LU (タイプ 0、1、2、3 のいずれか) の範囲を定義し、オプションでそれらの LU を LU プールに割り当てます。この verb を使用して既存の LU を変更することはできません。

この verb に指定するパラメーターには、新しい LU のベース名と NAU アドレスの範囲が含まれます。新しい LU の名前は、そのベース名と NAU アドレスを結合して生成されます。たとえば、LUNME というベース名が 11 ~ 14 の NAU 範囲と結合して、LUNME011、LUNME012、LUNME013、LUNME014 という LU が定義されます。

VCB 構造体

```
typedef struct define_lu_0_to_3_range
{
    AP_UINT16      opcode;           /* verb operation code */
    unsigned char  reserv2;         /* reserved */
    unsigned char  format;         /* reserved */
    AP_UINT16      primary_rc;     /* primary return code */
    AP_UINT32      secondary_rc;   /* secondary return code */
    unsigned char  base_name[6];   /* Base name */
    unsigned char  description[32]; /* resource description */
    unsigned char  reserv1[16];    /* reserved */
    unsigned char  min_nau;        /* Minimum NAU address */
    unsigned char  max_nau;        /* Maximum NAU address */
    unsigned char  pool_name[8];   /* LU Pool name */
    unsigned char  pu_name[8];    /* PU name */
    unsigned char  priority;       /* LU priority */
    unsigned char  lu_model;       /* LU model (type) */
    unsigned char  sscp_id[6];     /* SSCP ID */
    AP_UINT16      timeout;        /* Timeout */
    unsigned char  app_spec_def_data[16]; /* reserved */
    unsigned char  reserv3[7];     /* reserved */
    unsigned char  name_attributes; /* Extension type */
    unsigned char  base_number;    /* First extension number */
    unsigned char  term_method;    /* session termination type */
    unsigned char  disconnect_on_unbind; /* disconnect on UNBIND flag */
    unsigned char  reserv4[13];    /* reserved */
} DEFINE_LU_0_TO_3_RANGE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_LU_0_TO_3_RANGE

base_name

新しい LU の名前のベース名。これは 6 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、ベース名が 6 文字に満たない場合、右側に EBCDIC のスペースを入れます。

- *name_attributes* パラメーターが AP_USE_HEX_IN_NAME に設定されている場合、この名前は 6 文字までとなることがあります。Communications Server for Linux は、この名前に 2 桁の 16 進数を付加し、各 LU の LU 名を生成します (*base_number* パラメーターに指定した基本数から始まります)。
- 上記以外の場合、この名前は 5 文字までとなることがあります。Communications Server for Linux は、この名前に 3 桁の 10 進数を付加し、各 LU の LU 名を生成します (NAU アドレスまたは *name_attributes* パラメーターに指定した定義済み基本数から取られます)。

description

LU を記述したヌルで終了するテキスト・String (0 ~ 31 文字のあとにヌル文字を続けたString)。範囲内の各 LU に同じStringが使用されます。このStringは情報専用です。このStringはノードの構成ファイル内に保管され、QUERY_LU_0_TO_3 verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

min_nau

最初の LU の NAU アドレス (範囲は 1 ~ 255)。

max_nau

最後の LU の NAU アドレス (範囲は 1 ~ 255)。

pool_name

これらの LU が所属するプールの名前。これは 8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。指定した名前前のプールがまだ定義されていない場合、Communications Server for Linux はその名前が付いた新しいプールを追加し、LU をそのプールに割り当てます。

LU がプールに所属しない場合は、このフィールドを 8 桁の 2 進ゼロに設定します。

pu_name

これらの LU が使用する PU の (DEFINE_LS verb で指定した) 名前。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC ストリングで、右側に EBCDIC のスペースを入れます。

priority

ホストへ送信する場合の LU 優先順位。値は次のいずれかです。

AP_NETWORK

AP_HIGH

AP_MEDIUM

AP_LOW

lu_model

LU のタイプ。値は次のいずれかです。

AP_3270_DISPLAY_MODEL_2

AP_3270_DISPLAY_MODEL_3

AP_3270_DISPLAY_MODEL_4

AP_3270_DISPLAY_MODEL_5

AP_PRINTER

AP_SSCP_PRINTER

AP_RJE_WKSTN

AP_UNKNOWN (LU タイプは、ホストへのセッションが確立されたときに決定されます。)

3270 エミュレーションに LU を使用していない場合は、明示的な LU タイプを指定する必要はなく、このパラメーターは AP_UNKNOWN に設定します。

ホスト・システムが DDDL (従属 LU の動的定義) をサポートしていて、このパラメーターを AP_UNKNOWN 以外の値に設定した場合、Communications Server for Linux は、ホストへの通信リンクが確立されるときに、そのホストで LU を動的に定義します。ユーザーが指定する値に応じて、

Communications Server for Linux は、標準の VTAM テーブルで使用される値と一致する、以下のストリングのいずれかを DDDL NMVT のホストへ送信します。

AP_3270_DISPLAY_MODEL_2 の場合は 3270002

DEFINE_LU_0_TO_3_RANGE

AP_3270_DISPLAY_MODEL_3 の場合は 3270003

AP_3270_DISPLAY_MODEL_4 の場合は 3270004

AP_3270_DISPLAY_MODEL_5 の場合は 3270005

AP_PRINTER の場合は 3270DSC

AP_SCS_PRINTER の場合は 3270SCS

AP_RJE_WKSTN の場合は 3270000

TN3270 クライアントで AP_UNKNOWN の場合は 327000n、ここで n はクライアントが提供する型式番号 (2 ~ 5) です。

LUA クライアントで AP_UNKNOWN の場合は 3270000

ホストが DDDLU をサポートしていない場合、またはこのパラメーターを AP_UNKNOWN に設定する場合、この LU はホスト構成に含まれていなければなりません。

sscp_id この LU の活動化が許可される SSCP の ID を指定します。0 ~ 65,535 の範囲で値を指定します。このパラメーターのすべてのビットを 0 (ゼロ) に設定した場合は、すべての SSCP がこの LU を活動化できます。

timeout 指定した LU のタイムアウト (秒単位)。このタイムアウトをゼロ以外の値に設定し、この LU のユーザーがセッション非活動タイムアウトをサポートする場合、LU は、PLU-SLU セッションが指定した期間に非活動状態で、次のいずれかの条件が当てはまるときに非活動化されます。

- セッションが、限定リソース・リンクを通る。
- セッションが再度使用される前に、別のアプリケーションがその LU の使用を要求した。

このタイムアウトを 0 (ゼロ) に設定した場合、LU は非活動化されません。

セッション非活動タイムアウトのサポートは、LU (3270 エミュレーション・プログラムなど) を使用しているアプリケーションによって決まります。SNA ゲートウェイが LU を使用している場合、セッション非活動タイムアウトがサポートされるのは DEFINE_DOWNSTREAM_LU verb で *allow_timeout* が指定されている場合のみです。

name_attributes

定義する LU の属性。値は次のいずれかです。

AP_NONE

LU 名には NAU 番号に対応する番号があります。それらの番号は 10 進数で指定され、*base_name* パラメーターは必ず 5 文字にします。

AP_USE_BASE_NUMBER

この範囲内の LU の命名を *base_number* パラメーターで指定した値から開始します。

AP_USE_HEX_IN_NAME | AP_USE_BASE_NUMBER

この範囲内の LU の命名を *base_number* パラメーターで指定した値から開始し、その LU 名に 10 進ではなく 16 進の拡張子を付けます。この値を指定した場合は、*base_name* パラメーターに 6 文字まで指定できます。

base_number

name_attributes パラメーターに AP_USE_BASE_NUMBER を指定した場合は、範囲内の LU の命名を開始する番号を指定します。この値は、*min_nau* パラメーター値の代わりに使用されます。

term_method

このパラメーターは、PLU-SLU セッションの LU の 1 つからからホストへの PLU-SLU セッションを Communications Server for Linux が終了させる方法を指定します。値は次のいずれかです。

AP_USE_NODE_DEFAULT

DEFINE_NODE の *send_term_self* パラメーターで指定された、ノードのデフォルト終了方法を使用します。

AP_SEND_UNBIND

UNBIND を送信してセッションを終了します。

AP_SEND_TERM_SELF

TERM_SELF を送信してセッションを終了します。

disconnect_on_unbind

このパラメーターは、この範囲の LU が TN3270 クライアントによって使用される場合のみ適用されます。これは、ホストが VTAM MSG10 を表示するか、またはホスト・セッション・マネージャーに戻る代わりに UNBIND を送信するときに、セッションを終了させるかどうかを指定します。値は次のいずれかです。

AP_YES ホストがタイプ 2 (BIND が用意されている) でない UNBIND を送信する場合は、セッションを終了させます。

AP_NO ホストが UNBIND を送信してもセッションを終了させません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

base_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_POOL_NAME

pool_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_NAU_ADDRESS

1 つ以上の LU アドレスが、許可された範囲内にありませんでした。

AP_INVALID_PRIORITY

priority パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_TERM_METHOD

term_method パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PU_NAME

pu_name パラメーターが、有効ではありませんでした。

AP_PU_NOT_DEFINED

pu_name パラメーターが、定義されているどの PU 名にも一致しませんでした。

AP_INVALID_PU_TYPE

pu_name パラメーターで指定した PU は、ホスト PU ではありません。

AP_LU_NAME_POOL_NAME_CLASH

この範囲内の LU 名の 1 つが、LU プールの名前と矛盾します。

AP_LU_ALREADY_DEFINED

この範囲内のいずれかの LU の名前を使用して、既に LU が定義されています。

AP_LU_NAU_ADDR_ALREADY_DEFD

この範囲内のいずれかの LU のアドレスを使用して、既に LU が定義されています。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_LU_LU_PASSWORD

DEFINE_LU_LU_PASSWORD は、ローカル LU とパートナー LU 間のセッション・レベルのセキュリティー検証に使用するパスワードを提供します。

VCB 構造体

```
typedef struct define_lu_lu_password
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;               /* reserved                  */
    AP_UINT16      primary_rc;           /* primary return code      */
    AP_UINT32      secondary_rc;         /* secondary return code    */
    unsigned char  lu_name[8];           /* local LU name            */
    unsigned char  lu_alias[8];          /* local LU alias           */
    unsigned char  fqplu_name[17];        /* fully qualified partner  */
                                                /* LU name                  */
    unsigned char  verification_protocol; /* verification protocol    */
    unsigned char  description[32];      /* resource description     */
    unsigned char  reserv1[16];          /* reserved                  */
    unsigned char  reserv3[8];           /* reserved                  */
    unsigned char  password[8];          /* password                  */
} DEFINE_LU_LU_PASSWORD;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_LU_LU_PASSWORD

lu_name

Communications Server for Linux に対して定義されているローカル LU の LU 名。これは 8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。LU が LU 名でなく LU 別名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

lu_alias

Communications Server for Linux に対して定義されているローカル LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。これは、*lu_name* をゼロに設定した場合にのみ使用します。

CP へ関連付けられている LU (デフォルト LU) を示すには、*lu_name* と *lu_alias* の両方を 8 桁の 2 進ゼロに設定します。

fqplu_name

Communications Server for Linux に対して定義されているパートナー LU の完全修飾 LU 名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

verification_protocol

使用するよう要求される LU-LU 検証プロトコル。値は次のいずれかです。

DEFINE_LU_LU_PASSWORD

AP_BASIC

基本 LU-LU 検証プロトコルを指定します。

AP_ENHANCED

拡張 LU-LU 検証プロトコルを指定します。

AP_EITHER

基本検証プロトコルと拡張を受け入れます。

description

パスワードを記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、`QUERY_LU_LU_PASSWORD verb` で戻されますが、`Communications Server for Linux` では他の用途に使用されません。

password

パスワード。これは 8 バイトの 16 進数ストリングで、すべて空白またはすべてゼロに設定しないでください。これは、リモート・システム上でパートナー LU に構成された同様のパラメーターに一致しなければなりません (ただし、各バイトの最下位ビット (LSB) はセッション・レベルのセキュリティ検証に使用されないため、一致する必要はありません)。

アプリケーションでこのパラメーターに指定したすべての値は、すぐにこのパスワードを暗号化したものに置き換えられます。したがって、`password` パラメーターに指定した値は、決して書き出されません。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、`Communications Server for Linux` は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、`Communications Server for Linux` は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_ALIAS

`lu_alias` パラメーターが、定義されているどの LU 別名とも一致しませんでした。

AP_INVALID_LU_NAME

`lu_name` パラメーターが、定義されているどのローカル LU 名にも一致しませんでした。

AP_INVALID_PLU_NAME

fqplu_name パラメーターが、定義されているどのパートナー LU 名にも一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_LU_POOL

この verb は、LU プールを定義して LU をそのプールに割り当てるために使用するか、既存のプールへ追加 LU を割り当てるために使用します。それらの LU は、プールへ追加する前に定義されていなければなりません。LU を定義するときにプール名を指定することによっても、プールを定義できます。詳しくは、169 ページの『DEFINE_LU_0_TO_3』を参照してください。

この verb を使用し、プールから LU を除去することによって既存のプールを変更することはできません。プールから LU を除去するには、DELETE_LU_POOL verb を使用します。

VCB 構造体

```
typedef struct define_lu_pool
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  pool_name[8];   /* LU pool name             */
    unsigned char  description[32]; /* resource description     */
    unsigned char  reserv1[16];    /* reserved                  */
    unsigned char  reserv3[4];     /* reserved                  */
    AP_UINT16      num_lus;        /* number of LUs to add    */
    unsigned char  lu_names[10][8]; /* LU names                  */
} DEFINE_LU_POOL;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_LU_POOL

pool_name

LU プールの名前。これは 8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。この名前のプールがまだ定義されていなければ、Communications Server for Linux はそのプールを作成します。

description

プールを記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。この

DEFINE_LU_POOL

ストリングはノードの構成ファイル内に保管され、QUERY_LU_POOL verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

num_lus

このプールに追加する LU の数。このパラメーターは、LU を追加せずにプールを定義するためにゼロに設定するか、1 ~ 10 に設定できます。10 個を超える LU が入ったプールを作成するには、同じプール名に対して複数の DEFINE_LU_POOL verb を発行します。

lu_names

プールへ割り当てる LU の名前。これらの LU はそれぞれタイプ 0-3 の LU として、事前に Communications Server for Linux に定義されていなければなりません。それぞれの LU 名は 8 バイトからなる (英字で始まる) タイプ A の EBCDIC ストリングで、右側に EBCDIC のスペースを入れます。

指定した LU が現在、別のプールへ割り当てられている場合、Communications Server for Linux は、これをそのプールから除去し (1 つの LU を複数のプールに入れることはできないため)、この verb で指定したプールへ割り当てます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_LU_NAME

指定した LU 名の 1 つ以上が、定義されている LU 名に一致しませんでした。

AP_INVALID_POOL_NAME

pool_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_NUM_LUS

num_lus パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

AP_LU_NAME_POOL_NAME_CLASH

指定したプール名が LU の名前と矛盾します。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_MODE

DEFINE_MODE `verb` は、モード (セッション・グループが使用するネットワーク特性セット) を定義するか、以前定義したモードを変更します。SNA 定義モード CPSVCMG を変更したり、SNA 定義モード SNASVCMG で使用する COS 名を変更したりすることはできません。

この `verb` を使用して既存のモードを変更した場合、その変更は、変更されたあとにそのモードの使用を開始する新しいローカル LU とパートナー LU の組み合わせに適用されます。ただし、既にそのモードを使用している LU の組み合わせについては、次回にローカルまたはリモートで CNOS コマンドが開始されるまで、その変更が適用されません。

この `verb` を使用して、認識されないモードのマップ先にするデフォルトの COS を指定することもできます。デフォルトの COS を指定しなかった場合、SNA 定義の COS の #CONNECT が使用されます。

VCB 構造体

```
typedef struct define_mode
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;              /* reserved                      */
    unsigned char  format;               /* reserved                      */
    AP_UINT16      primary_rc;           /* primary return code          */
    AP_UINT32      secondary_rc;        /* secondary return code        */
    unsigned char  mode_name[8];        /* mode name                    */
    AP_UINT16      reserv3;              /* reserved                      */
    MODE_CHARS     mode_chars;           /* mode characteristics         */
} DEFINE_MODE;

typedef struct mode_chars
{
    unsigned char  description[32];      /* resource description          */
    unsigned char  reserv2[16];         /* reserved                      */
    AP_UINT16      max_ru_size_upp;     /* maximum RU size upper bound  */
    unsigned char  receive_pacing_win;  /* receive pacing window        */
}
```

DEFINE_MODE

```
unsigned char  default_ru_size;          /* default RU size to maximize */
/* performance */
AP_UINT16     max_neg_sess_lim;         /* maximum negotiable session limit*/
AP_UINT16     plu_mode_session_limit;  /* LU-mode session limit */
AP_UINT16     min_conwin_src;          /* minimum source contention winner*/
/* sessions */
unsigned char  cos_name[8];             /* class of service name */
unsigned char  cryptography;           /* reserved */
unsigned char  compression;            /* data compression supported? */
AP_UINT16     auto_act;                 /* initial auto-activation count */
AP_UINT16     min_conloser_src;        /* min source contention loser */
AP_UINT16     max_ru_size_low;         /* maximum RU size lower bound */
AP_UINT16     max_receive_pacing_win; /* maximum receive pacing window */
unsigned char  max_compress_lvl;       /* max level of data compression */
unsigned char  max_decompress_lvl;     /* max level of data decompression */
unsigned char  comp_in_series;         /* reserved */
unsigned char  reserv4[25];            /* reserved */
} MODE_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_MODE

mode_name

モードの名前。これは 8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。名前の先頭は文字にするか、#INTER のように SNA 定義モードの 1 つとして # にしなければなりません。SNA 定義モードについては、「*Communications Server for Linux 管理ガイド*」を参照してください。

認識されないモード名に使用するデフォルトの COS を指定するには、このパラメーターを 8 桁の 2 進ゼロに設定します。その場合、

mode_chars.cos_name パラメーターがデフォルト COS 名として採用されます。この verb で指定したその他のすべてのパラメーターは無視されます。

mode_chars.description

モードを記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、

QUERY_MODE_DEFINITION verb および QUERY_MODE verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

mode_chars.max_ru_size_upp

このモードのセッションで送受信する RU の最大サイズの上限。この値は、セッション活動化中に最大 RU サイズを折衝する場合に使用されません。

範囲は 256 ~ 61,440 です。*default_ru_size* パラメーター (以下を参照) を AP_YES に設定した場合、このパラメーターは無視されます (値は検査されません)。

mode_chars.receive_pacing_win

このモードを使用するセッションのセッション・ペーシング・ウィンドウ。範囲は 1 ~ 63 です。この値は固定ペーシングにのみ使用され (適応ペーシングには使用されません)、ローカル LU が応答を送信する前にパートナー LU から受信できるフレームの最大数を指定します。Communications

Server for Linux は、隣接ノードが適応ペーシングをサポートしないように指定されている場合を除き、常時、適応ペーシングを使用します。

mode_chars.default_ru_size

最大 RU サイズのデフォルト上限を使用するかどうかを指定します。値は次のいずれかです。

AP_YES Communications Server for Linux は *max_ru_size_upp* パラメーターを無視し、最大 RU サイズの上限をリンク BTU サイズに収まる最大値に設定します。

AP_NO Communications Server for Linux は、最大 RU サイズを定義するために *max_ru_size_upp* パラメーターを使用します。

mode_chars.max_neg_sess_lim

このモードでローカル LU とパートナー LU 間に許可されるセッションの最大数。この値は、*initialize_session_limit* または *change_session_limit* を発行した場合、特定の LU-LU モードの組み合わせでは小さくなる可能性があります。

範囲は 1 ~ 32,767 です。ゼロを指定すると、このモードを使用するセッションをアプリケーションから起動しようとしたときに、Communications Server for Linux は暗黙の CNOS 交換を開始しません。したがって、*initialize_session_limit* を使用してセッション限度を明示的に指定する必要があります。

モードが全二重 APPC 会話で使用される場合、全二重会話ごとに 2 つのセッションが必要となるため注意してください。

mode_chars.plu_mode_session_limit

このモードにおけるデフォルトのセッション限度。1 つのローカル LU とパートナー LU の間でこのモードで成立するセッションの最大数が、このパラメーターにより決まります。この値は CNOS (セッション数の変更) 交換が暗黙に開始されるときに使用されます。

1 ~ 32,767 の範囲で値を指定します (これは *max_neg_sess_lim* の値を超えてはなりません)。ゼロを指定すると、このモードを使用するセッションをアプリケーションから起動しようとしたときに、Communications Server for Linux は暗黙の CNOS 交換を開始しません。したがって、*initialize_session_limit* を使用してセッション限度を明示的に指定する必要があります。

明示的な限度を指定する場合、このモードを使用する LU の LU セッション限度は、その LU が使用するすべてのモードのセッション限度を合計した値に等しいか、それ以上でなければなりません。

モードが全二重 APPC 会話で使用される場合、全二重会話ごとに 2 つのセッションが必要となるため注意してください。

mode_chars.min_conwin_src

このモードを使用するローカル LU が活動化できる競合勝者セッションの最小数。この値は CNOS (セッション数変更) 交換がリモート・システムによって開始されるか、Communications Server for Linux によって暗黙に開始される場合に使用されます。0 ~ 32,767 の範囲の値を指定してください。

い。 *min_conwin_src* パラメーターと *min_conloser_src* パラメーターの合計が、*plu_mode_session_limit* を超えてはいけません。

mode_chars.cos_name

このモードでセッションを活動化するとき要求するサービス・クラスの名前。

ノードが、モードから COS へのマッピングを (DEFINE_NODE の *mode_to_cos_map_supp* パラメーターで定義したように) サポートしている場合、このフィールドで指定する COS は、SNA で定義した COS であるか、事前に DEFINE_COS verb を発行して定義した COS でなければなりません。それ以外の場合、このパラメーターは無視されます。

この名前は 8 バイトからなるタイプ A の文字ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

mode_chars.compression

このモードを使用して活動化されるセッションで圧縮が使用されるかどうかを指定します。値は次のいずれかです。

AP_COMP_PROHIBITED

圧縮は、このモードを使用するセッションではサポートされません。

AP_COMP_REQUESTED

圧縮は、このモードを使用するセッションでサポートおよび要求されます (ただし必須ではありません。圧縮は、パートナーの BIND が要求しない場合は使用されません)。

mode_chars.auto_act

このモードで自動的に活動化されるセッションの数。この値は CNOS (セッション数の変更) 交換が暗黙に開始されるときに使用されます。0 ~ 32,767 の範囲の値を指定してください。

mode_chars.min_conloser_src

このモードを使用するどのローカル LU でも活動化できる競合敗者セッションの最小数。この値は CNOS (セッション数の変更) 交換が暗黙に開始されるときに使用されます。0 ~ 32,767 の範囲の値を指定してください。*min_conwin_src* パラメーターと *min_conloser_src* パラメーターの合計が、*plu_mode_session_limit* を超えてはいけません。

mode_chars.max_ru_size_low

このモードを使用するセッションで送受信される RU の最大サイズの下限。256 ~ 61,440 の範囲で値を指定します。値 0 は、下限がないことを意味します。

この値は、セッション活動化中に最大 RU サイズを折衝する場合に使用されます。このパラメーターは、*default_ru_size* パラメーターを AP_YES に設定した場合には無視されます。

mode_chars.max_receive_pacing_win

このモードにおけるセッションの最大セッション・ペーシング・ウィンドウ。適応ペーシングの場合、この値は、セッションが認可する受信ペーシング・ウィンドウを制限するために使用されます。固定ペーシングの場合は、このパラメーターは使用されません。(Communications Server for Linux

は、隣接ノードが適応ペーシングをサポートしないように指定されている場合を除き、常時、適応ペーシングを使用します。)

0 ~ 32,767 の範囲の値を指定してください。値ゼロは、上限がないことを意味します。

mode_chars.max_compress_lvl

ローカル・ノードからのデータ・フローに関して Communications Server for Linux が折衝を行う最大圧縮レベルを指定します。値は次のいずれかです。

- AP_NONE
- AP_RLE_COMPRESSION
- AP_LZ9_COMPRESSION
- AP_LZ10_COMPRESSION

最大圧縮レベルを指定しない非拡張 BIND を使用して圧縮が折衝される場合、RLE 圧縮が使用されます。

mode_chars.max_decompress_lvl

ローカル・ノードからのデータ・フローに関して Communications Server for Linux が折衝を行う最大解凍レベルを指定します。値は次のいずれかです。

- AP_NONE
- AP_RLE_COMPRESSION
- AP_LZ9_COMPRESSION
- AP_LZ10_COMPRESSION

最大圧縮レベルを指定しない非拡張 BIND を使用して圧縮が折衝される場合、RLE 圧縮が使用されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_CPSVCMG_ALREADY_DEFD
SNA 定義モード CPSVCMG は変更できません。

AP_INVALID_CNOS_SLIM
plu_mode_session_limit パラメーターが、有効ではありません。

AP_INVALID_COMPRESSION
compression パラメーターが、有効ではありません。

AP_INVALID_COS_NAME

cos_name パラメーターが、定義されているどの COS 名にも一致しませんでした。

AP_INVALID_COS_SNASVCMG_MODE

SNA 定義モード SNASVCMG の COS は変更できません。

AP_INVALID_DEFAULT_RU_SIZE

default_ru_size パラメーターが、有効な範囲内にありませんでした。

AP_INVALID_MAX_COMPRESS_LVL

max_compress_lvl パラメーターが、有効ではありません。

AP_INVALID_MAX_DECOMPRESS_LVL

max_decompress_lvl パラメーターが、有効ではありません。

AP_INVALID_MAX_NEGOT_SESS_LIM

max_neg_sess_lim パラメーターが、有効な範囲内にありませんでした。

AP_INVALID_MAX_RU_SIZE_UPPER

max_ru_size_upper パラメーターが、有効な範囲内にありませんでした。

AP_INVALID_MIN_CONLOSERS

min_conloser_src パラメーターが、有効な範囲内にはいか、*plu_mode_session_limit* より大きくなっています。

AP_INVALID_MIN_CONWINNERS

min_conwin_src パラメーターが、有効な範囲内にはいか、*plu_mode_session_limit* より大きくなっています。

AP_INVALID_MIN_CONTENTION_SUM

min_conwin_src と *min_conloser_src* パラメーターの合計が、*plu_mode_session_limit* より大きくなっています。

AP_INVALID_MODE_NAME

mode_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_RECV_PACING_WINDOW

receive_pacing_win パラメーターが、有効な範囲内にありませんでした。

AP_INVALID_SNASVCMG_MODE_LIMIT

SNA 定義モード SNASVCMG は、セッション限度が 2 で *min_conwin_src* が 1 でなければなりません。これらのパラメーターに別の値を指定して SNASVCMG を定義することはできません。

AP_MODE_SESS_LIM_EXCEEDS_NEG

plu_mode_session_limit に指定した値が、*max_neg_sess_lim* に指定した値より大きい値でした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_NODE

この verb は、アプリケーションで新しいノードを定義するか、非活動ノードのパラメーターを変更するために発行します。

この verb は、ノードを実行していないサーバーに対して発行しなければなりません。実行中のノードに対して発行することはできません。

VCB 構造体

```
typedef struct define_node
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;          /* reserved                      */
    unsigned char  format;           /* reserved                      */
    AP_UINT16      primary_rc;       /* primary return code          */
    AP_UINT32      secondary_rc;     /* secondary return code        */
    unsigned char  node_name[128];   /* name of Node                 */
    AP_UINT32      target_handle;    /* handle for subsequent verbs  */
    CP_CREATE_PARMS cp_create_parms; /* CP create parameters         */
} DEFINE_NODE;

typedef struct cp_create_parms
{
    AP_UINT16      crt_parms_len;    /* length of CP_CREATE_PARMS    */
    unsigned char  description[32];  /* resource description          */
    unsigned char  reserv1[2];       /* reserved                      */
    unsigned char  ms_support;       /* MS API level                 */
    unsigned char  queue_nmvt;      /* queue/reject NMVTs          */
    unsigned char  ms_support;       /* reserved                      */
    unsigned char  queue_nmvt;      /* reserved                      */
    unsigned char  reserv3[12];      /* reserved                      */
    unsigned char  node_type;        /* node type                    */
    unsigned char  fqcp_name[17];    /* fully qualified CP name      */
    unsigned char  cp_alias[8];      /* CP alias                     */
    unsigned char  mode_to_cos_map_supp; /* mode to COS mapping support */
    unsigned char  mds_supported;    /* MDS and MS capabilities      */
    unsigned char  node_id[4];       /* node ID                      */
    AP_UINT16      max_locates;      /* maximum locates node can process */
    AP_UINT16      dir_cache_size;   /* directory cache size (reserved */
    /* is not NN)                    */
    AP_UINT16      max_dir_entries;  /* maximum directory entries (zero */
    /* means unlimited)              */
    AP_UINT16      locate_timeout;   /* locate timeout in seconds     */
    unsigned char  reg_with_nn;      /* register resources with NNS   */
    unsigned char  reg_with_cds;     /* register resources with CDS   */
    AP_UINT16      mds_send_alert_q_size; /* size of MDS send alert queue */
    AP_UINT16      cos_cache_size;   /* number of cos definitions     */
    AP_UINT16      tree_cache_size;  /* Topology Database routing tree */
    /* cache size                    */
    AP_UINT16      tree_cache_use_limit; /* number of times a tree can be used */
    AP_UINT16      max_tdm_nodes;    /* max number of nodes that can be */
    /* stored in Topology Database   */
    AP_UINT16      max_tdm_tgs;     /* max number of TGs that can be */
    /* stored in Topology Database   */
    AP_UINT32      max_isr_sessions; /* maximum ISR sessions         */
    AP_UINT32      isr_sessions_upper_threshold; /* upper threshold for ISR */
    /* sessions                      */
    AP_UINT32      isr_sessions_lower_threshold; /* lower threshold for ISR */
}
```

DEFINE_NODE

```

/* sessions */
AP_UINT16    isr_max_ru_size; /* max RU size for ISR */
AP_UINT16    isr_rcv_pac_window; /* ISR receive pacing window size */
unsigned char store_endpt_rscvs; /* endpoint RSCV storage */
unsigned char store_isr_rscvs; /* ISR RSCV storage */
unsigned char store_dlur_rscvs; /* DLUR RSCV storage */
unsigned char dlur_support; /* is DLUR supported? */
unsigned char pu_conc_support; /* is PU conc supported? */
unsigned char nn_rar; /* Route additional resistance */
unsigned char hpr_support; /* Level of hpr support */
unsigned char mobile; /* reserved */
unsigned char discovery_support; /* reserved */
unsigned char discovery_group_name[8]; /* reserved */
unsigned char implicit_lu_0_to_3; /* reserved */
unsigned char default_preference; /* reserved */
unsigned char anynet_supported; /* reserved */
AP_UINT16    max_ls_exception_events; /* Max # exception entries */
unsigned char comp_in_series; /* reserved */
unsigned char max_compress_lvl; /* reserved */
unsigned char node_spec_data_len; /* reserved */
unsigned char ptf[64]; /* program temporary fix array */
unsigned char cos_table_version; /* version of COS tables to use */
unsigned char send_term_self; /* default PLU-SLU session term */
unsigned char disable_branch_awareness; /* disable BrNN awareness */
unsigned char cplu_syncpt_support; /* syncpoint support on CP LU? */
unsigned char cplu_attributes; /* attributes for CP LU */
unsigned char reserved[95]; /* reserved */
} CP_CREATE_PARMS;
```

指定パラメーター

opcode AP_DEFINE_NODE

node_name

アプリケーションで定義したい Communications Server for Linux ノードの名前。

ノード名に . (ピリオド) 文字が含まれている場合、Communications Server for Linux はこれが完全修飾名であると想定します。ピリオドが含まれていない場合は、DNS ルックアップを実行してノード名を判別します。

cp_create_parms.crt_parms_len

パラメーター作成構造体の長さ。

cp_create_parms.description

ノードを記述したテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、QUERY_NODE verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

cp_create_parms.node_type

次のいずれかのノード・タイプです。

AP_NETWORK_NODE

AP_BRANCH_NETWORK_NODE

AP_END_NODE

AP_LEN_NODE

cp_create_parms.fqcp_name

ノードの完全修飾 CP 名。この名前は、右側に EBCDIC のスペースを入れ

た 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

cp_create_parms.cp_alias

ローカル使用の CP 別名。この別名は、APPC アプリケーションで CP LU にアクセスするために使用できます。この名前は、8 バイトからなる ASCII ストリングです。8 バイトはすべて有効であり、設定する必要があります。

cp_create_parms.mode_to_cos_map_supp

ノードがモードから COS へのマッピングを提供するかどうかを指定します。ネットワーク・ノードの場合は、モードから COS へのマッピングは常にサポートされるため、このパラメーターは無視されます。LEN ノードの場合、モードから COS へのマッピングはサポートされません。値は次のいずれかです。

AP_YES モードから COS へのマッピングをサポートします。このノード用に定義するモードには関連する COS 名が含まれていなければならず、その COS 名は、SNA 定義の COS を指定するか、DEFINE_COS で定義した COS の 1 つを指定していなければなりません。

AP_NO モードから COS へのマッピングをサポートしません。デフォルトの COS 名が使用されます。

cp_create_parms.mds_supported

Management Services で複数ドメイン・サポート (MDS) および MS 機能をサポートするかどうかを指定します。値は次のいずれかです。

AP_YES MDS はサポートされます。

AP_NO MDS はサポートされません。

cp_create_parms.node_id

XID 交換で使用されるノード ID。これは 4 バイトの 16 進数ストリングで、ブロック番号 (3 桁の 16 進数字) とノード番号 (5 桁の 16 進数字) で構成されます。

cp_create_parms.max_locates

ノードが同時に処理できる検索要求の最大数。未解決の検索要求 (応答がまだ受信されていない要求) の数がこの限度に達すると、それ以上の検索要求は拒否されます。最小値は 8 です。

cp_create_parms.dir_cache_size

ネットワーク・ノードのみ: ディレクトリー・キャッシュのサイズ。最小値は 3 です。QUERY_DIRECTORY_STATS で戻された情報を使用して、適切なサイズを判別できます。

cp_create_parms.max_dir_entries

ディレクトリー・エントリーの最大数。無制限の場合はゼロを指定してください。

cp_create_parms.locate_timeout

ネットワーク検索がタイムアウトになるまでの時間を秒単位で指定します。タイムアウトを設定しない場合は、ゼロを指定してください。

cp_create_parms.reg_with_nn

ノードの始動時に、そのノードのリソースをネットワーク・ノード・サーバーに登録するかどうかを指定します。このパラメーターの有効値は、ノードがエンド・ノードであるのか分岐ネットワーク・ノードであるのかによって異なることに注意してください。ローカル・ノードがネットワーク・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

エンド・ノードの場合、値は次のいずれかです。

AP_YES リソースを NN に登録します。エンド・ノードのネットワーク・ノード・サーバーは、エンド・ノードに有向検索要求のみを転送します。

AP_NO リソースを登録しません。ネットワーク・ノード・サーバーはすべてのブロードキャスト検索をエンド・ノードに転送します。

分岐ネットワーク・ノードの場合、値は次のいずれかです。

AP_REGISTER_NONE

ローカル・ノードは、どの LU も NN サーバーに登録しません。NN サーバーはすべてのブロードキャスト検索を分岐ネットワーク・ノードに転送します。

AP_REGISTER_ALL

ローカル・ノードは、すべてのドメイン独立 LU を NN サーバーに登録します。また、NN サーバーがオプション・セット 1116 をサポートしている場合には、すべてのローカル従属 LU も登録します。NN サーバーは、(登録できなかった従属 LU を所有していない限り) 有向検索要求のみを転送します。

AP_REGISTER_LOCAL_ONLY

ローカル・ノードは、すべてのローカル独立 LU を NN サーバーに登録します。また、NN サーバーがオプション・セット 1116 をサポートしている場合には、すべてのローカル従属 LU も登録します。NN サーバーはすべてのブロードキャスト検索を分岐ネットワーク・ノードに転送します。

cp_create_parms.reg_with_cds

エンド・ノード: ネットワーク・ノード・サーバーがエンド・ノード・リソースを中央ディレクトリー・サーバーに登録できるかどうかを指定します。*reg_with_nn* を AP_NO に設定した場合は、このフィールドは無視されます。

ネットワーク・ノード: ローカル・リソースまたはドメイン・リソースをオプションで中央ディレクトリー・サーバー (CDS) に登録できるかどうかを指定します。

値は次のいずれかです。

AP_YES リソースを CDS に登録します。

AP_NO リソースを登録しません。

分岐ネットワーク・ノード: BrNN リソース (分岐ネットワーク・ノードのローカル部分となっているリソース、または分岐ネットワーク・ノードのドメインから取り出すリソース) をネットワーク・サーバーが中央ディレクト

リー・サーバーに登録できるかどうかを指定します。 *reg_with_nn* を *AP_REGISTER_NONE* に設定した場合、このフィールドは無視されます。値は次のいずれかです。

AP_YES リソースを CDS に登録します。

AP_NO リソースを登録しません。

cp_create_parms.mds_send_alert_q_size

MDS 送信アラート・キューのサイズ。キューに入れられたアラートの数がこの限度に達すると、Communications Server for Linux はキューの最も古いアラートを削除します。2 を超える値を指定した場合を除き、Communications Server for Linux は値 2 を使用します。

cp_create_parms.cos_cache_size

COS データベースの重みキャッシュのサイズ。この値は、必要な COS 定義の最大数に設定してください。8 を超える値を指定した場合を除き、Communications Server for Linux は値 8 を使用します。

cp_create_parms.tree_cache_size

ネットワーク・ノード: トポロジー・データベースの経路指定ツリー・キャッシュのサイズ。最小値は 8 です。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.tree_cache_use_limit

ネットワーク・ノード: キャッシュに入れられたツリーの最大使用数。この数値を超えると、ツリーはいったん廃棄され、再度計算されます。これにより、ノードは同じ重みの経路間でセッションのバランスを取ることができます。この値を小さくすると、ロード・バランシングが改善されますが、活動化の待ち時間が増大します。最小値は 1 です。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.max_tdm_nodes

ネットワーク・ノード: トポロジー・データベースに格納できるノードの最大数 (ゼロは無制限です)。エンド・ノードまたは LEN ノードの場合は、このパラメーターは予約済みです。

cp_create_parms.max_tdm_tgs

ネットワーク・ノード: トポロジー・データベースに格納できる TG の最大数 (ゼロは無制限です)。エンド・ノードまたは LEN ノードの場合は、このパラメーターは予約済みです。

cp_create_parms.max_isr_sessions

ネットワーク・ノード: ノードが同時に参加できる ISR セッションの最大数。100 を超える値を指定した場合を除き、Communications Server for Linux は値 100 を使用します。エンド・ノードまたは LEN ノードの場合は、このパラメーターは予約済みです。

cp_create_parms.isr_sessions_upper_threshold,

cp_create_parms.isr_sessions_lower_threshold

ネットワーク・ノード: これらのしきい値は、ノードの輻輳状況を制御します。輻輳状況はネットワーク内の他のノードへ報告され、経路の計算に使用されます。ISR セッションの数が上限しきい値を超えた場合、ノードの状態は非輻輳から輻輳に変わります。ISR セッションの数が下限のしきい値

を下回った場合、ノードの状態は非輻輳状態に戻ります。下限のしきい値は上限のしきい値より小さくなくてはならず、上限のしきい値は *max_isr_sessions* より小さくなくてはなりません。エンド・ノードまたは LEN ノードの場合は、このパラメーターは予約済みです。

cp_create_parms.isr_max_ru_size

ネットワーク・ノード: 中間セッション用にサポートされる最大 RU サイズ。指定した値が有効な RU サイズ (SNA フォーマットで記述されたもの) でない場合、Communications Server for Linux はこの値を次に有効な値まで丸めます。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.isr_rcv_pac_window

ネットワーク・ノード: 中間セッションで推奨される、1 ~ 63 の範囲の受信ペーシング・ウィンドウ・サイズ。隣接ノードが適応ペーシングをサポートしていない場合、この値は中間セッションの 2 次ホップでのみ使用されます。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.store_endpt_rscvs

RSCV を診断に使用するために保管するかどうかを指定します (AP_YES または AP_NO)。このフィールドを AP_YES に設定した場合は、RSCV は QUERY_SESSION verb で戻されます。(この値を AP_YES に設定すると、それぞれのエンドポイント・セッションごとに RSCV が格納されることとなります。この追加ストレージは、1 セッションあたり最大 256 バイトです。)

cp_create_parms.store_isr_rscvs

ネットワーク・ノード: RSCV を診断に使用するために保管するかどうかを指定します (AP_YES または AP_NO)。このフィールドを AP_YES に設定した場合は、RSCV は QUERY_ISR_SESSION verb で戻されます。(この値を AP_YES に設定すると、それぞれの ISR セッションごとに RSCV が格納されることとなります。この追加ストレージは、1 セッションあたり最大 256 バイトです。) エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.store_dlur_rscvs

RSCV を診断に使用するために保管するかどうかを指定します (AP_YES または AP_NO)。このフィールドを AP_YES に設定した場合は、RSCV は QUERY_DLUR_LU verb で戻されます。(この値を AP_YES に設定すると、DLUR を使用した PLU-SLU セッションごとに RSCV が格納されることとなります。この追加ストレージは、1 セッションあたり最大 256 バイトです。)

cp_create_parms.dlur_support

DLUR がサポートされるかどうかを指定します。LEN ノードの場合、このパラメーターは予約済みです。値は次のいずれかです。

AP_YES DLUR は使用できます。

AP_LIMITED_DLUR_MULTI_SUBNET | AP_YES

エンド・ノードまたは分岐ネットワーク・ノードの場合: DLUR はサポートされますが、別のサブネットにある DLUR への接続には

使用されません。複数サブネットの操作が必要ない場合、AP_YES ではなくこの値を使用して、ネットワーク・ノードのネットワーク・トラフィックおよび輻輳を削減してください。

この値はネットワーク・ノードではサポートされていません。

AP_NO DLUR は使用できません。

cp_create_parms.pu_conc_support

SNA ゲートウェイがサポートされるかどうかを指定します (AP_YES または AP_NO)。

ノードがダウンストリーム LU と通信するプライマリー RUI アプリケーションの実行のために使用される場合は、このパラメーターを AP_YES に設定する必要があります。

cp_create_parms.nn_rar

ネットワーク・ノードの経路追加抵抗。範囲は 0 ~ 255。

cp_create_parms.hpr_support

ノードが提供する HPR (高性能経路指定) サポートのレベルを指定します。値は次のいずれかです。

AP_NONE

HPR はサポートされません。

AP_BASE

このノードは、自動ネットワーク経路指定 (ANR) を実行できますが、HPR のための RTP (高速トランスポート・プロトコル) エンドポイントとして機能することはできません。

AP_RTP このノードは、自動ネットワーク経路指定 (ANR) を実行でき、HPR セッションのための RTP (高速トランスポート・プロトコル) エンドポイントとして機能することができます。

AP_CONTROL_FLOWS

このノードは、制御フローを含め、すべての HPR 機能を実行できます。

ローカル・ノード LEN ノードの場合、このパラメーターは AP_NONE に設定しなければなりません。そのほかの場合は、AP_CONTROL_FLOWS に設定することをお勧めします。

Enterprise Extender (HPR/IP) リンクまたは MPC+ リンクを使用している場合、このパラメーターは AP_CONTROL_FLOWS に設定しなければなりません。

cp_create_parms.max_ls_exception_events

ノードが記録する LS 例外イベントの最大数。

cp_create_parms.ptf

将来のプログラム一時修正 (PTF) 操作を構成して制御するための次のような配列。

cp_create_parms.ptf[0]

REQDISCONT サポートおよび必須検索状況サポート。

Communications Server for Linux は、通常、REQDISCONT を使用してセッション・トラフィックで不要になった限定リソース・ホスト・リンクを非活

動化します。このバイトを使用して、REQDISCONT の使用を抑止したり、Communications Server for Linux が送信した REQDISCONT 要求で使用された設定を変更したりすることができます。このバイトは、次のいずれかの値に設定してください。

AP_NONE

通常の REQDISCONT サポートを使用します。

AP_SUPPRESS_REQDISCONT

REQDISCONT を使用しません。

AP_OVERRIDE_REQDISCONT

変更された REQDISCONT サポートを使用します。REQDISCONT を指定する場合は、論理 OR 演算を使用して次の値の 1 つまたは両方と結合する必要があります。

AP_REQDISCONT_TYPE

REQDISCONT でタイプ「immediate (即時)」を使用します。この値を指定しなかった場合、Communications Server for Linux はタイプ「normal (標準)」を使用します。

AP_REQDISCONT_RECONTACT

REQDISCONT でタイプ「immediate recontact (即時に再接続する)」を使用します。この値を指定しなかった場合は、Communications Server for Linux はタイプ「no immediate recontact (即時に再接続しない)」を使用します。

AP_ALLOW_BB_RQE

Communications Server for Linux は、通常、要求で方向転換 (CD) も示す必要のある SNA プロトコルにホストが従っている場合を除き、ホストからのいずれの開始ブラケット (BB) 例外 (RQE) 要求もセンス・コード 2003 でリジェクトします。このフラグを設定すると、Communications Server for Linux は、このプロトコルに従っていないホストとのセッションを続行できます。

Communications Server for Linux は、エンド・ノード または分岐ネットワーク・ノードとして実行しているときに、そのネットワーク・ノード・サーバー (NNS) からのネットワーク検索を送信勧誘するかどうかを選択することが可能です。ネットワーク全体のブロードキャスト検索の処理速度が遅くなるので、ネットワーク検索を要求するのは望ましくありません。しかし、ローカル・ノードがそのすべてのリソース (LU) をその NNS に登録できない場合は、検索を要求することが、ネットワークでこれらのリソースを可視にする唯一の方法になります。

通常、Communications Server for Linux はすべての LU を登録できるかどうかを判別して、その NNS からのネットワーク検索を的確に要求します。このノードが、LU を特殊な方法でネットワークにアクセス可能にしている場合 (たとえば、他のノードのゲートウェイとして機能している場合など)、上記の値と以下の値を組み合わせて、標準の操作を変更してください。

AP_SET_SEARCH_STATUS

NNS からのネットワーク検索を無条件で要求します。

cp_create_parms.ptf[1]

ERP サポート。通常、Communications Server for Linux は ACTPU(ERP) を ERP として処理します。これにより、PU-SSCP セッションはリセットされますが、付随する LU-SSCP セッションや PLU-SLU セッションを暗黙的に非活動化することはありません。インプリメントされた SNA によっては、ACTPU(ERP) を ACTPU(cold) として処理し、付随する LU-SSCP セッションや PLU-SLU セッションを暗黙に非活動化する場合があります。このバイトは、次のいずれかの値に設定してください。

AP_NONE

通常の処理を使用します。

AP_OVERRIDE_ERP

すべての ACTPU 要求を ACTPU(cold) として処理します。

cp_create_parms.ptf[2]

BIS サポート。通常、Communications Server for Linux は、限定リソース LU 6.2 セッションを非活動化する前に、BIS プロトコルを使用します。このバイトは、次のいずれかの値に設定してください。

AP_NONE

通常の処理を使用します。

AP_SUPPRESS_BIS

BIS プロトコルは使用しません。限定リソース LU 6.2 セッションは UNBIND(cleanup) を使用して即時に非活動化されます。

cp_create_parms.ptf[3]

APINGD サポート。通常、Communications Server for Linux には、APING 接続テスター用のパートナー・プログラムが組み込まれます。このバイトを使用すると、ノードに到着する APING プログラムによる要求が自動的に処理されないように、ノード内で APING デーモンを使用不可にすることができます。このバイトは、次のいずれかの値に設定してください。

AP_NONE

ノード内に APINGD サポートを組み込みます (通常の処理)。

AP_EXTERNAL_APINGD

ノード内で APINGD を使用不可にします。

cp_create_parms.ptf[4]

LU 0-3 の RU の検査。このバイトを使用して、非標準 SNA データを送信するホスト・システムに予備手段を提供します。このバイトは、以下で説明する特定の問題が発生しない限り、AP_NONE に設定する必要があります。

LU 0-3 RU で Communications Server for Linux の通常の検査を使用するには、このパラメーターを AP_NONE に設定します。

LU 0-3 の RU で特定の検査を緩和するには、以下の値を指定します。

AP_ALLOW_BB_RQE

SNA プロトコルでは、LU 0-3 の PLU-SLU セッションの BB !EB RU は、RQD でなければならないと指定されています。いくつかのホストは、RQE BB !EB CD を送信します。これはプロトコル違反

DEFINE_NODE

ですが、Communications Server for Linux では常に許容します。この値を設定すると、Communications Server for Linux は RQE BB !EB !CD EC RU も許容します。

cp_create_parms.ptf[5]

受信した Attach のセキュリティ検査。

ローカルの呼び出し可能 TP が、会話セキュリティを必要としないように定義されている場合、あるいは定義されていないために会話セキュリティを必要としないデフォルトが使用される場合には、呼び出し側 TP は、ローカルの呼び出し可能 TP にアクセスするために、ユーザー ID とパスワードを送信する必要はありません。呼び出し側 TP にこれらのパラメーターが設定されていて、Communications Server for Linux が受信する Attach メッセージにそのパラメーターが含まれている場合、Communications Server for Linux は、通常、呼び出し可能 TP が会話セキュリティを必要としていなくてもパラメーターを検査します (パラメーターが無効な場合には、Attach をリジェクトします)。このパラメーターを使用すると、検査を使用不可にすることができます。このバイトは、次のいずれかの値に設定してください。

AP_NONE

セキュリティ・パラメーターが受信した Attach に組み込まれている場合には、呼び出し可能 TP のセキュリティ要件に関係なく、これらのパラメーターを常に検査します (通常の処理)。

AP_LIMIT_TP_SECURITY

呼び出し可能 TP がセキュリティを必要としない場合には、受信した Attach でのセキュリティ・パラメーターを検査しません。

cp_create_parms.ptf[6]

HPR の RTP オプション。

Communications Server for Linux の通常の RTP 処理を使用するには、このパラメーターを AP_NONE に設定します。

RTP 操作をカスタマイズするには、以下の値を指定します。

AP_FORCE_STANDARD_ARB

通常、Communications Server for Linux は、標準 ARB アルゴリズムと ARB 応答可能モード・アルゴリズムの両方を RTP 接続でサポートすることを公示します。この値を設定すると、Communications Server for Linux は標準 ARB アルゴリズムのサポートのみを公示します。

cp_create_parms.ptf[7]

DACTLU での DLUR アンバインド。通常、Communications Server for Linux は、DLUR を使用しているセッションのホストからの DACTLU を受信しても、PLU-SLU セッションを終了しません。このバイトは、次のいずれかの値に設定してください。

AP_NONE

通常の処理を使用します。

AP_DLUR_UNBIND_ON_DACTLU

DLUR を使用したセッションで DACTLU を受信すると、PLU-SLU セッションを終了します。

cp_create_parms.ptf[8]

REQACTPU で PU 名を抑止します。 Communications Server for Linux は、DLUR PU を活動化するとき REQACTPU メッセージで PU 名を識別します。このバイトは、次のいずれかの値に設定してください。

AP_NONE

通常の処理を使用します。

AP_SUPPRESS_PU_NAME_ON_REQACTPU

DLUR PU を活動化するとき PU 名を抑止します。

cp_create_parms.ptf[9]

RUI ブラケット・レース・オプションおよび限定リソースによって、接続ネットワークのオプション、および TCP/IP 情報制御ベクトル・オプションが上書きされます。

RUI アプリケーションでブラケット・プロトコルを使用し、RUI アプリケーションが既に BB (開始ブラケット) を送信した後にホストが BB を送信した場合、Communications Server for Linux は、通常、センス・データ 0813 でこれをリジェクトし、BB をアプリケーションに渡しません。このバイトは、次のいずれかの値に設定してください。

AP_NONE

通常の処理を使用します。

AP_LUA_PASSTHRU_BB_RACE

BB を RUI アプリケーションに渡します。アプリケーションは、センス・データ 0813 または 0814 のいずれかを持つ否定応答を送信する必要があります。

接続ネットワークを使用する Communications Server for Linux 内のリンクは、通常、限定リソースです。これを上書きするには、上記の値と次の値を組み合わせます。

AP_CN_OVERRIDE_LIM_RES

それぞれの接続ネットワーク・リンクに関連付けられているポートで *implicit_limited_resource* パラメーターを使用し、限定リソースであるかどうかを判別します。

通常、Communications Server for Linux は、TN3270 または LUA セッション用のホストへの NOTIFY 要求に TCP/IP 情報制御ベクトル (0x64) を含めます。このベクトルにはホスト・コンソールで表示できる情報、またはホストで使用される情報 (たとえば、請求情報: クライアントによって使用される TCP/IP アドレスとポート番号、およびクライアントのアドレスに対応した IP 名) が含まれています。TN3270 の場合、通常、TN3270 サーバーはドメイン・ネーム・サーバー (DNS) ルックアップを実行して、クライアント IP 名を判別します。

クライアント・アドレスが IPv6 アドレスであっても、ホストで IPv6 アドレスを解釈できないバックレベル・バージョンの VTAM を稼働している場合は、クライアント・アドレスがホスト・コンソール上で正しく表示されないことがあります。

この動作は以下のフラグでオーバーライドできます。これを行うには、上記の値と以下の値のいずれかを結合します。

AP_NO_TCPIP_VECTOR

TCP/IP 情報制御ベクトル (0x64) を TN3270 または LUA 用のホストへの NOTIFY 要求に含めません。

ホストがこの制御ベクトルをサポートしない古いバージョンの VTAM を稼働している場合は、この値を使用します。

AP_NO_TCPIP_NAME

DNS ルックアップを実行しないで、IP 名ではなくクライアント IP アドレスを指定して CV64 制御ベクトルを送信します。

この値は TN3270 のみに適用します。したがって、LUA クライアントには DNS ルックアップは必要ありません。この値は DNS 環境が遅いか、またはクライアントが DNS データに含まれていないことが分かっている場合 (たとえば、クライアントが DDNS のない DHCP クライアントの場合) に使用します。

cp_create_parms.cos_table_version

ノードが使用する COS テーブルのバージョンを指定します。このバイトは、次のいずれかの値に設定してください。

AP_VERSION_0_COS_TABLES

本来、「APPN Architecture Reference」で定義されている COS テーブルを使用します。

AP_VERSION_1_COS_TABLES

ATM を介して本来 HPR に定義されている COS テーブルを使用します。

cp_create_parms.send_term_self

ホストに対する PLU-SLU セッションを終了するデフォルトの方式を指定します。指定した値は、LU 定義に異なる値を指定して変更しない限り、ノード上の 0-3 のすべてのタイプの LU で使用されます。値は次のいずれかです。

AP_YES CLOSE_PLU_SLU_SEC_RQ の受信時に TERM_SELF を送信します。

AP_NO CLOSE_PLU_SLU_SEC_RQ の受信時に UNBIND を送信します。

cp_create_parms.disable_branch_awareness

このパラメータは、*node_type* が AP_NETWORK_NODE の場合にのみ適用されます。他のノード・タイプの場合は予約済みです。

ローカル・ノードが、分岐認識 (APPN オプション・セット 1120) をサポートするかどうか、以下のいずれかの値を使用して指定します。

AP_YES ローカル・ノードは分岐認識をサポートしません。このノードと実行済み分岐ネットワーク・ノードの間の TG は、ネットワーク・ト

ポロジーには現れないため、ローカル・ノードはローカル・ノード自体が分岐として認識されているとは報告しません。

AP_NO ローカル・ノードは分岐認識をサポートします。

cp_create_parms.cplu_syncpt_support

ノードの制御点 LU が同期点機能をサポートするかどうかを指定します。このパラメーターは、DEFINE_LOCAL_LU の *syncpt_support* パラメーターと同等のものです。ノードの制御点 LU (明示的な LU 定義を持たないもの) にのみ適用されます。

このパラメーターは、標準 Communications Server for Linux 製品のほかに同期点マネージャー (SPM) と会話保護リソース・マネージャー (C-PRM) が組み込まれている場合にのみ AP_YES に設定してください。値は次のいずれかです。

AP_YES 同期点はサポートされます。

AP_NO 同期点はサポートされません。

cp_create_parms.cplu_attributes

ノードの制御点 LU に関する追加情報を識別します。このパラメーターは、DEFINE_LOCAL_LU の *lu_attributes* パラメーターと同等のものです。ノードの制御点 LU (明示的な LU 定義を持たないもの) にのみ適用されます。

値は次のいずれかです。

AP_NONE

追加情報は識別されません。

AP_DISABLE_PWSUB

制御点 LU のパスワード置換サポートを使用不可にします。パスワード置換とは、パスワードを平文で送信せずに、ローカル LU とリモート LU の間での伝送を行う前にパスワードを暗号化することを言います。通常、Communications Server for Linux は、リモート・システムがパスワード置換をサポートしている場合、パスワード置換を使用します。

この値は、パスワード置換を正しくインプリメントしていないリモート・システムと通信するための予備手段として提供されています。このオプションを使用する場合、平文でのパスワードの送受信 (セキュリティ上のリスクがある場合があります) が行われることに注意しなければなりません。このオプションは、リモート・システムにおけるパスワード置換のインプリメンテーションに関して問題がある場合以外は設定しないでください。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

target_handle

あとに続く verb で使用する戻り値。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_ISR_THRESHOLDS

ISR しきい値パラメーターが、有効ではありませんでした (下限のしきい値が上限より大きいか、上限のしきい値が *max_isr_sessions* を超えていました)。

AP_INVALID_NODE_NAME

node_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_CP_NAME

cp_alias パラメーターまたは *fqcp_name* パラメーターに、有効でない文字が入っていました。

AP_INVALID_NODE_TYPE

node_type パラメーターが、有効な値に設定されていませんでした。

AP_PU_CONC_NOT_SUPPORTED

このバージョンの Communications Server for Linux は SNA ゲートウェイ機能をサポートしていません。

AP_DLUR_NOT_SUPPORTED

このバージョンの Communications Server for Linux は DLUR 機能をサポートしていません。

AP_INVALID_REG_WITH_NN

reg_with_nn パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_COS_TABLE_VERSION

cos_table_version パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_SEND_TERM_SELF

send_term_self パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_DISABLE_BRANCH_AWRN

disable_branch_awareness パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_DLUR_SUPPORT

dlur_support パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_HPR_SUPPORT

hpr_support パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_STATE_CHECK

secondary_rc

AP_NODE_ALREADY_STARTED

ターゲット・ノードが活動状態であるので、この verb を使用してターゲット・ノードの構成を変更することはできません。

DEFINE_NODE は、非活動ノードに対してのみ発行できます。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_PARTNER_LU

DEFINE_PARTNER_LU verb は、ローカル LU とパートナー LU 間の LU-LU セッションについて、パートナー LU のパラメーターを定義するか、既存のパートナー LU を変更します。既存のパートナー LU のパートナー LU 別名を変更することはできません。

通常、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 へのセッションを確立するとき、Communications Server for Linux が暗黙に定義をセットアップするためです。パートナー LU を定義する必要があるのは、論理レコード・サイズ、会話セキュリティー・サポート、並列セッション・サポートのいずれかについて、デフォルト以外の値を強制する必要がある場合のみです。また、セッションを割り振るときにパートナー LU 別名を使用する APPC アプリケーションを使用する場合もあり、その場合は別名を完全修飾パートナー LU 名にマップするため、パートナー LU を定義する必要があります。

ローカル・ノードまたはリモート・ノード (パートナー LU が配置されている) が LEN ノードである場合は、Communications Server for Linux がパートナー LU にアクセスできるように、パートナー LU のディレクトリー・エントリーを定義する必要がありますことに注意してください。これを行うには、DEFINE_ADJACENT_LEN_NODE または DEFINE_DIRECTORY_ENTRY を使用します。ローカル・ノードおよびリモート・ノードの両方がネットワーク・ノードで

DEFINE_PARTNER_LU

ある場合、あるいは一方がネットワーク・ノードであり、他方がエンド・ノードである場合には、Communications Server for Linux は LU を動的に見つけることができるため、ディレクトリー・エントリーは必要ありません。

VCB 構造体

```
typedef struct define_partner_lu
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    PLU_CHARS      plu_chars;      /* partner LU characteristics  */
} DEFINE_PARTNER_LU;

typedef struct plu_chars
{
    unsigned char  fqplu_name[17]; /* fully qualified partner LU name */
    unsigned char  plu_alias[8];  /* partner LU alias              */
    unsigned char  description[32]; /* resource description          */
    unsigned char  reserv2[16];   /* reserved                      */
    unsigned char  plu_un_name[8]; /* partner LU uninterpreted name */
    unsigned char  preference;    /* reserved                      */
    AP_UINT16      max_mc_ll_send_size; /* maximum MC send LL size    */
    unsigned char  conv_security_ver; /* already-verified security   */
                                     /* supported?                   */
    unsigned char  parallel_sess_supp; /* parallel sessions supported? */
    unsigned char  reserv3[8];    /* reserved                      */
} PLU_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_PARTNER_LU

plu_chars.fqplu_name

パートナー LU の完全修飾 LU 名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

plu_chars.plu_alias

パートナー LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

前記の *fqplu_name* パラメーターが既存のパートナー LU の完全修飾名に一致する場合、このパラメーターは既に定義されているパートナー LU 別名に一致しなければなりません。既存のパートナー LU のパートナー LU 別名を変更したり、同じ完全修飾名について複数の LU 別名をセットアップしたりすることはできません。

plu_chars.description

パートナー LU を記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、

QUERY_PARTNER_LU verb および QUERY_PARTNER_LU_DEFINITION verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

plu_chars.plu_un_name

パートナー LU の非解釈名 (リモート SSCP に対して定義された LU の名前)。この名前は 8 バイトの EBCDIC 文字ストリングです。

デフォルトの非解釈名 (前記の *fqplu_name* パラメーターから取ったネットワーク名と同じ) を使用するには、このパラメーターを 8 桁の 2 進ゼロに設定します。このパラメーターは、パートナー LU がホスト上にあり、そのパートナー LU へのアクセスに従属 LU 6.2 を使用する場合にはのみ関連します。

plu_chars.max_mc_ll_send_size

パートナー LU 側のマップ式会話サービスが送受信できる論理レコード最大サイズ。1 ~ 32,767 の範囲で値を指定するか、無制限 (この場合、最大値は 32,767 です) を示すゼロを指定します。

plu_chars.conv_security_ver

パートナー LU に、ローカル LU の代わりにユーザー ID の妥当性を検査する許可を与えるかどうか (つまり、パートナー LU が接続要求で検査されたインディケーターを設定できるかどうか) を指定します。値は次のいずれかです。

AP_YES パートナー LU はユーザー ID の妥当性を検査できます。

AP_NO パートナー LU はユーザー ID の妥当性を検査できません。

plu_chars.parallel_sess_supp

パートナー LU が並列セッションをサポートするかどうかを指定します。値は次のいずれかです。

AP_YES パートナー LU は並列セッションをサポートします。

AP_NO パートナー LU は並列セッションをサポートしません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_DEF_PLU_INVALID_FQ_NAME
fqplu_name パラメーターに、有効でない文字が入っていました。

DEFINE_PARTNER_LU

AP_INVALID_UNINT_PLU_NAME

plu_un_name パラメーターに、有効でない文字が入っていました。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_PLU_ALIAS_CANT_BE_CHANGED

既存のパートナー LU の *plu_alias* パラメーターは変更できません。

AP_PLU_ALIAS_ALREADY_USED

plu_alias パラメーターは、別の LU 名が付いた既存のパートナー LU に既に使用されています。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_PORT

DEFINE_PORT は、新しいポートを定義するか、既存のポートを変更するために使用します。この verb を発行する前に、DEFINE_DLC verb を発行して、このポートが使用する DLC を定義しておかなければなりません。

既存のポートは、始動されていない場合にのみ変更できます。既存のポートに使用されている DLC を変更することはできません。既存のポートを変更するときに指定する *dlc_name* は、そのポートの初期定義で指定された DLC に一致しなければなりません。

着呼を受け付けるポートを定義するには、223 ページの『着呼』を参照してください。

VCB 構造体

```
typedef struct define_port
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;              /* reserved                  */
    unsigned char  format;               /* reserved                  */
    AP_UINT16      primary_rc;          /* primary return code      */
}
```



```

    AP_UINT32      secondary_rc;          /* secondary return code      */
    unsigned char  port_name[8];         /* name of port                */
    PORT_DEF_DATA  def_data;             /* port defined data           */
} DEFINE_PORT;

typedef struct port_def_data
{
    unsigned char  description[32];      /* resource description        */
    unsigned char  initially_active;     /* is the port initially active? */
    unsigned char  reserv2[15];         /* reserved                     */
    unsigned char  dlc_name[8];         /* DLC name associated with port */
    unsigned char  port_type;           /* port type                   */
    unsigned char  port_attributes[4];   /* port attributes             */
    unsigned char  implicit_uplink_to_en; /* implicit EN links up or down? */
    unsigned char  implicit_appn_links_len; /* reserved                    */
    unsigned char  reserv3;            /* reserved                     */
    AP_UINT32      port_number;          /* port number                 */
    AP_UINT16      max_rcv_btu_size;     /* max receive BTU size        */
    AP_UINT16      tot_link_act_lim;     /* total link activation limit  */
    AP_UINT16      inb_link_act_lim;     /* inbound link activation limit */
    AP_UINT16      out_link_act_lim;     /* outbound link activation limit */
    unsigned char  ls_role;             /* initial link station role   */
    unsigned char  retry_flags;         /* reserved                     */
    AP_UINT16      max_activation_attempts; /* reserved                    */
    AP_UINT16      activation_delay_timer; /* reserved                    */
    unsigned char  mltg_pacing_algorithm; /* reserved                    */
    unsigned char  implicit_tg_sharing_prohibited; /* reserved                    */
    unsigned char  link_spec_data_format; /* reserved                     */
    unsigned char  limit_enable;        /* reserved                     */
    unsigned char  reserv1[6];         /* reserved                     */
    unsigned char  implicit_dspu_template[8]; /* implicit dspu template      */
    AP_UINT16      implicit_ls_limit;    /* implicit ls limit           */
    unsigned char  reserv4;            /* reserved                     */
    unsigned char  implicit_dspu_services; /* implicit DSPU support       */
    AP_UINT16      implicit_deact_timer; /* deact timer for implicit LSs */
    AP_UINT16      act_xid_exchange_limit; /* activation XID exchange limit */
    AP_UINT16      nonact_xid_exchange_limit; /* non-activation XID          */
    /* exchange limit */
    unsigned char  ls_xmit_rcv_cap;     /* LS transmit-receive capability */
    unsigned char  max_ifrm_rcvd;       /* maximum number of I-frames that */
    /* can be received */
    AP_UINT16      target_pacing_count;  /* Target pacing count         */
    AP_UINT16      max_send_btu_size;    /* Desired maximum send BTU size */
    LINK_ADDRESS   dlc_data;            /* DLC data                    */
    LINK_ADDRESS   hpr_dlc_data;        /* reserved                     */
    unsigned char  implicit_cp_cp_sess_support; /* implicit links allow      */
    /* CP-CP sessions */
    unsigned char  implicit_limited_resource; /* implicit links are        */
    /* limited resource */
    unsigned char  implicit_hpr_support; /* Is HPR supported?          */
    unsigned char  implicit_link_lvl_error; /* Send HPR traffic on implicit */
    /* links using link-level error */
    /* recovery? */
    unsigned char  retired1;           /* reserved                     */
    TG_DEFINED_CHARS default_tg_chars; /* default TG chars           */
    unsigned char  discovery_supported; /* reserved                     */
    AP_UINT16      port_spec_data_len;  /* length of port specification */
    /* data */
    AP_UINT16      link_spec_data_len;  /* length of link specification */
    /* data */
} PORT_DEF_DATA;

typedef struct link_address
{
    unsigned char  format;              /* type of link address        */
    unsigned char  reservel;           /* reserved                     */
    AP_UINT16      length;              /* length                      */
    unsigned char  address[135];       /* address                      */
} LINK_ADDRESS;

```

DEFINE_PORT

```
typedef struct tg_defined_chars
{
    unsigned char    effect_cap;           /* effective capacity          */
    unsigned char    reserve1[5];         /* reserved                    */
    unsigned char    connect_cost;        /* connection cost            */
    unsigned char    byte_cost;           /* byte cost                  */
    unsigned char    reserve2;           /* reserved                    */
    unsigned char    security;            /* security                   */
    unsigned char    prop_delay;          /* propagation delay          */
    unsigned char    modem_class;         /* reserved                    */
    unsigned char    user_def_parm_1;     /* user-defined parameter 1   */
    unsigned char    user_def_parm_2;     /* user-defined parameter 2   */
    unsigned char    user_def_parm_3;     /* user-defined parameter 3   */
} TG_DEFINED_CHARS;
```

SDLIC の場合、ポート固有またはリンク固有データは含まれていません。

QLLC のポート固有データ:

```
typedef struct vql_port_spec_data
{
    V0_MUX_INFO     mux_info;             /* streams config info        */
    unsigned char   driver_name[13];      /* reserved                   */
    unsigned char   cud_mode;             /* matching required on CUD   */
    AP_UINT16       cud_len;              /* length of Call User Data in octets */
    unsigned char   cud[128];            /* Call User Data             */
    unsigned char   add_mode;             /* matching reqd on called address */
    AP_UINT16       add_len;              /* length of called address to match */
    AP_UINT32       xtras;               /* reserved                   */
    AP_UINT32       xtra_len;             /* reserved                   */
} VQL_PORT_SPEC_DATA;
```

QLLC のリンク固有データ:

```
typedef struct vql_ls_spec_data
{
    V0_MUX_INFO     mux_info;             /* streams config info        */
    AP_UINT16       reserve1;             /* reserved                   */
    AP_UINT16       reserve2;             /* reserved                   */
    unsigned char   vc_type;              /* Virtual Circuit type       */
    unsigned char   req_rev_charge;        /* reserved                   */
    unsigned char   loc_packet;           /* reserved                   */
    unsigned char   rem_packet;           /* reserved                   */
    unsigned char   loc_wsize;            /* reserved                   */
    unsigned char   rem_wsize;            /* reserved                   */
    AP_UINT16       fac_len;              /* X.25 facilities length    */
    unsigned char   fac[128];             /* X.25 facilities           */
    AP_UINT16       retry_limit;          /* reserved                   */
    AP_UINT16       retry_timeout;        /* reserved                   */
    AP_UINT16       idle_timeout;         /* reserved                   */
    AP_UINT16       pvc_id;               /* PVC logical channel identifier */
    AP_UINT16       sn_id_len;            /* reserved                   */
    unsigned char   sn_id[4];             /* reserved                   */
    AP_UINT16       cud_len;              /* length of any call user data */
    /* to send */
    unsigned char   cud[128];             /* actual call user data     */
    AP_UINT32       xtras;               /* reserved                   */
    AP_UINT32       xtra_len;             /* reserved                   */
    unsigned char   rx_thruput_class;     /* reserved                   */
    unsigned char   tx_thruput_class;     /* reserved                   */
    unsigned char   cugo;                 /* reserved                   */
    unsigned char   cug;                  /* reserved                   */
    AP_UINT16       cug_index;            /* reserved                   */
    AP_UINT16       nuid_length;          /* reserved                   */
    unsigned char   nuid_data[109];       /* reserved                   */
}
```

```

    unsigned char  reserve3[2];           /* reserved */
    unsigned char  rpoa_count;           /* reserved */
    AP_UINT16      rpoa_ids[30];        /* reserved */
} VQL_LS_SPEC_DATA;

```

トークンリング、イーサネットのポート固有データ:

```

typedef struct llc_port_spec_data
{
    V0_MUX_INFO    mux_info;             /* Streams config info */
    PROM_MODE_DATA prom_data;           /* reserved */
    LLC_SAP_SPEC_DATA sap_spec_data;    /* LLC2 timeouts and thresholds */
    unsigned char  adapter_id[8];       /* adapter ID */
    unsigned char  adapt_spec_data[16]; /* reserved */
    AP_UINT32      max_rcv_pool_kb;     /* max size of receive buf pool */
    unsigned char  throttle_back_pcent; /* throttle back threshold */
    unsigned char  pad[3];              /* reserved */
} LLC_PORT_SPEC_DATA;

typedef struct prom_mode_data
{
    AP_UINT16      port_spec_data_size; /* reserved */
    unsigned char  promiscuous;         /* reserved */
    unsigned char  dls_w_flag;          /* reserved */
    AP_UINT16      vrn;                 /* reserved */
    AP_UINT16      bridge_num;          /* reserved */
} PROM_MODE_DATA;

typedef struct llc_sap_spec_data
{
    AP_UINT16      ack_timeout;         /* acknowlegment timeout in ms */
    AP_UINT16      p_bit_timeout;      /* Poll response timeout in ms */
    AP_UINT16      t2_timeout;         /* acknowledgment delay in ms */
    AP_UINT16      rej_timeout;        /* *REJ response timeout in seconds*/
    AP_UINT16      busy_state_timeout; /* remote busy timeout in seconds*/
    AP_UINT16      idle_timeout;       /* idle RR interval in seconds */
    AP_UINT16      max_retry;          /* retry limit for any response */
    AP_UINT16      upward_cred_q_threshold; /* reserved */
    AP_UINT16      window_inc_threshold; /* window increment count for */
    AP_UINT16      pad;                /* dynamic window algorithm */
    AP_UINT16      pad;                /* reserved */
} LLC_SAP_SPEC_DATA;

```

トークンリング、イーサネットのリンク固有データ:

```

typedef struct llc_link_spec_data
{
    V0_MUX_INFO    mux_info;             /* Streams config info */
    AP_UINT16      reserve1;            /* reserved */
    AP_UINT16      reserve2;            /* reserved */
    AP_UINT16      length;              /* reserved */
    AP_UINT16      xid_timer;           /* XID timeout value in seconds */
    AP_UINT16      xid_timer_retry;     /* XID retry limit */
    AP_UINT16      test_timer;          /* TEST timeout value in seconds */
    AP_UINT16      test_timer_retry;    /* TEST retry limit */
    AP_UINT16      ack_timeout;         /* acknowledgment timeout in ms */
    AP_UINT16      p_bit_timeout;      /* POLL response timeout in ms */
    AP_UINT16      t2_timeout;         /* acknowledgment delay in ms */
    AP_UINT16      rej_timeout;        /* REJ response timeout in seconds */
    AP_UINT16      busy_state_timeout; /* remote busy timeout in seconds */
    AP_UINT16      idle_timeout;       /* idle RR interval in seconds */
    AP_UINT16      max_retry;          /* retry limit for any response */
} LLC_LINK_SPEC_DATA;

```

マルチパス・チャンネル (MPC) のポート固有データ、System z 上の Communications Server for Linux のみ:

DEFINE_PORT

```
typedef struct chnl_port_spec_data
{
    V0_MUX_INFO    mux_info;           /* streams information */
    AP_UINT16      tx_buffers;         /* reserved */
    AP_UINT16      rx_buffers;         /* reserved */
    AP_UINT32      speed;              /* reserved */
    unsigned char  reserv1[32]         /* pad and future expansion */
} CHNL_PORT_SPEC_DATA;
```

マルチパス・チャネル (MPC) のリンク固有データ:

```
typedef struct chnl_link_spec_data
{
    V0_MUX_INFO    mux_info;           /* streams information */
    AP_UINT16      device_end;         /* BlkMux protocol flag */
    unsigned char  escd_port;          /* reserved */
    unsigned char  cuadd;              /* reserved */
    unsigned char  local_name[8];      /* reserved */
    unsigned char  remote_name[8];    /* reserved */
    unsigned char  reserv1[32]         /* pad and future expansion */
} CHNL_LINK_SPEC_DATA;
```

Enterprise Extender (HPR/IP) のポート固有データ:

```
typedef struct ipdlc_port_spec_data
{
    V0_MUX_INFO    mux_info;           /* streams information */
    unsigned char  if_name[46];        /* Local interface id or IP address */
} IPDLC_PORT_SPEC_DATA;
```

Enterprise Extender (HPR/IP) のリンク固有データ:

```
typedef struct ipdlc_link_spec_data
{
    V0_MUX_INFO    mux_info;           /* streams information */
    AP_UINT16      ack_timeout;        /* ACK timer for command frames */
    AP_UINT16      max_retry;          /* Retry limit for command frames */
    AP_UINT16      liveness_timeout;   /* Liveness timer */
    unsigned char  short_hold_mode;    /* Run in short-hold mode */
    unsigned char  remote_hostname[255]; /* Name of remote host to contact */
} IPDLC_LINK_SPEC_DATA;
```

すべての DLC タイプのデータ:

```
typedef struct v0_mux_info
{
    AP_UINT16      dlc_type;           /* DLC implementation type */
    unsigned char  need_vrfy_fixup;    /* reserved */
    unsigned char  num_mux_ids;        /* reserved */
    AP_UINT32      card_type;          /* type of adapter card */
    AP_UINT32      adapter_number;     /* DLC adapter number */
    AP_UINT32      oem_data_length;    /* reserved */
    AP_INT32       mux_ids[5];         /* reserved */
} V0_MUX_INFO;
```

トークンリングまたはイーサネットの場合、link_address 構造体にある address パラメータは次のように置換されます。

```
typedef struct tr_address
{
    unsigned char  mac_address[6];     /* reserved */
    unsigned char  lsap_address;       /* local SAP address */
} TR_ADDRESS;
```

Enterprise Extender (HPR/IP) の場合、link_address 構造体にある address パラメータは次のように置換されます。

```
typedef struct ip_address_info
{
    unsigned char lsap;                /* Local Service Access Point addr */
    unsigned char version;            /* IPv4 or IPv6 */
    unsigned char address[272];       /* IP Address or hostname */
} IP_ADDRESS_INFO;
```

MPC の場合、完全な link_address 構造体は予約済みです。

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_PORT

port_name

定義するポートの名前。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

def_data.description

ポートを記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、QUERY_PORT verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

def_data.initially_active

このポートをノードの始動時に自動的に起動するかどうかを指定します。値は次のいずれかです。

AP_YES このポートは、ノードの始動時に自動的に起動します。

AP_NO このポートは、このポートを使用する LS が初期設定で活動状態になるように定義されている場合にのみ自動的に始動し、それ以外の場合は、手動で始動する必要があります。

def_data.dlc_name

関連する DLC の名前。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。指定する DLC は、既に DEFINE_DLC verb で定義されていなければなりません。

def_data.port_type

このポートが使用する回線のタイプ。

SDLC の場合、以下の値を指定できます。

AP_PORT_SWITCHED

交換回線。

AP_PORT_NONSWITCHED

非交換回線。

QLLC の場合、このパラメーターは AP_PORT_SWITCHED に設定しなければなりません。

トークンリング、イーサネットの場合、このパラメーターは `AP_PORT_SATF` (共用アクセス・トランスポート機能) に設定しなければなりません。

Enterprise Extender (HPR/IP) の場合、このパラメーターは `AP_PORT_SATF` (共用アクセス・トランスポート機能) に設定しなければなりません。

MPC の場合、このパラメーターは `AP_PORT_SWITCHED` に設定しなければなりません。

def_data.port_attributes

これは 1 ビットのパラメーターで、次の値を取ることができます。

AP_NO 着呼は、CP 名によって解決されます。

AP_RESOLVE_BY_LINK_ADDRESS

これは、着呼を解決するために、まず `CONNECT_IN` 上のリンク・アドレスを使用してから、受信した `XID3` 上の CP 名 (またはノード ID) を使用して解決を試みることを指定します。このビットは、*port_type* パラメーターを `AP_PORT_SWITCHED` に設定した場合以外は無視されます。

def_data.implicit_uplink_to_en

このパラメーターが適用されるのは、ローカル・ノードが分岐ネットワーク・ノードである場合のみです。ローカル・ノードが他のタイプの場合は予約済みです。MPC ポートの場合、暗黙リンクをサポートしていないため、このパラメーターは予約済みです。

隣接ノードがエンド・ノードの場合、このパラメーターは、このポートから切り離された暗黙リンク・ステーションがアップリンクであるのかダウンリンクであるのかを指定します。同じ隣接ノードに既にリンクがある場合、このパラメーターは無視されます。これは、この場合、既存のリンクがリンク・タイプの判別に使用されるためです。値は次のいずれかです。

AP_YES エンド・ノードへの暗黙リンクはアップリンクです。

AP_NO エンド・ノードへの暗黙リンクはダウンリンクです。

def_data.port_number

ポートの番号。

Enterprise Extender (HPR/IP) の場合、このパラメーターは予約済みです。

MPC の場合、このパラメーターは、マルチパス・チャネル・デバイスに対応する番号です。例えば、ポート 0 は `/dev/mpc0` で、ポート 1 は `/dev/mpc1` です。

def_data.max_rcv_btu_size

受信できる BTU の最大サイズ。この値には、RU のみでなく TH と RH の長さ (合計 9 バイト) も含まれます。265 ~ 65535 (SDLC の場合は 265 ~ 4105) の範囲で値を指定します。

def_data.tot_link_act_lim

リンク活動化の合計の限度 (このポートを使用して一度に活動状態にできるリンクの最大数)。

port_type が `AP_NONSWITCHED` に設定され、*ls_role* が `AP_LS_PRI` または `AP_LS_SEC` に設定されている SDLC の場合、範囲は 1 ~ 256 (1 より大き

い値はマルチドロップ 1 次リンクまたはマルチ PU 2 次リンクの定義です)。その他のすべての SDLC ポートには、このパラメーターは 1 に設定します。

MPC リンクの場合は、このパラメーターを 1 に設定しなければなりません。

def_data.inb_link_act_lim

インバウンド・リンク活動化限度 (インバウンド活動化に予約されるリンクの数)。 *inb_link_act_lim* と *out_link_act_lim* の合計が、*tot_link_act_lim* を超えてはなりません。 *inb_link_act_lim* と *tot_link_act_lim* の差は一度にアウトバウンドで活動化できるリンクの最大数を定義します。

port_type が AP_NONSWITCHED に設定された SDLC ポートの場合、このパラメーターはゼロにする必要があります。 *port_type* が AP_SWITCHED に設定されている場合、このポートには着呼 (*inb_link_act_lim* = 1 および *out_link_act_lim* = 0) または発呼 (*inb_link_act_lim* = 0 および *out_link_act_lim* = 1) のいずれかを受け入れるように定義する必要があります。

def_data.out_link_act_lim

アウトバウンド・リンク活動化限度 (アウトバウンド活動化に予約されるリンクの数)。 *inb_link_act_lim* と *out_link_act_lim* の合計が、*tot_link_act_lim* を超えてはなりません。 *out_link_act_lim* と *tot_link_act_lim* の差は一度にインバウンドで活動化できるリンクの最大数を定義します。

port_type が AP_NONSWITCHED に設定された SDLC ポートの場合、このパラメーターは *tot_link_act_lim* と等しくする必要があります。 *port_type* が AP_SWITCHED に設定されている場合、このポートには着呼 (*inb_link_act_lim* = 1 および *out_link_act_lim* = 0) または発呼 (*inb_link_act_lim* = 0 および *out_link_act_lim* = 1) のいずれかを受け入れるように定義する必要があります。

def_data.ls_role

リンク・ステーションの役割。

SDLC または QLLC の場合、以下の値を指定できます。

AP_LS_PRI

1 次

AP_LS_SEC

2 次

AP_LS_NEG

折衝可能

トークンリング、イーサネット、Enterprise Extender (HPR/IP) の場合、これは AP_LS_NEG に設定しなければなりません。

MPC の場合、AP_LS_NEG に設定しなければなりません。

def_data.implicit_dspu_template

Enterprise Extender (HPR/IP)、MPC の場合、このパラメーターは予約済みです。

DEFINE_DSPU_TEMPLATE verb で定義された DSPU テンプレートを指定します。このテンプレートは、ローカル・ノードが、このポート上で活動化される暗黙リンクに SNA ゲートウェイを提供する場合に、定義用に使われます。指定したテンプレートが存在しない場合、またはリンクを活動化したときに、既にそのインスタンス限度に達している場合、活動化は失敗します。このテンプレート名は、ローカル・システム上で表示可能な文字セットで表した 8 バイトのストリングです。

implicit_dspu_services パラメーターが AP_PU_CONCENTRATION に設定されていない場合、*implicit_dspu_template* パラメーターは予約済みです。

def_data.implicit_ls_limit

このポート上で同時に活動状態にできる暗黙リンク・ステーションの最大数を指定します。ダイナミック・リンクと、ディスカバリーに活動化されるリンクも含まれます。1 ~ 65,534 の範囲で値を指定するか、無制限を示す 0 (ゼロ) を指定してください。AP_NO_IMPLICIT_LINKS の値は、暗黙リンクが許容されないことを示します。

def_data.implicit_dspu_services

Enterprise Extender (HPR/IP)、MPC の場合、このパラメーターは予約済みです。

ローカル・ノードが、このポート上で活動化された暗黙リンクを介してダウンストリーム PU に提供するサービスを指定します。値は次のいずれかです。

AP_DLUR

ローカル・ノードは、(DEFINE_DLUR_DEFAULTS verb によって構成されたデフォルト DLUR を使用する) ダウンストリーム PU に DLUR サービスを提供します。

AP_PU_CONCENTRATION

ローカル・ノードは、ダウンストリーム PU に SNA ゲートウェイを提供します。また、ローカル・ノードは、*implicit_dspu_template* パラメーターに指定した DSPU テンプレートが指定したとおりに定義します。

AP_NONE

ローカル・ノードは、このダウンストリーム PU にサービスを提供しません。

def_data.implicit_deact_timer

implicit_hpr_support を AP_YES に設定し、*implicit_limited_resource* を AP_NO_SESSIONS に設定した場合、HPR 可能な暗黙リンクは、このパラメーターで指定したときにそのリンク上でのデータ・フローがなく、そのリンクを使用しているセッションが存在しない場合、自動的に非活動化されます。

暗黙の限定リソース・リンクの非活動化タイマー (秒単位)。

implicit_limited_resource を AP_INACTIVITY に設定した場合、このポートを使用する暗黙リンクは、このパラメーターで指定したときにそのリンク上でのデータ・フローがない場合、非活動化されます。

最小値は 5 で、1 ~ 4 の範囲の値は 5 として解釈されます。ゼロは、タイムアウトがないことを示します (リンクは、*implicit_limited_resource* を

AP_NO に設定した場合と同じように非活動化されません)。このパラメータは、*implicit_limited_resource* を AP_INACTIVITY 以外の値に設定した場合は予約済みです。

def_data.act_xid_exchange_limit
活動化 XID 交換限度。

def_data.nonact_xid_exchange_limit
非活動化 XID 交換限度。

def_data.ls_xmit_rcv_cap
このパラメータは予約済みです。
リンク・ステーションの送受信機能を指定します。値は次のいずれかです。

AP_LS_TWS
両方向同時

AP_LS_TWA
両方向交互

Enterprise Extender (HPR/IP) の場合、このパラメータは AP_LS_TWS に設定しなければなりません。

def_data.max_ifrm_rcvd
ローカル・リンク・ステーションが、肯定応答を送信するまでの間に受信できる I フレームの最大数。範囲は、1 ~ 127 です。

def_data.target_pacing_count
求めるペーシング・ウィンドウ・サイズを示す 1 以上 32,767 以下の数値。(現行バージョンの Communications Server for Linux では、この値は使用されません。)

def_data.max_send_btu_size
このポートから送信できる最大 BTU サイズ。この値は、1 対のリンク・ステーションが相互の通信に使用できる最大 BTU サイズを折衝するとき使用されます。この値には、RU のみでなく TH と RH の長さ (合計 9 バイト) も含まれます。265 ~ 65535 (SDLC の場合は 265 ~ 4105) の範囲で値を指定します。

def_data.dlc_data.format
このポートに指定したリンク・アドレスのタイプ。値は次のいずれかです。

AP_IP_ADDRESS_INFO
IP アドレス。Enterprise Extender (HPR/IP) ポートでこの値を指定します。

AP_UNSPECIFIED
未指定のアドレス・フォーマット。Enterprise Extender (HPR/IP) 以外のすべてのポート・タイプでこの値を指定します。

def_data.dlc_data.length
MPC の場合、このパラメータは予約済みです。
ポート・アドレス (次のパラメータ) の長さ。

def_data.dlc_data.address
ポート・アドレス。

MPC の場合、このパラメーターは予約済みです。

SDLC の場合、これは 1 バイト・アドレスです。 *ls_role* が AP_LS_SEC に設定されているか、*ls_role* が AP_LS_NEG に設定されていて、かつ、ローカル・ステーションが LS ロール・ネゴシエーション後に 2 次になる場合、このアドレスは着呼への応答に使用されます。ローカル・ステーションが 1 次の場合、またはポートが発呼だけに使用される場合、このパラメーターは予約済みです。

QLLC の場合、これはポートのローカル X.25 DTE アドレスを指定するための 1 ~ 14 バイトのストリングです。このアドレスは、このネットワークに使用される X.25 ドライバー用に構成されたアドレスと一致していなければなりません。

注: QLLC ポートにアドレスが指定されていない場合、Communications Server for Linux によって生成される発呼要求には、X.25 起呼アドレスが入っていません。一部のホストでは、このアドレスが着呼でセキュリティ手段として必要であり、このアドレスなしでは接続を受け入れられない可能性があります。

def_data.dlc_data.tr_address.lsap_address

トークンリングまたはイーサネットの場合、ポートのローカル SAP アドレス。 0x04 ~ 0xEC の範囲で 0x04 の倍数を指定してください。

(アドレス構造の最初のパラメーターには、通常、MAC アドレスが入っていますが、この値は LS でのみ使用され、ポートでは予約済みです。)

def_data.dlc_data.ip_address_info.lsap

Enterprise Extender の場合: ポートのローカル SAP アドレス。 0x04 ~ 0xEC の範囲で 0x04 の倍数を指定してください。通常値は 0x04 ですが、VTAM は一部の環境では 0x08 を使用できます。

同じ TCP/IP インターフェースで異なる LSAP アドレスを指定して複数のポートを使用する必要がある場合は、複数の Enterprise Extender DLC を作成する必要があり、そのために、それぞれの DLC には同じ *if_name* (ただし、異なる LSAP アドレス) を指定して、個別の Enterprise Extender ポートを作成します。

def_data.dlc_data.ip_address_info.version

Enterprise Extender の場合: 以下のフィールドが IPv4 または IPv6 アドレスを表すかどうかを指定します。値は次のいずれかです。

IP_VERSION_4_HOSTNAME

address フィールドには、IPv4 アドレス、あるいは IPv4 アドレスに解決するホスト名または別名を指定します。

IP_VERSION_6_HOSTNAME

address フィールドには、IPv6 アドレス、あるいは IPv6 アドレスに解決するホスト名または別名を指定します。

def_data.dlc_data.ip_address_info.address

Enterprise Extender の場合: ポートの IP アドレス。指定できるのは、以下のいずれかです。

- IPv4 ドット 10 進アドレス (たとえば、193.1.11.100)

- IPv6 コロン 16 進アドレス (たとえば、
2001:0db8:0000:0000:0000:0000:1428:57ab または
2001:db8::1428:57ab)
- 名前 (たとえば、newbox.this.co.uk)
- 別名 (たとえば、newbox)

def_data.implicit_cp_cp_sess_support

このポートを使用する暗黙リンク・ステーションに CP-CP セッションを許可するかどうかを指定します。値は次のいずれかです。

AP_YES 暗黙 LS に CP-CP セッションを許可します。

AP_NO 暗黙 LS に CP-CP セッションを許可しません。

def_data.implicit_limited_resource

このポート外の暗黙リンク・ステーションを限定リソースとして定義するかどうかを指定します。値は次のいずれかです。

AP_NO 暗黙リンクは限定リソースではなく、自動的に非活動化されません。

AP_NO_SESSIONS

暗黙リンクは限定リソースであり、それを使用する活動セッションがなくなると、自動的に非活動化されます。

AP_INACTIVITY

暗黙リンクは限定リソースであり、それらのリンクを使用している活動セッションがない場合、または *implicit_deact_timer* フィールドに指定された期間にデータ・フローがなかった場合には、自動的に非活動化されます。

- リンク中に活動状態である SSCP-PU セッションがなくなると、ノードはリンクを即時に非活動化します。
- SSCP-PU セッションが活動状態であっても、指定の時間枠内にトラフィックがない場合、ノードはホストに REQDISCONT(normal) を送信します。その後、ホストがすべての LU および PU の非活動化を担当し、同時にローカル・ノードがリンクを非活動化します。しかし、ホストは活動状態である PLU-SLU セッションを持つ LU を非活動化しないことがあります。この場合、これらのセッションがすべて非活動化される (たとえば、ユーザーのログアウト) まで、リンクは活動状態のままです。この振る舞いは、**START_NODE verb** の *ptf* フィールドのオプションを使用して変更することができます。

def_data.implicit_hpr_support

暗黙リンクで高性能経路指定 (HPR) をサポートするかどうかを指定します。値は次のいずれかです。

AP_YES 暗黙リンクで HPR をサポートします。

AP_NO 暗黙リンクで HPR をサポートしません。

Enterprise Extender (HPR/IP) の場合、このパラメーターは **AP_YES** に設定しなければなりません。

DEFINE_PORT

def_data.implicit_link_lvl_error

SDLC、Enterprise Extender (HPR/IP)MPC の場合、このパラメーターは予約済みです。

リンク・レベルのエラー回復を使用して暗黙リンクに HPR トラフィックを送信するかどうかを指定します (AP_YES または AP_NO)。

implicit_hpr_support を AP_NO に設定した場合、このパラメーターは予約済みです。

def_data.default_tg_chars

デフォルト TG 特性。これらの特性は、このポートを使用する暗黙リンク・ステーションに使用されるか、TG 特性を明示的に定義されていない定義済みリンク・ステーションのデフォルト TG 特性として使用されます。TG 特性パラメーターは、LS がダウンストリーム PU への LS である場合には無視されます。

これらのパラメーターについて詳しくは、135 ページの『DEFINE_LS』を参照してください。

def_data.port_spec_data_len

ポート固有データの長さ。このデータは、基本 VCB 構造体に連結されなければなりません。

ポート固有データは SDLC には使用されません。このパラメーターはゼロに設定します。

def_data.link_spec_data_len

リンク固有データの長さ。リンク固有データは、ポート固有データの直後に連結してください。

これらのパラメーターについて詳しくは、135 ページの『DEFINE_LS』を参照してください。DEFINE_PORT で指定した値は、着呼の処理のデフォルト値として使用されます。(LS 名が初期設定で不明である場合)

QLLC のポート固有データ:

mux_info.dlc_type

DLC のタイプ。これは AP_IMPL_NLI_QLLC に設定します。

cus_mode

着呼で提供される呼び出し利用者データ (CUD) と以下の *cus* パラメーター間で必要になるマッチングのタイプを指定します。値は次のいずれかです。

VQL_DONTCARE

着呼で CUD は検査されない。

VQL_IDENTITY

受信した CUD は *cus* パラメーターに指定されたストリングと一致する必要がある。

VQL_STARTSWITH

受信した CUD の (*cus_len* までの) 初期バイトは *cus* パラメーターに指定されたストリングと一致する必要がある。*cus_len* 以後のバイトは検査されない。

cuda_len

呼び出し利用者データの (下の *cuda* パラメーターで指定) の長さを指定します。

cuda

着呼を検査するために使用される呼び出し利用者データ。上記の *cuda_mode* が *VQL_IDENTITY* または *VQL_STARTSWITH* に設定されているときは、着呼にこのパラメーターで定義された値と一致する CUD スtringが指定されている場合のみ、着呼は受け入れられます。 *cuda_mode* が *VQL_DONTCARE* に設定されている場合、このパラメーターは無視されて、着呼の CUD スtringは検査されません。

add_mode

着呼で提供されるアドレスと *address* パラメーターで定義されたポート・アドレス間で必要になるマッチングのタイプを指定します。値は次のいずれかです。

VQL_DONTCARE

着呼のアドレスは検査されない。

VQL_IDENTITY

受信したアドレスは *address* パラメーターに指定されたスStringと一致する必要がある。

VQL_STARTSWITH

受信したアドレスの (*add_len* までの) 初期バイトは *address* パラメーターに指定されたスStringと一致する必要がある。 *add_len* 以後のバイトは検査されない。

address パラメーターがヌル・スStringに設定されている場合、このパラメーターは *VQL_DONTCARE* に設定しなければなりません。

add_len

add_mode が *VQL_STARTSWITH* に設定されている場合、このパラメーターには検査されるポート・アドレスのバイト数を指定します。

たとえば、*add_len* が 2 が設定されている場合、着呼で提供されたアドレスの先頭 2 バイトが *address* パラメーターの先頭 2 バイトと一致する場合に (それ以降のバイトが一致していなくても)、着呼は受け入れられます。

add_mode がその他の値であれば、このパラメーターは無視されます。

トークンリングまたはイーサネットのポート固有データ:

mux_info.dlc_type

DLC のタイプ。

値は次のいずれかです。

AP_IMPL_TR_SNAP_LLC2

トークンリング

AP_IMPL_ETHER_SNAP_LLC2

イーサネット

DEFINE_PORT

sap_spec_data.ack_timeout

隣接リンク・ステーションに送信された I フレームへの応答がその時間内にあることを Communications Server for Linux が予期するタイムアウト (ミリ秒)。

sap_spec_data.p_bit_timeout

Communications Server for Linux が、ポーリング・ビット・セットとともに送信されたフレームに対する応答を待機する時間 (ミリ秒)。

sap_spec_data.t2_timeout

さらに I フレームを受信し、通知トラフィックを削減するために同じ RR とともに通知することを許可する、受信した I フレームに対する応答を Communications Server for Linux が保留する期間 (ミリ秒)。
Communications Server for Linux は、少なくともこの期間の後に通知を送信しますが、この期間が終了する前に通知を送信することもあります。

sap_spec_data.rej_timeout

Communications Server for Linux が REJ フレームに対する応答を受信することを予期する期間 (秒)。

sap_spec_data.busy_state_timeout

Communications Server for Linux が、隣接リンク・ステーションでの使用中状態の解放の指示を待機する時間 (秒)。

sap_spec_data.idle_timeout

別のアイドル・リンクに対する RR キープアライブの間隔 (秒)。

sap_spec_data.max_retry

応答またはビジー状態の解放を待機している際に許可される再試行の最大数。

sap_spec_data.window_inc_threshold

作業ウィンドウ・サイズを大きくする前に正常に通知される必要がある I フレームの数。この値は、ダイナミック・ウィンドウ・アルゴリズムで使用され、エラー状態に従って縮小されたウィンドウ・サイズを拡大します。

adapter_id

このポートが使用する MAC デバイスのアダプター ID。この ID は、`DEFINE_DLC verb` で定義されているように、このポートが使用する DLC の名前に設定します。

throttle_back_pcent

リモート LS に対する新規送信ウィンドウの認可を保留する使用レベル。受信バッファ・プールの割合です。

max_rcv_pool_kb

受信バッファ・プールが使用できる最大メモリー (キロビット)。

マルチパス・チャンネル (MPC) のポート固有データ:

mux_info.dlc_type

DLC のタイプ。これは `AP_IMPL_MPC_GDLC` に設定してください。

Enterprise Extender (HPR/IP) のポート固有データ:

mux_info.dlc_type

DLC のタイプ。これは AP_IP に設定します。

if_name

複数の IP ネットワークへのアクセス権限を持っている場合は、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 です。1 つだけの IP ネットワークへのアクセス権限を持っている場合は、このフィールドを 2 進ゼロのままにすることができます。

インターフェースを指定する必要がある場合は、以下のいずれかを使用できます。

- IPv4 ドット 10 進アドレス (たとえば、193.1.11.100)
- IPv6 コロン 16 進アドレス (たとえば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)
- インターフェース ID (たとえば、eth0 または en0)

インターフェース ID を判別するには、以下のようになります。

- カードが装着されているサーバーで、コマンド **ifconfig** を実行します。
- 出力された「addr」または「addr6」フィールドを検索して、このパラメーターに使用されている ID を判別します。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PORT_NAME

port_name パラメーターが、有効ではありませんでした。

AP_INVALID_DLC_NAME

指定した *dlc_name* が、定義されているどの DLC にも一致しませんでした。

AP_INVALID_PORT_TYPE

port_type パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_BTU_SIZE

max_rcv_btu_size パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LS_ROLE

ls_role パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LINK_ACTIVE_LIMIT

活動化制限パラメーターの 1 つが、有効な値に設定されていませんでした。

AP_INVALID_MAX_IFRM_RCVD

max_ifrm_rcvd パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LS_ROLE

ls_role パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_DSPU_SERVICES

implicit_dspu_services パラメーターが、有効な値に設定されていませんでした。

AP_PU_CONC_NOT_SUPPORTED

implicit_dspu_services パラメーターが、予約済みの値に設定されていました。

AP_INVALID_TEMPLATE_NAME

implicit_dspu_template パラメーターで指定した DSPU テンプレートが、有効ではありませんでした。

AP_UNKNOWN_IP_HOST

remote_hostname パラメーターに指定したストリングが、有効な IP アドレスに解決できませんでした。

AP_INVALID_SHARING_PROHIBITED

予約済みパラメーターが、ゼロ以外の値に設定されました。

AP_INVALID_LINK_SPEC_FORMAT

予約済みパラメーターが、ゼロ以外の値に設定されました。

AP_INVALID_IMPLICIT_UPLINK

implicit_uplink_to_en パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_PORT_ACTIVE

指定したポートは現在活動状態であるため、変更できません。

AP_DUPLICATE_PORT_NUMBER

指定した *port_number* のポートは、既に定義されています。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター：その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

着呼

着呼を受け付けるポートを構成する場合 (*tot_link_act_lim*、*inb_link_act_lim*、および *out_link_act_lim* の各パラメーターで定義したように)、一般的にはこれらの呼び出しに使用する LS を定義する必要はありません。Communications Server for Linux が着呼の受信時に動的に LS を定義するためです。ただし、従属 LU をサポートするホスト・コンピューターか、SNA ゲートウェイを使用するダウンストリーム・コンピューターから着呼がある場合は、LS を明示的に定義する必要があります。それは、LS 定義が、従属 LU に関連した PU の名前やダウンストリーム PU の名前を含んでいるためです。

着呼がポートに達すると、Communications Server for Linux は呼び出しで指定されているアドレスを、そのポート上で定義されている LS に指定されたアドレス (ある場合) と照合し、その呼び出しに既に定義されている LS があるかどうかを判別します。アドレスが一致しない場合は、動的に LS が定義されます。明示的な LS 定義 (必要な PU 名も含む) が必ず使用されるようにするには、その LS について定義したアドレスが、ホスト・コンピューターまたはダウンストリーム・コンピューターから着呼で提供されるアドレスに一致するようにします。トークンリング、イーサネットの場合、正しい LS を選択するためには、MAC アドレスと SAP アドレスの両方が一致しなければなりません。

DEFINE_RCF_ACCESS

DEFINE_RCF_ACCESS は、Communications Server for Linux リモート・コマンド機能 (RCF) へのアクセス (UNIX コマンド機能 (UCF) のコマンドを実行するためのユーザー ID と、サービス・ポイント・コマンド機能 (SPCF) を使用して管理コマンドを発行するときの制約事項) を指定します。SPCF および UCF について詳しくは、「*Communications Server for Linux 管理ガイド*」を参照してください。この verb を使用して、SPCF と UCF の両方またはそのどちらか一方のみへのアクセスを行うことができます。

この verb は、ドメイン構成ファイルに対して発行しなければなりません。この verb を使用すると、初回の RCF アクセスを指定したり、既存の定義を変更したりできます。Communications Server for Linux は、ノードの始動時にこれらのパラメーターに基づいて動作します。ノードの実行中にこれらのパラメーターを変更した場合、ノードが実行されているサーバー上では、その変更はノードを停止して再始動するまで有効になりません。

VCB 構造体

```
typedef struct define_rcf_access
{
    AP_UINT16      opcode;                /* Verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;                /* reserved                  */
    AP_UINT16      primary_rc;            /* primary return code      */
    AP_UINT32      secondary_rc;          /* secondary return code    */
    unsigned char  ucf_username[32];     /* UCF username             */
    AP_UINT32      spcf_permissions;     /* SPCF permissions        */
    unsigned char  reserv3[8];           /* Reserved                  */
} DEFINE_RCF_ACCESS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_RCF_ACCESS

ucf_username

UCF ユーザーの Linux ユーザー名を指定します。このパラメーターはヌルで終了する ASCII ストリングです。root という名前を指定してはなりません。Communications Server for Linux では、セキュリティ上の理由から UCF コマンドを root として実行することはできません。

すべての UCF コマンドは、デフォルト・シェル、デフォルト・グループ ID、およびこのユーザー用に Linux システム上で定義されたアクセス許可により、このユーザーのユーザー ID を使用して実行されます。

UCF へのアクセスを禁止するには、このパラメーターをヌル・ストリングに設定します。

spcf_permissions

SPCF を使用してアクセスできる Communications Server for Linux verb のタイプを指定します。SPCF へアクセスできないようにするには、これを AP_NONE に設定するか、次の 1 つ以上の値に (論理 OR で結合して) 設定します。

AP_ALLOW_QUERY_LOCAL

QUERY_* verb を許可します。

AP_ALLOW_DEFINE_LOCAL

DEFINE_*, SET_*, DELETE_*, ADD_*, REMOVE_* の各 verb と INIT_NODE を許可します。

AP_ALLOW_ACTION_LOCAL

「アクション」verb、つまり、START_*, STOP_*, ACTIVATE_*, DEACTIVATE_*、および APING、INITIALIZE_SESSION_LIMIT、CHANGE_SESSION_LIMIT、RESET_SESSION_LIMIT も許可します。

AP_ALLOW_QUERY_REMOTE

QUERY_* verb をドメイン内の任意のノードに送信できます。

AP_ALLOW_DEFINE_REMOTE

DEFINE_*, SET_*, DELETE_*, ADD_*, REMOVE_*、および INIT_NODE verb をドメイン内の任意のノードに送信できます。

AP_ALLOW_ACTION_REMOTE

START_*、STOP_*、ACTIVATE_*、DEACTIVATE_*、APING、INITIALIZE_SESSION_LIMIT、CHANGE_SESSION_LIMIT、および RESET_SESSION_LIMIT verb をドメイン内の任意のノードに送信できます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_UCF_USER_CANNOT_BE_ROOT

ucf_username パラメーターで、root という許可されていない名前を指定しました。

AP_INVALID_SPCF_SECURITY

spcf_permissions パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_RTP_TUNING

DEFINE_RTP_TUNING により、RTP 接続をセットアップするときに使用されるパラメーターを指定します。この verb の発行後は、その後のすべての RTP 接続には、新規 DEFINE_RTP_TUNING verb を発行して変更するまでは、指定するパラメーターを使用します。

VCB 構造体

```
typedef struct define_rtp_tuning
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;     /* primary return code          */
}
```

DEFINE_RTP_TUNING

```
AP_UINT32      secondary_rc;          /* secondary return code      */
unsigned char  path_switch_attempts; /* number of path switch attempts */
unsigned char  short_req_retry_limit; /* short request timer retry limit */
AP_UINT16     path_switch_times[4]; /* path switch times          */
AP_UINT32     refifo_cap;           /* maximum for refifo timer    */
AP_UINT32     srt_cap;             /* maximum for short request timer */
unsigned char  reserved[80];       /* reserved                    */
} DEFINE_RTP_TUNING;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_RTP_TUNING

path_switch_attempts

新規 RTP 接続で設定を試行されるパス・スイッチの数。1 ~ 255 の範囲で値を指定します。0 (ゼロ) を指定すると、Communications Server for Linux はデフォルト値の 6 を使用します。

short_req_retry_limit

Communications Server for Linux が RTP 接続の切断およびパス・スイッチ処理の開始を決定する前に、状況要求が送信される回数。1 ~ 255 の範囲で値を指定します。0 (ゼロ) を指定すると、Communications Server for Linux はデフォルト値の 6 を使用します。

path_switch_times

Communications Server for Linux が切断された RTP 接続のパス・スイッチを試みる時間長 (秒単位)。このパラメーターには、それぞれの有効な伝送優先順位に対する 4 つの個別の時間制限として、AP_LOW、AP_MEDIUM、AP_HIGH、および AP_NETWORK の順序で指定します。それぞれの範囲は 1 ~ 65535 です。それぞれの伝送優先順位に指定する値は、それ自体の伝送優先順位より低い伝送優先順位の値を超えないようにしてください。

これらのいずれかの値に 0 (ゼロ) を指定すると、Communications Server for Linux は、以下のように対応するデフォルト値を使用します。

- AP_LOW の場合は 480 秒 (8 分)
- AP_MEDIUM の場合は 240 秒 (4 分)
- AP_HIGH の場合は 120 秒 (2 分)
- AP_NETWORK の場合は 60 秒 (1 分)

refifo_cap

RTP プロトコルは「Re-FIFO Timer」と呼ばれるタイマーを使用します。このタイマーの値はプロトコルの一部として計算されますが、このパラメーターには最大値 (タイマーはこの値を超えて増やすことができない) をミリ秒で指定します。ある状態では、この最大値の設定によりパフォーマンスを改善できます。0 (ゼロ) 値に設定するということは、タイマーの制限がないということであり、プロトコルで計算された、どのような値でも取り込むことができます。

このパラメーターのデフォルト値は 4000 ミリ秒であり、最小値は 250 ミリ秒です。1 ~ 249 の範囲の値が指定されると、250 の値が使用されません。

srt_cap RTP プロトコルは「Short Request Timer」と呼ばれるタイマーを使用しま

す。このタイマーの値はプロトコルの一部として計算されますが、このパラメーターには最大値 (タイマーはこの値を超えて増やすことができない) をミリ秒で指定します。ある状態では、この最大値の設定によりパフォーマンスを改善できます。0 (ゼロ) 値に設定するという事は、タイマーの制限がないということであり、プロトコルで計算された、どのような値でも取り込むことができます。

このパラメーターのデフォルト値は 8000 ミリ秒であり、最小値は 500 ミリ秒です。1 ~ 499 の範囲の値が指定されると、500 の値が使用されます。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_PATH_SWITCH_TIMES

`path_switch_times` パラメーターは無効でした。たとえば、ある伝送優先順位に対して、それより低い伝送優先順位に指定された値を超える値が指定された場合などです。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_SECURITY_ACCESS_LIST

`DEFINE_SECURITY_ACCESS_LIST` は、特定のローカル LU または呼び出し可能 TP にアクセスできるユーザーのリストを定義して、その LU または TP へのアクセスを指定ユーザーに制限します。また、これを使用して、既存のセキュリティー・アクセス・リストにユーザー名を追加することもできます。リストのユーザー名は、`DEFINE_USERID_PASSWORD verb` を使用して定義します。

特定のローカル LU または呼び出し可能 TP へのアクセスを制限するには、次のようにします。

DEFINE_SECURITY_ACCESS_LIST

1. LU または TP の各許可ユーザーは、DEFINE_USERID_PASSWORD verb を使用して定義します。
2. DEFINE_SECURITY_ACCESS_LIST verb を使用して、これらのすべてのユーザー ID が含まれたセキュリティー・アクセス・リストを定義する。
3. LU または TP を定義する DEFINE_LOCAL_LU verb または DEFINE_TP verb を使用して、このセキュリティー・アクセス・リストの名前を指定する。

セキュリティー・アクセス・リストが定義されたローカル LU または呼び出し可能 TP に着呼 Allocate 要求が到着したとき、呼び出し側アプリケーションは、会話セキュリティーを使用することを指示して、ユーザー ID を指定しなければなりません。標準会話セキュリティー検査 (DEFINE_USERID_PASSWORD verb を使用して指定したユーザー ID の検査) のほかに、Communications Server for Linux は、着呼割り振り要求のユーザー ID を LU または TP に定義されたセキュリティー・アクセス・リストで検査し、ユーザー ID が一致しない場合には会話をリジェクトします。LU と TP の両方にセキュリティー・アクセス・リストが定義されている場合、どちらのリストにもユーザー ID がなければなりません。

ローカル LU または呼び出し可能 TP にセキュリティー・アクセス・リストが定義されていないが、それでも会話セキュリティーが必要な構成になっている場合、標準の会話セキュリティー検査は利用されます。

VCB 構造体

DEFINE_SECURITY_ACCESS_LIST verb には、可変数の security_user_data 構造体が含まれており、それらの構造体は、セキュリティー・アクセス・リストに追加するユーザー名を定義しています。ユーザー名構造体は、def_data 構造体の末尾に組み込まれます。それらの構造体の数は、num_users パラメーターで指定します。

```
typedef struct define_security_access_list
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  list_name[14];  /* name of this list           */
    unsigned char  reserv3[2];     /* reserved                    */
    SECURITY_LIST_DEF def_data;    /* security access list        */
} DEFINE_SECURITY_ACCESS_LIST;

typedef struct security_list_def
{
    unsigned char  description[32]; /* description                  */
    unsigned char  reserv3[16];   /* reserved                    */
    AP_UINT32      num_users;     /* number of users being added */
    unsigned char  reserv2[16];   /* reserved                    */
} SECURITY_LIST_DEF;

typedef struct security_user_data
{
    AP_UINT16      sub_overlay_size; /* reserved                    */
    unsigned char  user_name[10];   /* user name                   */
} SECURITY_USER_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_SECURITY_ACCESS_LIST

list_name

セキュリティー・アクセス・リストの名前。これは、ASCII スtringで、右側にスペースを入れます。

この名前が既存のセキュリティー・アクセス・リストと一致すると、この `verb` で定義されたユーザーはそのリストに追加されます。一致しなければ、新規リストが作成されます。

def_data.description

セキュリティー・アクセス・リストを記述した、ヌルで終了するテキスト・String (0 ~ 31 文字のあとにヌル文字を続けたString)。このStringは情報専用です。このStringはノードの構成ファイル内に保管され、`QUERY_SECURITY_ACCESS_LIST verb` で戻されますが、`Communications Server for Linux` では他の用途に使用されません。

def_data.num_users

この `verb` で定義するユーザー名の数。`def_data` 構造体のあとに、各ユーザーの `security_user_data` 構造体を指定しなければなりません。

リストにある、`num_users` で指定した数までのそれぞれのユーザー名ごとに、以下の情報を持った `security_user_data` 構造体が 1 つ必要です。

user_name

ユーザーの名前。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、`Communications Server for Linux` は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、`Communications Server for Linux` は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LIST_NAME

list_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_USER_NAME

1 つ以上の指定ユーザー名が無効でした。

851 ページの『付録 B. 共通戻りコード』に、すべての `NOF verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_TN3270_ACCESS

DEFINE_TN3270_ACCESS は Communications Server for Linux の TN3270 サーバー機能を使用して、特定のクライアントに関する TN3270 のアクセスの詳細 (または、すべてのクライアントに関するデフォルトの TN3270 のアクセスの詳細) を定義します。(TN リダイレクターを使用してクライアントに関するアクセスの詳細を定義するには、DEFINE_TN_REDIRECT を使用します。)

各 verb は、1 つ以上のセッションの詳細を指定します。各セッションは、クライアント・アドレスとサーバー・ポート番号によって一意的に識別されます。

DEFINE_TN3270_ACCESS verb を使用すると、新しいクライアントを定義したり、既存のクライアントが使用する新しいセッションを定義したり、セッション・パラメーターを変更したりできます。(既存のクライアントからセッションを削除するには、DELETE_TN3270_ACCESS を使用します。)

VCB 構造体

DEFINE_TN3270_ACCESS verb には、可変数の tn3270_session_def_data 構造体が含まれており、それらの構造体は、そのユーザーのセッションを定義しています。セッション構造体は、def_data 構造体の末尾に組み込まれます。それらの構造体の数は、num_sessions パラメーターで指定します。

```
typedef struct define_tn3270_access
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    AP_UINT16      default_record; /* is this the DEFAULT record? */
    unsigned char  client_address[256]; /* address of TN3270 user      */
    TN3270_ACCESS_DEF_DATA def_data;
} DEFINE_TN3270_ACCESS;

typedef struct tn3270_access_def_data
{
    unsigned char  description[32]; /* Description - null terminated */
    unsigned char  reserv1[16];   /* reserved                     */
    AP_UINT16      address_format; /* Format of client address      */
    AP_UINT32      num_sessions;  /* Number of sessions being added */
    unsigned char  reserv3[64];   /* reserved                     */
} TN3270_ACCESS_DEF_DATA;

typedef struct tn3270_session_def_data
{
    AP_UINT16      sub_overlay_size; /* reserved                     */
    unsigned char  description[32]; /* Session description          */
    unsigned char  tn3270_support;  /* Level of TN3270 support      */
    unsigned char  allow_specific_lu; /* Allow access to specific LUs */
    unsigned char  printer_lu_name[8]; /* Generic printer LU/pool     */
    /* accessed */
    unsigned char  reserv1[6];     /* reserved                     */
    AP_UINT16      port_number;    /* TCP/IP port used to access   */
    /* server */
    unsigned char  lu_name[8];     /* Generic display LU/pool     */
    /* accessed */
}
```



```

unsigned char    session_type;        /* Unused in current version */
unsigned char    model_override;      /* Unused in current version */
unsigned char    ssl_enabled;         /* Is this an SSL session? */
unsigned char    security_level;      /* SSL encryption strength */
unsigned char    cert_key_label[80];  /* Certificate key label */
unsigned char    listen_local_address[46];
/* Local addr client connects to */
unsigned char    allow_ssl_timeout_to_nonssl;
/* Allow non-SSL clients on SSL? */

unsigned char    reserv3[17];
AP_UINT32        reserv4;            /* reserved */
} TN3270_SESSION_DEF_DATA;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_TN3270_ACCESS

default_record

この verb では、TCP/IP アドレスによって明示的に識別されない TN3270 ユーザーが使用するデフォルト・レコードを定義するかどうかを指定します。TN3270 ユーザーが TN サーバー・ノードに接続を試み、そのユーザーの TCP/IP アドレスが構成情報のどの DEFINE_TN3270_ACCESS レコードにも一致しない場合、デフォルトのレコードが定義されていれば、そのレコードに入っているパラメーターが使用されます。値は次のいずれかです。

AP_YES この verb はデフォルト・レコードを定義します。 *client_address* パラメーターと *address_format* パラメーターは予約済みです。

AP_NO この verb は、通常の TN3270 ユーザー・レコードを定義します。デフォルト・レコードは、TN サーバーを実行しているコンピューターの TCP/IP アドレスを判別できるすべての TN3270 ユーザーに対し、TN サーバー機能へアクセスできるようにします。特定のユーザー・グループのみが TN サーバーを使用できるようにするには、デフォルト・レコードを組み込まないか、デフォルト・レコードが使用されないようにデフォルト・レコードに 3270 LU または LU プールを構成しないことです。

ほとんどのユーザーに対してデフォルト・レコードをセットアップし、1 つ以上の TCP/IP アドレスのみを明示的に除外することもできます。これを行うには、除外するアドレスを TN サーバー・ユーザーとして定義し、それらのアドレスに 3270 LU または LU プールを構成しないようにします。

client_address

TN3270 プログラムを実行するコンピューターの TCP/IP アドレス。これはヌル終了 ASCII ストリングです。以下のいずれかの値を指定できます。

address_format パラメーターで IP アドレスか、または名前のいずれかを指示します。

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

DEFINE_TN3270_ACCESS

名前または別名を使用する場合は、次の制約事項が適用されます。

- その名前または別名を Linux システムが (ローカル TCP/IP 構成を使用するか、ドメイン・ネーム・サーバーを使用して) 完全修飾名に解決できなければなりません。
- それぞれの名前または別名は、固有の完全修飾名に展開できなければなりません。同じ完全修飾名に解決される同じ TN サーバー・ノードのユーザーに、2 つの名前を構成しないでください。
- 完全修飾名の大文字小文字の区別はありません。たとえば、Newbox.THIS.CO.UK と newbox.this.co.uk は同じです。

def_data.description

オプションのテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングは構成ファイル内に保管され、QUERY_TN3270_ACCESS_DEF verb の query_tn3270_access_def 構造体上で戻されますが、Communications Server for Linux では使用されません。このストリングを使用すると、ユーザーを区別するのに役立つ追加情報を保管できます。

def_data.address_format

client_address パラメーターのフォーマットを指定します。値は次のいずれかです。

AP_ADDRESS_IP

IP アドレス (IPv4 または IPv6 のいずれか)

AP_ADDRESS_FQN

別名または完全修飾名

def_data.num_sessions

この verb で定義または変更するセッションの数。各 TN3270 ユーザーは、セッションごとに異なる TCP/IP ポートを使用することにより、同じ TN サーバー・ノードに複数のセッションを使用してアクセスできます。tn3270_access_def_data 構造体のあとに各セッションに対応する、tn3270_session_def_data 構造体を指定します。

各セッションに、次の情報が入った tn3270_session_data 構造体が必要です。

description

オプションのテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングは構成ファイル内に保管され、QUERY_TN3270_ACCESS_DEF verb の query_tn3270_access_def 構造体上で戻されますが、Communications Server for Linux では使用されません。

tn3270_support

TN3270 サポートのレベルを指定します。値は次のいずれかです。

AP_TN3270

TN3270E プロトコルを使用不可にすることを指定します。

AP_TN3270E

TN3270E プロトコルを使用可能にすることを指定します。

TN3270 プロトコルと TN3287 プロトコルは、常に使用可能です。

AS/400® TN3270 クライアントの場合、このパラメーターは AP_TN3270E に設定しなければなりません。

allow_specific_lu

特定の LU へのアクセスを許可するかどうかを示します。値は次のいずれかです。

AP_YES 特定の LU へのアクセスを許可します。

AP_NO 特定の LU へのアクセスを許可しません。

printer_lu_name

このセッションが、汎用プリンター LU を要求する接続に使用するプリンター LU または LU プールの名前。これはタイプ A の EBCDIC スtring で、右側に EBCDIC のスペースを入れます。この名前は、このノード上で定義された LU タイプ 0-3 のプリンター LU の名前か、このノード上の LU が入っている LU プールの名前に一致しなければなりません。

単一のプリンター LU を指定する場合、そのプリンター LU を DEFINE_TN3270_ASSOCIATION verb でディスプレイ LU に関連付けないでください。プリンター LU プールを指定する場合は、そのプール内にディスプレイ LU へ関連付けられたプリンター LU があってはなりません。単一の LU を、汎用プリンター LU と関連プリンター LU のいずれでもアクセスできるようにすると、その LU が既に使用中となり、関連プリンター LU として使用できなくなる場合があります。(これらの規則は、NOF API では適用されません。)

このフィールドは、特定のプリンター LU セッションには影響しません。

port_number

TN3270 プログラムが TN サーバー・ノードにアクセスするために使用するサーバー TCP/IP ポートの番号。このポート番号が、この TN3270 ユーザーのいずれかのセッションに定義された既存のポート番号に一致する場合、そのセッションの情報が置換されます。一致しない場合は、新しいセッションが追加されます。

複数のセッション構造体が (同じ *client_address* または別のものに対して) 同じ *port_number* を使用する場合、*listen_local_address* パラメーターは、それらのすべてに指定するか、あるいはすべてに指定しないようにします。すなわち、このパラメーターを一部のセッションに指定して、他のセッションでは未指定のままにするということではできません。

lu_name

このセッションが、汎用ディスプレイ LU を要求する接続に使用する LU または LU プールの名前。これはタイプ A の EBCDIC スtring で、右側に EBCDIC のスペースを入れます。この名前は、このノード上で定義されたタイプ 0-3 のディスプレイ LU の名前か、このノード上の LU が入っている LU プールの名前に一致しなければなりません。

LU 名を指定した場合、指定した TCP/IP アドレスの TN3270 プログラムは、この TN サーバー・ノード上の指定されたサーバー・ポート番号へ接続することにより、一度に 1 つのセッションのみを使用できます。LU プールを指定した場合、プログラムは複数の汎用ディスプレイ LU セッションを使用でき (または、プログラムの複数のコピーが、この TN サーバーを

DEFINE_TN3270_ACCESS

使用して汎用ディスプレイ LU セッションにアクセスでき)、使用できるセッション数の上限は、プールから使用できるこのノード上の LU の数です。

このパラメーターは、特定のディスプレイ LU セッションには影響しません。

ssl_enabled

このセッションで、サーバーへのアクセスにセキュア・ソケット層 (SSL) を使用するかどうかを示します。

サーバーで SSL をサポートするために必要なソフトウェアを追加インストールしていない場合、このパラメーターは予約済みです。これを確認するには、NOF verb の QUERY_NODE_LIMITS を使用するか、*ssl_support* パラメーターの値を確認します。

値は次のいずれかです。

AP_NO このセッションで SSL は使用しません。

AP_YES このセッションで SSL を使用します。

AP_YES_WITH_CLI_AUTH

このセッションで SSL を使用します。TN サーバーがクライアント認証をするために必要とします。クライアントは、有効な証明書 (そのクライアントが、TN サーバーの使用を許可された有効なクライアントであることを示す情報) を送信しなければなりません。

TN サーバーは、証明書が有効であるかどうかを検査すると同時に、そのユーザーの権限が取り消されないように、外部 LDAP サーバーの証明書取り消しリストでその証明書を検査する必要があります。この場合、DEFINE_TN3270_SSL_LDAP 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。ユーザーに TN3270 高速ログオン機能の使用許可がある場合は、DEFINE_TN3270_EXPRESS_LOGON を使用してこの機能をセットアップする必要があります。

注:

1. このセッションの *port_number* パラメーターが、Telnet デーモンの TCP/IP ポートを使用するよう指定されている場合、このセッションでは SSL を使用しないでください。Telnet デーモンの TCP/IP ポートを使用するセッションで SSL を使用すると、ノードが活動状態であるときに、Telnet クライアントが **telnet** を使用して Communications Server for 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。
2. 非常に多くのクライアントが同じポートを使用していて、そのクライアントを非 SSL から SSL 構成に移行する場合は、移行の進行中に同じポートで SSL と非 SSL 接続の両方を受け入れるように構成をセットアップできます。以下の *allow_ssl_timeout_to_nonssl* パラメーターを参照してください。

security_level

このセッションに必要な SSL セキュリティ・レベルを示します。セッションには、クライアントとサーバーの両方でサポートできるセキュリティ

ー・レベルの中で最高レベルのものを使用します。クライアントが必須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。

ssl_enabled パラメーターを *AP_NO* に設定した場合、このパラメーターは予約済みです。

値は次のいずれかです。

AP_SSL_AUTHENTICATE_MIN

証明書を交換しなければなりません。暗号化は必須ではありません (クライアントが要求した場合は使用できます)。

AP_SSL_AUTHENTICATE_ONLY

証明書を交換しなければなりません。暗号化は使用しません。このオプションは、通常クライアントがセキュア・イントラネットに接続している間、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。

AP_SSL_40_BIT_MIN

最低、40 ビットの暗号化を使用します。

AP_SSL_56_BIT_MIN

最低、56 ビットの暗号化を使用します。

AP_SSL_128_BIT_MIN

最低、128 ビットの暗号化を使用します。

AP_SSL_168_BIT_MIN

最低、168 ビットの暗号化を使用します。

注: 暗号化を使用するには、Communications Server for Linux にソフトウェアを追加インストールする必要があります。詳しくは、

「*Communications Server for Linux 入門*」を参照してください。国によって、各暗号化レベルをサポートするためのソフトウェアが利用できないことがあるので、すべての地域で、リストに挙げたすべての暗号化レベルが使用できるとは限りません。

cert_key_label

このセッションの SSL で使用する証明書と鍵ペアを識別するラベルです。これは SSL 鍵リング・データベースのセットアップ時に指定したラベルと一致していなければなりません。詳しくは、「*Communications Server for Linux 入門*」を参照してください。

このラベルは、ヌルで終了する ASCII 文字ストリングです。SSL 鍵リング・データベースのセットアップ時に指定したデフォルトの SSL の証明書および鍵ペアを使用するには、このパラメーターをヌル・ストリングに設定します。

listen_local_address

TN3270 クライアントの接続先であるローカル TN サーバー・コンピューター上のアドレス。

- いずれのローカル・アドレスにも接続することができる TN3270 クライアントが存在しているか、または TN サーバーにローカル・アドレスが 1 つだけある場合は、このパラメーターを 2 進ゼロに設定します。この

場合、この構造体と同じ *port_number* を使用する (同じ *client_address* または別のものに対して) いずれの *tn3270_session_data* 構造体も、すべて 2 進ゼロに設定されたこのパラメーターが必要になります。

- TN3270 クライアントを特定のローカル・アドレスに制限する必要がある場合、このパラメーターはヌル終了 ASCII スtringとして指定します。指定できるアドレスは、以下のいずれかです。
 - IPv4 ドット 10 進アドレス (たとえば、193.1.11.100)
 - IPv6 コロン 16 進アドレス (たとえば、
2001:0db8:0000:0000:0000:0000:1428:57ab または
2001:db8::1428:57ab)

アドレスを指定する場合、この構造体と同じ *port_number* を使用する (同じ *client_address* または別のものに対して) いずれの *tn3270_session_data* 構造体も、このアドレスがすべてのセッションに対して同じである必要がなくても、このパラメーターに指定された値にする必要があります。

注: 1 つ以上のセッションにローカル・アドレスを指定する場合、このクライアント・レコードは Motif 管理プログラムに表示されません。したがって、このプログラムはこのレコードの表示や管理には使用できません。このレコードを管理するには、現在も、コマンド行管理プログラムの **snaadmin** または NOF アプリケーションを使用して管理します。

allow_ssl_timeout_to_nonssl

このパラメーターは *ssl_enabled* が AP_NO に設定されている場合は適用されません。非 SSL TN3270 クライアントは、SSL を使用するように構成されていても、このセッション・レコードを使用してサーバーをアクセスできません。値は次のいずれかです。

AP_YES SSL を使用しない TN3270 クライアントはサーバーをアクセスできます。SSL ネゴシエーションの開始を待ってサーバーの起動を 5 秒間遅らせることができます。この 5 秒が経過すると、サーバーはクライアントが SSL を使用しないと想定し、通常の TN3270 通信に戻ります。

AP_NO SSL を使用する TN3270 クライアントのみ、サーバーをアクセスできます。

注: このオプションは移行目的として提供されます。非常に多くのクライアントが同じポートを使用していて、そのクライアントを非 SSL から SSL 構成に移行する場合は、移行の進行中に同じポートで SSL と非 SSL 接続の両方を受け入れるように構成をセットアップできます。

非 SSL クライアントに SSL リソースを使用できるようにすると、機密漏れを起こす可能性があります。したがって、このオプションは長期に使用するためのものではありません。このパラメーターは移行を行っている短期間だけ AP_YES に設定し、移行が完了したら AP_NO に設定してください。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを返します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを返します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_UNKNOWN_CLIENT_ADDRESS

指定した名前または別名を完全修飾名へマップできませんでした。

AP_CLIENT_ADDRESS_CLASH

`client_address` パラメーターから解決した完全修飾名が、既に定義されたものと矛盾しています。

AP_DUPLICATE_PORT_NUMBER

別の TN3270 アクセス・セッション・レコードはこのレコードと同じ `port_number` パラメーターを使用しますが、`listen_local_address` パラメーターは一貫性なく設定されます。`listen_local_address` はすべてのレコードで同じポート番号を指定するか、またはいずれのレコードにも指定しないようにします。すなわち、1 つに指定できない場合は、別のものにも指定できません。

AP_TCPIP_PORT_IN_USE

TCP/IP ポート番号が既に別のプログラムによって使用されているため、TN サーバーがその番号を使用できません。

AP_INVALID_TN3270_SUPPORT

1 つ以上のセッションの `tn3270_support` パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_TN3270_ASSOCIATION

DEFINE_TN3270_ASSOCIATION は、ディスプレイ LU とプリンター LU 間の関連付けを定義します。この関連付けにより、TN3270E クライアントはプリンター LU の名前を知らなくても、ディスプレイ LU に関連したプリンター LU に接続できます。DEFINE_TN3270_ASSOCIATION verb を使用すると、新しい関連付けを定義するか、特定のディスプレイ LU についての既存の関連付けを上書きすることができます。

VCB 構造体

```
typedef struct define_tn3270_association
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  display_lu_name[8]; /* Display LU name             */
    TN3270_ASSOCIATION_DEF_DATA def_data; /* association definition      */
} DEFINE_TN3270_ASSOCIATION;

typedef struct tn3270_association_def_data
{
    unsigned char  description[32]; /* description                  */
    unsigned char  reserve0[16];   /* reserved                    */
    unsigned char  printer_lu_name[8]; /* Printer LU name            */
    unsigned char  reserv2[8];     /* reserved                    */
} TN3270_ASSOCIATION_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_TN3270_ASSOCIATION

display_lu_name

def_data.printer_lu_name パラメーターで指定したプリンターへ関連付けるディスプレイ LU の名前。これはタイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

指定するディスプレイ LU は、ローカル・ノード上で定義されたディスプレイ LU でなければなりません、これは NOF API では適用されません。

def_data.description

定義する関連付けの説明。このパラメーターはオプションです。

def_data.printer_lu_name

display_lu_name パラメーターで指定したディスプレイ LU へ関連付けるプリンター LU の名前。これはタイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

指定されたプリンター LU は、ローカル・ノード上で定義されたプリンター LU でなければなりません。

単一のプリンター LU を 2 つの TN3270E エミュレーターで共用することはできません。2 つの TN3270 の関連付けに同じプリンター LU を指定することはできません。

このプリンター LU を、汎用プリンター LU としてアクセス可能にしないでください。アクセス可能にすると、この LU が既に使用中となり、関連プリンター LU として使用できなくなる場合があります。したがって、DEFINE_TN3270_ACCESS verb で関連プリンター LU を *printer_lu_name* として (直接、または LU プールのメンバーとして間接的に) 構成しないでください。

(これらの規則は、NOF API では適用されません。)

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_LU_NAME

指定したディスプレイ LU 名と指定したプリンター LU 名のどちらかが、有効な EBCDIC スtringではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_TN3270_DEFAULTS

DEFINE_TN3270_DEFAULTS は、すべてのクライアント・セッションで使用する TN3270 パラメーターを定義します。

セキュア・ソケット層 (SSL) クライアント認証を使用していて、外部 LDAP サーバー上の証明書取り消しリストでクライアントを検査する場合、このサーバーへのアクセス方法の詳細を構成する必要もあります。それには、DEFINE_TN3270_SSL_LDAP verb を使用してください。

VCB 構造体

```
typedef struct define_tn3270_defaults
{
    AP_UINT16      opcode;          /* verb operation code      */
```

DEFINE_TN3270_DEFAULTS

```
    unsigned char    reserv2;                /* reserved */
    unsigned char    format;                 /* reserved */
    AP_UINT16        primary_rc;             /* primary return code */
    AP_UINT32        secondary_rc;          /* secondary return code */
    TN3270_DEFAULTS_DEF_DATA def_data;      /* TN3270 defaults */
} DEFINE_TN3270_DEFAULTS;
typedef struct tn3270_defaults_def_data
{
    unsigned char    force_responses;        /* force printer responses? */
    unsigned char    keepalive_method;       /* method for sending keep-alives */
    AP_UINT32        keepalive_interval;     /* interval between keep-alives */
    unsigned char    reserv2[32];           /* reserved */
} TN3270_DEFAULTS_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_TN3270_DEFAULTS

def_data.force_responses

プリンター・セッションでのクライアント応答を制御します。値は次のいずれかです。

AP_YES 常にクライアント・プリンター・セッションからの確定応答を要求します。一部の 3270 エミュレーターでは、確定応答が要求されなかった場合、大きなジョブを印刷できません。必要に応じて、*force_responses* を AP_YES に設定して、問題を回避してください。

AP_NO SNA トラフィックに適合した応答を要求します。

def_data.keepalive_method

キープアライブ・メッセージの送信方式。キープアライブ・メッセージとは、接続に他のアクティビティーがない場合に TN3270 クライアントへの TCP/IP 接続をアクティブなままにするために、このクライアントに対して送信されるメッセージのことです。これにより、失敗した接続およびクライアントが検出されます。TCP/IP 接続にトラフィックがない場合は、接続やクライアントの失敗は検出されないため、TN サーバーのリソースが浪費され、LU が他のセッションで使用されなくなります。

値は次のいずれかです。

AP_NONE

キープアライブ・メッセージは送信しません。

AP_TN3270_NOP

Telnet NOP メッセージを送信します。

AP_TN3270_TM

Telnet DO TIMING-MARK メッセージを送信します。

def_data.keepalive_interval

連続したキープアライブ・メッセージ間隔 (秒単位)。特に多数のアイドル状態のクライアント接続が常にある場合には、この間隔を十分長くして、ネットワーク・トラフィックを最小化します。キープアライブの間隔を短くすると、より素早く障害を検出することができますが、ネットワーク・トラ

フィックは増大します。キープアライブの間隔が必要以上に短く、また多数のクライアントが存在するとき、このトラフィックが重大なものになることがあります。

通常の値は 600 ~ 7200 (10 分 ~ 2 時間) です。値 0 (ゼロ) は、*keepalive_method* パラメーターを AP_TN3270_NOP または AP_TN3270_TM に設定した場合には無効です。

TCP/IP の作動上の理由から、構成したキープアライブ間隔は、サーバーがクライアントの消失を認識する際の正確な所要時間にはなりません。正確な時間は、さまざまな要因によって異なりますが、構成されたタイムアウトにさらに数分加えた時間の 2 倍以上にはなりません (正確な数字は、TCP/IP の構成方法によって異なります)。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_KEEPALIVE

keepalive_method パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_TN3270_EXPRESS_LOGON

DEFINE_TN3270_EXPRESS_LOGON は、TN3270 高速ログオン機能をセットアップします。この機能により、Secure Sockets Layer (SSL) クライアント認証機能を使用して Communications Server for Linux TN サーバーまたは TN リダイレクターに接続する TN3270 クライアント・ユーザーは、TN3270 セキュリティーに対して通常使用するユーザー ID およびパスワードを指定する必要はありません。このユーザーのセキュリティ証明書は、必要なユーザー ID およびパスワードを提供するホストの Digital Certificate Access Server (DCAS) に照らして検査されます。

VCB 構造体

```
typedef struct define_tn3270_express_logon
{
    AP_UINT16      opcode;          /* verb operation code          */
    unsigned char  reserv2;        /* reserved                      */
    unsigned char  format;        /* reserved                      */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code        */
    unsigned char  dcas_server[256]; /* IP hostname of DCAS server  */
    AP_UINT16      dcas_port;     /* port number to access server */
    unsigned char  enabled;       /* is Express Logon enabled?    */
    unsigned char  reserv3[33];   /* reserved                      */
} DEFINE_TN3270_EXPRESS_LOGON;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_TN3270_EXPRESS_LOGON

dcas_server

高速ログオン許可を処理するホスト DCAS サーバーの TCP/IP アドレス。
指定できるのは、以下のいずれかです。

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

名前または別名を使用する場合、Linux システムが (ローカル TCP/IP 構成を使用するか、ドメイン・ネーム・サーバーを使用して) この名前または別名を完全修飾名に解決できなければなりません。完全修飾名の大きい文字小文字の区別はありません。たとえば、Newbox.THIS.CO.UK と newbox.this.co.uk は同じです。

dcas_port

DCAS サーバーへのアクセスに使用する TCP/IP ポート番号。

enabled

TN3270 高速ログオン機能を使用可能にするかどうかを指定します。値は次のいずれかです。

AP_YES この機能を使用可能にします。TN3270 クライアントは、ホストにアクセスする場合にユーザー ID およびパスワードを指定する必要がありません。

AP_NO この機能を使用不可にします。TN3270 クライアントは、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。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_TN3270_SSL_LDAP

DEFINE_TN3270_SSL_LDAP は、セキュア・ソケット層 (SSL) のクライアント認証機能で使用する証明書取り消しリストへのアクセス方法を定義します。取り消しリストは外部 LDAP サーバーにあり、TN サーバーまたは TN リダイレクターの使用許可が取り消された (たとえば、ユーザーのセキュリティー情報が無許可通話者に漏れた場合や、そのユーザーが、許可を受けた組織で労働しなくなった場合など) 各 Telnet クライアントの詳細情報が含まれています。

この機能を使用している場合、Communications Server for Linux TN サーバーあるいは TN リダイレクターに接続する TN3270 クライアントは、証明書 (そのクライアントが、サーバーの使用を許可された有効なクライアントであることを示す情報) を提示しなければなりません。サーバーは、取り消しリストでこの証明書を検査して、証明書を有効に保つようにします。

LDAP サーバーへのアクセスを定義する場合、アクセス情報を変更する (たとえば、ユーザー ID やパスワードの変更) 場合、または外部 LDAP サーバーの取り消しリストを Communications Server for Linux が使用しないよう指定する場合は、この verb を使用することができます。

この verb は非活動ノードに対して発行しなければなりません。ノードの実行中に LDAP サーバーのアクセス情報を変更することはできません。

VCB 構造体

```
typedef struct define_tn3270_ssl_ldap
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
} DEFINE_TN3270_SSL_LDAP;
```

define_tn3270_ssl_ldap 構造体の直後には、以下のように、VCB の最後に連結して tn3270_ssl_ldap_def_data 構造体が続かなければなりません。

```
typedef struct tn3270_ssl_ldap_def_data
{
    AP_UINT16      overlay_size;     /* reserved                  */
    unsigned char  auth_type;       /* type of authorization checking */
    unsigned char  reserv1;         /* reserved                  */
    unsigned char  ldap_addr[256];  /* address of LDAP server    */
    AP_UINT16      ldap_port;       /* port number to access server */
    unsigned char  ldap_user[1024]; /* user ID on LDAP server    */
    unsigned char  ldap_password[128]; /* password on LDAP server */
    unsigned char  reserv2[256];    /* reserved                  */
} TN3270_SSL_LDAP_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

DEFINE_TN3270_SSL_LDAP

opcode AP_DEFINE_TN3270_SSL_LDAP

def_data.auth_type

TN サーバーまたは TN リダイレクターが実行する許可検査のタイプを指定します。値は次のいずれかです。

AP_LOCAL_ONLY

サーバーはクライアント証明書をローカルで検査しますが、外部の証明書取り消しリストは使用しません。パラメーター *ldap_addr*—*ldap_password* は予約済みです。

AP_LOCAL_X500

サーバーは証明書をローカルで検査しますが、外部の証明書取り消しリストも検査します。このデータ構造体の残りのパラメーターでは、このリストの場所を指定します。

def_data.ldap_addr

証明書取り消しリストを保持する LDAP サーバーの TCP/IP アドレス。指定できるのは、以下のいずれかです。

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

名前または別名を使用する場合、Linux システムが (ローカル TCP/IP 構成を使用するか、ドメイン・ネーム・サーバーを使用して) この名前または別名を完全修飾名に解決できなければなりません。完全修飾名の大文字小文字の区別はありません。たとえば、Newbox.THIS.CO.UK と newbox.this.co.uk は同じです。

def_data.ldap_port

LDAP サーバーへのアクセスに使用する TCP/IP ポート番号。

def_data.ldap_user

LDAP サーバー上の証明書取り消しリストへのアクセスに使用するユーザー名。このパラメーターをどのように指定するかについては、LDAP サーバーのシステム管理者に確認してください。

def_data.ldap_password

LDAP サーバー上の証明書取り消しリストへのアクセスに使用するパスワード。このパラメーターをどのように指定するかについては、LDAP サーバーのシステム管理者に確認してください。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_AUTH_TYPE

auth_type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_TN_REDIRECT

DEFINE_TN_REDIRECT は Communications Server for Linux の TN リダイレクター機能を使用して、特定の Telnet クライアントに関するアクセスの詳細 (または、すべてのクライアントに関するデフォルトのアクセスの詳細) を定義します。これは、新規クライアントの定義、または既存定義の変更に使えます。(TN3270 サーバーを使用してクライアントに関するアクセスの詳細を定義するには、DEFINE_TN3270_ACCESS を使います。)

VCB 構造体

```
typedef struct define_tn_redirect
{
    AP_UINT16          opcode;           /* verb operation code          */
    unsigned char     reserv2;          /* reserved                     */
    unsigned char     format;          /* reserved                     */
    AP_UINT16          primary_rc;      /* primary return code         */
    AP_UINT32          secondary_rc;    /* secondary return code       */
    TN_REDIRECT_ADDRESS addr;          /* Uniquely defines record     */
    TN_REDIRECT_DEF_DATA def_data;     /* verb data                   */
} DEFINE_TN_REDIRECT;

typedef struct tn_redirect_address
{
    AP_UINT16          default_record;  /* Is this the default record? */
    unsigned char     address_format;   /* IP address or fully-qualified name */
    unsigned char     client_address[256]; /* Client address              */
    AP_UINT16          port_number;     /* Port number that client connects on */
    unsigned char     listen_local_address[46]; /* Local addr client connects to */
    unsigned char     reserved[34];     /* reserved                    */
} TN_REDIRECT_ADDRESS;

typedef struct tn_redirect_def_data
{
    unsigned char     description[32];  /* Description - null terminated */
    unsigned char     reserve0[16];    /* Reserved                     */
    unsigned char     cli_conn_ssl_enabled; /* Is the client session SSL? */
    unsigned char     serv_conn_ssl_enabled; /* Is the host session SSL? */
    unsigned char     host_address_format; /* Type of IP address for the host */
}
```

DEFINE_TN_REDIRECT

```
unsigned char  reserv1;           /* Reserved */
unsigned char  host_address[256]; /* Host address */
AP_UINT16     host_port_number;  /* Port number to connect to host */
unsigned char  cli_conn_security_level; /* SSL encryption strength */
unsigned char  serv_conn_security_level; /* SSL encryption strength */
unsigned char  cli_conn_cert_key_label[80]; /* Key label for certificate */
unsigned char  serv_conn_cert_key_label[80]; /* Key label for certificate */
unsigned char  reserved[46];     /* Reserved */
} TN_REDIRECT_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_TN_REDIRECT

addr.default_record

この *verb* が TCP/IP アドレスによって明示的に識別されない Telnet クライアントが使用するデフォルト・レコードを定義するかどうかを指定します。Telnet クライアントが TN リダイレクター・ノードに接続を試み、そのユーザーの TCP/IP アドレスが構成情報のどの DEFINE_TN_REDIRECT レコードにも一致しない場合、クライアントが使用するポート番号にデフォルトのレコードが定義されていれば、そのレコードのパラメーターが使用されます。値は次のいずれかです。

AP_YES この *verb* はデフォルト・レコードを定義します。 *client_address* パラメーターと *address_format* パラメーターは予約済みです。

AP_NO この *verb* は、通常の TN リダイレクター・ユーザー・レコードを定義します。

デフォルト・レコードは、TN サーバーを実行しているコンピューターの TCP/IP アドレスを判別できるすべての Telnet クライアントに対し、TN リダイレクター機能へアクセスできるようにします。特定のユーザー・グループのみが TN リダイレクターを使用できるようにするには、デフォルト・レコードを組み込まないか、デフォルト・レコードが使用されないようにデフォルト・レコードにホスト・アドレスを構成しないことです。

ほとんどのユーザーに対してデフォルト・レコードをセットアップし、1 つ以上の TCP/IP アドレスのみを明示的に除外することもできます。これを行うには、除外するアドレスを TN リダイレクター・ユーザーとして定義し、それらのアドレスにホスト・アドレスを構成しないようにします。

addr.address_format

client_address パラメーターのフォーマットを指定します。値は次のいずれかです。

AP_ADDRESS_IP

IP アドレス (IPv4 または IPv6 のいずれか)

AP_ADDRESS_FQN

別名または完全修飾名

addr.client_address

Telnet クライアントを実行するコンピューターの TCP/IP アドレス。これはヌル終了 ASCII スtringです。以下のいずれかの値を指定できます。

address_format パラメーターで IP アドレスか、または名前のいずれかを指示します。

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

名前または別名を使用する場合は、次の制約事項が適用されます。

- その名前または別名を Linux システムが (ローカル TCP/IP 構成を使用するか、ドメイン・ネーム・サーバーを使用して) 完全修飾名に解決できなければなりません。
- それぞれの名前または別名は、固有の完全修飾名に展開できなければなりません。同じ完全修飾名に解決される同じ TN リダイレクター・ノードのユーザーに対して、2 つの名前を構成しないでください。
- 完全修飾名の大小文字の区別はありません。たとえば、Newbox.THIS.CO.UK と newbox.this.co.uk は同じです。

addr.port_number

Telnet クライアントが TN リダイレクター・ノードにアクセスするために使用するサーバーの TCP/IP ポート番号。

default_record パラメーターで、これがデフォルトの TN リダイレクター・アクセス・レコードであると指定されている場合、このパラメーターは、デフォルトの TN3270 サーバー・アクセス・レコード (DEFINE_TN3270_ACCESS を使用して定義) が使用するポート・アドレスと異なっていなければなりません。定義できるのは、各ポート番号のデフォルト・レコードの 2 つのタイプのうち 1 つのみです。

複数の *tn_redirect_address* 構造体が (同じ *client_address* または別のものに対して) 同じ *port_number* を使用する場合、*listen_local_address* パラメーターはそれらのすべてに指定するか、あるいはすべてに指定しないようにします。すなわち、このパラメーターを一部のセッションに指定して、他のセッションでは未指定のままにするということはできません。

addr.listen_local_address

TN3270 クライアントの接続先であるローカル TN サーバー・コンピュータ上のアドレス。

- いずれのローカル・アドレスにも接続することができる TN3270 クライアントが存在しているか、または TN サーバーにローカル・アドレスが 1 つだけある場合は、このパラメーターを 2 進ゼロに設定します。この場合、この構造体と同じ *port_number* を使用する (同じ *client_address* または別のものに対して) いずれの *tn_redirect_address* 構造体も、すべて 2 進ゼロに設定されたこのパラメーターが必要になります。
- TN3270 クライアントを特定のローカル・アドレスに制限する必要がある場合、このパラメーターはヌル終了 ASCII スtring として指定します。指定できるアドレスは、以下のいずれかです。
 - IPv4 ドット 10 進アドレス (たとえば、193.1.11.100)

DEFINE_TN_REDIRECT

- IPv6 コロン 16 進アドレス (たとえば、
2001:0db8:0000:0000:0000:0000:1428:57ab または
2001:db8::1428:57ab)

この場合、この構造体と同じ *port_number* を使用する (同じ *client_address* または別のものに対して) いずれの *tn_redirect_address* 構造体も、このアドレスがすべてのセッションに対して同じである必要がなくとも、このパラメーターに指定された値にする必要があります。

注: 1 つ以上のセッションにローカル・アドレスを指定する場合、このクライアント・レコードは Motif 管理プログラムに表示されません。したがって、このプログラムはこのレコードの表示や管理には使用できません。このレコードを管理するには、現在も、コマンド行管理プログラムの **snaadmin** または NOF アプリケーションを使用して管理します。

def_data.description

オプションのテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングは構成ファイル内に保管され、**QUERY_TN_REDIRECT_DEF verb** で戻されますが、**Communications Server for Linux** では使用しません。このストリングを使用すると、ユーザーを区別するのに役立つ追加情報を保管できます。

def_data.cli_conn_ssl_enabled

クライアントが、TN リダイレクターへのアクセスにセキュア・ソケット層 (SSL) を使用するかどうかを示します。

サーバーで SSL をサポートするために必要なソフトウェアを追加インストールしていない場合、このパラメーターは予約済みです。これを確認するには、**NOF verb** の **QUERY_NODE_LIMITS** を使用するか、**ssl_support** パラメーターの値を確認します。

値は次のいずれかです。

AP_NO クライアントは SSL を使用しません。

AP_YES クライアントは SSL を使用します。

AP_YES_WITH_CLI_AUTH

クライアントは SSL を使用します。TN リダイレクターがクライアント認証をするために必要とします。クライアントは、有効な証明書 (そのクライアントが、TN リダイレクターの使用を許可された有効なクライアントであることを示す情報) を送信しなければなりません。

TN リダイレクターは、証明書が有効であるかどうかを検査すると同時に、そのユーザーの権限が取り消されないように、外部 LDAP サーバーの証明書取り消しリストでその証明書を検査する必要もあります。この場合、**DEFINE_TN3270_SSL_LDAP** を使用して、このサーバーへのアクセス方法も指定する必要があります。

def_data.serv_conn_ssl_enabled

TN リダイレクターが、このクライアントのためにホストへアクセスする際、セキュア・ソケット層 (SSL) を使用するかどうかを示します。

サーバーで SSL をサポートするために必要なソフトウェアを追加インストールしていない場合、このパラメーターは予約済みです。これを確認するには、NOF verb の QUERY_NODE_LIMITS を使用するか、*ssl_support* パラメーターの値を確認します。

値は次のいずれかです。

AP_NO ホストは SSL を使用しません。

AP_YES ホストは SSL を使用します。

def_data.host_address_format

host_address パラメーターのフォーマットを指定します。値は次のいずれかです。

AP_ADDRESS_IP

IP アドレス (IPv4 または IPv6 のいずれか)

AP_ADDRESS_FQN

別名または完全修飾名

def_data.host_address

クライアントが通信するホスト・コンピューターの TCP/IP アドレス。これはヌル終了 ASCII スtringです。以下のいずれかの値を指定できます。*host_address_format* パラメーターで IP アドレスか、または名前のいずれかを指示します。

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

名前または別名を使用する場合、Linux システムが (ローカル TCP/IP 構成を使用するか、ドメイン・ネーム・サーバーを使用して) この名前または別名を完全修飾名に解決できなければなりません。完全修飾名の大小文字の区別はありません。たとえば、Newbox.THIS.CO.UK と newbox.this.co.uk は同じです。

def_data.host_port_number

TN リダイレクター・ノードがホストにアクセスするために使用する TCP/IP ポート番号。

def_data.cli_conn_security_level

このセッションのクライアント接続に必要な SSL セキュリティー・レベルを示します。セッションには、クライアントとサーバーの両方でサポートできるセキュリティ・レベルの中で最高レベルのものを使用します。クライアントが必須レベル以上のセキュリティをサポートできない場合、セッションは開始されません。

cli_conn_ssl_enabled パラメーターを AP_NO に設定した場合、このパラメーターは予約済みです。

値は次のいずれかです。

AP_SSL_AUTHENTICATE_MIN

証明書を交換しなければなりません。暗号化は必須ではありません (クライアントが要求した場合は使用できます)。

AP_SSL_AUTHENTICATE_ONLY

証明書を交換しなければなりません。暗号化は使用しません。このオプションは、通常クライアントがセキュア・イントラネットに接続している間、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。

AP_SSL_40_BIT_MIN

最低、40 ビットの暗号化を使用します。

AP_SSL_56_BIT_MIN

最低、56 ビットの暗号化を使用します。

AP_SSL_128_BIT_MIN

最低、128 ビットの暗号化を使用します。

AP_SSL_168_BIT_MIN

最低、168 ビットの暗号化を使用します。

注: 暗号化を使用するには、Communications Server for Linux にソフトウェアを追加インストールする必要があります。詳しくは、

「*Communications Server for Linux 入門*」を参照してください。国によって、各暗号化レベルをサポートするためのソフトウェアが利用できないことがあるので、すべての地域で、リストに挙げたすべての暗号化レベルが使用できるとは限りません。

def_data.serv_conn_security_level

このセッションのホスト接続に必要な SSL セキュリティ・レベルを示します。セッションには、ホストと Communications Server for Linux の両方でサポートできるセキュリティ・レベルの中で最高レベルのものを使用します。ホストが必須レベル以上のセキュリティをサポートできない場合、セッションは開始されません。

serv_conn_ssl_enabled パラメーターを AP_NO に設定した場合、このパラメーターは予約済みです。

値は次のいずれかです。

AP_SSL_AUTHENTICATE_MIN

証明書を交換しなければなりません。暗号化は必須ではありません (ホストが要求した場合は使用できます)。

AP_SSL_AUTHENTICATE_ONLY

証明書を交換しなければなりません。暗号化は使用しません。このオプションは、通常クライアントがセキュア・イントラネットに接続している間、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。

AP_SSL_40_BIT_MIN

最低、40 ビットの暗号化を使用します。

AP_SSL_56_BIT_MIN

最低、56 ビットの暗号化を使用します。

AP_SSL_128_BIT_MIN

最低、128 ビットの暗号化を使用します。

AP_SSL_168_BIT_MIN

最低、168 ビットの暗号化を使用します。

注: 暗号化を使用するには、Communications Server for Linux にソフトウェアを追加インストールする必要があります。詳しくは、「*Communications Server for Linux 入門*」を参照してください。国によって、各暗号化レベルをサポートするためのソフトウェアが利用できないことがあるので、すべての地域で、リストに挙げたすべての暗号化レベルが使用できるとは限りません。

def_data.cli_conn_cert_key_label

クライアント・セッションの SSL で使用する証明書と鍵ペアを識別するラベルです。これは、SSL 鍵リング・データベースのセットアップ時に指定したラベルと一致していなければなりません。詳しくは、「*Communications Server for Linux 入門*」を参照してください。

cli_conn_ssl_enabled パラメーターを AP_NO に設定した場合、このパラメーターは予約済みです。

このラベルは、ヌルで終了する ASCII 文字ストリングです。SSL 鍵リング・データベースのセットアップ時に指定したデフォルトの SSL の証明書および鍵ペアを使用するには、このパラメーターをヌル・ストリングに設定します。

def_data.serv_conn_cert_key_label

ホスト・セッションの SSL で使用する証明書と鍵ペアを識別するラベルです。これは、SSL 鍵リング・データベースのセットアップ時に指定したラベルと一致していなければなりません。詳しくは、「*Communications Server for Linux 入門*」を参照してください。

serv_conn_ssl_enabled パラメーターを AP_NO に設定した場合、このパラメーターは予約済みです。

このラベルは、ヌルで終了する ASCII 文字ストリングです。SSL 鍵リング・データベースのセットアップ時に指定したデフォルトの SSL の証明書および鍵ペアを使用するには、このパラメーターをヌル・ストリングに設定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

DEFINE_TN_REDIRECT

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_UNKNOWN_CLIENT_ADDRESS

指定した名前または別名を完全修飾名へマップできませんでした。

AP_CLIENT_CLASH

client_address パラメーターから解決したポート番号と完全修飾名の組み合わせが、既に定義されたものと重複しています。

AP_DUPLICATE_PORT_NUMBER

別の TN リダイレクター・レコードはこのレコードと同じ *port_number* パラメーターを使用しますが、*listen_local_address* パラメーターは一貫性なく設定されます。*listen_local_address* はすべてのレコードで同じポート番号を指定するか、またはいずれのレコードにも指定しないようにします。すなわち、1 つに指定できない場合は、別のものにも指定できません。

AP_TCPIP_PORT_IN_USE

TCP/IP ポート番号が既に別のプログラムによって使用されているため、TN リダイレクターがその番号を使用できません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター：その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_TP

DEFINE_TP verb は、パートナー LU からの着呼接続の結果として、Communications Server for Linux で TP を起動するために必要な情報を提供します。この verb を使用して、以前定義した TP に関する 1 つ以上のフィールドを変更することもできます。

呼び出し先 TP の標準パラメーターは、呼び出し可能 TP 情報ファイルに定義されています (詳しくは、「*Communications Server for Linux* 管理ガイド」を参照してください)。DEFINE_TP が必要になるのは、そのファイル内で設定できない追加パラメーターを指定する必要がある場合のみです。つまり、会話セキュリティー、確認同期、会話のタイプ (マップ式または基本) について特定のオプションを TP が使用しないようにしたり、一度に実行できる TP のインスタンスの数を制限したりする必要のある場合です。

VCB 構造体

```
typedef struct define_tp
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
}
```

```

    unsigned char    format;                /* reserved                */
    AP_UINT16        primary_rc;           /* primary return code     */
    AP_UINT32        secondary_rc;        /* secondary return code   */
    unsigned char    tp_name[64];         /* TP name                  */
    TP_CHARS         tp_chars;            /* TP characteristics     */
} DEFINE_TP;

typedef struct tp_chars
{
    unsigned char    description[32];      /* resource description     */
    unsigned char    security_list_name[14]; /* security access list name */
    unsigned char    reserv1[2];          /* reserved                 */
    unsigned char    conv_type;           /* conversation type       */
    unsigned char    security_rqd;        /* security support        */
    unsigned char    sync_level;          /* synchronisation level support */
    unsigned char    dynamic_load;        /* dynamic load (AP_YES)   */
    unsigned char    enabled;             /* is the TP enabled?     */
    unsigned char    pip_allowed;          /* program initialization   */
                                        /* parameters supported    */
    unsigned char    reserv3[10];          /* reserved                 */
    AP_UINT16        tp_instance_limit;    /* limit on currently active TP */
                                        /* instances               */
    AP_UINT16        incoming_alloc_timeout; /* incoming allocation timeout */
    AP_UINT16        rcv_alloc_timeout;    /* receive allocation timeout */
    AP_UINT16        tp_data_len;         /* reserved                 */
    unsigned char    tp_data[120];        /* reserved                 */
} TP_CHARS;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_TP

tp_name

定義する TP の名前。

tp_chars.description

TP を記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、QUERY_TP_DEFINITION verb および QUERY_TP verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

tp_chars.security_list_name

この TP が使用するセキュリティー・アクセス・リストの名前 (DEFINE_SECURITY_ACCESS_LIST verb を使用して定義されたもの)。このパラメーターは、指定リストに名前のあるユーザーのみが TP に会話を割り振れるように TP を制限します。セキュリティー・アクセス・リストを指定した場合、*tp_chars.security_rqd* パラメーターは AP_YES に設定しなければなりません。

すべてのユーザーが TP を使用できるように指定するには、このパラメーターを 14 桁の 2 進ゼロに設定します。

tp_chars.conv_type

この TP がサポートする会話のタイプを指定します。値は次のいずれかです。

AP_BASIC

TP は基本会話のみをサポートします。

AP_MAPPED

TP はマップ式会話のみをサポートします。

AP_EITHER

TP は基本会話またはマップ式会話をサポートします。

tp_chars.security_rqd

TP を始動するために会話セキュリティー情報が必要かどうかを指定します。値は次のいずれかです。

AP_YES TP を起動するためにユーザー ID とパスワードが必要です。

AP_NO セキュリティー情報は必要ありません。

tp_chars.sync_level

TP がサポートする同期レベルの値を指定します。値は次のいずれかです。

AP_NONE

TP は *sync_level* NONE のみをサポートします。

AP_CONFIRM_SYNC_LEVEL

TP は *sync_level* CONFIRM のみをサポートします。

AP_EITHER

TP は *sync_level* NONE か CONFIRM のいずれかをサポートします。

AP_SYNCPT_REQUIRED

TP は *sync_level* SYNCPT のみをサポートします (同期点が必要です)。

AP_SYNCPT_NEGOTIABLE

TP は、NONE、CONFIRM、SYNCPT の 3 つの *sync_level* 値をすべてサポートします。

tp_chars.dynamic_load

このパラメーターは、AP_YES に設定しなければなりません。

tp_chars.enabled

TP を正常に接続できるかどうかを指定します。値は次のいずれかです。

AP_YES TP を接続できます。

AP_NO TP を接続できません。

tp_chars.pip_allowed

TP がプログラム初期設定パラメーター (PIP) を受信できるかどうかを指定します。値は次のいずれかです。

AP_YES TP は PIP を受信できます。

AP_NO TP は PIP を受信できません。

tp_chars.tp_instance_limit

一度に活動状態にできるこの TP のインスタンス数に関する限度。値ゼロは無制限です。

tp_chars.incoming_alloc_timeout

着呼 Attach 要求が RECEIVE_ALLOCATE を待つキューに入っている秒数を指定します。値 0 (ゼロ) を指定すると、タイムアウトがないことになり、着呼 Attach 要求のキュー内での滞在期間は無制限になります。

tp_chars.rcv_alloc_timeout

RECEIVE_ALLOCATE verb が着呼 Attach 要求を待つキューに入っている秒数。値 0 (ゼロ) を指定すると、タイムアウトがないことになり、RECEIVE_ALLOCATE verb のキュー内での滞在期間は無制限になります。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_SYSTEM_TP_CANT_BE_CHANGED

指定した TP 名は Communications Server for Linux が内部で使用する TP の名前であり、この名前が付いた TP を定義したり変更したりすることはできません。

AP_INVALID_CONV_TYPE

conv_type パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_SYNC_LEVEL

sync_level パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_DYNAMIC_LOAD

dynamic_load パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_ENABLED

enabled パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_PIP_ALLOWED

pip_allowed パラメーターが、有効な値に設定されていませんでした。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_STATE_CHECK

secondary_rc
値は次のいずれかです。

AP_SECURITY_LIST_NOT_DEFINED

security_list_name パラメーターが、定義されているどのセキュリティー・リスト名にも一致しませんでした。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_TP_LOAD_INFO

DEFINE_TP_LOAD_INFO は、トランザクション・プログラムのロード時に使用する情報を記述したエントリーを定義するか、変更します。アプリケーションは、DEFINE_TP_LOAD_INFO verb を発行する前に、要求された AP_TP_LOAD_INFO の役割を指定して OPEN_FILE を発行しなければなりません。

VCB 構造体

```
typedef struct define_tp_load_info
{
    AP_UINT16          opcode;           /* verb operation code      */
    unsigned char     reserv2;          /* reserved                  */
    unsigned char     format;           /* reserved                  */
    AP_UINT16         primary_rc;       /* primary return code      */
    AP_UINT32         secondary_rc;     /* secondary return code    */
    unsigned char     tp_name[64];      /* TP name                   */
    unsigned char     lu_alias[8];      /* LU alias                  */
    TP_LOAD_INFO_DEF_DATA def_data;     /* defined data              */
} DEFINE_TP_LOAD_INFO;

typedef struct tp_load_info_def_data
{
    unsigned char     description[32];   /* Description               */
    unsigned char     reserv1[16];      /* reserved                  */
    unsigned char     user_id[64];      /* User ID                   */
    unsigned char     group_id[64];     /* Group ID                  */
    AP_UINT32         timeout;          /* Timeout value             */
    unsigned char     type;             /* TP type                   */
    unsigned char     style;           /* reserved                  */
    AP_UINT16         ltv_length;        /* Length of LTV data       */
} TP_LOAD_INFO_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_TP_LOAD_INFO

tp_name

定義する TP ロード情報エントリーの TP 名。これは 64 バイトからなる EBCDIC スtringで、名前が 64 文字に満たない場合は、右側にスペースを入れます。

lu_alias

定義する LU ロード情報エントリーの LU 別名。これは 8 バイトからなる ASCII 文字Stringです。

注: このパラメーターは、TP が APPC TP である場合にのみ使用できます。TP が CPI-C アプリケーションの場合、このパラメーターは予約

済みであり、すべてゼロに設定しなければなりません。CPI-C は、特定のローカル LU からの着呼 Attach 要求の受け入れをサポートしていません。CPI-C アプリケーションに LU の別名を (LU の別名がブランクの場合でも) 指定すると、着呼 Attach 要求を TP に経路指定する際のエラーの原因となります。

def_data.description

TP ロード情報を記述したヌルで終了するテキスト・ストリング (0 ~ 32 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、QUERY_TP verb で戻されますが、Communications Server for Linux では他の用途に使用されません。

def_data.user_id

TP にアクセスして実行するために必要なユーザー ID。

def_data.group_id

TP にアクセスして実行するために必要なグループ ID。

def_data.timeout

TP ロード後のタイムアウト (秒単位)。

def_data.type

TP タイプを指定します。値は次のいずれかです。

- AP_TP_TYPE_QUEUED
- AP_TP_TYPE_QUEUED_BROADCAST
- AP_TP_TYPE_NON_QUEUED

def_data.ltv_length

この verb に付加する LTV データ・ブロックの長さ。それぞれの LTV 構造体は、TP_LOAD_INFO_LTV で指定します。

TP_LOAD_INFO_LTV

LTV データは一連の非バイト整合の LTV として指定し、それぞれの LTV の内容は次のとおりです。

- 2 バイトの長さフィールド。このフィールドの最大値は 258 バイトです。このフィールドは行フォーマットで、NB_PUT_SHORT または NB_GET_SHORT を使用して読み書きが行われます。
- 1 バイトのタイプ・フィールド。以下のいずれかの値に設定します。

AP_TYPE_TP_PATH

パス。値ストリングは、TP 実行可能ファイルの絶対パス名を指定します。

AP_TYPE_TP_ARGUMENTS

引数。値ストリングは、TP が必要とするコマンド行引数を指定します。

AP_TYPE_TP_STDIN

標準入力。値ストリングは、標準入力ファイルまたは標準入力デバイスの絶対パス名を指定します。この LTV を指定しないときのデフォルトは、**/dev/null** です。

AP_TYPE_TP_STDOUT

標準出力。値ストリングは、標準出力ファイルまたは標準出力デバイスの絶対パス名を指定します。この LTV を指定しないときのデフォルトは、`/dev/null` です。

AP_TYPE_TP_STDERR

標準エラー。値ストリングは、標準エラー・ファイルまたは標準エラー・デバイスの絶対パス名を指定します。この LTV を指定しないときのデフォルトは、`/dev/null` です。

AP_TYPE_TP_ENV

環境。値ストリングは、TP が必要とする環境変数を `VARIABLE = VALUE` の形式で指定します。

TP が CPI-C アプリケーションの場合、この LTV を使用して環境変数 `APPCLLU` を設定することはできないことに注意してください。自動ロードされる CPI-C アプリケーションの TP ロード情報に、ローカル LU を指定することはできません。

- 最大 255 バイトの ASCII データからなる値フィールド。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_TP_TYPE

`type` パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LTV_LENGTH

LTV の `length` パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LTV_TYPE

LTV の `type` パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LTV_VALUE

LTV の `value` パラメーターに、有効でないデータが入っていました。

AP_INVALID_TP_STYLE

TP の `style` パラメーターに、有効でない値が入っています。

AP_INVALID_TP_NAME

TP の *name* パラメーターに、EBCDIC のスペースが入っています。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DEFINE_USERID_PASSWORD

DEFINE_USERID_PASSWORD は、APPC および CPI-C の会話セキュリティーに使用するユーザー ID/パスワードの対を定義するか、定義されたユーザー ID とパスワードにプロファイルを追加します。

VCB 構造体

```
typedef struct define_userid_password
{
    AP_UINT16          opcode;           /* verb operation code      */
    unsigned char      reserv2;         /* reserved                 */
    unsigned char      format;          /* reserved                 */
    AP_UINT16          primary_rc;      /* primary return code     */
    AP_UINT32          secondary_rc;    /* secondary return code   */
    AP_UINT16          define_type;     /* what the define type is */
    unsigned char      user_id[10];     /* user id                  */
    unsigned char      reserv3[8];      /* reserved                 */
    USERID_PASSWORD_CHARS password_chars; /* password characteristics */
} DEFINE_USERID_PASSWORD;

typedef struct userid_password_chars
{
    unsigned char      description[32]; /* resource description     */
    unsigned char      reserv2[16];    /* reserved                 */
    AP_UINT16          profile_count;   /* number of profiles     */
    AP_UINT16          reserv1;         /* reserved                 */
    unsigned char      password[10];   /* password                 */
    unsigned char      profiles[10][10]; /* profiles                 */
} USERID_PASSWORD_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DEFINE_USERID_PASSWORD

define_type

この verb の使用方法を指定します。値は次のいずれかです。

AP_ADD_USER

新しいユーザーを追加するか、既存のユーザーのパスワードを変更します。

AP_ADD_PROFILES

既存のユーザーのプロファイルに追加します。

DEFINE_USERID_PASSWORD

user_id ユーザー ID。これは 10 バイトからなるタイプ AE の EBCDIC 文字ストリングで、右側に EBCDIC のスペースを入れます。

一部の CPI-C のインプリメンテーションでは、ユーザー ID の最大長は 8 文字です。9 文字または 10 文字のユーザー ID を指定した場合、他のシステム上で実行されている CPI-C アプリケーションが、このユーザー ID とパスワードを使用すると、Communications Server for Linux システム上のアプリケーションにアクセスできなくなる可能性があります。

password_chars.description

ユーザー ID とパスワードを記述したヌルで終了するテキスト・ストリング (0 ~ 31 文字のあとにヌル文字を続けたストリング)。このストリングは情報専用です。このストリングはノードの構成ファイル内に保管され、`QUERY_USERID_PASSWORD verb` で戻されますが、Communications Server for Linux では他の用途に使用されません。

password_chars.profile_count

プロファイルの数。通常、このパラメーターはゼロに設定されます。詳しくは、以下の *password_chars.profiles* の項を参照してください。

password_chars.password

ユーザーのパスワード。これは 10 バイトからなるタイプ AE の EBCDIC 文字ストリングで、右側に EBCDIC のスペースを入れます。

一部の CPI-C のインプリメンテーションでは、パスワードの最大長は 8 文字です。9 文字または 10 文字のユーザー ID を指定した場合、他のシステム上で実行されている CPI-C アプリケーションが、このユーザー ID とパスワードを使用すると、Communications Server for Linux システム上のアプリケーションにアクセスできなくなる可能性があります。

アプリケーションでこのパラメーターに指定したすべての値は、すぐにこのパスワードを暗号化したものに置き換えられます。したがって、*password_chars.password* パラメーターに指定した値は、書き出されることはありません。

password_chars.profiles

このユーザー ID とパスワードに関連したプロファイル名。それぞれのプロファイル名は 10 バイトからなるタイプ AE の EBCDIC 文字ストリングで、右側に EBCDIC のスペースを入れます。

リモート TP がこのユーザー ID とパスワードを使用してローカル TP に接続し、接続要求にプロファイルを指定する場合、そのプロファイルはここに定義したプロファイル名の 1 つに一致しなければなりません。リモート・システムの管理者と共に、プロファイルが使用されるかどうかを調べ、使用される各プロファイルごとに、そのプロファイル名をこの *verb* の *profiles* パラメーターの 1 つとして指定してください。多くの場合、プロファイル名は使用されないため、この *verb* でプロファイル名を指定する必要はありません。*password_chars.profile_count* をゼロに設定し、プロファイルは指定しないでください。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
AP_PARAMETER_CHECK
```

```
secondary_rc
値は次のいずれかです。
```

AP_INVALID_PASSWORD
password パラメーターに、有効でない文字が入っていました。

AP_INVALID_PROFILE
指定したプロファイルの 1 つ以上が、有効ではありませんでした。

AP_INVALID_UPDATE_TYPE
define_type パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_USERID
user_id パラメーターに、有効でない文字が入っていました。

AP_NO_PROFILES
既存のユーザーにプロファイルを追加するためにこの `verb` が使用されましたが、プロファイルが指定されませんでした。

AP_TOO_MANY_PROFILES
profile_count パラメーターが、有効な値に設定されていませんでした。

AP_UNKNOWN_USER
既存のユーザーにプロファイルを追加するためにこの `verb` が使用されましたが、*user_id* パラメーターが既存のユーザー ID に一致していませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_ADJACENT_LEN_NODE

`DELETE_ADJACENT_LEN_NODE` は、隣接 `LEN` ノードとそれに関連した `LU` に関するノード・ディレクトリー・データベース内のエントリーを除去するか、`LEN` ノード自体は除去せずにその `LEN` ノードの `LU` エントリーを除去します。この `verb` は、`LEN` ノードとそれに関連した `LU` について、一連の `DELETE_DIRECTORY_ENTRY verb` を発行することと同様です。

VCB 構造体

```
typedef struct delete_adjacent_len_node
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                 */
    unsigned char  format;          /* reserved                 */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
    unsigned char  cp_name[17];     /* CP name                  */
    unsigned char  num_of_lus;      /* number of LUs           */
    unsigned char  lu_names[10][8]; /* LU names                 */
    unsigned char  wildcard_lus;    /* wildcard LUs            */
} DELETE_ADJACENT_LEN_NODE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_ADJACENT_LEN_NODE

cp_name

隣接 LEN ノード内の CP の完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

num_of_lus

1 ~ 10 の範囲の、削除する LU の数。LEN ノード定義全体を削除するには、ゼロを指定します。

lu_names

LEN ノード上の削除する LU の名前。それぞれの名前は 8 バイトからなるタイプ A の EBCDIC 文字ストリングで、右側に EBCDIC のスペースを入れます。LEN ノード定義全体を削除する場合 (*num_of_lus* がゼロの場合) は、LU 名を指定しないでください。

名前の最初の部分の文字のみを指定することにより、複数の LU 名に一致する「ワイルドカード」LU 名を指定できます。たとえば、ワイルドカード LU 名の APPN.LU は、APPN.LUNAME または APPN.LU01 と一致します (ただし、APPN.NAME.LU とは一致しません)。ただし、1 つの verb で指定するすべての LU 名は、次の *wildcard_lus* パラメーターで定義するのと同じタイプ (ワイルドカードか明示的) の LU 名でなければなりません。同じ LEN ノードから両方のタイプの LU 名を除去するには、複数の DELETE_ADJACENT_LEN_NODE verb を使用します。

wildcard_lus

指定する LU 名がワイルドカード・エンタリーであるか、それとも明示的な LU 名であるかを示します。値は次のいずれかです。

AP_YES 指定する LU 名はワイルドカード・エンタリーです。

AP_NO 指定する LU 名は明示的なエンタリーです。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

```
AP_INVALID_CP_NAME
    cp_name パラメーターに、有効でない文字が入っていました。
```

```
AP_INVALID_LU_NAME
    指定した LU 名の 1 つ以上に、有効でない文字が入っていました。
```

```
AP_INVALID_NUM_LUS
    num_of_lus パラメーターが、有効な範囲内にありませんでした。
```

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

```
AP_INVALID_CP_NAME
    指定した CP 名がありません。
```

```
AP_INVALID_LU_NAME
    指定した LU 名 (1 つ以上) がありません。
```

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_BACKUP

アプリケーションは、この verb を使用して、**sna.net** ファイルのバックアップ・マスター・サーバーのリストからサーバーを削除します。これにより、このサーバーはマスター構成ファイル・サーバーとして動作できなくなります。

DELETE_BACKUP

この verb を使用すると、削除するサーバーで SNA ソフトウェアが実行されているかどうかに関わらず、リスト内の任意のサーバー (マスター・サーバーを含む) を削除できます。ただし、リストには、SNA ソフトウェアを実行中のサーバーが常に 1 つ以上含まれている必要があります (このサーバーがマスター・サーバーとしての役割を継承します)。リスト内にサーバーが 1 つしかない場合や SNA ソフトウェアを実行中のサーバーが 1 つしかリストされていない場合は、サーバーを削除することはできません。

この verb は、**sna.net** ファイルに対して発行する必要があります。

VCB 構造体

```
typedef struct delete_backup
{
    AP_UINT16      opcode;          /* verb operation code      */
    unsigned char  reserv2;        /* reserved                  */
    unsigned char  format;        /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  backup_name[128]; /* name of server to delete */
    unsigned char  reserv3[4];     /* reserved                  */
} DELETE_BACKUP;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_BACKUP

backup_name

バックアップ・サーバーのリストから削除するサーバーの名前。

サーバー名に . (ピリオド) 文字が含まれている場合、Communications Server for Linux はこれが完全修飾名であると想定します。ピリオドが含まれていない場合は、DNS ルックアップを実行してサーバー名を判別します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_RECORD_NOT_FOUND

指定したサーバー名は、ファイルにリストされていません。

AP_CANT_DELETE_LAST_BACKUP

SNA ソフトウェアを実行中のサーバーが 1 つしかリストされていない (つまり、現在マスターとして機能している唯一のサーバーである) ため、リストからサーバーを削除できません。このサーバーの削除を試行する場合は、リストされている他の 1 つ以上のサーバーで SNA ソフトウェアを開始するか、または (ADD_BACKUP を使用して) 1 つ以上の新規バックアップ・サーバーを追加し、これらのサーバーで SNA ソフトウェアを開始します。

AP_INVALID_TARGET

NOF API 呼び出しのターゲット・ハンドルで構成ファイルまたはノードが指定されています。この verb は、**sna.net** ファイルに対して発行する必要があります。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_CN

DELETE_CN は、接続ネットワークを削除するか、選択したポートを接続ネットワークから削除します。

この verb はネットワーク・ノードまたはエンド・ノードでのみ有効であり、LEN ノードでは無効です。

VCB 構造体

```
typedef struct delete_cn
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code        */
    unsigned char  fqcn_name[17];  /* name of Connection Network   */
    unsigned char  reserv1;         /* reserved                      */
    AP_UINT16      num_ports;      /* number of ports to delete    */
    unsigned char  port_name[8][8]; /* names of ports to delete     */
} DELETE_CN;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_CN

fqcn_name

接続ネットワークの完全修飾名。この名前は、右側に EBCDIC のスペース

を入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

num_ports

接続ネットワークを削除するにはゼロを指定し、接続ネットワークを削除せずにポートを除去する場合は、削除するポートの数を指定します。

port_name

ポートを除去する場合 (*num_ports* がゼロ以外の場合) は、削除するポートの名前を指定します。それぞれのポート名は 8 バイトの ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。接続ネットワークを削除する場合 (*num_ports* がゼロの場合) は、これらの名前を 2 進ゼロに設定しなければなりません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_CN_NAME

指定された完全修飾 CN 名が、定義されたどの CN 名にも一致しませんでした。

AP_INVALID_NUM_PORTS_SPECIFIED

num_ports パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードが LEN ノードであるために *verb* が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは LEN ノードです。この *verb* は、ネットワーク・ノードまたはエンド・ノードでのみ有効です。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_COS

DELETE_COS はサービス・クラス・エントリーを削除します。ローカルで定義したサービス・クラスのみを削除できます。SNA によって定義されたデフォルトのサービス・クラスは削除できません。

ノードがモードから COS へのマッピング (DEFINE_NODE の *mode_to_cos_map_supp* パラメーターで定義します) をサポートしており、削除しようとしている COS へマップされるモードが構成に含まれている場合、Communications Server for Linux は、これらのモードをデフォルトの COS (DEFINE_MODE verb でヌルのモード名を使用して指定します) に再マップするか、デフォルトの COS が指定されていないければ、SNA 定義 COS の #CONNECT に再マップします。

VCB 構造体

```
typedef struct delete_cos
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                 */
    unsigned char  format;         /* reserved                 */
    AP_UINT16      primary_rc;      /* primary return code     */
    AP_UINT32      secondary_rc;    /* secondary return code   */
    unsigned char  cos_name[8];     /* class of service name   */
} DELETE_COS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_COS

cos_name

サービス・クラス名。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

DELETE_COS

secondary_rc

値は次のいずれかです。

AP_COS_NAME_NOT_DEFD

指定した名前は Communications Server for Linux システム上に定義されている COS の名前ではありません。

AP_SNA_DEFD_COS_CANT_BE_DELETED

指定した名前は SNA で定義されたサービス・クラスの名前であり、削除できません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_CPIC_SIDE_INFO

この verb は、サイド情報テーブルからエントリーを削除します。

この verb と CPI-C 関数 Delete_CPIC_Side_Information の違いに注意してください。この verb は構成ファイルを変更するため、すべての Communications Server for Linux CPI-C アプリケーションに影響を及ぼします。CPI-C 関数はサイド情報テーブルのメモリーにあるアプリケーション固有のコピーを変更し、他の CPI-C アプリケーションには影響を及ぼしません。

この verb は、ドメイン構成ファイルに対して発行しなければなりません。

VCB 構造体

```
typedef struct delete_cpic_side_info
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  reserv2a[8];    /* reserved                  */
    unsigned char  sym_dest_name[8]; /* Symbolic destination name */
} DELETE_CPIC_SIDE_INFO;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_CPIC_SIDE_INFO

sym_dest_name

サイド情報エントリーを識別する記号宛先名。これは大文字の A ~ Z および数字の 0 ~ 9 からなる 8 バイトの ASCII スtringで、必要に応じて右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

AP_INVALID_SYM_DEST_NAME

sym_dest_name パラメーターが、定義されている CPI-C サイド情報 エントリーの名前ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_DIRECTORY_ENTRY

DELETE_DIRECTORY_ENTRY は、ネットワーク・ディレクトリー内のエントリーを削除します。エンド・ノード CP のエントリーを、そのネットワーク・ノード・サーバーのディレクトリーから削除することはできません。

親子ソースのエントリーを削除すると、それに関連した子リソースのエントリーもすべて削除されます。たとえば、あるエンド・ノードの親であるネットワーク・ノードのエントリーを削除すると、そのネットワーク・ノードのエントリーが削除されるのみでなく、そのエンド・ノードのエントリーと、両方のノードに関連したすべての LU が (ワイルドカード LU エントリーも含め) 削除されます。

VCB 構造体

```
typedef struct delete_directory_entry
{
    AP_UINT16      opcode;           /* verb operation code */
    unsigned char  reserv2;         /* reserved */
    unsigned char  format;         /* reserved */
    AP_UINT16      primary_rc;     /* primary return code */
    AP_UINT32      secondary_rc;   /* secondary return code */
    unsigned char  resource_name[17]; /* fully qualified resource name */
    unsigned char  reserv3;       /* reserved */
    AP_UINT16      resource_type;  /* resource type */
} DELETE_DIRECTORY_ENTRY;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_DIRECTORY_ENTRY

resource_name

削除するリソースの完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

resource_type

削除するリソースのタイプを指定します。値は次のいずれかです。

AP_ENCP_RESOURCE

エンド・ノードまたは LEN ノード

AP_NNCP_RESOURCE

ネットワーク・ノード

AP_LU_RESOURCE

LU

AP_WILDCARD_LU_RESOURCE

ワイルドカード LU 名

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_FQ_LU_NAME

resource_name パラメーターが、定義されているディレクトリー・エントリーの名前ではありませんでした。

AP_INVALID_RESOURCE_TYPE

resource_type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

AP_CANT_DELETE_ADJ_ENDNODE

指定したエントリーはエンド・ノード・エントリーであり、この `verb` の発行先となったノードは、そのエンド・ノードのネットワーク・ノード・サーバーです。このエンド・ノード・エントリーを削除することはできません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_DLC

DELETE_DLC は DLC を削除します。この `verb` は、次のものも削除します。

- その DLC に関連したすべてのポート、リンク・ステーション、および接続ネットワーク TG。
- その DLC 上の各 LS に関連したすべての PU、それらの PU が所有するすべての LU、およびそれらの LU に関連したすべての LU-LU パスワード。

VCB 構造体

```
typedef struct delete_dlc
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  dlc_name[8];    /* name of DLC              */
} DELETE_DLC;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_DLC

dlc_name

削除する DLC の名前。これは 8 バイトからなる ASCII スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
```

AP_INVALID_DLC_NAME

指定した名前は Communications Server for Linux システム上に定義されている DLC の名前ではありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

AP_DLC_ACTIVE

その DLC は現在活動状態であるため、削除できません。削除を試みる前に、STOP_DLC verb を使用して DLC を停止してください。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_DOWNSTREAM_LU

この verb は、ダウンストリーム LU を削除するために使用します。

VCB 構造体

```
typedef struct delete_downstream_lu
{
    AP_UINT16          opcode;          /* verb operation code */
```

```

unsigned char    reserv2;           /* reserved          */
unsigned char    format;           /* reserved          */
AP_UINT16       primary_rc;       /* primary return code */
AP_UINT32       secondary_rc;     /* secondary return code */
unsigned char    dslu_name[8];     /* Downstream LU name  */
} DELETE_DOWNSTREAM_LU;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_DOWNSTREAM_LU

dslu_name

削除するダウンストリーム LU の名前。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側に EBCDIC のスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_NAME

dslu_name パラメーターに、有効でない文字が入っていました。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

dslu_name パラメーターが、定義されているどのダウンストリーム LU 名にも一致しませんでした。

AP_DSLU_ACTIVE

LU は現在活動状態であるため、削除できません。

DELETE_DOWNSTREAM_LU

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ノードの構成がこの verb をサポートしていないために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは SNA ゲートウェイをサポートしていません。これは、DEFINE_NODE verb の *pu_conc_support* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_DOWNSTREAM_LU_RANGE

この verb は、ダウンストリーム LU の範囲を削除するために使用します。

この verb に指定するパラメーターには、LU のベース名と NAU アドレスの範囲が含まれます。削除する LU 名は、そのベース名と NAU アドレスを結合して判別されます。たとえば、LUNME というベース名が 11 ~ 14 の NAU 範囲と結合され、LUNME011、LUNME012、LUNME013、LUNME014 という LU が削除されます。

指定した範囲の名前を持つすべての LU が削除されます。範囲内の名前が 1 つ以上存在しない場合でも、Communications Server for Linux はエラーを戻しません。

VCB 構造体

```
typedef struct delete_downstream_lu_range
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  dslu_base_name[5]; /* LU base name             */
    unsigned char  min_nau;        /* Minimum NAU address in range */
    unsigned char  max_nau;        /* Maximum NAU address in range */
} DELETE_DOWNSTREAM_LU_RANGE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_DOWNSTREAM_LU_RANGE

dslu_base_name

LU 名のベース名。これは 5 バイトからなる (英字で始まる) タイプ A の EBCDIC スtring で、ベース名が 5 文字に満たない場合、右側に EBCDIC のスペースを入れます。 Communications Server for Linux は、こ

の名前に 3 桁の 10 進値からなるそれぞれの NAU アドレスを付加することにより、削除する LU の名前を判別します。

min_nau

最初の LU の NAU アドレス (範囲は 1 ~ 255)。

max_nau

最後の LU の NAU アドレス (範囲は 1 ~ 255)。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_NAU_ADDRESS

min_nau パラメーターまたは *max_nau* パラメーターが、有効ではありませんでした。

AP_INVALID_LU_NAME

dslu_base_name パラメーターに、有効でない文字が入っていました。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

指定した範囲内の名前 で定義された LU はありません。

AP_DSLU_ACTIVE

範囲内の 1 つ以上の LU が、現在活動状態であるため削除できません。

DELETE_DOWNSTREAM_LU_RANGE

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ノードの構成がこの verb をサポートしていないために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは SNA ゲートウェイをサポートしていません。
これは、DEFINE_NODE verb の *pu_conc_support* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_DSPU_TEMPLATE

DELETE_DSPU_TEMPLATE verb は、以前に DEFINE_DSPU_TEMPLATE verb を使用して定義した特定のダウンストリーム物理装置 (DSPU) テンプレートを削除するか、DSPU テンプレートから 1 つ以上のダウンストリーム LU (DSL) を削除します。

VCB 構造体

```
typedef struct delete_dspu_template
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  template_name[8]; /* name of template        */
    AP_UINT16      num_of_dslu_templates; /* number of dslu templates */
    unsigned char  reserv1[10];    /* reserved                  */
} DELETE_DSPU_TEMPLATE;

typedef struct dslu_template
{
    unsigned char  min_nau;         /* Minimum NAU address in range */
    unsigned char  max_nau;         /* Maximum NAU address in range */
    unsigned char  allow_timeout;   /* Allow timeout of host LU?    */
    unsigned char  delayed_logon;   /* Allow delayed logon to host  */
    /* LU */
    unsigned char  reserv1[8];      /* reserved                    */
    unsigned char  host_lu[8];     /* Host LU or Pool name        */
} DSLU_TEMPLATE;
```

指定パラメーター

提供パラメーターは次のとおりです。

opcode AP_DELETE_DSPU_TEMPLATE

template_name

削除する DSPU テンプレートの名前、または削除する DSLU テンプレートを含む DSPU テンプレートの名前。1 ～ 8 文字のローカル・システム上で表示可能な文字を指定します。

num_of_dslu_templates

削除する DSLU テンプレートの数。1 ～ 255 の範囲で値を指定するか、DSPU テンプレート全体の削除を示す 0 (ゼロ) を指定してください。

削除されるそれぞれの DSLU テンプレートごとに、*num_of_dslu_templates* で指定された数まで、`DSLU_TEMPLATE` 構造体を `DELETE_DSPU_TEMPLATE` 構造体の終りに付加して、以下のパラメーターを収めます。

min_nau

削除する DSLU テンプレートの範囲内の最小 NAU アドレス。1 ～ 255 の範囲で値を指定します。

max_nau

削除する DSLU テンプレートの範囲内の最大 NAU アドレス。1 ～ 255 の範囲で値を指定します。

allow_timeout

ホスト LU 定義で指定されたタイムアウト期間にセッションが活動状態でないまま放置された場合、このダウンストリーム LU が使用しているホスト LU を Communications Server for Linux がタイムアウトにできるかどうかを指定します。値は次のいずれかです。

AP_YES Communications Server for Linux は、このダウンストリーム LU が使用しているホスト LU をタイムアウトにすることができます。

AP_NO Communications Server for Linux は、このダウンストリーム LU が使用しているホスト LU をタイムアウトにできません。

delayed_logon

Communications Server for Linux に、ダウンストリーム LU とホスト LU との接続をダウンストリーム LU から最初のデータを受信するまで遅らせるかどうかを指定します。接続を遅らせる場合は、シミュレート・ログオン画面がダウンストリーム LU へ送信されます。値は次のいずれかです。

AP_YES Communications Server for Linux は、ダウンストリーム LU とホスト LU との接続をダウンストリーム LU から最初のデータを受信するまで遅らせます。

AP_NO Communications Server for Linux は、ダウンストリーム LU とホスト LU との接続をダウンストリーム LU から最初のデータを受信するまで遅らせません。

host_lu 範囲内のすべてのダウンストリーム LU がマップされるホスト LU の名前またはホスト LU プールの名前。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

DELETE_DSPU_TEMPLATE

```
primary_rc
AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
AP_PARAMETER_CHECK
```

```
secondary_rc
値は次のいずれかです。
```

AP_INVALID_TEMPLATE_NAME

`template_name` パラメーターで指定したテンプレートが有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_FOCAL_POINT

`DELETE_FOCAL_POINT verb` は、指定した MS カテゴリのフォーカル・ポイント (そのカテゴリのメイン・フォーカル・ポイントまたはバックアップ・フォーカル・ポイント) の定義を除去します。定義したフォーカル・ポイント・アプリケーションが活動状態で、そのカテゴリの現行のフォーカル・ポイントとして機能している場合、Communications Server for Linux は、そのフォーカル・ポイントへ `MS_CAPABILITIES` メッセージを送信してそのフォーカル・ポイントを取り消し、フォーカル・ポイントとしてそれ以上機能しないようにします。

VCB 構造体

```
typedef struct delete_focal_point
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  reserved;      /* reserved                     */
    unsigned char  ms_category[8]; /* management services category */
    unsigned char  type;          /* type of focal point         */
} DELETE_FOCAL_POINT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

```
opcode AP_DELETE_FOCAL_POINT
```


ms_category

Management Services カテゴリー。これは、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照)の『MS Discipline-Specific Application Programs』の表に指定されたカテゴリー名の 1 つ (EBCDIC スペース (0x40) で埋め込まれている) か、またはユーザー定義のカテゴリーです。ユーザー定義カテゴリー名は、8 バイトからなるタイプ 1134 の EBCDIC ストリングで、必要に応じて EBCDIC のスペース (0x40) を入れます。

type

削除するフォーカル・ポイントのタイプを指定します。値は次のいずれかです。

AP_ACTIVE

現在活動状態であるフォーカル・ポイント (どのタイプでも構いません) を取り消します。

AP_IMPLICIT

暗黙の定義 (バックアップを AP_NO に設定した DEFINE_FOCAL_POINT を使用して定義したもの) を除去します。このフォーカル・ポイントは、現在活動状態であれば取り消されます。

AP_BACKUP

バックアップ定義 (バックアップを AP_YES に設定した DEFINE_FOCAL_POINT を使用して定義したもの) を除去します。このフォーカル・ポイントは、現在活動状態であれば取り消されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_CATEGORY_NAME

指定したカテゴリー名に、有効でない文字が入っていました。

AP_INVALID_TYPE

type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

DELETE_FOCAL_POINT

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは MS ネットワーク管理機能をサポートしていません。これは、DEFINE_NODE verb の *mds_supported* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_INTERNAL_PU

DELETE_INTERNAL_PU は DLUR からサービスを受ける ローカル PU を削除します。活動状態である SSCP-PU セッションがない PU のみを削除できます。

VCB 構造体

```
typedef struct delete_internal_pu
{
    AP_UINT16      opcode;          /* verb operation code      */
    unsigned char  reserv2;        /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  pu_name[8];     /* internal PU name         */
} DELETE_INTERNAL_PU;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_INTERNAL_PU

pu_name

削除する内部 PU の名前。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
```

AP_INVALID_PU_NAME

pu_name パラメーターが、定義されている内部 PU の名前ではありませんでした。

戻りパラメーター: 状態の検査

状態エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

値は次のいずれかです。

AP_PU_NOT_RESET

PU は、その PU にまだ活動状態である PU-SSCP セッションがあるため、削除できません。

AP_INVALID_PU_TYPE

指定した PU はリモート PU であり、内部 PU ではありません。

戻りパラメーター: サポートされていない機能

ノードの構成がこの *verb* をサポートしていないために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
```

AP_FUNCTION_NOT_SUPPORTED

ノードは DLUR をサポートしていません。これは、DEFINE_NODE の *dlur_support* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_LOCAL_LU

DELETE_LOCAL_LU *verb* はローカル LU を削除し、そのローカル LU に関連した LU-LU パスワードがあれば、それも削除します。

VCB 構造体

```
typedef struct delete_local_lu
{
    AP_UINT16          opcode;           /* verb operation code */
    unsigned char     reserv2;         /* reserved */
    unsigned char     format;         /* reserved */
}
```

DELETE_LOCAL_LU

```
AP_UINT16      primary_rc;      /* primary return code */
AP_UINT32      secondary_rc;    /* secondary return code */
unsigned char  lu_name[8];      /* local LU name */
} DELETE_LOCAL_LU;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_LOCAL_LU

lu_name

削除するローカル LU の名前。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtring で、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_CANT_DELETE_CP_LU

指定した LU 名がブランク (CP に関連した LU を示します) でした。この LU を削除することはできません。

AP_INVALID_LU_NAME

指定した LU 名は Communications Server for Linux システム上に定義されているローカル LU の名前ではありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_LS

DELETE_LS は、定義されているリンク・ステーション (LS) を削除します。この verb は、その LS に関連した PU、その PU が所有するすべての LU、およびそれらの LU に関連したすべての LU-LU パスワードも削除します。活動状態である LS を削除することはできません。

VCB 構造体

```
typedef struct delete_ls
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
    unsigned char  ls_name[8];      /* name of link station     */
} DELETE_LS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_LS

ls_name

削除するリンク・ステーションの名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 文字に満たない場合、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LINK_NAME

指定した LS 名に、有効でない文字が入っています。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

DELETE_LS

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_LS_ACTIVE

その LS は現在活動状態であるため、削除できません。

AP_INVALID_LINK_NAME

指定した LS 名は Communications Server for Linux システム上に定義されている LS の名前ではありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_LS_ROUTING

DELETE_LS_ROUTING verb は、前に DEFINE_LS_ROUTING verb を使用して定義したパートナー LU とリンク・ステーションの関連付けを削除します。

VCB 構造体

```
typedef struct delete_ls_routing
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;              /* reserved                  */
    unsigned char  format;              /* reserved                  */
    AP_UINT16      primary_rc;          /* primary return code      */
    AP_UINT32      secondary_rc;        /* secondary return code    */
    unsigned char  lu_name[8];          /* LU Name                   */
    unsigned char  lu_alias[8];         /* reserved                  */
    unsigned char  fq_partner_lu[17];   /* partner lu name          */
    unsigned char  wildcard_fqplu;     /* wildcard partner LU flag */
    unsigned char  reserv3[2];          /* reserved                  */
} DELETE_LS_ROUTING;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_LS_ROUTING

lu_name

パートナー LU (*fq_partner_lu* パラメーターで指定するもの) と通信したローカル LU の名前。1 ~ 8 文字のローカル・システム上で表示可能な文字を指定します。

fq_partner_lu

ローカル LU の LS 経路指定データから除去するパートナー LU の完全修飾名。1 ~ 8 文字のネットワーク名、1 つのピリオド、1 ~ 8 文字のパートナー LU 名の順序で続けた 3 ~ 17 個のローカル・システム上で表示可能な文字を指定します。

ワイルドカード・エントリーを削除するには、そのエントリーの定義に使用したのと同じワイルドカード LU 名を指定します。ワイルドカードを使用して、複数の明示的に定義したエントリーを削除することはできません。

wildcard_fqplu

fq_partner_lu パラメーターに、完全ワイルドカードまたは部分ワイルドカードが含まれているかどうかを示すワイルドカード・パートナー LU フラグ。このフラグはワイルドカード・エントリーを削除するために使用します。ワイルドカードを使用して、複数の明示的に定義したエントリーを削除することはできません。値は次のいずれかです。

AP_YES *fq_partner_lu* パラメーターには、ワイルドカード・エントリーが含まれています。

AP_NO *fq_partner_lu* パラメーターには、ワイルドカード・エントリーは含まれていません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LOCAL_LU

lu_name パラメーターに、有効でない文字が入っていました。

AP_INVALID_PARTNER_LU

fq_partner_lu パラメーターに、有効でない文字が入っていました。

AP_INVALID_WILDCARD_NAME

wildcard_fqplu パラメーターを AP_YES に設定しましたが、*fq_partner_lu* パラメーターは有効なワイルドカード名ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

DELETE_LS_ROUTING

secondary_rc

値は次のいずれかです。

AP_INVALID_LOCAL_LU

lu_name パラメーターが、既存の LS 経路指定レコードに一致しませんでした。

AP_INVALID_PARTNER_LU

fq_partner_lu パラメーターが、指定したローカル LU の既存の LS 経路指定レコードに一致しませんでした。

AP_INVALID_WILDCARD_NAME

wildcard_fqplu パラメーターを YES に設定しましたが、一致するエントリーがありませんでした。

AP_INVALID_RESOURCE_NAME

指定したパラメーターに一致する LS 経路指定エントリーがありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_LU62_TIMEOUT

DELETE_LU62_TIMEOUT verb は、前に DEFINE_LU62_TIMEOUT verb で定義した LU タイプ 6.2 セッションのタイムアウトの定義を削除します。

VCB 構造体

```
typedef struct delete_lu62_timeout
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;              /* reserved                  */
    unsigned char  format;               /* reserved                  */
    AP_UINT16      primary_rc;           /* primary return code      */
    AP_UINT32      secondary_rc;        /* secondary return code    */
    unsigned char  resource_type;        /* resource type             */
    unsigned char  resource_name[17];    /* resource name             */
} DELETE_LU62_TIMEOUT;
```

指定パラメーター

提供パラメーターは次のとおりです。

opcode AP_DELETE_LU62_TIMEOUT

resource_type

削除するタイムアウトのタイプを指定します。値は次のいずれかです。

AP_GLOBAL_TIMEOUT

ローカル・ノードのすべての LU 6.2 セッションに適用するタイムアウトを削除します。

AP_LOCAL_LU_TIMEOUT

resource_name パラメーターで指定したローカル LU のすべての LU 6.2 セッションに適用するタイムアウトを削除します。

AP_PARTNER_LU_TIMEOUT

resource_name パラメーターで指定したパートナー LU へのすべての LU 6.2 セッションに適用するタイムアウトを削除します。

AP_MODE_TIMEOUT

resource_name パラメーターで指定したモード上のすべての LU 6.2 セッションに適用するタイムアウトを削除します。

resource_name

タイムアウトを削除するリソースの名前。この値は次のいずれかです。

- *resource_type* を AP_GLOBAL_TIMEOUT に設定した場合は、このパラメーターを指定しないでください。
- *resource_type* を AP_LOCAL_LU_TIMEOUT に設定した場合は、ローカル LU 名として、1 ～ 8 文字のローカル・システム上で表示可能なタイプ A の文字を指定してください。
- *resource_type* を AP_PARTNER_LU_TIMEOUT に設定した場合は、パートナー LU の完全修飾名を指定するため、1 ～ 8 文字のネットワーク名、1 つのピリオド、1 ～ 8 文字のパートナー LU 名の順序で、17 文字からなるローカルに表示可能なタイプ A の文字を指定してください。
- *resource_type* を AP_MODE_TIMEOUT に設定した場合は、モード名として、1 ～ 8 文字のローカル・システム上で表示可能なタイプ A の文字を指定してください。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_RESOURCE_TYPE

resource_type パラメーターに指定した値が、有効ではありませんでした。

AP_INVALID_LU_NAME

resource_name パラメーターに指定した LU 名が有効ではありませんでした。

DELETE_LU62_TIMEOUT

AP_INVALID_PARTNER_LU

resource_name パラメーターに指定したパートナー LU 名が有効ではありませんでした。

AP_INVALID_MODE_NAME

resource_name パラメーターに指定したモード名が有効ではありませんでした。

AP_GLOBAL_TIMEOUT_NOT_DEFINED

resource_type パラメーターに AP_GLOBAL_TIMEOUT の値を指定しましたが、定義されているグローバル・タイムアウトがありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_LU_0_TO_3

この verb は、3270 エミュレーションまたは LUA に使用する LU (タイプ 0-3 の LU) を削除するために使用します。

VCB 構造体

```
typedef struct delete_lu_0_to_3
{
    AP_UINT16          opcode;          /* verb operation code      */
    unsigned char     reserv2;         /* reserved                  */
    unsigned char     format;         /* reserved                  */
    AP_UINT16         primary_rc;     /* primary return code      */
    AP_UINT32         secondary_rc;   /* secondary return code    */
    unsigned char     lu_name[8];     /* LU name                   */
} DELETE_LU_0_TO_3;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_LU_0_TO_3

lu_name

削除するローカル LU の名前。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtring で、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
```

```
    AP_INVALID_LU_NAME
```

指定した LU 名に、有効でない文字が入っていました。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

```
    AP_INVALID_LU_NAME
```

指定した LU 名は Communications Server for Linux システム上に定義されている LU の名前ではありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_STATE_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_LU_0_TO_3_RANGE

この `verb` は、3270 エミュレーションまたは LUA (タイプ 0-3 の LU) に使用する LU の範囲を削除するために使用します。

この `verb` に指定するパラメーターには、LU のベース名と NAU アドレスの範囲が含まれます。削除する LU 名は、そのベース名と NAU アドレスを結合して判別されます。たとえば、11 ~ 14 の範囲の NAU と結合された LUNME というベース名は、LUNME011、LUNME012、LUNME013、LUNME014 という LU を削除します。

指定した範囲の名前を持つすべての LU が削除されます。範囲内の名前が 1 つ以上存在しない場合でも、Communications Server for Linux はエラーを戻しません。

VCB 構造体

```
typedef struct delete_lu_0_to_3_range
{
    AP_UINT16          opcode;          /* verb operation code */
```

DELETE_LU_0_TO_3_RANGE

```
    unsigned char    reserv2;        /* reserved */
    unsigned char    format;        /* reserved */
    AP_UINT16        primary_rc;    /* primary return code */
    AP_UINT32        secondary_rc;  /* secondary return code */
    unsigned char    base_name[6];  /* Base name */
    unsigned char    min_nau;       /* Minimum NAU address in range */
    unsigned char    max_nau;       /* Maximum NAU address in range */
    unsigned char    name_attributes; /* Extension type */
    unsigned char    base_number;    /* First extension number */
    unsigned char    reserv5[16];   /* reserved */
} DELETE_LU_0_TO_3_RANGE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_LU_0_TO_3_RANGE

base_name

LU 名のベース名。これは、(英字で始まる) タイプ A の EBCDIC ストリングで、ベース名が 6 文字に満たない場合、右側に EBCDIC のスペースを入れます。これは、*name_attributes* パラメーターで決められた 5 バイトまたは 6 バイトになる場合があります。 Communications Server for Linux は、この名前に各 NAU アドレスの 10 進値 (または *base_number* パラメーターから始まる範囲にある数) を付加することにより、削除する LU の名前を判別します。

min_nau

最初の LU の NAU アドレス (範囲は 1 ~ 255)。

max_nau

最後の LU の NAU アドレス (範囲は 1 ~ 255)。

name_attributes

LU の拡張子のタイプを指定します。値は次のいずれかです。

AP_NONE

LU 名には NAU 番号に対応する番号があります。それらの番号は 10 進数で指定され、*base_name* パラメーターは必ず 5 文字にします。

AP_USE_BASE_NUMBER

この範囲内の LU の削除を *base_number* パラメーターで指定した値から開始します。

AP_USE_HEX_IN_NAME

LU 名の拡張子は、10 進数でなく 16 進数です。この値を指定した場合は、*base_name* パラメーターに 6 文字まで指定できます。

base_number

name_attributes パラメーターに AP_USE_BASE_NUMBER を指定した場合は、範囲内の LU の削除を開始する番号を指定します。この値は、*min_nau* パラメーター値の代わりに使用されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_NAU_ADDRESS

`min_nau` パラメーターまたは `max_nau` パラメーターが、有効ではありませんでした。

AP_INVALID_LU_NAME

`base_name` パラメーターに、有効でない文字が入っていました。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

AP_INVALID_LU_NAME

指定した範囲内の名前 で定義された LU はありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_STATE_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_LU_LU_PASSWORD

`DELETE_LU_LU_PASSWORD` は、ローカル LU に関連した LU-LU パスワードを削除します。LU-LU パスワードは、ローカル LU を削除したときに自動的に削除されます。この `verb` を使用する必要があるのは、パスワードのみを除去し、LU を構成した状態にしておく必要がある場合のみです。

VCB 構造体

```
typedef struct delete_lu_lu_password
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  lu_name[8];     /* LU name                     */
    unsigned char  lu_alias[8];   /* local LU alias              */
    unsigned char  fqplu_name[17]; /* fully qualified partner LU name */
    unsigned char  reserv3;       /* reserved                     */
} DELETE_LU_LU_PASSWORD;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_LU_LU_PASSWORD

lu_name

Communications Server for Linux に対して定義されているローカル LU の LU 名。これは 8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。LU が LU 名でなく LU 別名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

lu_alias

Communications Server for Linux に対して定義されているローカル LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。これは、*lu_name* をゼロに設定した場合にのみ使用します。

CP へ関連付けられている LU (デフォルト LU) を示すには、*lu_name* と *lu_alias* の両方を 8 桁の 2 進ゼロに設定します。

fqplu_name

Communications Server for Linux に対して定義されているパートナー LU の完全修飾 LU 名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PLU_NAME

fqplu_name パラメーターが、有効ではありませんでした。

AP_INVALID_LU_NAME

lu_name パラメーターが、有効ではありませんでした。

AP_INVALID_LU_ALIAS

lu_alias パラメーターが、有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_LU_POOL

DELETE_LU_POOL は、次のいずれかを行うために使用します。

- プールから 1 つ以上の LU を除去する。
- プールからすべての LU を除去し、そのプールを削除する。

この verb は LU を削除しません。LU は定義されたままですが、どのプールにも関連付けられません。

VCB 構造体

```
typedef struct delete_lu_pool
{
    AP_UINT16          opcode;           /* verb operation code      */
    unsigned char      reserv2;         /* reserved                 */
    unsigned char      format;          /* reserved                 */
    AP_UINT16          primary_rc;      /* primary return code     */
    AP_UINT32          secondary_rc;    /* secondary return code   */
    unsigned char      pool_name[8];    /* LU pool name            */
    AP_UINT16          num_lus;         /* Number of specified LUs */
    unsigned char      lu_names[10][8]; /* LU names                */
} DELETE_LU_POOL;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_LU_POOL

pool_name

LU プールの名前。これは 8 バイトからなる EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

num_lus

除去する LU の数 (*lu_names* リスト内の LU 名の数)。プール自体を削除

DELETE_LU_POOL

せずにプールから LU を除去する場合、この範囲は 1 ～ 10 です。プールからすべての LU を除去し、そのプールも削除するには、ゼロを指定します。

lu_names

プール自体を削除せずにプールから 1 つ以上の LU を除去するには、除去する LU の名前を指定します。指定する名前の数は、*num_lus* パラメーターに一致しなければなりません。それぞれの名前は 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側に EBCDIC のスペースを入れます。

プールからすべての LU を除去し、そのプールも削除するために *num_lus* をゼロに設定した場合は、このパラメーターを使用しません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_POOL_NAME
指定したプール名が、有効ではありませんでした。

AP_INVALID_LU_NAME
指定した LU 名の 1 つ以上が、プール内の LU の名前に一致しませんでした。

AP_INVALID_NUM_LUS
指定した *num_lus* パラメーターが、有効な範囲内にありませんでした。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_MODE

DELETE_MODE は、モードの定義を削除します。SNASVCMG および CPSVCMG などの SNA で定義されたモードを削除することはできません。

VCB 構造体

```
typedef struct delete_mode
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
    unsigned char  mode_name[8];    /* mode name                 */
} DELETE_MODE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_MODE

mode_name

モードの名前。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_CP_OR_SNA_SVCMG_UNDELETABLE

指定したモード名は SNA で定義されたモード名の 1 つであり、削除できません。

AP_MODE_NAME_NOT_DEFD

指定したモード名は Communications Server for Linux システム上に定義されているモードの名前ではありません。

AP_DEL_MODE_DEFAULT_SPCD

指定したモードは DEFINE_DEFAULTS *verb* でデフォルト・モードとして定義されているため、削除できません。

AP_MODE_UNDELETABLE

指定したモード名は SNA で定義されたモード名の 1 つであり、削除できません。

DELETE_MODE

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_PARTNER_LU

DELETE_PARTNER_LU verb は、パートナー LU 定義を削除します。

VCB 構造体

```
typedef struct delete_partner_lu
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;          /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  fqplu_name[17]; /* fully qualified partner LU name */
} DELETE_PARTNER_LU;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_PARTNER_LU

fqplu_name

削除するパートナー LU の完全修飾 LU 名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK
secondary_rc

AP_INVALID_PLU_NAME

指定した *fqplu_name* パラメーターが、定義されているどのパートナー LU 名にも一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_PORT

DELETE_PORT は、ポートを削除します。この verb は、次のものも削除します。

- そのポートに関連したすべてのリンク・ステーションと接続ネットワーク TG。
- そのポート上の各 LS に関連したすべての PU、それらの PU が所有するすべての LU、およびそれらの LU に関連したすべての LU-LU パスワード。

ポートは、この verb を発行するときに非活動状態になっていなければなりません。

VCB 構造体

```
typedef struct delete_port
{
    AP_UINT16      opcode;          /* verb operation code      */
    unsigned char  reserv2;        /* reserved                  */
    unsigned char  format;        /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  port_name[8];   /* name of port              */
} DELETE_PORT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_PORT

port_name

削除するポートの名前。この名前は、8 バイトからなる ASCII 文字ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

DELETE_PORT

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
```

AP_INVALID_PORT_NAME

指定したポート名は Communications Server for Linux システム上に定義されているポートの名前ではありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

AP_PORT_ACTIVE

指定したポートは現在活動状態であるため、削除できません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_RCF_ACCESS

DELETE_RCF_ACCESS は、前に DEFINE_RCF_ACCESS を使用して指定した Communications Server for Linux リモート・コマンド機能 (RCF) へのアクセスを防止します。RCF について詳しくは、「*Communications Server for Linux 管理ガイド*」を参照してください。

この verb は SPCF と UCF の両方へのアクセスを防止します。一方へのアクセスを許可し、他方へのアクセスを防止するには、DEFINE_RCF_ACCESS を使用します。

この verb は、ドメイン構成ファイルに対して発行しなければなりません。Communications Server for Linux は、ノードの始動時に RCF アクセス・パラメーターに基づいて動作します。ノードの実行中に RCF アクセスを削除した場合、ノードが実行されているサーバー上では、この変更はノードを停止して再始動するまで有効になりません。

VCB 構造体

```
typedef struct delete_rcf_access
{
    AP_UINT16          opcode;          /* Verb operation code */
```

```

        unsigned char    reserv2;        /* reserved          */
        unsigned char    format;        /* reserved          */
        AP_UINT16        primary_rc;    /* primary return code */
        AP_UINT32        secondary_rc;  /* secondary return code */
    } DELETE_RCF_ACCESS;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

```
opcode AP_DELETE_RCF_ACCESS
```

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_SECURITY_ACCESS_LIST

DELETE_SECURITY_ACCESS_LIST は、次のいずれかを行うために使用します。

- セキュリティー・アクセス・リストを削除する。
- セキュリティー・アクセス・リストから 1 人または複数のユーザーを削除するが、リストは構成したままにする。

ユーザー名は、そのユーザー名を使用してセットアップした活動状態である会話があるかどうかとは無関係に、セキュリティ・アクセス・リストから削除できます。ユーザー名を削除しても活動状態である会話に影響はありませんが、呼び出し側プログラムは、それ以降、削除されたユーザー名を使用して会話をセットアップすることはできなくなります。

VCB 構造体

DELETE_SECURITY_ACCESS_LIST verb には、可変数の security_user_name 構造体が含まれており、それらの構造体で、セキュリティ・アクセス・リストから削除するユーザー名を定義しています。ユーザー名構造体は、delete_security_access_list 構造体の末尾に組み込まれます。それらの構造体の数は、num_users パラメーターで指定します。

```

typedef struct delete_security_access_list
{
    AP_UINT16        opcode;            /* verb operation code */
    unsigned char    reserv2;          /* reserved             */
    unsigned char    format;          /* reserved             */
    AP_UINT16        primary_rc;      /* primary return code  */
    AP_UINT32        secondary_rc;    /* secondary return code */
    unsigned char    list_name[14];   /* name of this list    */
    unsigned char    reserv3[2];      /* reserved             */
    AP_UINT32        num_users;       /* number of users to delete */
} DELETE_SECURITY_ACCESS_LIST;

```

DELETE_SECURITY_ACCESS_LIST

```
typedef struct security_user_name
{
    unsigned char    user_name[10];          /* user name to delete      */
} SECURITY_USER_NAME;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_SECURITY_ACCESS_LIST

list_name

削除するセキュリティー・アクセス・リストの名前、または、削除するユーザー名を記載したリストの名前。これは 1 ~ 14 文字の ASCII ストリングで、名前が 14 文字に満たない場合は、右側にスペースを入れます。この名前は、既に定義されているセキュリティー・アクセス・リスト名と一致しなくてはなりません。

num_users

セキュリティー・アクセス・リストから削除するユーザー名の数。以下のように指定します。

- リストから 1 つ以上のユーザー名を削除し、その他のユーザー名は構成したままにしておくには、削除するユーザー名数を指定する。それぞれのユーザー名は、以下で説明するように、ユーザー名構造体で定義しなければなりません。
- セキュリティー・アクセス・リスト全体を削除するには、このパラメーターにゼロを指定し、ユーザー名は指定しない。

削除されるそれぞれのユーザー名ごとに、*num_users* で指定された数まで、SECURITY_USER_NAME 構造体を DELETE_SECURITY_ACCESS_LIST 構造体の終りに付加して、以下のパラメーターを収めます。

user_name

削除するユーザー名。この名前は、このセキュリティー・アクセス・リストに現在定義されているユーザー名と一致していなければなりません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_LIST_NAME

指定したセキュリティー・アクセス・リスト名が、セキュリティー・アクセス・リスト名として定義されていませんでした。

AP_INVALID_USER_NAME

指定したユーザー名の 1 つ以上が、このセキュリティー・アクセス・リストに定義されたユーザー名に一致しませんでした。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_TN3270_ACCESS

DELETE_TN3270_ACCESS は、次のいずれかを行うために使用します。

- TN3270 サーバー・ユーザーを削除し、そのユーザーが TN サーバーを使用してホストへアクセスできないようにする。
- ユーザーのセッションの 1 つ以上を削除するが、ユーザー自体は構成した状態にしておく。

VCB 構造体

```
typedef struct delete_tn3270_access
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    AP_UINT16      default_record; /* is this the DEFAULT record? */
    unsigned char  client_address[256]; /* address of TN3270 user  */
    AP_UINT32      num_sessions;   /* number of sessions to delete */
    unsigned char  delete_options; /* delete all sessions / delete user? */
} DELETE_TN3270_ACCESS;

typedef struct tn3270_session_name
{
    AP_UINT16      port_number;     /* TCP/IP port num of session to delete */
    unsigned char  listen_local_address[46]; /* Local addr client connects to */
} TN3270_SESSION_NAME;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_TN3270_ACCESS

default_record

この verb では、TCP/IP アドレスによって明示的に識別されない TN3270 ユーザーが使用するデフォルトの TN3270 ユーザーのレコードを参照するかどうかを指定します (このレコードを削除すると、それらのユーザーは TN サーバーへアクセスできなくなります)。値は次のいずれかです。

DELETE_TN3270_ACCESS

AP_YES この verb でデフォルトの TN3270 ユーザーのレコードを参照します。 *client_address* パラメーターは予約済みです。

AP_NO この verb で通常の TN3270 ユーザーのレコードを参照します。

client_address

DEFINE_TN3270_ACCESS verb での指定と同様に、削除する TN3270 ユーザーの TCP/IP アドレス。これはヌル終了 ASCII スtringであり、指定できるのは以下のいずれかです。

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

num_sessions

次のように削除するセッションの数。

- ユーザーのセッションの 1 つ以上を削除し、その他のセッションを構成した状態にしておくには、削除するセッションの数を指定します。それぞれのセッションは、次で述べるようにそのセッションの TCP/IP ポート番号によって定義しなければなりません。
- すべてのセッションを削除するか、ユーザーを削除するには、このパラメーターにゼロを指定し、TCP/IP ポート番号を組み込まないでください。次の *delete_options* パラメーターに、必要な削除のタイプを指定してください。

delete_options

num_sessions パラメーター (上記参照) がゼロでない場合、このパラメーターは無視されます。 *num_sessions* がゼロの場合は、次のいずれかの値を指定してください。

AP_ALL_SESSIONS

すべてのセッションを削除しますが、TN3270 ユーザーは構成した状態にしておきます。

AP_DELETE_USER

ユーザーと、そのユーザーのすべてのセッションを削除します。

削除されるそれぞれのセッションごとに、*num_sessions* で指定された数まで、TN3270_SESSION_NAME 構造体を DELETE_TN3270_ACCESS 構造体の終りに付加して、以下のパラメーターを収めます。

tn3270_session_name.port_number

セッションに使用する TCP/IP ポート番号。この番号は、この TN3270 ユーザーに定義したポート番号に一致しなければなりません。

tn3270_session_name.listen_local_address

TN3270 クライアントの接続先であるローカル TN サーバー・コンピュータ上のアドレス。

- セッションを構成するときにこのパラメーターを指定しない場合は、すべて 2 進ゼロを指定します。

- セッションを構成するときにアドレスを指定した場合は、このパラメーターには同じアドレスを指定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_CLIENT_ADDRESS

指定したクライアント・アドレスが、どの TN3270 ユーザーに定義された TCP/IP アドレスとも一致しませんでした。

AP_INVALID_PORT_NUMBER

指定した TCP/IP ポート番号が、このユーザーに定義されているどの TCP/IP ポート番号とも一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_TN3270_ASSOCIATION

DELETE_TN3270_ASSOCIATION は、指定したディスプレイ LU 名に基づいて、ディスプレイ LU とプリンター LU 間の関連付けを削除します。

VCB 構造体

```
typedef struct delete_tn3270_association
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;              /* reserved                      */
    unsigned char  format;               /* reserved                      */
    AP_UINT16      primary_rc;           /* primary return code          */
    AP_UINT32      secondary_rc;        /* secondary return code        */
    unsigned char  display_lu_name[8];   /* Display LU name              */
} DELETE_TN3270_ASSOCIATION;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_TN3270_ASSOCIATION

display_lu_name

関連付けを削除するディスプレイ LU の名前を指定します。これは、右側に EBCDIC のスペースを入れた EBCDIC ストリングです。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LU_NAME

ディスプレイ LU 名が、有効な EBCDIC ストリングではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

AP_INVALID_LU_NAME

指定したディスプレイ LU に定義されている関連付けはありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_TN_REDIRECT

DELETE_TN_REDIRECT は、TN リダイレクター・ユーザーを削除して、このユーザーが TN リダイレクターを使用してホストにアクセスできないようにするために使用します。

VCB 構造体

```
typedef struct delete_tn_redirect
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    TN_REDIRECT_ADDRESS addr;       /* Uniquely defines record      */
} DELETE_TN_REDIRECT;

typedef struct tn_redirect_address
{
    AP_UINT16      default_record;   /* Is this the default record ? */
    unsigned char  address_format;   /* IP address or fully-qualified name */
    unsigned char  client_address[256]; /* Client address                */
    AP_UINT16      port_number;      /* Port number that client connects on */
    unsigned char  listen_local_address[46]; /* Local addr client connects to */
    unsigned char  reserved[34];     /* reserved                      */
} TN_REDIRECT_ADDRESS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_TN_REDIRECT

addr.default_record

この *verb* が、TCP/IP アドレスによって明示的に識別されない TN リダイレクター・ユーザーが使用するデフォルトの TN リダイレクター・ユーザーのレコードを参照するかどうかを指定します (このレコードを削除すると、それらのユーザーは TN リダイレクターへアクセスできないことになります)。値は次のいずれかです。

AP_YES この *verb* は、デフォルトのレコードを参照します。 *client_address* パラメーターと *address_format* パラメーターは予約済みです。

AP_NO この *verb* は、通常の TN リダイレクター・ユーザー・レコードを参照します。

addr.address_format

client_address パラメーターのフォーマットを指定します。値は次のいずれかです。

AP_ADDRESS_IP

IP アドレス (IPv4 または IPv6 のいずれか)

AP_ADDRESS_FQN

別名または完全修飾名

addr.client_address

Telnet クライアントを実行するコンピューターの TCP/IP アドレス。これは

DELETE_TN_REDIRECT

ヌル終了 ASCII スtringです。以下のいずれかの値を指定できます。
address_format パラメーターで IP アドレスか、または名前のいずれかを指示します。

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

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

addr.port_number

Telnet クライアントが TN サーバー・ノードにアクセスするために使用するサーバーの TCP/IP ポートの数。

addr.listen_local_address

TN3270 クライアントの接続先であるローカル TN サーバー・コンピュータ上のアドレス。

- TN リダイレクト・レコードを構成するときにこのパラメーターを指定しない場合は、このパラメーターにはすべて 2 進ゼロを指定します。
- TN リダイレクト・レコードを構成するときにこのアドレスを指定した場合は、このパラメーターには同じアドレスを指定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc  
AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc  
AP_PARAMETER_CHECK
```

```
secondary_rc
```

AP_INVALID_CLIENT_ADDRESS

指定されたアドレッシング情報が、定義されているどの TN リダイレクター・ユーザーとも一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_TP

DELETE_TP は、TP 定義を削除します。

VCB 構造体

```
typedef struct delete_tp
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
    unsigned char  tp_name[64];     /* TP name                   */
} DELETE_TP;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_TP

tp_name

削除する TP の名前。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_TP_NAME

tp_name パラメーターが、定義されている TP の名前に一致しませんでした。

AP_SYSTEM_TP_CANT_BE_DELETED

指定した TP 名は Communications Server for Linux が内部で使用する TP の名前であり、この TP を削除することはできません。

DELETE_TP

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_TP_LOAD_INFO

DELETE_TP_LOAD_INFO verb は、TP ロード情報エントリーを削除します。

VCB 構造体

```
typedef struct delete_tp_load_info
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
    unsigned char  tp_name[64];     /* TP name                   */
    unsigned char  lu_alias[8];     /* LU alias                  */
} DELETE_TP_LOAD_INFO;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_TP_LOAD_INFO

tp_name

削除する TP ロード情報エントリーの TP 名。これは 64 バイトからなる EBCDIC スtringで、名前が 64 文字に満たない場合は、右側にスペースを入れます。

lu_alias

削除する TP ロード情報エントリーの LU 別名。これは 8 バイトからなる ASCII スtringで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

このパラメーターは、TP が APPC アプリケーションである場合にのみ使用できます。TP が CPI-C アプリケーションである場合は予約済みです。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_TP_NAME

tp_name パラメーターが、定義されている TP の名前に一致しませんでした。

AP_INVALID_LU_ALIAS

lu_alias パラメーターが、指定した TP 名の TP ロード情報エントリについて指定されている定義された LU 別名に一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DELETE_USERID_PASSWORD

DELETE_USERID_PASSWORD は、ユーザー ID に関連したパスワードを削除するか、ユーザー ID とパスワードについてのプロフィールを除去します。

VCB 構造体

```
typedef struct delete_userid_password
{
    AP_UINT16          opcode;           /* verb operation code      */
    unsigned char     reserv2;          /* reserved                  */
    unsigned char     format;           /* reserved                  */
    AP_UINT16         primary_rc;       /* primary return code      */
    AP_UINT32         secondary_rc;     /* secondary return code    */
    AP_UINT16         delete_type;      /* type of delete           */
    unsigned char     user_id[10];      /* user id                   */
    USERID_PASSWORD_CHARS password_chars; /* password characteristics */
} DELETE_USERID_PASSWORD;

typedef struct userid_password_chars
{
    unsigned char     description[32];   /* resource description     */
    unsigned char     reserv2[16];      /* reserved                  */
    AP_UINT16         profile_count;    /* number of profiles      */
    AP_UINT16         reserv1;          /* reserved                  */
    unsigned char     password[10];     /* password                  */
    unsigned char     profiles[10][10]; /* profiles                  */
} USERID_PASSWORD_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_DELETE_USERID_PASSWORD

delete_type

この verb の使用方法を指定します。値は次のいずれかです。

DELETE_USERID_PASSWORD

AP_REMOVE_USER

ユーザー、パスワード、およびすべての関連するプロファイルを削除します。

AP_REMOVE_PROFILES

指定したプロファイルを削除します。

user_id ユーザー ID。これは 10 バイトからなるタイプ AE の EBCDIC 文字ストリングで、名前が 10 文字に満たない場合は、右側に EBCDIC のスペースを入れます。

password_chars.description

このパラメーターは無視されます。

password_chars.profile_count

削除するプロファイルの数。*delete_type* を AP_REMOVE_USER に設定した場合、このパラメーターは予約済みです。

password_chars.password

このパラメーターは無視されます。

password_chars.profiles

ユーザーに関連したプロファイル。各プロファイル名は 10 バイトからなるタイプ AE の EBCDIC 文字ストリングで、プロファイル名が 10 文字に満たない場合は、右側に EBCDIC のスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_NO_PROFILES

delete_type パラメーターを AP_REMOVE_PROFILES に設定しましたが、プロファイルを指定しませんでした。

AP_UNKNOWN_USER

user_id パラメーターが、定義されているユーザー ID に一致しませんでした。

AP_INVALID_UPDATE_TYPE

delete_type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

DISCONNECT_NODE

アプリケーションは、ノードへの NOF verb の発行が終了したときに、この verb を使用して Communications Server for Linux ノードへのアプリケーションのハンドルを解放します。アプリケーションが切断を指定したノードは、コールでの *target_handle* パラメーターによって識別されます。この verb が正常に完了したあと、そのノードを識別するターゲット・ハンドルは有効でなくなります。

アプリケーションは終了する前に、オープンしているノード・ハンドルに対して必ず DISCONNECT_NODE を発行し、そのアプリケーションへ関連付けられているリソースを Communications Server for Linux が解放できるようにしてください。

この verb は、実行中のノード、ノードを実行していないサーバーのターゲット・ハンドルを解放するために発行されます。

VCB 構造体

```
typedef struct disconnect_node
{
    AP_UINT16          opcode;           /* Verb operation code          */
    unsigned char     reserv2;          /* reserved                      */
    unsigned char     format;          /* reserved                      */
    AP_UINT16         primary_rc;      /* Primary return code          */
    AP_UINT32         secondary_rc;    /* Secondary return code        */
} DISCONNECT_NODE;
```

指定パラメーター

opcode AP_DISCONNECT_NODE

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_STATE_CHECK

DISCONNECT_NODE

secondary_rc

AP_VERB_IN_PROGRESS

指定したターゲット・ハンドルは、前にそのハンドルに対して発行した *verb* が未処理なので、解放できません。ターゲット・ハンドルのすべての *verb* は、そのノードからの切断を試みる前に完了していなければなりません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

INIT_NODE

この *verb* は、以前定義したノードを始動します。アプリケーションでは、最初に CONNECT_NODE を発行してノードのターゲット・ハンドルを取得してから、そのターゲット・ハンドルを INIT_NODE コールに使用して、始動するノードを識別します。

この *verb* は、ノードを実行していないサーバーに対して発行しなければなりません。

VCB 構造体

```
typedef struct init_node
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
} INIT_NODE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_INIT_NODE

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_NODE_NAME

構成ファイルに指定したノード名が、`verb` の発行先に指定した Communications Server for Linux コンピューターの名前に一致しません。

AP_NOT_SERVER

構成ファイルで指定したノード名は Communications Server for Linux コンピューターの名前と一致しますが、指定したコンピューターは (サーバーではなく) クライアントであるため、ノードを実行できません。

AP_DLUR_NOT_SUPPORTED

ノードの構成で DLUR のサポートを指定しましたが、ノードは LEN ノードとして定義されています。DLUR は LEN ノード上ではサポートされません。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_NODE_ALREADY_STARTED

指定したノードは、既に始動されています。

AP_RESOURCE_NOT_LOADED

ノードの構成をロードしようとしたときに、Communications Server for Linux が 1 つ以上のエラーを検出したため、ノードは始動されませんでした。エラー・ログ・ファイルを検査して、エラーの詳細を示すメッセージを参照してください。

AP_INVALID_VERSION

Communications Server for Linux ソフトウェアのコンポーネント間にバージョンの不一致があったため、ノードが始動されませんでした。追加機能または追加ユーザーを組み込むために Communications Server for Linux ライセンスをアップグレードした場合は、正しいバージョンのライセンス・ソフトウェアを使用しているかどうかを検査してください。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

INITIALIZE_SESSION_LIMIT

INITIALIZE_SESSION_LIMIT verb は、ローカル LU、パートナー LU、モードの組み合わせについて、セッション限度を初期化します。

この verb は、ACTIVATE_SESSION verb を発行する前に発行しなければなりません。

この verb はクライアント上で実行中の NOF アプリケーションから発行することができます。この verb を AIX または Linux クライアント上で実行するには、NOF アプリケーションは root のユーザー ID か、あるいは sys グループ (AIX) または sna グループ (Linux) のメンバーのユーザー ID を使用して実行しなければなりません。

VCB 構造体

```
typedef struct initialize_session_limit
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;                /* reserved                  */
    AP_UINT16      primary_rc;            /* primary return code      */
    AP_UINT32      secondary_rc;          /* secondary return code    */
    unsigned char  lu_name[8];            /* local LU name             */
    unsigned char  lu_alias[8];           /* local LU alias           */
    unsigned char  plu_alias[8];          /* partner                   */
    unsigned char  fqplu_name[17];        /* fully qualified partner   */
                                           /* LU name                   */
    unsigned char  reserv3;               /* reserved                  */
    unsigned char  mode_name[8];          /* mode name                 */
    unsigned char  reserv3a;              /* reserved                  */
    unsigned char  set_negotiable;         /* set max negotiable limit? */
    AP_UINT16      plu_mode_session_limit; /* session limit            */
    AP_UINT16      min_conwinners_source; /* minimum source contention */
                                           /* winner sessions          */
    AP_UINT16      min_conwinners_target; /* minimum target contention */
                                           /* winner sessions          */
    AP_UINT16      auto_act;              /* auto activation limit    */
    unsigned char  reserv4[4];            /* reserved                  */
    AP_UINT32      sense_data;            /* sense data                */
} INITIALIZE_SESSION_LIMIT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_INITIALIZE_SESSION_LIMIT

lu_name

Communications Server for Linux に対して定義されているローカル LU の LU 名。これは 8 バイトからなるタイプ A の EBCDIC ストリングで、名

前が 8 バイトに満たない場合は、右側にスペースを入れます。LU が LU 名でなく LU 別名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

lu_alias

Communications Server for Linux に対して定義されているローカル LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。これは、*lu_name* をゼロに設定した場合にのみ使用します。

CP へ関連付けられている LU (デフォルト LU) を示すには、*lu_name* と *lu_alias* の両方を 8 桁の 2 進ゼロに設定します。

plu_alias

パートナー LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。パートナー LU が LU 別名でなく完全修飾 LU 名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

fqplu_name

Communications Server for Linux に対して定義されているパートナー LU の完全修飾 LU 名。このパラメーターは、*plu_alias* フィールドをゼロに設定した場合のみ使用し、*plu_alias* を指定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

mode_name

LU が使用するモードの名前。これは 8 バイトの英数字からなる (英文字で始まる) タイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。

set_negotiable

このモードの DEFINE_MODE で定義した折衝可能な最大セッション限度を変更するかどうかを指定します。値は次のいずれかです。

AP_YES *plu_mode_session_limit* に指定した値を、この LU-LU モードの組み合わせの折衝可能な最大セッション限度として使用します。

AP_NO 折衝可能な最大セッション限度を、そのモードに指定した値のままにしておきます。

plu_mode_session_limit

この LU-LU モードの組み合わせについて要求する合計セッション限度。つまり、このモードを使用するこれら 2 つの LU の間で許可される並列セッションの最大数。1 ~ 32,767 の範囲で値を指定します (値は、ローカル LU に対して DEFINE_LOCAL_LU verb で指定したセッション限度を超えてはなりません)。この値は、パートナー LU と折衝される場合があります。

INITIALIZE_SESSION_LIMIT

min_conwinners_source

ローカル LU がコンテンション勝者である、このモードを使用するセッションの最小数。 0 ~ 32,767 の範囲の値を指定してください。

min_conwinners_source パラメーターと *min_conwinners_target* パラメーターの合計は、*plu_mode_session_limit* パラメーターを超えてはなりません。

min_conwinners_target

パートナー LU がコンテンション勝者である、このモードを使用するセッションの最小数。 0 ~ 32,767 の範囲の値を指定してください。

min_conwinners_source パラメーターと *min_conwinners_target* パラメーターの合計は、*plu_mode_session_limit* パラメーターを超えてはなりません。

auto_act

自動的に活動化するセッションの数。 0 ~ 32,767 の範囲で値を指定します (値は、*plu_mode_session_limit* パラメーター、あるいはローカル LU に対して `DEFINE_LOCAL_LU verb` で指定したセッション限度を超えてはなりません)。自動的に活動化されるセッションの実際数は、この値と、ローカル LU のコンテンション勝者セッションの折衝された最小数のどちらか小さい方です。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

値は次のいずれかです。

AP_AS_NEGOTIATED

セッション限度が初期化されましたが、パートナー LU によって 1 つ以上の値が折衝されました。

AP_AS_SPECIFIED

パートナー LU による折衝なしに、セッション限度が要求どおりに初期化されました。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_EXCEEDS_MAX_ALLOWED

plu_mode_session_limit、*min_conwinners_source*、*min_conwinners_target*、*auto_act* のいずれかのパラメーターが、有効範囲外の値に設定されました。

AP_CANT_CHANGE_TO_ZERO

この verb を使用して *plu_mode_session_limit* パラメーターをゼロに設定することはできません。代わりに RESET_SESSION_LIMIT を使用してください。

AP_INVALID_LU_ALIAS

lu_alias パラメーターが、定義されているどのローカル LU 別名にも一致しませんでした。

AP_INVALID_LU_NAME

lu_name パラメーターが、定義されているどのローカル LU 名にも一致しませんでした。

AP_INVALID_MODE_NAME

mode_name パラメーターが、定義されているどのモード名にも一致しませんでした。

AP_INVALID_PLU_NAME

fqplu_name パラメーターが、定義されているどのパートナー LU 名にも一致しませんでした。

AP_INVALID_SET_NEGOTIABLE

set_negotiable パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

AP_MODE_NOT_RESET

この LU-LU モードの組み合わせについて、1 つ以上のセッションが現在活動状態です。 INITIALIZE_SESSION_LIMIT でなく CHANGE_SESSION_LIMIT を使用して限度を指定してください。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: セッションの割り振りエラー

セッション割り振りエラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_ALLOCATION_ERROR

secondary_rc

AP_ALLOCATION_FAILURE_NO_RETRY

訂正処置が必要な状態のために、セッションを割り振ることができませんでした。 *sense_data* パラメーターとログに記録されたメッセージを検査して障害の理由を判別し、必要な処置を実行してください。エラー状態を訂正してからこの verb を再試行してください。

sense_data

割り振りの障害に関連した SNA センス・データ。

パラメーター: CNOS 処理エラー

エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_CONV_FAILURE_NO_RETRY

処置が必要な状態 (たとえば、構成の不一致またはセッション・プロトコル・エラーなど) により、セッション限度を初期化できませんでした。エラー状態について Communications Server for Linux ログ・ファイルを確認し、エラー状態を訂正してから、この verb を再試行してください。

primary_rc

AP_CNOS_PARTNER_LU_REJECT

secondary_rc

AP_CNOS_COMMAND_RACE_REJECT

セッションを活動化または非活動化したり、セッション限度を処理するために別の管理プログラム (または内部的に Communications Server for Linux ソフトウェア) が指定したモードでアクセスしていたため、verb が失敗しました。アプリケーションで verb を再試行してください。ただし、競争状態が解決されるように、タイムアウトのあとで再試行するようにしてください。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

OPEN_FILE

アプリケーションはこの verb を使用して、Communications Server for Linux ドメイン構成ファイルにアクセスしてドメイン・リソースを管理したり、**sna.net** ファイルにアクセスして Communications Server for Linux LAN 上のバックアップ・マスター・サーバーを管理したりすることができます。

この verb は、ヌルのターゲット・ハンドルを指定して発行しなければなりません。正常に終了した場合、Communications Server for Linux はファイルを識別するハンドルを戻し、アプリケーションはそのハンドルを他の NOF verb で使用して、verb のターゲットを示すことができます。

VCB 構造体

```
typedef struct open_file
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    CONFIG_FILE    file_info;      /* definition of file requested */
    AP_UINT32      target_handle;  /* handle for subsequent verbs */
    unsigned char  reserv3[4];     /* reserved                     */
} OPEN_FILE;

typedef struct config_file
{
    unsigned char  requested_role;  /* config file requested       */
    unsigned char  role_supplied;  /* config file returned       */
    unsigned char  system_name[128]; /* computer name where file   */
    /* located                       */
    unsigned char  file_name[81];  /* file name                   */
} CONFIG_FILE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_OPEN_FILE

file_info.requested_role

オープンするファイルのタイプ。値は次のいずれかです。

AP_MASTER

ドメイン構成ファイルのマスター・コピーをオープンします。アプリケーションからドメイン・リソースの構成を変更する *verb* を発行しようとする場合は、必ずこの値を使用しなければなりません。

AP_BACKUP

ドメイン構成ファイルのマスター・コピーが使用可能であればそれをオープンし、使用可能でなければバックアップ・コピーをオープンします。この値は、アプリケーションから *QUERY_* verb* のみを使用しようとする場合に使用できます。構成内容を変更する必要がある場合は、*AP_MASTER* を使用しなければなりません。バックアップ構成ファイルへの書き込みアクセスができないためです。

AP_SNA_NET

マスター・サーバー上の *sna.net* ファイルを開きます。

AP_TP_LOAD_INFO

トランザクション・プログラム (TP) のロード方法に関する情報が入ったローカル・マシン上のファイルへの接続をオープンします。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

target_handle

このファイルに対してあとに続く `verb` に使用する戻り値。

file_info.role_supplied

requested_role を `AP_BACKUP` に設定した場合、このパラメーターは戻されたファイル・ハンドルがマスター構成ファイルのものか、バックアップ・ファイルのものかを示します。値は次のいずれかです。

AP_MASTER

マスター構成ファイル。

AP_BACKUP

バックアップ構成ファイル。

requested_role の値が上記以外の場合、このパラメーターは定義されません。

file_info.system_name

そのファイルが配置されている Communications Server for Linux コンピューターの名前。

file_info.file_name

ファイルの名前。このパラメーターは 1 ~ 80 文字の ASCII スtring で、そのあとにヌル文字 (0x00) が付きます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_PARAMETER_CHECK`

secondary_rc

値は次のいずれかです。

AP_INVALID_FILE_NAME

file_name パラメーターに有効な構成ファイル名を指定しませんでした。

AP_INVALID_FILE_INFO

file_info 構造体にあるパラメーターの 1 つが有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての `NOF verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_STATE_CHECK`

secondary_rc

値は次のいずれかです。

AP_CONNECTION_NOT_MADE

Communications Server for Linux は、そのファイルへのローカル通信パスをセットアップできませんでした。

AP_FILE_BAD_RECORD

Communications Server for Linux は、構成ファイルのエラーを検出しました。エラー・ログ・ファイルを検査して、エラーの詳細を示すメッセージを参照してください。

AP_FILE_ROLE_UNAVAILABLE

アプリケーションがマスター構成ファイルまたはバックアップ構成ファイル (**sna.net** ファイル) を要求しましたが、使用可能なマスター・サーバーまたはバックアップ・サーバーがありません。この状態は、通常、新規サーバーがマスターの動作を引き継ぐときに発生する一時的なものです。

アプリケーションがサーバー指示を受け取るように登録されている場合、このアプリケーションは、この指示の *flags* パラメータを検査して、新規サーバーが正常にマスターの動作を引き継いだことを確認してから、OPEN_FILE verb を再試行します。詳しくは、832 ページの『SERVER_INDICATION』を参照してください。または、成功するまで間をおいて OPEN_FILE を再試行します。

AP_INVALID_VERSION

構成ファイル・ヘッダーの Communications Server for Linux バージョン番号が、使用している Communications Server for Linux ソフトウェアのバージョンに一致しません。正しいファイルがあるかどうかを検査してください。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

PATH_SWITCH

PATH_SWITCH は、現在活動状態である高速トランスポート・プロトコル (RTP) 接続を別のパスに切り替えるよう Communications Server for Linux に要求します。Communications Server for Linux は、適切なパスを検出できない場合には、接続をそのままの状態にして変更しません。

VCB 構造体

```
typedef struct path_switch
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;              /* reserved                     */
    unsigned char  format;               /* reserved                     */
    AP_UINT16      primary_rc;           /* primary return code          */
    AP_UINT32      secondary_rc;         /* secondary return code        */
    unsigned char  rtp_connection_name[8]; /* RTP connection name          */
} PATH_SWITCH;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_PATH_SWITCH

rtp_connection_name

パスの変更を要求する RTP 接続。これは 8 バイトからなるローカル・システム上で表示可能な文字セットのストリングです。8 バイトすべてが有効なバイトであり、必ず設定しなければなりません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_RTP_CONNECTION

rtp_connection_name パラメーターに指定した値が、既存の RTP 接続の名前に一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_STATE_CHECK

secondary_rc

AP_PATH_SWITCH_IN_PROGRESS

Communications Server for Linux は、*rtp_connection_name* パラメーターで指定した RTP 接続のパスを現在変更中です。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: パス・スイッチが使用不可の場合

RTP パートナー・ノードがパス・スイッチ・タイマーをゼロに設定することによってパス・スイッチを使用不可にしたために、この *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PATH_SWITCH_DISABLED
```

```
secondary_rc
    (2 次戻りコードは戻されません。)
```

戻りパラメーター: パス・スイッチに障害がある場合

パス・スイッチの試みが失敗したために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_UNSUCCESSFUL
```

```
secondary_rc
    (2 次戻りコードは戻されません。)
```

戻りパラメーター: ノード検査

システムが RTP サポート付きで構築されていないために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_INVALID_VERB
```

```
secondary_rc
    (2 次戻りコードは戻されません。)
```

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_ACTIVE_TRANSACTION

QUERY_ACTIVE_TRANSACTION は、Communications Server for Linux 管理サービスのコンポーネントで既に認識されている活動状態である複数ドメイン・サポート (MDS) トランザクションに関する情報を戻します。活動状態であるトランザクションとは、それに関する情報がまだ受信されていない MDS 要求のことです。

この `verb` は、使用するオプションに応じて、1 つのトランザクションに関する情報を入手するか、複数のトランザクションに関する情報を入手するために使用できます。

この `verb` は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_active_transaction
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;              /* reserved                  */
    unsigned char  format;               /* reserved                  */
    AP_UINT16      primary_rc;           /* primary return code      */
    AP_UINT32      secondary_rc;        /* secondary return code    */
    unsigned char  *buf_ptr;             /* pointer to buffer        */
    AP_UINT32      buf_size;             /* buffer size              */
}
```

QUERY_ACTIVE_TRANSACTION

```
    AP_UINT32    total_buf_size;          /* total buffer size required */
    AP_UINT16    num_entries;             /* number of entries          */
    AP_UINT16    total_num_entries;      /* total number of entries    */
    unsigned char list_options;          /* listing options            */
    unsigned char reserv3;               /* reserved                    */
    unsigned char fq_req_loc_cp_name[17]; /* fq cp name of transaction  */
                                           /* requestor                   */
    unsigned char req_agent_appl_name[8]; /* appl name of transaction   */
                                           /* requestor                   */
    unsigned char seq_num_dt[17];        /* sequence number date/time  */
} QUERY_ACTIVE_TRANSACTION;

typedef struct active_transaction_data
{
    AP_UINT16    overlay_size;           /* size of returned entry     */
    unsigned char fq_origin_cp_name[17]; /* cp name of transaction origin */
    unsigned char origin_ms_appl_name[8]; /* appl name of transaction   */
                                           /* origin                      */
    unsigned char fq_dest_cp_name[17];   /* cp name of transaction     */
                                           /* destination                  */
    unsigned char dest_ms_appl_name[8];  /* appl name of transaction dest */
    unsigned char fq_req_loc_cp_name[17]; /* fq cp name of transaction  */
                                           /* requestor                    */
    unsigned char req_agent_appl_name[8]; /* appl name of transaction   */
                                           /* requestor                    */
    unsigned char seq_num_dt[17];        /* sequence number date/time  */
    unsigned char reserva[20];           /* reserved                    */
} ACTIVE_TRANSACTION_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_ACTIVE_TRANSACTION

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

トランザクションの最大数 (それらのトランザクションについてのデータが戻されます)。ある範囲でなく、特定のトランザクションに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリから開始します。

AP_LIST_INCLUSIVE

fq_req_loc_cp_name、*req_agent_appl_name*、*seq_num_dt* の各パラメーターで指定したエントリから開始します。

AP_LIST_FROM_NEXT

fq_req_loc_cp_name、*req_agent_appl_name*、*seq_num_dt* の各パラメーターで指定したエントリの直後のエントリから開始します。

リストは、最初に *fq_req_loc_cp_name* 順に並べられ、次に *req_agent_appl_name* 順、最後に *seq_num_dt* の数値順に並んでいます。リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

fq_req_loc_cp_name

トランザクション・リクエスターの完全修飾制御点名。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

req_agent_appl_name

トランザクション・リクエスターのアプリケーション名。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、通常、タイプ 1134 の文字 (大文字の A~Z と数字の 0 ~ 9) を使用した EBCDIC ストリングです。あるいは、「SNA Management Services Reference」で指定された『MS Discipline-Specific Application Program』の 1 つである場合もあります。このストリングは、長さが 8 文字でなければならず、必要に応じて EBCDIC のスペース文字 (0x40) を右側に埋め込みます。

seq_num_dt

IBM SNA フォーマット・マニュアルで定義されている、元のトランザクションの順序番号と日時の相関係数 (長さ 17 バイト)。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

QUERY_ACTIVE_TRANSACTION

active_transaction_data.overlay_size

戻された `active_transaction_data` 構造体のサイズ。すなわち、データ・バッファ内での次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 `active_transaction_data` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

active_transaction_data.fq_origin_cp_name

トランザクションの起点の完全修飾制御点名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

active_transaction_data.origin_ms_appl_name

トランザクションの起点のアプリケーション名。この名前は通常、タイプ 1134 の文字 (大文字の A~Z と数字の 0 ~ 9) を使用した 8 文字の EBCDIC ストリングです。あるいは、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照) で指定された『MS Discipline-Specific Application Program』の 1 つである場合もあります。

active_transaction_data.fq_dest_cp_name

トランザクションの宛先の完全修飾制御点名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

active_transaction_data.dest_ms_appl_name

トランザクションの宛先のアプリケーション名。この名前は通常、タイプ 1134 の文字 (大文字の A~Z と数字の 0 ~ 9) を使用した 8 文字の EBCDIC ストリングです。あるいは、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照) で指定された『MS Discipline-Specific Application Program』の 1 つである場合もあります。

active_transaction_data.fq_req_loc_cp_name

トランザクション・リクエストの完全修飾制御点名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

active_transaction_data.req_agent_appl_name

トランザクション・リクエストのアプリケーション名。この名前は通常、タイプ 1134 の文字 (大文字の A~Z と数字の 0 ~ 9) を使用した 8 文字

の EBCDIC ストリングです。あるいは、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照)で指定された『MS Discipline-Specific Application Program』の 1 つである場合があります。

active_transaction_data.seq_num_dt

IBM SNA フォーマット・マニュアルで定義されている、元のトランザクションの順序番号と日時の相関係数 (長さ 17 バイト)。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_ACTIVE_TRANSACTION

制御点名、アプリケーション名、順序番号相関係数のいずれかが、活動状態であるトランザクションのものと一致しませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの `verb` をサポートしていないために `verb` が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは MS ネットワーク管理機能をサポートしていません。これは、DEFINE_NODE `verb` の *mds_supported* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_ADJACENT_NN

QUERY_ADJACENT_NN verb は、隣接ネットワーク・ノード (そのノードへの CP-CP セッションが活動状態であるか、ある時点で活動状態だったことがあるネットワーク・ノード) に関する情報を戻します。この verb は Communications Server for Linux ノードがネットワーク・ノードである場合のみ使用可能であり、このノードがエンド・ノードまたは LEN ノードである場合は無効です。

この verb は、使用するオプションに応じて、特定の隣接ネットワーク・ノードに関する情報を入手するか、複数の隣接ネットワーク・ノードに関する情報を入手するために使用できます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_adjacent_nn
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;          /* reserved                      */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  *buf_ptr;        /* pointer to buffer            */
    AP_UINT32      buf_size;        /* buffer size                  */
    AP_UINT32      total_buf_size;  /* total buffer size required   */
    AP_UINT16      num_entries;     /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries     */
    unsigned char  list_options;    /* listing options              */
    unsigned char  reserv3;         /* reserved                      */
    unsigned char  adj_nncp_name[17]; /* CP name of adjacent Network Node */
} QUERY_ADJACENT_NN;

typedef struct adj_nncp_data
{
    AP_UINT16      overlay_size;    /* size of returned entry       */
    unsigned char  adj_nncp_name[17]; /* CP name of adjacent network node */
    unsigned char  cp_cp_sess_status; /* CP-CP session status        */
    AP_UINT32      out_of_seq_tdus; /* out of sequence TDUs        */
    AP_UINT32      last_frsn_sent;  /* last FRSN sent              */
    AP_UINT32      last_frsn_rcvd;  /* last FRSN received          */
    unsigned char  reserva[20];     /* reserved                     */
} ADJ_NNCP_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_ADJACENT_NN

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size
指定したデータ・バッファのサイズ。

num_entries
隣接 NN の最大数 (それらのノードについてのデータが戻されます)。ある範囲でなく、特定の隣接 NN に関するデータを要求するには、値 1 を指定

します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すための隣接 NN リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

adj_nncp_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

adj_nncp_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

adj_nncp_name

情報を求める隣接 NN の完全修飾名、または隣接 NN リストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A String 文字からなるネットワーク名で構成されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

QUERY_ADJACENT_NN

データ・バッファ内の各エントリは、次のパラメーターで構成されています。

adj_nncp_data.overlay_size

戻された *adj_nncp_data* 構造体のサイズ。すなわち、データ・バッファ内で次のエントリが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 *adj_nncp_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

adj_nncp_data.adj_nncp_name

隣接 NN の完全修飾名。この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A スtring文字からなるネットワーク名で構成されます。

adj_nncp_data.cp_cp_sess_status

隣接 NN への CP-CP セッションの状況。値は次のいずれかです。

AP_ACTIVE

セッションは活動状態です。

AP_CONWINNER_ACTIVE

セッション (競合勝者セッション) は活動状態です。

AP_CONLOSER_ACTIVE

セッション (競合敗者セッション) は活動状態です。

AP_INACTIVE

セッションは活動状態ではありません。

adj_nncp_data.out_of_seq_t dus

このノードから受信した順不同 TDU の数。

adj_nncp_data.last_frsn_sent

このノードへ最後に送信されたフロー縮小シーケンス番号 (Flow Reduction Sequence Number: FRSN)。

adj_nncp_data.last_frsn_rcvd

このノードから最後に受信されたフロー縮小シーケンス番号 (Flow Reduction Sequence Number: FRSN)。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_ADJ_NNCP_NAME

list_options パラメーターを、指定した名前から始まるすべてのエン
トリーをリスト出力する AP_LIST_INCLUSIVE に設定しましたが、
adj_nncp_name パラメーターが有効ではありませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんで
した。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、
AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードがネットワーク・ノードでないために verb が正常に実行されない
と、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは、エンド・ノードか LEN ノードです。この
verb は、ネットワーク・ノードでのみ有効です。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記
以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_AVAILABLE_TP

QUERY_AVAILABLE_TP は、活動状態で呼び出し可能 TP (RECEIVE_ALLOCATE
verb を発行した APPC アプリケーション、または Accept_Conversation コールまた
は Accept_Incoming コールを発行した CPI-C アプリケーション) に関する情報を戻
します。この verb は、使用するオプションに応じて、特定の TP に関する情報を
入手するか、複数の TP に関する情報を入手するために使用できます。この verb
は、TP が現在新しい着呼会話を受け付けようとする未処理の APPC verb または
CPI-C コールがあるかどうかに関係なく、実行中のすべての TP に関する情報を戻
します。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_available_tp
{
    AP_UINT16      opcode;           /* Verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* Primary return code         */
    AP_UINT32      secondary_rc;   /* Secondary return code       */
    unsigned char  *buf_ptr;       /* pointer to buffer           */
    AP_UINT32      buf_size;       /* buffer size                 */
    AP_UINT32      total_buf_size; /* total buffer size required  */
}
```

QUERY_AVAILABLE_TP

```
AP_UINT16      num_entries;      /* number of entries      */
AP_UINT16      total_num_entries; /* total number of entries */
unsigned char  list_options;     /* listing options        */
unsigned char  reserv3[3];       /* reserved                */
unsigned char  tp_name[64];      /* TP name                */
unsigned char  system_name[128]; /* computer name where TP is
                                /* running                */
} QUERY_AVAILABLE_TP;

typedef struct available_tp_data
{
    AP_UINT16      overlay_size;   /* size of returned entry  */
    unsigned char  tp_name[64];    /* TP name                  */
    unsigned char  reserv4[4];     /* reserved                 */
    unsigned char  system_name[128]; /* computer name where TP is
                                /* running                 */
} AVAILABLE_TP_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_AVAILABLE_TP

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファーへのポインター。

buf_size

指定したデータ・バッファーのサイズ。

num_entries

TP の最大数 (それらの TP についてのデータが戻されます)。ある範囲でなく、特定の TP に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すための TP リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

TP 名とシステム名の組み合わせによって指定したエントリーから開始します。

AP_LIST_FROM_NEXT

TP 名とシステム名の組み合わせによって指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

tp_name

TP 名。これは 64 バイトのストリングで、名前が 64 文字に満たない場合は、右側にスペースを入れます。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

system_name

TP 情報を求めるコンピューター名。このシステム名は 1 ~ 128 文字の ASCII スtringで、これは Communications Server for Linux コンピューター名に一致しなければなりません。この値は、*list_options* を *AP_FIRST_IN_LIST* に設定した場合には無視されます。

コンピューター名に . (ピリオド) 文字が含まれている場合、Communications Server for Linux はこれが完全修飾名であると想定します。ピリオドが含まれていない場合は、DNS ルックアップを実行してコンピューター名を判別します。

Communications Server for Linux が単一のコンピューター上にあるすべてのプログラムを使用して実行されている場合は、コンピューター名を指定する必要はありません (すべて 2 進ゼロのままです)。クライアント/サーバー・システムの場合は、特定のコンピューターの TP のみがリストされるようコンピューター名を指定するか、またはすべてのコンピューターの TP がリストされるようコンピューター名をすべて 2 進ゼロのままにします。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを返します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

available_tp_data.overlay_size

戻された *available_tp_data* 構造体のサイズ。すなわち、データ・バッファー内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *available_tp_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の *sizeof()* 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくな

QUERY_AVAILABLE_TP

る可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

available_tp_data.tp_name

TP 名。これは 64 バイトのストリングで、名前が 64 文字に満たない場合は、右側にスペースを入れます。

available_tp_data.system_name

その TP を実行しているコンピューターの名前。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

AP_UNKNOWN_TP

list_options パラメーターを、指定した名前から始まるすべてのエントリーをリスト出力する `AP_LIST_INCLUSIVE` に設定しましたが、*tp_name* パラメーターが有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_BUFFER_AVAILABILITY

この `verb` は Communications Server for Linux が現在使用している STREAMS バッファの容量、これが既に使用した最大容量、使用可能な最大容量 (`SET_BUFFER_AVAILABILITY verb` を使用して指定した) に関する情報を戻します。この `verb` を使用すると、STREAMS バッファの使用状況を検査して適切な限度を設定することが可能であり、それによって Communications Server for Linux のコンポーネントと Linux コンピューター上のその他のプログラムに十分なバッファを提供することができます。この `verb` は、Communications Server for Linux サポート担当員に対して、バッファの使用状況に関連した追加の内部値も戻します。

VCB 構造体

```
typedef struct query_buffer_availability
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;              /* reserved */
    AP_UINT16      primary_rc;          /* primary return code */
    AP_UINT32      secondary_rc;        /* secondary return code */
    AP_UINT32      reset_max_values;    /* set stored max values to
                                        /* current */

    AP_UINT32      buf_avail;           /* maximum buffer space
                                        /* available */

    AP_UINT32      buf_total_count;     /* current buffer usage - count */
    AP_UINT32      buf_total_bytes;    /* current buffer usage - bytes */
    AP_UINT32      buf_rsrv_count;     /* buffers reserved - count */
    AP_UINT32      buf_rsrv_bytes[2];  /* buffers reserved - bytes */
    AP_UINT32      buf_res_use_count;   /* usage of reserved buffers
                                        /* - count */
    AP_UINT32      buf_res_use_bytes;   /* usage of reserved buffers
                                        /* - bytes */

    AP_UINT32      peak_usage;         /* peak usage */
    AP_UINT32      peak_decay;         /* peak decay */
    unsigned char  throttle_status;    /* throttle status */
    unsigned char  buf_use_status;     /* congestion status */
    AP_UINT32      max_buf_total_count; /* maximum buffer usage - count */
    AP_UINT32      max_buf_total_bytes; /* maximum buffer usage - bytes */
    AP_UINT32      max_buf_rsrv_count;  /* max buffers reserved - count */
    AP_UINT32      max_buf_rsrv_bytes[2]; /* max buffers reserved - bytes */
    AP_UINT32      max_buf_res_use_count; /* max rsrv buffer usage - count */
    AP_UINT32      max_buf_res_use_bytes; /* max rsrv buffer usage - bytes */
    AP_UINT32      max_peak_usage;     /* maximum peak usage */
    unsigned char  max_throttle_status; /* maximum throttle status */
    unsigned char  max_buf_use_status; /* maximum congestion status */
    unsigned char  debug_param[32];    /* reserved */
    unsigned char  reserv3[8];         /* reserved */
} QUERY_BUFFER_AVAILABILITY;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_BUFFER_AVAILABILITY

reset_max_values

*max_** パラメーターの値を (この verb で戻したあと)、これらのパラメーターの現行値に一致するよう Communications Server for Linux にリセットさせるかどうかを指定します。これにより、以降の

QUERY_BUFFER_AVAILABILITY verb は、システムを始動してから (または *max_** パラメーターを最後にリセットしてから) の最大値でなく、この verb の発行後に到達した最大値を戻します。値は次のいずれかです。

AP_YES *max_** パラメーターの値を、現行値に一致するようリセットします。

AP_NO *max_** パラメーターの値をリセットしません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。他のフィールドに戻された値は、Communications Server for Linux サポート担当員が使用します。

QUERY_BUFFER_AVAILABILITY

primary_rc

AP_OK

secondary_rc

未使用。

buf_avail

Communications Server for Linux から使用できる STREAMS バッファースペースのバイト単位の最大容量 (SET_BUFFER_AVAILABILITY verb で定義したもの)。

buf_total_count

Communications Server for Linux のコンポーネントに、現在、割り振られているバッファースペースの数。

buf_total_bytes

Communications Server for Linux のコンポーネントに、現在、割り振られているバッファースペースの合計容量。

buf_rsrv_count

予約されたバッファースペースの合計数。

buf_rsrv_bytes

予約されたバッファースペースの合計記憶容量 (バイト単位)。

buf_res_use_count

使用中の予約済みバッファースペースの数。

buf_res_use_bytes

現在使用中の予約済みバッファースペース内のバイト数。

peak_usage

ピーク時のバッファースペース使用率 (実際に使用されたバッファースペースの平滑化したパーセント)。

peak_decay

平滑化パラメーター。

throttle_status

適応ペーシング状況。

buf_use_status

輻輳状況。値は次のいずれかです。

AP_CONGESTED

AP_UNCONGESTED

max_buf_total_count

Communications Server for Linux のコンポーネントに、任意の時点で割り振られているバッファースペースの最大数。

max_buf_total_bytes

Communications Server for Linux のコンポーネントに、任意の時点で割り振られているバッファースペースの最大容量。

max_buf_rsrv_count

予約できるバッファースペースの最大数。

max_buf_rsrv_bytes

予約できるバッファ・ストレージの最大容量 (バイト単位)。

max_buf_res_use_count

使用できる予約済みバッファの最大数。

max_buf_res_use_bytes

常時使用できる予約済みバッファの最大バイト数。

max_peak_usage

ピーク時のバッファ使用率の最大値 (実際に使用されたバッファの平滑化したパーセント)。

max_throttle_status

適応ペーシング状況の最大値。

max_buf_use_status

輻輳状況の最大値。値は次のいずれかです。

AP_CONGESTED
AP_UNCONGESTED

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_CENTRAL_LOGGER

この verb は、中央ロガー (すべてのサーバーの Communications Server for Linux ログ・メッセージの送信先となる中央ログ・ファイルを保持するノード) として、現在、定義されているノードの名前を戻します。この verb は、中央ロギングがアクティブであるかどうかに関する情報を戻しません。アクティブであるかどうかを判別するには、QUERY_CENTRAL_LOGGING を使用します。

この verb は、ヌルのターゲット・ハンドルを指定して発行しなければなりません。

VCB 構造体

```
typedef struct query_central_logger
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  reserv3[4];     /* reserved                  */
    unsigned char  node_name[128]; /* name of central logger   */
} QUERY_CENTRAL_LOGGER;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_CENTRAL_LOGGER

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

node_name
中央ロガーとして定義されているノードの名前。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_STATE_CHECK

secondary_rc
AP_NO_CENTRAL_LOG
マスター・サーバーは現在アクティブではありません。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_CENTRAL_LOGGING

この verb は、Communications Server for Linux ログ・メッセージがすべてのサーバーから中央ファイルに送信されるか、または各サーバー上の個々のファイルに送信されるかに関する情報を戻します。詳しくは、738 ページの『SET_LOG_FILE』を参照してください。

この verb は、中央ロガーとして現在機能しているノードに発行する必要があります。このノードへのアクセスについては、70 ページの『CONNECT_NODE』を参照してください。

VCB 構造体

```
typedef struct query_central_logging
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  enabled;        /* is central logging enabled? */
    unsigned char  reserv3[3];     /* reserved                    */
} QUERY_CENTRAL_LOGGING;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_CENTRAL_LOGGING

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

enabled
中央ロギングが使用可能であるか、または使用不可であることを指定します。値は次のいずれかです。

AP_YES 中央ロギングは使用可能です。すべてのログ・メッセージは、現在の中央ロガーであるノード上の単一のファイルに送信されます。

AP_NO 中央ロギングは使用不可です。各サーバーからのログ・メッセージは、そのサーバー上のファイル (SET_LOG_FILE *verb* を使用して指定される) に送信されます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_NOT_CENTRAL_LOGGER
この *verb* は、中央ロガーではないノードに発行されました。

状態の検査

状態エラーのためコマンドが実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_STATE_CHECK

secondary_rc
AP_NO_CENTRAL_LOG
マスター・サーバーは現在アクティブではありません。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_CN

QUERY_CN は、隣接接続ネットワークに関する情報を戻します。この情報は、「判別されたデータ」(実行中に動的に収集したデータ) および「定義されたデータ」(DEFINE_CN でアプリケーションから定義したデータ) として構造化されます。

この verb は、使用するオプションに応じて、特定の接続ネットワークに関する情報を入手するか、複数の接続ネットワークに関する情報を入手するために使用できます。この verb はネットワーク・ノードまたはエンド・ノードのみで発行でき、LEN ノードでは無効です。

VCB 構造体

```
typedef struct query_cn
{
    AP_UINT16      opcode;           /* Verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* Primary return code         */
    AP_UINT32      secondary_rc;   /* Secondary return code       */
    unsigned char  *buf_ptr;       /* pointer to buffer           */
    AP_UINT32      buf_size;       /* buffer size                 */
    AP_UINT32      total_buf_size; /* total buffer size required  */
    AP_UINT16      num_entries;    /* number of entries           */
    AP_UINT16      total_num_entries; /* total number of entries    */
    unsigned char  list_options;   /* listing options             */
    unsigned char  reserv3;       /* reserved                     */
    unsigned char  fqcn_name[17];  /* Name of Connection Network  */
} QUERY_CN;

typedef struct cn_data
{
    AP_UINT16      overlay_size;   /* size of returned entry      */
    unsigned char  fqcn_name[17]; /* Name of Connection Network  */
    unsigned char  reserv1;       /* reserved                     */
    CN_DET_DATA    det_data;      /* Determined data             */
    CN_DEF_DATA    def_data;      /* Defined data                */
} CN_DATA;

typedef struct cn_det_data
{
    AP_UINT16      num_act_ports;  /* number of active ports     */
    unsigned char  reserva[20];   /* reserved                    */
} CN_DET_DATA;

typedef struct cn_def_data
{
    unsigned char  description[32]; /* resource description        */
    unsigned char  reserve0[16];   /* reserved                    */
    unsigned char  num_ports;     /* number of ports on CN     */
    unsigned char  cn_type;       /* reserved                    */
    unsigned char  reserve1[15];   /* reserved                    */
    TG_DEFINED_CHARS tg_chars;    /* TG characteristics         */
} CN_DEF_DATA;

typedef struct tg_defined_chars
{
    unsigned char  effect_cap;     /* effective capacity         */
    unsigned char  reserve1[5];    /* reserved                    */
    unsigned char  connect_cost;  /* connection cost           */
    unsigned char  byte_cost;     /* byte cost                  */
    unsigned char  reserve2;     /* reserved                    */
    unsigned char  security;      /* security                   */
    unsigned char  prop_delay;    /* propagation delay         */
}
```

```

unsigned char    modem_class;        /* reserved                */
unsigned char    user_def_parm_1;    /* user-defined parameter 1 */
unsigned char    user_def_parm_2;    /* user-defined parameter 2 */
unsigned char    user_def_parm_3;    /* user-defined parameter 3 */
} TG_DEFINED_CHARS;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_CN

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

CN の最大数 (その CN のデータが戻されます)。ある範囲でなく、特定の CN に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すための CN リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

fqcn_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

fqcn_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

fqcn_name

情報を求める CN の完全修飾名、または CN リストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

cn_data.overlay_size

戻された *cn_data* 構造体のサイズ。すなわち、データ・バッファー内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *cn_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

cn_data.fqcn_name

CN の完全修飾名。この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String文字からなるネットワーク名で構成されます。

cn_data.det_data.num_act_ports

その接続ネットワーク上で活動状態であるポートの数。

cn_data.def_data.description

CN を記述したヌルで終了するテキスト・String (その CN の定義で指定したもの)。

cn_data.def_data.num_ports

その接続ネットワーク上のポートの合計数。

cn_data.def_data.tg_chars.effect_cap

実際のビット/秒率 (回線速度)。値は 1 バイトの浮動小数点数としてエンコードされます。このときの公式は、 $0.1 \text{ mmm} * 2^{\text{eeee}}$ で、バイトのビット表現は `eeeeemmm` です。300 ビット/秒の倍数で指定してください。

cn_data.def_data.tg_chars.connect_cost

接続時間当たりのコスト。有効な値は 0 ~ 255 の範囲の整数値で、0 が接続時間当たりの最低コスト、255 が最高コストです。

cn_data.def_data.tg_chars.byte_cost

バイト当たりのコスト。値は 0 ~ 255 の範囲の整数で、ゼロがバイト当たりの最低コスト、255 が最高コストです。

cn_data.def_data.tg_chars.security

ネットワークのセキュリティー・レベル。値は次のいずれかです。

AP_SEC_NONSECURE

セキュリティーなし。

AP_SEC_PUBLIC_SWITCHED_NETWORK

公衆交換網を介してデータを伝送します。

AP_SEC_UNDERGROUND_CABLE

保護された地下ケーブルを介してデータを伝送します。

AP_SEC_SECURE_CONDUIT

ガードされていない保護されたコンジット (配管) 内の回線を介してデータを伝送します。

AP_SEC_GUARDED_CONDUIT

物理接続による盗聴から保護されたコンジット内の回線を介してデータを伝送します。

AP_SEC_ENCRYPTED

回線を介して伝送する前にデータを暗号化します。

AP_SEC_GUARDED_RADIATION

物理接続による盗聴および電波盗聴から保護された回線を介してデータを伝送します。

AP_SEC_MAXIMUM

最大のセキュリティー。

cn_data.def_data.tg_chars.prop_delay

伝搬遅延: 信号をリンクの長さのみ伝送するのに要する時間。値は次のいずれかです。

AP_PROP_DELAY_MINIMUM

最小の伝搬遅延。

AP_PROP_DELAY_LAN

遅延は 480 マイクロ秒未満です (LAN の通常値)。

AP_PROP_DELAY_TELEPHONE

遅延は 480 ~ 49,512 マイクロ秒の範囲です (電話網の通常値)。

AP_PROP_DELAY_PKT_SWITCHED_NET

遅延は 49,512 ~ 245,760 マイクロ秒の範囲です (パケット交換ネットワークの通常値)。

AP_PROP_DELAY_SATELLITE

遅延は 245,760 マイクロ秒を超えます (サテライト・リンクの通常値)。

AP_PROP_DELAY_MAXIMUM

最大の伝搬遅延。

cn_data.def_data.tg_chars.user_def_parm_1 ~ def_data.tg_chars.user_def_parm_3

上記のパラメーターで対応できないその他の TG 特性を含むユーザー定義パラメーター。これらのパラメーターは、それぞれ 0 ~ 255 の範囲の値に設定されます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_CN_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、**AP_LIST_INCLUSIVE** に設定されていますが、*fqcn_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての **NOF verb** に共通し、**AP_PARAMETER_CHECK** に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードが **LEN** ノードであるために `verb` が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは **LEN** ノードです。この `verb` は、ネットワーク・ノードまたはエンド・ノードでのみ有効です。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての **NOF verb** に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_CN_PORT

QUERY_CN_PORT は、隣接接続ネットワーク上に定義されたポートに関する情報を戻します。

この `verb` は、使用するオプションに応じて、特定のポートに関する情報を入手するか、複数のポートに関する情報を入手するために使用できます。この `verb` はネットワーク・ノードまたはエンド・ノードのみで発行でき、**LEN** ノードでは無効です。

VCB 構造体

```

typedef struct query_cn_port
{
    AP_UINT16      opcode;           /* Verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* Primary return code      */
    AP_UINT32      secondary_rc;    /* Secondary return code    */
    unsigned char  *buf_ptr;        /* pointer to buffer        */
    AP_UINT32      buf_size;        /* buffer size              */
    AP_UINT32      total_buf_size;  /* total buffer size required */
    AP_UINT16      num_entries;     /* number of entries        */
    AP_UINT16      total_num_entries; /* total number of entries  */
    unsigned char  list_options;    /* listing options          */
    unsigned char  reserv3;         /* reserved                  */
    unsigned char  fqcn_name[17];   /* Name of Connection Network */
    unsigned char  port_name[8];    /* port name                 */
} QUERY_CN_PORT;

typedef struct cn_port_data
{
    AP_UINT16      overlay_size;     /* size of returned entry   */
    unsigned char  fqcn_name[17];    /* Name of Connection Network */
    unsigned char  port_name[8];     /* name of port              */
    unsigned char  tg_num;           /* transmission group number */
    unsigned char  reserva[20];      /* reserved                   */
} CN_PORT_DATA;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_CN_PORT

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size
指定したデータ・バッファのサイズ。

num_entries
ポートの最大数 (そのポートのデータが戻されます)。ある範囲でなく、特定のポートに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options
Communications Server for Linux がデータを戻すためのポート・リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST
リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE
port_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT
port_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

fqn_name

求めるポートが定義されている CN、または求めるポート・リストの対象である CN の完全修飾名。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

port_name

情報を求めるポートの名前、またはポート・リストの索引として使用する名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 文字に満たない場合、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。buf_size より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。num_entries より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

cn_port_data.overlay_size

戻された cn_port_data 構造体のサイズ。すなわち、データ・バッファー内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファーを調べて、各 cn_port_data 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の sizeof() 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

cn_port_data.fqcn_name

CN の完全修飾名。この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String文字からなるネットワーク名で構成されます。

cn_port_data.port_name

ポートの名前。これは 8 バイトからなる ASCII Stringで、名前が 8 文字に満たない場合、右側にスペースを入れます。

cn_port_data.tg_num

指定したポートの伝送グループ番号。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_CN_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*fqcn_name* パラメーターは無効でした。

AP_INVALID_PORT_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*port_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードが LEN ノードであるために `verb` が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

*primary_rc***AP_FUNCTION_NOT_SUPPORTED**

ローカル・ノードは LEN ノードです。この `verb` は、ネットワーク・ノードまたはエンド・ノードでのみ有効です。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_CONVERSATION

QUERY_CONVERSATION は、特定のローカル LU を使用する会話に関する情報を戻します。

この verb は、使用するオプションに応じて、特定の会話または会話の範囲に関する情報を入手するために使用できます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_conversation
{
    AP_UINT16    opcode;                /* verb operation code      */
    unsigned char reserv2;              /* reserved                  */
    unsigned char format;               /* reserved                  */
    AP_UINT16    primary_rc;            /* primary return code      */
    AP_UINT32    secondary_rc;          /* secondary return code    */
    unsigned char *buf_ptr;              /* pointer to buffer        */
    AP_UINT32    buf_size;               /* buffer size              */
    AP_UINT32    total_buf_size;         /* total buffer size required */
    AP_UINT16    num_entries;            /* number of entries        */
    AP_UINT16    total_num_entries;     /* total number of entries  */
    unsigned char list_options;          /* listing options          */
    unsigned char reserv3;               /* reserved                  */
    unsigned char lu_name[8];            /* LU Name                  */
    unsigned char lu_alias[8];           /* LU Alias                 */
    AP_UINT32    conv_id;                /* Conversation ID          */
    unsigned char session_id[8];         /* Session ID               */
    unsigned char reserv4[12];           /* reserved                  */
} QUERY_CONVERSATION;

typedef struct conv_summary
{
    AP_UINT16    overlay_size;           /* overlay size             */
    AP_UINT32    conv_id;                /* conversation ID          */
    unsigned char local_tp_name[64];     /* local TP name           */
    unsigned char partner_tp_name[64];   /* partner TP name         */
    unsigned char tp_id[8];              /* TP ID                   */
    unsigned char sess_id[8];            /* Session ID              */
    AP_UINT32    conv_start_time;        /* Conversation start time  */
    AP_UINT32    bytes_sent;              /* Number of bytes sent    */
    AP_UINT32    bytes_received;         /* Number of bytes received */
    unsigned char conv_state;            /* conversation state       */
    unsigned char duplex_type;           /* full- or half-duplex conv? */
} CONV_SUMMARY;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_CONVERSATION

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファーへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

会話の最大数 (その会話のデータが戻されます)。ある範囲でなく、特定の会話に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリから開始します。

AP_LIST_INCLUSIVE

ローカル LU と会話 ID を組み合わせて指定したエントリから開始します。

AP_LIST_FROM_NEXT

ローカル LU と会話 ID を組み合わせて指定したエントリの直後のエントリから開始します。

list_options を AP_LIST_INCLUSIVE または AP_LIST_FROM_NEXT に設定した場合、指定したローカル LU (*lu_name* または *lu_alias*) と会話 ID (*conv_id*) の組み合わせは、セッションのリストの索引として使用されます。

リストの順序と、アプリケーションでそのリストから特定のエントリを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

lu_name

LU 名。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。LU を LU 名ではなく、別名で識別することを指定するには、このパラメーターを 8 桁の 2 進ゼロに設定し、次のパラメーターで LU 別名を指定します。ローカル CP に関連した LU (デフォルト LU) を指定するには、*lu_name* と *lu_alias* の両方をすべて 2 進ゼロに設定します。

lu_alias

ローカルで定義された LU 別名。これは 8 バイトからなる ASCII String で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。このパラメーターは、*lu_name* を 8 桁の 2 進ゼロに設定した場合のみ使用され、それ以外の場合は無視されます。ローカル CP に関連した LU (デフォルト LU) を指定するには、*lu_name* と *lu_alias* の両方をすべて 2 進ゼロに設定します。

conv_id

情報を求める会話の ID、または会話のリストの索引として使用する会話 ID。会話 ID は、呼び出している TP 内の ALLOCATE verb によって、または呼び出された TP 内の RECEIVE_ALLOCATE verb によって戻されました。

このパラメーターは、*list_options* を `AP_FIRST_IN_LIST` に設定した場合には無視されます。

session_id

8 バイトからなるセッション ID。特定のセッションに関連した会話についての情報のみをリストするには、セッション ID を指定します。すべてのセッションについての完全なリストを取得するには、このフィールドをすべて 2 進ゼロに設定します。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_OK`

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

conv_summary.overlay_size

戻された `conv_summary` 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 `conv_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

conv_summary.conv_id

会話 ID。会話 ID は、呼び出している TP 内の `ALLOCATE verb` によって、または呼び出された TP 内の `RECEIVE_ALLOCATE verb` によって戻されました。

conv_summary.local_tp_name

会話内のローカル TP の名前。

conv_summary.partner_tp_name

会話内のパートナー TP の名前。このパラメーターが戻されるのは、会話がローカル TP によって開始された場合のみです。会話がリモート TP によって開始された場合は予約済みです。

conv_summary.tp_id

会話の TP ID。

conv_summary.session_id

会話に割り振られたセッションのセッション ID。

conv_summary.conv_start_time

Communications Server for Linux ノードが開始された時刻から会話が開始された時刻までの経過時間 (1/100 秒)。

conv_summary.bytes_sent

会話が開始されてから、ローカル TP からパートナー TP に送信されたバイト数。

conv_summary.bytes_received

会話が開始されてから、パートナー TP からローカル TP が受信したバイト数。

conv_summary.conv_state

会話の現行状態。半二重会話の値は、以下のとおりです。

AP_CONFIRM_STATE
 AP_CONFIRM_DEALL_STATE
 AP_CONFIRM_SEND_STATE
 AP_END_CONV_STATE
 AP_PEND_DEALL_STATE
 AP_PEND_POST_STATE
 AP_POST_ON_RECEIPT_STATE
 AP_RECEIVE_STATE
 AP_RESET_STATE
 AP_SEND_STATE
 AP_SEND_PENDING_STATE

全二重会話の値は、以下のとおりです。

AP_RESET_STATE
 AP_SEND_ONLY_STATE
 AP_SEND_RECEIVE_STATE
 AP_RECEIVE_ONLY_STATE

conv_summary.duplex_type

会話の二重化タイプ。以下の値があります。

AP_HALF_DUPLEX
 AP_FULL_DUPLEX

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

QUERY_CONVERSATION

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_BAD_CONV_ID

list_options パラメーターが、指定した値から始まるすべてのエンターリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*conv_id* パラメーターは無効でした。

AP_INVALID_LU_ALIAS

指定した *lu_alias* パラメーターが、有効ではありませんでした。

AP_INVALID_LU_NAME

指定した *lu_name* パラメーターが、有効ではありませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_COS

QUERY_COS は、特定のサービス・クラス (COS) に関する経路計算情報を戻します。

この verb は、使用するオプションに応じて、特定の COS に関する情報を入手するか、複数の COS に関する情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_cos
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;                /* reserved                  */
    AP_UINT16      primary_rc;            /* primary return code      */
    AP_UINT32      secondary_rc;          /* secondary return code    */
    unsigned char  *buf_ptr;              /* pointer to buffer        */
    AP_UINT32      buf_size;              /* buffer size              */
    AP_UINT32      total_buf_size;        /* total buffer size required */
    AP_UINT16      num_entries;           /* number of entries        */
    AP_UINT16      total_num_entries;     /* total number of entries  */
    unsigned char  list_options;          /* listing options          */
    unsigned char  reserv3;               /* reserved                  */
    unsigned char  cos_name[8];           /* cos name                  */
} QUERY_COS;

typedef struct cos_data
{
    AP_UINT16      overlay_size;           /* size of returned entry   */
}
```

```

unsigned char  cos_name[8];           /* cos name */
unsigned char  description[32];      /* resource description */
unsigned char  reserv1[16];          /* reserved */
unsigned char  transmission_priority; /* transmission priority */
AP_UINT16     num_of_node_rows;      /* number of node rows */
AP_UINT16     num_of_tg_rows;        /* number of tg rows */
AP_UINT32     trees;                 /* number of tree caches for COS */
AP_UINT32     calcs;                 /* number of route calculations
                                     /* for this COS */
AP_UINT32     rejs;                  /* number of route rejects for
                                     /* COS */
unsigned char  reserva[20];          /* reserved */
} COS_DATA;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_COS

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファーへのポインター。

buf_size

指定したデータ・バッファーのサイズ。

num_entries

COS の最大数 (それらの COS についてのデータが戻されます)。ある範囲でなく、特定の COS に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すための COS リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

cos_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

cos_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

cos_name

データを求めるサービス・クラスの名前、またはリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側に EBCDIC のスペースを入れます。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

cos_data.overlay_size

戻された `cos_data` 構造体のサイズ。すなわち、データ・バッファー内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファーを調べて、各 `cos_data` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があります。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

cos_data.cos_name

サービス・クラス名。これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC ストリングで、右側に EBCDIC のスペースを入れます。

cos_data.description

COS を記述したヌルで終了するテキスト・ストリング (その COS の定義で指定したもの)。

cos_data.transmission_priority

伝送優先順位。値は次のいずれかです。

AP_LOW

AP_MEDIUM

AP_HIGH

AP_NETWORK (最高の優先順位)

cos_data.num_of_node_rows

この COS に定義されたノード行の数。

cos_data.num_of_tg_rows

この COS に定義された TG 行の数。

cos_data.trees

前回の初期化以後に、この COS に作成された経路ツリー・キャッシュの数。

cos_data.calcs

このサービス・クラスを指定したセッション活動化要求の数 (つまり、経路計算の回数)。

cos_data.rejs

ネットワークを介してこのノードから指定した宛先までの受け入れ可能な経路が存在しなかったために失敗したセッション活動化要求の数。経路が受け入れ可能になるのは、その経路全体が、指定したサービス・クラスを提供できる活動状態である TG とノードで構成されている場合のみです。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_COS_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*cos_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_COS_NODE_ROW

QUERY_COS_NODE_ROW は、前に DEFINE_COS で定義した (または SNA 定義 COS のノードによって暗黙に定義された) 指定サービス・クラスに関するノード行情報を戻します。

QUERY_COS_NODE_ROW

この verb は、使用するオプションに応じて、特定の COS ノード行に関する情報を入手するか、複数の COS ノード行に関する情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_cos_node_row
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  *buf_ptr;       /* pointer to buffer        */
    AP_UINT32      buf_size;       /* buffer size              */
    AP_UINT32      total_buf_size; /* total buffer size required */
    AP_UINT16      num_entries;    /* number of entries        */
    AP_UINT16      total_num_entries; /* total number of entries */
    unsigned char  list_options;   /* listing options          */
    unsigned char  reserv3;       /* reserved                  */
    unsigned char  cos_name[8];    /* cos name                  */
    AP_UINT16      node_row_index; /* node row index           */
} QUERY_COS_NODE_ROW;

typedef struct cos_node_row_data
{
    AP_UINT16      overlay_size;   /* size of returned entry   */
    unsigned char  cos_name[8];    /* cos name                  */
    AP_UINT16      node_row_index; /* node row index           */
    COS_NODE_ROW   node_row;      /* cos node row information  */
} COS_NODE_ROW_DATA;

typedef struct cos_node_row
{
    COS_NODE_STATUS minimum;      /* minimum                  */
    COS_NODE_STATUS maximum;     /* maximum                  */
    unsigned char  weight;       /* weight                   */
    unsigned char  reserv1;      /* reserved                  */
} COS_NODE_ROW;

typedef struct cos_node_status
{
    unsigned char  rar;          /* route additional resistance */
    unsigned char  status;      /* node status                */
    unsigned char  reserv1[2];  /* reserved                    */
} COS_NODE_STATUS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_COS_NODE_ROW

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size
指定したデータ・バッファのサイズ。

num_entries
COS ノード行の最大数 (そのノード行のデータが戻されます)。ある範囲でなく、特定の COS ノード行に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すための COS ノード行のリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

cos_name パラメーターと *node_row_index* パラメーターを組み合わせ指定したエントリーから開始します。

AP_LIST_FROM_NEXT

cos_name パラメーターと *node_row_index* パラメーターを組み合わせ指定したエントリーの直後のエントリーから開始します。

リストは、最初に *cos_name*、次に各 COS の *node_row_index* の順番で並んでいます。アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

cos_name

ノード行情報を求めるサービス・クラスの名前、またはリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側に EBCDIC のスペースを入れます。

node_row_index

情報を求めるノード行の番号、またはリストの索引として使用する番号。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この COS に関連したノード行の数を知るには、QUERY_COS を使用します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファ内の各エントリは、次のパラメーターで構成されています。

cos_node_row_data.overlay_size

戻された *cos_node_row_data* 構造体のサイズ。すなわち、データ・バッファ内で次のエントリが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 *cos_node_row_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

cos_node_row_data.cos_name

サービス・クラス名。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

cos_node_row_data.node_row_index

ノード行の索引。

cos_node_row_data.node_row.minimum.rar

0 ~ 255 の範囲の経路追加抵抗の最小値。

cos_node_row_data.node_row.minimum.status

ノードの最小輻輳状況を指定します。このパラメーターは、AP_UNCONGESTED、次に表示する値のいずれか、論理 OR で組み合わせた 2 つ以上の値に設定できます。値は次のいずれかです。

AP_UNCONGESTED

ISR セッションの数は、ノードの構成で *isr_sessions_upper_threshold* の値未満です。

AP_CONGESTED

ISR セッションの数は、しきい値を超えています。

AP_IRR_DEPLETED

ISR セッションの数は、そのノードに指定した最大値に達しました。

AP_ERR_DEPLETED

エンドポイント・セッションの数は、指定した最大値に達しました。

AP QUIESCING

タイプが AP_QUIESCE または AP_QUIESCE_ISR の STOP_NODE が発行されました。

cos_node_row_data.node_row.maximum.rar

0 ~ 255 の範囲の経路追加抵抗の最大値。

cos_node_row_data.node_row.maximum.status

ノードの最大輻輳状況を指定します。このパラメーターは、

AP_UNCONGESTED、次に表示する値のいずれか、論理 OR で組み合わせた 2 つ以上の値に設定できます。値は次のいずれかです。

AP_UNCONGESTED

ISR セッションの数は、ノードの構成で *isr_sessions_upper_threshold* の値未満です。

AP_CONGESTED

ISR セッションの数は、しきい値を超えています。

AP_IRR_DEPLETED

ISR セッションの数は、そのノードに指定した最大値に達しました。

AP_ERR_DEPLETED

エンドポイント・セッションの数は、指定した最大値に達しました。

AP QUIESCING

タイプが AP_QUIESCE または AP_QUIESCE_ISR の STOP_NODE が発行されました。

cos_node_row_data.node_row.weight

このノード行に関連した重み。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_COS_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*cos_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_COS_TG_ROW

QUERY_COS_TG_ROW は、前に DEFINE_COS で定義した (または SNA 定義 COS のノードによって暗黙に定義された) 指定サービス・クラスに関する TG 行情報を戻します。

この verb は、使用するオプションに応じて、特定の COS TG 行に関する情報を入手するか、複数の COS TG 行に関する情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_cos_tg_row
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
    unsigned char  *buf_ptr;        /* pointer to buffer        */
    AP_UINT32      buf_size;        /* buffer size              */
    AP_UINT32      total_buf_size;  /* total buffer size required */
    AP_UINT16      num_entries;     /* number of entries        */
    AP_UINT16      total_num_entries; /* total number of entries  */
    unsigned char  list_options;    /* listing options          */
    unsigned char  reserv3;         /* reserved                  */
    unsigned char  cos_name[8];     /* cos name                  */
    AP_UINT16      tg_row_index;    /* TG row index             */
} QUERY_COS_TG_ROW;

typedef struct cos_tg_row_data
{
    AP_UINT16      overlay_size;     /* size of returned entry   */
    unsigned char  cos_name[8];     /* cos name                  */
    AP_UINT16      tg_row_index;    /* TG row index             */
    COS_TG_ROW     tg_row;          /* TG row information       */
} COS_TG_ROW_DATA;

typedef struct cos_tg_row
{
    TG_DEFINED_CHARS minimum;      /* minimum                  */
    TG_DEFINED_CHARS maximum;     /* maximum                  */
    unsigned char  weight;        /* weight                   */
    unsigned char  reserv1;       /* reserved                  */
} COS_TG_ROW;

typedef struct tg_defined_chars
{
    unsigned char  effect_cap;     /* Effective capacity       */
    unsigned char  reserve1[5];    /* Reserved                  */
    unsigned char  connect_cost;  /* Connection Cost          */
    unsigned char  byte_cost;     /* Byte cost                */
    unsigned char  reserve2;      /* Reserved                  */
    unsigned char  security;      /* Security                  */
    unsigned char  prop_delay;    /* Propagation delay        */
    unsigned char  modem_class;   /* reserved                  */
    unsigned char  user_def_parm_1; /* User-defined parameter 1 */
    unsigned char  user_def_parm_2; /* User-defined parameter 2 */
    unsigned char  user_def_parm_3; /* User-defined parameter 3 */
} TG_DEFINED_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_COS_TG_ROW

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size
指定したデータ・バッファのサイズ。

num_entries
COS TG 行の最大数 (その TG 行のデータが戻されます)。ある範囲でなく、特定の COS TG 行に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options
Communications Server for Linux がデータを戻すための COS TG 行のリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST
リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE
cos_name パラメーターと *tg_row_index* パラメーターを組み合わせで指定したエントリーから開始します。

AP_LIST_FROM_NEXT
cos_name パラメーターと *tg_row_index* パラメーターを組み合わせで指定したエントリーの直後のエントリーから開始します。

リストは、最初に *cos_name*、次に各 COS の *tg_row_index* の順番で並んでいます。リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

cos_name
データを求めるサービス・クラスの名前、またはリストの索引として使用する名前。この名前は 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側に EBCDIC のスペースを入れます。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

tg_row_index
データを求める TG 行番号、またはリストの索引として使用する番号 (1 行目の索引はゼロです)。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

buf_size
指定したバッファに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファ内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファ内の各エントリーは、次のパラメーターで構成されています。

cos_tg_row_data.overlay_size

戻された *cos_tg_row_data* 構造体のサイズ。すなわち、データ・バッファ内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 *cos_tg_row_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

cos_tg_row_data.cos_name

サービス・クラス名。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

cos_tg_row_data.tg_row_index

TG 行の索引 (1 行目の索引はゼロです)。

cos_tg_row_data.tg_row.minimum.effect_cap

実際のビット/秒率の下限 (回線速度)。値は 1 バイトの浮動小数点数としてエンコードされます。このときの公式は、 $0.1 \text{ mmm} * 2^{\text{eeee}}$ で、バイトのビット表現は `b'eeeeemmm'` です。300 ビット/秒の倍数で指定してください。

cos_tg_row_data.tg_row.minimum.connect_cost

接続時間当たりのコストの下限。0 ~ 255 の範囲の整数値で、0 が接続時間当たりの最低コスト、255 が最高コストです。

cos_tg_row_data.tg_row.minimum.byte_cost

バイト当たりのコストの下限。0 ~ 255 の範囲の整数値で、ゼロがバイト当たりの最低コスト、255 が最高コストです。

cos_tg_row_data.tg_row.minimum.security

最小セキュリティー・レベル。値は次のいずれかです。

AP_SEC_NONSECURE

セキュリティーなし。

AP_SEC_PUBLIC_SWITCHED_NETWORK

公衆交換網を介してデータを伝送します。

AP_SEC_UNDERGROUND_CABLE

保護された地下ケーブルを介してデータを伝送します。

AP_SEC_SECURE_CONDUIT

ガードされていない保護されたコンジット (配管) 内の回線を介してデータを伝送します。

AP_SEC_GUARDED_CONDUIT

物理接続による盗聴から保護されたコンジット内の回線を介してデータを伝送します。

AP_SEC_ENCRYPTED

回線を介して伝送する前にデータを暗号化します。

AP_SEC_GUARDED_RADIATION

物理接続による盗聴および電波盗聴から保護された回線を介してデータを伝送します。

cos_tg_row_data.tg_row.minimum.prop_delay

伝搬遅延の下限: 信号をリンクの長さのみ伝送するのに要する時間。値は次のいずれかです。

AP_PROP_DELAY_MINIMUM

最小の伝搬遅延。

AP_PROP_DELAY_LAN

遅延は 480 マイクロ秒未満です (LAN の通常値)。この verb を実行中のノードに対して発行した場合、この値は、DEFINE_COS で AP_PROP_DELAY_LAN または AP_PROP_DELAY_MINIMUM を指定した場合のみ戻されます。

AP_PROP_DELAY_TELEPHONE

遅延は 480 ~ 49,512 マイクロ秒の範囲です (電話網の通常値)。

AP_PROP_DELAY_PKT_SWITCHED_NET

遅延は 49,512 ~ 245,760 マイクロ秒の範囲です (パケット交換ネットワークの通常値)。

AP_PROP_DELAY_SATELLITE

遅延は 245,760 マイクロ秒を超えます (サテライト・リンクの通常値)。

AP_PROP_DELAY_MAXIMUM

最大の伝搬遅延。

cos_tg_row_data.tg_row.minimum.user_def_parm_1 ~

cos_tg_row_data.tg_row.minimum.user_def_parm_3

上記のパラメーターで対応できないその他の TG 特性を含むユーザー定義パラメーターの最小値。これらのパラメーターは、それぞれ 0 ~ 255 の範囲の値に設定されます。

cos_tg_row_data.tg_row.maximum.effect_cap

実際のビット/秒率の上限 (回線速度)。値は 1 バイトの浮動小数点数として

QUERY_COS_TG_ROW

エンコードされます。このときの公式は、 $0.1 \text{ mmm} * 2^{\text{eeee}}$ で、バイトのビット表現は `eeeeemmm` です。300 ビット/秒の倍数で指定してください。

cos_tg_row_data.tg_row.maximum.connect_cost

接続時間当たりのコストの上限。0 ~ 255 の範囲の整数値で、0 が接続時間当たりの最低コスト、255 が最高コストです。

cos_tg_row_data.tg_row.maximum.byte_cost

バイト当たりのコストの上限。0 ~ 255 の範囲の整数値で、0 がバイト当たりの最低コスト、255 が最高コストです。

cos_tg_row_data.tg_row.maximum.security

最大セキュリティー・レベル。値は次のいずれかです。

AP_SEC_NONSECURE

セキュリティーなし。

AP_SEC_PUBLIC_SWITCHED_NETWORK

公衆交換網を介してデータを伝送します。

AP_SEC_UNDERGROUND_CABLE

保護された地下ケーブルを介してデータを伝送します。

AP_SEC_SECURE_CONDUIT

ガードされていない保護されたコンジット (配管) 内の回線を介してデータを伝送します。

AP_SEC_GUARDED_CONDUIT

物理接続による盗聴から保護されたコンジット内の回線を介してデータを伝送します。

AP_SEC_ENCRYPTED

回線を介して伝送する前にデータを暗号化します。

AP_SEC_GUARDED_RADIATION

物理接続による盗聴および電波盗聴から保護された回線を介してデータを伝送します。

AP_SEC_MAXIMUM

最大のセキュリティー。

cos_tg_row_data.tg_row.maximum.prop_delay

伝搬遅延の上限: 信号をリンクの長さのみ伝送するのに要する時間。値は次のいずれかです。

AP_PROP_DELAY_MINIMUM

最小の伝搬遅延。

AP_PROP_DELAY_LAN

遅延は 480 マイクロ秒未満です (LAN の通常値)。

AP_PROP_DELAY_TELEPHONE

遅延は 480 ~ 49,512 マイクロ秒の範囲です (電話網の通常値)。

AP_PROP_DELAY_PKT_SWITCHED_NET

遅延は 49,512 ~ 245,760 マイクロ秒の範囲です (パケット交換ネットワークの通常値)。

AP_PROP_DELAY_SATELLITE

遅延は 245,760 マイクロ秒を超えます (サテライト・リンクの通常値)。この verb を実行中のノードに対して発行した場合、この値は、DEFINE_COS で AP_PROP_DELAY_SATELLITE または AP_PROP_DELAY_MAXIMUM を指定した場合のみ戻されます。

AP_PROP_DELAY_MAXIMUM

最大の伝搬遅延。

cos_tg_row_data.tg_row.maximum.user_def_parm_1 ~

cos_tg_row_data.tg_row.maximum.user_def_parm_3

上記のパラメーターで対応できないその他の TG 特性を含むユーザー定義パラメーターの最大値。これらのパラメーターは、それぞれ 0 ~ 255 の範囲の値に設定されます。

cos_tg_row_data.tg_row.weight

この TG 行に関連した重み。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_COS_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*cos_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_CPIC_SIDE_INFO

この verb は、使用するオプションに応じて、ある記号宛先名のサイド情報エントリーを戻すか、複数の記号宛先名のサイド情報エントリーを戻します。

この verb と CPI-C 関数 Extract_CPIC_Side_Information の違いに注意してください。この verb は構成ファイルを照会し、その結果、すべての Communications Server for Linux CPI-C アプリケーションで使用されるデフォルト情報を戻します。

QUERY_CPIC_SIDE_INFO

CPI-C 関数はサイド情報テーブルのメモリー内にあるアプリケーション独自のコピーに対して照会しますが、そのコピーは、アプリケーションで他の CPI-C サイド情報関数を使用して変更されている可能性があります。

この verb は、ドメイン構成ファイルに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_cplic_side_info
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  *buf_ptr;       /* pointer to buffer        */
    AP_UINT32      buf_size;       /* buffer size              */
    AP_UINT32      total_buf_size; /* total buffer size required */
    AP_UINT16      num_entries;    /* number of entries        */
    AP_UINT16      total_num_entries; /* total number of entries */
    unsigned char  list_options;   /* listing options          */
    unsigned char  reserv3;        /* reserved                  */
    unsigned char  sym_dest_name[8]; /* Symbolic destination name */
} QUERY_CPIC_SIDE_INFO;

typedef struct cplic_side_info_data
{
    AP_UINT16      overlay_size;   /* size of returned entry   */
    unsigned char  sym_dest_name[8]; /* Symbolic destination name */
    unsigned char  reserv1[2];     /* reserved                  */
    CPIC_SIDE_INFO_DEF_DATA def_data;
} CPIC_SIDE_INFO_DATA;

typedef struct cplic_side_info_def_data
{
    unsigned char  description[32]; /* resource description     */
    unsigned char  reserv1[16];    /* reserved                  */
    CPIC_SIDE_INFO side_info;     /* CPIC side info           */
    unsigned char  user_data[24];  /* reserved                  */
} CPIC_SIDE_INFO_DEF_DATA;

typedef struct cplic_side_info
{
    unsigned char  partner_lu_name[17]; /* Fully qualified partner */
                                        /* LU name                  */
    unsigned char  reserved[3];       /* Reserved                  */
    AP_UINT32      tp_name_type;     /* TP name type             */
    unsigned char  tp_name[64];     /* TP name                   */
    unsigned char  mode_name[8];     /* Mode name                 */
    AP_UINT32      conversation_security_type; /* Conversation security */
                                        /* type                     */
    unsigned char  security_user_id[10]; /* User ID                  */
    unsigned char  security_password[10]; /* Password                 */
    unsigned char  lu_alias[8];     /* LU alias                  */
} CPIC_SIDE_INFO;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_CPIC_SIDE_INFO

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

記号宛先名の最大数 (その記号宛先名のデータが戻されます)。ある範囲でなく、特定のシンボリック宛先名に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのシンボリック宛先名リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

sym_dest_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

sym_dest_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

sym_dest_name

データを求める記号宛先名、またはリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、大文字の A ~ Z と数字の 0 ~ 9 からなる ASCII スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファ内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファ内の各エントリーは、次のパラメーターで構成されています。

cpic_side_info_data.overlay_size

戻された *cpic_side_info_data* 構造体のサイズ。すなわち、データ・バッファ内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 *cpic_side_info_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは *Communications Server for Linux* の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

cpic_side_info_data.sym_dest_name

戻されたサイド情報エントリーの記号宛先名。

cpic_side_info_data.def_data.description

サイド情報エントリーを記述したヌルで終了するテキスト・ストリング (そのサイド情報エントリーの定義で指定したもの)。

cpic_side_info_data.def_data.side_info.partner_lu_name

パートナー LU の完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

cpic_side_info_data.def_data.side_info.tp_name_type

ターゲット TP のタイプ (TP 名に有効な文字は、TP タイプによって決まります)。値は次のいずれかです。

XC_APPLICATION_TP

アプリケーション TP。TP 名のすべての文字は、有効な ASCII 文字でなければなりません。

XC_SNA_SERVICE_TP

サービス TP。TP 名は、4 文字の 16 進数字を表す 8 文字の ASCII ストリングで指定しなければなりません。たとえば、名前の 16 進表記が 0x21F0F0F8 の場合、*def_data.side_info.tp_name* パラメーターは、8 文字のストリング「21F0F0F8」に設定します。

先頭文字 (2 バイトで表す) は、0x0 ~ 0x3F の範囲の 16 進値でなければなりません。ただし、0x0E と 0x0F は除きます。残りの文字 (それぞれ 2 バイトで表す) は有効な EBCDIC 文字でなければなりません。

cpic_side_info_data.def_data.side_info.tp_name

ターゲット TP の TP 名。これは、右側にスペースを入れた 64 バイトの ASCII 文字ストリングです。

cpic_side_info_data.def_data.side_info.mode_name

ターゲット TP にアクセスするために使用するモードの名前。これは、右側にスペースを入れた 8 バイトの ASCII 文字ストリングです。

cpic_side_info_data.def_data.side_info.conversation_security_type

ターゲット TP が会話セキュリティーを使用するかどうかを指定します。値は次のいずれかです。

XC_SECURITY_NONE

ターゲット TP は、会話セキュリティーを使用しません。

XC_SECURITY_PROGRAM

ターゲット TP は、会話セキュリティーを使用します。ターゲット TP へアクセスするために、以下に指定した *security_user_id* パラメーターと *security_password* パラメーターを使用します。

XC_SECURITY_PROGRAM_STRONG

ローカル・ノードがパスワードを平文でネットワーク上に送信してはならない点を除けば、XC_SECURITY_PROGRAM と同じです。この値を使用できるのは、リモート・システムがパスワード置換をサポートしている場合のみです。

XC_SECURITY_SAME

ターゲット TP は会話セキュリティーを使用し、ローカル TP から「検査済み」インディケーターを受け入れることができます。(これは、ローカル TP 自体が別の TP から呼び出され、その TP が提供したセキュリティー・ユーザー ID とパスワードをローカル TP が検証済みであることを示しています。) ターゲット TP にアクセスするために、以下に指定した *security_user_id* パラメーターが使用されます。したがって、パスワードは必要ありません。

cpic_side_info_data.def_data.side_info.security_user_id

パートナー TP にアクセスするために使用するユーザー ID。このパラメーターは、*conversation_security_type* パラメーターを XC_SECURITY_NONE に設定した場合には使用されません。

cpic_side_info_data.def_data.side_info.security_password

パートナー TP にアクセスするために使用するパスワード。このパラメーターは、*conversation_security_type* パラメーターを XC_SECURITY_PROGRAM または XC_SECURITY_PROGRAM_STRONG に設定した場合のみ使用されます。

cpic_side_info_data.def_data.side_info.lu_alias

ターゲット TP との通信に使用するローカル LU の別名。この別名は、ローカル・システム上で表示可能な文字を使用した文字ストリングです。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

QUERY_CPIC_SIDE_INFO

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
```

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

AP_INVALID_SYM_DEST_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*sym_dest_name* パラメーターは無効でした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_CS_TRACE

この verb は Communications Server for Linux LAN 上のコンピューター間で送信されるデータの現行トレース・オプションに関する情報を戻します。トレース・オプションについて詳しくは、「*Communications Server for Linux 管理ガイド*」を参照してください。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_cs_trace
{
    AP_UINT16      opcode;           /* verb operation code */
    unsigned char  reserv2;         /* reserved */
    unsigned char  format;         /* reserved */
    AP_UINT16      primary_rc;     /* primary return code */
    AP_UINT32      secondary_rc;   /* secondary return code */
    unsigned char  dest_sys[128];  /* node to which messages are traced */
    unsigned char  reserv4[4];     /* reserved */
}
```

```

AP_UINT16      trace_flags;      /* trace flags */
AP_UINT16      trace_direction; /* direction (send/rcv/both) to trace */
unsigned char   reserv3[8];      /* Reserved */
} QUERY_CS_TRACE;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_CS_TRACE

dest_sys

トレース・オプションが照会されるサーバー名。この名前は ASCII ストリングです。名前が 128 文字に満たない場合は、右側にスペースを入れます。

この *verb* の発行先となるコンピューター (NOF API 呼び出しの *target_handle* パラメーターで識別されます) と LAN 上の他のサーバーの間を流れるメッセージでトレース・オプションを照会する場合は、ここでそのサーバーの名前を指定します。

コンピューター名に . (ピリオド) 文字が含まれている場合、Communications Server for Linux はこれが完全修飾名であると想定します。ピリオドが含まれていない場合は、DNS ルックアップを実行してコンピューター名を判別します。

デフォルト・トレース・オプション (宛先システムの名前を指定しないで SET_CS_TRACE *verb* で設定) を照会する場合は、このパラメーターをすべて ASCII スペース文字に設定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

trace_flags

現在活動状態であるトレースのタイプ。これらのトレース・タイプについて詳しくは、732 ページの『SET_CS_TRACE』を参照してください。

活動状態であるトレースがない場合、またはすべてのタイプのトレースが活動状態である場合は、値は次のいずれかです。

AP_NO_TRACE

トレースしません。

AP_ALL_TRACE

すべてのタイプをトレースします。

特定のインターフェースのトレースを使用している場合、このパラメーターは、次のリストの値の 1 つ以上を論理 OR 演算で組み合わせた値に設定されます。

AP_CS_ADMIN_MSG

クライアント/サーバー・トポロジーに関する内部メッセージ

AP_CS_DATAGRAM

データグラム・メッセージ

AP_CS_DATA

データ・メッセージ

trace_direction

トレースがアクティブである 1 つ以上の方向を指定します。 *trace_flags* が AP_NO_TRACE に設定されている場合、このパラメーターは使用されません。値は次のいずれかです。

AP_CS_SEND

ターゲット・コンピューターから *dest_sys* で定義したコンピューターに流れるメッセージがトレースされます。

AP_CS_RECEIVE

dest_sys で定義したコンピューターからターゲット・コンピューターに流れるメッセージがトレースされます。

AP_CS_BOTH

両方向に流れるメッセージがトレースされます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_NAME_NOT_FOUND

dest_sys パラメーターで指定したサーバーは存在していないか、開始されていません。

AP_LOCAL_SYSTEM

dest_sys パラメーターで指定したサーバーは、この *verb* の発行先のターゲット・ノードと同じです。

AP_INVALID_TARGET

この *verb* はスタンドアロン・サーバーで発行されました。この *verb* はクライアント/サーバー・システムでのみ発行できます。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DEFAULT_PU

QUERY_DEFAULT_PU により、ユーザーはデフォルト PU (DEFINE_DEFAULT_PU を使用して定義された PU) について照会できます。

VCB 構造体

```
typedef struct query_default_pu
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  def_pu_name[8]; /* default PU name             */
    unsigned char  description[32]; /* resource description        */
    unsigned char  reserv1[16];    /* reserved                     */
    unsigned char  def_pu_sess[8]; /* PU name of active default session */
    unsigned char  reserv3[16];    /* reserved                     */
} QUERY_DEFAULT_PU;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DEFAULT_PU

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

def_pu_name

最新の DEFINE_DEFAULT_PU *verb* で指定した PU の名前。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。このフィールドがすべて 2 進ゼロに設定されている場合、それは DEFINE_DEFAULT_PU *verb* がまだ発行されたことがないか、*pu_name* パラメーターをすべて 2 進ゼロに指定した DEFINE_DEFAULT_PU *verb* を発行することによってデフォルトの PU が削除されたことを示しています。

description

デフォルト PU を記述したヌルで終了するテキスト・String (そのデフォルト PU 定義で指定したもの)。

def_pu_sess

現在活動状態であるデフォルト PU セッションに関連した PU の名前。

通常、このパラメーターには *def_pu_name* フィールドと同じ値が入っています。ただし、デフォルト PU が定義され、それに関連したセッションが活動状態でない場合、Communications Server for Linux は、定義されているデフォルト PU に関連したセッションが活動状態になるまで、以前のデフ

QUERY_DEFAULT_PU

オルト PU に関連したセッションを使用し続けます。その場合、このパラメーターは前のデフォルト PU の名前を示し、*def_pu_name* フィールドの内容とは異なります。

活動状態である PU セッションがない場合、このフィールドはすべてゼロに設定されます。

戻りパラメーター: ノードが始動しない場合

ノードがまだ始動されていないために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_NODE_NOT_STARTED
```

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DEFAULTS

QUERY_DEFAULTS により、ユーザーはノードに定義したデフォルト・パラメーター (DEFINE_DEFAULTS を使用して定義されたデフォルト・パラメーター) について照会することができます。

VCB 構造体

```
typedef struct query_defaults
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    DEFAULT_CHARS  default_chars;  /* default parameters         */
} QUERY_DEFAULTS;

typedef struct default_chars
{
    unsigned char  description[32]; /* resource description         */
    unsigned char  reserv2[16];    /* reserved                     */
    unsigned char  mode_name[8];   /* default mode name           */
    unsigned char  implicit_plu_forbidden; /* disallow implicit PLUs? */
    unsigned char  specific_security_codes; /* generic security sensecodes? */
    AP_UINT16      limited_timeout; /* timeout for limited sessions */
    unsigned char  reserv[244];    /* reserved                     */
} DEFAULT_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

```
opcode AP_QUERY_DEFAULTS
```

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

default_chars.description

デフォルト・パラメーターを記述したヌルで終了するテキスト・STRING (DEFINE_DEFAULTS で指定したもの)。

default_chars.mode_name

デフォルト・モードの名前。セッションを開始しようとするときに、認識されていないモード名をアプリケーションで指定した場合、認識されていないモードのデフォルト定義として、このモードからのパラメーターが使用されます。

このモード名は、8 バイトのタイプ A の EBCDIC 文字STRINGです。DEFINE_DEFAULTS verb を使用してデフォルトのモード名を指定しなかった場合、このパラメーターの 8 つのビットはゼロに設定されます。

default_chars.implicit_plu_forbidden

Communications Server for Linux で不明なパートナー LU の代わりに暗黙の定義を使用するかどうかを指定します。値は次のいずれかです。

AP_YES Communications Server for Linux で不明なパートナー LU の代わりに暗黙の定義を使用しません。すべてのパートナーは明示的に定義されなければなりません。

AP_NO Communications Server for Linux で不明なパートナー LU の代わりに暗黙の定義を使用します。

default_chars.specific_security_codes

セキュリティーの認証または許可が失敗したときに、Communications Server for Linux では特定のセンス・コードを使用するかどうかを指定します。特定のセンス・コードは、セッション上でそれらのセンス・コードのサポートを報告したパートナー LU にのみ戻されます。値は次のいずれかです。

AP_YES Communications Server for Linux では特定のセンス・コードを使用します。

AP_NO Communications Server for Linux では特定のセンス・コードを使用しません。

default_chars.limited_timeout

使用されていない限定リソース競合勝者セッションを非活動化するまでのタイムアウトを指定します。範囲は 0 ~ 65,535 秒です。

戻りパラメーター: ノードが始動しない場合

ノードがまだ始動されていないために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_NODE_NOT_STARTED

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DIRECTORY_ENTRY

QUERY_DIRECTORY_ENTRY は、ディレクトリー・データベース内のリソースに関する情報を戻します。この verb では、使用するオプションに応じて、特定のリソースまたは複数のリソースについての要約情報または詳細情報を戻すことができます。

実行中のモードに対して発行した場合、この verb は (DEFINE_DIRECTORY_ENTRY または DEFINE_ADJACENT_LEN_NODE を使用して) 明示的に定義したリソースと、動的に検索されたリソースの両方に関する情報を戻します。ノードが実行中でない場合は、明示的に定義したリソースのみが戻されます。

この verb をエンド・ノードに対して発行した場合、ディレクトリーにはエンド・ノードとそのリソースに関する情報のみが入っており、その他のノードに関する情報は入っていません。最初に戻されるエントリーはエンド・ノード自体のエントリーで、そのあとにそのエンド・ノードの各 LU のエントリーが続きます。(エンド・ノードのネットワーク・ノード・サーバーに関するエントリーは戻されません。)

この verb をネットワーク・ノードに対して発行した場合、ディレクトリーには複数のネットワーク・ノードとそれらのノードに関連したエンド・ノードおよび LU に関する情報が入っています。各ネットワーク・ノードについて、次の順序で情報が戻されます。

1. ネットワーク・ノード。
2. そのネットワーク・ノードが所有する LU。
3. そのネットワーク・ノードに関連した最初のエンド・ノード。
4. そのエンド・ノードが所有する LU。
5. そのネットワーク・ノードに関連した他のエンド・ノード。各エンド・ノードのあとにそのエンド・ノードの LU が続きます。

VCB 構造体

```
typedef struct query_directory_entry
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;          /* reserved                      */
    unsigned char  format;           /* reserved                      */
    AP_UINT16      primary_rc;       /* primary return code          */
    AP_UINT32      secondary_rc;     /* secondary return code        */
    unsigned char  *buf_ptr;         /* pointer to buffer            */
    AP_UINT32      buf_size;         /* buffer size                  */
    AP_UINT32      total_buf_size;   /* total buffer size required   */
    AP_UINT16      num_entries;      /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries      */
    unsigned char  list_options;     /* listing options              */
    unsigned char  reserv3;          /* reserved                      */
    unsigned char  resource_name[17]; /* network qualified resource   */
    /* name                          */
    unsigned char  reserv4;          /* reserved                      */
    AP_UINT16      resource_type;     /* Resource type                */
    unsigned char  parent_name[17];  /* parent name filter           */
    unsigned char  reserv5;          /* reserved                      */
    AP_UINT16      parent_type;      /* parent type                  */
    unsigned char  reserv6[24];      /* reserved                      */
} QUERY_DIRECTORY_ENTRY;
```

```

typedef struct directory_entry_summary
{
    AP_UINT16      overlay_size;          /* size of this entry      */
    unsigned char  resource_name[17];     /* network qualified resource
                                           /* name                    */
    unsigned char  reserve1;              /* reserved                */
    AP_UINT16      resource_type;         /* Resource type           */
    unsigned char  description[32];       /* resource description    */
    unsigned char  reserv1[16];           /* reserved                */
    AP_UINT16      real_owning_cp_type;    /* CP type of real owner   */
    unsigned char  real_owning_cp_name[17]; /* CP name of real owner   */
    unsigned char  reserve2;              /* reserved                */
} DIRECTORY_ENTRY_SUMMARY;

typedef struct directory_entry_detail
{
    AP_UINT16      overlay_size;          /* size of this entry      */
    unsigned char  resource_name[17];     /* network qualified res name
                                           /*                        */
    unsigned char  reserv1a;              /* reserved                */
    AP_UINT16      resource_type;         /* Resource type           */
    unsigned char  description[32];       /* resource description    */
    unsigned char  reserv2[16];           /* reserved                */
    unsigned char  parent_name[17];       /* Network qualified parent name
                                           /*                        */
    unsigned char  reserv1b;              /* reserved                */
    AP_UINT16      parent_type;           /* Parent resource type    */
    unsigned char  entry_type;            /* Type of the directory entry
                                           /*                        */
    unsigned char  location;              /* Resource location       */
    AP_UINT16      real_owning_cp_type;    /* CP type of real owner   */
    unsigned char  real_owning_cp_name[17]; /* CP name of real owner   */
    unsigned char  reserv1c;              /* reserved                */
    AP_UINT16      supplier_cp_type;      /* CP type of supplier     */
    unsigned char  supplier_cp_name[17];  /* CP name of supplier     */
    unsigned char  reserva;               /* reserved                */
} DIRECTORY_ENTRY_DETAIL;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DIRECTORY_ENTRY

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

リソースの最大数 (そのリソースのデータが戻されます)。ある範囲でなく、特定のリソースに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

QUERY_DIRECTORY_ENTRY

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

parent_name、*resource_name*、*resource_type* の各パラメーターを組み合わせて指定したエントリーから開始します。

AP_LIST_FROM_NEXT

parent_name、*resource_name*、*resource_type* の各パラメーターを組み合わせて指定したエントリーの直後のエントリーから開始します。

リストは、最初に *parent_name*、次に *resource_name*、最後に *resource_type* の順番で並んでいます。リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

resource_name

情報を求めるリソースの完全修飾名、またはリソース・リストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A String 文字からなるネットワーク名で構成されます。

resource_type

情報を求めるリソースのタイプ。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。値は次のいずれかです。

AP_ENCP_RESOURCE

エンド・ノードまたは LEN ノード

AP_NNCP_RESOURCE

ネットワーク・ノード

AP_LU_RESOURCE

LU

parent_name

親リソースの完全修飾リソース名。LU の場合、親リソースは所有する側の制御点であり、エンド・ノードまたは LEN ノードの場合、親リソースはネットワーク・ノード・サーバーです。指定した親に所属するエントリーのみを戻すには、このパラメーターを親リソースの名前に設定し、*parent_type* を親のリソース・タイプに設定します。すべてのエントリーを戻すには、両方のパラメーターをすべてのビットをゼロに設定します。

この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

parent_type

親リソースのリソース・タイプ。指定した親に所属するエントリーのみを戻すには、このパラメーターを親リソースのタイプに設定します。すべてのエントリーを戻すには、このパラメーターのすべてのビットをゼロに設定します。値は次のいずれかです。

AP_ENCP_RESOURCE

エンド・ノード (エンド・ノードによって所有された LU リソースの場合)

AP_NNCP_RESOURCE

ネットワーク・ノード (ネットワーク・ノードによって所有された LU リソースの場合、または EN リソースまたは LEN リソースの場合)。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

directory_entry_summary.overlay_size

戻された `directory_entry_summary` 構造体のサイズ。すなわち、データ・バッファー内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファーを調べて、各 `directory_entry_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後の

QUERY_DIRECTORY_ENTRY

リリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

directory_entry_summary.resource_name

リソースの完全修飾名。この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A スtring文字からなるネットワーク名で構成されます。

directory_entry_summary.resource_type

リソースのタイプ。次のいずれかです。

AP_ENCP_RESOURCE

エンド・ノードまたは LEN ノード

AP_NNCP_RESOURCE

ネットワーク・ノード

AP_LU_RESOURCE

LU

directory_entry_summary.description

ディレクトリー・エントリーを記述したヌルで終了するテキスト・String (そのディレクトリー・エントリーの定義で指定したもの)。

directory_entry_summary.real_owning_cp_type

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には予約済みです。

このディレクトリー・エントリーによって識別されるリソースを所有する実 CP が、親リソースであるのか、別のノードであるのかを指定します。次のいずれかです。

AP_NONE

実際の所有者は親リソースです。

AP_ENCP_RESOURCE

実際の所有者は、親リソースではないエンド・ノードです。たとえば、分岐ネットワーク・ノード (Branch Network Node: BrNN) のドメインでエンド・ノードがリソースを所有する場合、この BrNN のネットワーク・ノード・サーバーのディレクトリーには、親リソースとして BrNN を含みますが、実際の所有 CP はエンド・ノードです。

directory_entry_summary.real_owning_cp_name

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には予約済みです。

real_owning_cp_type パラメーターがリソースの実際の所有者は親でないことを示す場合には、このパラメーターは、リソースを所有する CP の完全修飾名を指定します。それ以外の場合には、使用しません。

この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ～ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ～ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

directory_entry_detail.overlay_size

戻された *directory_entry_detail* 構造体のサイズ。すなわち、データ・バッファ内では次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 *directory_entry_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

directory_entry_detail.resource_name

リソースの完全修飾名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ～ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ～ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

directory_entry_detail.resource_type

リソースのタイプ。次のいずれかです。

AP_ENCP_RESOURCE

エンド・ノードまたは LEN ノード

AP_NNCP_RESOURCE

ネットワーク・ノード

AP_LU_RESOURCE

LU

directory_entry_detail.description

ディレクトリー・エントリーを記述したヌルで終了するテキスト・ストリング (そのディレクトリー・エントリーの定義で指定したもの)。

directory_entry_detail.parent_name

親リソースの完全修飾リソース名。LU の場合、親リソースは所有する側の制御点であり、エンド・ノードまたは LEN ノードの場合、親リソースはネットワーク・ノード・サーバーです。ネットワーク・ノード・リソースの場合、このパラメーターは使用されません。

この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ～ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ～ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

QUERY_DIRECTORY_ENTRY

directory_entry_detail.parent_type

親リソースのリソース・タイプ。ネットワーク・ノード・リソースの場合、このパラメーターは使用されません。それ以外の場合は、このパラメーターは次のいずれかです。

AP_ENCP_RESOURCE

エンド・ノード (エンド・ノードによって所有された LU リソースの場合)

AP_NNCP_RESOURCE

ネットワーク・ノード (ネットワーク・ノードによって所有された LU リソースの場合、または EN リソースまたは LEN リソースの場合)。

directory_entry_detail.entry_type

ディレクトリー・エントリーのタイプを指定します。次のいずれかです。

AP_HOME

ローカル・リソース。

AP_CACHE

キャッシュに入れられたエントリー。

AP_REGISTER

登録済みリソース (NN のみ)。

directory_entry_detail.location

リソースの位置を指定します。次のいずれかです。

AP_LOCAL

リソースは、ローカル・ノードにあります。

AP_DOMAIN

リソースは、接続したエンド・ノードに所属します。

AP_CROSS_DOMAIN

リソースは、ローカル・ノードのドメイン内にありません。

directory_entry_detail.real_owning_cp_type

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には予約済みです。

このディレクトリー・エントリーによって識別されるリソースを所有する実 CP が、親リソースであるのか、別のノードであるのかを指定します。次のいずれかです。

AP_NONE

実際の所有者は親リソースです。

AP_ENCP_RESOURCE

実際の所有者は、親リソースではないエンド・ノードです。たとえば、分岐ネットワーク・ノード (Branch Network Node: BrNN) のドメインでエンド・ノードがリソースを所有する場合、この BrNN のネットワーク・ノード・サーバーのディレクトリーには、親リソースとして BrNN を含みますが、実際の所有 CP はエンド・ノードです。

directory_entry_detail.real_owning_cp_name

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には予約済みです。

real_owning_cp_type パラメーターがリソースの実際の所有者は親でないことを示す場合には、このパラメーターは、リソースを所有する CP の完全修飾名を指定します。それ以外の場合には、使用しません。

この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

directory_entry_detail.supplier_cp_type

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には予約済みです。

リソースの所有 CP ではない別のノードが、このディレクトリー・エントリーを登録したかどうかを指定します。次のいずれかです。

AP_NONE

ディレクトリー・エントリーは、登録されなかったか、その所有側 CP によって登録されました。

AP_ENCP_RESOURCE

ディレクトリー・エントリーは、その所有側 CP ではないノードによって登録されました。たとえば、それ自体ローカル・ノードのドメインにある分岐ネットワーク・ノード (BrNN) のドメインにあるエンド・ノードがリソースを所有している場合、そのリソースをローカル・ノードに登録した BrNN がそのリソースの提供者になりますが、実所有側 CP はエンド・ノードです。

directory_entry_detail.supplier_cp_name

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には予約済みです。

supplier_cp_type パラメーターが、所有リソースではないノードによってディレクトリー・エントリーが登録されたことを示す場合には、このパラメーターは、登録を提供した CP の完全修飾名を指定します。それ以外の場合には、使用しません。

この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

QUERY_DIRECTORY_ENTRY

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_RES_NAME

list_options パラメーターが、指定した名前から始まるすべてのエン
トリーをリストするために、AP_LIST_INCLUSIVE に設定されていま
すが、*resource_name* パラメーターは無効でした。

AP_INVALID_RES_TYPE

resouce_type パラメーターが、有効な値に設定されていませんで
した。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんで
した。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、
AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記
以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DIRECTORY_LU

QUERY_DIRECTORY_LU は、ディレクトリー・データベースから LU のリストを
戻します。この verb は、使用するオプションに応じて、特定の LU に関する情報
を入手するか、複数の LU に関する情報を入手するために使用できます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_directory_lu
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  *buf_ptr;       /* pointer to buffer           */
    AP_UINT32      buf_size;       /* buffer size                 */
    AP_UINT32      total_buf_size; /* total buffer size required  */
    AP_UINT16      num_entries;    /* number of entries          */
    AP_UINT16      total_num_entries; /* total number of entries    */
    unsigned char  list_options;   /* listing options            */
    unsigned char  reserv3;       /* reserved                   */
    unsigned char  lu_name[17];    /* network qualified lu name   */
} QUERY_DIRECTORY_LU;

typedef struct directory_lu_summary
{
    AP_UINT16      overlay_size;    /* size of returned entry      */
}
```

```

    unsigned char    lu_name[17];          /* network qualified lu name */
    unsigned char    description[32];     /* resource description */
    unsigned char    reserv1[16];        /* reserved */
} DIRECTORY_LU_SUMMARY;

typedef struct directory_lu_detail
{
    AP_UINT16        overlay_size;        /* size of returned entry */
    unsigned char    lu_name[17];        /* network qualified lu name */
    unsigned char    description[32];     /* resource description */
    unsigned char    reserv1[16];        /* reserved */
    unsigned char    server_name[17];     /* network qualified server name */
    unsigned char    lu_owner_name[17];   /* network qualified lu owner name */
    unsigned char    location;           /* Resource location */
    unsigned char    entry_type;         /* Type of the directory entry */
    unsigned char    wild_card;          /* type of wildcard entry */
    unsigned char    apparent_lu_owner_name[17]; /* name of apparent LU owner */
    unsigned char    reserva[3];        /* reserved */
} DIRECTORY_LU_DETAIL;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DIRECTORY_LU

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size
指定したデータ・バッファのサイズ。

num_entries
LU の最大数 (その LU のデータが戻されます)。ある範囲でなく、特定の LU に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options
Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY
要約情報のみ。

AP_DETAIL
詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST
リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE
lu_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

lu_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入力する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

lu_name

情報を求める LU の完全修飾名、または LU のリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A スtring 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring 文字からなるネットワーク名で構成されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

directory_lu_summary.overlay_size

戻された *directory_lu_summary* 構造体のサイズ。すなわち、データ・バッファー内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *directory_lu_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の *sizeof()* 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

directory_lu_summary.lu_name

LU の完全修飾名。この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、1 ～ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ～ 8 文字の A String文字からなるネットワーク名で構成されます。

directory_lu_summary.description

ディレクトリー・エントリーを記述したヌルで終了するテキスト・String (そのディレクトリー・エントリーの定義で指定したもの)。

directory_lu_detail.overlay_size

戻された `directory_lu_detail` 構造体のサイズ。すなわち、データ・バッファ内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 `directory_lu_detail` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

directory_lu_detail.lu_name

LU の完全修飾名。この名前は 17 バイトの EBCDIC Stringで、右側に EBCDIC スペースを入れます。この名前は、1 ～ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ～ 8 文字の A String文字からなるネットワーク名で構成されます。

directory_lu_detail.description

ディレクトリー・エントリーを記述したヌルで終了するテキスト・String (そのディレクトリー・エントリーの定義で指定したもの)。

directory_lu_detail.server_name

その LU の接続先ノードの完全修飾名。この名前は 17 バイトの EBCDIC Stringで、右側に EBCDIC スペースを入れます。この名前は、1 ～ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ～ 8 文字の A String文字からなるネットワーク名で構成されます。

directory_lu_detail.lu_owner_name

その LU を所有するノードの完全修飾名。この名前は 17 バイトの EBCDIC Stringで、右側に EBCDIC スペースを入れます。この名前は、1 ～ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ～ 8 文字の A String文字からなるネットワーク名で構成されます。

directory_lu_detail.location

リソースの位置を指定します。次のいずれかです。

AP_LOCAL

リソースは、ローカル・ノードにあります。

QUERY_DIRECTORY_LU

AP_DOMAIN

リソースは、接続したエンド・ノードに所属します。

AP_CROSS_DOMAIN

リソースは、ローカル・ノードのドメイン内にありません。

directory_lu_detail.entry_type

リソースのタイプを指定します。次のいずれかです。

AP_HOME

ローカル・リソース。

AP_CACHE

キャッシュに入れられたエントリー。

AP_REGISTER

登録済みリソース (NN のみ)。

directory_lu_detail.wild_card

LU エントリーが明示的な名前であるか、ある範囲の名前に一致するワイルドカード値であるかを指定します。次のいずれかです。

AP_EXPLICIT

エントリーは明示的な LU 名です。

AP_FULL_WILDCARD

エントリーはどの LU 名にも一致する完全ワイルドカード値です。

AP_PARTIAL_WILDCARD

エントリーは部分ワイルドカードで、名前のブランクでない文字を使用して LU 名と突き合わせられます。

AP_OTHER

不明なタイプの LU エントリー。

directory_lu_detail.apparent_lu_owner_name

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には予約済みです。

この LU の外見上の所有 CP が実際の所有 CP ではない場合、このパラメーターは、外見上の所有 CP の完全修飾名を指定します。それ以外の場合には、使用しません。たとえば、分岐ネットワーク・ノード (Branch Network Node: BrNN) のドメインでエンド・ノードがリソースを所有する場合、この BrNN のネットワーク・ノード・サーバーのディレクトリーには、外見上の所有者として BrNN を含みますが、実際の所有 CP はエンド・ノードです。

この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String文字からなるネットワーク名で構成されます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

list_options パラメーターが、指定した名前から始まるすべてのエン
トリーをリストするために、AP_LIST_INCLUSIVE に設定されていま
すが、*lu_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんで
した。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、
AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記
以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DIRECTORY_STATS

QUERY_DIRECTORY_STATS は、ディレクトリー・データベース統計を戻します。
この統計を使用して、ネットワーク検索のトラフィックのレベルを判断できます。
ネットワーク・ノードの場合は、ディレクトリー・キャッシュの使用量に関する情
報を戻します。この情報を使用して適切なキャッシュ・サイズを決定し、それを
DEFINE NODE verb で指定することができます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_directory_stats
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;              /* reserved                  */
    unsigned char  format;               /* reserved                  */
    AP_UINT16      primary_rc;           /* primary return code      */
    AP_UINT32      secondary_rc;         /* secondary return code    */
    AP_UINT32      max_caches;           /* maximum number of cache  */
                                                /* entries                   */
    AP_UINT32      cur_caches;           /* cache entry count        */
    AP_UINT32      cur_home_entries;     /* home entry count         */
    AP_UINT32      cur_reg_entries;      /* registered entry count   */
    AP_UINT32      cur_directory_entries; /* current number of directory
                                                /* entries                   */
    AP_UINT32      cache_hits;           /* count of cache finds     */
    AP_UINT32      cache_misses;        /* count of resources found
                                                /* by broadcast search     */
                                                /* (not in cache)         */
    AP_UINT32      in_locates;           /* locates in               */
    AP_UINT32      in_bcast_locates;     /* broadcast locates in    */
    AP_UINT32      out_locates;         /* locates out              */
    AP_UINT32      out_bcast_locates;    /* broadcast locates out   */
    AP_UINT32      not_found_locates;    /* unsuccessful locates     */
}
```

QUERY_DIRECTORY_STATS

```
AP_UINT32      not_found_bcast_locates; /* unsuccessful broadcast */
/* locates */
AP_UINT32      locates_outstanding; /* total outstanding locates */
unsigned char  reserva[20]; /* reserved */
} QUERY_DIRECTORY_STATS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DIRECTORY_STATS

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

max_caches

ネットワーク・ノードの場合は、許可されるキャッシュ・エントリーの最大数。

cur_caches

ネットワーク・ノードの場合は、現在のキャッシュ・エントリーの数。

cur_home_entries

現在のホーム・エントリーの数。

cur_reg_entries

現在登録されているエントリーの数。

cur_directory_entries

現在ディレクトリーにあるエントリーの合計数。

cache_hits

ネットワーク・ノードの場合は、成功したキャッシュが検索された数。このカウントは、ローカル・ディレクトリー・キャッシュ内でリソースが検索されるたびに増加します。

cache_misses

ネットワーク・ノードの場合は、あるリソースがブロードキャスト検索要求により検索された回数。このカウントは、ローカル・ディレクトリー・キャッシュ内で検索されなかったリソースがあとでブロードキャスト検索要求により検索されるたびに増加します。

注: *cache_hits* と *cache_misses* の 2 つのカウントは、ディレクトリー・キャッシュのサイズ (DEFINE_NODE で指定された) を調整できるように維持されます。時間が経過するにつれて *cache_misses* が増加する場合は、ディレクトリー・キャッシュのサイズが小さすぎることを示します。 *cache_hits* が一定して増加し、*cache_misses* が変わらない場合は、キャッシュのサイズがほぼ正しいことを示します。

in_locates

受信された有向検索要求の数。

in_bcast_locates

ネットワーク・ノードの場合は、受信されたブロードキャスト検索要求の数。

out_locates

送信された有向検索要求の数。

out_bcast_locates

ネットワーク・ノードの場合は、送信されたブロードキャスト検索要求の数。

not_found_locates

「not found」を戻した有向検索要求の数。

not_found_bcast_locates

ネットワーク・ノードの場合は、「not found」を戻したブロードキャスト検索要求の数。

locates_outstanding

現在の未処理の検索要求の数 (有向検索要求とブロードキャスト検索要求の両方)。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DLC

QUERY_DLC は、DLC に関する情報を戻します。この情報は、「決定済みデータ」(実行中に動的に収集されたデータ) および「定義済みデータ」(DEFINE_DLC で提供されたデータ) として構造化されます。

この verb は、使用するオプションに応じて、特定の DLC または複数の DLC についての要約情報または詳細情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_dlc
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  *buf_ptr;       /* pointer to buffer           */
    AP_UINT32      buf_size;       /* buffer size                 */
    AP_UINT32      total_buf_size; /* total buffer size required  */
    AP_UINT16      num_entries;    /* number of entries           */
    AP_UINT16      total_num_entries; /* total number of entries    */
    unsigned char  list_options;   /* listing options             */
    unsigned char  reserv3;       /* reserved                     */
    unsigned char  dlc_name[8];    /* name of DLC                 */
} QUERY_DLC;

typedef struct dlc_summary
{
    AP_UINT16      overlay_size;    /* size of returned entry      */
    unsigned char  dlc_name[8];    /* name of DLC                 */
}
```

QUERY_DLC

```
    unsigned char  description[32];    /* resource description    */
    unsigned char  reserv1[16];       /* reserved                */
    unsigned char  state;             /* State of the DLC       */
    unsigned char  dlc_type;          /* DLC type                */
} DLC_SUMMARY;

typedef struct dlc_detail
{
    AP_UINT16      overlay_size;      /* size of returned entry  */
    unsigned char  dlc_name[8];       /* name of DLC             */
    unsigned char  reserv2[2];        /* reserved                */
    DLC_DET_DATA   det_data;          /* Determined data         */
    DLC_DEF_DATA   def_data;          /* Defined data            */
} DLC_DETAIL;

typedef struct dlc_det_data
{
    unsigned char  state;             /* State of the DLC       */
    unsigned char  reserv3[3];        /* reserved                */
    unsigned char  reserva[20];       /* reserved                */
} DLC_DET_DATA;

typedef struct dlc_def_data
{
    unsigned char  description[32];    /* resource description    */
    unsigned char  initially_active;   /* is DLC initially active? */
    unsigned char  reserv1[15];       /* reserved                */
    unsigned char  dlc_type;          /* DLC type                */
    unsigned char  neg_ls_supp;       /* negotiable link station support */
    unsigned char  port_types;        /* port types supported by DLC type */
    unsigned char  hpr_only;          /* only support HPR?      */
    unsigned char  reserv3;           /* reserved                */
    unsigned char  retry_flags;       /* reserved                */
    AP_UINT16      max_activation_attempts; /* reserved                */
    AP_UINT16      activation_delay_timer; /* reserved                */
    unsigned char  reserv4[4];        /* reserved                */
    AP_UINT16      dlc_spec_data_len;  /* Length of DLC specific data */
} DLC_DEF_DATA;
```

DLC 固有のデータについて詳しくは、100 ページの『DEFINE_DLC』を参照してください。このデータのデータ構造体は `dlc_def_data` 構造体に従いますが、スペースが入るため、4 バイト目の境界から始まります。

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DLC

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

DLC の最大数 (その DLC のデータが戻されます)。ある範囲でなく、特定の DLC に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

dlc_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

dlc_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

dlc_name

情報を求める DLC の名前、または DLC のリストの索引として使用する名前。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は 8 バイトからなる ASCII スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファ内の各エントリは、次のパラメーターで構成されています。

dlc_summary.overlay_size

戻された `dlc_summary` 構造体のサイズ。すなわち、データ・バッファ内で次のエントリが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 `dlc_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

dlc_summary.dlc_name

DLC 名。この名前は 8 バイトからなる ASCII ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

dlc_summary.description

DLC を記述したヌルで終了するテキスト・ストリング (その DLC の定義で指定したもの)。

dlc_summary.state

DLC の状態。次のいずれかです。

AP_ACTIVE

DLC は活動状態です。

AP_NOT_ACTIVE

DLC は活動状態ではありません。

AP_PENDING_INACTIVE

STOP_DLC が進行中です。

dlc_summary.dlc_type

DLC のタイプ。次のいずれかです。

AP_SDLC

SDLC

AP_X25 QLLC

AP_TR トークンリング

AP_ETHERNET

イーサネット

AP_MPC マルチパス・チャンネル (MPC) アダプター、System z 上の Communications Server for Linux のみ

AP_IP Enterprise Extender (HPR/IP)

dlc_detail.overlay_size

戻された `dlc_detail` 構造体のサイズ。すなわち、データ・バッファ内で次のエントリが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 `dlc_detail` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子

は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

dlc_detail.dlc_name

DLC 名。この名前は 8 バイトからなる ASCII ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

dlc_detail.dlc_data.state

DLC の状態。次のいずれかです。

AP_ACTIVE

DLC は活動状態です。

AP_NOT_ACTIVE

DLC は活動状態ではありません。

AP_PENDING_INACTIVE

STOP_DLC が進行中です。

dlc_detail.dlc_data.description

DLC を記述したヌルで終了するテキスト・ストリング (その DLC の定義で指定したもの)。

dlc_detail.dlc_data.initially_active

ノードの始動時に DLC を自動的に起動するかどうかを指定します。値は次のいずれかです。

AP_YES DLC はノードの始動時に自動的に起動します。

AP_NO DLC は自動的に起動しません。手動で起動させる必要があります。

dlc_detail.dlc_data.dlc_type

DLC のタイプ。次のいずれかです。

AP_SDLC

SDLC

AP_X25 QLLC

AP_TR トークンリング

AP_ETHERNET

イーサネット

AP_IP Enterprise Extender (HPR/IP)

dlc_detail.dlc_data.neg_ls_supp

DLC が折衝可能リンク・ステーションをサポートするかどうかを指定します。値は次のいずれかです。

AP_YES この DLC を使用するリンク・ステーションは、折衝可能にできます。

AP_NO この DLC を使用するリンク・ステーションは、1 次リンク・ステーションまたは 2 次リンク・ステーションのいずれかとして定義する必要があります。折衝可能リンク・ステーションはサポートされません。

dlc_detail.def_data.port_types

dlc_type を AP_TR、AP_ETHERNET、AP_IP のいずれかに設定した場合、このパラメーターは AP_PORT_SATF に設定されます。それ以外の DLC タイプの場合、このパラメーターは予約済みです。

dlc_detail.def_data.hpr_only

DLC を Enterprise Extender のリンクに使用するかどうかを指定します。その結果、HPR トラフィックのみをサポートします。値は次のいずれかです。

AP_YES この DLC を Enterprise Extender のリンクに使用し、HPR トラフィックのみをサポートします。

AP_NO この DLC を Enterprise Extender のリンクに使用せず、非 HPR トラフィックをサポートします。HPR トラフィックをサポートする場合もあります。

dlc_detail.def_data.dlc_spec_data_len

このタイプの DLC に固有のデータでスペースが入っていない長さ (バイト単位)。このデータのデータ構造体は *def_data* 構造体に従いますが、スペースが入るため、4 バイト目の境界から始まります。DLC 固有のデータについて詳しくは、100 ページの『DEFINE_DLC』を参照してください。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_DLC_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*dlc_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DLC_TRACE

QUERY_DLC_TRACE は、ADD_DLC_TRACE *verb* を使用してセットアップされた DLC 回線トレースに関する情報を戻します。

この verb は、使用するオプションに応じて、すべてのリソースのトレース、特定の
リソース・タイプのトレース、または特定のリソースのトレースに関する情報を入
手するために使用できます。

VCB 構造体

```
typedef struct query_dlc_trace
{
    AP_UINT16      opcode;           /* Verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* Primary return code      */
    AP_UINT32      secondary_rc;    /* Secondary return code    */
    unsigned char  *buf_ptr;        /* pointer to buffer        */
    AP_UINT32      buf_size;        /* buffer size              */
    AP_UINT32      total_buf_size;  /* total buffer size required */
    AP_UINT16      num_entries;     /* number of entries        */
    AP_UINT16      total_num_entries; /* total number of entries  */
    unsigned char  list_options;    /* listing options          */
    unsigned char  list_type;       /* type of listing required  */
    DLC_TRACE_FILTER filter_entry;  /* resource to start at     */
} QUERY_DLC_TRACE;

typedef struct dlc_trace_data
{
    AP_UINT16      overlay_size;    /* size of returned entry   */
    DLC_TRACE_FILTER filter;        /* DLC trace filter information */
} DLC_TRACE_DATA;

typedef struct dlc_trace_filter
{
    unsigned char  resource_type;   /* type of resource         */
    unsigned char  resource_name[8]; /* name of resource         */
    SNA_LFSID      lfsid;           /* session identifier       */
    unsigned char  message_type;   /* type of messages        */
} DLC_TRACE_FILTER;

typedef struct sna_lfsid
{
    union
    {
        AP_UINT16      session_id;
        struct
        {
            unsigned char  sidh;
            unsigned char  sidl;
        } s;
    } uu;
    AP_UINT16      odai;
} SNA_LFSID;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DLC_TRACE

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

DLC_TRACE エントリーの最大数 (そのエントリーのデータが戻されます)。ある範囲でなく、特定のエントリーに関するデータを要求するには、

QUERY_DLC_TRACE

値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すための DLC_TRACE エントリー・リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

`filter_entry` 構造体で指定したエントリーから開始します。

AP_LIST_FROM_NEXT

`filter_entry` 構造体で指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入力する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

list_type

トレース・オプションを表示するリソースのタイプ。値は次のいずれかです。

AP_ALL_DLC_TRACES

指定したトレース・オプションを (どのリソース・タイプについても) すべてリストします。

AP_ALL_RESOURCES

すべてのリソースについて指定したトレース・オプション (リソース・タイプ `AP_ALL_RESOURCES` を指定した `ADD_DLC_TRACE` を使用して定義したもの) をリストします。

AP_DLC DLC リソースについてのトレース・オプションをリストします。

AP_PORT

すべての LS がトレースされるポート・リソースについてのトレース・オプションをリストします。

AP_LS LS リソースについてのトレース・オプションをリストします。

AP_RTP_RESOURCE_TYPE

RTP 接続リソースについてのトレース・オプションをリストします。

AP_PORT_DEFINED_LS

定義された LS のみがトレースされる (暗黙 LS はトレースされない) ポート・リソースについてのトレース・オプションをリストします。

AP_PORT_IMPLICIT_LS

暗黙 LS のみがトレースされる (定義された LS はトレースされない) ポート・リソースについてのトレース・オプションをリストします。

filter_entry.resource_type

戻されるエンタリーまたはリストの索引として使用されるエンタリーのリソース・タイプを指定します。このパラメーターは、*list_type* を AP_ALL_DLC_TRACES に設定し、*list_options* を AP_FIRST_IN_LIST に設定しない場合のみ使用します。値は次のいずれかです。

AP_ALL_RESOURCES

すべての DLC、ポート、LS のトレースに使用するオプションを指定する必須エンタリー。

AP_DLC *resource_name* で指定した DLC と、この DLC を使用するすべてのポートおよび LS についてのトレース・オプションを指定する必須エンタリー。

AP_PORT

resource_name で指定したポートと、このポートを使用するすべての LS についてのトレース・オプションを指定する必須エンタリー。

AP_LS *resource_name* で指定した LS についてのトレース・オプションを指定する必須エンタリー。

AP_RTP_RESOURCE_TYPE

resource_name パラメーターで指定した RTP 接続についてのトレース・オプションを指定する必須エンタリー。

AP_PORT_DEFINED_LS

resource_name で指定したポートと、このポートを使用する定義されたすべての LS (暗黙 LS ではない) についてのトレース・オプションを指定する必須エンタリー。

AP_PORT_IMPLICIT_LS

resource_name で指定したポートと、このポートを使用するすべての暗黙 LS (定義された LS ではない) についてのトレース・オプションを指定する必須エンタリー。

filter_entry.resource_name

戻されるエンタリーの名前、またはリストの索引として使用されるエンタリーの名前。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合、または *resource_type* を AP_ALL_RESOURCES に設定した場合には無視されます。

filter_entry.lfsid

指定した LS 上のセッションのローカル・フォーム・セッション ID。これは *resource_type* AP_LS の場合のみ有効で、必須エンタリーが、指定した LS の特定のセッションについてのメッセージを指定していることを示します。この構造体には次の 3 つの値が入っており、これらの値は QUERY_SESSION verb の SESSION_STATS セクションに戻されます。

filter_entry.lfsid.uu.s.sidh

セッション ID の上位バイト。

filter_entry.lfsid.uu.s.sidl

セッション ID の下位バイト。

filter_entry.lfsid.odai

起点宛先アサイナー・インディケーター。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーには、次のパラメーターが入ります。

overlay_size

戻された *dlc_trace_data* 構造体のサイズ。すなわち、データ・バッファー内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *dlc_trace_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

dlc_trace_filter.resource_type

トレースされるリソースのタイプ。値は次のいずれかです。

ALL_RESOURCES

すべてのリソースのトレース・オプションを指定するエントリー。

AP_DLC *resource_name* で指定した DLC と、この DLC を使用するすべてのポートおよび LS についてのトレース・オプションを指定するエントリー。

AP_PORT

resource_name で指定したポートと、このポートを使用するすべての LS についてのトレース・オプションを指定するエントリー。

AP_LS *resource_name* で指定した LS (またはこの LS の特定の LFSID) についてのトレース・オプションを指定するエントリー。

AP_RTP_RESOURCE_TYPE

resource_name で名前を指定した RTP 接続のトレース・オプションを指定するエントリー。

AP_PORT_DEFINED_LS

resource_name で指定したポートと、このポートを使用する定義されたすべての LS (暗黙 LS ではない) についてのトレース・オプションを指定するエントリー。

AP_PORT_IMPLICIT_LS

resource_name で指定したポートと、このポートを使用するすべての暗黙 LS (定義された LS ではない) についてのトレース・オプションを指定するエントリー。

dlc_trace_filter.resource_name

トレースする DLC、ポート、または LS の名前。

dlc_trace_filter.lfsid

指定した LS 上のセッションのローカル・フォーム・セッション ID。これは *resource_type* AP_LS の場合のみ有効で、このセッションについてのメッセージのみをトレースすることを示します。この構造体には次の 3 つの値が入っており、これらの値は QUERY_SESSION verb の SESSION_STATS セクションに戻されます。

dlc_trace_filter.lfsid.uu.s.sidh

セッション ID の上位バイト。

dlc_trace_filter.lfsid.uu.s.sidl

セッション ID の下位バイト。

dlc_trace_filter.lfsid.odai

起点宛先アサイナー・インディケータ。

dlc_trace_filter.message_type

特定のリソースまたはセッションについてトレースされるメッセージのタイプ。このパラメーターは、AP_TRACE_ALL (すべてのメッセージのトレース) に設定するか、次の値の 1 つ以上 (論理 OR を使用して組み合わせる) に設定します。

AP_TRACE_XID

XID メッセージ

AP_TRACE_SC

セッション制御 RU

AP_TRACE_DFC

データ・フロー制御 RU

AP_TRACE_FMD

FMD メッセージ

AP_TRACE_NLP

(このメッセージ・タイプは現在は使用されていません)

AP_TRACE_NC

(このメッセージ・タイプは現在は使用されていません)

QUERY_DLC_TRACE

AP_TRACE_SEGS

RH を含まない BBIU 以外のセグメント

AP_TRACE_CTL

MU と XID 以外のメッセージ

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のいずれかを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LIST_TYPE

list_type パラメーターに、無効な値が指定されました。

AP_INVALID_RESOURCE_TYPE

resource_type パラメーターに、無効な値が指定されました。

AP_ALL_RESOURCES_NOT_DEFINED

resource_type パラメーターを AP_ALL_RESOURCES に設定しましたが、すべてのリソースのトレース・オプションに対して定義された DLC_TRACE エントリーがありません。

AP_INVALID_RTP_CONNECTION

resource_name パラメーターで指定した RTP 接続には、トレース・オプションが設定されていません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DLUR_DEFAULTS

QUERY_DLUR_DEFAULTS `verb` は、ユーザーが DEFINE_DLUR_DEFAULTS `verb` を使用して定義されたデフォルトを照会できるようにします。

VCB 構造体

```
typedef struct query_dlur_defaults
{
    AP_UINT16      opcode;           /* verb operation code */
    unsigned char  reserv2;         /* reserved */
    unsigned char  format;          /* reserved */
    AP_UINT16      primary_rc;      /* primary return code */
    AP_UINT32      secondary_rc;    /* secondary return code */
    unsigned char  description[32]; /* resource description */
    unsigned char  reserv1[16];     /* reserved */
    unsigned char  dlus_name[17];   /* DLUS name */
    unsigned char  bkup_dlus_name[17]; /* Backup DLUS name */
};
```

```

unsigned char    reserv3;                /* reserved          */
AP_UINT16       dlus_retry_timeout;     /* DLUS retry timeout */
AP_UINT16       dlus_retry_limit;       /* DLUS retry limit   */
unsigned char    prefer_active_dlus;     /* retry using active DLUS */
unsigned char    persistent_pipe_support; /* reserved          */
unsigned char    reserv4[14];           /* reserved          */
} QUERY_DLUR_DEFAULTS;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DLUR_DEFAULTS

description

リソースの記述。このパラメーターの長さは、4 バイトの倍数で、ゼロではありません。

dlus_name

デフォルトの DLUS ノード名。この名前は、すべてゼロに設定するか、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringにします。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

bkup_dlus_name

バックアップ・デフォルトとして使用する DLUS ノード名。この名前は、すべてゼロに設定するか、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringにします。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

dlus_retry_timeout

DLUS との接続の 2 回目以降の間隔 (秒数)。1 回目の試行と 2 回目の試行の間隔は常に 1 秒です。

dlus_retry_limit

最初の DLUS との接続が失敗したあとの再試行の最大数。0xFFFF の値は Communications Server for Linux が無期限に再試行することを示します。

prefer_active_dlus

Communications Server for Linux が、DLUS から否定 RSP(REQACTPU) を受信した場合の動作、または失敗した DLUR PU の再活動化を試行する場合の動作を指定します。値は次のいずれかです。

AP_YES デフォルトのプライマリー DLUS またはデフォルトのバックアップ DLUS が活動状態である場合、Communications Server for Linux は活動状態である DLUS のみを使用して PU を活動化または再活動化しようとします。

AP_NO Communications Server for Linux は、標準再試行ロジックを使用して PU を活動化または再活動化しようとします。

QUERY_DLUR_DEFAULTS

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
```

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードが DLUR をサポートしていません。これは、DEFINE_NODE verb の *dlur_support* パラメーターで定義されています。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DLUR_LU

QUERY_DLUR_LU は、Communications Server for Linux の DLUR 機能を使用している活動状態の LU に関する情報を戻します。この verb は、使用するオプションに応じて、特定の LU または複数の LU に関する情報を入手するために使用できます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_dlur_lu
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
    unsigned char  *buf_ptr;        /* pointer to buffer        */
    AP_UINT32      buf_size;        /* buffer size              */
    AP_UINT32      total_buf_size;  /* total buffer size required */
    AP_UINT16      num_entries;     /* number of entries        */
    AP_UINT16      total_num_entries; /* total number of entries  */
    unsigned char  list_options;    /* listing options          */
    unsigned char  reserv3;         /* reserved                  */
    unsigned char  lu_name[8];      /* LU name                  */
    unsigned char  pu_name[8];     /* PU name filter           */
    unsigned char  filter;          /* local / downstream filter */
} QUERY_DLUR_LU;
```

```

typedef struct dlur_lu_summary
{
    AP_UINT16      overlay_size;      /* size of returned entry */
    unsigned char  lu_name[8];       /* LU name */
} DLUR_LU_SUMMARY;

typedef struct dlur_lu_detail
{
    AP_UINT16      overlay_size;      /* size of returned entry */
    unsigned char  lu_name[8];       /* LU name */
    unsigned char  pu_name[8];       /* PU name of owning PU */
    unsigned char  dlus_name[17];    /* DLUS name if SSCP-LU session */
    unsigned char  active;           /* active */
    unsigned char  lu_location;     /* downstream or local LU */
    unsigned char  nau_address;     /* NAU address of LU */
    unsigned char  plu_name[17];    /* PLU name if PLU-SLU session */
    unsigned char  active;           /* active */
    unsigned char  reserv1[27];     /* reserved */
    unsigned char  rscv_len;        /* length of appended RSCV */
} DLUR_LU_DETAIL;

```

注: DLUR_LU_DETAIL 構造体のあとに、SNA フォーマットで定義された経路選択制御ベクトル (RSCV) が続く場合があります。この制御ベクトルは、ネットワークを介するセッション経路を定義し、BIND を実行します。この RSCV が組み込まれるのは、DLUR セッションに対して RSCV が保管されるようにノードの構成 (DEFINE_NODE で指定される) で指示され、PLU-SLU セッションが活動状態である場合のみです。

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DLUR_LU

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

DLUR LU の最大数 (その DLUR LU のデータが戻されます)。ある範囲でなく、特定の LU に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

pu_name パラメーターと *lu_name* パラメーターを組み合わせて指定したエントリーから開始します。

AP_LIST_FROM_NEXT

pu_name パラメーターと *lu_name* パラメーターを組み合わせて指定したエントリーの直後のエントリーから開始します。

リストは、最初に *pu_name*、次に *lu_name* の順番で並んでいます。アプリケーションでそのリストから特定のエントリーを入手する方法については、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

lu_name

情報を求める LU の名前、または LU のリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、8 バイトからなる タイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

pu_name

LU 情報を求める PU 名。特定の PU に関連した LU についての情報のみを表示するには、PU 名を指定します。すべての PU について完全なリストを取得するには、このフィールドをすべて 2 進ゼロに設定します。この名前は、8 バイトからなる タイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

filter

戻された LU をその位置に応じてフィルターにかけるかどうかを指定します。ネットワーク・ノードの場合は、次の値を指定できます。

AP_INTERNAL

内部 LU に関する情報のみを戻します。

AP_DOWNSTREAM

ダウンストリーム LU に関する情報のみを戻します。

AP_NONE

位置に関係なく、すべての LU に関する情報を戻します。

エンド・ノードの場合は、このパラメーターは予約済みです (ダウンストリーム DLUR LU はサポートされません)。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファ内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファ内の各エントリーは、次のパラメーターで構成されています。

dlur_lu_summary.overlay_size

戻された *dlur_lu_summary* 構造体のサイズ。すなわち、データ・バッファ内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 *dlur_lu_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

dlur_lu_summary.lu_name

LU の名前。この名前は、8 バイトからなる タイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

dlur_lu_detail.overlay_size

戻された *dlur_lu_detail* 構造体のサイズ。すなわち、データ・バッファ内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 *dlur_lu_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

dlur_lu_detail.lu_name

LU の名前。この名前は、8 バイトからなる タイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

dlur_lu_detail.pu_name

LU に関連した PU の名前。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC ストリングで、右側に EBCDIC のスペースを入れます。

dlur_lu_detail.dlus_name

SSCP-LU セッションが活動状態である場合は、このフィールドには LU が使用する DLUS ノードの名前が入ります。それ以外の場合は、17 桁の 2 進ゼロに設定されます。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

dlur_lu_detail.lu_location

LU の位置。

次のいずれかに設定します。

AP_INTERNAL

LU はローカル・ノードにあります。

AP_DOWNSTREAM

LU はダウンストリーム・ノードにあります (ネットワーク・ノードのみ)。

dlur_lu_detail.nau_address

LU のネットワーク・アクセス可能単位アドレス。

dlur_lu_detail.plu_name

PLU-SLU セッションが活動状態である場合は、このフィールドには PLU の名前が入ります。それ以外の場合は、17 桁の 2 進ゼロに設定されます。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

dlur_lu_detail.rscv_len

dlur_lu_detail 構造体に追加される RSCV の長さ。ノードの構成で DLUR RSCV の保管を指定していない場合、または PLU-SLU セッションが活動状態でない場合、この長さはゼロに設定され、RSCV は組み込まれません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*lu_name* パラメーターは無効でした。

AP_INVALID_FILTER_OPTION

filter パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードが DLUR をサポートしていません。これは、DEFINE_NODE verb の *dlur_support* パラメーターで定義されています。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DLUR_PU

QUERY_DLUR_PU は、Communications Server for Linux の DLUR 機能を使用している PU に関する情報を戻します。

この verb は、使用するオプションに応じて、特定の PU または複数の PU に関する情報を入手するために使用できます。

非活動ノードに対してこの verb を発行した場合は、そのローカル・ノードで定義されている PU に関する情報のみが戻されます。実行中のノードに対して発行した場合は、このノードで DLUR を使用している活動ダウンストリーム PU に関する情報も戻されます。

VCB 構造体

```
typedef struct query_dlur_pu
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                 */
    unsigned char  format;         /* reserved                 */
    AP_UINT16      primary_rc;     /* primary return code     */
    AP_UINT32      secondary_rc;   /* secondary return code   */
    unsigned char  *buf_ptr;       /* pointer to buffer       */
    AP_UINT32      buf_size;       /* buffer size             */
    AP_UINT32      total_buf_size; /* total buffer size required */
    AP_UINT16      num_entries;    /* number of entries       */
    AP_UINT16      total_num_entries; /* total number of entries */
    unsigned char  list_options;   /* listing options         */
    unsigned char  reserv3;       /* reserved                 */
}
```

QUERY_DLUR_PU

```
    unsigned char    pu_name[8];           /* PU name */
    unsigned char    dlus_name[17];       /* fully-qualified DLUS name */
    unsigned char    filter;              /* local / downstream filter */
} QUERY_DLUR_PU;

typedef struct dlur_pu_summary
{
    AP_UINT16        overlay_size;        /* size of returned entry */
    unsigned char    pu_name[8];         /* PU name */
    unsigned char    description[32];     /* resource description */
    unsigned char    reserv1[16];        /* reserved */
} DLUR_PU_SUMMARY;

typedef struct dlur_pu_detail
{
    AP_UINT16        overlay_size;        /* size of returned entry */
    unsigned char    pu_name[8];         /* PU name */
    unsigned char    description[32];     /* resource description */
    unsigned char    initially_active;    /* is the PU initially active? */
    unsigned char    reserv1[15];        /* reserved */
    unsigned char    defined_dlus_name[17]; /* defined DLUS name */
    unsigned char    bkup_dlus_name[17]; /* backup DLUS name */
    unsigned char    pu_id[4];           /* PU identifier */
    unsigned char    pu_location;        /* downstream or local PU */
    unsigned char    active_dlus_name[17]; /* active DLUS name */
    unsigned char    ans_support;        /* auto network shutdown support */
    unsigned char    pu_status;          /* status of the PU */
    unsigned char    dlus_session_status; /* status of the DLUS pipe */
    unsigned char    reserv3;            /* reserved */
    FQPCID           fqpcid;             /* FQPCID used on pipe */
    AP_UINT16        dlus_retry_timeout; /* DLUR retry timeout */
    AP_UINT16        dlus_retry_limit;   /* DLUR retry limit */
} DLUR_PU_DETAIL;

typedef struct fqpcid
{
    unsigned char    pcid[8];             /* procedure correlator identifier */
    unsigned char    fqcp_name[17];       /* originator's network qualified */
                                           /* CP name */
    unsigned char    reserve3[3];         /* reserved */
} FQPCID;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DLUR_PU

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

DLUR PU の最大数 (その DLUR PU のデータが戻されます)。ある範囲でなく、特定の PU に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

pu_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

pu_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストは *pu_name* で配列されます。アプリケーションでそのリストから特定のエントリーを入手する方法については、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

pu_name

情報を求める PU の名前、または PU のリストへの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、8 バイトからなる タイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

dlus_name

PU 情報を求める DLUS 名。特定の DLUS に関連した PU に関する情報のみを表示するには、DLUS 名を指定します。PU が表示されるのは、指定した DLUS ノードとの SSCP-PU セッションがある場合のみです。すべての DLUS について完全なリストを取得するには、このフィールドをすべて 2 進ゼロに設定します。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A スtring 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring 文字からなるネットワーク名で構成されます。

filter

戻された PU をその位置に応じてフィルターにかけるかどうかを指定します。

ネットワーク・ノードの場合は、次の値を指定できます。

AP_INTERNAL

内部 PU に関する情報のみを戻します。

AP_DOWNSTREAM

ダウンストリーム PU に関する情報のみを戻します。

AP_NONE

位置に関係なく、すべての PU に関する情報を戻します。

エンド・ノードの場合は、このパラメーターは予約済みです (ダウンストリーム DLUR PU はサポートされません)。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

dlur_pu_summary.overlay_size

戻された `dlur_pu_summary` 構造体のサイズ。すなわち、データ・バッファー内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファーを調べて、各 `dlur_pu_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

dlur_pu_summary.pu_name

PU の名前。この名前は、8 バイトからなる タイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

dlur_pu_summary.description

PU を記述したヌルで終了するテキスト・string (その PU の定義で指定したもの)。PU が定義された内部 PU ではなく、活動ダウンストリーム PU である場合、このパラメーターは予約済みです。

dlur_pu_detail.overlay_size

戻された `dlur_lu_detail` 構造体のサイズ。すなわち、データ・バッファ内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 `dlur_pu_detail` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

dlur_pu_detail.pu_name

PU の名前。この名前は、8 バイトからなる タイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

dlur_pu_detail.description

PU を記述したヌルで終了するテキスト・ストリング (その PU の定義で指定したもの)。PU が定義された内部 PU ではなく、活動ダウンストリーム PU である場合、このパラメーターは予約済みです。

dlur_pu_detail.initially_active

ノードの始動時に PU が自動的に起動するかどうかを指定します。ダウンストリーム PU の場合、このパラメーターは予約済みです。内部 PU の場合は次の値を使用できます。

AP_YES PU はノードの始動時に自動的に起動します。

AP_NO PU は自動的に起動しません。手動で起動させる必要があります。

dlur_pu_detail.defined_dlus_name

DEFINE_INTERNAL_PU verb または DEFINE_LS verb (`dspu_services` を AP_DLUR に設定して) で定義された DLUR ノードの名前。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

dlur_pu_detail.bkup_dlus_name

DEFINE_INTERNAL_PU verb または DEFINE_LS verb (`dspu_services` を AP_DLUR に設定して) で定義されたバックアップ DLUR ノードの名前。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

dlur_pu_detail.pu_id

DEFINE_INTERNAL_PU で定義された PU ID またはダウンストリーム PU からの XID で取得された PU ID。これは 4 バイトの 16 進数ストリングで、ブロック番号 (3 桁の 16 進数字) とノード番号 (5 桁の 16 進数字) で構成されます。

dlur_pu_detail.pu_location

PU の位置。

次のいずれかに設定します。

AP_INTERNAL

PU はローカル・ノードにあります。

AP_DOWNSTREAM

PU はダウンストリーム・ノードにあります (ネットワーク・ノードのみ)。

dlur_pu_detail.active_dlus_name

PU が現在使用している DLUS ノードの名前。SSCP-PU セッションが活動状態でない場合は、このフィールドのビットは、すべてゼロに設定されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringです。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

dlur_pu_detail.ans_support

SSCP-PU の活動化の際に DLUS から DLUR に送信された自動ネットワーク終了サポート。PU を制御する SSCP に対してサブエリア・ノードが自動ネットワーク終了プロシージャーを開始した場合に、リンク・レベルの接続を続けるかどうかを指定します。値は次のいずれかです。

AP_CONT

リンク・レベルの接続を継続します。

AP_STOP

リンク・レベルの接続を停止します。

SSCP-LU セッションが非活動状態である場合、このフィールドは予約済みです。

dlur_pu_detail.pu_status

PU の状況 (DLUR から見た)。値は次のいずれかです。

AP_RESET

PU はリセット状態です。

AP_PEND_ACTPU

PU はホストからの ACTPU を待ちます。

AP_PEND_ACTPU_RSP

DLUR は ACTPU を PU に転送したあと、PU の応答を待ちます。

AP_ACTIVE

PU は活動状態です。

AP_PEND_DACTPU_RSP

DLUR は DACTPU を PU に転送したあと、PU の応答を待ちます。

AP_PEND_INOP

DLUR は PU を非活動化する前に、必要なイベントがすべて完了するのを待ちます。

dlur_pu_detail.dlus_session_status

PU が現在使用している DLUS パイプの状況。値は次のいずれかです。

AP_PENDING_ACTIVE

パイプは活動化中です。

AP_ACTIVE

パイプは活動状態です。

AP_PENDING_INACTIVE

パイプは活動化中ではありません。

AP_INACTIVE

パイプは活動状態ではありません。

dlur_pu_detail.fqpcid.pcid

パイプで使用するプロシージャー相関関係子 ID。これは 8 バイトからなる 16 進数ストリングです。SSCP-PU セッションが活動状態でない場合、このフィールドはすべて 2 進ゼロに設定されます。

dlur_pu_detail.fqpcid.fqcp_name

パイプで使用する制御点の完全修飾名。SSCP-PU セッションが活動状態でない場合、このフィールドはすべて 2 進ゼロに設定されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

pcid パラメーターと *fqcp_name* パラメーターを組み合わせることによって、DLUR を使用して転送されるセッションに関与している各 PU が個別に識別されます。*fqcp_name* パラメーターは、DLUR ノードまたは DLUS ノードのいずれかの CP 名で、どのノードで SSCP-PU セッションの活動化が開始されたかで決まります。

dlur_pu_detail.dlus_retry_timeout

def_data.dlus_name パラメーターおよび *def_data.bkup_dlus_name* パラメーターで指定した、DLUS との接続の 2 回目以降の試行の間隔 (秒数)。1 回目の試行と 2 回目の試行の間隔は常に 1 秒です。ゼロを指定した場合は、DEFINE_DLUR_DEFAULTS verb で指定したデフォルト値が使用されます。

dlur_pu_detail.dlus_retry_limit

最初の DLUS との接続が失敗したあとの再試行回数。ゼロの値は、DEFINE_DLUR_DEFAULTS verb からの値を使用することを示します。0xFFFF が戻されると、Communications Server for Linux は無期限に再試行します。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

QUERY_DLUR_PU

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PU_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*pu_name* パラメーターは無効でした。

AP_INVALID_FILTER_OPTION

filter パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードが DLUR をサポートしていません。これは、DEFINE_NODE verb の *dlur_support* パラメーターで定義されています。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DLUS

QUERY_DLUS は、Communications Server for Linux の DLUR 機能が認識する DLUS ノードについての情報を戻します。この verb はパイプ統計 (SSCP-PU セッションおよび SSCP-LU セッションの統計) を戻します。PLU-SLU セッションの統計を取得するには、QUERY_ISR_SESSION verb を使用できます。

この verb は、使用するオプションに応じて、特定の DLUS または複数の DLUS に関する情報を入手するために使用できます。

非活動ノードに対してこの verb を発行した場合は、DEFINE_INTERNAL_PU または DEFINE_DLUR_DEFAULTS を使用して定義した DLUS ノードに関する情報のみが戻されます。実行中のノードに対して発行した場合は、活動状態である DLUS

ノードに関する情報も戻されます。 DEFINE_DLUR_DEFAULTS を使用して定義したバックアップ DLUS についての情報は、この DLUS が活動状態でなければ戻されません。

VCB 構造体

```
typedef struct query_dlus
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;              /* reserved                     */
    unsigned char  format;              /* reserved                     */
    AP_UINT16      primary_rc;          /* primary return code          */
    AP_UINT32      secondary_rc;        /* secondary return code        */
    unsigned char  *buf_ptr;            /* pointer to buffer            */
    AP_UINT32      buf_size;            /* buffer size                  */
    AP_UINT32      total_buf_size;      /* total buffer size required   */
    AP_UINT16      num_entries;         /* number of entries            */
    AP_UINT16      total_num_entries;   /* total number of entries      */
    unsigned char  list_options;        /* listing options              */
    unsigned char  reserv3;            /* reserved                     */
    unsigned char  dlus_name[17];       /* fully-qualified DLUS name    */
} QUERY_DLUS;

typedef struct dlus_data
{
    AP_UINT16      overlay_size;        /* size of returned entry       */
    unsigned char  dlus_name[17];       /* fully qualified DLUS name    */
    unsigned char  is_default;          /* is the DLUS the default      */
    unsigned char  is_backup_default;   /* is DLUS the backup default   */
    unsigned char  pipe_state;         /* state of CPSVRMGR pipe       */
    AP_UINT16      num_active_pus;      /* num of active PUs using pipe */
    PIPE_STATS     pipe_stats;         /* pipe statistics              */
    unsigned char  persistent_pipe_support; /* reserved                     */
    unsigned char  persistent_pipe;     /* reserved                     */
} DLUS_DATA;

typedef struct pipe_stats
{
    AP_UINT32      reqactpu_sent;        /* REQACTPUs sent to DLUS      */
    AP_UINT32      reqactpu_rsp_received; /* RSP(REQACTPU)s received     */
                                        /* from DLUS                   */
    AP_UINT32      actpu_received;      /* ACTPUs received from DLUS   */
    AP_UINT32      actpu_rsp_sent;      /* RSP(ACTPU)s sent to DLUS    */
    AP_UINT32      reqdactpu_sent;      /* REQDACTPUs sent to DLUS     */
    AP_UINT32      reqdactpu_rsp_received; /* RSP(REQDACTPU)s received    */
                                        /* from DLUS                   */
    AP_UINT32      dactpu_received;     /* DACTPUs received from DLUS  */
    AP_UINT32      dactpu_rsp_sent;     /* RSP(DACTPU)s sent to DLUS   */
    AP_UINT32      actlu_received;      /* ACTLUs received from DLUS   */
    AP_UINT32      actlu_rsp_sent;      /* RSP(ACTLU)s sent to DLUS    */
    AP_UINT32      dactlu_received;     /* DACTLUs received from DLUS  */
    AP_UINT32      dactlu_rsp_sent;     /* RSP(DACTLU)s sent to DLUS   */
    AP_UINT32      sscp_pu_mus_rcvd;    /* MUs for SSCP-PU sessions rcvd */
    AP_UINT32      sscp_pu_mus_sent;    /* MUs for SSCP-PU sessions sent */
    AP_UINT32      sscp_lu_mus_rcvd;    /* MUs for SSCP-LU sessions rcvd */
    AP_UINT32      sscp_lu_mus_sent;    /* MUs for SSCP-LU sessions sent */
} PIPE_STATS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DLUS

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

DLUS の最大数 (その DLUS のデータが戻されます)。ある範囲でなく、特定の DLUS に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

dus_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

dus_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストは *dus_name* の順序で配列されます。アプリケーションでそのリストから特定のエントリーを入手する方法については、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

dus_name

情報を求める DLUS の名前、または DLUS のリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A String 文字からなるネットワーク名で構成されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファ内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファ内の各エントリーは、次のパラメーターで構成されています。

dlus_data.overlay_size

戻された *dlus_data* 構造体のサイズ。すなわち、データ・バッファ内で次のエントリーが始まるまでのオフセット。

アプリケーションが戻されたバッファを調べて、各 *dlus_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

dlus_data.dlus_name

DLUS の名前。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A スtring 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring 文字からなるネットワーク名で構成されます。

dlus_data.is_default

DLUS ノードが `DEFINE_DLUR_DEFAULTS verb (AP_YES または AP_NO)` によりデフォルトとして指定されているかどうかを指定します。

dlus_data.is_backup_default

DLUS ノードが `DEFINE_DLUR_DEFAULTS verb (AP_YES または AP_NO)` によりバックアップ・デフォルトとして指定されているかどうかを指定します。

dlus_data.pipe_state

DLUS へのパイプの状態。値は次のいずれかです。

AP_PENDING_ACTIVE

パイプは活動化中です。

AP_ACTIVE

パイプは活動状態です。

AP_PENDING_INACTIVE

パイプは活動化中ではありません。

AP_INACTIVE

パイプは活動状態ではありません。

dlus_data.num_active_pus

DLUS へのパイプを現在使用している PU の数。

dlus_data.pipe_stats.reqactpu_sent

パイプを介して DLUS に送信された `REQACTPU` の数。

dlus_data.pipe_stats.reqactpu_rsp_received
 パイプを介して DLUS から受信された RSP(REQACTPU) の数。

dlus_data.pipe_stats.actpu_received
 パイプを介して DLUS から受信された ACTPU の数。

dlus_data.pipe_stats.actpu_rsp_sent
 パイプを介して DLUS に送信された RSP(ACTPU) の数。

dlus_data.pipe_stats.reqdactpu_sent
 パイプを介して DLUS に送信された REQDACTPU の数。

dlus_data.pipe_stats.reqdactpu_rsp_received
 パイプを介して DLUS から受信された RSP(REQDACTPU) の数。

dlus_data.pipe_stats.dactpu_received
 パイプを介して DLUS から受信された DACTPU の数。

dlus_data.pipe_stats.dactpu_rsp_sent
 パイプを介して DLUS に送信された RSP(DACTPU) の数。

dlus_data.pipe_stats.actlu_received
 パイプを介して DLUS から受信された ACTLU の数。

dlus_data.pipe_stats.actlu_rsp_sent
 パイプを介して DLUS に送信された RSP(ACTLU) の数。

dlus_data.pipe_stats.dactlu_received
 パイプを介して DLUS から受信された DACTLU の数。

dlus_data.pipe_stats.dactlu_rsp_sent
 パイプを介して DLUS に送信された RSP(DACTLU) の数。

dlus_data.pipe_stats.sscp_pu_mus_rcvd
 パイプを介して DLUS から受信された SSCP-PU MU の数。

dlus_data.pipe_stats.sscp_pu_mus_sent
 パイプを介して DLUS に送信された SSCP-PU MU の数。

dlus_data.pipe_stats.sscp_lu_mus_rcvd
 パイプを介して DLUS から受信された SSCP-LU MU の数。

dlus_data.pipe_stats.sscp_lu_mus_sent
 パイプを介して DLUS に送信された SSCP-LU MU の数。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
 AP_PARAMETER_CHECK

secondary_rc
 値は次のいずれかです。

AP_INVALID_DLUS_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*dlus_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻しません。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードが DLUR をサポートしていません。これは、DEFINE_NODE verb の *dlur_support* パラメーターで定義されています。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DOMAIN_CONFIG_FILE

QUERY_DOMAIN_CONFIG_FILE は、Communications Server for Linux ドメイン構成ファイルに含まれたヘッダー情報 (Communications Server for Linux バージョン番号、ファイルの改訂レベル、および DEFINE_DOMAIN_CONFIG_FILE で指定されたオプションのコメント・ストリング) を戻します。

この verb は、ドメイン構成ファイルに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_domain_config_file
{
    AP_UINT16          opcode;           /* verb operation code */
    unsigned char     reserv2;          /* reserved */
    unsigned char     format;           /* reserved */
    AP_UINT16         primary_rc;       /* primary return code */
    AP_UINT32         secondary_rc;     /* secondary return code */
    unsigned char     reserv3[8];       /* Reserved */
    CONFIG_FILE_HEADER hdr;
} QUERY_DOMAIN_CONFIG_FILE;

typedef struct config_file_header
{
    AP_UINT16         major_version;    /* major version number */
    AP_UINT16         minor_version;    /* minor version number */
    AP_UINT16         update_release;   /* update release */
    AP_UINT32         revision_level;   /* file revision number */
    unsigned char     comment[100];     /* optional comment string */
    AP_UINT16         updating;         /* reserved */
} CONFIG_FILE_HEADER;
```

QUERY_DOMAIN_CONFIG_FILE

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DOMAIN_CONFIG_FILE

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

hdr.major_version, *hdr.minor_version*, *hdr.update_release*

このファイルの作成に使用された Communications Server for Linux のリリースの内部バージョン ID。

hdr.revision_level

ファイルの改訂レベル (Communications Server for Linux により内部に保管されている)。

hdr.comment

ファイルに関する情報が入ったオプションのコメント・ストリング。これは 0 ~ 99 文字の ASCII ストリングで、最後にヌル文字が続きます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DOWNSTREAM_LU

QUERY_DOWNSTREAM_LU は、SNA ゲートウェイまたは DLUR、あるいはその両方が使用するダウンストリーム LU に関する情報を戻します。これは Communications Server for Linux プライマリー RUI アプリケーションと通信するアプリケーションによって使用されるダウンストリーム LU に関する情報も戻します。プライマリー RUI について詳しくは、「Communications Server for Linux LUA Programmer's Guide」を参照してください。

戻り情報は、決定済みデータ (実行中に動的に収集されたデータで、ノードが活動状態である場合のみ戻される) および定義済みデータ (DEFINE_DOWNSTREAM_LU で指定したデータ) として構造化されます。DLUR サポート LU の場合は、ダウンストリーム LU が活動化されると、暗黙に定義されたデータが書き込まれます。

この *verb* は、使用するオプションに応じて、特定の LU または複数の LU に関する情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_downstream_lu
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
}
```



```

AP_UINT32      secondary_rc;          /* secondary return code */
unsigned char  *buf_ptr;              /* pointer to buffer */
AP_UINT32      buf_size;              /* buffer size */
AP_UINT32      total_buf_size;        /* total buffer size required */
AP_UINT16      num_entries;           /* number of entries */
AP_UINT16      total_num_entries;     /* total number of entries */
unsigned char  list_options;          /* listing options */
unsigned char  reserv3;               /* reserved */
unsigned char  dspu_name[8];          /* Downstream PU name filter */
unsigned char  dslu_name[8];          /* Downstream LU name */
unsigned char  dspu_services;         /* services provided to LU */
} QUERY_DOWNSTREAM_LU;

typedef struct downstream_lu_summary
{
    AP_UINT16      overlay_size;        /* size of returned entry */
    unsigned char  dspu_name[8];        /* PU name */
    unsigned char  dslu_name[8];        /* LU name */
    unsigned char  description[32];     /* resource description */
    unsigned char  reserv1[16];         /* reserved */
    unsigned char  dspu_services;       /* Type of services provided
                                        /* to downstream LU
    unsigned char  nau_address;          /* NAU address */
    unsigned char  lu_sscp_sess_active; /* Is LU-SSCP session active */
    unsigned char  plu_sess_active;     /* Is PLU-SLU session active */
} DOWNSTREAM_LU_SUMMARY;

typedef struct downstream_lu_detail
{
    AP_UINT16      overlay_size;        /* size of returned entry */
    unsigned char  dslu_name[8];        /* LU name */
    unsigned char  reserv1[2];          /* reserved */
    DOWNSTREAM_LU_DET_DATA det_data;    /* Determined data */
    DOWNSTREAM_LU_DEF_DATA def_data;    /* Defined data */
} DOWNSTREAM_LU_DETAIL;

typedef struct downstream_lu_det_data
{
    unsigned char  lu_sscp_sess_active; /* Is LU-SSCP session active */
    unsigned char  plu_sess_active;     /* Is PLU-SLU session active */
    unsigned char  dspu_services;       /* Type of service provided to
                                        /* downstream node
    unsigned char  reserv1;              /* reserved */
    SESSION_STATS lu_sscp_stats;         /* LU-SSCP session statistics */
    SESSION_STATS ds_plu_stats;          /* Downstream PLU-SLU session
                                        /* statistics
    SESSION_STATS us_plu_stats;          /* Upstream PLU-SLU session
                                        /* statistics
    unsigned char  host_lu_name[8];      /* Determined host LU name */
    unsigned char  host_pu_name[8];      /* Determined host PU name */

    unsigned char  reserva[4];          /* reserved */
} DOWNSTREAM_LU_DET_DATA;

typedef struct downstream_lu_def_data
{
    unsigned char  description[32];     /* resource description */
    unsigned char  reserv1[16];         /* reserved */
    unsigned char  nau_address;         /* downstream LU nau address */
    unsigned char  dspu_name[8];        /* Downstream PU name */
    unsigned char  host_lu_name[8];     /* Host LU or Pool name */
    unsigned char  allow_timeout;       /* Allow timeout of host LU */
    unsigned char  delayed_logon;       /* Allow delayed logon to
                                        /* host LU
    unsigned char  reserv2[6];          /* reserved */
} DOWNSTREAM_LU_DEF_DATA;

```

QUERY_DOWNSTREAM_LU

```
typedef struct session_stats
{
    AP_UINT16    rcv_ru_size;          /* session receive RU size      */
    AP_UINT16    send_ru_size;        /* session send RU size         */
    AP_UINT16    max_send_btu_size;   /* maximum send BTU size        */
    AP_UINT16    max_rcv_btu_size;    /* maximum rcv BTU size         */
    AP_UINT16    max_send_pac_win;    /* maximum send pacing window size */
    AP_UINT16    cur_send_pac_win;    /* current send pacing window size */
    AP_UINT16    max_rcv_pac_win;    /* maximum receive pacing window size*/
    AP_UINT16    cur_rcv_pac_win;    /* current receive pacing window size*/
    AP_UINT32    send_data_frames;    /* number of data frames sent    */
    AP_UINT32    send_fmd_data_frames; /* num fmd data frames sent     */
    AP_UINT32    send_data_bytes;     /* number of data bytes sent     */
    AP_UINT32    rcv_data_frames;     /* number of data frames received */
    AP_UINT32    rcv_fmd_data_frames; /* num fmd data frames received  */
    AP_UINT32    rcv_data_bytes;     /* number of data bytes received  */
    unsigned char sidh;               /* session ID high byte (from LFSID) */
    unsigned char sidl;               /* session ID low byte (from LFSID) */
    unsigned char odai;               /* ODAI bit set                  */
    unsigned char ls_name[8];         /* Link station name              */
    unsigned char pacing_type;        /* type of pacing in use         */
} SESSION_STATS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DOWNSTREAM_LU

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

ダウンストリーム LU の最大数 (そのダウンストリーム LU のデータが戻されます)。ある範囲でなく、特定の LU に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

dspu_name パラメーターと *dslu_name* パラメーターを組み合わせて指定したエントリーから開始します。

AP_LIST_FROM_NEXT

dspu_name パラメーターと *dslu_name* パラメーターを組み合わせて指定したエントリーの直後のエントリーから開始します。

リストは、最初に *dspu_name*、次に *dslu_name* の順番で並んでいます。アプリケーションでそのリストから特定のエントリーを入手する方法については、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

dspu_name

LU 情報を求める PU 名 (DEFINE_LS で指定したもの)。特定の PU に関連した LU についての情報のみを表示するには、PU 名を指定します。すべての PU について完全なリストを取得するには、このフィールドをすべて 2 進ゼロに設定します。この名前は、8 バイトからなる タイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

dslu_name

情報を求める LU の名前、または LU のリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、8 バイトからなる タイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

dspu_services

DSPU サービス・フィルター。実行中のノードに対してこの verb を発行した場合、このパラメーターは、LU に指定したサービスのタイプによって戻り情報をフィルターにかけるかどうかを指定します。値は次のいずれかです。

AP_PU_CONCENTRATION

SNA ゲートウェイからサービスを受けるダウンストリーム LU のみに関する戻り情報。

AP_DLUR

DLUR からサービスを受けるダウンストリーム LU のみに関する戻り情報。

AP_NONE

すべてのダウンストリーム LU に関する戻り情報。

ノードが実行中でない場合、このパラメーターは無視されます。

Communications Server for Linux はすべてのダウンストリーム LU に関する情報を戻します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

QUERY_DOWNSTREAM_LU

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

downstream_lu_summary.overlay_size

戻された *downstream_lu_summary* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *downstream_lu_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

downstream_lu_summary.dspu_name

LU に関連した PU の名前。この名前は、8 バイトからなる タイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

downstream_lu_summary.dslu_name

LU の名前。この名前は、8 バイトからなる タイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

downstream_lu_summary.description

ダウンストリーム LU を記述したヌルで終了するテキスト・String (そのダウンストリーム LU の定義で指定したもの)。

DLUR サポート LU の場合、このパラメーターは予約済みです。

downstream_lu_summary.dspu_services

実行中のノードに対してこの *verb* が発行された場合、このパラメーターは、ローカル・ノードがダウンストリーム LU に提供するサービスを指定します。

値は次のいずれかです。

AP_PU_CONCENTRATION

ダウンストリーム LU は SNA ゲートウェイからサービスを受けません。

AP_DLUR

ダウンストリーム LU は DLUR からサービスを受けます。

downstream_lu_summary.nau_address

LU のネットワーク・アクセス可能単位アドレス。

downstream_lu_summary.lu_sscp_sess_active

LU-SSCP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

downstream_lu_summary.plu_sess_active

PLU-SLU セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

downstream_lu_detail.overlay_size

戻された *downstream_lu_detail* 構造体のサイズ。すなわち、データ・バッファにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *downstream_lu_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

downstream_lu_detail.dslu_name

LU の名前。この名前は、8 バイトからなる タイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

downstream_lu_detail.det_data.lu_sscp_sess_active

LU-SSCP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

downstream_lu_detail.det_data.plu_sess_active

PLU-SLU セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

QUERY_DOWNSTREAM_LU

downstream_lu_detail.det_data.dspu_services

実行中のノードに対してこの `verb` が発行された場合、このパラメーターは、ローカル・ノードがダウンストリーム LU に提供するサービスを指定します。

値は次のいずれかです。

AP_PU_CONCENTRATION

ダウンストリーム LU は SNA ゲートウェイからサービスを受けます。

AP_DLUR

ダウンストリーム LU は DLUR からサービスを受けます。

3 つのセッション (LU-SSCP セッション、ダウンストリーム PLU-SLU セッション、アップストリーム PLU-SLU セッション) のそれぞれに `session_stats` 構造体が組み込まれます。この構造体のフィールドは次のとおりです。

rcv_ru_size

受信 RU の最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

send_ru_size

送信 RU の最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

max_send_btu_size

送信できる BTU の最大サイズ。

max_rcv_btu_size

受信できる BTU の最大サイズ。

max_send_pac_win

このセッションの送信ペーシング・ウィンドウの最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

cur_send_pac_win

このセッションの送信ペーシング・ウィンドウの現行サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

max_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

cur_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの現行サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

send_data_frames

送信された通常フロー・データ・フレームの数。

send_fmd_data_frames

送信された通常フロー FMD データ・フレームの数。

send_data_bytes

送信された通常フロー・データ・バイトの数。

rcv_data_frames

受信された通常フロー・データ・フレームの数。

rcv_fmd_data_frames

受信された通常フロー FMD データ・フレームの数。

rcv_data_bytes

受信された通常フロー・データ・バイトの数。

sidh セッション ID の上位バイト。(SNA ゲートウェイからサービスを受ける LU のアップストリーム PLU-SLU セッション統計では、このパラメーターは予約済みです。)

sidl セッション ID の下位バイト。(SNA ゲートウェイからサービスを受ける LU のアップストリーム PLU-SLU セッション統計では、このパラメーターは予約済みです。)

odai 起点宛先アサイナー・インディケータ。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合、BIND の送信側は、このフィールドをゼロに設定し、BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合は、このフィールドを 1 に設定します。(SNA ゲートウェイからサービスを受ける LU のアップストリーム PLU-SLU セッション統計では、このパラメーターは予約済みです。)

ls_name

統計に関連したリンク・ステーション名。この名前は、8 バイトからなる ASCII 文字ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。(SNA ゲートウェイからサービスを受ける LU のアップストリーム PLU-SLU セッション統計では、このパラメーターは予約済みです。)

pacings_type

このセッションで使用している受信ペースングのタイプ。値は次のいずれかです。

AP_NONE

AP_PACING_FIXED

downstream_lu_detail.det_data.host_lu_name

ダウンストリーム LU がマップされるホスト LU の名前、または PLU-SLU セッションが前に活動状態であったときにダウンストリーム LU がマップされていたホスト LU の名前。*def_data.host_lu_name* はホスト LU プールの名前であることがあるため、このパラメーター値は *def_data.host_lu_name* と異なる場合があります。

ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は、このフィールドを EBCDIC でストリング #PRIRUI# と設定します。

downstream_lu_detail.det_data.host_pu_name

ダウンストリーム LU がマップされるホスト PU の名前、または PLU-SLU セッションが前に活動状態であったときにダウンストリーム LU がマップされていたホスト PU の名前。

downstream_lu_detail.def_data.description

ダウンストリーム LU を記述したヌルで終了するテキスト・ストリング (そ

QUERY_DOWNSTREAM_LU

のダウンストリーム LU の定義で指定したもの)。DLUR サポート LU の場合、このパラメーターは予約済みです。

downstream_lu_detail.def_data.nau_address

ダウンストリーム LU のネットワーク・アクセス可能単位アドレス。

downstream_lu_detail.def_data.dspu_name

この LU に関連したダウンストリーム PU の名前 (DEFINE_LS verb で指定したもの)。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

downstream_lu_detail.def_data.host_lu_name

ダウンストリーム LU が使用するホスト LU またはホスト LU プールの名前。この名前は、8 バイトからなる EBCDIC スtringで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は、このフィールドを EBCDIC でスtring #PRIRUI# と設定します。

このフィールドは DLUR からサービスを受けるダウンストリーム LU に予約済みです。

downstream_lu_detail.allow_timeout

このダウンストリーム LU でアップストリーム LU とのセッションのタイムアウトが許可されるかどうかを指定します。値は次のいずれかです。

AP_YES このダウンストリーム LU では、アップストリーム LU とのセッションのタイムアウトが許可されます。

AP_NO このダウンストリーム LU では、アップストリーム LU とのセッションのタイムアウトは許可されません。

このフィールドは、ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は無視されます。

downstream_lu_detail.delayed_logon

このダウンストリーム LU で遅延ログオン (ユーザーが活動化を要求するまでアップストリーム LU は活動化されない) を使用するかどうかを指定します。値は次のいずれかです。

AP_YES このダウンストリーム LU では、遅延ログオンを使用します。

AP_NO このダウンストリーム LU では、遅延ログオンは使用しません。

このフィールドは、ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は無視されます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

list_options パラメーターが、指定した名前から始まるすべてのエン
トリーをリストするために、AP_LIST_INCLUSIVE に設定されていま
すが、*lu_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんで
した。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、
AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、
以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

AP_INVALID_PU_TYPE

dspu_name パラメーターで指定した PU はダウンストリーム PU
ではありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、
AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実
行されないと、Communications Server for Linux は、以下のパラメーターを戻しま
す。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードが SNA ゲートウェイまたは DLUR をサポート
していません。これは DEFINE_NODE verb の *pu_conc_support* パ
ラメーターおよび *dlur_support* パラメーターで定義されています。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記
以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DOWNSTREAM_PU

QUERY_DOWNSTREAM_PU は、SNA ゲートウェイまたは DLUR、あるいはその両方が使用するダウンストリーム PU に関する情報を戻します。この verb は、使用するオプションに応じて、特定の PU または複数の PU に関する情報を入手するために使用できます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_downstream_pu
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;          /* reserved                      */
    unsigned char  format;           /* reserved                      */
    AP_UINT16      primary_rc;       /* primary return code          */
    AP_UINT32      secondary_rc;     /* secondary return code        */
    unsigned char  *buf_ptr;         /* pointer to buffer            */
    AP_UINT32      buf_size;         /* buffer size                  */
    AP_UINT32      total_buf_size;   /* total buffer size required   */
    AP_UINT16      num_entries;      /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries      */
    unsigned char  list_options;     /* listing options              */
    unsigned char  reserv3;          /* reserved                      */
    unsigned char  dspu_name[8];     /* Downstream PU name filter    */
    unsigned char  dspu_services;    /* services provided to PU      */
} QUERY_DOWNSTREAM_PU;

typedef struct downstream_pu_data
{
    AP_UINT16      overlay_size;     /* size of returned entry       */
    unsigned char  dspu_name[8];     /* PU name                      */
    unsigned char  description[32];  /* resource description          */
    unsigned char  reserv1[16];     /* reserved                      */
    unsigned char  ls_name[8];       /* Link name                    */
    unsigned char  pu_sscp_sess_active; /* Is the PU-SSCP session active */
    unsigned char  dspu_services;    /* DSPU service type            */
    SESSION_STATS pu_sscp_stats;     /* SSCP-PU session statistics   */
    unsigned char  reserva[20];     /* reserved                      */
} DOWNSTREAM_PU_DATA;

typedef struct session_stats
{
    AP_UINT16      rcv_ru_size;      /* session receive RU size      */
    AP_UINT16      send_ru_size;     /* session send RU size         */
    AP_UINT16      max_send_btu_size; /* maximum send BTU size        */
    AP_UINT16      max_rcv_btu_size; /* maximum rcv BTU size         */
    AP_UINT16      max_send_pac_win; /* maximum send pacing window size */
    AP_UINT16      cur_send_pac_win; /* current send pacing window size */
    AP_UINT16      max_rcv_pac_win;  /* maximum receive pacing window */
    /* size */
    AP_UINT16      cur_rcv_pac_win;  /* current receive pacing window */
    /* size */
    AP_UINT32      send_data_frames; /* number of data frames sent   */
    AP_UINT32      send_fmd_data_frames; /* num fmd data frames sent   */
    AP_UINT32      send_data_bytes;  /* number of data bytes sent    */
    AP_UINT32      rcv_data_frames;  /* number of data frames received */
    AP_UINT32      rcv_fmd_data_frames; /* num fmd data frames received */
    AP_UINT32      rcv_data_bytes;   /* number of data bytes received */
    unsigned char  sidh;             /* session ID high byte (from LFSID) */
    unsigned char  sidl;             /* session ID low byte (from LFSID) */
}
```

```

unsigned char  odai;                /* ODAI bit set          */
unsigned char  ls_name[8];          /* Link station name     */
unsigned char  pacing_type;        /* type of pacing in use */
} SESSION_STATS;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DOWNSTREAM_PU

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size
指定したデータ・バッファのサイズ。

num_entries
ダウンストリーム PU の最大数 (そのダウンストリーム PU のデータが戻されます)。ある範囲でなく、特定の PU に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options
Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST
リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE
dspu_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT
dspu_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

dspu_name
情報を求める PU の名前 (DEFINE_LS で定義されたもの)、または PU のリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、8 バイトからなる タイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

dspu_services
DSPU サービス・フィルター。PU に提供されたサービスのタイプによって戻り情報をフィルターにかけるかどうかを指定します。値は次のいずれかです。

AP_PU_CONCENTRATION
SNA ゲートウェイからサービスを受けるダウンストリーム PU のみに関する戻り情報。

AP_DLUR

DLUR からサービスを受けるダウンストリーム PU のみに関する戻り情報。

AP_NONE

すべてのダウンストリーム PU に関する戻り情報。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

downstream_pu_data.overlay_size

戻された *downstream_pu_data* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *downstream_pu_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

downstream_pu_data.dspu_name

ダウンストリーム PU の名前。この名前は、8 バイトからなる タイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

downstream_pu_data.description

ダウンストリーム PU に LS を記述したヌルで終了するテキスト・String (その LU の定義したもの)。

downstream_pu_data.ls_name

ダウンストリーム PU へのアクセスに使用した LS の名前。これは 8 バイトからなる ASCII スtringで、名前が 8 文字に満たない場合、右側にスペースを入れます。

downstream_pu_data.pu_sscp_sess_active

ダウンストリーム PU との PU-SSCP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

downstream_pu_data.dspu_services

PU に提供されるサービスのタイプを指定します。

値は次のいずれかです。

AP_PU_CONCENTRATION

ダウンストリーム PU は SNA ゲートウェイからサービスを受けます。

AP_DLUR

ダウンストリーム PU は DLUR からサービスを受けます。

downstream_pu_data.pu_sscp_stats.rcv_ru_size

受信 RU の最大サイズ。ダウンストリーム PU が SNA ゲートウェイのサービスを受ける場合、このフィールドは予約済みです (ゼロに設定されません)。

downstream_pu_data.pu_sscp_stats.send_ru_size

送信 RU の最大サイズ。ダウンストリーム PU が SNA ゲートウェイのサービスを受ける場合、このフィールドは予約済みです (ゼロに設定されません)。

downstream_pu_data.pu_sscp_stats.max_send_btu_size

送信できる BTU の最大サイズ。

downstream_pu_data.pu_sscp_stats.max_rcv_btu_size

受信できる BTU の最大サイズ。

downstream_pu_data.pu_sscp_stats.max_send_pac_win

予約済み (常にゼロに設定される)。

downstream_pu_data.pu_sscp_stats.cur_send_pac_win

予約済み (常にゼロに設定される)。

downstream_pu_data.pu_sscp_stats.max_rcv_pac_win

予約済み (常にゼロに設定される)。

downstream_pu_data.pu_sscp_stats.cur_rcv_pac_win

予約済み (常にゼロに設定される)。

downstream_pu_data.pu_sscp_stats.send_data_frames

送信された通常フロー・データ・フレームの数。

downstream_pu_data.pu_sscp_stats.send_fmd_data_frames

送信された通常フロー FMD データ・フレームの数。

QUERY_DOWNSTREAM_PU

downstream_pu_data.pu_sscp_stats.send_data_bytes

送信された通常フロー・データ・バイトの数。

downstream_pu_data.pu_sscp_stats.rcv_data_frames

受信された通常フロー・データ・フレームの数。

downstream_pu_data.pu_sscp_stats.rcv_fmd_data_frames

受信された通常フロー FMD データ・フレームの数。

downstream_pu_data.pu_sscp_stats.rcv_data_bytes

受信された通常フロー・データ・バイトの数。

downstream_pu_data.pu_sscp_stats.sidh

セッション ID の上位バイト。

downstream_pu_data.pu_sscp_stats.sidl

セッション ID の下位バイト。

downstream_pu_data.pu_sscp_stats.odai

起点宛先アサイナー・インディケータ。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合、BIND の送信側は、このフィールドをゼロに設定し、BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合は、このフィールドを 1 に設定します。

downstream_pu_data.pu_sscp_stats.ls_name

統計に関連したリンク・ステーション名。この名前は、8 バイトからなる ASCII 文字ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

downstream_pu_data.pacing_type

PU-SSCP で使用している受信ペーシングのタイプ。このパラメーターは常に AP_NONE に設定されます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PU_NAME

list_options パラメーターは、指定した名前から始まるすべてのエントリーを表示する AP_LIST_INCLUSIVE に設定されましたが、

dspu_name パラメーターが有効ではありませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは SNA ゲートウェイまたは DLUR をサポートしていません。これは DEFINE_NODE verb の *pu_conc_support* パラメーターおよび *dlur_support* パラメーターで定義されています。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_DSPU_TEMPLATE

QUERY_DSPU_TEMPLATE verb は、暗黙的なリンクを介して SNA ゲートウェイに使用する定義されたダウンストリーム PU テンプレートに関する情報を戻します。

この verb は、使用するオプションに応じて、特定のダウンストリーム PU テンプレートまたは複数のダウンストリーム PU テンプレートに関する情報を入手するために使用できます。特定のダウンストリーム PU テンプレートまたは複数のダウンストリーム PU テンプレートに関する情報を入手するには、*template_name* パラメーターを設定します。*list_options* オプションを AP_FIRST_IN_LIST に設定した場合、*template_name* パラメーターは無視されます。

VCB 構造体

```
typedef struct query_dspu_template
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv1;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  *buf_ptr;        /* pointer to buffer            */
    AP_UINT32      buf_size;        /* buffer size                  */
    AP_UINT32      total_buf_size;  /* total buffer size required    */
    AP_UINT16      num_entries;     /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries      */
    unsigned char  list_options;    /* listing options              */
    unsigned char  reserv3;         /* reserved                      */
    unsigned char  template_name[8]; /* name of DSPU template        */
} QUERY_DSPU_TEMPLATE;

typedef struct dspu_template_data
{
    AP_UINT16      overlay_size;    /* size of returned entry       */
    unsigned char  template_name[8]; /* name of DSPU template        */
    unsigned char  description[32]; /* resource description          */
    unsigned char  reserv2[16];     /* reserved                      */
    unsigned char  reserv1[12];     /* reserved                      */
}
```

QUERY_DSPU_TEMPLATE

```
    AP_UINT16      max_instance;    /* max active template instance */
    AP_UINT16      active_instance; /* current active instances */
    unsigned char  num_of_dslu_templates; /* number of DSLU templates */
} DSPU_TEMPLATE_DATA;
```

各 `dspu_template_data` 構造体には 1 つ以上のダウンストリーム LU テンプレートが続きます。ダウンストリーム LU テンプレートの数は `number_of_dslu_templates` パラメーターで指定します。各ダウンストリーム LU テンプレートのフォーマットは次のとおりです。

```
typedef struct dslu_template_data
{
    AP_UINT16      overlay_size;    /* size of this entry */
    unsigned char  reserv1[2];     /* reserved */
    DSLU_TEMPLATE  dslu_template;  /* downstream LU template */
} DSLU_TEMPLATE_DATA;

typedef struct dslu_template
{
    unsigned char  min_nau;        /* minimum NAU address in range */
    unsigned char  max_nau;        /* maximum NAU address in range */
    unsigned char  allow_timeout; /* allow timeout of host LU? */
    unsigned char  delayed_logon; /* allow delayed logon to host LU */
    unsigned char  reserv1[8];    /* reserved */
    unsigned char  host_lu[8];    /* host LU or pool name */
} DSLU_TEMPLATE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_DSPU_TEMPLATE

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

テンプレートの最大数 (そのテンプレートのデータが戻されます)。ある範囲でなく、特定のテンプレートに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

template_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

template_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

template_name

情報を求める DSPU の名前、または DSPU のリストの索引として使用する名前。これは 8 バイトからなるローカル・システム上で表示可能な文字セットのストリングです。このパラメーターは、*list_options* を *AP_FIRST_IN_LIST* に設定した場合には無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

バッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。この値は *buf_size* より大きい場合があります。

num_entries

実際に戻されたエントリーの数。

total_num_entries

戻されている可能性があるエントリーの合計数。この値は *num_entries* より大きい場合があります。

dspu_template_data.overlay_size

このエントリーのバイト数。ダウンストリーム LU テンプレートを含み、次に戻されたエントリーがある場合は、そのエントリーへのオフセットも含みます。

アプリケーションが戻されたバッファーを調べて、各 *dspu_template_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の *sizeof()* 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

dspu_template_data.template_name

DSPU テンプレートの名前。

dspu_template_data.description

DEFINE_DSPU_TEMPLATE verb で定義したリソースの記述。

dspu_template_data.max_instance

同時に活動状態にできるテンプレートのインスタンスの最大数。

QUERY_DSPU_TEMPLATE

dspu_template_data.active_instance

現在活動状態であるテンプレートのインスタンスの数。

dspu_template_data.num_of_dslu_templates

このダウンストリーム PU テンプレートに対応するダウンストリーム LU テンプレートの数。このパラメーターに続いて、 DSLU テンプレートごとに *num_of_dslu_templates* エントリーが 1 つずつあります。

dslu_template_data.overlay_size

このエントリーのバイト数。次に戻されたエントリーがある場合は、そのエントリーへのオフセットも含まれます。

アプリケーションが戻されたバッファを調べて、各 *dslu_template_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

dslu_template_data.min_nau

DSLU テンプレートの範囲内の最小 NAU アドレス。

dslu_template_data.max_nau

DSLU テンプレートの範囲内の最大 NAU アドレス。

dslu_template_data.allow_timeout

ホスト LU 定義で指定されたタイムアウト期間にセッションが活動状態でないまま放置された場合、このダウンストリーム LU が使用しているホスト LU を Communications Server for Linux がタイムアウトにできるかどうかを指定します。値は次のいずれかです。

AP_YES Communications Server for Linux は、このダウンストリーム LU が使用しているホスト LU をタイムアウトにすることができます。

AP_NO Communications Server for Linux は、このダウンストリーム LU が使用しているホスト LU をタイムアウトにすることができません。

このフィールドは、ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は無視されます。

dslu_template_data.delayed_logon

Communications Server for Linux に、ダウンストリーム LU とホスト LU との接続をダウンストリーム LU から最初のデータを受信するまで遅らせるかどうかを指定します。接続を遅らせる場合は、シミュレート・ログオン画面がダウンストリーム LU へ送信されます。値は次のいずれかです。

AP_YES Communications Server for Linux はダウンストリーム LU のホスト LU への接続を遅らせます。

AP_NO Communications Server for Linux はダウンストリーム LU のホスト LU への接続を遅らせません。

このフィールドは、ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は無視されます。

dslu_template_data.host_lu_name

範囲内のすべてのダウンストリーム LU がマップされるホスト LU の名前またはホスト LU プールの名前。

ダウンストリーム LU が、ホストの代わりに、Communications Server for Linux プライマリー RUI アプリケーションと通信するために使用される場合は、このフィールドを EBCDIC でストリング #PRIRUI# と設定します。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_TEMPLATE_NAME

template_name パラメーターで指定したテンプレートが有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_FOCAL_POINT

QUERY_FOCAL_POINT は、使用するオプションに応じて、特定の Management Services カテゴリーのフォーカル・ポイントについての情報、または複数のフォーカル・ポイントに関する情報を戻します。

VCB 構造体

```
typedef struct query_focal_point
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;               /* reserved                      */
    unsigned char  format;               /* reserved                      */
    AP_UINT16      primary_rc;           /* primary return code          */
    AP_UINT32      secondary_rc;        /* secondary return code        */
    unsigned char  *buf_ptr;             /* pointer to buffer            */
    AP_UINT32      buf_size;             /* buffer size                  */
    AP_UINT32      total_buf_size;      /* total buffer size required   */
    AP_UINT16      num_entries;          /* number of entries            */
    AP_UINT16      total_num_entries;   /* total number of entries      */
    unsigned char  list_options;        /* listing options              */
    unsigned char  reserv3;             /* reserved                      */
    unsigned char  ms_category[8];      /* name of MS category          */
} QUERY_FOCAL_POINT;
```

QUERY_FOCAL_POINT

```
typedef struct fp_data
{
    AP_UINT16    overlay_size;           /* size of returned entry */
    unsigned char ms_appl_name[8];      /* focal point application name */
    unsigned char ms_category[8];      /* focal point category */
    unsigned char description[32];     /* resource description */
    unsigned char reserv1[16];         /* reserved */
    unsigned char fp_fqcp_name[17];    /* focal point fully qualified
                                        /* cp name */
    unsigned char bkup_appl_name[8];   /* backup focal point
                                        /* application name */
    unsigned char bkup_fp_fqcp_name[17]; /* backup fp fully qualified cp
                                        /* name */
    unsigned char implicit_appl_name[8]; /* implicit focal point appl name */
    unsigned char implicit_fp_fqcp_name[17]; /* implicit fp fully qualified
                                        /* cp name */
    unsigned char fp_type;             /* focal point type */
    unsigned char fp_status;           /* focal point status */
    unsigned char fp_routing;          /* type of MDS routing to use */
    unsigned char reserva[20];         /* reserved */
    AP_UINT16    number_of_appls;      /* number of applications */
} FP_DATA;
```

各 `fp_data` 構造体には 1 つ以上のアプリケーション名が続きます。アプリケーション名は `number_of_appls` パラメーターで指定します。各アプリケーション名のフォーマットは次のとおりです。

```
    unsigned char    appl_name[8];           /* application name */
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_FOCAL_POINT

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

エントリーの最大数 (そのエントリーのデータが戻されます)。ある範囲でなく、特定のエントリーに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのフォーカル・ポイント・リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

ms_category パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

ms_category パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

ms_category

Management Services カテゴリ。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には使用されません。

これは、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照)の『MS Discipline-Specific Application Programs』の表に指定されたカテゴリ名の 1 つ (EBCDIC スペース (0x40) が埋め込まれている) か、またはユーザー定義のカテゴリです。ユーザー定義カテゴリ名は、8 バイトからなるタイプ 1134 の EBCDIC ストリングで、必要に応じて EBCDIC のスペース (0x40) を入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを返します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて返すために必要だったバッファーのサイズを示す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

fp_data.overlay_size

戻された *fp_data* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *fp_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

fp_data.ms_appl_name

現在活動状態であるフォーカル・ポイント・アプリケーションの名前。これは、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの

ページの『関連資料』を参照) の『MS Discipline-Specific Application Programs』の 1 つか、タイプ 1134 の文字を使用した EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

fp_data.ms_category

Management Services カテゴリー。これは、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照) に指定されたカテゴリー名の 1 つか、タイプ 1134 の文字を使用した EBCDIC String で、その名前が 8 文字に満たない場合は、右側にスペースを入れます。

fp_data.description

フォーカル・ポイントを記述したヌルで終了するテキスト・String (フォーカル・ポイントの定義で指定したもの)。

fp_data.fp_fqcp_name

現在活動状態であるフォーカル・ポイントの制御点の完全修飾名。この名前は 17 バイトの EBCDIC String で、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String 文字からなるネットワーク名で構成されます。

fp_data.bkup_appl_name

バックアップ・フォーカル・ポイント・アプリケーション名。これは、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照) の『MS Discipline-Specific Application Programs』の 1 つか、タイプ 1134 の文字を使用した EBCDIC String で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

fp_data.bkup_fp_fqcp_name

バックアップ・フォーカル・ポイントの制御点の完全修飾名。この名前は 17 バイトの EBCDIC String で、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String 文字からなるネットワーク名で構成されます。

fp_data.implicit_appl_name

暗黙のフォーカル・ポイント・アプリケーションの名前 (DEFINE_FOCAL_POINT を使用して指定されたもの)。これは、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照) の『MS Discipline-Specific Application Programs』の 1 つか、タイプ 1134 の文字を使用した EBCDIC String で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

fp_data.implicit_fp_fqcp_name

暗黙のフォーカル・ポイントの制御点の完全修飾名 (DEFINE_FOCAL_POINT を使用して指定されたもの)。この名前は 17 バイトの EBCDIC String で、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String 文字からなるネットワーク名で構成されます。

fp_data.fp_type

フォーカル・ポイントのタイプ。詳しくは、「*IBM Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照)をご覧ください。次のいずれかです。

AP_EXPLICIT_PRIMARY_FP

AP_IMPLICIT_PRIMARY_FP

AP_BACKUP_FP

AP_DEFAULT_PRIMARY_FP

AP_DOMAIN_FP

AP_HOST_FP

AP_NO_FP

fp_data.fp_status

フォーカル・ポイントの状況。次のいずれかです。

AP_ACTIVE

フォーカル・ポイントは現在活動状態です。

AP_NOT_ACTIVE

フォーカル・ポイントは現在活動状態ではありません。

AP_PENDING

フォーカル・ポイントは保留活動状態です。これは暗黙の要求がフォーカル・ポイントに送信されたあと、応答が受信されるまでの状態です。

AP_NEVER_ACTIVE

指定されたカテゴリのアプリケーション登録は受け付けられましたが、このカテゴリに使用できるフォーカル・ポイント情報がありません。

fp_data.fp_routing

アプリケーションでフォーカル・ポイントへの経路トラフィックを指定するため、デフォルト経路指定または直接経路指定のどちらを使用するかを指定します。次のいずれかです。

AP_DEFAULT

MDS_MU は、デフォルト経路指定を使用してフォーカル・ポイントへ送達されます。

AP_DIRECT

MDS_MU は、セッションで直接フォーカル・ポイントへ経路指定されます。

fp_data.number_of_appls

このフォーカル・ポイント・カテゴリに登録されたアプリケーションの数。

appl_name

フォーカル・ポイント・カテゴリに登録されたアプリケーションの名前。これは、「*Systems Network Architecture: Management Services Reference*」(xxiii ページの『関連資料』を参照)の『MS Discipline-Specific Application

QUERY_FOCAL_POINT

Programs』の 1 つか、タイプ 1134 の文字を使用した EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_MS_CATEGORY

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*ms_category* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは MS ネットワーク管理機能をサポートしていません。これは、DEFINE_NODE verb の *mds_supported* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_GLOBAL_LOG_TYPE

この verb により、Communications Server for Linux がログ・ファイルに記録する情報のタイプを NOF アプリケーションが判別できるようになります。これにより、すべてのサーバー上で使用されるデフォルト値 (SET_LOG_TYPE によって特定のサーバー上で指定変更されない場合) が指定されます。QUERY_LOG_TYPE を使用すると、特定のサーバー上で使用される値を決定することができます。

Communications Server for Linux は、以下のタイプのイベントについてメッセージをログに記録します。

問題 ユーザーが感知できる、システムの劣化を引き起こす異常イベント (セッションの異常終了など)。

例外 システムを劣化させることがあってもユーザーが感知できない異常イベント (リモート・システムからの無効なメッセージの受信など)。

監査 通常のイベント (セッションの開始など)。

問題メッセージおよび例外メッセージは、エラー・ログ・ファイルに記録されます。監査メッセージは監査ログ・ファイルに記録されます。問題メッセージは常に記録され、使用不可にできませんが、他の 2 つのタイプのメッセージは、それぞれ記録するかどうかを指定できます。2 つのファイル (監査とエラー) のそれぞれについて、簡略ログ (メッセージのテキストとメッセージ送信元の要約のみを含む) またはフル・ログ (メッセージの送信元、原因、および必要なすべての処置の詳細すべてを含む) のどちらを使用するかを指定できます。

この verb は、中央ロガーとして現在機能しているノードに発行する必要があります。詳しくは、70 ページの『CONNECT_NODE』を参照してください。

VCB 構造体

```
typedef struct query_global_log_type
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code         */
    AP_UINT32      secondary_rc;    /* secondary return code       */
    unsigned char  audit;          /* audit logging on or off     */
    unsigned char  exception;      /* exception logging on or off */
    unsigned char  succinct_audits; /* use succinct logging in audit file? */
    unsigned char  succinct_errors; /* use succinct logging in error file? */
    unsigned char  reserv3[4];     /* reserved                    */
} QUERY_GLOBAL_LOG_TYPE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_GLOBAL_LOG_TYPE

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

audit このパラメーターは、監査メッセージを記録するかどうかを指定します。値は次のいずれかです。

AP_YES 監査メッセージを記録します。

AP_NO 監査メッセージを記録しません。

QUERY_GLOBAL_LOG_TYPE

exception

このパラメーターは、例外メッセージを記録するかどうかを指定します。値は次のいずれかです。

AP_YES 例外メッセージを記録します。

AP_NO 例外メッセージを記録しません。

succinct_audits

このパラメーターは、監査ログ・ファイルで簡略ログまたはフル・ログのどちらを使用するかを指定します。値は次のいずれかです。

AP_YES 簡略ログ: ログ・ファイル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、ログ・タイプ、システム名など)、メッセージ・テキスト・ストリング、およびパラメーターが入ります。ログの原因と必要な処置の詳細を入手するには、**snahelp** ユーティリティを使用してください。

AP_NO フル・ログ: ログ・ファイル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、ログの原因と必要な処置に関する追加情報が入っています。

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ついて簡略ログまたはフル・ログのいずれを選択するかは、中央ロガーとして機能しているサーバー上でのこのパラメーターの設定によって決定されます。この設定は、デフォルトを指定変更するために、**SET_GLOBAL_LOG_TYPE verb** から、またはそのサーバーに発行された **SET_LOG_TYPE verb** から生じます。

succinct_errors

このパラメーターは、エラー・ログ・ファイルで簡略ログまたはフル・ログのどちらを使用するかを指定します。これは例外ログと問題ログの両方に適用されます。指定できる値とその意味は、*succinct_audits* パラメーターの場合と同じです。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために **verb** が実行されないと、**Communications Server for Linux** は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_NOT_CENTRAL_LOGGER

この **verb** は、中央ロガーではないノードに発行されました。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての **NOF verb** に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_ISR_SESSION

QUERY_ISR_SESSION は、ネットワーク・ノードが中間セッション経路指定を提供しているセッションに関するリスト情報を戻します。

この verb は、使用するオプションに応じて、特定のセッションまたは複数のセッションに関する情報を入手するために使用できます。この verb は Communications Server for Linux ノードがネットワーク・ノードである場合のみ使用可能であり、このノードがエンド・ノードまたは LEN ノードである場合は無効です。

このリストは、最初に *fqpcid.pcid*、次に *fqpcid.fqcp_name* の順番で EBCDIC の字句解釈配列により並べられます。fqpcid 構造体のフォーマットは、8 バイトの PCID (プロシージャール相関関係子 ID) と、セッション開始元のネットワーク修飾 CP 名です。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_isr_session
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  *buf_ptr;       /* pointer to buffer            */
    AP_UINT32      buf_size;       /* buffer size                  */
    AP_UINT32      total_buf_size; /* total buffer size required   */
    AP_UINT16      num_entries;     /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries      */
    unsigned char  list_options;    /* listing options              */
    unsigned char  session_type;    /* is this query for DLUR or regular */
                                   /* ISR sessions?                */
    FQPCID         fqpcid;         /* fully qualified procedure     */
                                   /* correlator ID                 */
} QUERY_ISR_SESSION;

typedef struct isr_session_summary
{
    AP_UINT16      overlay_size;    /* size of returned entry       */
    FQPCID         fqpcid;         /* fully qualified procedure     */
                                   /* correlator ID                 */
} ISR_SESSION_SUMMARY;

typedef struct isr_session_detail
{
    AP_UINT16      overlay_size;    /* size of returned entry       */
    AP_UINT16      sub_overlay_size; /* offset to appended RSCV     */
    FQPCID         fqpcid;         /* fully qualified procedure     */
                                   /* correlator ID                 */
    unsigned char  trans_pri;      /* Transmission priority:       */
    unsigned char  cos_name[8];    /* Class of Service name        */
    unsigned char  ltd_res;        /* Session spans a limited resource */
    unsigned char  reserv1[2];     /* reserved                      */
    EXTENDED_SESSION_STATS pri_ext_sess_stats; /* primary hop session stats */
    EXTENDED_SESSION_STATS sec_ext_sess_stats; /* secondary hop session stats */
    unsigned char  sess_lu_type;   /* session LU type              */
    unsigned char  sess_lu_level; /* session LU level             */
    unsigned char  pri_tg_number; /* Primary session TG number    */
    unsigned char  sec_tg_number; /* Secondary session TG number  */
    AP_UINT32      rtp_tcid;       /* RTP TC identifier            */
    AP_UINT32      time_active;    /* time elapsed since activation */
}
```

QUERY_ISR_SESSION

```
    unsigned char   isr_state;           /* current state of ISR session */
    unsigned char   reserv2[11];        /* reserved */
    unsigned char   mode_name[8];       /* mode name */
    unsigned char   pri_lu_name[17];    /* primary LU name */
    unsigned char   sec_lu_name[17];    /* secondary LU name */
    unsigned char   pri_adj_cp_name[17]; /* primary stage adjacent CP name */
    unsigned char   sec_adj_cp_name[17]; /* secondary stage adjacent CP name */
    unsigned char   reserv3[3];        /* reserved */
    unsigned char   rscv_len;          /* length of following RSCV */
} ISR_SESSION_DETAIL;
```

ISR セッション詳細構造体のあとに、SNA フォーマットで定義された経路選択制御ベクトル (RSCV) が続く場合があります。この制御ベクトルは、ネットワークを紹介するセッション経路を定義し、BIND を実行します。この RSCV が組み込まれるのは、ISR セッションについての RSCV の保管がノードの構成 (DEFINE_NODE で指定される) で指示されている場合のみです。

```
typedef struct fqpcid
{
    unsigned char   pcid[8];           /* procedure correlator identifier */
    unsigned char   fqcp_name[17];    /* originator's network qualified */
                                        /* CP name */
    unsigned char   reserve3[3];      /* reserved */
} FQPCID;

typedef struct extended_session_stats
{
    AP_UINT16       rcv_ru_size;      /* session receive RU size */
    AP_UINT16       send_ru_size;     /* session send RU size */
    AP_UINT16       max_send_btu_size; /* maximum send BTU size */
    AP_UINT16       max_rcv_btu_size; /* maximum rcv BTU size */
    AP_UINT16       max_send_pac_win; /* maximum send pacing window size */
    AP_UINT16       cur_send_pac_win; /* current send pacing window size */
    AP_UINT16       send_rpc;        /* send residual pacing count */
    AP_UINT16       max_rcv_pac_win; /* maximum rcv pacing window size */
    AP_UINT16       cur_rcv_pac_win; /* current rcv pacing window size */
    AP_UINT16       rcv_rpc;        /* receive residual pacing count */
    AP_UINT32       send_data_frames; /* number of data frames sent */
    AP_UINT32       send_fmd_data_frames; /* num fmd data frames sent */
    AP_UINT32       send_data_bytes; /* number of data bytes sent */
    AP_UINT32       send_fmd_data_bytes; /* number of fmd data bytes sent */
    AP_UINT32       rcv_data_frames; /* number of data frames received */
    AP_UINT32       rcv_fmd_data_frames; /* num fmd data frames received */
    AP_UINT32       rcv_data_bytes; /* number of data bytes received */
    AP_UINT32       rcv_fmd_data_bytes; /* number of fmd data bytes received */
    unsigned char   sidh;            /* session ID high byte (from LFSID) */
    unsigned char   sidl;            /* session ID low byte (from LFSID) */
    unsigned char   odai;            /* ODAI bit set */
    unsigned char   ls_name[8];      /* link station name */
    unsigned char   pacing_type;     /* type of pacing in use */
    unsigned char   reserv1[100];    /* reserved */
} EXTENDED_SESSION_STATS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_ISR_SESSION

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファーへのポインター。

buf_size

指定したデータ・バッファーのサイズ。

num_entries

セッションの最大数 (それらのセッションについてのデータが戻されます)。ある範囲でなく、特定のセッションに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

pcid パラメーターと *fqcp_name* パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

pcid パラメーターと *fqcp_name* パラメーターで指定したエントリーの直後のエントリーから開始します。

リストは、最初に *pcid* で (数値により)、次に *fqcp_name* の順番で並べられます。アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

session_type

DLUR 保守セッションまたは正規の ISR セッションのどちらを照会するかを指定します。値は次のいずれかです。

AP_DLUR_SESSIONS

DLUR 保守セッションを照会します。

AP_ISR_SESSIONS

正規の ISR セッションを照会します。

fqpcid.pcid

プロシージャer相関関係子 ID。これは 8 バイトからなる 16 進数ストリングです。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

fqpcid.fqcp_name

情報を求めるセッションの制御点の完全修飾名、またはセッションのリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

isr_session_summary.overlay_size

戻された *isr_session_summary* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *isr_session_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

isr_session_summary.fqpcid.pcid

プロシージャー相関関係子 ID。

isr_session_summary.fqpcid.fqcp_name

CP の完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

isr_session_detail.overlay_size

戻された *isr_session_detail* 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *isr_session_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

isr_session_detail.fqpcid.pcid

プロシージャー相関関係子 ID。

isr_session_detail.fqpcid.fqcp_name

CP の完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

isr_session_detail.trans_pri

伝送優先順位。このパラメーターの値は次のいずれかです。

AP_LOW AP_MEDIUM
AP_HIGH AP_NETWORK

isr_session_detail.cos_name

サービス・クラス名。これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC ストリングで、右側に EBCDIC のスペースを入れます。

isr_session_detail.ltd_res

セッションで限定リソース・リンクを使用するかどうかを指定します。値は次のいずれかです。

AP_YES セッションでは、限定リソース・リンクを使用します。

AP_NO セッションでは、限定リソース・リンクは使用しません。

2 つのセッション (1 次および 2 次) では、*extended_session_stats* 構造体には次のフィールドが入り、1 次セッションの場合は各フィールドの前に *isr_session_detail.pri_ext_sess_stats.*_** が付き、2 次セッションの場合は各フィールドの前に *isr_session_detail.sec_ext_sess_stats.*_** が付きます。

rcv_ru_size

受信 RU の最大サイズ。

send_ru_size

送信 RU の最大サイズ。

max_send_btu_size

送信できる BTU の最大サイズ。

QUERY_ISR_SESSION

<i>max_rcv_btu_size</i>	受信できる BTU の最大サイズ。
<i>max_send_pac_win</i>	送信ペーシング・ウィンドウの最大サイズ。
<i>cur_send_pac_win</i>	送信ペーシング・ウィンドウの現行サイズ。
<i>send_rpc</i>	送信残余ペーシング・カウント。
<i>max_rcv_pac_win</i>	受信ペーシング・ウィンドウの最大サイズ。
<i>cur_rcv_pac_win</i>	受信ペーシング・ウィンドウの現行サイズ。
<i>rcv_rpc</i>	受信残余ペーシング・カウント。
<i>send_data_frames</i>	送信された通常フロー・データ・フレームの数。
<i>send_fmd_data_frames</i>	送信された通常フロー FMD データ・フレームの数。
<i>send_data_bytes</i>	送信された通常フロー・データ・バイトの数。
<i>send_fmd_data_bytes</i>	送信された通常フロー FMD データ・バイトの数。
<i>rcv_data_frames</i>	受信された通常フロー・データ・フレームの数。
<i>rcv_fmd_data_frames</i>	受信された通常フロー FMD データ・フレームの数。
<i>rcv_data_bytes</i>	受信された通常フロー・データ・バイトの数。
<i>rcv_fmd_data_bytes</i>	受信された通常フロー FMD データ・バイトの数。
<i>sidh</i>	セッション ID の上位バイト。
<i>sidl</i>	セッション ID の下位バイト。
<i>odai</i>	起点宛先アサイナー・インディケーター。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合、BIND の送信側は、このフィールドをゼロに設定し、BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合は、このフィールドを 1 に設定します。
<i>ls_name</i>	統計に関連したリンク・ステーション名または RTP 接続の名前。これは 8 バイトからなるローカル・システム上で表示可能な文字セットのストリングです。8 バイトはすべて有効です。このフィールドは、中間セッション統計を特定のリンク・ステーションに相関させるために使用できます。

pacings_type

セッションで使用している受信ペーシング。値は次のいずれかです。

```
AP_NONE
AP_PACING_FIXED
AP_PACING_ADAPTIVE
```

次のパラメーターも戻されます (これらのパラメーターは `session_stats` 構造体の一部ではありません)。

isr_session.detail.sess_lu_type

BIND で指定されたセッションの LU タイプ。指定できる値は次のとおりです (LU タイプ 5 は意図的に除外してあります)。

```
AP_LU_TYPE_0
AP_LU_TYPE_1
AP_LU_TYPE_2
AP_LU_TYPE_3
AP_LU_TYPE_4
AP_LU_TYPE_6
AP_LU_TYPE_7
AP_LU_TYPE_UNKNOWN
```

isr_session.detail.sess_lu_level

セッションの LU レベル。値は次のいずれかです。

```
AP_LU_LEVEL_0
AP_LU_LEVEL_1
AP_LU_LEVEL_2
AP_LU_LEVEL_UNKNOWN
```

LU タイプが 6 以外の場合、このパラメーターは `AP_LU_LEVEL_0` に設定されます。 `DEFINE_ISR_STATS` を使用して名前の集合を使用可能にしない限り、常に、値 `AP_LU_LEVEL_UNKNOWN` が戻されます。

isr_session.detail.pri_tg_number

1 次セッション・ホップが通過したリンクに関連した TG 番号。1 次セッション・ステージが RTP 接続を通過した場合は、ゼロが戻されます。

`DEFINE_ISR_STATS` を使用して名前の集合を使用可能にしない限り、常に値ゼロになります。

isr_session.detail.sec_tg_number

2 次セッション・ホップが通過したリンクに関連した TG 番号。2 次セッション・ステージが RTP 接続を通過した場合は、ゼロが戻されます。

`DEFINE_ISR_STATS` を使用して名前の集合を使用可能にしない限り、常に値ゼロになります。

isr_session.detail.rtp_tcid

RTP 接続の合計 TC ID。これは、この ISR セッションが ANR/ISR 境界の一部である場合に戻されます。それ以外の場合は、このパラメーターはゼロに設定されます。 `DEFINE_ISR_STATS` を使用して名前の集合を使用可能にしない限り、常に値ゼロになります。

QUERY_ISR_SESSION

isr_session.detail.time_active

セッションを活動化してから経過した時間 (0.01 秒単位)。

DEFINE_ISR_STATS を使用して名前の集合を使用可能にしない限り、常に値ゼロになります。

isr_session.detail.isr_state

セッションの現行状態。値は次のいずれかです。

AP_ISR_INACTIVE
AP_ISR_PENDING_ACTIVE
AP_ISR_ACTIVE
AP_ISR_PENDING_INACTIVE

isr_session.detail.mode_name

セッションのモード名。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC 英数字ストリングで、右側に EBCDIC スペースを入れます。DEFINE_ISR_STATS を使用して名前の集合を使用可能にしない限り、すべてゼロになります。

isr_session.detail.pri_lu_name

セッションの 1 次 LU 名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなる LU 名で構成されます。この名前が使用できない場合、このフィールドはすべてゼロになります。DEFINE_ISR_STATS を使用して名前の集合を使用可能にしない限り、常にすべてゼロになります。

isr_session.detail.sec_lu_name

セッションの 2 次 LU 名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなる LU 名で構成されます。この名前が使用できない場合、このフィールドはすべてゼロになります。DEFINE_ISR_STATS を使用して名前の集合を使用可能にしない限り、常にすべてゼロになります。

isr_session.detail.pri_adj_cp_name

このセッションの 1 次ステージ隣接 CP 名。1 次セッションが RTP 接続を通過した場合は、リモート RTP エンドポイントの CP 名が戻されます。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなる CP 名で構成されます。この名前が使用できない場合、このフィールドはすべてゼロになります。DEFINE_ISR_STATS を使用して名前の集合を使用可能にしない限り、常にすべてゼロになります。

isr_session.detail.sec_adj_cp_name

このセッションの 2 次ステージ隣接 CP 名。2 次セッションが RTP 接続を通過した場合は、リモート RTP エンドポイントの CP 名が戻されます。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の

A スtring文字からなる CP 名で構成されます。この名前が使用できない場合、このフィールドはすべてゼロになります。 DEFINE_ISR_STATS を使用して名前の集合を使用可能にしない限り、常にすべてゼロになります。

isr_session_detail.rscv_len

session_detail 構造体に追加される RSCV の長さ。(何も追加しない場合は、長さはゼロです。)

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、 Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_FQPCID

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*pcid* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードがネットワーク・ノードでないために verb が正常に実行されないと、 Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_INVALID_VERB

ローカル・ノードがネットワーク・ノードではありません。この verb はネットワーク・ノードでのみ使用できます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_KERNEL_MEMORY_LIMIT

この verb は、 Communications Server for Linux が現在使用しているカーネル・メモリーの容量、これまでに使用した最大容量、および構成限度に関する情報を戻します。この verb を使用すると、メモリーの使用状況を検査して適切な限度を設定することが可能であり、それによって Communications Server for Linux のコンポーネントと Linux コンピューター上のその他のプログラムに、十分なメモリーを確実に使用可能にすることができます。

QUERY_KERNEL_MEMORY_LIMIT

Communications Server for Linux ソフトウェアの起動時にカーネル・メモリーの限度を指定したり (詳しくは、「*Communications Server for Linux 管理ガイド*」を参照)、あとでノードの実行時に (SET_KERNEL_MEMORY_LIMIT verb を使用して) 変更したりできます。

VCB 構造体

```
typedef struct query_kernel_memory_limit
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    AP_UINT32      limit;           /* kernel memory limit, 0 => no limit */
    AP_UINT32      actual;          /* current amount of memory allocated */
    AP_UINT32      max_used;        /* maximum amount of memory allocated */
    unsigned char  reset_max_used; /* set max_used = actual        */
    unsigned char  reserv3[8];     /* Reserved                     */
} QUERY_KERNEL_MEMORY_LIMIT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_KERNEL_MEMORY_LIMIT

reset_max_used

Communications Server for Linux が現在割り振られているメモリー容量に一致するように *max_used* 値を (この verb に戻したあと) リセットするかどうかを指定します。これにより、システムが始動してから (または *max_used* 値が最後にリセットされてから) のメモリーの最大使用量ではなく、この verb 以後のメモリーの最大使用量を、あとに続く QUERY_KERNEL_MEMORY_LIMIT verb で戻すことができるようになります。値は次のいずれかです。

AP_YES 現在のメモリー割り振りに一致させるために *max_used* 値をリセットします。

AP_NO *max_used* 値をリセットしません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

limit Communications Server for Linux で任意の時点で使用できるカーネル・メモリーの最大容量 (バイト単位)。Communications Server for Linux のコンポーネントの 1 つがカーネル・メモリーを割り振ろうとした結果、現在、Communications Server for Linux のコンポーネントに割り振られているメモリーの合計容量がこの限度を超えると、その割り振りの試みは失敗します。ゼロの値は制限がないことを示します。

actual 現在、Communications Server for Linux のコンポーネントに割り振られているカーネル・メモリーの容量 (バイト単位)。

max_used

max_used パラメーターが最後にリセットされてから (上記の *reset_max_used* の説明)、または Communications Server for Linux ソフトウェアが起動されてから、Communications Server for Linux のコンポーネントに任意の時点で割り振られていたカーネル・メモリーの最大容量 (バイト単位)。

reset_max_used

Communications Server for Linux が現在割り振られているメモリー容量に一致するように *max_used* 値を (このコマンドに戻したあと) リセットするかどうかを指定します。これにより、システムが始動してから (または *max_used* 値が最後にリセットされてから) の最大使用量ではなく、このコマンドが発行されてからのメモリーの最大使用量を、あとに続く QUERY_KERNEL_MEMORY_LIMIT verb で戻すことができるようになります。値は次のいずれかです。

AP_YES Communications Server for Linux は、現在のメモリー割り振りに一致するように *max_used* 値をリセットします。

AP_NO Communications Server for Linux は、*max_used* 値をリセットしません。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_LOCAL_LU

QUERY_LOCAL_LU は、ローカル LU に関する情報を戻します。

この verb は、使用するオプションに応じて、特定の LU または複数の LU に関する要約情報または詳細情報を入手するために使用できます。また、CP に関連した LU (デフォルト LU) に関する情報も入手できます。

VCB 構造体

```
typedef struct query_local_lu
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                 */
    unsigned char  format;          /* reserved                 */
    AP_UINT16      primary_rc;      /* primary return code     */
    AP_UINT32      secondary_rc;    /* secondary return code   */
    unsigned char  *buf_ptr;        /* pointer to buffer       */
    AP_UINT32      buf_size;        /* buffer size             */
    AP_UINT32      total_buf_size;  /* total buffer size required */
    AP_UINT16      num_entries;     /* number of entries       */
    AP_UINT16      total_num_entries; /* total number of entries */
    unsigned char  list_options;    /* listing options         */
    unsigned char  reserv3;         /* reserved                 */
}
```

QUERY_LOCAL_LU

```
    unsigned char    lu_name[8];           /* LU name */
    unsigned char    lu_alias[8];         /* LU alias */
    unsigned char    pu_name[8];         /* PU name filter */
} QUERY_LOCAL_LU;

typedef struct local_lu_summary
{
    AP_UINT16        overlay_size;        /* size of returned entry */
    unsigned char    lu_name[8];         /* LU name */
    unsigned char    lu_alias[8];         /* LU alias */
    unsigned char    description[32];     /* resource description */
    unsigned char    reserv1[16];        /* reserved */
} LOCAL_LU_SUMMARY;

typedef struct local_lu_detail
{
    AP_UINT16        overlay_size;        /* size of returned entry */
    unsigned char    lu_name[8];         /* LU name */
    LOCAL_LU_DEF_DATA def_data;          /* defined data */
    LOCAL_LU_DET_DATA det_data;         /* determined data */
} LOCAL_LU_DETAIL;

typedef struct local_lu_def_data
{
    unsigned char    description[32];     /* resource description */
    unsigned char    reserv1;            /* reserved */
    unsigned char    security_list_name[14]; /* security access list name */
    unsigned char    reserv3;            /* reserved */
    unsigned char    lu_alias[8];        /* local LU alias */
    unsigned char    nau_address;        /* NAU address */
    unsigned char    syncpt_support;     /* is Syncpoint supported? */
    AP_UINT16        lu_session_limit;   /* LU session limit */
    unsigned char    default_pool;       /* is LU in the pool of default
                                           /* LUs? */
    unsigned char    reserv2;            /* reserved */
    unsigned char    pu_name[8];         /* PU name */
    unsigned char    lu_attributes;      /* LU attributes */
    unsigned char    sscp_id[6];         /* SSCP ID */
    unsigned char    disable;            /* disable or enable local LU */
    ROUTING_DATA    attach_routing_data; /* routing data for incoming
                                           /* attaches */
    unsigned char    reserv6;            /* reserved */
    unsigned char    reserv4[7];         /* reserved */
    unsigned char    reserv5[16];        /* reserved */
} LOCAL_LU_DEF_DATA;

typedef struct local_lu_det_data
{
    unsigned char    lu_sscp_sess_active; /* Is LU-SSCP session active */
    unsigned char    appl_conn_active;    /* application is using LU */
    unsigned char    reserv1[2];          /* reserved */
    SESSION_STATS    lu_sscp_stats;       /* LU-SSCP session statistics */
    unsigned char    sscp_id[6];         /* SSCP ID */
} LOCAL_LU_DET_DATA;

typedef struct session_stats
{
    AP_UINT16        rcv_ru_size;         /* session receive RU size */
    AP_UINT16        send_ru_size;        /* session send Ru size */
    AP_UINT16        max_send_btu_size;   /* max send BTU size */
    AP_UINT16        max_rcv_btu_size;    /* max rcv BTU size */
    AP_UINT16        max_send_pac_win;    /* max send pacing window size */
    AP_UINT16        cur_send_pac_win;    /* current send pacing win size */
    AP_UINT16        max_rcv_pac_win;     /* max receive pacing win size */
    AP_UINT16        cur_rcv_pac_win;     /* current receive pacing
                                           /* window size */
    AP_UINT32        send_data_frames;    /* number of data frames sent */
    AP_UINT32        send_fmd_data_frames; /* num of fmd data frames sent */
    AP_UINT32        send_data_bytes;     /* number of data bytes sent */
}
```

```

AP_UINT32      rcv_data_frames;      /* num data frames received */
AP_UINT32      rcv_fmd_data_frames; /* num of fmd data frames recvd */
AP_UINT32      rcv_data_bytes;      /* number of data bytes received*/
unsigned char  sidh;                 /* session ID high byte */
unsigned char  sidl;                 /* session ID low byte */
unsigned char  odai;                 /* ODAI bit set */
unsigned char  ls_name;              /* link station name */
unsigned char  pacing_type;          /* type of pacing in use */
} SESSION_STATS;
typedef struct routing_data
{
    unsigned char  sys_name[128];     /* Name of target system for TP */
    AP_INT32      timeout;            /* timeout value in seconds */
    unsigned char  back_level;        /* reserved */
    unsigned char  reserved[59];     /* reserved */
} ROUTING_DATA;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_LOCAL_LU

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

LU の最大数 (その LU のデータが戻されます)。ある範囲でなく、特定の LU に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

lu_name パラメーターまたは *lu_alias* パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

lu_name パラメーターまたは *lu_alias* パラメーターで指定したエントリーの直後のエントリーから開始します。

AP_FIRST_IN_LIST を指定した場合は、論理 OR 演算を使用して、次のオプションも組み込むことができます。

AP_LIST_BY_ALIAS

リストは LU 名の順ではなく LU 別名の順に戻されます。このオプションは、AP_FIRST_IN_LIST と共に指定した場合のみ有効です。(AP_LIST_FROM_NEXT または AP_LIST_INCLUSIVE の場合は、リストの索引としてどちらを指定したかに応じて、リストは LU 名順または LU 別名順になります。)

アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。リストは EBCDIC 字句解釈配列です (各名前の長さに関係なく)。

lu_name

情報を求める LU の完全修飾名、または LU のリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。LU を名前ではなく別名で識別するには、このパラメーターを 8 桁の 2 進ゼロに設定し、*lu_alias* パラメーターで別名を指定します。デフォルト LU を識別するには、*lu_name* と *lu_alias* の両方の 8 桁の 2 進ゼロに設定します。

この名前は 8 バイトからなる EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

lu_alias

情報を求める LU の LU 別名、または LU のリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

これは 8 バイトからなる ASCII ストリングで、名前が 8 文字に満たない場合、右側にスペースを入れます。LU を別名ではなく名前で識別するには、このパラメーターを 8 桁の 2 進ゼロに設定し、*lu_name* パラメーターで名前を指定します。デフォルト LU を識別するには、*lu_name* と *lu_alias* を 8 桁の 2 進ゼロに設定します。

pu_name

PU 名フィルター。特定の PU に関連した LU のみに関する情報を戻すには、PU 名を指定します。PU 名で選別しないで情報を戻すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

この名前は 8 バイトからなるタイプ A の EBCDIC ストリング (英字で始まる) で、右側に EBCDIC スペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

local_lu_summary.overlay_size

戻された *local_lu_summary* 構造体のサイズ。すなわち、データ・バッファー内で次のエントリーが開始するまでのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *local_lu_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

local_lu_summary.lu_name

LU 名。この名前は 8 バイトからなるタイプ A の EBCDIC 文字ストリングです。

local_lu_summary.lu_alias

LU 別名。これは 8 バイトからなる ASCII 文字ストリングです。

local_lu_summary.description

ローカル LU を記述したヌルで終了するテキスト・ストリング (その LU の定義で指定したもの)。

local_lu_detail.overlay_size

戻された *local_lu_detail* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *local_lu_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

local_lu_detail.lu_name

LU 名。この名前は 8 バイトからなるタイプ A の EBCDIC 文字ストリングです。

QUERY_LOCAL_LU

local_lu_detail.def_data.description

ローカル LU を記述したヌルで終了するテキスト・ストリング (その LU の定義で指定したもの)。

local_lu_detail.def_data.security_list_name

このローカル LU が使用するセキュリティー・アクセス・リスト (DEFINE_SECURITY_ACCESS_LIST verb を使用して定義されたもの) の名前。このパラメーターが 14 桁の 2 進ゼロに設定されている場合は、すべてのユーザーがこの LU を使用できます。

local_lu_detail.def_data.lu_alias

LU 別名。これは 8 バイトからなる ASCII 文字ストリングです。

local_lu_detail.def_data.nau_address

LU のネットワーク・アクセス可能単位アドレス。LU が従属 LU の場合は 1 ~ 255 の範囲で、LU が独立 LU の場合はゼロです。

local_lu_detail.def_data.syncpt_support

LU が同期点機能をサポートするかどうかを指定します。値は次のいずれかです。

AP_YES 同期点はサポートされます。

AP_NO 同期点はサポートされません。

local_lu_detail.def_data.lu_session_limit

ローカル LU のセッションの最大合計数 (すべてのモード)。値ゼロは制限がないことを示します。

local_lu_detail.def_data.default_pool

LU がデフォルトの従属 LU のプールにあるかどうかを指定します。アプリケーションがローカル LU 名を指定しないで会話を開始しようとすると、Communications Server for Linux はこのプールから未使用の LU を選択します。値は次のいずれかです。

AP_YES LU はデフォルト LU のプールに入り、LU 名を指定しないアプリケーションから使用できます。

AP_NO LU はプールに入りません。

LU が独立 LU の場合、このパラメーターは予約済みです。

local_lu_detail.def_data.pu_name

従属 LU の場合、このパラメーターはこの LU が使用する PU を示します。これは 8 バイトからなる (英字で始まる) タイプ A の EBCDIC ストリングで、必要に応じて右側に EBCDIC のスペースを入れます。独立 LU の場合、このフィールドは使用しません。8 つのビットがゼロに設定されます。

local_lu_detail.def_data.lu_attributes

構成されている LU 属性。値は次のいずれかです。

AP_NONE

追加情報は識別されません。

AP_DISABLE_PWSUB

ローカル LU のパスワード置換サポートを使用不可にします。パスワード置換とは、パスワードを平文で送信せずに、ローカル LU と

リモート LU の間での伝送を行う前にパスワードを暗号化することを言います。通常、Communications Server for Linux は、リモート・システムがパスワード置換をサポートしている場合、パスワード置換を使用します。

この値は、パスワード置換を正しくインプリメントしていないリモート・システムと通信するための予備手段として提供されています。このオプションを使用する場合、平文でのパスワードの送受信(セキュリティ上のリスクがある場合があります)が行われることに注意しなければなりません。このオプションは、リモート・システムにおけるパスワード置換のインプリメンテーションに関して問題がある場合以外は設定しないでください。

local_lu_detail.def_data.sscp_id

この LU の活動化が許可される SSCP の ID を指定します。ID は 6 バイトの 2 進数フィールドです。このパラメーターは従属 LU のみが使用し、独立 LU の場合または SSCP が LU を活動化できる場合は、このパラメーターはすべてゼロに設定されます。

local_lu_detail.def_data.attach_routing_data.sys_name

このローカル LU に着信する着呼割り振り要求 (APPC または CPI-C 会話を開始するためのパートナー TP からの要求) のターゲット・コンピューターの名前。これにより、ターゲット TP が実行されるコンピューターが識別されます。

このパラメーターが 2 進ゼロに設定されている場合、Communications Server for Linux は、着呼割り振り要求を TP の実行コピーに動的に経路指定する (使用可能な場合) か、またはローカル LU と同じコンピューター上で TP を開始しようとします。

local_lu_detail.def_data.attach_routing_data.timeout

ダイナミック・ロード要求のタイムアウト値 (秒数)。呼び出された TP がこの時間内に Receive_Allocate verb (APPC)、Accept_Conversation または Accept_Incoming (CPI-C) を発行しなかった場合、要求はタイムアウトになります。-1 の値はタイムアウトなし (ダイナミック・ロード要求を無限に待つ) を示します。

以下のパラメーターは従属 LU だけに使用されます。独立 LU には、これらのパラメーターは予約済みです (2 進ゼロに設定される)。この LU とパートナー LU 間の適切なセッションに対しては、QUERY_SESSION verb を発行することにより、同等の情報を入手できます。

local_lu_detail.det_data.lu_sscp_session_active

LU-SSCP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES LU-SSCP セッションは活動状態です。

AP_NO LU-SSCP セッションは活動状態ではありません。

local_lu_detail.det_data.appl_conn_active

APPC または CPI-C トランザクション・プログラム (TP) が、このローカル LU からのセッションでホスト LU と会話状態であるかどうかを指定します。値は次のいずれかです。

QUERY_LOCAL_LU

AP_YES 会話は LU を使用して進行中です。

AP_NO この LU に対して 1 つ以上の TP が TP_STARTED を発行した可能性があります。LU を使用する会話は進行中ではありません。

local_lu_detail.det_data.lu_sscp_stats

LU-SSCP セッションの統計情報。

local_lu_detail.det_data.lu_sscp_stats.rcv_ru_size

このパラメーターは常に予約済みです。

local_lu_detail.det_data.lu_sscp_stats.send_ru_size

このパラメーターは常に予約済みです。

local_lu_detail.det_data.lu_sscp_stats.max_send_btu_size

送信できる基本伝送単位 (BTU) の最大サイズ。

local_lu_detail.det_data.lu_sscp_stats.max_rcv_btu_size

受信できる BTU の最大サイズ。

local_lu_detail.det_data.lu_sscp_stats.max_send_pac_win

このパラメーターは常にゼロに設定されます。

local_lu_detail.det_data.lu_sscp_stats.cur_send_pac_win

このパラメーターは常にゼロに設定されます。

local_lu_detail.det_data.lu_sscp_stats.max_rcv_pac_win

このパラメーターは常にゼロに設定されます。

local_lu_detail.det_data.lu_sscp_stats.cur_rcv_pac_win

このパラメーターは常にゼロに設定されます。

local_lu_detail.det_data.lu_sscp_stats.send_data_frames

送信された通常フロー・データ・フレームの数。

local_lu_detail.det_data.lu_sscp_stats.send_fmd_data_frames

送信された通常フロー機能管理データ (FMD) フレームの数。

local_lu_detail.det_data.lu_sscp_stats.send_data_bytes

送信された通常フロー・データ・バイトの数。

local_lu_detail.det_data.lu_sscp_stats.rcv_data_frames

受信された通常フロー・データ・フレームの数。

local_lu_detail.det_data.lu_sscp_stats.rcv_fmd_data_frames

受信された通常フロー FMD データ・フレームの数。

local_lu_detail.det_data.lu_sscp_stats.rcv_data_bytes

受信された通常フロー・データ・バイトの数。

local_lu_detail.det_data.lu_sscp_stats.sidh

セッション ID の上位バイト。

local_lu_detail.det_data.lu_sscp_stats.sidl

セッション ID の下位バイト。

local_lu_detail.det_data.lu_sscp_stats.odai

起点宛先アサイナー・インディケーター。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合は、ACTLU の送

信側は、このパラメーターをゼロに設定し、ACTLU 送信側のノードに 2 次リンク・ステーションが含まれる場合には、1 に設定します。

local_lu_detail.det_data.lu_sscp_stats.ls_name

統計に関連したリンク・ステーション名。これは 8 バイトからなるローカル・システム上で表示可能な文字セットのストリングです。8 バイトはすべて有効です。このパラメーターは、セッションの転送経路となるリンクにこのセッションを相互に関連付けるために使用できます。

local_lu_detail.det_data.lu_sscp_stats.pacing_type

LU-SSCP セッションで使用している受信ペーシングのタイプ。このパラメーターは AP_NONE に設定されます。

local_lu_detail.det_data.sscp_id

このパラメーターは、この LU が使用する PU の ACTPU で受信された SSCP ID が入る 6 バイトのフィールドです。

lu_sscp_sess_active が AP_YES に設定されていない場合、このパラメーターは予約済みです。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_ALIAS

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*lu_alias* パラメーターは無効でした。

AP_INVALID_LU_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*lu_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_LOCAL_TOPOLOGY

すべての APPN ノードは、すべての隣接ノードに対する TG に関する情報を保持するローカル・トポロジー・データベースを保守します。

QUERY_LOCAL_TOPOLOGY により、これらの TG に関する情報を戻すことができます。

この verb は、使用するオプションに応じて、特定の TG または複数の TG に関する要約情報または詳細情報のいずれかを、入手するために使用できます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_local_topology
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;          /* reserved                  */
    unsigned char  format;           /* reserved                  */
    AP_UINT16      primary_rc;       /* primary return code      */
    AP_UINT32      secondary_rc;     /* secondary return code    */
    unsigned char  *buf_ptr;         /* pointer to buffer        */
    AP_UINT32      buf_size;         /* buffer size              */
    AP_UINT32      total_buf_size;   /* total buffer size required */
    AP_UINT16      num_entries;      /* number of entries        */
    AP_UINT16      total_num_entries; /* total number of entries  */
    unsigned char  list_options;     /* listing options          */
    unsigned char  reserv3;          /* reserved                  */
    unsigned char  dest[17];         /* TG destination node      */
    unsigned char  dest_type;        /* TG destination node type */
    unsigned char  tg_num;           /* TG number                 */
} QUERY_LOCAL_TOPOLOGY;

typedef struct local_topology_summary
{
    AP_UINT16      overlay_size;     /* size of returned entry   */
    unsigned char  dest[17];         /* TG destination node      */
    unsigned char  dest_type;        /* TG destination node type */
    unsigned char  tg_num;           /* TG number                 */
} LOCAL_TOPOLOGY_SUMMARY;

typedef struct local_topology_detail
{
    AP_UINT16      overlay_size;     /* size of returned entry   */
    unsigned char  dest[17];         /* TG destination node      */
    unsigned char  dest_type;        /* TG destination node type */
    unsigned char  tg_num;           /* TG number                 */
    unsigned char  reserv1;          /* reserved                  */
    LINK_ADDRESS   dlc_data;         /* DLC signalling data      */
    AP_UINT32      rsn;              /* resource sequence number */
    unsigned char  status;           /* tg status                 */
    TG_DEFINED_CHARS tg_chars;       /* TG characteristics       */
    unsigned char  cp_cp_session_active; /* CP-CP sessions active? */
    unsigned char  branch_link_type; /* Up or down link?        */
    unsigned char  branch_tg;       /* Branch TG?               */
    unsigned char  appended_data_format; /* Format of appended data */
    unsigned char  appended_data_len; /* Length of appended data  */
    unsigned char  reserva[11];     /* reserved                  */
} LOCAL_TOPOLOGY_DETAIL;

typedef struct link_address
{
    unsigned char  format;           /* type of link address     */
}
```

```

unsigned char  reservel;           /* reserved          */
AP_UINT16     length;            /* length           */
unsigned char  address[32];      /* address          */
} LINK_ADDRESS;

```

TG_DEFINED_CHARS 構造体について詳しくは、135 ページの『DEFINE_LS』を参照してください。

list_options パラメーターが詳細情報を指定すると、TG Descriptor CV が戻された情報に付加される場合があります。詳しくは、パラメーター *local_topology_detail.appended_data_format* および *local_topology_detail.appended_data_len* の説明を参照してください。

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_LOCAL_TOPOLOGY

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

エントリーの最大数 (そのエントリーのデータが戻されます)。ある範囲でなく、特定のエントリーに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

dest、*dest_type*、*tg_num* の各パラメーターを組み合わせ指定したエントリーから開始します。

AP_LIST_FROM_NEXT

dest、*dest_type*、*tg_num* の各パラメーターを組み合わせ指定したエントリーの直後のエントリーから開始します。

リストは、最初に *dest*、次に *dest_type* (AP_NETWORK_NODE、AP_END_NODE、AP_VRN の順)、最後に *tg_num* の数値順で並んでいます。リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

dest 情報を求める TG の宛先ノードの完全修飾名、または TG のリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A スtring 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring 文字からなるネットワーク名で構成されます。

dest_type

この TG の宛先ノードのノード・タイプ。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。値は次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_VRN 仮想経路指定ノード。

AP_END_NODE

エンド・ノードまたは LEN ノード。

AP_LEARN_NODE

不明なノード・タイプ。

tg_num TG に関連した番号。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファ内の各エントリは、次のパラメーターで構成されています。

local_topology_summary.overlay_size

戻された `local_topology_summary` 構造体のサイズ。すなわち、データ・バッファにおける次のエントリの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 `local_topology_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは `Communications Server for Linux` の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

local_topology_summary.dest

TG の宛先ノードの完全修飾名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

local_topology_summary.dest_type

この TG の宛先ノードのノード・タイプ。次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_VRN 仮想経路指定ノード。

AP_END_NODE

エンド・ノードまたは LEN ノード。

local_topology_summary.tg_num

TG に関連した番号。

local_topology_detail.overlay_size

戻された `local_topology_detail` 構造体のサイズ。すなわち、データ・バッファにおける次のエントリの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 `local_topology_detail` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは `Communications Server for Linux` の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

local_topology_detail.dest

TG の宛先ノードの完全修飾名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の

QUERY_LOCAL_TOPOLOGY

A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A スtring文字からなるネットワーク名で構成されます。

local_topology_detail.dest_type

この TG の宛先ノードのノード・タイプ。次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_VRN 仮想経路指定ノード。

AP_END_NODE

エンド・ノードまたは LEN ノード。

local_topology_detail.tg_num

TG に関連した番号。

local_topology_detail.dlc_data.length

dest_type が AP_VRN の場合は、このフィールドは VRN と接続する DLC アドレスの長さを指定します。それ以外の場合は、このフィールドは使用されず、ゼロに設定されます。

local_topology_detail.dlc_data.address

dest_type が AP_VRN の場合は、このフィールドは VRN と接続する DLC アドレス (16 進数) を指定します。アドレスのバイト数は直前のフィールド (length) で指定され、このフィールドの残りのバイトは定義されません。それ以外の場合は、このフィールドは使用されません。

トークンリングまたはイーサネットの場合は、このフィールドは 6 バイトの MAC アドレスと 1 バイトのローカル SAP アドレスの 2 つの部分で構成されます。MAC アドレスのビット配列は予期したフォーマットと異なる場合があります。2 つのアドレス・フォーマット間の変換については、163 ページの『MAC アドレス内のビット順序』を参照してください。

local_topology_detail.rsn

リソース・シーケンス番号。この番号は、このリソースを所有しているネットワーク・ノードにより割り当てられます。

local_topology_detail.status

TG の状況を指定します。値は次のうちの 1 つ以上を論理 OR 演算で組み合わせます。

AP_TG_OPERATIVE

AP_TG_CP_CP_SESSIONS

AP_TG QUIESCING

AP_TG_HPR

AP_TG_RTP

local_topology_detail.tg_chars

TG 特性。これらのパラメーターについて詳しくは、135 ページの『DEFINE_LS』を参照してください。

local_topology_detail.cp_cp_session_active

所有側ノードのコンテンション勝者 CP-CP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES CP-CP セッションは活動状態です。

AP_NO CP-CP セッションは活動状態ではありません。

AP_UNKNOWN

CP-CP セッションの状況は不明です。

local_topology_detail.branch_link_type

このパラメーターは、ノードが分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

この TG の分岐リンク・タイプを指定します。値は次のいずれかです。

AP_UPLINK

TG はアップリンクです

AP_DOWNLINK

TG はエンド・ノードへのダウンリンクです。

AP_DOWNLINK_TO_BRNN

TG は、ローカル・ノードの全体像のエンド・ノードとして表示される分岐ネットワーク・ノードへのダウンリンクです。

AP_OTHERLINK

TG は VRN へのリンクです。

local_topology_detail.branch_tg

このパラメーターは、ノードがネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

TG が分岐 TG であるかどうかを指定します。値は次のいずれかです。

AP_YES TG は分岐 TG です。

AP_NO TG は分岐 TG ではありません。

AP_UNKNOWN

TG のタイプは不明です。

local_topology_detail.appended_data_format

この NOF VCB 構造体に付加されるデータのフォーマットを指定します。

パラメーター *local_topology_detail.appended_data_len* を、付加されたデータが組み込まれていることを示すゼロ以外の値に設定すると、このパラメーターは以下の値に設定されます。

AP_TG_DESCRIPTOR_CV

付加されたデータには、SNA フォーマットで定義された TG Descriptor CV が含まれます。

local_topology_detail.appended_data_len が、付加されたデータが組み込まれていないことを示すゼロの場合は、このパラメーターは予約済みです。

QUERY_LOCAL_TOPOLOGY

local_topology_detail.appended_data_len

この NOF VCB 構造体に付加される TG Descriptor CV データの長さを指定します。この値がゼロに設定されている場合は、付加されたデータは組み込まれません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_TG

list_options パラメーターは、指定した名前から始まるすべてのエンタリーを表示する AP_LIST_INCLUSIVE に設定しましたが、*tg_num* パラメーターが有効ではありませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_LOG_FILE

この *verb* を使用すると、Communications Server for Linux が監査、エラー、または使用量の各ログ・メッセージの記録に使用するファイルの名前、バックアップ・ログ・ファイルの名前、およびバックアップ・ファイルにログ情報をコピーする際のファイル・サイズについて、アプリケーションで判別できます。

VCB 構造体

```
typedef struct query_log_file
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;               /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;         /* secondary return code */
    unsigned char  log_file_type;        /* type of log file */
    unsigned char  file_name[81];        /* file name */
    unsigned char  backup_file_name[81]; /* backup file name */
    AP_UINT32      file_size;            /* log file size */
    unsigned char  succinct;             /* reserved */
    unsigned char  reserv3[3];          /* reserved */
} QUERY_LOG_FILE;
```

指定パラメーター

opcode AP_QUERY_LOG_FILE

log_file_type

照会するログ・ファイルのタイプ。値は次のいずれかです。

AP_AUDIT_FILE

監査ログ・ファイル (監査メッセージのみ)。

AP_ERROR_FILE

エラー・ログ・ファイル (問題メッセージおよび例外メッセージ)。

AP_USAGE_FILE

使用量ログ・ファイル (Communications Server for Linux リソースの現在とピーク時の使用量に関する情報)。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

file_name

ログ・ファイルの名前。このパラメーターは 1 ~ 80 文字の ASCII スtringで、最後にヌル文字 (0x00) が続きます。

パスが指定されていない場合、このファイルは診断ファイルのデフォルト・ディレクトリー **/var/opt/ibm/sna** に保管されます。パスが指定されている場合は、絶対パス (/ 文字で始まる) か、デフォルト・ディレクトリーに対する相対パスのいずれかです。

backup_file_name

バックアップ・ログ・ファイルの名前。このパラメーターは 1 ~ 80 文字の ASCII Stringで、最後にヌル文字 (0x00) が続きます。

ログ・ファイルが次の *file_size* で指定したサイズになると、Communications Server for Linux は、ログ・ファイルの現在の内容をこの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。また、SET_LOG_FILE *verb* を使用すれば、いつでもバックアップを要求できます。

パスが指定されていない場合、このファイルは診断ファイルのデフォルト・ディレクトリー **/var/opt/ibm/sna** に保管されます。パスが指定されている場合は、絶対パス (/ 文字で始まる) か、デフォルト・ディレクトリーに対する相対パスのいずれかです。

file_size

log_file_type で指定したログ・ファイルの最大サイズ。ファイルへのメッセージの書き込みでファイル・サイズがこの限度を超えると、Communications Server for Linux はバックアップ・ログ・ファイルをクリアし、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

QUERY_LOG_FILE

グ・ファイルをクリアします。したがって、ログ・ファイルが占有するディスク・スペースの最大容量は *file_size* の値の約 2 倍になります。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_FILE_TYPE

log_file_type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_LOG_TYPE

この *verb* により、NOF アプリケーションは、Communications Server for Linux が特定のサーバー上でログ・ファイルに記録する情報のタイプを判別し、これらが SET_GLOBAL_LOG_TYPE に指定されたデフォルト設定であるか、または前の SET_LOG_TYPE *verb* によって指定されたローカル設定であるかを判別できるようになります。

Communications Server for Linux は、以下のタイプのイベントについてメッセージをログに記録します。

問題 ユーザーが感知できる、システムの劣化を引き起こす異常イベント (セッションの異常終了など)。

例外 システムを劣化させることがあってもユーザーが感知できない異常イベント (リモート・システムからの無効なメッセージの受信など)。

監査 通常のイベント (セッションの開始など)。

問題メッセージおよび例外メッセージは、エラー・ログ・ファイルに記録されません。監査メッセージは監査ログ・ファイルに記録されます。問題メッセージは常に記録され、使用不可にできませんが、他の 2 つのタイプのメッセージは、それぞれ記録するかどうかを指定できます。2 つのファイル (監査とエラー) のそれぞれについて、簡略ログ (メッセージのテキストとメッセージ送信元の要約のみを含む) またはフル・ログ (メッセージの送信元、原因、および必要なすべての処置の詳細すべてを含む) のどちらを使用するかを指定できます。

VCB 構造体

```
typedef struct query_log_type
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  override;       /* overriding global settings? */
    unsigned char  audit;          /* audit logging on or off     */
    unsigned char  exception;      /* exception logging on or off */
    unsigned char  succinct_audits; /* use succinct logging in audit file? */
    unsigned char  succinct_errors; /* use succinct logging in error file? */
    unsigned char  reserv3[3];     /* reserved                     */
} QUERY_LOG_TYPE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_LOG_TYPE

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

override

この *verb* で戻されたログ・タイプおよび簡略ログまたはフル・ログのオプションが、SET_GLOBAL_LOG_TYPE 上で指定されたグローバル・ログ・タイプであるか、または SET_LOG_TYPE 上で指定されたローカル値であるかを指定します。値は次のいずれかです。

AP_YES 戻された *audit*、*exception*、および *succinct_** パラメーターは、グローバル設定を指定変更するローカル設定です。

AP_NO 戻された *audit*、*exception*、および *succinct_** パラメーターはグローバル設定で、指定変更されません。

audit

このパラメーターは、監査メッセージを記録するかどうかを指定します。値は次のいずれかです。

AP_YES 監査メッセージを記録します。

AP_NO 監査メッセージを記録しません。

exception

このパラメーターは、例外メッセージを記録するかどうかを指定します。値は次のいずれかです。

AP_YES 例外メッセージを記録します。

AP_NO 例外メッセージを記録しません。

QUERY_LOG_TYPE

succinct_audits

このパラメーターは、監査ログ・ファイルで簡略ログまたはフル・ログのどちらを使用するかを指定します。値は次のいずれかです。

AP_YES 簡略ログ: ログ・ファイル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、ログ・タイプ、システム名など)、メッセージ・テキスト・ストリング、およびパラメーターが入ります。ログの原因と必要な処置の詳細を入手するには、**snahelp** ユーティリティを使用してください。

AP_NO フル・ログ: ログ・ファイル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、ログの原因と必要な処置に関する追加情報が入っています。

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ついて簡略ログまたはフル・ログのいずれを選択するかは、中央ログャーとして機能しているサーバー上でのこのパラメーターの設定によって決定されます。この設定は、デフォルトを指定変更するために、

SET_GLOBAL_LOG_TYPE verb から、またはそのサーバーに発行された **SET_LOG_TYPE verb** から生じます。

succinct_errors

このパラメーターは、エラー・ログ・ファイルで簡略ログまたはフル・ログのどちらを使用するかを指定します。これは例外ログと問題ログの両方に適用されます。指定できる値とその意味は、*succinct_audits* パラメーターの場合と同じです。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての **NOF verb** に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_LS

QUERY_LS は、ノードで定義されているリンク・ステーションについての情報のリストを戻します。この情報は、「決定済みデータ」(実行中に動的に収集されたデータ、ノードが活動状態である場合のみ戻される) および「定義済みデータ」(**DEFINE_LS** で提供されたデータ) として構造化されます。

この **verb** は、使用するオプションに応じて、特定の **LS** または複数の **LS** に関する要約情報または詳細情報のいずれかを入手するために使用できます。

VCB 構造体

```
typedef struct query_ls
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;                /* reserved                  */
    AP_UINT16      primary_rc;            /* Primary return code      */
    AP_UINT32      secondary_rc;          /* Secondary return code    */
    unsigned char  *buf_ptr;              /* pointer to buffer        */
    AP_UINT32      buf_size;              /* buffer size              */
    AP_UINT32      total_buf_size;        /* total buffer size required */
}
```



```

    AP_UINT16      num_entries;          /* number of entries      */
    AP_UINT16      total_num_entries;    /* total number of entries */
    unsigned char  list_options;         /* listing options        */
    unsigned char  reserv3;              /* reserved                */
    unsigned char  ls_name[8];           /* name of link station    */
    unsigned char  port_name[8];         /* port used by link station */
} QUERY_LS;

typedef struct ls_summary
{
    AP_UINT16      overlay_size;          /* size of returned entry  */
    unsigned char  ls_name[8];            /* link station name       */
    unsigned char  description[32];       /* resource description     */
    unsigned char  reserv1[16];           /* reserved                 */
    unsigned char  dlc_type;              /* DLC type                */
    unsigned char  state;                 /* link station state      */
    AP_UINT16      act_sess_count;        /* currently active sessions */
                                          /* count                   */
    unsigned char  det_adj_cp_name[17];   /* determined adjacent CP name */
    unsigned char  det_adj_cp_type;       /* determined adjacent node type*/
    unsigned char  port_name[8];         /* port name               */
    unsigned char  adj_cp_name[17];       /* adjacent CP name        */
    unsigned char  adj_cp_type;           /* adjacent node type      */
} LS_SUMMARY;

typedef struct ls_detail
{
    AP_UINT16      overlay_size;          /* size of returned entry  */
    unsigned char  ls_name[8];            /* link station name       */
    LS_DET_DATA    det_data;              /* determined data         */
    LS_DEF_DATA    def_data;              /* defined data            */
} LS_DETAIL;

typedef struct ls_det_data
{
    AP_UINT16      act_sess_count;        /* currently active sessions */
                                          /* count                   */
    unsigned char  dlc_type;              /* DLC type                */
    unsigned char  state;                 /* link station state      */
    unsigned char  sub_state;             /* link station sub state  */
    unsigned char  det_adj_cp_name[17];   /* adjacent CP name        */
    unsigned char  det_adj_cp_type;       /* adjacent node type      */
    unsigned char  dlc_name[8];           /* name of DLC             */
    unsigned char  dynamic;               /* specifies whether LS is */
                                          /* dynamic                  */
    unsigned char  migration;             /* supports migration partners */
    unsigned char  tg_num;                 /* TG number               */
    LS_STATS       ls_stats;               /* link station statistics */
    AP_UINT32      start_time;             /* time LS started         */
    AP_UINT32      stop_time;             /* time LS stopped         */
    AP_UINT32      up_time;                /* total time LS active    */
    AP_UINT32      current_state_time;     /* time in current state   */
    unsigned char  deact_cause;            /* deactivation cause      */
    unsigned char  hpr_support;            /* TG HPR support          */
    unsigned char  anr_label[2];           /* local ANR label        */
    unsigned char  hpr_link_lvl_error;     /* HPR link-level error    */
    unsigned char  auto_act;               /* auto-activation supported */
    unsigned char  ls_role;                /* LS role                  */
    unsigned char  ls_type;                /* LS type (defined,dynamic,..) */
    unsigned char  node_id[4];             /* determined node ID      */
    AP_UINT16      active_isr_count;        /* active isr count        */
    AP_UINT16      active_lu_sess_count;    /* count of active LU sessions */
    AP_UINT16      active_sscp_sess_count; /* count of active SSCP sessions*/
    ANR_LABEL      reverse_anr_label;       /* Reverse ANR label       */
    LINK_ADDRESS   local_address;           /* Local address           */
    AP_UINT16      max_send_btu_size;      /* Max send BTU size       */
    unsigned char  brnn_link_type;         /* type of branch link     */
    unsigned char  adj_cp_is_brnn;         /* is adjacent node a BrNN? */
}

```

QUERY_LS

```

    unsigned char    mltg_member;          /* reserved */
    unsigned char    tg_sharing;          /* reserved */
    unsigned char    reservb[62];        /* reserved */
} LS_DET_DATA;

typedef struct ls_def_data
{
    unsigned char    description[32];     /* resource description */
    unsigned char    initially_active;    /* is this LS initially active? */
    AP_UINT16        reserv2;            /* reserved */
    AP_UINT16        react_timer;        /* timer for retrying failed LS */
    AP_UINT16        react_timer_retry;  /* retry count for failed LS */
    AP_UINT16        activation_count;    /* reserved */
    unsigned char    restart_on_normal_deact; /* restart the link on any
                                           /* failure */

    unsigned char    reserv3[7];         /* reserved */
    unsigned char    port_name[8];       /* name of associated port */
    unsigned char    adj_cp_name[17];    /* adjacent CP name */
    unsigned char    adj_cp_type;        /* adjacent node type */
    LINK_ADDRESS     dest_address;        /* destination address */
    unsigned char    auto_act_supp;      /* auto-activate supported */
    unsigned char    tg_number;          /* pre-assigned TG number */
    unsigned char    limited_resource;   /* limited resource */
    unsigned char    solicit_sscp_sessions; /* solicit SSCP sessions */
    unsigned char    pu_name[8];         /* Local PU name (reserved if
                                           /* solicit_sscp_sessions is
                                           /* set to AP_NO)

    unsigned char    disable_remote_act; /* disable remote activation */
    unsigned char    dspu_services;      /* Services provided for
                                           /* downstream PU

    unsigned char    dspu_name[8];       /* Downstream PU name (reserved
                                           /* if dspu_services is AP_NONE)*/
    unsigned char    dlus_name[17];      /* DLUS name if dspu_services
                                           /* is AP_DLUR

    unsigned char    bkup_dlus_name[17]; /* Backup DLUS name if
                                           /* dspu_services is AP_DLUR

    unsigned char    hpr_supported;      /* does the link support HPR? */
    unsigned char    hpr_link_lvl_error; /* does link use link-level
                                           /* recovery for HPR frames?

    AP_UINT16        link_deact_timer;    /* deact timer for limited
                                           /* resource

    unsigned char    reserv1;            /* reserved */
    unsigned char    default_nn_server;  /* default LS to NN server? */
    unsigned char    ls_attributes[4];   /* LS attributes */
    unsigned char    adj_node_id[4];     /* adjacent node ID */
    unsigned char    local_node_id[4];   /* local node ID */
    unsigned char    cp_cp_sess_support; /* CP-CP session support */
    unsigned char    use_default_tg_chars; /* Use default tg_chars */
    TG_DEFINED_CHARS tg_chars;           /* TG characteristics */
    AP_UINT16        target_pacing_count; /* target pacing count */
    AP_UINT16        max_send_btu_size;   /* maximum send BTU size */
    AP_UINT16        ls_role;            /* link station role */
    unsigned char    max_ifrm_rcvd;      /* no. before acknowledgment */
    AP_UINT16        dlus_retry_timeout;  /* seconds to recontact a DLUS */
    AP_UINT16        dlus_retry_limit;    /* attempts to recontact a DLUS */
    unsigned char    conventional_lu_compression; /* compression for LU 0-3? */
    unsigned char    conventional_lu_cryptography; /* reserved */
    unsigned char    reserv3a;           /* reserved */
    unsigned char    retry_flags;        /* reserved */
    AP_UINT16        max_activation_attempts; /* reserved */
    AP_UINT16        activation_delay_timer; /* reserved */
    unsigned char    branch_link_type;   /* is link an up or down link */
    unsigned char    adj_brnn_cp_support; /* adj CP allowed to be BrNN? */
    unsigned char    mltg_pacing_algorithm; /* reserved */
    unsigned char    reserv5;           /* reserved */
    AP_UINT16        max_rcv_btu_size;    /* reserved */
    unsigned char    tg_sharing_prohibited; /* reserved */
    unsigned char    link_spec_data_format; /* reserved */

```

```

    unsigned char    pu_can_send_dddllu_offline; /* does the PU send NMVT      */
                                                    /* (power off) to the host? */
    unsigned char    reserv4[13];                /* reserved                  */
    AP_UINT16        link_spec_data_len;        /* length of link specific data */
} LS_DEF_DATA;
typedef struct link_address
{
    unsigned char    format;                    /* type of link address      */
    unsigned char    reserve1;                 /* reserved                  */
    AP_UINT16        length;                   /* length                    */
    unsigned char    address[32];              /* address                    */
} LINK_ADDRESS;

```

トークンリングまたはイーサネットの場合、link_address 構造体にある address パラメータは次のように置換されます。

```

typedef struct tr_address
{
    unsigned char    mac_address[6];           /* MAC address               */
    unsigned char    lsap_address;            /* local SAP address         */
} TR_ADDRESS;

```

Enterprise Extender (HPR/IP) の場合、link_address 構造体にある address パラメータは次のように置換されます。

```

typedef struct ip_address_info
{
    unsigned char    lsap;                    /* Local Service Access Point addr */
    unsigned char    version;                 /* IPv4 or IPv6              */
    unsigned char    address[272];           /* IP Address or hostname     */
} IP_ADDRESS_INFO;

```

マルチパス・チャンネル (MPC) または MPC+ の場合、link_address 構造体にある address パラメータは次のように置換されます。

```
typedef unsigned char GDLC_MPC_ADDRESS[20];
```

すべてのリンク・タイプの場合:

```

typedef struct tg_defined_chars
{
    unsigned char    effect_cap;              /* Effective capacity        */
    unsigned char    reserve1[5];            /* Reserved                  */
    unsigned char    connect_cost;           /* Connection Cost          */
    unsigned char    byte_cost;              /* Byte cost                 */
    unsigned char    reserve2;               /* Reserved                  */
    unsigned char    security;               /* Security                  */
    unsigned char    prop_delay;             /* Propagation delay        */
    unsigned char    modem_class;            /* reserved                  */
    unsigned char    user_def_parm_1;        /* User-defined parameter 1 */
    unsigned char    user_def_parm_2;        /* User-defined parameter 2 */
    unsigned char    user_def_parm_3;        /* User-defined parameter 3 */
} TG_DEFINED_CHARS;

typedef struct ls_stats
{
    AP_UINT32        in_xid_bytes;           /* number of XID bytes received */
    AP_UINT32        in_msg_bytes;           /* number of message bytes received */
    AP_UINT32        in_xid_frames;          /* number of XID frames received */
    AP_UINT32        in_msg_frames;          /* number of message frames received */
    AP_UINT32        out_xid_bytes;          /* number of XID bytes sent */
    AP_UINT32        out_msg_bytes;          /* number of message bytes sent */
    AP_UINT32        out_xid_frames;         /* number of XID frames sent */
    AP_UINT32        out_msg_frames;         /* number of message frames sent */
    AP_UINT32        in_invalid_sna_frames; /* number of invalid frames */
}

```

```

AP_UINT32      in_session_control_frames; /* received */
AP_UINT32      out_session_control_frames; /* number of control frames */
AP_UINT32      echo_rsps;                /* received */
AP_UINT32      current_delay;             /* number of control frames */
AP_UINT32      max_delay;                 /* sent */
AP_UINT32      min_delay;                 /* reserved */
AP_UINT32      max_delay_time;            /* reserved */
AP_UINT32      good_xids;                  /* reserved */
AP_UINT32      bad_xids;                  /* reserved */
AP_UINT32      /* successful XID on LS count */
AP_UINT32      /* unsuccessful XID on LS count */
} LS_STATS;

```

リンク固有のデータについて詳しくは、135 ページの『DEFINE_LS』を参照してください。このデータのデータ構造体は `ls_def_data` 構造体に続きますが、スペースが入るため、4 バイト目の境界から始まります。

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_LS

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

LS の最大数 (その LS のデータが戻されます)。ある範囲でなく、特定の LS に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

ls_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

ls_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入力する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

ls_name

リンク・ステーション名。これは 8 バイトからなる ASCII スtring で、名前が 8 文字に満たない場合、右側にスペースを入れます。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

port_name

ポート名フィルター。特定のポートに関連した LS のみにに関する情報を戻すには、ポートの名前を指定します。これは 8 バイトからなる ASCII スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。ポート名で選別しないですべての LS に関する情報を戻すには、このパラメーターの 8 桁の 2 進ゼロに設定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

ls_summary.overlay_size

戻された *ls_summary* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *ls_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

ls_summary.ls_name

リンク・ステーション名。これは 8 バイトからなる ASCII スtringで、名前が 8 文字に満たない場合、右側にスペースを入れます。

ls_summary.description

LS を記述したヌルで終了するテキスト・String (その LS の定義で指定したもの)。

ls_summary.dlc_type

DLC のタイプ。次のいずれかです。

AP_SDLC

SDLC

AP_X25 QLLC

AP_TR トークンリング

AP_ETHERNET

イーサネット

AP_MPC マルチパス・チャンネル (MPC)、System z 上の Communications Server for Linux のみ

AP_IP Enterprise Extender (HPR/IP)

ls_summary.state

リンク・ステーションの状態。次のいずれかです。

AP_ACTIVE

LS は活動状態です。

AP_NOT_ACTIVE

LS は活動状態ではありません。

AP_PENDING_ACTIVE

LS は活動化中です。

AP_PENDING_INACTIVE

LS は活動化されていません。

AP_PENDING_ACTIVE_BY_LR

LS に障害が起こり (または失敗した LS を活動化しようとして)、Communications Server for Linux が LS を再活動化しようとしています。

ls_summary.act_sess_count

リンクを使用している活動セッションの合計数 (エンドポイント・セッションと中間セッションの両方)。

ls_summary.det_adj_cp_name

隣接制御点の完全修飾名。この名前は 17 バイトの EBCDIC Stringで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String文字からなるネットワーク名で構成されます。

通常、この名前は活動化中に決定され、LS が活動状態でない場合はヌルです。ただし、バックレベル LEN ノードに対する LS (DEFINE_LS の

adj_cp_type パラメーターで定義された LS) の場合は、この名前は LS 定義から取られ、活動化中には決定されません。

ls_summary.det_adj_cp_type

隣接ノードのタイプ。次のいずれかです。

AP_APPN_NODE

ノード・タイプが不明、または LS が活動状態ではありません。

AP_END_NODE

エンド・ノード、ローカル・ノードの全体像のエンド・ノードとして機能する分岐ネットワーク・ノード、または上位レベル LEN ノード (XID3 にネットワーク名 CV が含まれているノード)。

AP_NETWORK_NODE

ネットワーク・ノード、またはローカル・ノードの全体像のネットワーク・ノードとして機能する分岐ネットワーク・ノード。

AP_VRN 仮想経路指定ノード。

通常、ノード・タイプは活動化中に決定され、LS が活動状態でない場合はヌルです。ただし、バックレベル LEN ノードに対する LS (DEFINE_LS の *adj_cp_type* パラメーターで定義された LS) の場合は、ノード・タイプは LS 定義から取られ、活動化中には決定されません。

ls_summary.port_name

このリンク・ステーションに関連したポートの名前。これは 8 バイトからなる ASCII スtring で、名前が 8 文字に満たない場合、右側にスペースを入れます。

ls_summary.adj_cp_name

隣接制御点の完全修飾名。暗黙的なリンクの場合、このパラメーターはヌルです。この名前は 17 バイトの EBCDIC String で、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String 文字からなるネットワーク名で構成されます。

ls_summary.adj_cp_type

リンクの活動化中に決定される隣接ノードのタイプ。次のいずれかです。

AP_APPN_NODE

ノード・タイプが不明、または LS が活動状態ではありません。

AP_END_NODE

エンド・ノード、ローカル・ノードの全体像のエンド・ノードとして機能する分岐ネットワーク・ノード、または上位レベル LEN ノード (XID3 にネットワーク名 CV が含まれているノード)。

AP_NETWORK_NODE

ネットワーク・ノード、またはローカル・ノードの全体像のネットワーク・ノードとして機能する分岐ネットワーク・ノード。

AP_BACK_LEVEL_LEN_NODE

バックレベル LEN ノード (XID3 にネットワーク名 CV が含まれていないノード)。

AP_HOST_XID3

ホスト・ノード。Communications Server for Linux はノードからのポーリング XID にフォーマット 3 XID で応答します。

AP_HOST_XID0

ホスト・ノード。Communications Server for Linux はノードからのポーリング XID にフォーマット 0 XID で応答します。

AP_DSPU_XID

ダウンストリーム PU。Communications Server for Linux はリンクの活動化に XID 交換を含めます。 *dspu_name* フィールドと *dspu_services* フィールドも設定する必要があります。

AP_DSPU_NOXID

ダウンストリーム PU。Communications Server for Linux はリンクの活動化に XID 交換を含めません。 *dspu_name* フィールドと *dspu_services* フィールドも設定する必要があります。

AP_VRN 仮想経路指定ノード。

ls_detail.overlay_size

戻された *ls_detail* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *ls_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

ls_detail.ls_name

リンク・ステーション名。これは 8 バイトからなる ASCII スtring で、名前が 8 文字に満たない場合、右側にスペースを入れます。

ls_detail.det_data.act_sess_count

リンクを使用している活動セッションの合計数 (エンドポイント・セッションと中間セッションの両方)。

ls_detail.det_data.dlc_type

DLC のタイプ。次のいずれかです。

AP_SDLC

SDLC

AP_X25 QLLC

AP_TR トークンリング

AP_ETHERNET

イーサネット

AP_MPC マルチパス・チャネル (MPC)、System z 上の Communications Server for Linux のみ

AP_IP Enterprise Extender (HPR/IP)

ls_detail.det_data.state

リンク・ステーションの状態。次のいずれかです。

AP_ACTIVE

LS は活動状態です。

AP_NOT_ACTIVE

LS は活動状態ではありません。

AP_PENDING_ACTIVE

LS は活動化中です。

AP_PENDING_INACTIVE

LS は活動化されていません。

AP_PENDING_ACTIVE_BY_LR

LS に障害が起こり (または失敗した LS を活動化しようとして)、
Communications Server for Linux が LS を再活動化しようとしています。

ls_detail.det_data.sub_state

このフィールドは、このリンク・ステーションの状態に関する詳細情報を提供します。値は次のいずれかです。

AP_SENT_CONNECT_OUT

AP_PENDING_XID_EXCHANGE

AP_SENT_ACTIVATE_AS

AP_SENT_SET_MODE

AP_ACTIVE

AP_SENT_DEACTIVATE_AS_ORDERLY

AP_SENT_DISCONNECT

AP_WAITING_STATS

AP_RESET

ls_detail.det_data.det_adj_cp_name

隣接制御点の完全修飾名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

通常、この名前は活動化中に決定され、LS が活動状態でない場合はヌルです。ただし、バックレベル LEN ノードに対する LS (DEFINE_LS の *adj_cp_type* パラメーターで定義された LS) の場合は、この名前は LS 定義から取られ、活動化中には決定されません。

ls_detail.det_data.det_adj_cp_type

隣接ノードのタイプ。次のいずれかです。

AP_END_NODE

エンド・ノード、ローカル・ノードの全体像のエンド・ノードとして機能する分岐ネットワーク・ノード。

AP_NETWORK_NODE

ネットワーク・ノード、またはローカル・ノードの全体像のネットワーク・ノードとして機能する分岐ネットワーク・ノード。

AP_LEARN_NODE

ノード・タイプは不明です。

通常、ノード・タイプは活動化中に決定され、LS が活動状態でない場合はヌルです。ただし、バックレベル LEN ノードに対する LS (DEFINE_LS の *adj_cp_type* パラメーターで定義された LS) の場合は、ノード・タイプは LS 定義から取られ、活動化中には決定されません。

ls_detail.det_data.dlc_name

DLC の名前。これは 8 バイトからなる ASCII スtring で、名前が 8 文字に満たない場合、右側にスペースを入れます。

ls_detail.det_data.dynamic

リンクが動的に定義されているかどうかを指定します。値は次のいずれかです。

AP_YES リンクは (隣接ノードからの接続要求に応答するように、または接続ネットワークを介して別のノードに動的に接続するように) 動的に定義されています。

AP_NO リンクは (DEFINE_LS により) 明示的に定義されています。

ls_detail.det_data.migration

隣接ノードが移行レベルのノード (ロー・エンタリー・ネットワーキングまたは LEN ノードなど) であるか、完全な APPN ネットワーク・ノードまたはエンド・ノードであるかを指定します。値は次のいずれかです。

AP_YES 隣接ノードは移行レベルのノードです。

AP_NO 隣接ノードはネットワーク・ノードまたはエンド・ノードです。

AP_UNKNOWN

隣接ノード・レベルは不明です。

ls_detail.det_data.tg_num

TG に関連した番号。

ls_detail.det_data.ls_stats.in_xid_bytes

このリンク・ステーションで受信した XID (交換識別) バイトの合計数。

ls_detail.det_data.ls_stats.in_msg_bytes

このリンク・ステーションで受信したデータ・バイトの合計数。

ls_detail.det_data.ls_stats.in_xid_frames

このリンク・ステーションで受信した XID (交換識別) フレームの合計数。

ls_detail.det_data.ls_stats.in_msg_frames

このリンク・ステーションで受信したデータ・フレームの合計数。

ls_detail.det_data.ls_stats.out_xid_bytes

このリンク・ステーションで送信した XID (交換識別) バイトの合計数。

ls_detail.det_data.ls_stats.out_msg_bytes

このリンク・ステーションで送信したデータ・バイトの合計数。

ls_detail.det_data.ls_stats.out_xid_frames

このリンク・ステーションで送信した XID (交換識別) フレームの合計数。

ls_detail.det_data.ls_stats.out_msg_frames

このリンク・ステーションで送信したデータ・フレームの合計数。

ls_detail.det_data.ls_stats.in_invalid_sna_frames

このリンク・ステーションで受信した無効な SNA フレームの合計数。

ls_detail.det_data.ls_stats.in_session_control_frames

このリンク・ステーションで受信したセッション制御フレームの合計数。

ls_detail.det_data.ls_stats.out_session_control_frames

このリンク・ステーションで送信したセッション制御フレームの合計数。

ls_detail.det_data.ls_stats.good_xids

このリンク・ステーションで始動したあとに発生して成功した XID 交換の合計数。

ls_detail.det_data.ls_stats.bad_xids

このリンク・ステーションで始動したあとに発生して失敗した XID 交換の合計数。

ls_detail.det_data.start_time

システムが始動してから、リンク・ステーションが最後に活動化される (つまり、モード設定コマンドの完了) までの時間 (0.01 秒単位)。

ls_detail.det_data.stop_time

システムが始動してから、リンク・ステーションが最後に非活動化されるまでの時間 (0.01 秒単位)。

ls_detail.det_data.up_time

システムが始動してから、このリンク・ステーションが活動状態である合計時間 (0.01 秒単位)。

ls_detail.det_data.current_state_time

システムが始動してから、このリンク・ステーションが現在の状態である合計時間 (0.01 秒単位)。

ls_detail.det_data.deact_cause

リンク・ステーションが最後に活動化されなかった原因。値は次のいずれかです。

AP_NONE

リンク・ステーションは活動化されています。

AP_DEACT_OPER_ORDERLY

リンク・ステーションは、オペレーターが通常 STOP コマンドを入力したため、活動化されませんでした。

AP_DEACT_OPER_IMMEDIATE

リンク・ステーションは、オペレーターが即時 STOP コマンドを入力したため、活動化されませんでした。

AP_DEACT_AUTOMATIC

リンク・ステーションは、このリンク・ステーションを使用するセッションがなくなったなどの理由で自動的に活動化されませんでした。

AP_DEACT_FAILURE

リンク・ステーションは、障害のために活動化されませんでした。

ls_detail.det_data.hpr_support

ローカル・ノードおよび隣接ノードの機能を考慮した、この伝送グループ (TG) でサポートされている高性能経路指定 (High Performance Routing: HPR) のレベル。値は次のいずれかです。

AP_NONE

この TG は HPR プロトコルをサポートしていません。

AP_BASE

この TG は基本レベルの HPR をサポートしています。

AP_RTP この TG は高速トランスポート・プロトコル (Rapid Transport Protocol: RTP) をサポートしています。

ls_detail.det_data.anr_label

ローカル・リンクに割り振られた HPR 自動ネットワーク経路指定 (automatic network routing: ANR) ラベル。

ls_detail.det_data.hpr_link_lvl_error

SDLC の場合、このパラメーターは予約済みです。

その他のポート・タイプの場合、このパラメーターでリンク上の HPR トラフィックにリンク・レベルのエラー回復を使用するかどうかを指定します。

ls_detail.det_data.auto_act

リンクで現在リモート活動化またはオンデマンド活動化が許可されているかどうかを指定します。このパラメーターは、AP_NONE (どちらも許されていない場合) に設定するか、次の値の 1 つまたは両方 (論理 OR を使用して組み合わせる) に設定します。

AP_AUTO_ACT

リンクは、セッションが要求するときにローカル・ノードによりオンデマンドで活動化できます。

AP_REMOTE_ACT

リンクは、リモート・ノードにより活動化できます。

ls_detail.det_data.ls_role

このリンクの LS の役割。これは通常、LS を所有しているポートの定義から (または、このリンクにより、ポート定義での LS の役割が上書きされる場合は LS の定義から) とられます。ただし、LS の役割が折衝可能として定義されている場合は、LS が活動状態であるときに 1 次役割または 2 次役割のいずれかとして折衝されます。この verb を使用して活動状態である LS を照会する場合、LS の役割は現在使用している折衝された役割であり、定義された役割ではありません。値は次のいずれかです。

AP_LS_PRI

1 次。

AP_LS_SEC

2 次。

AP_LS_NEG

折衝可能。

ls_detail.det_data.ls_type

このリンクが定義された方法または検出された方法を指定します。値は次のいずれかです。

AP_LS_DEFINED

リンク・ステーションは、Communications Server for Linux 管理プログラムによって、明示的に定義されました。

AP_LS_DYNAMIC

リンク・ステーションは、ローカル・ノードが接続ネットワークを通じて他のノードへ接続したときに作成されました。

AP_LS_TEMPORARY

リンク・ステーションは着呼を処理するために一時的に作成されましたが、まだ活動状態になっていません。

AP_LS_IMPLICIT

リンク・ステーションは、定義済みのリンク・ステーションと一致しなかった着呼を Communications Server for Linux が受け取ったときに、暗黙的に定義されました。

AP_LS_DLUS_DEFINED

リンク・ステーションは、DLUR サービスを受けるダウンストリーム PU へのダイナミック・リンク・ステーションであり、ローカル・ノードが DLUS から ACTPU を受信したときに定義されました。

ls_detail.det_data.node_id

XID 交換中に隣接ノードから受信されたノード ID。これは 4 バイトの 16 進数ストリングです。

ls_detail.det_data.active_isr_count

リンクを使用している活動状態である中間セッションの数。

ls_detail.det_data.active_lu_sess_count

このリンクを使用している活動 LU-LU セッションのカウント。

ls_detail.det_data.active_sscp_sess_count

このリンクを使用している活動 PU-SSCP セッションのカウント。

ls_detail.det_data.reverse_anr_label

このリンク・ステーションの反転自動ネットワーク経路指定 (Reverse Automatic Network Routing: ANR) ラベル。

SDLC の場合:

ls_detail.det_data.local_address

このリンク・ステーションのローカル・アドレス。

QLLC の場合:

ls_detail.det_data.local_address

このリンク・ステーションのローカル・アドレス。

トークンリング、イーサネットの場合:

ls_detail.det_data.local_address.mac_address

ローカル・リンク・ステーションの MAC アドレス。

ls_detail.det_data.local_address.lsap_address

ローカル・リンク・ステーションのローカル SAP アドレス。

Enterprise Extender の場合:

ls_detail.det_data.local_address.ip_address_info.lsap

Enterprise Extender の場合: ポートのローカル SAP アドレス。 0x04 ~ 0xEC の範囲で 0x04 の倍数を指定してください。通常の場合は 0x04 ですが、VTAM は一部の環境では 0x08 を使用できます。

同じ TCP/IP インターフェースで異なる LSAP アドレスを指定して複数のポートを使用する必要がある場合は、複数の Enterprise Extender DLC を作成する必要があり、そのために、それぞれの DLC には同じ *if_name* (ただし、異なる LSAP アドレス) を指定して、個別の Enterprise Extender ポートを作成します。

ls_detail.det_data.local_address.ip_address_info.version

Enterprise Extender の場合: 以下のフィールドが IPv4 または IPv6 アドレスを表すかどうかを指定します。値は次のいずれかです。

IP_VERSION_4_HOSTNAME

address フィールドには、IPv4 アドレス、あるいは IPv4 アドレスに解決するホスト名または別名を指定します。

IP_VERSION_6_HOSTNAME

address フィールドには、IPv6 アドレス、あるいは IPv6 アドレスに解決するホスト名または別名を指定します。

ls_detail.det_data.local_address.ip_address_info.address

Enterprise Extender の場合: ポートの IP アドレス。指定できるのは、以下のいずれかです。

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

マルチパス・チャネル (MPC)、System z 上の Communications Server for Linux のみ:

ls_detail.det_data.local_address.address

このパラメーターは予約済みです。

ls_detail.det_data.max_send_btu_size

隣接ノードとの折衝で決定された、このリンクへ送信できる BTU の最大サイズ。リンクがまだ活動化されていない場合は、ゼロの値が戻されます。

ls_detail.det_data.brnn_link_type

このパラメーターは、ローカル・ノードが分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

このリンクの分岐リンク・タイプを指定します。値は次のいずれかです。

AP_UPLINK

リンクはアップリンクです。

AP_DOWNLINK

リンクはダウンリンクです。

AP_OTHERLINK

VRN へのリンクです。

AP_UNKNOWN_LINK_TYPE

分岐リンク・タイプは不明です。

AP_BRNN_NOT_SUPPORTED

リンクは PU 2.0 トラフィックのみをサポートします。

ls_detail.det_data.adj_cp_is_brnn

隣接ノードが分岐ネットワーク・ノードであるかどうかを指定します。値は次のいずれかです。

AP_YES 隣接ノードは分岐ネットワーク・ノードです。

AP_NO 隣接ノードは分岐ネットワーク・ノードではありません。

AP_UNKNOWN

隣接ノードのタイプは不明です。

ls_detail.def_data.description

LS を記述したヌルで終了するテキスト・ストリング (その LS の定義で指定したもの)。

ls_detail.def_data.initially_active

この LS をノードの始動時に自動的に起動するかどうかを指定します。値は次のいずれかです。

AP_YES LS はノードの起動時に自動的に始動します。

AP_NO LS は自動的に始動しません。手動で始動させる必要があります。

ls_detail.def_data.react_timer

障害が起きた LS を再活動化するための再活動化タイマー。下の *react_timer_retry* パラメーターがゼロ以外の場合、このパラメーターは、障害が起きた LS の活動化を Communications Server for Linux が再試行するように指定するため、再試行の時間間隔を秒単位で指定します。LS で障害が起きた場合、または障害が起きた LS の再活動化の試行が失敗した場合、Communications Server for Linux は、指定した時間まで待機してから活動化を再試行します。*react_timer_retry* がゼロの場合は、このパラメーターは無視されます。

ls_detail.def_data.react_timer_retry

障害が起きた LS を再活動化するための再試行カウント。このパラメーターは、LS の使用中に障害が起きた (または、LS 開始の試行が失敗した) ときに、Communications Server for Linux が LS の再活動化を試行するかどうかを指定する場合に使用されます。

ゼロは Communications Server for Linux が LS を再活動化しないことを示します。ゼロ以外の値は再試行を行う回数を指定します。65,535 の値は、LS が活動化されるまで Communications Server for Linux が無制限に再試行することを示します。

Communications Server for Linux は、再試行から次の再試行までの間、上記の *react_timer* パラメーターで指定した時間を待機します。LS が正常に再活動化されないで再試行カウントに達した場合、または Communications Server for Linux が活動化を再試行しているときに STOP_LS が発行された場合、再試行はそれ以上実行されず、LS に対して START_LS が発行されない限り、LS は非活動状態のままになります。

auto_act_supp パラメーターを AP_YES に設定した場合、再活動化タイマーの各フィールドは無視されます。リンクに障害が起きた場合、Communications Server for Linux は、そのセッションを使用していたユーザー・アプリケーションがセッションの再起動を試行するまで、そのリンクの再活動化を試行しません。

ls_detail.def_data.restart_on_normal_deact

LS がリモート・システムで正常に非活動化された場合に、Communications Server for Linux に LS の再活動化を試行させるかどうかを指定します。値は次のいずれかです。

AP_YES リモート・システムが LS を正常に非活動化した場合、Communications Server for Linux は、障害のあった LS を再活動化する場合と同じ再試行タイマー値およびカウント値 (上記の *react_timer* および *react_timer_retry* パラメーター) を使用して、LS の再活動化を試行します。

AP_NO リモート・システムが LS を正常に非活動化した場合、Communications Server for Linux は LS の再活動化を試行しません。

LS が (*adj_cp_type* パラメーターで指定した) ホスト・リンクである場合、あるいはノードの始動時に自動的に開始された (*initially_active* パラメーターを AP_YES に設定) 場合、このパラメーターは無視されます。リモート・システムが通常どおり LS を非活動化すると、Communications Server for Linux は常に LS の再活動化を試みます (*react_timer_retry* がゼロでない場合)。

ls_detail.def_data.port_name

このリンク・ステーションに関連したポートの名前。これは 8 バイトからなる ASCII スtring で、名前が 8 文字に満たない場合、右側にスペースを入れます。リンクが VRN の場合は、このフィールドは VRN と接続するために使用された実際のポートの名前 (DEFINE_CN verb で指定されたもの) を指定します。

ls_detail.def_data.adj_cp_name

隣接制御点の完全修飾名。このパラメーターは、*adj_cp_type* で隣接ノードに APPN ノードまたはバックレベル LEN ノードが指定されている場合のみ使用されます。

この名前は 17 バイトの EBCDIC String で、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String 文字からなるネ

ットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A スtring文字からなるネットワーク名で構成されます。

ls_detail.def_data.adj_cp_type

隣接ノード・タイプ。次のいずれかです。

AP_APPN_NODE

APPN 可能ノード。このノード・タイプは XID 交換中に認識されます。

AP_NETWORK_NODE

ネットワーク・ノード、またはローカル・ノードの全体像のネットワーク・ノードとして機能する分岐ネットワーク・ノード。

AP_END_NODE

エンド・ノード、ローカル・ノードの全体像のエンド・ノードとして機能する分岐ネットワーク・ノード、または上位レベル LEN ノード (XID3 にネットワーク名 CV が含まれているノード)。

AP_BACK_LEVEL_LEN_NODE

バックレベル LEN ノード (XID3 にネットワーク名 CV が含まれていないノード)。

AP_HOST_XID3

ホスト・ノード。 Communications Server for Linux はノードからのポーリング XID にフォーマット 3 XID で応答します。

AP_HOST_XID0

ホスト・ノード。 Communications Server for Linux はノードからのポーリング XID にフォーマット 0 XID で応答します。

AP_DSPU_XID

ダウンストリーム PU。 Communications Server for Linux はリンクの活動化に XID 交換を含めます。

AP_DSPU_NOXID

ダウンストリーム PU。 Communications Server for Linux はリンクの活動化に XID 交換を含めません。

SDLC の場合:

ls_detail.def_data.dest_address

2 次リンク・ステーションのアドレス。

このパラメーターの値は、この LS を所有するポートの構成方法で決まります。その方法は以下のとおりです。

- ポートが着呼だけに使用される場合 (DEFINE_PORT の *out_link_act_lim* が 0)、このパラメーターは予約済みです。
- このポートが交換用の 1 次であり、発呼用を使用される場合 (DEFINE_PORT 上の *port_type* は AP_SWITCHED、*ls_role* は AP_LS_PRI、および *out_link_act_lim* はゼロ以外の値)、このパラメーターには、2 次ステーションにどのようなアドレスが構成されていても受け入れるようにするための 0xFF を設定するか、または (2 次ステーションに構成されている値と一致しなければならない) 0x01 ~ 0xFE の範囲の 1 バイト値に設定します。

- その他のポート構成には、リンク・ステーションを識別するための 0x01 ~ 0xFE の範囲の 1 バイト値に設定します。ポートが 1 次のマルチドロップの場合 (DEFINE_PORT 上の *ls_role* が AP_LS_PRI、*tot_link_act_lim* が 1 より大きい)、このアドレスはポート上の各アドレスごとに異ならなければなりません。

QLLC の場合:

ls_detail.def_data.dest_address

隣接ノード上でのリンク・ステーションの宛先アドレス。このパラメータは SVC 発呼だけに使用されます (リンク固有データの *vc_type* パラメータ、および DEFINE_PORT 上のリンク活動化限度パラメータによって定義される)。着呼または PVC に対しては無視されます。

トークンリング、イーサネットの場合:

ls_detail.def_data.dest_address.mac_address

隣接ノード上でのリンク・ステーションの MAC アドレス。

このパラメータがヌルである場合は、LS は非選択 listen LS (着呼にのみ使用できるが、そこで定義済みの LU を持ち、従属型 LU トラフィックをサポートすることが可能) です。LS は、すべてのリモート・リンク・ステーションからの着呼を受信するために使用できますが、発呼には使用できません。

ローカル・ノードと隣接ノードが、ブリッジで接続した異なるタイプ (一方がトークンリングで、他方がイーサネット) の LAN 上にあるときは、ほとんどの場合、MAC アドレス内のバイトのビット順序を逆にする必要があります。詳しくは、163 ページの『MAC アドレス内のビット順序』を参照してください。2 つのノードが同じ LAN にある場合、またはブリッジで接続された同じタイプの複数の LAN にある場合は、変更する必要はありません。

ls_detail.def_data.dest_address.lsap_address

隣接ノード上でのリンク・ステーションのローカル SAP アドレス。

マルチパス・チャネル (MPC)、System z 上の Communications Server for Linux のみ:

def_data.dest_address.address

このパラメータは予約済みです。

Enterprise Extender (HPR/IP) の場合:

ls_detail.def_data.dest_address.ip_address_info.lsap

隣接ノード上でのリンク・ステーションのローカル SAP アドレス。0x04 ~ 0xEC の範囲で 0x04 の倍数を指定してください。通常の値は 0x04 ですが、VTAM は一部の環境では 0x08 を使用できます。

ls_detail.def_data.dest_address.ip_address_info.version

以下のフィールドが IPv4 または IPv6 アドレスを表すかどうかを指定します。値は次のいずれかです。

IP_VERSION_4_HOSTNAME

address フィールドには、IPv4 アドレス、あるいは IPv4 アドレスに解決するホスト名または別名を指定します。

IP_VERSION_6_HOSTNAME

address フィールドには、IPv6 アドレス、あるいは IPv6 アドレスに解決するホスト名または別名を指定します。

ls_detail.def_data.dest_address.ip_address_info.address

隣接ノード上でのリンク・ステーションの IP アドレス。指定できるのは、以下のいずれかです。

- IPv4 ドット 10 進アドレス (たとえば、193.1.11.100)
- IPv6 コロン 16 進アドレス (たとえば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)

すべてのリンク・タイプの場合:

ls_detail.def_data.auto_act_supp

セッションで必要なときに、リンクを自動的に活動化できるかどうかを指定します。値は次のいずれかです。

AP_YES リンクは自動的に活動化できます。

AP_NO リンクは自動的に活動化できません。

ls_detail.def_data.tg_number

リンクの活動化時にリンクを表すために使用する事前に割り当てられた TG 番号。このパラメーターは、隣接ノードが APPN ノードの場合 (*adj_cp_type* が AP_NETWORK_NODE または AP_END_NODE の場合) のみ使用され、それ以外の場合は無視されます。ゼロは、TG 番号を事前に割り当てないで、リンクの活動化時に折衝することを示します。

ls_detail.def_data.limited_resource

リンクを使用するセッションがないとき、このリンク・ステーションを非活動化するかどうかを指定します。値は次のいずれかです。

AP_NO リンクは限定リソースでないため、自動的に非活動化されません。

AP_NO_SESSIONS

リンクは限定リソースであるため、それを使用する活動状態であるセッションがなくなると、自動的に非活動化されます。

AP_INACTIVITY

リンクは限定リソースであるため、それを使用する活動状態であるセッションがなくなると、*link_deact_timer* フィールドで指定した時間内にリンクを流れるデータがないと、自動的に非活動化されます。

ls_detail.def_data.solicit_sscp_sessions

隣接ノードに、SSCP とローカル CP および従属 LU 間のセッションを開始するよう要求するかどうかを指定します。このパラメーターは、隣接ノードが APPN ノードの場合 (*adj_cp_type* が AP_NETWORK_NODE または AP_END_NODE の場合) のみ使用され、それ以外の場合は無視されます。隣接

ノードがホストである場合 (*adj_cp_type* が AP_HOST_XID3 または AP_HOST_XID0 のいずれかである場合)、Communications Server for Linux は常にホストに SSCP セッションを開始するよう要求します。

値は次のいずれかです。

AP_YES 隣接ノードに SSCP セッションの開始を要求します。

AP_NO 隣接ノードに SSCP セッションの開始を要求しません。

ls_detail.def_data.pu_name

このリンクを使用するローカル PU の名前。このパラメーターは、*adj_cp_type* が AP_HOST_XID3 または AP_HOST_XID0 に設定されている場合、または *solicit_sscp_sessions* が AP_YES に設定されている場合のみ使用され、それ以外の場合は予約済みです。

PU 名は、8 バイトからなるタイプ A の EBCDIC 英数字ストリング (英字で始まる) で、右側に EBCDIC スペースを入れます。

ls_detail.def_data.disable_remote_act

LS をリモート・ノードから活動化できるかどうかを指定します。値は次のいずれかです。

AP_YES LS は、ローカル・ノードのみから活動化できます。LS をリモート・ノードが活動化しようとした場合、Communications Server for Linux はその活動化をリジェクトします。

AP_NO LS は、リモート・ノードから活動化できます。

ls_detail.def_data.dspu_services

ローカル・ノードがこのリンクを介してダウンストリーム PU に提供するサービスを指定します。このパラメーターは、隣接ノードがダウンストリーム PU であるか、*solicit_sscp_sessions* を AP_NO に設定した APPN ノードである場合のみ使用されます。それ以外の場合は予約済みです。値は次のいずれかです。

AP_PU_CONCENTRATION

ローカル・ノードは、ダウンストリーム PU に SNA ゲートウェイを提供します。

AP_DLUR

ローカル・ノードは、ダウンストリーム PU に DLUR サービスを提供します。

AP_NONE

ローカル・ノードは、このダウンストリーム PU にサービスを提供しません。

ls_detail.def_data.dspu_name

ダウンストリーム PU の名前。このパラメーターは、*solicit_sscp_sessions* が AP_NO に設定され、*dspu_services* が AP_PU_CONCENTRATION または AP_DLUR に設定されている場合のみ必須で、それ以外の場合は予約済みです。この名前は 8 バイトからなるタイプ A の EBCDIC ストリング (英字で始まる) で、右側に EBCDIC スペースを入れます。

ls_detail.def_data.dlus_name

ダウンストリーム・ノードに対してリンクを活動化するときに、DLUR が

SSCP サービスを請求する請求元の DLUS ノードの名前。 *dspu_services* が AP_DLUR に設定されていない場合、このフィールドは予約済みです。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringです。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

17 桁の 2 進ゼロで構成されるStringは、DEFINE_DLUR_DEFAULTS verb を使用して定義されたグローバル・デフォルト DLUS を示します。このパラメーターをゼロに設定したときにグローバル・デフォルト DLUS がないと、DLUR はリンクを活動化するときに SSCP 接続を開始しません。

ls_detail.def_data.bkup_dlus_name

ダウンストリーム・ノードに対してリンクを活動化するときに、DLUR が SSCP サービスを請求する請求元の DLUS ノードの名前。 *dspu_services* が AP_DLUR に設定されていない場合、このフィールドは予約済みです。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringです。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

17 桁の 2 進ゼロで構成されるStringは、DEFINE_DLUR_DEFAULTS verb を使用して定義されたグローバル・バックアップ・デフォルト DLUS を示します。

ls_detail.def_data.hpr_supported

このリンクで HPR がサポートされているかどうかを指定します。値は次のいずれかです。

AP_YES このリンクでは HPR がサポートされています。

AP_NO このリンクでは HPR はサポートされていません。

ls_detail.def_data.hpr_link_lvl_error

SDLC の場合、このパラメーターは予約済みです。

その他のポート・タイプの場合、このパラメーターでリンク上の HPR トラフィックにリンク・レベルのエラー回復をサポートするかどうかを指定します。

ls_detail.def_data.hpr_supported パラメーターが AP_NO に設定されている場合、このパラメーターは予約済みです。以下の値を設定できます。

AP_YES このリンクでは HPR のリンク・レベル・エラー回復タワーがサポートされています。

AP_NO このリンクでは HPR のリンク・レベル・エラー回復タワーはサポートされていません。

ls_detail.def_data.link_deact_timer

限定リソース・リンク非活動化タイマー (単位、最小 5)。 *limited_resource* が AP_INACTIVITY に設定されている場合は、このパラメーターで指定した時間内にデータ・フローがなかった場合にリンクが非活動化されます。ゼロ

の値は タイムアウトがないことを示し (*limited_resource* が AP_NO に設定された場合と同様に、リンクは非活動化されません)、1 ~ 4 の範囲の値は 5 と解釈されます。

ls_detail.def_data.default_nn_server

エンド・ノード。ローカル・ノードのネットワーク・ノード・サーバーとして機能できる、ネットワーク・ノードとの CP-CP セッションをサポートするリンクであるかどうかを指定します。ローカル・ノードにネットワーク・ノード・サーバーとの CP-CP セッションがなく、それを確立する必要がある場合は、ローカル・ノードはその定義済み LS でこのパラメーターを検査し、活動化に適した LS を検索します。これにより、どの 隣接 NN が NN サーバーとして機能するのに適しているかを指定できます (たとえば、コストのかかるリンクまたは低速のリンクを介してアクセスする NN を使用しないようにします)。

値は次のいずれかです。

AP_YES このリンクは、ローカル・ノードの NN サーバーとして機能できるネットワーク・ノードとの CP-CP セッションをサポートします。ローカル・ノードは、NN サーバーと接続する必要がある場合にこのリンクを自動的に活動化できます。

AP_NO ネットワーク・ノード・サーバーと接続しようとしても、このリンクは自動的に活動化しません。

ローカル・ノードがエンド・ノードでない場合、このパラメーターは予約済みです。

ls_detail.def_data.ls_attributes

この配列には、次のパラメーターで記述する隣接ノードに関する詳しい情報が入ります。

ls_detail.def_data.ls_attributes[0]

ホスト・タイプ (通常は標準 SNA)。値は次のいずれかです。

AP_SNA 標準 SNA ホスト。

AP_FNA 富士通 Network Architecture (VTAM-F) ホスト。

AP_HNA 日立 Network Architecture ホスト。

ls_detail.def_data.ls_attributes[1]

バックレベル LEN ノードとのリンクのネットワーク名 CV 抑止。

adj_cp_type が AP_BACK_LEVEL_LEN_NODE に設定されている場合、このパラメーターは、LEN ノードへ送信される 3 XID フォーマットでのネットワーク名 CV の組み込みを抑止するかどうかを指定します。値は次のいずれかです。

AP_NO XID にネットワーク名 CV を組み込みます。

AP_SUPPRESS_CP_NAME

ネットワーク名 CV を組み込みません。

adj_cp_type がこれ以外の値に設定されている場合、このパラメーターは予約済みです。

ls_detail.def_data.adj_node_id

隣接ノードのノード ID。これは 4 バイトの 16 進数ストリングで、ゼロが 4 つの場合にはノード ID 検査が使用できないことを示します。

ls_detail.def_data.local_node_id

この LS の XID で送信されるノード ID。これは 4 バイトの 16 進数ストリングで、ブロック番号 (3 桁の 16 進数字) とノード番号 (5 桁の 16 進数字) で構成されます。値がすべてゼロの場合は、Communications Server for Linux が DEFINE_NODE verb で指定したノード ID を使用することを示します。

ls_detail.def_data.cp_cp_sess_support

CP-CP セッションがサポートされるかどうかを指定します。値は次のいずれかです。

AP_YES CP-CP セッションがサポートされます。

AP_NO CP-CP セッションはサポートされません。

ls_detail.def_data.use_default_tg_chars

DEFINE_PORT verb で指定したデフォルト TG 特性を使用するかどうかを指定します。値は次のいずれかです。

AP_YES デフォルト TG 特性を使用します。この verb の tg_chars 構造体は無視されます。

AP_NO この verb の tg_chars 構造体を使用します。

ls_detail.def_data.tg_chars.effect_cap

実際のビット/秒率 (回線速度)。値は 1 バイトの浮動小数点数としてエンコードされます。このときの公式は、 $0.1 \text{ mmm} * 2^{\text{eeee}}$ で、バイトのビット表現は `b'eeeeemmm'` です。300 ビット/秒の倍数で指定してください。

ls_detail.def_data.tg_chars.connect_cost

接続時間当たりのコスト。有効な値は 0 ~ 255 の範囲の整数値で、0 が接続時間当たりの最低コスト、255 が最高コストです。

ls_detail.def_data.tg_chars.byte_cost

バイト当たりのコスト。有効な値は 0 ~ 255 の範囲の整数値で、0 がバイト当たりの最低コスト、255 が最高コストです。

ls_detail.def_data.tg_chars.security

ネットワークのセキュリティー・レベル。値は次のいずれかです。

AP_SEC_NONSECURE

セキュリティーなし。

AP_SEC_PUBLIC_SWITCHED_NETWORK

公衆交換網を介してデータを伝送します。

AP_SEC_UNDERGROUND_CABLE

保護された地下ケーブルを介してデータを伝送します。

AP_SEC_SECURE_CONDUIT

ガードされていない保護されたコンジット (配管) 内の回線を介してデータを伝送します。

AP_SEC_GUARDED_CONDUIT

物理接続による盗聴から保護されたコンジット内の回線を介してデータを伝送します。

AP_SEC_ENCRYPTED

回線を介して伝送する前にデータを暗号化します。

AP_SEC_GUARDED_RADIATION

物理接続による盗聴および電波盗聴から保護された回線を介してデータを伝送します。

AP_SEC_MAXIMUM

最大のセキュリティー。

ls_detail.def_data.tg_chars.prop_delay

伝搬遅延: 信号をリンクの長さのみ伝送するのに要する時間。値は次のいずれかです。

AP_PROP_DELAY_MINIMUM

最小の伝搬遅延。

AP_PROP_DELAY_LAN

遅延は 480 マイクロ秒未満です (LAN の通常値)。

AP_PROP_DELAY_TELEPHONE

遅延は 480 ~ 49,512 マイクロ秒の範囲です (電話網の通常値)。

AP_PROP_DELAY_PKT_SWITCHED_NET

遅延は 49,512 ~ 245,760 マイクロ秒の範囲です (パケット交換ネットワークの通常値)。

AP_PROP_DELAY_SATELLITE

遅延は 245,760 マイクロ秒を超えます (サテライト・リンクの通常値)。

AP_PROP_DELAY_MAXIMUM

最大の伝搬遅延。

ls_detail.def_data.tg_chars.user_def_parm_1 ~ def_data.tg_chars.user_def_parm_3

上記のパラメーターで対応できないその他の TG 特性を含むユーザー定義パラメーター。これらのパラメーターは、それぞれ 0 ~ 255 の範囲の値に設定されます。

ls_detail.def_data.target_pacing_count

求めるペーシング・ウィンドウ・サイズを示す 1 以上 32,767 以下の数値。(現行バージョンの Communications Server for Linux では、この値は使用されません。)

ls_detail.def_data.max_send_btu_size

送信できる BTU の最大サイズ。

ls_detail.def_data.ls_role

リンク・ステーションの役割。このパラメーターは通常、この LS を所有しているポートの定義から LS の役割を取ることを指定する AP_USE_PORT_DEFAULTS に設定します。

ポート定義を上書きする特定の LS の役割で LS が定義される場合は、値は次のいずれかです。

AP_LS_PRI

1 次

AP_LS_SEC

2 次

AP_LS_NEG

折衝可能

ls_detail.def_data.max_ifrm_rcvd

肯定応答を送信する前にこのリンク・ステーションで受信できる I フレームの最大数。この値は 0 ~ 127 の範囲です。このフィールドがゼロの場合は、DEFINE_PORT からの *max_ifrm_rcvd* の値がデフォルトとして使用されます。

ls_detail.def_data.dlus_retry_timeout

ls_detail.def_data.dlus_name パラメーターおよび

ls_detail.def_data.bkup_dlus_name パラメーターで指定された DLUS との接続を 2 番目以降に試みる場合の間隔 (秒単位)。1 回目の試行と 2 回目の試行の間隔は常に 1 秒です。ゼロが戻された場合は、DEFINE_DLUR_DEFAULTS で構成されたデフォルト値が使用されます。このパラメーターは、*ls_detail.def_data.dspu_services* が AP_DLUR に設定されていない場合には無視されます。

ls_detail.def_data.dlus_retry_limit

ls_detail.def_data.dlus_name パラメーターおよび

ls_detail.def_data.bkup_dlus_name パラメーターで指定された DLUS の最初の接続に失敗したあとに再試行する最大回数。ゼロが戻された場合は、DEFINE_DLUR_DEFAULTS で構成されたデフォルト値が使用されます。0x0FFFF が戻されると、Communications Server for Linux は無期限に再試行します。このパラメーターは、*ls_detail.def_data.dspu_services* が AP_DLUR に設定されていない場合には無視されます。

def_data.conventional_lu_compression

このリンク上の LU 0 から 3 のセッションでデータ圧縮が要求されるかどうかを指定します。このパラメーターは、このリンクが LU 0 から 3 のトラフィックを伝送する場合にのみ使用されます。LU 6.2 セッションには適用されません。

値は次のいずれかです。

AP_YES データ圧縮は、ホストが要求した場合にこのリンク上の LU 0 から 3 のセッションに対して使用されます。

AP_NO データ圧縮は、このリンク上の LU 0 から 3 のセッションに対して使用されません。

ls_detail.def_data.branch_link_type

このパラメーターが適用されるのは、ローカル・ノードが分岐ネットワーク・ノードである場合のみです。ローカル・ノードが他のタイプの場合は予約済みです。

パラメーター *def_data.adj_cp_type* が AP_NETWORK_NODE、AP_END_NODE、AP_APPN_NODE、または AP_BACK_LEVEL_LEN_NODE のいずれかに設定されてい

る場合は、このパラメーターは、リンクがアップリンクであるかまたはダウンリンクであるかを定義します。値は次のいずれかです。

AP_UPLINK

リンクはアップリンクです。

AP_DOWNLINK

リンクはダウンリンクです。

ls_detail.def_data.adj_brnn_cp_support

このパラメーターが適用されるのは、ローカル・ノードが分岐ネットワーク・ノードで、隣接ノードがネットワーク・ノードの場合 (パラメーター *def_data.adj_cp_type* が AP_NETWORK_NODE に設定されているか、あるいは AP_APPN_NODE に設定され、XID 交換時に発見されたノード・タイプがネットワーク・ノードの場合) のみです。ローカル・ノードとリモート・ノードが他のタイプの場合は、予約済みです。

このパラメーターは、隣接ノードを、ローカル・ノードから見てネットワーク・ノードとして機能している分岐ネットワーク・ノードにすることができるかどうかを定義します。値は次のいずれかです。

AP_BRNN_ALLOWED

隣接ノードを分岐ネットワーク・ノードにすることができます (ただし、必須ではありません)。

AP_BRNN_REQUIRED

隣接ノードは分岐ネットワーク・ノードでなければなりません。

AP_BRNN_PROHIBITED

隣接ノードを分岐ネットワーク・ノードにすることはできません。

ls_detail.def_data.pu_can_send_dddlu_offline

ローカル PU がホストに NMVT (電源オフ) メッセージを送信するかどうかを指定します。ホスト・システムが DDDL (従属 LU の動的定義) をサポートしている場合、Communications Server for Linux は、動的に定義された LU を使用し終わったときに、NMVT (電源オフ) をホストへ送信します。これによって、ホストは、定義が不要になった際にその定義を除去することによって、リソースを節約することができます。

このパラメーターを使用するのは、これがホストへのリンクである (*solicit_sscp_sessions* が AP_YES に設定され、*dspu_services* が AP_NONE に設定されていない) 場合のみです。

値は次のいずれかです。

AP_YES ローカル PU は、NMVT (電源オフ) メッセージをホストに送信します。

AP_NO ローカル PU は、NMVT (電源オフ) メッセージをホストに送信しません。

ホストが DDDL はサポートしていても NMVT (電源オフ) メッセージをサポートしていない場合、このパラメーターは AP_NO に設定しなければなりません。

ls_detail.def_data.link_spec_data_len

初期化中に変更されないでリンク・ステーション・コンポーネントに渡され

るリンク固有のデータの長さ。このデータのデータ構造体は `ls_def_data` 構造体に続きますが、スペースが入るため、4 バイト目の境界から始まります。リンク固有のデータについて詳しくは、135 ページの『DEFINE_LS』を参照してください。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

`primary_rc`

`AP_PARAMETER_CHECK`

`secondary_rc`

値は次のいずれかです。

AP_INVALID_LINK_NAME

`list_options` パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、`AP_LIST_INCLUSIVE` に設定されていますが、`ls_name` パラメーターは無効でした。

AP_INVALID_LIST_OPTION

`list_options` パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_LS_ROUTING

`QUERY_LS_ROUTING verb` は、指定されたローカル LU のパートナー LU の中で使用して、リンク・ステーションを使用しているパートナー LU の位置に関する情報を戻します。複数のローカル LU に関する情報が要求された場合、情報は、まずローカル LU 名に基づいて順序づけられ、次に各ローカル LU 名に関連したパートナー LU 名に基づいて順序づけされます。ワイルドカード・パートナー LU 名とワイルドカードを含まないエントリーを混在させることができます。

VCB 構造体

```
typedef struct query_ls_routing
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  *buf_ptr;       /* buffer pointer              */
    AP_UINT32      buf_size;       /* buffer size                 */
    AP_UINT32      total_buf_size; /* total buffer size           */
    AP_UINT16      num_entries;    /* number of entries           */
    AP_UINT16      total_num_entries; /* total number of entries     */
}
```

QUERY_LS_ROUTING

```
    unsigned char    list_options;           /* list options          */
    unsigned char    reserv3;               /* reserved              */
    unsigned char    lu_name[8];           /* LU Name               */
    unsigned char    lu_alias[8];          /* reserved              */
    unsigned char    fq_partner_lu[17];     /* partner lu name       */
    unsigned char    wildcard_fqplu;       /* wildcard partner LU flag */
    unsigned char    reserv4[2];           /* reserved              */
} QUERY_LS_ROUTING;

typedef struct ls_routing_data
{
    AP_UINT16        overlay_size;
    unsigned char    lu_name[8];           /* local LU name         */
    unsigned char    lu_alias[8];          /* reserved              */
    unsigned char    fq_partner_lu[17];     /* partner lu            */
    unsigned char    wildcard_fqplu;       /* wildcard partner LU flag */
    unsigned char    ls_name[8];           /* link to use           */
    unsigned char    reserv3[2];           /* reserved              */
} LS_ROUTING_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_LS_ROUTING

num_entries

LS 経路指定エントリーの最大数 (その LS のデータが戻されます)。ある範囲でなく、特定の LS に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すための LS 経路指定エントリーのリスト内の開始位置。

値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

lu_name パラメーターと *fq_partner_lu* パラメーターを組み合わせ、指定したエントリーから開始します。

AP_LIST_FROM_NEXT

lu_name、*fq_partner_lu*、*wildcard_fqplu* の各パラメーターを組み合わせ、指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

lu_name

Communications Server for Linux に対して定義されているローカル LU の名前。これは 8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されません。

lu_alias

このパラメーターは予約済みです。2 進ゼロに設定してください。

fq_partner_lu

Communications Server for Linux に対して定義されているパートナー LU の完全修飾名。この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。このパラメーターは、指定したローカル LU のパートナー LU 名のリストに戻すエントリーを修飾するために使用します。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

このパラメーターはすべてゼロに設定され、*list_options* が AP_LIST_FROM_NEXT に設定されている場合、戻されたリストは *lu_name* パラメーターで指定した LU の最初のパートナー LU 名から開始されます。

wildcard_fqplu

fq_partner_lu パラメーターに、完全ワイルドカードまたは部分ワイルドカードが含まれているかどうかを示すワイルドカード・パートナー LU フラグ。このフラグは、戻された最初のレコードを識別するためのみに使用されます。ワイルドカード指定に一致するエントリーのみを戻すことを指定するためには、このフラグは使用できません。値は次のいずれかです。

AP_YES *fq_partner_lu* パラメーターには、ワイルドカード・エントリーが含まれています。

AP_NO *fq_partner_lu* パラメーターには、ワイルドカード・エントリーは含まれていません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

lu_name

ローカル LU の名前。

fq_partner_lu

パートナー LU の完全修飾名。

wildcard_fqplu

fq_partner_lu パラメーターに完全ワイルドカードまたは部分ワイルドカードが含まれているかどうかを示すフラグ。値は次のいずれかです。

AP_YES *fq_partner_lu* パラメーターには、完全ワイルドカードまたは部分ワイルドカードが含まれています。

AP_NO *fq_partner_lu* パラメーターには、完全ワイルドカードまたは部分ワイルドカードは含まれていません。

ls_name

lu_name パラメーターで指定した LU と *fq_plu_name* パラメーターで指定したパートナー LU 間のセッションに使用するリンク・ステーションの名前。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、`Communications Server for Linux` は、以下のパラメーターを戻します。

primary_rc

`AP_PARAMETER_CHECK`

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

list_options パラメーターを `AP_LIST_INCLUSIVE` に設定しましたが、*lu_name* パラメーターで指定した値は、既存の LS 経路指定データ・レコードに一致しませんでした。

AP_INVALID_PARTNER_LU_NAME

list_options パラメーターを `AP_LIST_INCLUSIVE` に設定しましたが、*fq_partner_lu* パラメーターで指定した値は、指定したパートナー LU 名の既存の LS 経路指定データ・レコードに一致しませんでした。

AP_INVALID_WILDCARD_NAME

wildcard_fqplu パラメーターを `AP_YES` に設定しましたが、*fq_partner_lu* パラメーターは有効なワイルドカード名ではありませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての `NOF verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての `NOF verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_LU_0_TO_3

`QUERY_LU_0_TO_3` は、タイプ 0、1、2、または 3 のローカル LU に関する情報を戻します。この情報は、「決定済みデータ」(実行中に動的に収集されたデータ。ノードが活動状態である場合のみ戻される) および「定義済みデータ」(アプリケーションにより `DEFINE_LU_0_TO_3` で提供されたデータ) として構造化されます。

この `verb` は、使用するオプションに応じて、特定の LU または複数の LU に関する要約情報または詳細情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_lu_0_to_3
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
}
```

```

    AP_UINT16      primary_rc;          /* primary return code      */
    AP_UINT32      secondary_rc;        /* secondary return code    */
    unsigned char  *buf_ptr;            /* pointer to buffer        */
    AP_UINT32      buf_size;            /* buffer size              */
    AP_UINT32      total_buf_size;      /* total buffer size required */
    AP_UINT16      num_entries;          /* number of entries        */
    AP_UINT16      total_num_entries;    /* total number of entries   */
    unsigned char  list_options;        /* listing options          */
    unsigned char  reserv3;             /* reserved                  */
    unsigned char  pu_name[8];          /* PU name filter           */
    unsigned char  lu_name[8];          /* LU name                   */
    unsigned char  host_attachment;     /* host attachment filter    */
} QUERY_LU_0_TO_3;

typedef struct lu_0_to_3_summary
{
    AP_UINT16      overlay_size;        /* size of returned entry   */
    unsigned char  pu_name[8];          /* PU name                   */
    unsigned char  lu_name[8];          /* LU name                   */
    unsigned char  description[32];     /* resource description      */
    unsigned char  reserv1[16];         /* reserved                   */
    unsigned char  nau_address;         /* NAU address               */
    unsigned char  lu_sscp_sess_active; /* Is LU-SSCP session active */
    unsigned char  appl_conn_active;    /* Is connection to appl active */
    unsigned char  plu_sess_active;     /* Is PLU-SLU session active */
    unsigned char  host_attachment;     /* LU's host attachment      */
} LU_0_TO_3_SUMMARY;

typedef struct lu_0_to_3_detail
{
    AP_UINT16      overlay_size;        /* size of returned entry   */
    unsigned char  lu_name[8];          /* LU name                   */
    unsigned char  reserv1[2];         /* reserved                   */
    LU_0_TO_3_DET_DATA det_data;       /* Determined data           */
    LU_0_TO_3_DEF_DATA def_data;       /* Defined data              */
} LU_0_TO_3_DETAIL;

typedef struct lu_0_to_3_det_data
{
    unsigned char  lu_sscp_sess_active; /* Is LU-SSCP session active */
    unsigned char  appl_conn_active;    /* Application is using LU    */
    unsigned char  plu_sess_active;     /* Is PLU-SLU session active */
    unsigned char  host_attachment;     /* Host attachment            */
    SESSION_STATS lu_sscp_stats;        /* reserved                   */
    SESSION_STATS plu_stats;            /* reserved                   */
    unsigned char  plu_name[8];          /* PLU name                   */
    unsigned char  session_id[8];        /* Internal ID of PLU-SLU sess */
    unsigned char  app_spec_det_data[360]; /* Application specified data */
    unsigned char  app_type;             /* Type of application using LU */
    unsigned char  sscp_id[6];           /* sscp id                    */
    unsigned char  bind_lu_type;         /* LU type from BIND message  */
    unsigned char  compression;         /* data compression level     */
    unsigned char  cryptography;        /* reserved                    */
    unsigned char  reserva[10];         /* reserved                    */
} LU_0_TO_3_DET_DATA;

typedef struct session_stats
{
    AP_UINT16      rcv_ru_size;          /* session receive RU size   */
    AP_UINT16      send_ru_size;         /* session send RU size      */
    AP_UINT16      max_send_btu_size;    /* maximum send BTU size     */
    AP_UINT16      max_rcv_btu_size;     /* maximum rcv BTU size      */
    AP_UINT16      max_send_pac_win;     /* maximum send pacing window size */
    AP_UINT16      cur_send_pac_win;     /* current send pacing window size */
    AP_UINT16      max_rcv_pac_win;     /* maximum receive pacing window size */
    AP_UINT16      cur_rcv_pac_win;     /* current receive pacing window size */
    AP_UINT32      send_data_frames;     /* number of data frames sent */
}

```

QUERY_LU_0_TO_3

```
AP_UINT32    send_fmd_data_frames; /* num fmd data frames sent */
AP_UINT32    send_data_bytes;      /* number of data bytes sent */
AP_UINT32    rcv_data_frames;      /* number of data frames received */
AP_UINT32    rcv_fmd_data_frames;  /* num fmd data frames received */
AP_UINT32    rcv_data_bytes;       /* number of data bytes received */
unsigned char sidh;                 /* session ID high byte (from LFSID) */
unsigned char sidl;                 /* session ID low byte (from LFSID) */
unsigned char odai;                 /* ODAI bit set */
unsigned char ls_name[8];           /* Link station name */
unsigned char pacing_type;         /* type of pacing in use */
} SESSION_STATS;
typedef struct lu_0_to_3_def_data
{
    unsigned char description[32];    /* resource description */
    unsigned char reserv1[16];       /* reserved */
    unsigned char nau_address;        /* LU NAU address */
    unsigned char pool_name[8];      /* LU Pool name */
    unsigned char pu_name[8];        /* PU name */
    unsigned char priority;          /* LU priority */
    unsigned char lu_model;          /* LU model (type) */
    unsigned char sscp_id[6];        /* SSCP ID */
    AP_UINT16    timeout;             /* Timeout */
    unsigned char app_spec_def_data[16]; /* application-specified data */
    unsigned char model_name[7];     /* reserved */
    unsigned char term_method;       /* session termination type */
    unsigned char disconnect_on_unbind; /* disconnect on UNBIND flag */
    unsigned char reserv3[15];       /* reserved */
} LU_0_TO_3_DEF_DATA;
```

lu_0_to_3_det_data 構造体の *app_type* パラメーターを AP_LUA_APPLICATION に設定した場合、*app_spec_det_data* フィールドには、以下の構造体が入ります。

```
typedef struct lua_session_user_info
{
    unsigned char user_ip_address[40]; /* IP address of LUA application */
    unsigned char user_host_address[256]; /* Host name of LUA application */
    unsigned char reserved[24];        /* reserved */
} SESSION_USER_INFO;
```

lu_0_to_3_det_data 構造体の *app_type* パラメーターを AP_FMI_APPLICATION に設定した場合、*app_spec_det_data* フィールドには次の構造体が入ります。

```
typedef struct session_user_info
{
    unsigned char user_name[32];      /* 3270 user name */
    unsigned char system_name[128];  /* computer name */
    AP_UINT32    user_pid;           /* process ID */
    AP_UINT32    user_type;          /* type of application using LU */
    AP_UINT32    user_uid;           /* user ID */
    AP_UINT32    user_gid;           /* group ID */
    unsigned char user_gname[32];    /* group name */
    unsigned char reserv4[32];       /* reserved */
} SESSION_USER_INFO;
```

lu_0_to_3_det_data 構造体の *app_type* パラメーターを AP_PU_CONCENTRATION に設定した場合、*app_spec_det_data* フィールドには上記の 3270 構造体と同じ構造体が入りますが、*app_type* パラメーターは AP_PU_CONCENTRATION に設定され、*user_name* パラメーターから *user_gname* パラメーターまでは *pu_conc_downstream_lu* パラメーターで置換される点が異なります。

lu_0_to_3_det_data 構造体の *app_type* パラメーターを AP_LUA_APPLICATION に設定した場合、*app_spec_det_data* フィールドには上記の 3270 構造体と同じ構造体が

入りますが、*app_type* パラメーターは *AP_LUA_APPLICATION* に設定され、*user_name* パラメーターから *user_gname* パラメーターまでは戻されない点が異なります。

lu_0_to_3_det_data 構造体の *app_type* パラメーターを *AP_TN_SERVER* に設定した場合、*app_spec_det_data* フィールドには次の構造体が入ります。

```
typedef struct tn_server_session_user_info
{
    unsigned char    user_ip_address[40];    /* user's IP address          */
    AP_UINT16        port_number;           /* TCP/IP port number        */
    AP_UINT16        cb_number;             /* reserved                   */
    AP_UINT16        cfg_default;           /* using the default record?  */
    unsigned char    cfg_address[68];       /* address from config record */
    AP_UINT16        cfg_format;            /* format of address          */
    unsigned char    tn3270_level;          /* TN3270 level used:        */
                                                /* AP_LEVEL_TN3270           */
                                                /* AP_LEVEL_TN3270E          */
    unsigned char    lu_select;             /* method of LU selection:    */
                                                /* AP_GENERIC_LU             */
                                                /* AP_SPECIFIC_LU           */
                                                /* AP_ASSOCIATED_LU         */
    unsigned char    request_lu_name[8];     /* requested LU name or      */
                                                /* associated display LU name */
                                                /* (in EBCDIC)              */
    unsigned char    cipher_spec;           /* SSL cipher specification   */
    unsigned char    reserv3[21];           /* reserved                   */
} TN_SERVER_SESSION_USER_INFO;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode *AP_QUERY_LU_0_TO_3*

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

LU の最大数 (その LU のデータが戻されます)。ある範囲でなく、特定の LU に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すには 0 を指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

lu_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

lu_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

pu_name

LU 情報を求める PU 名。特定の PU に関連した LU についての情報のみを表示するには、PU 名を指定します。すべての PU について完全なリストを取得するには、このフィールドをすべて 2 進ゼロに設定します。

lu_name

ローカル LU の名前。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtring で、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

host_attachment

ホスト接続フィルター。実行中のノードに対してこの verb が発行された場合、このパラメーターは、LU がホストに直接接続されるか、DLUR または PU 集信を介して接続されるかにより、戻り情報をフィルターにかけるかどうかを指定します。値は次のいずれかです。

AP_DIRECT_ATTACHED

ホスト・システムに直接接続された LU のみの情報を戻します。

AP_DLUR_ATTACHED

ローカル・ノードで DLUR がサポートする LU のみの情報を戻します。

AP_DLUR

ダウンストリーム・ノードからパススルー DLUR がサポートする LU のみの情報を戻します。このオプションが有効なのは、ローカル・ノードがネットワーク・ノードの場合のみです。

AP_PU_CONCENTRATION

ダウンストリーム・ノードから SNA ゲートウェイがサポートする LU のみの情報を戻します。

AP_NONE

ホスト接続に関係なく、すべての LU に関する情報を戻します。

ノードが実行中でない場合、このパラメーターは無視されます。Communications Server for Linux はホスト接続に関係なく、すべての LU に関する情報を戻します。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

lu_0_to_3_summary.overlay_size

戻された `lu_0_to_3_summary` 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 `lu_0_to_3_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

lu_0_to_3_summary.pu_name

LU が使用するローカル PU の名前。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC ストリングで、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

lu_0_to_3_summary.lu_name

ローカル LU の名前。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC ストリングで、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

lu_0_to_3_summary.description

LU を記述したヌルで終了するテキスト・ストリング (その LU の定義で指定したもの)。

lu_0_to_3_summary.nau_address

LU のネットワーク・アクセス可能単位アドレス。1 ~ 255 の範囲です。

lu_0_to_3_summary.lu_sscp_sess_active

LU-SSCP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

lu_0_to_3_summary.appl_conn_active

この LU を使用しているアプリケーションがあるかどうかを指定します。値は次のいずれかです。

AP_YES LU を使用しているアプリケーションがあります。

AP_NO LU を使用しているアプリケーションはありません。

lu_0_to_3_summary.plu_sess_active

PLU-SLU セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

lu_0_to_3_summary.host_attachment

LU ホスト接続タイプ。

実行中のノードに対してこの verb が発行された場合、このパラメーターは、次のいずれかの値をとります。

AP_DIRECT_ATTACHED

LU はホスト・システムに直接接続されています。

AP_DLUR_ATTACHED

ローカル・ノードで DLUR がサポートする LU。

AP_DLUR

ダウンストリーム・ノードからパススルー DLUR がサポートする LU。

AP_PU_CONCENTRATION

ダウンストリーム・ノードから SNA ゲートウェイがサポートする LU。

lu_0_to_3_detail.overlay_size

戻された *lu_0_to_3_detail* 構造体のサイズ。すなわち、データ・バッファにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *lu_0_to_3_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

lu_0_to_3_detail.lu_name

ローカル LU の名前。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtring で、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

lu_0_to_3_detail.det_data.lu_sscp_sess_active

LU-SSCP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

lu_0_to_3_detail.det_data.appl_conn_active

この LU を使用しているアプリケーションがあるかどうかを指定します。値は次のいずれかです。

AP_YES LU を使用しているアプリケーションがあります。

AP_NO LU を使用しているアプリケーションはありません。

lu_0_to_3_detail.det_data.plu_sess_active

PLU-SLU セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

lu_0_to_3_detail.det_data.host_attachment

LU ホスト接続タイプ。

実行中のノードに対してこの verb が発行された場合、このパラメーターは、次のいずれかの値をとります。

AP_DIRECT_ATTACHED

LU はホスト・システムに直接接続されています。

AP_DLUR_ATTACHED

ローカル・ノードで DLUR がサポートする LU。

AP_DLUR

ダウンストリーム・ノードからパススルー DLUR がサポートする LU。

AP_PU_CONCENTRATION

ダウンストリーム・ノードから SNA ゲートウェイがサポートする LU。

2 つのセッション (LU-SSCP セッションおよび PLU-SLU セッション) のそれぞれについて、*session_stats* 構造体には、次のパラメーターが含まれます。

rcv_ru_size

受信 RU の最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

send_ru_size

送信 RU の最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

<i>max_send_btu_size</i>	送信できる BTU の最大サイズ。
<i>max_rcv_btu_size</i>	受信できる BTU の最大サイズ。
<i>max_send_pac_win</i>	このセッションの送信ペーシング・ウィンドウの最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)
<i>cur_send_pac_win</i>	このセッションの送信ペーシング・ウィンドウの現行サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)
<i>max_rcv_pac_win</i>	このセッションの受信ペーシング・ウィンドウの最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)
<i>cur_rcv_pac_win</i>	このセッションの受信ペーシング・ウィンドウの現行サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)
<i>send_data_frames</i>	送信された通常フロー・データ・フレームの数。
<i>send_fmd_data_frames</i>	送信された通常フロー FMD データ・フレームの数。
<i>send_data_bytes</i>	送信された通常フロー・データ・バイトの数。
<i>rcv_data_frames</i>	受信された通常フロー・データ・フレームの数。
<i>rcv_fmd_data_frames</i>	受信された通常フロー FMD データ・フレームの数。
<i>rcv_data_bytes</i>	受信された通常フロー・データ・バイトの数。
<i>sidh</i>	セッション ID の上位バイト。
<i>sidl</i>	セッション ID の下位バイト。
<i>odai</i>	起点宛先アサイナー・インディケータ。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合、BIND の送信側は、このフィールドをゼロに設定し、BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合は、このフィールドを 1 に設定します。
<i>ls_name</i>	統計に関連したリンク・ステーション名。この名前は、8 バイトからなる ASCII 文字ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。
<i> pacing_type</i>	PLU-SLU セッションで使用している受信ペーシングのタイプ。値は次のいずれかです。 AP_NONE

AP_PACING_FIXED

lu_0_to_3_detail.det_data.plu_name

1 次 LU の名前。これは 8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。PLU-SLU セッションが活動状態でない場合、このフィールドは予約済みです。

lu_0_to_3_detail.det_data.session_id

8 バイトからなる PLU-SLU セッションの内部 ID。

lu_0_to_3_detail.det_data.app_spec_det_data

このフィールドのデータのフォーマットは、次のように、*app_type* フィールドの値によって異なります。

- *app_type* を AP_NONE に設定した場合、このパラメーターは予約済みです。
- *app_type* が AP_PU_CONCENTRATION に設定されている場合、このフィールドの最初の 8 バイトには、現在このローカル LU を使用しているダウンストリーム LU の LU 名が入ります。これは EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。残りのバイトは予約済みです。
- *app_type* が AP_LUA_APPLICATION に設定されている場合、以下の説明のように、このフィールドは *lua_session_user_info* 構造体で置換されません。
- *app_type* が AP_FMI_APPLICATION に設定されている場合、以下の説明のように、このフィールドは *session_user_info* 構造体で置換されます。

app_type が AP_LUA_APPLICATION に設定されている場合、*app_spec_det_data* フィールドは、この LU を使用する LUA アプリケーションに関する情報が入った *lua_session_user_info* 構造体で置換されます。構造体は次のフィールドで構成されます。

user_ip_address

LUA アプリケーションが実行しているコンピューター (クライアントまたはサーバー) の IP アドレス。これはヌル終了 ASCII ストリングであり、指定できるのは以下のいずれかです。

- IPv4 ドット 10 進アドレス (たとえば、193.1.11.100)
- IPv6 コロン 16 進アドレス (たとえば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)

user_host_address

LUA アプリケーションが実行しているコンピューター (クライアントまたはサーバー) の名前。これはヌル終了 ASCII ストリングであり、IP ホスト名 (newbox.this.co.uk など) を表します。

app_type が AP_FMI_APPLICATION に設定されている場合、*app_spec_det_data* フィールドは、この LU のユーザーに関する情報が入った *session_user_info* 構造体で置換されます。構造体は次のフィールドで構成されます。

QUERY_LU_0_TO_3

user_name

この LU を使用する 3270 エミュレーション・プログラムを実行している Linux ユーザー名。これは 1 ～ 32 文字の ASCII ストリングです。

system_name

プログラムを実行しているコンピューターの名前。

user_pid

LU を使用しているプログラムのプロセス ID。

user_type

LU を使用しているセッションのタイプ (3270 表示セッション、3270 印刷装置セッション)。値は次のいずれかです。

AP_3270_DISPLAY_MODEL_2

AP_3270_DISPLAY_MODEL_3

AP_3270_DISPLAY_MODEL_4

AP_3270_DISPLAY_MODEL_5

AP_PRINTER

AP_SCS_PRINTER

AP_UNKNOWN

user_uid

プログラムを実行している Linux ユーザー ID。

user_gid

プログラムを実行している Linux グループ ID。

user_gname

プログラムを実行している Linux グループ名。これは 1 ～ 32 文字の ASCII ストリングです。

app_type が AP_TN_SERVER に設定されている場合、このフィールドは、この LU を使用している TN3270 プログラムに関する情報が入った *tn_server_session_user_info* 構造体で置換されます。構造体は次のフィールドで構成されます。

user_ip_address

TN3270 プログラムを実行しているコンピューターの IP アドレス。これはヌル終了 ASCII ストリングであり、指定できるのは以下のいずれかです。

- IPv4 ドット 10 進アドレス (たとえば、193.1.11.100)
- IPv6 コロン 16 進アドレス (たとえば、
2001:0db8:0000:0000:0000:0000:1428:57ab または
2001:db8::1428:57ab)

port_number

TN サーバーにアクセスするために TN3270 プログラムが使用する TCP/IP ポート番号。

cb_number

TN サーバー制御ブロック番号。

cfg_default

TN3270 プログラムが、明示的に定義された TN サーバーのユーザー・レコードを使用しているか、構成されたデフォルト・レコードを使用しているかを指定します。デフォルト TN サーバーのユーザー・レコードの構成について詳しくは、230 ページの『DEFINE_TN3270_ACCESS』を参照してください。値は次のいずれかです。

AP_YES プログラムはデフォルト・レコードを使用しています。以下の *cfg_address* パラメーターと *cfg_format* パラメーターは予約済みです。

AP_NO プログラムは明示的に定義されたレコードを使用しています。

cfg_address

このユーザーが使用している構成レコードで定義された、TN3270 プログラムを実行しているコンピューターの TCP/IP アドレス。入力できるのは、以下のいずれかです。フォーマットは *cfg_format* パラメーターで指示されます。

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

cfg_format

cfg_address パラメーターのフォーマットを指定します。値は次のいずれかです。

AP_ADDRESS_IP
IP アドレス

AP_ADDRESS_FQN
別名または完全修飾名

tn3270_level

TN3270 サポートのレベル。値は次のいずれかです。

AP_LEVEL_TN3270
TN3270E プロトコルは使用不可です。

AP_LEVEL_TN3270E
TN3270E プロトコルは使用可能です。

lu_select

LU 選択の方式。値は次のいずれかです。

AP_GENERIC_LU
TN3270 プログラムは総称ディスプレイ LU またはプリンター LU を選択しました。

AP_SPECIFIC_LU
TN3270 プログラムはこの LU を特定して選択しました。

AP_ASSOCIATED_LU
これは DEFINE_TN3270_ASSOCIATION verb によりディスプレイ

LU に関連付けられたプリンター LU か、
 DEFINE_TN3270_ASSOCIATION verb によりプリンター LU に関
 連付けられたディスプレイ LU です。LU はこの関連により
 TN3270 で使用中です。

request_lu_name

要求した LU 名または関連したディスプレイ LU 名。この名前は、8 バイ
 トからなるタイプ A の EBCDIC スtringで、名前が 8 文字に満たない
 場合は、右側にスペースを入れます。

cipher_spec

このセッションに使用している SSL セキュリティーのタイプと暗号化レベ
 ルを示します。値は次のいずれかです。

AP_SSL_NO_SSL

SSL は使用していません。

AP_SSL_NULL_MD5

証明書は交換されていますが、暗号化は使用されていません。

AP_SSL_NULL_SHA

証明書は交換されていますが、暗号化は使用されていません。

AP_SSL_RC4_MD5_EXPORT

40 ビットの暗号化

AP_SSL_RC2_MD5_EXPORT

40 ビットの暗号化

AP_SSL_DES_SHA_EXPORT

56 ビットの暗号化

AP_SSL_RC4_MD5_US

128 ビットの暗号化

AP_SSL_RC4_SHA_US

128 ビットの暗号化

AP_SSL_3DES_SHA_US

triple-DES (168 ビット) の暗号化

lu_0_to_3_detail.det_data.app_type

LU を使用しているアプリケーションがある場合、そのアプリケーションの
 タイプ。値は次のいずれかです。

AP_NONE

LU は使用中ではありません。

AP_LUA_APPLICATION

LU は LUA アプリケーションで使用中です。

AP_PU_CONCENTRATION

LU は SNA ゲートウェイを使用するダウンストリーム LU で使用
 中です。

AP_FMI_APPLICATION

LU は 3270 エミュレーション・プログラムで使用中です。

AP_TN_SERVER

LU は TN サーバーにアクセスする TN3270 プログラムで使用
中
です。

lu_0_to_3_detail.det_data.sscp_id

この LU が使用する PU の ACTPU で受信された SSCP ID が入る 6 バ
イトのフィールド。 *lu_sscp_sess_active* が AP_NO でない場合、このパラメ
ーターはすべてゼロになります。

lu_0_to_3_detail.det_data.bind_lu_type

元の BIND を発行した LU の LU タイプを指定します (活動化の LU-LU
セッションがある場合)。値は次のいずれかです。

AP_LU_TYPE_0

LU タイプ 0

AP_LU_TYPE_1

LU タイプ 1

AP_LU_TYPE_2

LU タイプ 2

AP_LU_TYPE_3

LU タイプ 3

AP_LU_TYPE_6

ダウンストリーム従属型 LU 6.2

AP_LU_TYPE_UNKNOWN

活動状態である LU-LU セッションはありません。

lu_0_to_3_detail.det_data.compression

PLU-SLU セッションで使用中の圧縮レベル (存在する場合)。値は次のい
ずれかです。

AP_NO PLU-SLU セッションで流れるデータが Communications Server for
Linux で圧縮されていないか、または 活動状態の PLU-SLU セッシ
ョンがありません。

AP_YES Communications Server for Linux は、 PLU-SLU セッション・デー
タを圧縮および解凍します。 RLE 圧縮は、アップストリームから
1 次 LU に流れるデータで使用します。 LZ9 圧縮は、ダウンスト
リームから 1 次 LU に流れるデータで使用します。

AP_PASSTHRU

このセッションでの圧縮は、Communications Server for Linux では
なく、セッション・エンドポイント (ホスト LU およびローカル・
アプリケーションまたはダウンストリーム LU) によって実行され
ます。

lu_0_to_3_detail.def_data.description

LU を記述したヌルで終了するテキスト・ストリング (その LU の定義で指
定したもの)。

lu_0_to_3_detail.def_data.nau_address

LU のネットワーク・アクセス可能単位アドレス (1 ~ 255)。

lu_0_to_3_detail.def_data.pool_name

この LU が属する LU プールの名前。これは 8 バイトからなる EBCDIC スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。LU がプールに属さない場合は、このフィールドの 8 つのビットをゼロに設定します。

lu_0_to_3_detail.def_data.pu_name

この LU が使用する PU の名前 (DEFINE_LS verb で指定したもの)。この名前は、8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

lu_0_to_3_detail.def_data.priority

ホストへ送信する場合の LU 優先順位。次のいずれか 1 つに設定します。

AP_NETWORK
AP_HIGH
AP_MEDIUM
AP_LOW

lu_0_to_3_detail.def_data.lu_model

LU のタイプ。次のいずれか 1 つに設定します。

AP_3270_DISPLAY_MODEL_2
AP_3270_DISPLAY_MODEL_3
AP_3270_DISPLAY_MODEL_4
AP_3270_DISPLAY_MODEL_5
AP_PRINTER
AP_SCS_PRINTER
AP_UNKNOWN

lu_0_to_3_detail.def_data.sscp_id

この LU の活動化が許可される SSCP の ID を指定します。これは 6 バイトの 2 進数フィールドです。このパラメーターのすべてのビットがゼロに設定されている場合は、LU はどの SSCP からでも活動化できます。

lu_0_to_3_detail.def_data.timeout

LU のタイムアウト (秒単位)。タイムアウトを指定し、LU のユーザーが OPEN_LU_SSCP_SEC_RQ (または、SNA ゲートウェイの場合は、ダウンストリーム LU の定義) で allow_timeout を指定した場合は、PLU-SLU セッションが活動状態であったこの期間が経過し、次の条件を満たすと、LU は非活動化されます。

- セッションが、限定リソース・リンクを通る。
- セッションが再使用される前に LU の使用を求めている別のアプリケーションがある。

タイムアウトがゼロに設定されている場合は、LU は非活動化されません。

lu_0_to_3_detail.def_data.term_method

このパラメーターは、この LU からホストへの PLU-SLU セッションを Communications Server for Linux が終了させる方法を指定します。値は次のいずれかです。

AP_USE_NODE_DEFAULT

DEFINE_NODE の *send_term_self* パラメーターで指定された、ノードのデフォルト終了方法を使用します。

AP_SEND_UNBIND

UNBIND を送信してセッションを終了します。

AP_SEND_TERM_SELF

TERM_SELF を送信してセッションを終了します。

lu_0_to_3_detail.def_data.disconnect_on_unbind

このパラメーターは、この LU が TN3270 クライアントによって使用される場合のみ適用されます。これは、ホストが VTAM MSG10 を表示するか、またはホスト・セッション・マネージャーに戻る代わりに UNBIND を送信するときに、セッションを終了させるかどうかを指定します。値は次のいずれかです。

AP_YES ホストがタイプ 2 (BIND が用意されている) でない UNBIND を送信する場合は、セッションを終了させます。

AP_NO ホストが UNBIND を送信してもセッションを終了させません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*lu_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_LU_LU_PASSWORD

QUERY_LU_LU_PASSWORD は、ローカル LU とパートナー LU 間のセッション・レベル・セキュリティー検査に使用するパスワードに関する情報を戻します。この verb は使用するオプションに応じて、特定のパートナー LU のパスワードまたは複数のパートナー LU のパスワードに関する情報を入手するのに使用できます。

VCB 構造体

```
typedef struct query_lu_lu_password
{
    AP_UINT16      opcode;                /* Verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;                /* reserved                  */
    AP_UINT16      primary_rc;            /* Primary return code      */
    AP_UINT32      secondary_rc;          /* Secondary return code    */
    unsigned char  *buf_ptr;              /* pointer to buffer        */
    AP_UINT32      buf_size;              /* buffer size              */
    AP_UINT32      total_buf_size;        /* total buffer size required */
    AP_UINT16      num_entries;           /* number of entries        */
    AP_UINT16      total_num_entries;     /* total number of entries  */
    unsigned char  list_options;          /* listing options          */
    unsigned char  reserv3;               /* reserved                  */
    unsigned char  lu_name[8];            /* LU name                   */
    unsigned char  lu_alias[8];           /* LU alias                  */
    unsigned char  plu_alias[8];          /* partner LU alias         */
    unsigned char  fqplu_name[17];        /* fully-qual. partner LU name */
} QUERY_LU_LU_PASSWORD;

typedef struct password_info
{
    AP_UINT16      overlay_size;          /* size of returned entry   */
    unsigned char  plu_alias[8];          /* partner LU alias         */
    unsigned char  fqplu_name[17];        /* fully-qual. partner LU name */
    unsigned char  description[32];       /* resource description     */
    unsigned char  reserv1[16];           /* reserved                  */
    unsigned char  password[8];           /* password                  */
    unsigned char  protocol_defined;      /* protocol defined         */
    unsigned char  protocol_in_use;       /* protocol in use          */
} PASSWORD_INFO;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_LU_LU_PASSWORD

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファーへのポインター。

buf_size
指定したデータ・バッファーのサイズ。

num_entries
パートナー LU の最大数 (そのパスワード情報が戻されます)。ある範囲でなく、特定のエントリーに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

plu_alias パラメーターまたは *fqplu_name* パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

plu_alias パラメーターまたは *fqplu_name* パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

lu_name

LU 名。この名前は 8 バイトからなるタイプ A の EBCDIC 文字ストリングです。LU を LU 名ではなく、LU 別名で識別することを示すには、このパラメーターを 8 桁の 2 進ゼロに設定し、*lu_alias* パラメーター LU で別名を指定します。

lu_alias

ローカルで定義された LU 別名。これは 8 バイトからなる ASCII 文字ストリングです。このパラメーターは、*lu_name* をすべてゼロに設定した場合のみ使用され、それ以外の場合は無視されます。CP に関連した LU (デフォルト LU) を示すには、*lu_name* と *lu_alias* の両方のビットをゼロに設定します。

plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII 文字ストリングです。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。それ以外の場合、パートナー LU 名に対して LU 別名または LU の完全修飾名を指定する必要があります。パートナー LU を LU 別名ではなく、LU の完全修飾名で識別することを示すには、このパラメーターを 8 桁の 2 進ゼロに設定し、*fqplu_name* パラメーターで LU 別名を指定します。

fqplu_name

パートナー LU の完全修飾ネットワーク名。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。それ以外の場合、パートナー LU 名に対して LU 別名または LU の完全修飾名を指定する必要があります。このパラメーターは、*plu_alias* をすべてゼロに設定した場合のみ使用され、それ以外の場合は無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

buf_size
指定したバッファーに戻された情報の長さ。

total_buf_size
要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries
データ・バッファー内に戻されたエントリーの数。

total_num_entries
使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。
データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

password_info.overlay_size
戻された `password_info` 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。
アプリケーションが戻されたバッファーを調べて、各 `password_info` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

password_info.plu_alias
パートナー LU 別名。これは 8 バイトからなる ASCII 文字ストリングです。

password_info.fqplu_name
パートナー LU の完全修飾ネットワーク名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

password_info.description
LU-LU パスワードを記述したヌルで終了するテキスト・ストリング (パスワードの定義で指定したもの)。

password_info.password
DEFINE_LU_LU_PASSWORD `verb` で指定したパスワードの暗号化されたバージョン。これは 8 バイトからなる 16 進数ストリングです。

password_info.protocol_defined

このパートナー LU で使用するために定義され、要求された LU-LU 検査プロトコル。値は次のいずれかです。

AP_BASIC

基本セキュリティ・プロトコルが要求されました。

AP_ENHANCED

拡張セキュリティ・プロトコルが要求されました。

AP_EITHER

基本セキュリティ・プロトコルまたは拡張セキュリティ・プロトコルが受け入れられました。

password_info.protocol_in_use

このパートナー LU で使用している LU-LU 検査プロトコル。値は次のいずれかです。

AP_BASIC

基本セキュリティ・プロトコルを使用中。

AP_ENHANCED

拡張セキュリティ・プロトコルを使用中。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_ALIAS

指定した *lu_alias* パラメーターは、構成されたどの LU の別名にも一致しませんでした。

AP_INVALID_LU_NAME

指定した *lu_name* パラメーターは、構成されたどの LU の名前にも一致しませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、**AP_PARAMETER_CHECK** に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_LU_POOL

QUERY_LU_POOL は、LU プールと、そのプールに属する LU についての情報を戻します。

この verb は使用するオプションに応じて、特定の LU またはプール、あるいは複数の LU またはプールに関する情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_lu_pool
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;        /* secondary return code */
    unsigned char  *buf_ptr;             /* pointer to buffer */
    AP_UINT32      buf_size;             /* buffer size */
    AP_UINT32      total_buf_size;      /* total buffer size required */
    AP_UINT16      num_entries;          /* number of entries */
    AP_UINT16      total_num_entries;    /* total number of entries */
    unsigned char  list_options;        /* listing options */
    unsigned char  reserv3;             /* reserved */
    unsigned char  pool_name[8];        /* Pool name */
    unsigned char  lu_name[8];          /* LU name */
} QUERY_LU_POOL;

typedef struct lu_pool_summary
{
    AP_UINT16      overlay_size;         /* size of returned entry */
    unsigned char  pool_name[8];        /* Pool name */
    unsigned char  description[32];     /* resource description */
    unsigned char  reserv1[16];         /* reserved */
    AP_UINT16      num_active_lus;      /* number of active lus */
    AP_UINT16      num_avail_lus;       /* number of available lus */
} LU_POOL_SUMMARY;

typedef struct lu_pool_detail
{
    AP_UINT16      overlay_size;         /* size of returned entry */
    unsigned char  pool_name[8];        /* Pool name */
    unsigned char  description[32];     /* resource description */
    unsigned char  reserv1[16];         /* reserved */
    unsigned char  lu_name[8];          /* LU name */
    unsigned char  lu_sscp_sess_active; /* Is LU-SSCP session active */
    unsigned char  appl_conn_active;    /* Is appl connection open */
    unsigned char  plu_sess_active;     /* Is PLU-SLU session active */
} LU_POOL_DETAIL;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_LU_POOL

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

エントリーの最大数 (そのエントリーのデータが戻されます)。 *list_options*

を AP_SUMMARY に設定した場合は、各エントリーが単一の LU プールになります。*list_options* を AP_DETAIL に設定した場合は、各エントリーがプール内の 1 つの LU (または空の LU プールを示すエントリー) になります。

ある範囲でなく、特定のエントリーに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ (LU プールのリスト)。

AP_DETAIL

詳細情報 (LU プール内の個別の LU のリスト)。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

pool_name パラメーターと *lu_name* パラメーターを組み合わせて指定したエントリーから開始します。

AP_LIST_FROM_NEXT

pool_name パラメーターと *lu_name* パラメーターを組み合わせて指定したエントリーの直後のエントリーから開始します。

リストは、最初に *pool_name*、次に *lu_name* の順番で並んでいます。リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

pool_name

LU プールの名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。これは 8 バイトからなる EBCDIC タイプ A スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

lu_name

LU 名。この値は、*list_options* を AP_FIRST_IN_LIST または AP_SUMMARY に設定した場合には無視されます。この名前は、8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

プール内のすべての LU に関する情報を入手するには、*pool_name* をプールの名前に設定し、*num_entries* のビットを 0 に設定し、*lu_name* を 8 桁の 2 進ゼロに設定します。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

buf_size
指定したバッファーに戻された情報の長さ。

total_buf_size
要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries
データ・バッファー内に戻されたエントリーの数。

total_num_entries
使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。
データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

lu_pool_summary.overlay_size
戻された *lu_pool_summary* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。
アプリケーションが戻されたバッファーを調べて、各 *lu_pool_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

lu_pool_summary.pool_name
LU プールの名前。これは 8 バイトからなる EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

lu_pool_summary.description
LU プールを記述したヌルで終了するテキスト・ストリング (そのプールの定義で指定したもの)。

lu_pool_summary.num_active_lus
プール内の活動状態である LU の数。

lu_pool_summary.num_avail_lus
強制されたオープン要求による活動化に使用可能なプール内の LU の数。
この数は、PU が活動状態であるか、またはホスト・リンクが自動起動可能で、接続が解放されているすべての LU を示します。

このカウントでは、PU の LU *model_type*、*model_name* および DDDL U サポートを考慮に入れていません。オープン要求で *model_type* に特定の値を

指定する場合には、このカウントに組み込まれている一部の LU は使用できない可能性があります。これは、このような LU のモデル・タイプが間違っているためです。

lu_pool_detail.overlay_size

戻された `lu_pool_detail` 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 `lu_pool_detail` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

lu_pool_detail.pool_name

LU が属する LU プールの名前。これは 8 バイトからなる EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

lu_pool_detail.description

LU プールを記述したヌルで終了するテキスト・ストリング (そのプールの定義で指定したもの)。

lu_pool_detail.lu_name

LU の LU 名。この名前は、8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。LU 名に 8 桁の 2 進ゼロからなるストリングを指定した特定のプールについて単一の `lu_pool_detail` 構造体が戻された場合は、指定されたプールが空であることを示します。

lu_pool_detail.lu_sscp_sess_active

LU-SSCP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

lu_pool_detail.appl_conn_active

この LU を使用しているアプリケーションがあるかどうかを指定します。値は次のいずれかです。

AP_YES LU を使用しているアプリケーションがあります。

AP_NO LU を使用しているアプリケーションはありません。

lu_pool_detail.plu_sess_active

PLU-SLU セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*lu_name* パラメーターは無効でした。

AP_INVALID_POOL_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*pool_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_LU62_TIMEOUT

QUERY_LU62_TIMEOUT `verb` は、前に DEFINE_LU62_TIMEOUT `verb` で定義した LU タイプ 6.2 セッションのタイムアウトに関する情報を戻します。

情報はリストとして戻されます。特定のタイムアウトに関する情報または複数のタイムアウト値に関する情報を入手するには、*resource_type* パラメーターと *resource_name* パラメーターに値を指定します。 *list_options* パラメーターを AP_FIRST_IN_LIST に設定した場合、*resource_type* パラメーターと *resource_name* パラメーターは無視されます。戻されるリストは、最初に *resource_type*、次に *resource_name* の順番で並んでいます。

resource_type の場合は、次の順序になります。

1. グローバル・タイムアウト
2. ローカル LU タイムアウト
3. パートナー LU タイムアウト
4. モード・タイムアウト

resource_name の場合は、次の順序になります。

1. 名前の長さ

2. 同じ長さの名前については ASCII の字句解釈配列

list_options パラメーターを AP_LIST_FROM_NEXT に設定した場合、戻されるリストは、定義した順序 (指定したエントリーが存在するかどうか) に従って次のエントリーから開始されます。

VCB 構造体

```
typedef struct query_lu62_timeout
{
    AP_UINT16    opcode;                /* verb operation code      */
    unsigned char reserv2;              /* reserved                  */
    unsigned char format;               /* reserved                  */
    AP_UINT16    primary_rc;            /* primary return code      */
    AP_UINT32    secondary_rc;          /* secondary return code    */
    unsigned char *buf_ptr;              /* buffer pointer           */
    AP_UINT32    buf_size;               /* buffer size              */
    AP_UINT32    total_buf_size;        /* total buffer size        */
    AP_UINT16    num_entries;            /* number of entries        */
    AP_UINT16    total_num_entries;     /* total number of entries  */
    unsigned char list_options;          /* list options             */
    unsigned char reserv3;              /* reserved                  */
    unsigned char resource_type;        /* resource type            */
    unsigned char resource_name[17];    /* resource name            */
} QUERY_LU62_TIMEOUT;

typedef struct lu62_timeout_data
{
    AP_UINT16    overlay_size;          /* overlay size             */
    unsigned char resource_type;        /* resource type            */
    unsigned char resource_name[17];    /* resource name            */
    AP_UINT16    timeout;               /* timeout                  */
} LU62_TIMEOUT_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_LU62_TIMEOUT

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size
指定したデータ・バッファのサイズ。

num_entries
エントリーの最大数 (そのエントリーのデータが戻されます)。ある範囲でなく、特定のエントリーに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すには 0 を指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options
Communications Server for Linux がデータを戻すためのエントリー・リスト内の開始位置。リストは、まず *resource_type* により AP_GLOBAL_TIMEOUT、AP_LOCAL_LU_TIMEOUT、AP_PARTNER_LU_TIMEOUT、AP_MODE_TIMEOUT の順に並べられ、次に *resource_name* により名前の長さの順に配列され、さらに同じ長さの名前については ASCII の字句解釈配列により並べられます。

値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

resource_type パラメーターと *resource_name* パラメーターを組み合わせ指定したエントリーから開始します。

AP_LIST_FROM_NEXT

resource_type パラメーターと *resource_name* パラメーターを組み合わせ指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

resource_type

照会するタイムアウトのタイプを指定します。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

値は次のいずれかです。

AP_GLOBAL_TIMEOUT

ローカル・ノードのすべての LU 6.2 セッションにタイムアウトが適用されます。

AP_LOCAL_LU_TIMEOUT

resource_name パラメーターで指定したローカル LU のすべての LU 6.2 セッションにタイムアウトが適用されます。

AP_PARTNER_LU_TIMEOUT

resource_name パラメーターで指定したパートナー LU のすべての LU 6.2 セッションにタイムアウトが適用されます。

AP_MODE_TIMEOUT

resource_name パラメーターで指定したモードを使用するすべての LU 6.2 セッションにタイムアウトが適用されます。

resource_name

照会するリソースの名前。この値は次のいずれかです。

- *resource_type* を AP_GLOBAL_TIMEOUT に設定した場合は、このパラメーターを指定しないでください。
- *resource_type* を AP_LOCAL_LU_TIMEOUT に設定した場合は、*resource_name* の最初の 8 バイトのみが有効で、この 8 バイトをローカル LU の名前に設定します。これは英字で始まり 8 バイトからなるタイプ A の EBCDIC 英数字ストリングで、右側に EBCDIC スペースを入れます。残りの 9 バイトは、すべてゼロに設定します。
- *resource_type* を AP_PARTNER_LU_TIMEOUT に設定した場合は、*resource_name* の 17 バイトがすべて有効で、この 17 バイトをパートナー LU の完全修飾名に設定し、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 の A ストリング文字からなるネットワーク名、1 つの EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 の A ストリング文字からなるパートナー LU 名で構成されます。
- *resource_type* を AP_MODE_TIMEOUT に設定した場合は、*resource_name* の最初の 8 バイトのみが有効で、この 8 バイトをモードの名前に設定しま

す。これは英字で始まり 8 バイトからなるタイプ A の EBCDIC 英数字
 スtringで、右側に EBCDIC スペースを入れます。残りの 9 バイト
 は、すべてゼロに設定します。

このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には
 無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメー
 ターを戻します。

primary_rc
 AP_OK

buf_size
 指定したバッファーに戻された情報の長さ。

total_buf_size
 要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示
 す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻
 されなかったことを示しています。

num_entries
 データ・バッファー内に戻されたエントリーの数。

total_num_entries
 使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能な
 エントリーがすべて戻されなかったことを示しています。
 データ・バッファー内の各エントリーは、次のパラメーターで構成されてい
 ます。

resource_type
 タイムアウトのタイプ。値は次のいずれかです。

AP_GLOBAL_TIMEOUT

ローカル・ノードのすべての LU 6.2 セッションにタイムアウトが
 適用されます。 *resource_name* パラメーターはすべてゼロに設定し
 ます。

AP_LOCAL_LU_TIMEOUT

resource_name パラメーターで指定したローカル LU のすべての
 LU 6.2 セッションにタイムアウトが適用されます。

AP_PARTNER_LU_TIMEOUT

resource_name パラメーターで指定したパートナー LU のすべての
 LU 6.2 セッションにタイムアウトが適用されます。

AP_MODE_TIMEOUT

resource_name パラメーターで指定したモードを使用するすべての
 LU 6.2 セッションにタイムアウトが適用されます。

resource_name
 リソースの名前。この名前は、*resource_type* パラメーターの値に応じて、ロ

QUERY_LU62_TIMEOUT

ーカル LU、パートナー LU、またはモードです。 *resource_type* を AP_GLOBAL_TIMEOUT に設定した場合は、このパラメーターのビットはゼロに設定されます。

timeout タイムアウト期間 (秒単位)。0 (ゼロ) の値は、セッションが解放されるとすぐにタイムアウトになることを示します。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_RESOURCE_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*resource_type* と *resource_name* の組み合わせが設定されたものの値にも一致しません。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_MDS_APPLICATION

QUERY_MDS_APPLICATION は、MS *verb* の REGISTER_MS_APPLICATION を発行して MDS レベルのメッセージに登録されたアプリケーションのリストを戻します。この *verb* について詳しくは、「*Communications Server for Linux MS Programmer's Guide*」を参照してください。

この *verb* は、使用するオプションに応じて、特定のアプリケーションまたは複数のアプリケーションに関する情報を入手するために使用できます。

この *verb* は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_mds_application
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code        */
}
```

```

unsigned char *buf_ptr; /* pointer to buffer */
AP_UINT32 buf_size; /* buffer size */
AP_UINT32 total_buf_size; /* total buffer size required */
AP_UINT16 num_entries; /* number of entries */
AP_UINT16 total_num_entries; /* total number of entries */
unsigned char list_options; /* listing options */
unsigned char reserv3; /* reserved */
unsigned char application[8]; /* application */
} QUERY_MDS_APPLICATION;

typedef struct mds_application_data
{
    AP_UINT16 overlay_size; /* size of returned entry */
    unsigned char application[8]; /* application name */
    AP_UINT16 max_rcv_size; /* max data size appl can receive */
    unsigned char reserva[20]; /* reserved */
} MDS_APPLICATION_DATA;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_MDS_APPLICATION

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

アプリケーションの最大数 (そのアプリケーションのデータが戻されます)。ある範囲でなく、特定のアプリケーションに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリを戻します。

list_options

Communications Server for Linux がデータを戻すためのアプリケーション・リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリから開始します。

AP_LIST_INCLUSIVE

application パラメーターで指定したエントリから開始します。

AP_LIST_FROM_NEXT

application パラメーターで指定したエントリの直後のエントリから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

application

情報を求めるアプリケーションの名前、またはリストの索引として使用する名前。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場

合には無視されます。この名前は、8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

buf_size
指定したバッファに戻された情報の長さ。

total_buf_size
要求したリスト情報をすべて戻すために必要だったバッファのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries
データ・バッファ内に戻されたエントリーの数。

total_num_entries
使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。
データ・バッファ内の各エントリーは、次のパラメーターで構成されています。

mds_application_data.overlay_size
戻された *mds_application_data* 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。
アプリケーションが戻されたバッファを調べて、各 *mds_application_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

mds_application_data.application
登録されたアプリケーションの名前。この名前は、8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

mds_application_data.max_rcv_size
アプリケーションが 1 つのメッセージで受信できるバイトの最大数 (これはアプリケーションが MDS に登録されるときに指定されます)。MDS レベルのアプリケーション登録について詳しくは、「*Communications Server for Linux MS Programmer's Guide*」を参照してください。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_APPLICATION_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*application* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの `verb` をサポートしていないために `verb` が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは MS ネットワーク管理機能をサポートしていません。これは、DEFINE_NODE `verb` の *mds_supported* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_MDS_STATISTICS

QUERY_MDS_STATISTICS は、Management Services 統計情報を戻します。この `verb` を使用すると、MDS 経路指定トラフィックのレベルを適切に評価できます。この情報は、DEFINE_NODE `verb` を使用して構成された送信アラート・キューの必要サイズを決定するときにも使用できます。

この `verb` は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_mds_statistics
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;              /* reserved                  */
}
```

QUERY_MDS_STATISTICS

```
unsigned char  format;                /* reserved */
AP_UINT16     primary_rc;             /* primary return code */
AP_UINT32     secondary_rc;          /* secondary return code */
AP_UINT32     alerts_sent;           /* number of alert sends */
AP_UINT32     alert_errors_rcvd;     /* error messages received
/* for alert sends */
AP_UINT32     uncorrelated_alert_errors; /* uncorrelated alert errors
/* received */
AP_UINT32     mds_mus_rcvd_local;    /* number of MDS_MUs received
/* from local applications */
AP_UINT32     mds_mus_rcvd_remote;   /* number of MDS_MUs received
/* from remote applications */
AP_UINT32     mds_mus_delivered_local; /* number of MDS_MUs delivered
/* to local applications */
AP_UINT32     mds_mus_delivered_remote; /* number of MDS_MUs delivered
/* to remote applications */
AP_UINT32     parse_errors;          /* number of MDS_MUs received
/* with parse errors */
AP_UINT32     failed_deliveries;     /* number of MDS_MUs where
/* delivery failed */
AP_UINT32     ds_searches_performed; /* number of DS searches
/* performed */
AP_UINT32     unverified_errors;    /* number of unverified errors */
unsigned char  reserva[20];         /* reserved */
} QUERY_MDS_STATISTICS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

```
opcode AP_QUERY_MDS_STATISTICS
```

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

alerts_sent

MDS トランスポート・システムを使用して送信されたローカル起点のアラートの数。

alert_errors_rcvd

アラートが入っているメッセージの送達の失敗を示す、MDS が受信したエラー・メッセージの数。

uncorrelated_errors_rcvd

アラートが入っているメッセージの送達の失敗を示す、MDS が受信したエラー・メッセージの数。エラー・メッセージを MDS の送信アラート・キュー上のアラートに相関させることができない場合に、送達が失敗します。MDS は固定サイズのキューを保持し、問題判別フォーカル・ポイントに送信されたアラートをキャッシュに入れます。キューが最大サイズに達すると、最も古いアラートは廃棄され、新しいアラートで置換されます。送達エラー・メッセージを受信すると、MDS は、問題判別フォーカル・ポイントが復元されるまでアラートが保持されるように、キャッシュに入れられたアラートにエラー・メッセージを相関させようとしています。

注: *alert_errors_rcvd* と *uncorrelated_errors_rcvd* の 2 つのカウントを使用すると、送信アラート・キューのサイズ (DEFINE_NODE で指定された) が適切かどうかを検査することができます。
uncorrelated_errors_rcvd の値が時間の経過と共に増加する場合は、送信アラート・キューが小さすぎることを示しています。

mds_mus_rcvd_local

ローカル・アプリケーションから受信された MDS_MU の数。

mds_mus_rcvd_remote

MDS_RECEIVE トランザクション・プログラムおよび MSU_HANDLER トランザクション・プログラムを使用してリモート・ノードから受信された MDS_MU の数。

mds_mus_delivered_local

ローカル・アプリケーションに正常に送達された MDS_MU の数。

mds_mus_delivered_remote

MDS_SEND トランザクション・プログラムを使用してリモート・ノードに正常に送達された MDS_MU の数。

parse_errors

ヘッダー・フォーマットにエラーがあった受信済み MDS_MU の数。

failed_deliveries

このノードが送達に失敗した MDS_MU の数。

ds_searches_performed

ある MDS_MU のネクスト・ホップを見つけるために使用されたディレクトリー・サービスの数。(ネットワーク・ノードのみに有効)。

unverified_errors

ある MDS_MU のネクスト・ホップを判別するために未検査の (ローカル・ディレクトリー・サービス検索) データを使用したために発生した経路指定エラー検索の数。これらのエラーの 1 つが発生するたびに、ディレクトリー・サービスは中央ディレクトリー検索またはブロードキャスト検索の機能を使用して検索を繰り返す必要があります。(ネットワーク・ノードのみに有効)。

戻りパラメーター: サポートされていない機能

ローカル・ノードの構成がこの verb をサポートしていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは MS ネットワーク管理機能をサポートしていません。これは、DEFINE_NODE verb の *mds_supported* パラメーターによって定義されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_MODE

QUERY_MODE は、ローカル LU がパートナー LU と共に使用しているモードまたは使用したモードに関する情報を戻します。

この verb は使用するオプションに応じて、特定のパートナー LU とモードの組み合わせまたは複数のモードに関する情報、およびセッションが現在活動状態であるモード、または使用したすべてのモードに関する情報を入手するために使用できます。この verb が戻すのはモードと LU の現在の使用状況に関する情報であって、モードと LU の定義ではありません。モードと LU の定義は、QUERY_MODE_DEFINITION を使用して入手してください。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_mode
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  *buf_ptr;        /* pointer to buffer            */
    AP_UINT32      buf_size;        /* buffer size                  */
    AP_UINT32      total_buf_size;  /* total buffer size required   */
    AP_UINT16      num_entries;     /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries      */
    unsigned char  list_options;    /* listing options              */
    unsigned char  reserv3;         /* reserved                     */
    unsigned char  lu_name[8];      /* LU name                      */
    unsigned char  lu_alias[8];    /* LU alias                     */
    unsigned char  plu_alias[8];   /* partner LU alias            */
    unsigned char  fqplu_name[17]; /* fully qualified partner LU name */
    unsigned char  mode_name[8];   /* mode name                   */
    unsigned char  active_sessions; /* active sessions only filter  */
} QUERY_MODE;

typedef struct mode_summary
{
    AP_UINT16      overlay_size;    /* size of returned entry       */
    unsigned char  mode_name[8];    /* mode name                    */
    unsigned char  description[32]; /* resource description         */
    unsigned char  reserv2[16];     /* reserved                     */
    AP_UINT16      sess_limit;      /* current session limit        */
    AP_UINT16      act_sess_count;  /* currently active sessions count */
    unsigned char  fqplu_name[17]; /* fully-qualified partner LU name */
    unsigned char  reserv1[3];     /* reserved                     */
} MODE_SUMMARY;

typedef struct mode_detail
{
    AP_UINT16      overlay_size;    /* size of returned entry       */
    unsigned char  mode_name[8];    /* mode name                    */
    unsigned char  description[32]; /* resource description         */
    unsigned char  reserv2[16];     /* reserved                     */
    AP_UINT16      sess_limit;      /* session limit                */
    AP_UINT16      act_sess_count;  /* currently active sessions count */
    unsigned char  fqplu_name[17]; /* fully-qualified partner LU name */
    unsigned char  reserv1[3];     /* reserved                     */
    AP_UINT16      min_conwinners_source; /* minimum conwinner sess limit */
    AP_UINT16      min_conwinners_target; /* minimum conloser sess limit  */
    unsigned char  drain_source;    /* drain source?                */
}
```



```

unsigned char  drain_partner;      /* drain partner?          */
AP_UINT16     auto_act;           /* auto activated conwinner */
/* session limit          */
AP_UINT16     act_cw_count;       /* active conwinner sessions count */
AP_UINT16     act_cl_count;      /* active conloser sessions count */
unsigned char  sync_level;       /* synchronization level    */
unsigned char  default_ru_size;  /* default RU size to maximize */
/* performance          */
AP_UINT16     max_neg_sess_limit; /* maximum negotiated session limit*/
AP_UINT16     max_rcv_ru_size;   /* maximum receive RU size   */
AP_UINT16     pending_session_count; /* pending sess count for mode */
AP_UINT16     termination_count; /* termination count for mode */
AP_UINT16     implicit;         /* implicit or explicit entry */
unsigned char  reserva[15];     /* reserved                  */
} MODE_DETAIL;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_MODE

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

モードの最大数 (そのモードのデータが戻されます)。ある範囲でなく、特定モードに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリー (指定したローカル LU の最初のパートナー LU およびモード) から開始します。

AP_LIST_INCLUSIVE

指定したパートナー LU 名とモード名で指定したエントリーから開始します。

AP_LIST_FROM_NEXT

指定されたパートナー LU 名とモード名で指定したエントリーの直後のエントリーから開始します。

AP_FIRST_IN_LIST の場合、リストの索引として使用されるエントリーは、*lu_name* (または *lu_alias*) と *fqplu_name* (または *plu_alias*) を組み合わせて定義されます。 *fqplu_name* または *plu_alias* を指定しなかった場合は、索引として使用されるエントリーは *lu_name* (または *lu_alias*) です。

AP_LIST_INCLUSIVE または AP_LIST_FROM_NEXT の場合、リストの索引として使用されるエントリーは、*lu_name* (または *lu_alias*)、*fqplu_name* (または *plu_alias*)、および指定された *mode_name* を組み合わせて定義されます。リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

lu_name

LU 名。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。LU を LU 名ではなく、別名で識別することを指定するには、このパラメーターを 8 桁の 2 進ゼロに設定し、次のパラメーターで LU 別名を指定します。

lu_alias

ローカルで定義された LU 別名。このパラメーターは、*lu_name* を 8 桁の 2 進ゼロに設定した場合のみ使用され、それ以外の場合は無視されます。

この名前は 8 バイトからなる ASCII スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。CP に関連した LU (デフォルト LU) を示すには、*lu_name* と *lu_alias* の両方のビットをゼロに設定します。

plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。それ以外の場合は、パートナー LU 名に対して LU 別名または LU の完全修飾名を指定する必要があります。LU を別名ではなく、LU 名で識別することを指定するには、このパラメーターを 8 桁の 2 進ゼロに設定し、次のパラメーターで LU 名を指定します。

fqplu_name

パートナー LU の完全修飾ネットワーク名。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。それ以外の場合は、パートナー LU 名に対して LU 別名または LU の完全修飾名を指定する必要があります。このパラメーターは、*plu_alias* を 8 桁の 2 進ゼロに設定した場合のみ使用され、それ以外の場合は無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A スtring 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring 文字からなるネットワーク名で構成されます。

mode_name

1 グループのセッションのネットワーク特性を指定するモード名。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

active_sessions

セッションが活動状態であるモードのみに関する情報を戻すか、すべてのモードに関する情報を戻すかを指定します。値は次のいずれかです。

AP_YES セッションが現在活動状態であるモードのみに関する情報を戻します。

AP_NO セッションが活動状態であるか、活動状態であったすべてのモードに関する情報を戻します。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

mode_summary.overlay_size

戻された `mode_summary` 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 `mode_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

mode_summary.mode_name

モード名。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

mode_summary.description

モードを記述したヌルで終了するテキスト・String (そのモードの定義で指定したもの)。

mode_summary.sess_limit

現行のセッション限度。

mode_summary.act_sess_count

このモードを使用する、指定したローカル LU とパートナー LU 間の活動状態であるセッションの合計数。

mode_summary.fqplu_name

パートナー LU の完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

mode_detail.overlay_size

戻された *mode_detail* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *mode_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

mode_detail.mode_name

モード名。この名前は、8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

mode_detail.description

モードを記述したヌルで終了するテキスト・ストリング (そのモードの定義で指定したもの)。

mode_detail.sess_limit

現行のセッション限度。

mode_detail.act_sess_count

このモードを使用する、指定したローカル LU とパートナー LU 間の活動状態であるセッションの合計数。

mode_detail.fqplu_name

パートナー LU の完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

mode_detail.min_conwinners_source

ローカル LU が競合勝者であるセッションの最小数を指定します。

mode_detail.min_conwinners_target

ローカル LU が競合敗者であるセッションの最小数を指定します。

mode_detail.drain_source

セッション限度が変更またはリセットされたとき、ローカル LU が待機状態のセッション要求をセッションの非活動化前に満たすかどうかを指定します。値は次のいずれかです。

AP_YES 待機状態のセッション要求は、セッションを非活動化する前に満たされます。

AP_NO 待機状態のセッション要求は満たされません。

mode_detail.drain_partner

セッション限度が変更またはリセットされたとき、パートナー LU が待機状態のセッション要求をセッションの非活動化前に満たすかどうかを指定します。値は次のいずれかです。

AP_YES 待機状態のセッション要求は、セッションを非活動化する前に満たされます。

AP_NO 待機状態のセッション要求は満たされません。

mode_detail.auto_act

パートナー LU との CNOS 交換に続いて自動的に活動化される競合勝者セッションの数。

mode_detail.act_cw_count

このモードを使用している活動状態である競合勝者セッションの数。(ローカル LU は、これらのセッションの 1 つを使用する前に「送信権要求」を必要としません。)

mode_detail.act_cl_count

このモードを使用している活動状態である競合敗者セッションの数。(ローカル LU は、これらのセッションの 1 つを使用する前に「送信権要求」を必要とします。)

mode_detail.sync_level

モードがサポートする同期レベルを指定します。値は次のいずれかです。

AP_CONFIRM

モードは CONFIRM verb および CONFIRMED verb を使用する同期をサポートします。

AP_SYNCPT

モードは同期点機能をサポートします。

AP_NONE

モードは同期をサポートしません。

mode_detail.default_ru_size

最大 RU サイズのデフォルト上限を使用するかどうかを指定します。値は次のいずれかです。

AP_YES Communications Server for Linux は、モードの定義で指定された最大 RU サイズを無視し、最大 RU サイズの上限をリンク BTU サイズに収まる最大値に設定します。

AP_NO Communications Server for Linux は、モードの定義で指定された最大 RU サイズを使用します。

mode_detail.max_neg_sess_limit

折衝可能な最大セッション限度。ターゲット LU として CNOS 処理中にローカル LU がこのモード名で使用できる最大セッション限度を指定します。

mode_detail.max_rcv_ru_size

受信 RU の最大サイズ。

mode_detail.pending_session_count

保留セッション (セッション活動化の完了を待つセッション) の数を指定します。

mode_detail.termination_count

前の CNOS verb でモード・セッション限度がゼロに設定されているが、会話で使用済または使用待ちであったためにセッションがまだ活動状態である場合は、このパラメーターは、まだ非活動化されていないセッションの数を指定します。

mode_detail.implicit

エントリーが暗黙的な定義で作成されたか明示的な定義で作成されたかを指定します。値は次のいずれかです。

AP_YES エントリーは暗黙的なエントリーです。

AP_NO エントリーは明示的なエントリーです。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LU_ALIAS

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*lu_alias* パラメーターは無効でした。

AP_INVALID_LU_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*lu_name* パラメーターは無効でした。

AP_INVALID_MODE_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*mode_name* パラメーターは無効でした。

AP_INVALID_PLU_NAME

list_options パラメーターは、指定した名前から始まるすべてのエントリーを表示する AP_LIST_INCLUSIVE に設定しましたが、次の条件のいずれかが適用されます。

- *fqplu_name* パラメーターは、このローカル LU のどのパートナーの名前とも一致しません。
- 指定したローカル LU、パートナー LU、およびモードの組み合わせに対して (ノードを最後に起動してから) 活動状態であったセッションはありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_MODE_DEFINITION

QUERY_MODE_DEFINITION は、DEFINE_MODE を使用して定義されたモード、または SNA 定義モードに関する情報を戻します。

この verb は使用するオプションに応じて、特定のモードまたは複数のモードに関する要約情報または詳細情報のを入手するために使用できます。この verb が戻す情報は、モードの定義であり、その現在の使用状況についてはありません。ローカル LU およびパートナー LU によるモードの現在の使用状況に関する情報を入手するには、QUERY_MODE を使用してください。

この verb は、認識されないモード名に使用されるデフォルト COS 名に関する情報を戻すために使用することはできません。それには QUERY_MODE_TO_COS_MAPPING を使用してください。

VCB 構造体

```
typedef struct query_mode_definition
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;               /* reserved */
    unsigned char  format;                /* reserved */
    AP_UINT16      primary_rc;            /* primary return code */
    AP_UINT32      secondary_rc;          /* secondary return code */
    unsigned char  *buf_ptr;              /* pointer to buffer */
    AP_UINT32      buf_size;              /* buffer size */
    AP_UINT32      total_buf_size;        /* total buffer size required */
    AP_UINT16      num_entries;           /* number of entries */
    AP_UINT16      total_num_entries;     /* total number of entries */
    unsigned char  list_options;          /* listing options */
    unsigned char  reserv3;               /* reserved */
    unsigned char  mode_name[8];         /* mode name */
} QUERY_MODE_DEFINITION;

typedef struct mode_def_summary
{
    AP_UINT16      overlay_size;          /* size of returned entry */
}
```

QUERY_MODE_DEFINITION

```
    unsigned char    mode_name[8];           /* mode name          */
    unsigned char    description[32];        /* resource description */
    unsigned char    reserv1[16];           /* reserved            */
} MODE_DEF_SUMMARY;

typedef struct mode_def_detail
{
    AP_UINT16        overlay_size;           /* size of returned entry */
    unsigned char    mode_name[8];          /* mode name            */
    MODE_CHARS       mode_chars;            /* mode characteristics  */
} MODE_DEF_DETAIL;

typedef struct mode_chars
{
    unsigned char    description[32];        /* resource description  */
    unsigned char    reserv2[16];          /* reserved              */
    AP_UINT16        max_ru_size_upper;     /* maximum RU size upper bound*/
    unsigned char    receive_pacing_win;    /* receive pacing window */
    unsigned char    default_ru_size;      /* default RU size to    */
    /* maximize performance */
    AP_UINT16        max_neg_sess_lim;      /* maximum negotiable session */
    /* limit */
    AP_UINT16        plu_mode_session_limit; /* LU-mode session limit */
    AP_UINT16        min_conwin_src;        /* minimum source contention */
    /* winner sessions */
    unsigned char    cos_name[8];           /* class of service name */
    unsigned char    cryptography;          /* cryptography (reserved) */
    unsigned char    compression;          /* data compression supported? */
    AP_UINT16        auto_act;              /* number of sessions to be */
    /* activated automatically */
    AP_UINT16        min_conloser_src;      /* minimum source contention */
    /* loser */
    AP_UINT16        max_ru_size_lower;     /* maximum RU size lower bound*/
    AP_UINT16        max_receive_pacing_win; /* maximum receive pacing */
    /* window */
    unsigned char    max_compress_lvl;      /* max level of data compression */
    unsigned char    max_decompress_lvl;    /* max level of data decompression */
    unsigned char    comp_in_series;        /* reserved */
    unsigned char    reserv4[25];          /* reserved */
} MODE_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_MODE_DEFINITION

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

モードの最大数 (そのモードのデータが戻されます)。ある範囲でなく、特定モードに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

mode_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

mode_name パラメーターで指定したエントリーの直後のエントリーから開始します。

アプリケーションでそのリストから特定のエントリーを入手する方法については、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。この verb は、モードを作成順に表示する点で他の QUERY_* verb と異なります。

mode_name

1 グループのセッションのネットワーク特性を指定するモード名。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

QUERY_MODE_DEFINITION

mode_def_summary.overlay_size

戻された `mode_def_summary` 構造体のサイズ。すなわち、データ・バッファ一内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 `mode_def_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

mode_def_summary.mode_name

モード名。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

mode_def_summary.description

モードを記述したヌルで終了するテキスト・String (そのモードの定義で指定したもの)。

mode_def_detail.overlay_size

戻された `mode_def_detail` 構造体のサイズ。すなわち、データ・バッファ一内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 `mode_def_detail` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

mode_def_detail.mode_name

モード名。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

mode_def_detail.mode_chars.description

モードを記述したヌルで終了するテキスト・String (そのモードの定義で指定したもの)。

mode_def_detail.mode_chars.max_ru_size_upp

このモード名のセッションで使用される最大 RU サイズの上限。この値は、セッション活動化中に最大 RU サイズを折衝する場合に使用されません。

範囲は 256 ~ 61,440 です。 `default_ru_size` パラメーター (以下を参照) を `AP_YES` に設定した場合は、このフィールドは無視されます。

mode_def_detail.mode_chars.receive_pacing_win

このモードを使用するセッションのセッション・ペーシング・ウィンドウ。固定ペーシングの場合、これはローカル LU が応答を送信する前にパートナー LU から受信できるフレームの最大数です。適応ペーシングの場合は、この値は受信ウィンドウの初期サイズとして使用されます。

Communications Server for Linux は、隣接ノードが適応ペーシングをサポートしないように指定されている場合を除き、常時、適応ペーシングを使用します。

範囲は 1 ~ 63、またはペーシング・ウィンドウなしの場合 (つまり、受信できるフレームの数の制限がなく、応答が必要でない場合) はゼロです。

mode_def_detail.mode_chars.default_ru_size

最大 RU サイズのデフォルト上限を使用するかどうかを指定します。値は次のいずれかです。

AP_YES Communications Server for Linux は *max_ru_size_upp* パラメーターを無視し、最大 RU サイズの上限をリンク BTU サイズに収まる最大値に設定します。

AP_NO Communications Server for Linux は、最大 RU サイズを定義するために *max_ru_size_upp* パラメーターを使用します。

mode_def_detail.mode_chars.max_neg_sess_lim

このモードでローカル LU とパートナー LU 間に許可されるセッションの最大数。範囲: 1 ~ 32,767、または暗黙の CNOS 交換がないことを指定するゼロ。

mode_def_detail.mode_chars.plu_mode_session_limit

このモードにおけるデフォルトのセッション限度。1 つのローカル LU とパートナー LU の間でこのモードで成立するセッションの最大数が、このパラメーターにより決まります。この値は CNOS (セッション数の変更) 交換が暗黙に開始されるときに使用されます。範囲: 1 ~ 32,767、または暗黙の CNOS 交換がないことを指定するゼロ。

mode_def_detail.mode_chars.min_conwin_src

このモードを使用するローカル LU が活動化できる競合勝者セッションの最小数。この値は CNOS (セッション数の変更) 交換が暗黙に開始されるときに使用されます。範囲: 1 ~ 32,767、または暗黙の CNOS 交換がないことを指定するゼロ。

mode_def_detail.mode_chars.cos_name

このモードでセッションを活動化するとき要求するサービス・クラスの名前。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

mode_def_detail.mode_chars.compression

このモードを使用して活動化されるセッションで圧縮が使用されるかどうかを指定します。値は次のいずれかです。

AP_COMP_PROHIBITED

圧縮は、このモードを使用するセッションではサポートされません。

AP_COMP_REQUESTED

圧縮は、このモードを使用するセッションでサポートおよび要求されます (ただし必須ではありません。圧縮は、パートナー の BIND が要求しない場合は使用されません)。

mode_def_detail.mode_chars.auto_act

このモードに自動的に活動化されるセッションの数を指定します。この値は

QUERY_MODE_DEFINITION

CNOS (セッション数の変更) 交換が暗黙に開始されるときに使用されます。この値は 0 ~ 32,767 の範囲です。

mode_def_detail.mode_chars.min_conloser_src

このモードを使用する際のローカル LU でも活動化できる競合敗者セッションの最小数。この値は CNOS (セッション数の変更) 交換が暗黙に開始されるときに使用されます。この値は 0 ~ 32,767 の範囲です。

mode_def_detail.mode_chars.max_ru_size_low

このモードを使用するセッションで送受信される RU の最大サイズの下限。

この値は、256 ~ 61,440 の範囲、または下限がないことを示すゼロです。

mode_def_detail.mode_chars.max_receive_pacing_win

このモードにおけるセッションの最大セッション・ペーシング・ウィンドウ。適応ペーシングの場合、この値は、セッションが認可する受信ペーシング・ウィンドウを制限するために使用されます。固定ペーシングの場合は、このパラメーターは使用されません。(Communications Server for Linux は、隣接ノードが適応ペーシングをサポートしないように指定されている場合を除き、常時、適応ペーシングを使用します。)

この値は、0 ~ 32,767 の範囲、またはペーシング・ウィンドウの下限がないことを示すゼロです。

mode_def_detail.mode_chars.max_compress_lvl

ローカル・ノードからのデータ・フローに関して Communications Server for Linux が折衝を行う最大圧縮レベルを指定します。値は次のいずれかです。

- AP_NONE
- AP_RLE_COMPRESSION
- AP_LZ9_COMPRESSION
- AP_LZ10_COMPRESSION

最大圧縮レベルを指定しない非拡張 BIND を使用して圧縮が折衝される場合、RLE 圧縮が使用されます。

mode_def_detail.mode_chars.max_decompress_lvl

ローカル・ノードからのデータ・フローに関して Communications Server for Linux が折衝を行う最大解凍レベルを指定します。値は次のいずれかです。

- AP_NONE
- AP_RLE_COMPRESSION
- AP_LZ9_COMPRESSION
- AP_LZ10_COMPRESSION

最大圧縮レベルを指定しない非拡張 BIND を使用して圧縮が折衝される場合、RLE 圧縮が使用されます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_MODE_NAME

list_options パラメーターが、指定した名前から始まるすべてのエン
トリーをリストするために、AP_LIST_INCLUSIVE に設定されていま
すが、*mode_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんで
した。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、
AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記
以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_MODE_TO_COS_MAPPING

QUERY_MODE_TO_COS_MAPPING は、特定のモードに関連した COS (サービ
ス・クラス) に関する情報を戻します。この verb は、使用するオプションに応じ
て、特定のモードまたは複数のモードに関する情報を入手するために使用できま
す。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_mode_to_cos_mapping
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    unsigned char  *buf_ptr;       /* pointer to buffer        */
    AP_UINT32      buf_size;       /* buffer size              */
    AP_UINT32      total_buf_size; /* total buffer size required */
    AP_UINT16      num_entries;    /* number of entries        */
    AP_UINT16      total_num_entries; /* total number of entries */
    unsigned char  list_options;   /* listing options          */
    unsigned char  reserv3;       /* reserved                  */
    unsigned char  mode_name[8];  /* mode name                 */
} QUERY_MODE_TO_COS_MAPPING;

typedef struct mode_to_cos_mapping_data
{
    AP_UINT16      overlay_size;   /* size of returned entry   */
    unsigned char  mode_name[8];  /* mode name                 */
    unsigned char  cos_name[8];   /* cos name                  */
    unsigned char  reserva[20];   /* reserved                  */
} MODE_TO_COS_MAPPING_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_MODE_TO_COS_MAPPING

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

モードの最大数 (そのモードのデータが戻されます)。ある範囲でなく、特定モードに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのモード・リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

mode_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

mode_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

mode_name

情報を求めるモードの名前、またはリストへの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されません。

この名前は、8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。認識されないモード名に使用されるデフォルト COS に関する情報を戻すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファ内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファ内の各エントリーは、次のパラメーターで構成されています。

mode_to_cos_mapping_data.overlay_size

戻された *mode_to_cos_mapping_data* 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *mode_to_cos_mapping_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

mode_to_cos_mapping_data.mode_name

モード名。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

mode_to_cos_mapping_data.cos_name

このモード名に関連したサービス・クラス名。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_MODE_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*mode_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

QUERY_MODE_TO_COS_MAPPING

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_NMVT_APPLICATION

QUERY_NMVT_APPLICATION は、MS verb の REGISTER_NMVT_APPLICATION を発行して NMVT レベルのメッセージについて登録されたアプリケーションのリストを戻します。この verb について詳しくは、「*Communications Server for Linux MS Programmer's Guide*」を参照してください。

この verb は、使用するオプションに応じて、特定のアプリケーションまたは複数のアプリケーションに関する情報を入手するために使用できます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_nmvt_application
{
    AP_UINT16      opcode;                /* Verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;         /* secondary return code */
    unsigned char  *buf_ptr;             /* pointer to buffer */
    AP_UINT32      buf_size;             /* buffer size */
    AP_UINT32      total_buf_size;       /* total buffer size required*/
    AP_UINT16      num_entries;          /* number of entries */
    AP_UINT16      total_num_entries;    /* total number of entries */
    unsigned char  list_options;         /* listing options */
    unsigned char  reserv3;              /* reserved */
    unsigned char  application[8];       /* application */
} QUERY_NMVT_APPLICATION;

typedef struct nmvt_application_data
{
    AP_UINT16      overlay_size;         /* size of returned entry */
    unsigned char  application[8];       /* application name */
    AP_UINT16      ms_vector_key_type;   /* MS vector key accepted
                                         /* by appl */
    unsigned char  conversion_required;  /* is conversion to MDS_MU
                                         /* required */
    unsigned char  reserv[5];           /* reserved */
    unsigned char  reserva[20];         /* reserved */
} NMVT_APPLICATION_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_NMVT_APPLICATION

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

アプリケーションの最大数 (そのアプリケーションのデータが戻されます)。ある範囲でなく、特定のアプリケーションに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのアプリケーション・リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

application パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

application パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

application

アプリケーション名。このパラメーターは、*list_options* を **AP_FIRST_IN_LIST** に設定した場合には無視されます。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファのサイズを示す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファ内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファ内の各エントリは、次のパラメーターで構成されています。

nmvt_application_data.overlay_size

戻された `nmvt_application_data` 構造体のサイズ。すなわち、データ・バッファ内において次のエントリの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 `nmvt_application_data` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

nmvt_application_data.application

登録済みアプリケーションの名前。この名前は、8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

nmvt_application_data.ms_vector_key_type

アプリケーションが受け入れる MS ベクトル・キー。アプリケーションは、NMVT メッセージに登録される時、受け入れる MS ベクトル・キーを指定します。

nmvt_application_data.conversion_required

着信メッセージを NMVT フォーマットから MDS_MU フォーマットへ変換することが登録されたアプリケーションに必要なかどうかを指定します。アプリケーションは、NMVT メッセージに登録される時、この変換が必要かどうかを指定します。値は次のいずれかです。

AP_YES 着信メッセージは MDS_MU フォーマットに変換されます。

AP_NO 着信メッセージは変換されません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_APPLICATION_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリをリストするために、**AP_LIST_INCLUSIVE** に設定されていますが、`application` パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_NN_TOPOLOGY_NODE

各ネットワーク・ノードは、すべてのネットワーク・ノード、仮想経路指定ノード (VRN)、およびネットワーク内のネットワーク・ノードからネットワーク・ノードへの TG に関する情報を保持するネットワーク・トポロジー・データベースを維持します。QUERY_NN_TOPOLOGY_NODE は、このデータベース内のネットワーク・ノード・エントリーおよび VRN エントリーに関する情報を戻します。

この verb は、使用するオプションに応じて、特定のノードまたは複数のノードに関する要約情報または詳細情報を入手するために使用できます。この verb はネットワーク・ノードに対してのみ発行でき、エンド・ノードまたは LEN ノードでは有効ではありません。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_nn_topology_node
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;          /* reserved                     */
    unsigned char  format;           /* reserved                     */
    AP_UINT16      primary_rc;       /* primary return code          */
    AP_UINT32      secondary_rc;     /* secondary return code        */
    unsigned char  *buf_ptr;         /* pointer to buffer            */
    AP_UINT32      buf_size;         /* buffer size                  */
    AP_UINT32      total_buf_size;   /* total buffer size required   */
    AP_UINT16      num_entries;      /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries      */
    unsigned char  list_options;     /* listing options              */
    unsigned char  reserv3;          /* reserved                     */
    unsigned char  node_name[17];    /* network qualified node name  */
    unsigned char  node_type;        /* node type                    */
    AP_UINT32      frsn;             /* flow reduction sequence number */
} QUERY_NN_TOPOLOGY_NODE;
```

frsn フィールドをゼロ以外の値に設定した場合は、指定した値に等しいか、それより大きい *FRSN* を持つノード・エントリーのみが戻されます。ゼロに設定した場合は、すべてのノード・エントリーが戻されます。

```
typedef struct nn_topology_node_summary
{
    AP_UINT16      overlay_size;     /* size of returned entry       */
    unsigned char  node_name[17];    /* network qualified node name  */
    unsigned char  node_type;        /* node type                    */
} NN_TOPOLOGY_NODE_SUMMARY;
```

```
typedef struct nn_topology_node_detail
{
    AP_UINT16      overlay_size;     /* size of returned entry       */
    unsigned char  node_name[17];    /* network qualified node name  */
}
```

QUERY_NN_TOPOLOGY_NODE

```
unsigned char    node_type;           /* node type */
AP_UINT16       days_left;           /* days left until entry purged */
unsigned char    reserv1[2];         /* reserved */
AP_UINT32       frsn;                /* flow reduction sequence number */
AP_UINT32       rsrn;                /* resource sequence number */
unsigned char    rar;                /* route additional resistance */
unsigned char    status;             /* node status */
unsigned char    function_support;   /* function support */
unsigned char    reserv2;            /* reserved */
unsigned char    branch_aware;       /* is the node branch aware? */
unsigned char    reserva[19];        /* reserved */
} NN_TOPOLOGY_NODE_DETAIL;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_NN_TOPOLOGY_NODE

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

ノードの最大数 (そのノードのデータが戻されます)。ある範囲でなく、特定のノードに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

node_name パラメーター、*node_type* パラメーター、および *frsn* パラメーターを組み合わせて指定したエントリーから開始します。

AP_LIST_FROM_NEXT

node_name、*node_type*、*frsn* の各パラメーターを組み合わせて指定したエントリーの直後のエントリーから開始します。

リストは、まず *node_name* の順、次に *node_type* の順 (AP_NETWORK_NODE、AP_VRN の順)、最後に *frsn* の値の順に並べられます。 リストの順序と、アプリケーションでそのリストから特定の

エントリーを入手する方法については、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

node_name

情報を求めるノードの完全修飾名、またはノードのリストの索引として使用する名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A スtring 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring 文字からなるネットワーク名で構成されます。

node_type

ノードのタイプ。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。値は次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_VRN 仮想経路指定ノード。

AP_LEARN_NODE

ノード・タイプは不明です。

frsn

フロー縮小シーケンス番号 (FRSN)。すべてのノードに関する情報を戻すためにゼロを指定するか、FRSN がこの値より大きいか等しいノードに関する情報を戻すためにゼロ以外の値を指定します。

このパラメーターを使用すると、すべての情報を入手するために複数の *verb* をアプリケーションで発行する必要がある場合に、整合性のある情報を入手ようにすることができます。アプリケーションで以下の手順を実行してください。

frsn パラメーターを使用して整合性のある情報を入手するためには、以下の手順を実行します。

1. QUERY_NODE を発行して、ノードの現行 FRSN を読み取ります。
2. *frsn* パラメーターをゼロに設定して、すべてのデータベース・エントリーを読み取るために必要な QUERY_NN_TOPOLOGY_NODE *verb* を発行します。
3. QUERY_NODE を再度発行し、新しい FRSN をステップ 1 で戻された FRSN と比較します。
4. 2 つの FRSN が異なっている場合は、データベースが変更されています。ステップ 1 で読み取った FRSN に 1 を加え、*frsn* パラメーターをこの新しい値に設定して、さらに QUERY_NN_TOPOLOGY_NODE *verb* を発行します。これらの *verb* は変更されたエントリーのみを戻します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

QUERY_NN_TOPOLOGY_NODE

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

nn_topology_node_summary.overlay_size

戻された *nn_topology_node_summary* 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *nn_topology_node_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

nn_topology_node_summary.node_name

ノードの完全修飾名。この名前は、17 バイトの EBCDIC スtringで、この右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A スtring文字からなるネットワーク名で構成されます。

nn_topology_node_summary.node_type

ノードのタイプ。次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_END_NODE

エンド・ノード。

AP_VRN 仮想経路指定ノード。

nn_topology_node_detail.node_name

ノードの完全修飾名。この名前は、17 バイトの EBCDIC スtringで、この右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A

ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

nn_topology_node_detail.node_type

ノードのタイプ。次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_END_NODE

エンド・ノード。

AP_VRN 仮想経路指定ノード。

nn_topology_node_detail.overlay_size

戻された *nn_topology_node_detail* 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *nn_topology_node_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

nn_topology_node_detail.days_left

このノード・エントリーをトポロジー・データベースから削除する前の日数。ローカル・ノード・エントリーの場合は、この値をゼロに設定し、このエントリーは削除されないことを示します。

nn_topology_node_detail.frsn

フロー縮小シーケンス番号 (FRSN)。ローカル・ノードでこのリソースが最後に更新された時刻を示します。

nn_topology_node_detail.rsn

リソース・シーケンス番号。この番号は、このリソースを所有しているネットワーク・ノードにより割り当てられます。

nn_topology_node_detail.rar

ノードの経路の追加抵抗。値は 0 ~ 255 の範囲です。

nn_topology_node_detail.status

ノードの状況を指定します。このパラメーターは、AP_UNCONGESTED、次に表示する値のいずれか、論理 OR で組み合わせた 2 つ以上の値に設定できます。値は次のいずれかです。

AP_UNCONGESTED

ISR セッションの数は、ノードの構成で *isr_sessions_upper_threshold* の値未満です。

AP_CONGESTED

ISR セッションの数は、しきい値を超えています。

QUERY_NN_TOPOLOGY_NODE

AP_IRR_DEPLETED

ISR セッションの数は、そのノードに指定した最大値に達しました。

AP_ERR_DEPLETED

エンドポイント・セッションの数は、指定した最大値に達しました。

AP QUIESCING

タイプが AP_QUIESCE または AP_QUIESCE_ISR の STOP_NODE が発行されました。

nn_topology_node_detail.function_support

サポートされる機能を指定します。値は次のいずれか 1 つ以上を論理 OR で組み合わせたものです。

AP_BORDER_NODE

ボーダー・ノード。

AP_EXTENDED_BORDER_NODE

戻りボーダー・ノード機能がサポートされます。

AP_CDS 中央ディレクトリー・サーバー。

AP_GATEWAY

ゲートウェイ・ノード。

AP_INTERCHANGE_NODE

交換ノード機能がサポートされます。

AP_ISR 中間セッション経路指定。

AP_HPR ノードは高性能経路指定 (High Performance Routing: HPR) の基本機能をサポートします。

AP_RTP_TOWER

ノードは HPR の高速トランスポート・プロトコル (Rapid Transport Protocol) タワーをサポートします。

AP_CONTROL_OVER_RTP_TOWER

ノードは高速トランスポート・プロトコル (Rapid Transport Protocol) タワーを介した HPR 制御フローをサポートします。

nn_topology_node_detail.branch_aware

ノードが分岐認識 (APPN オプション・セット 1120) をサポートするかどうかを指定します。

AP_NO ノードはオプション・セット 1120 をサポートしません。

AP_YES ノードはオプション・セット 1120 をサポートします。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_NODE

list_options パラメーターが、指定した名前から始まるすべてのエン
トリーをリストするために、AP_LIST_INCLUSIVE に設定されていま
すが、*node_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんで
した。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、
AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードがネットワーク・ノードでないために *verb* が正常に実行されな
いと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードがネットワーク・ノードではありません。この
verb はネットワーク・ノードでのみ使用できます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記
以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_NN_TOPOLOGY_STATS

QUERY_NN_TOPOLOGY_STATS は、トポロジー・データベースに関する統計情報
を戻します。これは Communications Server for Linux ノードがネットワーク・ノ
ードである場合のみ使用可能であり、このノードがエンド・ノードである場合は無効
です。

この *verb* は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_nn_topology_stats
{
    AP_UINT16    opcode;                /* verb operation code */
    unsigned char reserv2;              /* reserved */
    unsigned char format;               /* reserved */
    AP_UINT16    primary_rc;            /* primary return code */
    AP_UINT32    secondary_rc;          /* secondary return code */
    AP_UINT32    max_nodes;             /* max number of nodes in database */
    AP_UINT32    cur_num_nodes;         /* current number of nodes in
                                        /* database
    AP_UINT32    node_in_tdus;          /* number of TDUs received
    AP_UINT32    node_out_tdus;         /* number of TDUs sent
    AP_UINT32    node_low_rsns;         /* node updates received with low
                                        /* RSNs
    AP_UINT32    node_equal_rsns;       /* node updates in with equal RSNs */

```

QUERY_NN_TOPOLOGY_STATS

```
AP_UINT32    node_good_high_rsns; /* node updates in with high RSNs */
AP_UINT32    node_bad_high_rsns; /* node updates in with high and */
/* odd RSNs */
AP_UINT32    node_state_updates; /* number of node updates sent */
AP_UINT32    node_errors; /* number of node entry errors found*/
AP_UINT32    node_timer_updates; /* number of node records built */
/* due to timer updates */
AP_UINT32    node_purges; /* number of node records purged */
AP_UINT32    tg_low_rsns; /* TG updates received with low RSNs*/
AP_UINT32    tg_equal_rsns; /* TG updates in with equal RSNs */
AP_UINT32    tg_good_high_rsns; /* TG updates in with high RSNs */
AP_UINT32    tg_bad_high_rsns; /* TG updates in with high and */
/* odd RSNs */
AP_UINT32    tg_state_updates; /* number of TG updates sent */
AP_UINT32    tg_errors; /* number of TG entry errors found */
AP_UINT32    tg_timer_updates; /* number of node records built */
/* due to timer updates */
AP_UINT32    tg_purges; /* number of node records purged */
AP_UINT32    total_route_calcs; /* number of routes calculated */
/* for COS */
AP_UINT32    total_route_rejs; /* number of failed route */
/* calculations */
AP_UINT32    total_tree_cache_hits; /* total number of tree cache hits */
AP_UINT32    total_tree_cache_misses; /* total number of tree cache */
/* misses */
AP_UINT32    total_tdu_wars; /* total number TDU war detections */
unsigned char reserva[16]; /* reserved */
} QUERY_NN_TOPOLOGY_STATS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

```
opcode AP_QUERY_NN_TOPOLOGY_STATS
```

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

max_nodes

トポロジー・データベース内のノード・レコードの最大数。この値は DEFINE_NODE で指定されています。ゼロの値は制限がないことを示します。

cur_num_nodes

このノードのトポロジー・データベース内の現在のノード数。この値が許されるノードの最大数を超えた場合は、アラートが発行されます。

node_in_tdus

このノードが受信したトポロジー・データベース更新 (TDU) の合計数。

node_out_tdus

このノードが作成し、最後の初期化後にすべての隣接ネットワーク・ノードに送信されるトポロジー・データベース更新 (TDU) の合計数。

node_low_rsns

現行 RSN より少ない RSN でこのノードが受信したトポロジー・ノード更新の合計数。このカウントには偶数と奇数の両方の RSN が含まれます。

(これらの TDU はエラーではなく、TDU がすべての隣接ネットワーク・ノードへブロードキャストされる時の結果です。このノードのトポロジー・データベースの更新は行われませんが、このノードは、この小さい RSN を送信した隣接ノードへ、より大きい RSN の TDU を送信します。)

node_equal_rsns

現行 RSN に等しい RSN でこのノードが受信したトポロジー・ノード更新の合計数。このカウントには偶数と奇数の両方の RSN が含まれます。(これらの TDU はエラーではなく、TDU がすべての隣接ネットワーク・ノードへブロードキャストされる時の結果です。このノードのトポロジー・データベースの更新は行われません。)

node_good_high_rsns

現行 RSN より大きい RSN でこのノードが受信したトポロジー・ノード更新の合計数。ノードは、そのトポロジーを更新し、すべての隣接ネットワーク・ノードへ TDU をブロードキャストします。送信側のノードは既にこの更新情報を持っているため、送信側ノードへ TDU を送信する必要はありません。

node_bad_high_rsns

現行 RSN より大きい奇数 RSN でこのノードが受信したトポロジー・ノード更新の合計数。これらの更新は、APPN ネットワーク・ノードの 1 つによって検出されたトポロジーの不整合を表します。ノードは、そのトポロジーを更新し、すべての隣接ネットワーク・ノードへ TDU をブロードキャストします。

node_state_updates

APPN トポロジーおよび経路指定に影響を及ぼす内部的に検出されたノードの状態変更の結果として作成された、トポロジー・ノード更新の合計数。更新は TDU を介してすべての隣接ネットワーク・ノードへ送信されます。

node_errors

このノードが検出したトポロジー・ノード更新不整合の合計数。これは、このノードがそのトポロジー・データベースを更新しようとしてデータの不整合を検出したときに起こります。このノードは、現行 RSN を次の奇数に増やして TDU を作成し、すべての隣接ネットワーク・ノードへブロードキャストします。

node_timer_updates

タイマー更新のため、このノードのリソースについて作成されたトポロジー・ノード更新の合計数。更新は TDU を介してすべての隣接ネットワーク・ノードへ送信されます。これらの更新により、他のネットワーク・ノードはそのトポロジー・データベースからこのノードのリソースを削除しないようになります。

node_purges

このノードのトポロジー・データベースから除去されたトポロジー・ノード・レコードの合計数。これは、指定された時間内にノード・レコードが更新されなかった場合に起こります。ノードがネットワーク・トポロジーに保持しようとしているリソースの更新をブロードキャストするのは、リソースを所有するノードの役割です。

QUERY_NN_TOPOLOGY_STATS

tg_low_rsns

現行 RSN より小さい RSN でこのノードが受信したトポロジー TG 更新の合計数。このカウントには偶数と奇数の両方の RSN が含まれます。(これらの TDU はエラーではなく、TDU がすべての隣接ネットワーク・ノードへブロードキャストされるべきの結果です。このノードのトポロジー・データベースの更新は行われませんが、このノードは、この小さい RSN を送信した隣接ノードへ、より大きい RSN の TDU を送信します。)

tg_equal_rsns

現行 RSN に等しい RSN でこのノードが受信したトポロジー TG 更新の合計数。このカウントには偶数と奇数の両方の RSN が含まれます。(これらの TDU はエラーではなく、TDU がすべての隣接ネットワーク・ノードへブロードキャストされるべきの結果です。このノードのトポロジー・データベースの更新は行われません。)

tg_good_high_rsns

現行 RSN より大きい RSN でこのノードが受信したトポロジー TG 更新の合計数。ノードは、そのトポロジーを更新し、すべての隣接ネットワーク・ノードへ TDU をブロードキャストします。

tg_bad_high_rsns

現行 RSN より大きい奇数 RSN でこのノードが受信したトポロジー TG 更新の合計数。これらの更新は、APPN ネットワーク・ノードの 1 つによって検出されたトポロジーの不整合を表します。ノードは、そのトポロジーを更新し、すべての隣接ネットワーク・ノードへ TDU をブロードキャストします。

tg_state_updates

APPN トポロジーおよび経路指定に影響を及ぼす内部的に検出されたノードの状態変更の結果として作成された、トポロジー TG 更新の合計数。更新は TDU を介してすべての隣接ネットワーク・ノードへ送信されます。

tg_errors

このノードが検出したトポロジー TG 更新不整合の合計数。これは、このノードがそのトポロジー・データベースを更新しようとしてデータの不整合を検出したときに起こります。このノードは、現行 RSN を次の奇数に増やして TDU を作成し、すべての隣接ネットワーク・ノードへブロードキャストします。

tg_timer_updates

タイマー更新のためこのノードのリソースについて作成されたトポロジー TG 更新の合計数。更新は TDU を介してすべての隣接ネットワーク・ノードへ送信されます。これらの更新により、他のネットワーク・ノードはそのトポロジー・データベースからこのノードのリソースを削除しないようになります。

tg_purges

このノードのトポロジー・データベースから除去されたトポロジー TG レコードの合計数。これは、指定された時間内に TG レコードが更新されなかった場合に起こります。ノードがネットワーク・トポロジーに保持しようとしているリソースの更新をブロードキャストするのは、リソースを所有するノードの役割です。

total_route_calcs

最後の初期化以後にすべてのサービス・クラスについて計算された経路の数。

total_route_rejs

最後の初期化以後に計算できなかったすべてのサービス・クラスについての経路要求の数。

total_tree_cache_hits

キャッシュにある経路ツリーに従って行われた経路計算の数。経路によっては複数のツリーによる検査を必要とするため、この数は計算された経路の合計数より大きい場合があります。

total_tree_cache_misses

キャッシュにある経路指定ツリーが適用できなかったため、新しい経路指定ツリーの作成が必要となった経路計算の数。

total_tdu_wars

ローカル・ノードが検出し、防止した TDU war の数。

戻りパラメーター: サポートされていない機能

ローカル・ノードがネットワーク・ノードでないために `verb` が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

*primary_rc***AP_FUNCTION_NOT_SUPPORTED**

ローカル・ノードがネットワーク・ノードではありません。この `verb` はネットワーク・ノードでのみ使用できます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_NN_TOPOLOGY_TG

各ネットワーク・ノードは、すべてのネットワーク・ノード、VRN、およびネットワーク内のネットワーク・ノードからネットワーク・ノードへの TG に関する情報を保持するネットワーク・トポロジー・データベースを維持します。

QUERY_NN_TOPOLOGY_TG は、このデータベース内の TG エントリーに関する情報を戻します。

この `verb` は、使用するオプションに応じて、特定の TG または複数の TG に関する要約情報または詳細情報のいずれかを、入手するために使用できます。この `verb` はネットワーク・ノードに対してのみ発行でき、エンド・ノードまたは LEN ノードでは有効ではありません。

この `verb` は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```

typedef struct query_nn_topology_tg
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;          /* reserved                     */
    unsigned char  format;           /* reserved                     */
    AP_UINT16      primary_rc;       /* primary return code          */
    AP_UINT32      secondary_rc;     /* secondary return code        */
    unsigned char  *buf_ptr;         /* pointer to buffer            */
    AP_UINT32      buf_size;         /* buffer size                  */
    AP_UINT32      total_buf_size;   /* total buffer size required   */
    AP_UINT16      num_entries;      /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries      */
    unsigned char  list_options;     /* listing options              */
    unsigned char  reserv3;          /* reserved                     */
    unsigned char  owner[17];        /* node that owns the TG        */
    unsigned char  owner_type;       /* type of node that owns the TG */
    unsigned char  dest[17];         /* TG destination node          */
    unsigned char  dest_type;        /* TG destination node type     */
    unsigned char  tg_num;           /* TG number                    */
    unsigned char  reserv1;          /* reserved                     */
    AP_UINT32      frsn;             /* flow reduction sequence number */
} QUERY_NN_TOPOLOGY_TG;

typedef struct topology_tg_summary
{
    AP_UINT16      overlay_size;     /* size of returned entry       */
    unsigned char  owner[17];        /* node that owns the TG        */
    unsigned char  owner_type;       /* type of node that owns the TG */
    unsigned char  dest[17];         /* TG destination node          */
    unsigned char  dest_type;        /* TG destination node type     */
    unsigned char  tg_num;           /* TG number                    */
    unsigned char  reserv3[1];       /* reserved                     */
    AP_UINT32      frsn;             /* flow reduction sequence number */
} TOPOLOGY_TG_SUMMARY;

typedef struct topology_tg_detail
{
    AP_UINT16      overlay_size;     /* size of returned entry       */
    unsigned char  owner[17];        /* node that owns the TG        */
    unsigned char  owner_type;       /* type of node that owns the TG */
    unsigned char  dest[17];         /* TG destination node          */
    unsigned char  dest_type;        /* TG destination node type     */
    unsigned char  tg_num;           /* TG number                    */
    unsigned char  reserv3[1];       /* reserved                     */
    AP_UINT32      frsn;             /* flow reduction sequence number */
    AP_UINT16      days_left;        /* days left until entry purged */
    LINK_ADDRESS   dlc_data;         /* DLC signalling data          */
    AP_UINT32      rsn;              /* resource sequence number     */
    unsigned char  status;           /* tg status                    */
    TG_DEFINED_CHARS tg_chars;       /* TG characteristics           */
    unsigned char  subarea_number;   /* subarea number               */
    unsigned char  tg_type;          /* TG type                      */
    unsigned char  intersubnet_tg;   /* TG between subnets         */
    unsigned char  cp_cp_session_active; /* Are CP-CP sessions active? */
    unsigned char  branch_tg;       /* TG branch aware?            */
    unsigned char  multilink_tg;     /* reserved                     */
    unsigned char  appended_data_format; /* format of appended data     */
    unsigned char  appended_data_len; /* length of appended data     */
    unsigned char  reserva[9];       /* reserved                     */
} TOPOLOGY_TG_DETAIL;

typedef struct link_address
{
    unsigned char  format;           /* type of link address         */
}

```

```

unsigned char  reservel;          /* reserved          */
AP_UINT16     length;           /* length            */
unsigned char  address[32];     /* address           */
} LINK_ADDRESS;

```

TG_DEFINED_CHARS 構造体について詳しくは、135 ページの『DEFINE_LS』を参照してください。

frsn フィールドをゼロ以外の値に設定した場合は、指定した値に等しいかそれより大きい FRSN を持つノード・エントリーのみが戻されます。ゼロに設定した場合は、すべてのノード・エントリーが戻されます。

list_options パラメーターが詳細情報を指定すると、TG Descriptor CV が戻された情報に付加される場合があります。詳しくは、パラメーター

topology_tg_detail.appended_data_format および *topology_tg_detail.appended_data_len* の説明を参照してください。

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_NN_TOPOLOGY_TG

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

TG の最大数 (その TG のデータが戻されます)。ある範囲でなく、特定の TG に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

所有者、宛先、TG 番号、および FRSN を組み合わせて指定したエントリーから開始します。

AP_LIST_FROM_NEXT

所有者、宛先、TG 番号、および FRSN を組み合わせて指定したエントリーの直後のエントリーから開始します。

list_options パラメーターを AP_LIST_INCLUSIVE または AP_LIST_FROM_NEXT に設定した場合は、*owner*、*owner_type*、*dest*、*dest_type*、*tg_num*、および *frsn* のパラメーターの組み合わせは、TG のリストへの索引として使用されます。

リストは、*owner*、*owner_type* (AP_NETWORK_NODE、AP_VRN の順)、*dest*、*dest_type* (AP_NETWORK_NODE、AP_VRN の順)、*tg_num* (数値順)、最後に *frsn* (数値順) の順に並べられます。リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法については、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

owner TG を所有するノードの名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A スtring 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring 文字からなるネットワーク名で構成されます。

owner_type

TG を所有するノードのタイプ。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。値は次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_VRN 仮想経路指定ノード。

AP_LEARN_NODE

ノード・タイプは不明です。

dest この TG のための宛先ノードの名前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtring です。この名前は、最大 8 文字の A スtring 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring 文字からなるネットワーク名で構成されます。

dest_type

TG の宛先ノードのタイプ。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。値は次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_VRN 仮想経路指定ノード。

AP_LEARN_NODE

ノード・タイプは不明です。

tg_num TG に関連した番号。

frsn フロー縮小シーケンス番号 (FRSN)。すべての TG に関する情報を戻すため

にゼロを指定するか、FRSN がこの値より大きいか等しい TG に関する情報を戻すためにゼロ以外の値を指定します。

このパラメーターを使用すると、すべての情報を入手するために複数の `verb` をアプリケーションで発行する必要がある場合に、整合性のある情報を入手ようにすることができます。アプリケーションで以下の手順を実行してください。

`frsn` パラメーターを使用して整合性のある情報を入手するためには、以下の手順を実行します。

1. `QUERY_NODE` を発行して、ノードの現行 `FRSN` を読み取ります。
2. `frsn` パラメーターをゼロに設定して、すべてのデータベース・エントリーを読み取るために必要な `QUERY_NN_TOPOLOGY_TG verb` を発行します。
3. `QUERY_NODE` を再度発行し、新しい `FRSN` をステップ 1 で戻された `FRSN` と比較します。
4. 2 つの `FRSN` が異なっている場合は、データベースが変更されています。ステップ 1 で読み取った `FRSN` に 1 を加え、`frsn` パラメーターをこの新しい値に設定して、さらに `QUERY_NN_TOPOLOGY_TG verb` を発行します。これらの `verb` は変更されたエントリーのみを戻します。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

`primary_rc`
AP_OK

`buf_size`
指定したバッファーに戻された情報の長さ。

`total_buf_size`
要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 `buf_size` より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

`num_entries`
データ・バッファー内に戻されたエントリーの数。

`total_num_entries`
使用可能なエントリーの合計数。 `num_entries` より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

`topology_tg_summary.overlay_size`
戻された `topology_tg_summary` 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 `topology_tg_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の

QUERY_NN_TOPOLOGY_TG

sizeof() 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

topology_tg_summary.owner

TG を所有するノードの名前。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

topology_tg_summary.owner_type

TG を所有するノードのタイプ。値は次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_END_NODE

エンド・ノード。

AP_VRN 仮想経路指定ノード。

topology_tg_summary.dest

この TG のための宛先ノードの名前。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

topology_tg_summary.dest_type

TG の宛先ノードのタイプ。値は次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_END_NODE

エンド・ノード。

AP_VRN 仮想経路指定ノード。

topology_tg_summary.tg_num

TG に関連した番号。

topology_tg_summary.frsn

フロー縮小シーケンス番号 (FRSN)。ローカル・ノードでこのリソースが最後に更新された時刻を示します。

topology_tg_detail.overlay_size

戻された *topology_tg_detail* 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *topology_tg_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の sizeof() 演算子は使用してはいけません。これは、戻されたオーバーレイ

のサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

topology_tg_detail.owner

TG を所有するノードの名前。この名前は 17 バイトの EBCDIC スtring で、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String 文字からなるネットワーク名で構成されます。

topology_tg_detail.owner_type

TG を所有するノードのタイプ。値は次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_END_NODE

エンド・ノード。

AP_VRN 仮想経路指定ノード。

topology_tg_detail.dest

この TG のための宛先ノードの名前。この名前は 17 バイトの EBCDIC String で、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String 文字からなるネットワーク名で構成されます。

topology_tg_detail.dest_type

TG の宛先ノードのタイプ。値は次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_END_NODE

エンド・ノード。

AP_VRN 仮想経路指定ノード。

topology_tg_detail.tg_num

TG に関連した番号。

topology_tg_detail.frsn

フロー縮小シーケンス番号 (FRSN)。ローカル・ノードでこのリソースが最後に更新された時刻を示します。

topology_tg_detail.days_left

この TG エントリーをトポロジー・データベースから削除する前の日数。

topology_tg_detail.dlc_data.length

dest_type または *owner_type* が AP_VRN の場合は、このフィールドは次のフィールドの DLC アドレスの長さを指定します。それ以外の場合は、このフィールドは使用されません。

topology_tg_detail.dlc_data.address

dest_type または *owner_type* が AP_VRN の場合は、このフィールドは VRN

との接続の DLC アドレス (16 進数) を指定します。アドレスのバイト数は直前のフィールド (length) で指定され、このフィールドの残りのバイトは定義されません。それ以外の場合は、このフィールドは使用されません。

トークンリングまたはイーサネットの場合は、このフィールドは 6 バイトの MAC アドレスと 1 バイトのローカル SAP アドレスの 2 つの部分で構成されます。MAC アドレスのビット配列は予期したフォーマットと異なる場合があります。2 つのアドレス・フォーマット間の変換については、163 ページの『MAC アドレス内のビット順序』を参照してください。

Enterprise Extender (HPR/IP) の場合は、アドレス・フォーマットについて詳しくは 478 ページの『QUERY_LS』を参照してください。

topology_tg_detail.rsn

リソース・シーケンス番号。この番号は、このリソースを所有しているネットワーク・ノードにより割り当てられます。

topology_tg_detail.status

TG の状況を指定します。これは次のいずれか 1 つ以上を論理 OR 演算で組み合わせたものです。

AP_NONE
 AP_TG_OPERATIVE
 AP_TG QUIESCING
 AP_TG_CP_CP_SESSIONS
 AP_HPR
 AP_RTP

topology_tg_detail.tg_chars

TG 特性。これらのパラメーターについて詳しくは、135 ページの『DEFINE_LS』を参照してください。

topology_tg_detail.subarea_number

TG の宛先の所有者がサブエリア対応ノードである場合は、このサブエリア対応ノード上で TG に関連付けられたリンク・ステーションを所有するタイプ 4 またはタイプ 5 のサブエリア番号が、このパラメーターに入ります。それ以外の場合は、このパラメーターはすべて 2 進ゼロに設定されます。

topology_tg_detail.tg_type

TG のタイプ。値は次のいずれかです。

AP_APPN_OR_BOUNDARY_TG
 TG を基にした APPN TG または境界機能。

AP_INTERCHANGE_TG
 交換 TG。

AP_VIRTUAL_ROUTE_BASED_TG
 TG を基にした仮想経路。

AP_UNKNOWN
 TG のタイプは不明です。

topology_tg_detail.intersubnet_tg

TG がサブネットワーク間 TG であるかどうかを指定します。値は次のいずれかです。

AP_YES TG はサブネットワーク間 TG です。

AP_NO TG はサブネットワーク間 TG ではありません。

topology_tg_detail.cp_cp_session_active

所有側ノードのコンテンツ勝者 CP-CP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES CP-CP セッションは活動状態です。

AP_NO CP-CP セッションは活動状態ではありません。

AP_UNKNOWN

CP-CP セッションの状況は不明です。

topology_tg_detail.branch_tg

TG が分岐 TG であるかどうかを指定します。値は次のいずれかです。

AP_YES TG は分岐 TG です。

AP_NO TG は分岐 TG ではありません。

AP_UNKNOWN

TG のタイプは不明です。

topology_tg_detail.appended_data_format

この NOF VCB 構造体に付加されるデータのフォーマットを指定します。

パラメーター *topology_tg_detail.appended_data_len* を、付加されたデータが組み込まれていることを示すゼロ以外の値に設定すると、このパラメーターは以下の値に設定されます。

AP_TG_DESCRIPTOR_CV

付加されたデータには、SNA フォーマットで定義された TG Descriptor CV が含まれます。

topology_tg_detail.appended_data_len が、付加されたデータが組み込まれていないことを示すゼロの場合は、このパラメーターは予約済みです。

topology_tg_detail.appended_data_len

この NOF VCB 構造体に付加される TG Descriptor CV データの長さを指定します。この値がゼロに設定されている場合は、付加されたデータは組み込まれません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

QUERY_NN_TOPOLOGY_TG

AP_INVALID_TG

list_options パラメーターは、指定した名前から始まるすべてのエン
トリーを表示する AP_LIST_INCLUSIVE に設定しましたが、*tg_num*
パラメーターが有効ではありませんでした。

AP_INVALID_ORIGIN_NODE

list_options パラメーターは、指定した名前から始まるすべてのエン
トリーを表示する AP_LIST_INCLUSIVE に設定しましたが、*owner* パ
ラメーターが有効ではありませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんで
した。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、
AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

ローカル・ノードがネットワーク・ノードでないために verb が正常に実行されな
いと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードがネットワーク・ノードではありません。この
verb はネットワーク・ノードでのみ使用できます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記
以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_NODE

QUERY_NODE は、Communications Server for Linux ノードの定義に関する情報を
戻し、ノードが活動状態である場合はその状況に関する情報も戻します。

VCB 構造体

```
typedef struct query_node
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;               /* reserved */
    unsigned char  format;                /* reserved */
    AP_UINT16      primary_rc;            /* primary return code */
    AP_UINT32      secondary_rc;          /* secondary return code */
    CP_CREATE_PARMS cp_create_parms;      /* create parameters */
    AP_UINT32      up_time;                /* time since node started */
    AP_UINT32      mem_size;               /* reserved */
    AP_UINT32      mem_used;               /* reserved */
    AP_UINT32      mem_warning_threshold; /* reserved */
    AP_UINT32      mem_critical_threshold; /* reserved */
    unsigned char  nn_functions_supported; /* NN functions supported */
    unsigned char  functions_supported;    /* functions supported */
    unsigned char  en_functions_supported; /* EN functions supported */
    unsigned char  nn_status;              /* node status */
    AP_UINT32      nn_frsn;                /* NN flow reduction sequence */
}
```

```

/* number */
AP_UINT32    nn_rsn; /* Resource sequence number */
AP_UINT16    def_ls_good_xids; /* Good XIDS for defined link */
/* stations */
AP_UINT16    def_ls_bad_xids; /* Bad XIDS for defined link */
/* stations */
AP_UINT16    dyn_ls_good_xids; /* Good XIDS for dynamic link */
/* stations */
AP_UINT16    dyn_ls_bad_xids; /* Bad XIDS for dynamic link */
/* stations */
unsigned char dlur_release_level; /* Current DLUR release level */
unsigned char nns_dlur_served_lu_reg_supp; /* NNS supports DLUR-served */
/* LU registration? */
unsigned char nns_en_reg_diff_owning_cp; /* NNS supports option 1123? */
unsigned char reserva[17]; /* reserved */
unsigned char fq_nn_server_name[17]; /* fully qualified NN server */
/* name */
AP_UINT32    current_isr_sessions; /* number of ISR sessions */
unsigned char nn_functions2; /* further NN fns supported */
unsigned char branch_ntwk_arch_version; /* level of BrNN support */
unsigned char reservb[28]; /* reserved */
} QUERY_NODE;

typedef struct cp_create_parms
{
    AP_UINT16    crt_parms_len; /* length of CP_CREATE_PARMS */
    unsigned char description[32]; /* resource description */
    unsigned char reserv1[2]; /* reserved */
    unsigned char ms_support; /* MS API level */
    unsigned char queue_nmvt; /* queue/reject NMVTs */
    unsigned char ms_support; /* reserved */
    unsigned char queue_nmvt; /* reserved */
    unsigned char reserv3[12]; /* reserved */
    unsigned char node_type; /* node type */
    unsigned char fqcp_name[17]; /* fully qualified CP name */
    unsigned char cp_alias[8]; /* CP alias */
    unsigned char mode_to_cos_map_supp; /* mode to COS mapping support */
    unsigned char mds_supported; /* MDS and MS capabilities */
    unsigned char node_id[4]; /* node ID */
    AP_UINT16    max_locates; /* maximum locates node can process */
    AP_UINT16    dir_cache_size; /* directory cache size */
    AP_UINT16    max_dir_entries; /* maximum directory entries */
/* (0 means unlimited) */
    AP_UINT16    locate_timeout; /* locate timeout in seconds */
    unsigned char reg_with_nn; /* register resources with NNS */
    unsigned char reg_with_cds; /* register resources with CDS */
    AP_UINT16    mds_send_alert_q_size; /* size of MDS send alert queue */
    AP_UINT16    cos_cache_size; /* number of cos definitions */
    AP_UINT16    tree_cache_size; /* Topology Database routing tree */
/* cache size */
    AP_UINT16    tree_cache_use_limit; /* number of times a tree can be */
/* used */
    AP_UINT16    max_tdm_nodes; /* max number of nodes that can be */
/* stored in Topology Database */
    AP_UINT16    max_tdm_tgs; /* max number of TGs that can be */
/* stored in Topology Database */
    AP_UINT32    max_isr_sessions; /* maximum ISR sessions */
    AP_UINT32    isr_sessions_upper_threshold; /* upper threshold for ISR */
/* sessions */
    AP_UINT32    isr_sessions_lower_threshold; /* lower threshold for ISR */
/* sessions */
    AP_UINT16    isr_max_ru_size; /* max RU size for ISR */
    AP_UINT16    isr_rcv_pac_window; /* ISR receive pacing window size */
    unsigned char store_endpt_rscvs; /* endpoint RSCV storage */
    unsigned char store_isr_rscvs; /* ISR RSCV storage */
    unsigned char store_dlur_rscvs; /* DLUR RSCV storage */
    unsigned char dlur_support; /* is DLUR supported? */
    unsigned char pu_conc_support; /* is PU conc supported? */
}

```

QUERY_NODE

```
unsigned char nn_rar; /* route additional resistance */
unsigned char hpr_support; /* level of HPR support */
unsigned char mobile; /* reserved */
unsigned char discovery_support; /* reserved */
unsigned char discovery_group_name[8]; /* reserved */
unsigned char implicit_lu_0_to_3; /* reserved */
unsigned char default_preference; /* reserved */
unsigned char anynet_supported; /* reserved */
AP_UINT16 max_ls_exception_events; /* max # exception entries */
unsigned char reserv2[1]; /* reserved */
unsigned char max_compress_lvl; /* Max compresssion level (reserved)*/
unsigned char node_spec_data_len; /* reserved */
unsigned char ptf[64]; /* program temporary fix array */
unsigned char cos_table_version; /* version of COS tables to use */
unsigned char send_term_self; /* default PLU-SLU session term */
unsigned char disable_branch_awareness; /* disable BrNN awareness */
unsigned char cplu_syncpt_support; /* syncpoint support on CP LU? */
unsigned char cplu_attributes; /* attributes for CP LU */
unsigned char reserved[95]; /* reserved */
} CP_CREATE_PARMS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_NODE

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

cp_create_parms.crt_parms_len

パラメーター作成構造体の長さ。

cp_create_parms.description

ノードを記述したヌルで終了するテキスト・ストリング (そのノードの定義で指定したもの)。

cp_create_parms.node_type

ノードのタイプ。値は次のいずれかです。

AP_NETWORK_NODE

AP_BRANCH_NETWORK_NODE

AP_END_NODE

AP_LEN_NODE

cp_create_parms.fqcp_name

ノードの完全修飾名。この名前は、17 バイトの EBCDIC ストリングで、この右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

cp_create_parms.cp_alias

ローカル使用の CP 別名。この名前は、8 バイトからなる ASCII ストリングです。8 バイトはすべて有効です。

cp_create_parms.mode_to_cos_map_supp

モードから COS へのマッピングをノードでサポートするかどうかを指定します。ネットワーク・ノードの場合は、モードから COS へのマッピングは常にサポートされるため、このパラメーターは無視されます。LEN ノードの場合、モードから COS へのマッピングはサポートされません。値は次のいずれかです。

AP_YES モードから COS へのマッピングをサポートします。

AP_NO モードから COS へのマッピングをサポートしません。

cp_create_parms.mds_supported

Management Services で複数ドメイン・サポート (MDS) および MS 機能をサポートするかどうかを指定します。値は次のいずれかです。

AP_YES MDS はサポートされます。

AP_NO MDS はサポートされません。

cp_create_parms.node_id

XID 交換で使用されるノード ID。これは 4 バイトの 16 進数ストリングです。

cp_create_parms.max_locates

ノードが処理できる検索の最大数。

cp_create_parms.dir_cache_size

ネットワーク・ノードのみ: ディレクトリー・キャッシュのサイズ。

cp_create_parms.max_dir_entries

ディレクトリー・エントリーの最大数。ゼロは制限がないことを示します。

cp_create_parms.locate_timeout

ネットワーク検索がタイムアウトになる前の時間 (秒単位) を指定します。ゼロはタイムアウトにならないことを示します。

cp_create_parms.reg_with_nn

エンド・ノードのみ: ノードの起動時にノードのリソースをネットワーク・ノード・サーバーに登録するかどうかを指定します。値は次のいずれかです。

AP_YES リソースを NN に登録します。エンド・ノードのネットワーク・ノード・サーバーは、エンド・ノードに有向検索要求のみを転送します。

AP_NO リソースを登録しません。ネットワーク・ノード・サーバーはすべてのブロードキャスト検索をエンド・ノードに転送します。

cp_create_parms.reg_with_cds

エンド・ノード: ネットワーク・ノード・サーバーがエンド・ノード・リソースを中央ディレクトリー・サーバーに登録できるかどうかを指定します。*reg_with_nn* を **AP_NO** に設定した場合は、このフィールドは無視されます。

ネットワーク・ノード: ローカル・リソースまたはドメイン・リソースをオプションで中央ディレクトリー・サーバー (CDS) に登録できるかどうかを指定します (AP_YES または AP_NO)。

値は次のいずれかです。

AP_YES リソースを CDS に登録します。

AP_NO リソースを登録しません。

cp_create_parms.mds_send_alert_q_size

MDS 送信アラート・キューのサイズ。キューに入れられたアラートの数がこの限度に達すると、Communications Server for Linux はキューの最も古いアラートを削除します。

cp_create_parms.cos_cache_size

ネットワーク・ノード: COS データベースの重みキャッシュのサイズ (必要な COS 定義の最大数)。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.tree_cache_size

ネットワーク・ノード: トポロジー・データベースの経路指定ツリー・キャッシュのサイズ。最小値は 8。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.tree_cache_use_limit

ネットワーク・ノード: キャッシュに入れられたツリーの最大使用数。この数を超えると、ツリーは廃棄され、再計算されます。これにより、ノードは等しい重みの経路間でセッションを平衡にさせることができます。この値を小さくすると、ロード・バランシングが改善されますが、活動化の待ち時間が増大します。使用の最小数は 1 です。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.max_tdm_nodes

ネットワーク・ノード: トポロジー・データベースに保管できるノードの最大数。0 (ゼロ) の値は、ノードの数に制限がないことを示します。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.max_tdm_tgs

ネットワーク・ノード: トポロジー・データベースに保管できる TG の最大数。0 (ゼロ) の値は、ノードの数に制限がないことを示します。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.max_isr_sessions

ネットワーク・ノード: ノードが同時に参加できる ISR セッションの最大数。100 を超える値を指定した場合を除き、Communications Server for Linux は値 100 を使用します。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.isr_sessions_upper_threshold および

cp_create_parms.isr_sessions_lower_threshold

ネットワーク・ノード: これらのしきい値は、ノードの輻輳状況を制御します。輻輳状況はネットワーク内の他のノードへ報告され、経路の計算に使用されます。ISR セッションの数が上限しきい値を超えた場合、ノードの状態は非輻輳から輻輳に変わります。ISR セッションの数が下限しきい値よ

り低くなると、ノードの状態は非輻輳に戻ります。エンド・ノードまたは LEN ノードの場合は、これらのパラメーターは予約済みです。

cp_create_parms.isr_max_ru_size

ネットワーク・ノード: 中間セッション用にサポートされる最大 RU サイズ。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.isr_rcv_pac_window

ネットワーク・ノード: 中間セッションで推奨される、1 ~ 63 の範囲の受信ペーシング・ウィンドウ・サイズ。隣接ノードが適応ペーシングをサポートしていない場合、この値は中間セッションの 2 次ホップでのみ使用されます。エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.store_endpt_rscvs

RSCV を診断に使用するために保管するかどうかを指定します。値は次のいずれかです。

AP_YES RSCV を保管します。

AP_NO RSCV を保管しません。

このフィールドを AP_YES に設定した場合は、RSCV は QUERY_SESSION verb で戻されます。(この値を AP_YES に設定すると、それぞれのエンドポイント・セッションごとに RSCV が格納されることとなります。この追加ストレージは、1 セッションあたり最大 256 バイトです。)

cp_create_parms.store_isr_rscvs

ネットワーク・ノード: RSCV を診断に使用するために保管するかどうかを指定します (AP_YES または AP_NO)。このフィールドを AP_YES に設定した場合は、RSCV は QUERY_ISR_SESSION verb で戻されます。(この値を AP_YES に設定すると、それぞれの ISR セッションごとに RSCV が格納されることとなります。この追加ストレージは、1 セッションあたり最大 256 バイトです。) エンド・ノードまたは LEN ノードの場合、このパラメーターは予約済みです。

cp_create_parms.store_dlur_rscvs

RSCV を診断に使用するために保管するかどうかを指定します (AP_YES または AP_NO)。このフィールドを AP_YES に設定した場合は、RSCV は QUERY_DLUR_LU verb で戻されます。(この値を AP_YES に設定した場合は、PLU-SLU セッションごとに 1 つの RSCV が保管されることを示しています。この追加ストレージは、1 セッションあたり最大 256 バイトです。)

cp_create_parms.dlur_support

DLUR がサポートされるかどうかを指定します。LEN ノードの場合は、このパラメーターは予約済みです。値は次のいずれかです。

AP_YES DLUR は使用できます。

AP_LIMITED_DLUR_MULTI_SUBNET | AP_YES

エンド・ノードまたは分岐ネットワーク・ノードの場合: DLUR はサポートされますが、別のサブネットにある DLUS への接続には使用されません。

この値はネットワーク・ノードではサポートされていません。

AP_NO DLUR は使用できません。

cp_create_parms.pu_conc_support

SNA ゲートウェイがサポートされるかどうかを指定します (AP_YES または AP_NO)。

cp_create_parms.nn_rar

ネットワーク・ノードの経路の追加抵抗。

cp_create_parms.hpr_support

ノードが提供する HPR (高性能経路指定) サポートのレベルを指定します。値は次のいずれかです。

AP_NONE

HPR はサポートされません。

AP_BASE

このノードは、自動ネットワーク経路指定 (ANR) を実行できますが、HPR のための RTP (高速トランスポート・プロトコル) エンドポイントとして機能することはできません。

AP_RTP

このノードは、自動ネットワーク経路指定 (ANR) を実行でき、HPR セッションのための RTP (高速トランスポート・プロトコル) エンドポイントとして機能することができます。

AP_CONTROL_FLOWS

このノードは、制御フローを含め、すべての HPR 機能を実行できます。

cp_create_parms.max_ls_exception_events

ノードが記録した LS 例外イベントの最大数。

cp_create_parms.ptf

将来のプログラム一時修正 (PTF) 操作を構成して制御するための次のような配列。

cp_create_parms.ptf[0]

REQDISCONT サポートおよび必須検索状況サポート。

Communications Server for Linux は、通常、REQDISCONT を使用してセッション・トラフィックで不要になった限定リソース・ホスト・リンクを非活性化します。このバイトを使用して、REQDISCONT の使用を抑止したり、Communications Server for Linux が送信した REQDISCONT 要求で使用された設定を変更したりすることができます。値は次のいずれかです。

AP_NONE

通常の REQDISCONT サポートを使用します。

AP_SUPPRESS_REQDISCONT

REQDISCONT を使用しません。

AP_OVERRIDE_REQDISCONT

変更された REQDISCONT サポートを使用します。REQDISCONT を指定する場合は、論理 OR 演算を使用して次の値の 1 つまたは両方と結合する必要があります。

AP_REQDISCONT_TYPE

REQDISCONT でタイプ「immediate (即時)」を使用します。この値を指定しなかった場合、Communications Server for Linux はタイプ「normal (標準)」を使用します。

AP_REQDISCONT_RECONTACT

REQDISCONT でタイプ「immediate recontact (即時に再接続する)」を使用します。この値を指定しなかった場合は、Communications Server for Linux はタイプ「no immediate recontact (即時に再接続しない)」を使用します。

AP_ALLOW_BB_RQE

Communications Server for Linux は、通常、要求で方向転換 (CD) も示す必要のある SNA プロトコルにホストが従っている場合を除き、ホストからのいずれの開始ブラケット (BB) 例外 (RQE) 要求もセンス・コード 2003 でリジェクトします。このフラグを設定すると、Communications Server for Linux は、このプロトコルに従っていないホストとのセッションを続行できます。

Communications Server for Linux は、エンド・ノードまたは分岐ネットワーク・ノードとして実行しているときに、そのネットワーク・ノード・サーバー (NNS) からのネットワーク検索を送信勧誘するかどうか選択することが可能です。ネットワーク全体のブロードキャスト検索の処理速度が遅くなるので、ネットワーク検索を要求するのは望ましくありません。しかし、ローカル・ノードがそのすべてのリソース (LU) をその NNS に登録できない場合は、検索を要求することが、ネットワークでこれらのリソースを可視にする唯一の方法になります。

通常、Communications Server for Linux はすべての LU を登録できるかどうかを判別して、その NNS からのネットワーク検索を的確に要求します。このノードが、LU が通常と異なる方法でネットワークにアクセスできるようにする場合 (たとえば、他のノードのゲートウェイとして機能している場合) には、上記の値を以下の値と組み合わせて標準操作を変更します。

AP_SET_SEARCH_STATUS

NNS からのネットワーク検索を無条件で要求します。

cp_create_parms.ptf[1]

ERP サポート。通常、Communications Server for Linux は ACTPU(ERP) を ERP として処理します。これにより、PU-SSCP セッションはリセットされますが、付随する LU-SSCP セッションや PLU-SLU セッションを暗黙的に非活動化することはありません。インプリメントされた SNA によっては、ACTPU(ERP) を ACTPU(cold) として処理し、付随する LU-SSCP セッションや PLU-SLU セッションを暗黙に非活動化する場合があります。値は次のいずれかです。

AP_NONE

通常の処理を使用します。

AP_OVERRIDE_ERP

すべての ACTPU 要求を ACTPU(cold) として処理します。

cp_create_parms.ptf[2]

BIS サポート。通常、Communications Server for Linux は、限定リソース LU 6.2 セッションを非活動化する前に、BIS プロトコルを使用します。値は次のいずれかです。

AP_NONE

通常の処理を使用します。

AP_SUPPRESS_BIS

BIS プロトコルは使用しません。限定リソース LU 6.2 セッションは UNBIND(cleanup) を使用して即時に非活動化されます。

cp_create_parms.ptf[3]

APINGD サポート。通常、Communications Server for Linux には、APING 接続テスター用のパートナー・プログラムが組み込まれます。このバイトを使用すると、ノードに到着する APING プログラムによる要求が自動的に処理されないように、ノード内で APING デーモンを使用不可にすることができます。値は次のいずれかです。

AP_NONE

ノード内に APINGD サポートを組み込みます (通常の処理)。

AP_EXTERNAL_APINGD

ノード内で APINGD を使用不可にします。

cp_create_parms.ptf[4]

LU 0-3 の RU の検査。このバイトを使用して、非標準 SNA データを送信するホスト・システムに予備手段を提供します。このバイトは、以下で説明する特定の問題が発生しない限り、AP_NONE に設定する必要があります。

値 AP_NONE は、LU 0-3 RU での Communications Server for Linux の通常の検査を指示します。

LU 0-3 RU での特定の検査が終了すると、以下の値が戻されます。

AP_ALLOW_BB_RQE

SNA プロトコルでは、LU 0-3 の PLU-SLU セッションの BB !EB RU は、RQD でなければならないと指定されています。いくつかのホストは、RQE BB !EB CD を送信します。これはプロトコル違反ですが、Communications Server for Linux では常に許容します。この値を設定すると、Communications Server for Linux は RQE BB !EB !CD EC RU も許容します。

cp_create_parms.ptf[5]

受信した Attach のセキュリティー検査。

ローカルの呼び出し可能 TP が、会話セキュリティーを必要としないように定義されている場合、あるいは定義されていないために会話セキュリティーを必要としないデフォルトが使用される場合には、呼び出し側 TP は、ローカルの呼び出し可能 TP にアクセスするために、ユーザー ID とパスワードを送信する必要はありません。呼び出し側 TP にこれらのパラメーターが設定されていて、Communications Server for Linux が受信する Attach メッセージにそのパラメーターが含まれている場合、Communications Server for Linux は、通常、呼び出し可能 TP が会話セキュリティーを必要としていなくてもパラメーターを検査します (パラメーターが無効な場合には、Attach

をリジェクトします)。このパラメーターを使用すると、検査を使用不可にすることができます。値は次のいずれかです。

AP_NONE

セキュリティー・パラメーターが受信した Attach に組み込まれている場合には、呼び出し可能 TP のセキュリティー要件に関係なく、これらのパラメーターを常に検査します (通常の処理)。

AP_LIMIT_TP_SECURITY

呼び出し可能 TP がセキュリティーを必要としない場合には、受信した Attach でのセキュリティー・パラメーターを検査しません。

cp_create_parms.ptf[6]

HPR の RTP オプション。

値 AP_NONE は、Communications Server for Linux の通常の RTP 処理を指示します。

カスタマイズ済みの RTP 操作の場合は、以下の値が戻されます。

AP_FORCE_STANDARD_ARB

通常、Communications Server for Linux は、標準 ARB アルゴリズムと ARB 応答可能モード・アルゴリズムの両方を RTP 接続でサポートすることを公示します。この値を設定すると、Communications Server for Linux は標準 ARB アルゴリズムのサポートのみを公示します。

cp_create_parms.ptf[7]

DACTLU での DLUR アンバインド。通常、Communications Server for Linux は、DLUR を使用しているセッションのホストからの DACTLU を受信しても、PLU-SLU セッションを終了しません。このパラメーターを使用すると、PLU-SLU セッションを強制終了することができます。値は次のいずれかです。

AP_NONE

通常の処理を使用します。

AP_DLUR_UNBIND_ON_DACTLU

DLUR を使用したセッションで DACTLU を受信すると、PLU-SLU セッションを終了します。

cp_create_parms.ptf[8]

REQACTPU で PU 名を抑止します。通常、Communications Server for Linux は、DLUR PU を活動化するときに REQACTPU メッセージで PU 名を識別します。値は次のいずれかです。

AP_NONE

通常の処理を使用します。

AP_SUPPRESS_PU_NAME_ON_REQACTPU

DLUR PU を活動化するときに PU 名を抑止します。

cp_create_parms.ptf[9]

RUI ブラケット・レース・オプションおよび限定リソースによって、接続ネットワークのオプション、および TCP/IP 情報制御ベクトル・オプションが上書きされます。

RUI アプリケーションでブラケット・プロトコルを使用し、RUI アプリケーションが既に BB (開始ブラケット) を送信した後にホストが BB を送信した場合、Communications Server for Linux は、通常、センス・データ 0813 でこれをリジェクトし、BB をアプリケーションに渡しません。値は次のいずれかです。

AP_NONE

通常の処理を使用します。

AP_LUA_PASSTHRU_BB_RACE

BB を RUI アプリケーションに渡します。アプリケーションは、センス・データ 0813 または 0814 のいずれかを持つ否定応答を送信する必要があります。

接続ネットワークを使用する Communications Server for Linux 内のリンクは、通常、限定リソースです。以下の値でこのデフォルトをオーバーライドします。

AP_CN_OVERRIDE_LIM_RES

それぞれの接続ネットワーク・リンクに関連付けられているポートで *implicit_limited_resource* パラメーターを使用し、限定リソースであるかどうかを判別します。

通常、Communications Server for Linux は、TN3270 または LUA セッション用のホストへの NOTIFY 要求に TCP/IP 情報制御ベクトル (0x64) を含めます。このベクトルにはホスト・コンソールで表示できる情報、またはホストで使用される情報 (たとえば、請求情報: クライアントによって使用される TCP/IP アドレスとポート番号、およびクライアントのアドレスに対応した IP 名) が含まれています。TN3270 の場合、通常、TN3270 サーバーはドメイン・ネーム・サーバー (DNS) ルックアップを実行して、クライアント IP 名を判別します。

クライアント・アドレスが IPv6 アドレスであっても、ホストで IPv6 アドレスを解釈できないバックレベル・バージョンの VTAM を稼働している場合は、クライアント・アドレスがホスト・コンソール上で正しく表示されないことがあります。

この動作は以下のフラグでオーバーライドできます。

AP_NO_TCPIP_VECTOR

TCP/IP 情報制御ベクトル (0x64) を TN3270 または LUA 用のホストへの NOTIFY 要求に含めません。

ホストがこの制御ベクトルをサポートしない古いバージョンの VTAM を稼働している場合は、この値を使用します。

AP_NO_TCPIP_NAME

DNS ルックアップを実行しないで、IP 名ではなくクライアント IP アドレスを指定して CV64 制御ベクトルを送信します。

この値は TN3270 のみに適用します。したがって、LUA クライアントには DNS ルックアップは必要ありません。この値は DNS 環境が遅いか、またはクライアントが DNS データに含まれていないことが分かっている場合 (たとえば、クライアントが DDNS のない DHCP クライアントの場合) に使用します。

cp_create_parms.cos_table_version

ノードが使用する COS テーブルのバージョンを指定します。値は次のいずれかです。

AP_VERSION_0_COS_TABLES

本来、「APPN Architecture Reference」で定義されている COS テーブルを使用します。

AP_VERSION_1_COS_TABLES

ATM を介して本来 HPR に定義されている COS テーブルを使用します。

cp_create_parms.send_term_self

ホストに対する PLU-SLU セッションを終了するデフォルトの方式を指定します。指定した値は、LU 定義に異なる値を指定して変更しない限り、ノード上の 0-3 のすべてのタイプの LU で使用されます。値は次のいずれかです。

AP_YES CLOSE_PLU_SLU_SEC_RQ の受信時に TERM_SELF を送信します。

AP_NO CLOSE_PLU_SLU_SEC_RQ の受信時に UNBIND を送信します。

cp_create_parms.disable_branch_awareness

このパラメーターは、*node_type* が AP_NETWORK_NODE の場合にのみ適用されます。他のノード・タイプの場合は予約済みです。

ローカル・ノードが分岐認識 (APPN オプション・セット 1120) をサポートするかどうかを指定します。値は次のいずれかです。

AP_YES ローカル・ノードは分岐認識をサポートしません。このノードと実行済み分岐ネットワーク・ノードの間の TG は、ネットワーク・トポロジーには現れないため、ローカル・ノードはローカル・ノード自体が分岐として認識されているとは報告しません。

AP_NO ローカル・ノードは分岐認識をサポートします。

cp_create_parms.cplu_syncpt_support

ノードの制御点 LU が同期点機能をサポートするかどうかを指定します。このパラメーターは、DEFINE_LOCAL_LU の *syncpt_support* パラメーターと同等のものですが、ノードの制御点 LU (明示的な LU 定義を持たないもの) にのみ適用されます。

このパラメーターは、標準 Communications Server for Linux 製品のほかに同期点マネージャー (SPM) と会話保護リソース・マネージャー (C-PRM) が組み込まれている場合にのみ AP_YES に設定してください。値は次のいずれかです。

AP_YES 同期点はサポートされます。

AP_NO 同期点はサポートされません。

cp_create_parms.cplu_attributes

ノードの制御点 LU に関する追加情報を識別します。このパラメーターは、DEFINE_LOCAL_LU の *lu_attributes* パラメーターと同等のものですが、ノードの制御点 LU (明示的な LU 定義を持たないもの) にのみ適用されます。

値は次のいずれかです。

AP_NONE

追加情報は識別されません。

AP_DISABLE_PWSUB

制御点 LU のパスワード置換サポートを使用不可にします。パスワード置換とは、パスワードを平文で送信せずに、ローカル LU とリモート LU の間での伝送を行う前にパスワードを暗号化することを言います。通常、Communications Server for Linux は、リモート・システムがパスワード置換をサポートしている場合、パスワード置換を使用します。

この値は、パスワード置換を正しくインプリメントしていないリモート・システムと通信するための予備手段として提供されています。このオプションを使用する場合、平文でのパスワードの送受信(セキュリティ上のリスクがある場合があります)が行われることに注意しなければなりません。このオプションは、リモート・システムにおけるパスワード置換のインプリメンテーションに関して問題がある場合以外は設定しないでください。

up_time

ノードの始動 (または再始動) 以後の時間 (0.01 秒単位)。ゼロの値はノードが稼働中でないことを示します。

nn_functions_supported

ネットワーク・ノードのみ: サポートされるネットワーク・ノード機能を指定します。値は次のいずれか 1 つ以上を論理 OR で組み合わせたものです。

AP_RCV_REG_CHAR

ノードは登録済み特性の受信をサポートします。

AP_GATEWAY

ノードはゲートウェイ・ノードです。

AP_CDS ノードは中央ディレクトリー・サーバー機能をサポートします。

AP_TREE_CACHING

ノードは経路ツリー・キャッシュをサポートします。

AP_TREE_UPDATES

ノードは増分ツリー更新をサポートします。これをサポートする場合は、ツリー・キャッシュもサポートする必要があります。

AP_ISR ノードは中間セッション経路指定をサポートします。

functions_supported

サポートされる機能を指定します。値は次のいずれか 1 つ以上を論理 OR で組み合わせたものです。

AP_NEGOTIABLE_LS

AP_SEGMENT_REASSEMBLY

AP_BIND_REASSEMBLY

AP_PARALLEL_TGS

AP_CALL_IN

AP_ADAPTIVE_PACING

AP_TOPOLOGY_AWARENESS

en_functions_supported

エンド・ノードのみ: サポートされるエンド・ノード機能を指定します。値は次のいずれか 1 つ以上を論理 OR で組み合わせたものです。

AP_SEGMENT_GENERATION

ノードはセグメント生成をサポートします。

AP_MODE_TO_COS_MAP

ノードはモード名から COS 名へのマッピングをサポートします。

AP_LOCATE_CDINIT

ノードはリモート LU を見つけるために、検索およびドメイン間開始 GDS 変数の生成をサポートします。

AP_REG_WITH_NN

ノードはその LU を隣接サービス・ネットワーク・ノードに登録します。

AP_REG_CHARS_WITH_NN

ノードは送信レジスター特性をサポートします。この機能をサポートする場合は、送信レジスター名もサポートする必要があります。

nn_status

ネットワーク・ノードのみ: ノードの状況を指定します。このパラメータは、AP_UNCONGESTED、次に表示する値のいずれか、論理 OR で組み合わせた 2 つ以上の値に設定できます。値は次のいずれかです。

AP_UNCONGESTED

ISR セッションの数は、ノードの構成で *isr_sessions_upper_threshold* の値未満です。

AP_CONGESTED

ISR セッションの数は、しきい値を超えています。

AP_IRR_DEPLETED

ISR セッションの数は、そのノードに指定した最大値に達しました。

AP_ERR_DEPLETED

エンドポイント・セッションの数は、指定した最大値に達しました。

AP QUIESCING

タイプが AP_QUIESCE または AP_QUIESCE_ISR の STOP_NODE が発行されました。

nn_frsn

ネットワーク・ノードのみ: ネットワーク・ノードの現行フロー縮小シーケンス番号 (FRSN)。

nn_rsn ネットワーク・ノードのみ: リソース・シーケンス番号。

def_ls_good_xids

ノードの最後の起動以後にすべての定義済みリンク・ステーションで発生して成功した XID 交換の合計数。

def_ls_bad_xids

ノードの最後の起動以後にすべての定義済みリンク・ステーションで発生して失敗したノード XID 交換の合計数。

dyn_ls_good_xids

ノードの最後の起動以後にすべてのダイナミック・リンク・ステーションで発生して成功したノード XID 交換の合計数。

dyn_ls_bad_xids

ノードの最後の起動以後にすべてのダイナミック・リンク・ステーションで発生して失敗したノード XID 交換の合計数。

dlur_release_level

ノードがサポートする DLUR アーキテクチャーのリリース・レベル。これは値 1 (現在定義されている唯一の DLUR リリース・レベル) に設定されます。将来のバージョンには新しいリリース・レベルの DLUR アーキテクチャーが組み込まれて、別の値を戻すようになる可能性があります。

nns_dlus_served_lu_reg_supp

このパラメーターが適用されるのは、ローカル・ノードがエンド・ノードであるかまたは分岐ネットワーク・ノードである場合のみです。それ以外の場合は使用しません。

ネットワーク・ノード・サーバーが DLUS 実行済み LU 登録をサポートするかどうかを指定します。値は次のいずれかです。

AP_YES ネットワーク・ノード・サーバーは、DLUS 実行済み LU の登録をサポートします。

AP_NO ネットワーク・ノード・サーバーは、DLUS 実行済み LU の登録をサポートしません。

AP_UNKNOWN

ノードにはネットワーク・ノード・サーバーがありません。

nns_en_reg_diff_owning_cp

このパラメーターは、ローカル・ノードが分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

ネットワーク・ノード・サーバーが、オプション・セット 1123、別の所有 CP 名 NNS(BrNN) サポートをもつエンド・ノード・リソース登録をサポートするかどうかを指定します。

AP_YES ネットワーク・ノード・サーバーは、オプション・セット 1123 をサポートします。

AP_NO ネットワーク・ノード・サーバーは、オプション・セット 1123 をサポートしません。

AP_UNKNOWN

ノードにはネットワーク・ノード・サーバーがありません。

fq_nm_server_name

エンド・ノードのみ。ノードのためのネットワーク・ノード・サーバーの名前。

current_isr_sessions

このパラメーターが適用されるのは、ローカル・ノードがネットワーク・ノードであるかまたは分岐ネットワーク・ノードである場合のみです。それ以外の場合は使用しません。

このノードを介して経路指定される ISR セッションの数。

nm_functions_2

このパラメーターが適用されるのは、ローカル・ノードがネットワーク・ノードである場合のみです。それ以外の場合は使用しません。

ノードが分岐認識 (APPN オプション・セット 1120) をサポートするかどうかを指定します。

AP_NONE

ネットワーク・ノード・サーバーは、オプション・セット 1120 をサポートしません。

AP_BRANCH_AWARENESS

ノードはオプション・セット 1120 をサポートします。

branch_ntwk_arch_version

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

サポートされている分岐ネットワーク・アーキテクチャーのバージョンを指定します。これは 1 に設定されます。あるいは、ノードが分岐ネットワーク・アーキテクチャーをサポートしない場合には 0 (ゼロ) に設定されます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_NODE_ALL

QUERY_NODE_ALL は、Communications Server for Linux LAN 上のノードに関する情報を戻します。この verb は各ノードの名前および構成ファイル・ロールのみを戻し、ノードの構成に関する詳細情報は提供しません。アプリケーションは、特定のノード名に対して QUERY_NODE を使用して、そのノードに関する詳細情報を取得できます。

この verb は、ヌルのターゲット・ハンドルを指定して発行しなければなりません。

VCB 構造体

```
typedef struct query_node_all
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;              /* reserved                      */
}
```

QUERY_NODE_ALL

```
unsigned char  format;          /* reserved */
AP_UINT16     primary_rc;      /* primary return code */
AP_UINT32     secondary_rc;    /* secondary return code */
unsigned char  *buf_ptr;       /* pointer to buffer */
AP_UINT32     buf_size;        /* buffer size */
AP_UINT32     total_buf_size;  /* total buffer size required */
AP_UINT16     num_entries;     /* number of entries */
AP_UINT16     total_num_entries; /* total number of entries */
unsigned char  list_options;   /* listing options */
unsigned char  reserv3;        /* reserved */
unsigned char  node_name[128]; /* node name */
} QUERY_NODE_ALL;

typedef struct node_summary
{
    AP_UINT16     overlay_size; /* size of returned entry */
    unsigned char  node_name[128]; /* node name */
    unsigned char  config_role; /* server's config file role */
    unsigned char  reserv3[12]; /* reserved */
} NODE_SUMMARY;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_NODE_ALL

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

ノードの最大数 (そのノードのデータが戻されます)。ある範囲でなく、特定のノードに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

ノードのリスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

node_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

node_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストは、ノード名の順序には並べられません。ただし、この順序は、後続の QUERY_NODE_ALL verb でも同じになるため、アプリケーションは、通常の方法で複数の verb を使用して、複数のセクションにおける完全なリストを取得できます。アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

node_name

リストのインデックスとして使用されるノードの名前。このパラメーターは、*list_options* を `AP_FIRST_IN_LIST` に設定した場合には無視されます。

この名前は 1 ~ 128 文字の ASCII ストリングです。名前が 128 文字に満たない場合は、右側にスペースを入れます。

コンピューター名に `.` (ピリオド) 文字が含まれている場合、`Communications Server for Linux` はこれが完全修飾名であると想定します。ピリオドが含まれていない場合は、DNS ルックアップを実行してコンピューター名を判別します。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、`Communications Server for Linux` は、以下のパラメーターを戻します。

primary_rc

`AP_OK`

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

node_summary.overlay_size

戻された `node_summary` 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 `node_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは `Communications Server for Linux` の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

node_summary.node_name

`Communications Server for Linux` ノードの名前。

node_summary.config_role

ノードが実行されているサーバーの構成ファイル・ロール。構成ファイルの

QUERY_NODE_ALL

役割について詳しくは、「*Communications Server for Linux 管理ガイド*」を参照してください。値は次のいずれかです。

AP_ROLE_MASTER

サーバーは、マスター構成ファイルを保持します。

AP_ROLE_BACKUP

サーバーは、バックアップ構成ファイルを保持します。

AP_ROLE_NONE

サーバーは、構成ファイルのコピーを共有しません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、*Communications Server for Linux* は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_NODE_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、`AP_LIST_INCLUSIVE` に設定されていますが、*node_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての `NOF verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての `NOF verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_NODE_LIMITS

`QUERY_NODE_LIMITS` は、使用している *Communications Server for Linux* ライセンスにより特定のノードで使用できる機能とこれらの機能の使用量に関する情報を戻します。これらの機能は、以下のカテゴリに分割されます。

- 使用できる *Communications Server for Linux* 機能を指定するノード・オプション。
- 現在およびピーク時の *Communications Server for Linux* リソースの使用量を指定するノード・リソースの使用量。

この `verb` によって戻される情報を使用して、*Communications Server for Linux* リソースの使用量が、ユーザーのライセンスに許可された制限の範囲内であるかどうかを検査することができます。ライセンス交付要件については、「*Communications Server for Linux 入門*」を参照してください。

この verb によって戻される情報は、ときどき使用量ログ・ファイルにも書き込まれます。このファイルについて詳しくは、「*Communications Server for Linux 診断ガイド*」を参照してください。

VCB 構造体

```
typedef struct query_node_limits
{
    AP_UINT16      opcode;          /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code        */
    NODE_RESOURCE_LIMITS max_limits; /* reserved                      */
    NODE_RESOURCE_LIMITS curr_usage; /* current usage of LUs/sessions/users*/
    NODE_OPTIONS   node_options;   /* permitted functions          */
    unsigned char  reserv4[4];     /* reserved                      */
    NODE_RESOURCE_LIMITS max_usage; /* highest usage counts         */
} QUERY_NODE_LIMITS;

typedef struct node_resource_limits
{
    AP_INT32      lu62_tps;         /* APPC/CPI-C applications     */
    AP_INT32      lua_tps;         /* LUA applications             */
    AP_INT32      fmapi_tps;       /* reserved                     */
    AP_INT32      link_stations;   /* Active link stations         */
    AP_INT32      tn3270_connections; /* TN3270 server connections  */
    AP_INT32      tn_redirector_connections; /* TN redirector connections */
    AP_INT32      v4_sna_channels; /* reserved                     */
    AP_INT32      v4_g sna_channels; /* reserved                     */
    AP_INT32      data_sessions;  /* Active PLU-SLU sessions     */
    AP_INT32      reserv1[11];     /* Reserved                     */
} NODE_RESOURCE_LIMITS;

typedef struct node_options
{
    unsigned char network_node;    /* is Network Node supported?  */
    unsigned char end_node;       /* is End Node supported?      */
    unsigned char len_node;      /* is LEN Node supported?      */
    unsigned char dlur_support;   /* is DLUR supported?          */
    unsigned char pu_conc_support; /* is PU Conc supported?       */
    unsigned char tn_server_support; /* is TN Server supported?    */
    unsigned char hpr_support;    /* level of HPR support        */
    unsigned char back_level_client; /* are back-level clients supported?*/
    unsigned char reserv2;       /* reserved                     */
    unsigned char ssl_support;    /* is SSL supported?           */
    unsigned char branch_network_node; /* is BrNN supported?       */
    unsigned char reserv1[21];   /* reserved                     */
} NODE_OPTIONS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_NODE_LIMITS

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

QUERY_NODE_LIMITS

curr_usage.lu62_tps

このノードで現在活動状態である APPC アプリケーションおよび CPI-C アプリケーションの数。

curr_usage.lua_tps

このノードで現在活動状態である LUA アプリケーションの数。

curr_usage.link_stations

このノードで現在活動状態であるリンク・ステーションの数。

curr_usage.tn3270_connections

このノードで現在活動状態である TN3270 クライアントからの接続の数。

curr_usage.tn_redirector_connections

このノードで現在活動状態である TN リダイレクター・クライアントからの接続の数。

curr_usage.data_sessions

このノードで現在活動状態である PLU-SLU セッションの数。

全二重 APPC 会話が使用されている場合は、各全二重会話でセッションが 2 つ必要となります。

max_usage.lu62_tps

Linux コンピューターが再始動されてから、任意の時点でこのノードで活動状態であった APPC アプリケーションおよび CPI-C アプリケーションの最大数。

max_usage.lua_tps

Linux コンピューターが再始動されてから、任意の時点でこのノードで活動状態であった LUA アプリケーションの最大数。

max_usage.link_stations

Linux コンピューターが再始動されてから、任意の時点でこのノードで活動状態であったリンク・ステーションの最大数。

max_usage.tn3270_connections

Linux コンピューターが再始動されてから、任意の時点でこのノードで活動状態であった TN3270 クライアントからの接続の最大数。

max_usage.tn_redirector_connections

Linux コンピューターが再始動されてから、任意の時点でこのノードで活動状態であった TN リダイレクター・クライアントからの接続の最大数。

max_usage.data_sessions

Linux コンピューターが再始動されてから、任意の時点でこのノードで活動状態であった PLU-SLU セッションの最大数。

全二重 APPC 会話が使用されている場合は、各全二重会話でセッションが 2 つ必要となります。

node_options.network_node

保持しているライセンスでこのノードをネットワーク・ノードとして定義できるかどうかを指定します。値は次のいずれかです。

AP_YES ネットワーク・ノードとして定義できます。

AP_NO ネットワーク・ノードとして定義できません。

node_options.end_node

保持しているライセンスでこのノードをエンド・ノードとして定義できるかどうかを指定します。値は次のいずれかです。

AP_YES エンド・ノードとして定義できます。

AP_NO エンド・ノードとして定義できません。

node_options.len_node

保持しているライセンスでこのノードを LEN ノードとして定義できるかどうかを指定します。値は次のいずれかです。

AP_YES LEN ノードとして定義できます。

AP_NO LEN ノードとして定義できません。

node_options.dlur_support

このパラメーターは予約済みです。

保持しているライセンスでこのノードで従属 LU リクエスター (Dependent LU Requester: DLUR) を使用できるかどうかを指定します。値は次のいずれかです。

AP_YES DLUR は使用できます。

AP_NO DLUR は使用できません。

node_options.pu_conc_support

保持しているライセンスでこのノードで SNA ゲートウェイを使用できるかどうかを指定します。値は次のいずれかです。

AP_YES SNA ゲートウェイは使用できます。

AP_NO SNA ゲートウェイは使用できません。

node_options.tn_server_support

保持しているライセンスでこのノードで TN サーバーを使用できるかどうかを指定します。値は次のいずれかです。

AP_YES TN サーバーは使用できます。

AP_NO TN サーバーは使用できません。

node_options.hpr_support

保持しているライセンスを使用してこのノードで HPR (ハイパフォーマンス経路指定) を使用できるかどうかを指定します。値は次のいずれかです。

AP_YES HPR はサポートされます。

AP_NO HPR はサポートされません。

node_options.back_level_client

このパラメーターは予約済みです。

node_options.ssl_support

セキュア・ソケット層ソフトウェアがノードにインストールされているかどうかを指定します (TN サーバーでの使用)。値は次のいずれかです。

AP_YES SSL ソフトウェアはインストールされています。

AP_NO SSL ソフトウェアはインストールされていません。

QUERY_NODE_LIMITS

node_options.branch_network_node

保持しているライセンスでこのノードを分岐ネットワーク・ノードとして定義できるかどうかを指定します。 値は次のいずれかです。

AP_YES 分岐ネットワーク・ノードはサポートされています。

AP_NO 分岐ネットワーク・ノードはサポートされていません。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_PARTNER_LU

QUERY_PARTNER_LU は、ローカル LU が現在使用しているパートナー LU または使用したパートナー LU に関する情報を戻します。この verb が戻すのはパートナー LU の使用状況に関する情報であって、その定義ではありません。パートナー LU の定義は、QUERY_PARTNER_LU_DEFINITION を使用して入手してください。

この verb は、使用するオプションに応じて、特定の LU または複数の LU に関する要約情報または詳細情報を入手するために使用できます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_partner_lu
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;               /* reserved                      */
    unsigned char  format;                /* reserved                      */
    AP_UINT16      primary_rc;            /* primary return code          */
    AP_UINT32      secondary_rc;          /* secondary return code        */
    unsigned char  *buf_ptr;              /* pointer to buffer            */
    AP_UINT32      buf_size;              /* buffer size                   */
    AP_UINT32      total_buf_size;        /* total buffer size required   */
    AP_UINT16      num_entries;           /* number of entries            */
    AP_UINT16      total_num_entries;     /* total number of entries      */
    unsigned char  list_options;          /* listing options              */
    unsigned char  reserv3;               /* reserved                      */
    unsigned char  lu_name[8];            /* LU name                       */
    unsigned char  lu_alias[8];           /* LU alias                      */
    unsigned char  plu_alias[8];          /* partner LU alias             */
    unsigned char  fqplu_name[17];        /* fully qualified partner LU name */
    unsigned char  active_sessions;       /* active sessions only filter  */
} QUERY_PARTNER_LU;

typedef struct plu_summary
{
    AP_UINT16      overlay_size;          /* size of returned entry       */
    unsigned char  plu_alias[8];          /* partner LU alias             */
    unsigned char  fqplu_name[17];        /* fully qualified partner LU name */
    unsigned char  reserv1;               /* reserved                      */
    unsigned char  description[32];        /* resource description          */
    unsigned char  reserv2[16];           /* reserved                      */
    AP_UINT16      act_sess_count;        /* currently active sessions count */
    unsigned char  partner_cp_name[17];   /* partner LU CP name           */
    unsigned char  partner_lu_located;    /* CP name resolved?           */
} PLU_SUMMARY;
```

```

typedef struct plu_detail
{
    AP_UINT16    overlay_size;           /* size of returned entry      */
    unsigned char plu_alias[8];         /* partner LU alias            */
    unsigned char fqplu_name[17];      /* fully qualified partner LU name */
    unsigned char reserv1;             /* reserved                    */
    unsigned char description[32];     /* resource description        */
    unsigned char reserv2[16];        /* reserved                    */
    AP_UINT16    act_sess_count;       /* currently active sessions count */
    unsigned char partner_cp_name[17]; /* partner LU CP name          */
    unsigned char partner_lu_located;  /* CP name resolved?          */
    unsigned char plu_un_name[8];     /* partner LU uninterpreted name */
    unsigned char parallel_sess_supp;  /* parallel sessions supported? */
    unsigned char conv_security;       /* conversation security       */
    AP_UINT16    max_mc_ll_send_size;  /* maximum send LL size for mapped */
                                           /* conversations                */
    unsigned char implicit;           /* implicit or explicit entry  */
    unsigned char security_details;   /* session security details    */
    unsigned char duplex_support;     /* full-duplex support         */
    unsigned char preference;         /* reserved                    */
    unsigned char reserva[16];        /* reserved                    */
} PLU_DETAIL;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_PARTNER_LU

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

LU の最大数 (その LU のデータが戻されます)。ある範囲でなく、特定の LU に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

指定したローカル LU に関連したパートナー LU のリストの最初のエントリーから開始します。

AP_LIST_INCLUSIVE

ローカル LU 名とパートナー LU 名を組み合わせで指定したエン
トリーから開始します。

AP_LIST_FROM_NEXT

ローカル LU 名とパートナー LU 名を組み合わせで指定したエン
トリーの直後のエントリーから開始します。

AP_LIST_BY_ALIAS

リストは LU 名の順ではなく LU 別名の順に戻されます。このオ
プションは、AP_FIRST_IN_LIST と共に指定した場合のみ有効です。
(AP_LIST_FROM_NEXT または AP_LIST_INCLUSIVE の場合は、リスト
の索引としてどちらを指定したかに応じて、リストは LU 名順また
は LU 別名順になります。)

list_options パラメーターを AP_LIST_INCLUSIVE または AP_LIST_FROM_NEXT
に設定した場合、指定したローカル LU (*lu_name* または *lu_alias*) とパート
ナー LU (*plu_alias* または *fqplu_name*) の組み合わせは、パートナー LU
のリストの索引として使用されます。

リストは *fqplu_name* で並べられます。リストの順序と、アプリケーション
でそのリストから特定のエントリーを入手する方法について詳しくは、47
ページの『QUERY_* verb のリスト・オプション』を参照してください。

lu_name

ローカル LU の LU 名。この名前は、8 バイトからなるタイプ A の
EBCDIC スtringで、名前が 8 文字に満たない場合は、右側にスペース
を入れます。LU が LU 名でなく LU 別名で識別されていることを示すに
は、このパラメーターを 8 桁の 2 進ゼロに設定し、次に示すパラメーター
に LU 別名を指定します。

lu_alias

ローカル LU の LU 別名。このパラメーターは、*lu_name* フィールドを 8
桁の 2 進ゼロに設定した場合のみ使用され、それ以外の場合は無視されま
す。この名前は 8 バイトからなる ASCII スtringで、名前が 8 文字に
満たない場合は、右側にスペースを入れます。ローカル CP に関連した LU
(デフォルト LU) を示すには、*lu_name* と *lu_alias* の両方のビットをゼロに
設定します。

plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII スtringで、名
前が 8 バイトに満たない場合は、右側にスペースを入れます。このパラメ
ーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されま
す。それ以外の場合は、パートナー LU 名に対して LU 別名または LU の
完全修飾名を指定する必要があります。LU を別名ではなく完全修飾名に
より識別することを示すには、このパラメーターを 8 桁の 2 進ゼロに設定
し、次のパラメーターで LU 名を指定します。

fqplu_name

17 バイトからなるパートナー LU の完全修飾ネットワーク名。このパラメ
ーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されま
す。それ以外の場合は、パートナー LU 名に対して LU 別名または LU の

完全修飾名を指定する必要があります。このパラメーターは、*plu_alias* フィールドを 8 桁の 2 進ゼロに設定した場合のみ使用され、それ以外の場合は無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC スtringです。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

active_sessions

セッションが活動状態であるパートナー LU のみにに関する情報を戻すか、すべてのパートナー LU に関する情報を戻すかを指定します。値は次のいずれかです。

AP_YES セッションが現在活動状態であるパートナー LU のみにに関する情報を戻します。

AP_NO セッションが活動状態であるか活動状態であったすべてのパートナー LU に関する情報を戻します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを戻す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

plu_summary.overlay_size

戻された *plu_summary* 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *plu_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があります。

るためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

plu_summary.plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

plu_summary.fqplu_name

17 バイトからなるパートナー LU の完全修飾ネットワーク名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

plu_summary.description

パートナー LU を記述したヌルで終了するテキスト・ストリング (そのパートナー LU の定義で指定したもの)。

plu_summary.act_sess_count

ローカル LU とパートナー LU 間の活動状態であるセッションの合計数。

plu_summary.partner_cp_name

パートナー LU に関連した CP の 17 バイトからなる完全修飾ネットワーク名。次の *partner_lu_located* を AP_NO に設定した場合は、このパラメーターは使用されません。

この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

plu_summary.partner_lu_located

ローカル・ノードがパートナー LU を見つけた場所で CP を見つけたかどうかを指定します。値は次のいずれかです。

AP_YES パートナー LU が見つかりました。 *partner_cp_name* パラメーターにはパートナー LU の CP 名が入っています。

AP_NO パートナー LU はまだ見つかっていません。 *partner_cp_name* パラメーターは検査しないでください。

plu_detail.overlay_size

戻された *plu_detail* 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *plu_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

plu_detail.plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

plu_detail.fqplu_name

17 バイトからなるパートナー LU の完全修飾ネットワーク名。この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、1 ～ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ～ 8 文字の A String文字からなるネットワーク名で構成されます。

plu_detail.description

パートナー LU を記述したヌルで終了するテキスト・String (そのパートナー LU の定義で指定したもの)。

plu_detail.act_sess_count

ローカル LU とパートナー LU 間の活動状態であるセッションの合計数。

plu_detail.partner_cp_name

パートナー LU に関連した CP の 17 バイトからなる完全修飾ネットワーク名。次の *partner_lu_located* を AP_NO に設定した場合は、このパラメーターは使用されません。

この名前は 17 バイトの EBCDIC Stringで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A String文字からなるネットワーク名で構成されます。

plu_detail.partner_lu_located

ローカル・ノードがパートナー LU を見つけた場所で CP を見つけたかどうかを指定します。値は次のいずれかです。

AP_YES パートナー LU が見つかりました。 *partner_cp_name* パラメーターにはパートナー LU の CP 名が入っています。

AP_NO パートナー LU はまだ見つかっていません。 *partner_cp_name* パラメーターは検査しないでください。

plu_detail.plu_un_name

パートナー LU の非解釈名。この名前は 8 バイトからなるタイプ A の EBCDIC 文字Stringです。

plu_detail.parallel_sess_supp

並列セッションがサポートされるかどうかを指定します。値は次のいずれかです。

AP_YES 並列セッションがサポートされます。

AP_NO 並列セッションはサポートされません。

plu_detail.conv_security

このパートナー LU に会話セキュリティー情報を送信できるかどうかを指定します。値は次のいずれかです。

AP_YES ローカル TP が提供する会話セキュリティー情報がパートナー LU に送信されます。

AP_NO ローカル TP が提供する会話セキュリティー情報はパートナー LU に送信されません。

AP_UNKNOWN

パートナー LU とともに活動化されるセッションはありません。

plu_detail.max_mc_ll_send_size

パートナー LU に送信できる論理レコードの最大サイズ (バイト単位)。この値の範囲は、1 ~ 32,767 か制限がないことを示すゼロです (その場合の最大値は 32,767 です)。これより大きいデータ・レコードは、パートナー LU に送信する前にいくつかの LL レコードに分割されます。

plu_detail.implicit

エントリーが暗黙的な定義で作成されたか明示的な定義で作成されたかを指定します。値は次のいずれかです。

AP_YES エントリーは暗黙的なエントリーです。

AP_NO エントリーは明示的なエントリーです。

plu_detail.security_details

BIND で折衝された会話セキュリティー・サポートを戻します。値は次のいずれかです。

AP_CONVERSATION_LEVEL_SECURITY

会話セキュリティー情報は、パートナー LU に対する会話割り振り要求またはパートナー LU からの会話割り振り要求に基づいて受け入れられます。特定のタイプの会話セキュリティー・サポートは、次の値により記述されます。

AP_ALREADY_VERIFIED

ローカル LU とパートナー LU の両方が、会話割り振りの検査済み要求を受け入れます。検査済み要求にはユーザー ID のみが必要です。パスワードは必要ありません。

AP_PERSISTENT_VERIFICATION

ローカル LU とパートナー LU 間のセッションで、持続検査がサポートされます。会話に対する初期要求 (ユーザー ID と、通常はパスワードも必要) が検査されると、それ以後の会話要求に必要なのはユーザー ID のみになります。

AP_PASSWORD_SUBSTITUTION

ローカル LU とパートナー LU がパスワード置換会話セキュリティーをサポートします。会話割り振り要求を発行するとき、暗号化されたパスワードが必要です。パスワード置換がサポートされていない場合は、パスワードを平文 (暗号化されていない) フォーマットにする必要があります。セッションがパスワード置換をサポートしていない場合は、セキュリティー・タイプを AP_PGM_STRONG に設定した Allocate または Send_Conversation は失敗します。

AP_UNKNOWN

パートナー LU とともに活動化されるセッションはありません。

plu_detail.duplex_support

BIND で折衝された会話二重サポートを戻します。値は次のいずれかです。

AP_HALF_DUPLEX

半二重会話のみがサポートされます。

AP_FULL_DUPLEX

全二重セッションと半二重セッションの両方がサポートされます。優先データもサポートされます。

AP_UNKNOWN

パートナー LU との活動状態であるセッションがないため、会話二重サポートは不明です。

preference

このパラメーターは予約済みです。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、`Communications Server for Linux` は、以下のパラメーターを戻します。

primary_rc

`AP_PARAMETER_CHECK`

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_ALIAS

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、`AP_LIST_INCLUSIVE` に設定されていますが、`lu_alias` パラメーターは無効でした。

AP_INVALID_LU_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、`AP_LIST_INCLUSIVE` に設定されていますが、`lu_name` パラメーターは無効でした。

AP_INVALID_PLU_NAME

list_options パラメーターは、指定した名前から始まるすべてのエントリーを表示する `AP_LIST_INCLUSIVE` に設定しましたが、次の条件のいずれかが適用されます。

- `fqplu_name` パラメーターは、このローカル LU のどのパートナーの名前とも一致しません。
- 指定したローカル LU とパートナー LU の組み合わせに対して (ノードを最後に起動してから) 活動状態であったセッションはありません。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての `NOF verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての `NOF verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_PARTNER_LU_DEFINITION

QUERY_PARTNER_LU_DEFINITION は、ローカル LU に対するパートナー LU に関する情報を戻します。この verb が戻すのは LU の定義に関する情報であって、LU の現在の使用状況ではありません。使用状況の情報を入手するには、QUERY_PARTNER_LU を使用してください。

この verb は、使用するオプションに応じて、特定の LU または複数の LU に関する要約情報または詳細情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_partner_lu_definition
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;               /* reserved                     */
    unsigned char  format;               /* reserved                     */
    AP_UINT16      primary_rc;           /* primary return code          */
    AP_UINT32      secondary_rc;        /* secondary return code        */
    unsigned char  *buf_ptr;             /* pointer to buffer            */
    AP_UINT32      buf_size;             /* buffer size                  */
    AP_UINT32      total_buf_size;      /* total buffer size required   */
    AP_UINT16      num_entries;         /* number of entries            */
    AP_UINT16      total_num_entries;   /* total number of entries      */
    unsigned char  list_options;        /* listing options              */
    unsigned char  reserv3;             /* reserved                     */
    unsigned char  plu_alias[8];        /* partner LU alias             */
    unsigned char  fqplu_name[17];     /* fully qualified partner LU name */
} QUERY_PARTNER_LU_DEFINITION;

typedef struct partner_lu_def_summary
{
    AP_UINT16      overlay_size;        /* size of returned entry       */
    unsigned char  plu_alias[8];        /* partner LU alias             */
    unsigned char  fqplu_name[17];     /* fully qualified partner LU name */
    unsigned char  description[32];    /* resource description         */
    unsigned char  reserv1[16];        /* reserved                     */
} PARTNER_LU_DEF_SUMMARY;

typedef struct partner_lu_def_detail
{
    AP_UINT16      overlay_size;        /* size of returned entry       */
    unsigned char  plu_alias[8];        /* partner LU alias             */
    unsigned char  fqplu_name[17];     /* fully qualified partner LU name */
    unsigned char  reserv1;            /* reserved                     */
    PLU_CHARS      plu_chars;          /* partner LU characteristics   */
} PARTNER_LU_DEF_DETAIL;

typedef struct plu_chars
{
    unsigned char  fqplu_name[17];     /* fully qualified partner LU name */
    unsigned char  plu_alias[8];        /* partner LU alias             */
    unsigned char  description[32];    /* resource description         */
    unsigned char  reserv2[16];        /* reserved                     */
    unsigned char  plu_un_name[8];     /* partner LU uninterpreted name */
    unsigned char  preference;         /* reserved                     */
    AP_UINT16      max_mc_ll_send_size; /* maximum MC send LL size     */
    unsigned char  conv_security_ver;  /* already-verified security   */
    unsigned char  supported;          /* supported?                   */
    unsigned char  parallel_sess_supp; /* parallel sessions supported? */
    unsigned char  reserv3[8];         /* reserved                     */
} PLU_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_PARTNER_LU_DEFINITION

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

LU の最大数 (その LU のデータが戻されます)。ある範囲でなく、特定の LU に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

plu_alias パラメーターまたは *fqplu_name* パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

plu_alias パラメーターまたは *fqplu_name* パラメーターで指定したエントリーの直後のエントリーから開始します。

AP_FIRST_IN_LIST を指定した場合は、論理 OR 演算を使用して、次のオプションも組み込むことができます。

AP_LIST_BY_ALIAS

リストは LU 名の順ではなく LU 別名の順に戻されます。このオプションは、AP_FIRST_IN_LIST と共に指定した場合のみ有効です。(AP_LIST_FROM_NEXT または AP_LIST_INCLUSIVE の場合は、リストの索引としてどちらを指定したかに応じて、リストは LU 名順または LU 別名順になります。)

アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

plu_alias

パートナー LU 別名。この名前は 8 バイトからなる ASCII スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。それ以外の場合は、パートナー LU 名に対して LU 別名または LU の完全修飾名を指定する必要があります。パートナー LU を別名ではなく完全修飾名で定義することを示すには、このパラメーターを 8 桁の 2 進ゼロに設定し、次のパラメーターで LU 名を指定します。

fqplu_name

情報を求めるパートナー LU の完全修飾名、または LU のリストの索引として使用する名前。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。それ以外の場合は、パートナー LU 名に対して LU 別名または LU の完全修飾名を指定する必要があります。このパラメーターは、*plu_alias* パラメーターをゼロに設定した場合のみ使用され、それ以外の場合は無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC String です。この名前は、最大 8 文字の A String 文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A String 文字からなるネットワーク名で構成されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

partner_lu_def_summary.overlay_size

戻された *partner_lu_def_summary* 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *partner_lu_def_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。

せん。また、C の sizeof() 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

partner_lu_def_summary.plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

partner_lu_def_summary.fqplu_name

パートナー LU の完全修飾ネットワーク名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

partner_lu_def_summary.description

パートナー LU を記述したヌルで終了するテキスト・ストリング (そのパートナー LU の定義で指定したもの)。

partner_lu_def_detail.overlay_size

戻された *partner_lu_def_detail* 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *partner_lu_def_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の sizeof() 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

partner_lu_def_detail.plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

partner_lu_def_detail.fqplu_name

パートナー LU の完全修飾ネットワーク名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

partner_lu_def_detail.plu_chars.fqplu_name

パートナー LU の完全修飾ネットワーク名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

QUERY_PARTNER_LU_DEFINITION

partner_lu_def_detail.plu_chars.plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

partner_lu_def_detail.plu_chars.description

パートナー LU を記述したヌルで終了するテキスト・ストリング (そのパートナー LU の定義で指定したもの)。

partner_lu_def_detail.plu_chars.plu_un_name

パートナー LU の非解釈名。この名前は、8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

partner_lu_def_detail.plu_chars.preference

このパラメーターは予約済みです。

partner_lu_def_detail.plu_chars.max_mc_ll_send_size

パートナー LU に送信できる論理レコードの最大長 (バイト単位)。この値の範囲は、1 ~ 32,767 か制限がないことを示すゼロです (その場合の最大値は 32,767 です)。これより大きいデータ・レコードは、パートナー LU に送信する前にいくつかの LL レコードに分割されます。

partner_lu_def_detail.plu_chars.conv_security_ver

パートナー LU がローカル LU の代わりにユーザー ID の検査を行う許可を持っているかどうか、つまりパートナー LU が接続要求に検査済みのインディケータを設定できるかどうかを指定します。値は次のいずれかです。

AP_YES パートナー LU はユーザー ID の検査を許可されています。

AP_NO パートナー LU はユーザー ID の検査を許可されています。

partner_lu_def_detail.plu_chars.parallel_sess_supp

並列セッションがサポートされるかどうかを指定します。値は次のいずれかです。

AP_YES 並列セッションがサポートされます。

AP_NO 並列セッションはサポートされません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PLU_NAME

list_options パラメーターを、指定した名前から始まるすべてのエントリーを表示する AP_LIST_INCLUSIVE に設定しましたが、*plu_alias* パラメーターまたは *fqplu_name* パラメーターが有効ではありませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_PORT

QUERY_PORT は、ノードのポートについての情報のリストを戻します。この情報は、「決定済みデータ」(実行中に動的に収集されたデータ) および「定義済みデータ」(アプリケーションにより DEFINE_PORT で提供されたデータ) として構造化されます。

この verb は、使用するオプションに応じて、特定のポートまたは複数のポートに関する要約情報または詳細情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_port
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;               /* reserved                     */
    unsigned char  format;                /* reserved                     */
    AP_UINT16      primary_rc;            /* primary return code          */
    AP_UINT32      secondary_rc;          /* secondary return code        */
    unsigned char  *buf_ptr;              /* pointer to buffer            */
    AP_UINT32      buf_size;              /* buffer size                  */
    AP_UINT32      total_buf_size;        /* total buffer size required   */
    AP_UINT16      num_entries;           /* number of entries            */
    AP_UINT16      total_num_entries;     /* total number of entries      */
    unsigned char  list_options;          /* listing options              */
    unsigned char  reserv3;               /* reserved                     */
    unsigned char  port_name[8];          /* port name                    */
    unsigned char  dlc_name[8];           /* DLC name filter              */
} QUERY_PORT;

typedef struct port_summary
{
    AP_UINT16      overlay_size;           /* size of returned entry       */
    unsigned char  port_name[8];           /* port name                    */
    unsigned char  description[32];        /* resource description          */
    unsigned char  reserv2[16];           /* reserved                     */
    unsigned char  port_state;             /* port state                   */
    unsigned char  reserv1[1];            /* reserved                     */
    unsigned char  dlc_name[8];           /* name of DLC                  */
} PORT_SUMMARY;

typedef struct port_detail
{
    AP_UINT16      overlay_size;           /* size of returned entry       */
    unsigned char  port_name[8];           /* port name                    */
    unsigned char  reserv1[2];            /* reserved                     */
    PORT_DET_DATA  det_data;               /* determined data              */
    PORT_DEF_DATA  def_data;               /* defined data                  */
} PORT_DETAIL;
```

QUERY_PORT

```
typedef struct port_det_data
{
    unsigned char    port_state;           /* port state */
    unsigned char    dlc_type;             /* DLC type */
    unsigned char    port_sim_rim;         /* port initialization options */
    unsigned char    reserv1;              /* reserved */
    AP_UINT16        def_ls_good_xids;     /* number of successful XIDs */
    AP_UINT16        def_ls_bad_xids;      /* number of unsuccessful XIDs */
    AP_UINT16        dyn_ls_good_xids;     /* successful XIDs on dynamic
    /* LS count */
    AP_UINT16        dyn_ls_bad_xids;      /* failed XIDs on dynamic LS
    /* count */
    AP_UINT16        num_implicit_links;   /* number of implicit links */
    unsigned char    neg_ls_supp;          /* negotiable? */
    unsigned char    abm_ls_supp;          /* ABM support? */
    AP_UINT32        start_time;           /* Start time of the port */
    unsigned char    reserva[12];         /* reserved */
} PORT_DET_DATA;

typedef struct port_def_data
{
    unsigned char    description[32];      /* resource description */
    unsigned char    initially_active;     /* is the port initially active? */
    unsigned char    reserv2[15];          /* reserved */
    unsigned char    dlc_name[8];          /* DLC name associated with port */
    unsigned char    port_type;            /* port type */
    unsigned char    port_attributes[4];   /* port attributes */
    unsigned char    implicit_uplink_to_en; /* implicit EN links up or down? */
    unsigned char    implicit_appn_links_len; /* reserved */
    unsigned char    reserv3;              /* reserved */
    AP_UINT32        port_number;           /* port number */
    AP_UINT16        max_rcv_btu_size;     /* max receive BTU size */
    AP_UINT16        tot_link_act_lim;     /* total link activation limit */
    AP_UINT16        inb_link_act_lim;     /* inbound link activation limit */
    AP_UINT16        out_link_act_lim;     /* outbound link activation limit */
    unsigned char    ls_role;              /* initial link station role */
    unsigned char    retry_flags;           /* reserved */
    AP_UINT16        max_activation_attempts; /* reserved */
    AP_UINT16        activation_delay_timer; /* reserved */
    unsigned char    mltg_pacing_algorithm; /* reserved */
    unsigned char    implicit_tg_sharing_prohibited; /* reserved */
    unsigned char    link_spec_data_format; /* reserved */
    unsigned char    limit_enable;          /* reserved */
    unsigned char    reserv1[6];           /* reserved */
    unsigned char    implicit_dspu_template[8]; /* implicit dspu template */
    AP_UINT16        implicit_ls_limit;     /* implicit ls limit */
    unsigned char    reserv4;              /* reserved */
    unsigned char    implicit_dspu_services; /* reserved */
    unsigned char    implicit_deact_timer;  /* deact timer for implicit LSs */
    AP_UINT16        act_xid_exchange_limit; /* activation XID exchange limit */
    AP_UINT16        nonact_xid_exchange_limit; /* non-act. XID exchange
    /* limit */
    unsigned char    ls_xmit_rcv_cap;       /* LS transmit-receive capability */
    unsigned char    max_ifrm_rcvd;         /* maximum number of I-frames
    /* that can be received */
    AP_UINT16        target_pacing_count;   /* target pacing count */
    AP_UINT16        max_send_btu_size;     /* maximum send BTU size */
    LINK_ADDRESS     dlc_data;              /* DLC data */
    LINK_ADDRESS     hpr_dlc_data;          /* reserved */
    unsigned char    implicit_cp_cp_sess_support; /* implicit links allow
    /* CP-CP sessions */
    unsigned char    implicit_limited_resource; /* implicit links are
    /* limited resource */
    unsigned char    implicit_hpr_support;  /* Implicit links support HPR */
    unsigned char    implicit_link_lvl_error; /* Send HPR traffic on implicit
    /* links using link-level error
    /* recovery? */
    unsigned char    retired1;              /* reserved */
}
```

```

TG_DEFINED_CHARS default_tg_chars;      /* default TG chars      */
unsigned char   discovery_supported;    /* reserved               */
AP_UINT16      port_spec_data_len;     /* length of port specific data */
AP_UINT16      link_spec_data_len;     /* length of link specific data */
} PORT_DEF_DATA;

```

`link_address` 構造体に関する詳細については、478 ページの『QUERY_LS』を参照してください。ポート固有のデータおよびリンク固有のデータに関する詳細については、206 ページの『DEFINE_PORT』および 135 ページの『DEFINE_LS』を参照してください。ポート固有のデータのデータ構造体は `port_def_data` 構造体に続き、リンク固有のデータのデータ構造体がこれに続きますが、どちらの構造体もスペースが入るため 4 バイト目の境界から始まります。

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

`opcode` AP_QUERY_PORT

`buf_ptr` Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

`buf_size`

指定したデータ・バッファのサイズ。

`num_entries`

ポートの最大数 (そのポートのデータが戻されます)。ある範囲でなく、特定のポートに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリを戻します。

`list_options`

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリから開始します。

AP_LIST_INCLUSIVE

`port_name` パラメーターで指定したエントリから開始します。

AP_LIST_FROM_NEXT

`port_name` パラメーターで指定したエントリの直後のエントリから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

port_name

照会するポートの名前。これは 8 バイトからなる ASCII スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されません。

dlc_name

DLC 名フィルター。特定の DLC に関連したポートのみに関する情報を戻すには、DLC 名を指定します。これは 8 バイトからなる ASCII スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。DLC 名で選別しないですべてのポートに関する情報を戻すには、このパラメーターの 8 つのビットをゼロに設定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを戻す戻り値。buf_size より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。num_entries より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

port_summary.overlay_size

戻された port_summary 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 port_summary 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の sizeof() 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

port_summary.port_name

ポートの名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

port_summary.description

ポートを記述したヌルで終了するテキスト・ストリング (そのポートの定義で指定したもの)。

port_summary.port_state

ポートの現行状態を指定します。値は次のいずれかです。

AP_ACTIVE

ポートは活動状態です。

AP_NOT_ACTIVE

ポートは活動状態ではありません。

AP_PENDING_ACTIVE

START_PORT が進行中です。

AP_PENDING_INACTIVE

STOP_PORT が進行中です。

port_summary.dlc_name

このポートに関連した DLC の名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

port_detail.overlay_size

戻された `port_detail` 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 `port_detail` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

port_detail.port_name

ポートの名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

port_detail.det_data.port_state

ポートの現行状態を指定します。値は次のいずれかです。

AP_ACTIVE

ポートは活動状態です。

AP_NOT_ACTIVE

ポートは活動状態ではありません。

AP_PENDING_ACTIVE

START_PORT が進行中です。

AP_PENDING_INACTIVE

STOP_PORT が進行中です。

port_detail.det_data.dlc_type

ポートの DLC タイプ。次のいずれかです。

AP_SDLC

SDLC

AP_X25 QLLC

AP_TR トークンリング

AP_ETHERNET

イーサネット

AP_MPC マルチパス・チャネル (MPC)、System z 上の Communications Server for Linux のみ

AP_IP Enterprise Extender (HPR/IP)

port_detail.det_data.port_sim_rim

設定初期化モード (SIM) と受信初期化モード (RIM) がサポートされているかどうかを指定します。値は次のいずれかです。

AP_YES SIM と RIM がサポートされます。

AP_NO SIM と RIM はサポートされません。

port_detail.det_data.def_ls_good_xids

ポートを最後に起動してから、このポートのすべての定義されたリンク・ステーションで発生して成功した XID 交換の合計数。

port_detail.det_data.def_ls_bad_xids

ポートを最後に起動してから、このポートのすべての定義されたリンク・ステーションで発生して失敗した XID 交換の合計数。

port_detail.det_data.dyn_ls_good_xids

ポートを最後に起動してから、このポートのすべてのダイナミック・リンク・ステーションで発生して成功した XID 交換の合計数。

port_detail.det_data.dyn_ls_bad_xids

ポートを最後に起動してから、このポートのすべてのダイナミック・リンク・ステーションで発生して失敗した XID 交換の合計数。

port_detail.det_data.num_implicit_links

このポートで現在活動状態である暗黙リンクの合計数。これにはディスカバリーの使用に続いて作成されたダイナミック・リンクおよび暗黙リンクが含まれます。このポートで許可されるこの種のリンクの数は、*port_def_data* の *implicit_ls_limit* パラメーターにより制限されます。

port_detail.det_data.neg_ls_supp

折衝可能リンク・ステーションのサポート。値は次のいずれかです。

AP_YES リンク・ステーションは折衝可能です。

AP_NO リンク・ステーションは折衝できません。

port_detail.det_data.abm_ls_supp

ABM リンク・ステーションのサポート。値は次のいずれかです。

AP_YES ABM リンク・ステーションはサポートされます。

AP_NO ABM リンク・ステーションはサポートされません。

AP_UNKNOWN

このポートに関連した DLC がまだ始動されていないため、ABM リンク・ステーションのサポートを決定できません。

port_detail.det_data.start_time

ノードが始動された時刻とこのポートが始動された最後の時刻の間の経過時間 (1/100 秒)。このポートがまだ始動されていない場合には、このパラメーターはゼロに設定されます。

port_detail.def_data.description

ポートを記述したヌルで終了するテキスト・ストリング (そのポートの定義で指定したもの)。

port_detail.def_data.dlc_name

このポートに関連した DLC の名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

port_detail.def_data.port_type

ポートが使用する回線のタイプ。

SDLC の場合、以下の値が戻されます。

AP_PORT_SWITCHED

交換回線。

AP_PORT_NONSWITCHED

非交換回線。

QLLC の場合は、AP_PORT_SWITCHED に設定されます。

トークンリングまたはイーサネットの場合は、AP_PORT_SATF (共用アクセス・トランスポート機能) に設定されます。

Enterprise Extender (HPR/IP) の場合は、AP_PORT_SATF (共用アクセス・トランスポート機能) に設定されます。

port_detail.def_data.port_attributes

これはビット・フィールドです。値は AP_NO か、次の値です。

AP_RESOLVE_BY_LINK_ADDRESS

この値は、受信した XID3 上の CP 名 (またはノード ID) を使用する前に CONNECT_IN のリンク・アドレスを使用して着呼の解決を試みることを指定します。この値は、*port_type* パラメーターを AP_PORT_SWITCHED に設定しない場合には無視されます。

def_data.implicit_uplink_to_en

このパラメーターが適用されるのは、ローカル・ノードが分岐ネットワーク・ノードである場合のみです。ローカル・ノードが他のタイプの場合は予約済みです。

隣接ノードがエンド・ノードの場合、このパラメーターは、このポートから切り離された暗黙リンク・ステーションがアップリンクであるのかダウンリンクであるのかを指定します。同じ隣接ノードに既にリンクがある場合、このパラメーターは無視されます。これは、この場合、既存のリンクがリンク・タイプの判別に使用されるためです。値は次のいずれかです。

AP_YES エンド・ノードへの暗黙リンクはアップリンクです。

AP_NO エンド・ノードへの暗黙リンクはダウンリンクです。

port_detail.def_data.port_number

ポート番号。

port_detail.def_data.max_rcv_btu_size

受信できる BTU の最大サイズ。

port_detail.def_data.tot_link_act_lim

合計リンク活動化限度。

port_detail.def_data.inb_link_act_lim

インバウンド・リンク活動化限度。

port_detail.def_data.out_link_act_lim

アウトバウンド・リンク活動化限度。

port_detail.def_data.ls_role

リンク・ステーションの役割。

SDLC または QLLC の場合、以下の値が戻されます。

AP_LS_PRI

1 次

AP_LS_SEC

2 次

AP_LS_NEG

折衝可能

トークンリングまたはイーサネットの場合は、AP_LS_NEG (折衝可能) に設定されます。

port_detail.def_data.implicit_dspu_template

DEFINE_DSPU_TEMPLATE verb で定義された DSPU テンプレートを指定します。このテンプレートは、このポートで活動化される暗黙リンクに SNA ゲートウェイをローカル・ノードで提供する場合に、定義に使用されます。指定したテンプレートがリンク活動化時にない (または既にインスタンス限度に達している) 場合は、活動化が失敗します。これは 8 バイトからなるローカル・システム上で表示可能な文字セットのストリングです。8 バイトすべてが有効なバイトであり、必ず設定しなければなりません。

def_data.implicit_dspu_services パラメーターを AP_PU_CONCENTRATION に設定した場合は、このパラメーターは予約済みです。

port_detail.def_data.implicit_ls_limit

このポートで同時に活動状態にできる暗黙リンク・ステーションの最大数で、ダイナミック・リンクとディスクバリーのために活動化されるリンクを含みます。ゼロの値は制限がないことを示します。AP_NO_IMPLICIT_LINKS の値は暗黙リンクが許可されないことを示します。

port_detail.def_data.implicit_deact_timer

限定リソース・リンクの非活動化タイマー (秒単位)。

implicit_limited_resource を AP_YES または AP_NO_SESSIONS に設定した場合は、このタイマーの設定時間の間リンク上にデータ・フローがなく、このリンクを使用しているセッションがない場合は、HPR 可能な暗黙的なリンクは自動的に非活動化されます。

implicit_limited_resource を AP_INACTIVITY に設定した場合は、このタイマーの設定時間内にデータ・フローがリンク上にないと、暗黙的なリンクは自動的に非活動化されます。

port_detail.def_data.act_xid_exchange_limit
活動化 XID 交換限度。

port_detail.def_data.nonact_xid_exchange_limit
非活動化 XID 交換限度。

port_detail.def_data.ls_xmit_rcv_cap
リンク・ステーションの送受信機能を指定します。値は次のいずれかです。

AP_LS_TWS
両方向同時

AP_LS_TWA
両方向交互

port_detail.def_data.max_ifrm_rcvd
肯定応答を送信する前にローカル・リンク・ステーションで受信できる I フレームの最大数。範囲は、1 ~ 127 です。

port_detail.def_data.target_pacing_count
求めるペーシング・ウィンドウ・サイズを示す 1 ~ 32,767 の数値。(現行バージョンの Communications Server for Linux では、この値は使用されません。)

port_detail.def_data.max_send_btu_size
送信できる BTU の最大サイズ。

port_detail.def_data.dlc_data
ポート・アドレス。dlc_data 構造体に関する詳細については、478 ページの『QUERY_LS』を参照してください。

def_data.implicit_cp_cp_sess_support
このポートを使用する暗黙リンク・ステーションに CP-CP セッションを許可するかどうかを指定します。値は次のいずれかです。

AP_YES 暗黙 LS に CP-CP セッションを許可します。

AP_NO 暗黙 LS に CP-CP セッションを許可しません。

def_data.implicit_limited_resource
このポートから切り離された暗黙リンク・ステーションを限定リソースとして定義するかどうかを指定します。値は次のいずれかです。

AP_NO 暗黙リンクは限定リソースではなく、自動的に非活動化されません。

AP_NO_SESSIONS
暗黙リンクは限定リソースであり、それを使用する活動セッションがなくなると、自動的に非活動化されます。

AP_INACTIVITY

暗黙リンクは限定リソースであり、それを使用する活動セッションがなくなるか、*link_deact_timer* フィールドで指定した期間にデータ・フローがないと、自動的に非活動化されます。

def_data.implicit_hpr_support

暗黙リンクで高性能経路指定 (HPR) をサポートするかどうかを指定します。値は次のいずれかです。

AP_YES 暗黙リンクで HPR をサポートします。

AP_NO 暗黙リンクで HPR をサポートしません。

def_data.implicit_link_lvl_error

SDLC の場合、このパラメーターは使用されません。

その他のリンク・タイプの場合、このパラメーターはリンク・レベルのエラー回復を使用して暗黙リンクに HPR トラフィックを送信するかどうかを指定します (AP_YES または AP_NO)。 *implicit_hpr_support* を AP_NO に設定した場合、このパラメーターは予約済みです。

def_data.default_tg_chars

デフォルト TG 特性。これらの特性は、このポートを使用する暗黙リンク・ステーションに使用されるか、TG 特性を明示的に定義されていない定義済みリンク・ステーションのデフォルト TG 特性として使用されます。これらのパラメーターについて詳しくは、135 ページの『DEFINE_LS』を参照してください。

port_detail.def_data.port_spec_data_len

ポート固有のデータの、スペースが入っていない長さ (バイト単位)。このデータのデータ構造体は *port_def_data* 構造体に続きますが、スペースが入るため、4 バイト目の境界から始まります。ポート固有のデータについて詳しくは、206 ページの『DEFINE_PORT』を参照してください。

port_detail.def_data.link_spec_data_len

リンク固有のデータの、スペースが入っていない長さ (バイト単位)。リンク固有のデータのデータ構造体は、ポート固有のデータのデータ構造体に続きますが、スペースが入るため、4 バイト目の境界から始まります。リンク固有のデータについて詳しくは、206 ページの『DEFINE_PORT』を参照してください。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PORT_NAME

list_options パラメーターを、指定した名前から始まるすべてのエントリーを表示する AP_LIST_INCLUSIVE に設定しましたが、
port_name パラメーターが有効ではありませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_PU

QUERY_PU は、ローカル PU およびそれに関連したリンクに関する情報を戻します。この verb は、使用するオプションに応じて、特定の PU または複数の PU に関する情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_pu
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code        */
    unsigned char  *buf_ptr;       /* pointer to buffer            */
    AP_UINT32      buf_size;       /* buffer size                  */
    AP_UINT32      total_buf_size; /* total buffer size required   */
    AP_UINT16      num_entries;    /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries     */
    unsigned char  list_options;   /* listing options              */
    unsigned char  reserv3;       /* reserved                      */
    unsigned char  pu_name[8];     /* PU name                      */
    unsigned char  host_attachment; /* host attachment filter       */
} QUERY_PU;

typedef struct pu_data
{
    AP_UINT16      overlay_size;   /* size of returned entry       */
    unsigned char  pu_name[8];     /* PU name                      */
    unsigned char  description[32]; /* resource description         */
    unsigned char  reserv1[16];   /* reserved                      */
    unsigned char  ls_name[8];    /* LS name                      */
    unsigned char  pu_sscp_sess_active; /* Is PU-SSCP session active */
    unsigned char  host_attachment; /* Host attachment              */
    SESSION_STATS pu_sscp_stats;  /* PU-SSCP session statistics  */
    unsigned char  sscp_id[6];    /* SSCP ID                      */
    unsigned char  conventional_lu_compression; /* compression for LU 0-3? */
    unsigned char  conventional_lu_cryptography; /* reserved                    */
    unsigned char  dddlu_supported; /* does the host support DDDL? */
    unsigned char  tcpcv_supported; /* does the host support TCPCVs? */
    unsigned char  dddlu_offline_supported; /* does the PU support sending */
    /* NMVT (power off) to host? */
    unsigned char  reserva[9];    /* reserved                      */
} PU_DATA;

typedef struct session_stats
{
    AP_UINT16      rcv_ru_size;    /* session receive RU size     */
    AP_UINT16      send_ru_size;  /* session send RU size        */
}
```

QUERY_PU

```
AP_UINT16    max_send_btu_size;    /* maximum send BTU size */
AP_UINT16    max_rcv_btu_size;     /* maximum rcv BTU size */
AP_UINT16    max_send_pac_win;     /* maximum send pacing window size */
AP_UINT16    cur_send_pac_win;     /* current send pacing window size */
AP_UINT16    max_rcv_pac_win;     /* maximum receive pacing window
/* size */
AP_UINT16    cur_rcv_pac_win;     /* current receive pacing window
/* size */
AP_UINT32    send_data_frames;     /* number of data frames sent */
AP_UINT32    send_fmd_data_frames; /* num fmd data frames sent */
AP_UINT32    send_data_bytes;     /* number of data bytes sent */
AP_UINT32    rcv_data_frames;     /* number of data frames received */
AP_UINT32    rcv_fmd_data_frames; /* num fmd data frames received */
AP_UINT32    rcv_data_bytes;     /* number of data bytes received */
unsigned char sidh;                /* session ID high byte (from
/* LFSID) */
unsigned char sidl;                /* session ID low byte (from LFSID)*/
unsigned char odai;                /* ODAI bit set */
unsigned char ls_name[8];          /* Link station name */
unsigned char pacing_type;        /* type of pacing in use */
} SESSION_STATS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_PU

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

PU の最大数 (そのデータが戻されます)。ある範囲でなく、特定の PU に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリから開始します。

AP_LIST_INCLUSIVE

pu_name パラメーターで指定したエントリから開始します。

AP_LIST_FROM_NEXT

pu_name パラメーターで指定したエントリの直後のエントリから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリを手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

pu_name

情報を求める PU の名前、または PU のリストへの索引として使用する名

前。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。

host_attachment

PU がホストに直接接続されるか、DLUR を介して接続されるかによって戻り情報をフィルターにかけるかどうかを指定します。値は次のいずれかです。

AP_DIRECT_ATTACHED

ホスト・システムに直接接続された PU についてのみの情報を戻します。

AP_DLUR_ATTACHED

DLUR がサポートする PU についてのみの情報を戻します。

AP_NONE

ホスト接続に関係なく、すべての PU に関する情報を戻します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

pu_data.overlay_size

戻された *pu_data* 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *pu_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

pu_data.pu_name

PU 名。この名前は、8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

pu_data.description

PU を記述したヌルで終了するテキスト・ストリング (その LS または 内部 PU の定義で指定したもの)。

pu_data.ls_name

この PU に関連したリンク・ステーションの名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

pu_data.pu_sscp_sess_active

PU-SSCP セッションが活動状態かどうかを指定します。値は次のいずれかです。

AP_YES PU-SSCP セッションは活動状態です。

AP_NO PU-SSCP セッションは活動状態ではありません。

pu_data.host_attachment

ローカル PU ホスト接続タイプ。

値は次のいずれかです。

AP_DIRECT_ATTACHED

PU はホスト・システムに直接接続されます。

AP_DLUR_ATTACHED

PU は DLUR によりサポートされています。

pu_data.pu_sscp_stats.rcv_ru_size

予約済み (常にゼロに設定される)。

pu_data.pu_sscp_stats.send_ru_size

予約済み (常にゼロに設定される)。

pu_data.pu_sscp_stats.max_send_btu_size

送信できる BTU の最大サイズ。

pu_data.pu_sscp_stats.max_rcv_btu_size

受信できる BTU の最大サイズ。

pu_data.pu_sscp_stats.max_send_pac_win

予約済み (常にゼロに設定される)。

pu_data.pu_sscp_stats.cur_send_pac_win

予約済み (常にゼロに設定される)。

pu_data.pu_sscp_stats.max_rcv_pac_win

予約済み (常にゼロに設定される)。

pu_data.pu_sscp_stats.cur_rcv_pac_win

予約済み (常にゼロに設定される)。

pu_data.pu_sscp_stats.send_data_frames

送信された通常フロー・データ・フレームの数。

- pu_data.pu_sscp_stats.send_fmd_data_frames*
送信された通常フロー FMD データ・フレームの数。
- pu_data.pu_sscp_stats.send_data_bytes*
送信された通常フロー・データ・バイトの数。
- pu_data.pu_sscp_stats.rcv_data_frames*
受信された通常フロー・データ・フレームの数。
- ppu_data.pu_sscp_stats.rcv_fmd_data_frames*
受信された通常フロー FMD データ・フレームの数。
- pu_data.pu_sscp_stats.rcv_data_bytes*
受信された通常フロー・データ・バイトの数。
- pu_data.pu_sscp_stats.sidh*
セッション ID の上位バイト。
- pu_data.pu_sscp_stats.sidl*
セッション ID の下位バイト。
- pu_data.pu_sscp_stats.odai*
起点宛先アサイナー・インディケータ。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合、BIND の送信側は、このフィールドをゼロに設定し、BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合は、このフィールドを 1 に設定します。
- pu_data.pu_sscp_stats.ls_name*
統計に関連したリンク・ステーション名。この名前は、8 バイトからなる ASCII 文字ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。
- pu_data.pu_sscp_stats.pacing_type*
PU-SSCP セッションで使用中の受信ペーシングのタイプ。このパラメータは AP_NONE に設定されます。
- pu_data.sscp_id*
従属 LU セッションの場合、このパラメータは、ローカル LU がマップされる PU についてホストからの ACTPU で受信された SSCP ID です。独立 LU セッションの場合、このパラメータは 0 (ゼロ) に設定されます。この値は、16 進値で表示される 6 バイトの配列です。
- pu_data.conventional_lu_compression*
この PU を使用して、LU 0 から 3 のセッションにデータ圧縮を要求するかどうかを指定します。値は次のいずれかです。
- AP_YES** ホストが要求した場合、データ圧縮は、この PU を使用して LU 0 から 3 のセッションに対して使用する必要があります。
- AP_NO** データ圧縮は、この PU を使用して LU 0 から 3 のセッションに対して使用しないでください。
- pu_data.dddlu_supported*
ホスト・システムが DDDL (従属型 LU の動的定義) をサポートするかどうかを指定します。値は次のいずれかです。
- AP_YES** ホストは DDDL をサポートします。

AP_NO ホストは DDDLU をサポートしません。

pu_data.tcpcv_supported

ホスト・システムが TCP/IP 情報制御ベクトル (0x64) の受信をサポートするかどうかを指定します。Communications Server for Linux は、このベクトルを使用して TN3270 または LUA クライアントの TCP/IP アドレス情報をホストへ送信できます。値は次のいずれかです。

AP_YES ホスト・サポートは TCP CV をサポートします。

AP_NO ホスト・サポートは TCP CV をサポートしません。

pu_data.dddlu_offline_supported

ローカル PU が、NMVT (電源オフ) メッセージのホストへの送信をサポートするかどうかを指定します。ホスト・システムが DDDLU (従属 LU の動的定義) をサポートしている場合、Communications Server for Linux は、動的に定義された LU を使用し終わったときに、NMVT (電源オフ) をホストへ送信します。これによって、ホストは、定義が不要になった際にその定義を除去することによって、リソースを節約することができます。

値は次のいずれかです。

AP_YES ローカル PU は、NMVT (電源オフ) メッセージをホストに送信します。

AP_NO ローカル PU は、NMVT (電源オフ) メッセージをホストに送信しません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PU_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*pu_name* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。


```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

AP_INVALID_PU_TYPE

pu_name パラメーターで指定された PU は、ローカル PU ではなく、リモート PU です。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_RAPI_CLIENTS

QUERY_RAPI_CLIENTS は、現在、マスターとして機能している Communications Server for Linux LAN 上の特定のサーバーの (AIX、Linux、または Windows 上の) Remote API Client に関する情報を戻します。

この verb はサーバーに対して発行する必要があります。この場合、サーバーでノードが始動されているかどうかについては関係ありません。

注: クライアントが Web サーバー経由でサーバーに接続されていて、クライアントのソフトウェアを停止するときに、Web サーバーが Communications Server for Linux マスター・サーバーへの接続を終了するまでに 1 ~ 2 分遅れる場合があります。これは QUERY_RAPI_CLIENTS verb がサーバーの使用を停止してからでも、少しの間クライアントをこの verb の中に包含しているためです。

VCB 構造体

```
typedef struct query_rapi_clients
{
    AP_UINT16          opcode;           /* verb operation code          */
    unsigned char     reserv2;          /* reserved                     */
    unsigned char     format;           /* reserved                     */
    AP_UINT16         primary_rc;       /* primary return code          */
    AP_UINT32         secondary_rc;     /* secondary return code        */
    unsigned char     *buf_ptr;         /* pointer to buffer            */
    AP_UINT32         buf_size;         /* buffer size                  */
    AP_UINT32         total_buf_size;   /* total buffer size required   */
    AP_UINT16         num_entries;      /* number of entries            */
    AP_UINT16         total_num_entries; /* total number of entries      */
    unsigned char     list_options;     /* listing options              */
    AP_UINT16         max_clients;      /* maximum number of clients    */
    unsigned char     sys_name[128];    /* RAPI Client to start query   */
} QUERY_RAPI_CLIENTS;

typedef struct rapi_client_info
{
    AP_UINT16         overlay_size;     /* overlay size                 */
    unsigned char     sys_name[128];    /* RAPI Client System name      */
    SNA_IP_ADDR      rapi_client_origin_ip_addr; /* IP addr client sends us    */
    SNA_IP_ADDR      rapi_client_adj_ip_addr; /* IP addr client comes in on */
    AP_UINT16         rapi_client_adj_port; /* port IP client comes in on */
} RAPI_CLIENT_INFO;
```

QUERY_RAPI_CLIENTS

```
typedef struct sna_ip_addr
{
    AP_UINT16      family;                /* IPv4 or IPv6          */
    union
    {
        unsigned char  ipv4_addr[4];
        unsigned char  ipv6_addr[16];
    } ip_addr;
} SNA_IP_ADDR;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_RAPI_CLIENTS

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

データを戻す必要があるクライアントの最大数。ある範囲でなく、特定のクライアントに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

クライアントのリスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

sys_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

sys_name パラメーターで指定したエントリーの直後のエントリーから開始します。

sys_name

リストのインデックスとして使用されるクライアントの完全修飾システム名 (newbox.this.co.uk など)。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

この名前は 1 ~ 128 文字の ASCII ストリングです。名前が 128 文字に満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

max_clients

Communications Server for Linux ソフトウェアが始動されてから、任意の時点でマスター・サーバーとしてサーバーを使用するクライアントの最大数。

rapi_client_info.overlay_size

戻された *rapi_client_info* 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *rapi_client_info* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

rapi_client_info.sys_name

クライアントの完全修飾システム名 (たとえば、`newbox.this.co.uk`)。

rapi_client_info.rapi_client_origin_ip_addr

クライアントの IP アドレス。

rapi_client_info.rapi_client_origin_ip_addr.family

クライアントに指定された TCP/IP アドレスのタイプ。可能な値は以下のとおりです。

AF_INET

IPv4 アドレス。ドット 10 進アドレスで指定します (たとえば、`193.1.11.100`)。

AF_INET6

IPv6 アドレス。コロン 16 進アドレスで指定します (たとえば、`2001:0db8:0000:0000:0000:0000:1428:57ab` または `2001:db8::1428:57ab`)。

注: 値 **AF_INET** および **AF_INET6** はシステム・ヘッダー・ファイルから取り込まれます。これらは、Communications Server for Linux を使用して定義する標準 `AF_*` 値ではありません。システム・ヘッダー・ファイル

は、Linux サーバーまたはクライアント上の `/usr/include/linux/socket.h`、および AIX クライアント上の `/usr/include/sys/socket.h` です。

使用する NOF アプリケーションでこれらの値についてテストする必要がある場合は、`#include` を使用してこのシステム・ファイルを `nof_c.h` ヘッダー・ファイルに組み込む必要があります。

rapi_client_info.rapi_client_origin_ip_addr.ip_addr.ipv4_addr

このフィールドは `family` パラメーターが `AF_INET` に設定されている場合のみ使用します。クライアント・コンピューターの IPv4 (ドット 10 進) アドレス。

rapi_client_info.rapi_client_origin_ip_addr.ip_addr.ipv6_addr

このフィールドは `family` パラメーターが `AF_INET6` に設定されている場合のみ使用します。クライアント・コンピューターの IPv6 (コロン 16 進) アドレス。

rapi_client_info.rapi_client_adj_ip_addr

クライアントが Communications Server for Linux に接続するために使用する IP アドレス。以下の項目の 1 つでも真の場合、*rapi_client_origin_ip_addr* と同じにすることはできません。

- クライアントは Web サーバー経由で接続される。
- クライアントは TCP/IP プロキシまたは NAT ルーターを経由して接続される。たとえば、Linux iptables ツール。
- クライアントに複数の IP アドレスがある。

rapi_client_info.rapi_client_adj_ip_addr.family

クライアントが Communications Server for Linux に接続するために使用する TCP/IP アドレスのタイプ。可能な値は以下のとおりです。

AF_INET

IPv4 アドレス。ドット 10 進アドレスで指定します (たとえば、193.1.11.100)。

AF_INET6

IPv6 アドレス。コロン 16 進アドレスで指定します (たとえば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)。

注: 値 `AF_INET` および `AF_INET6` はシステム・ヘッダー・ファイルから取り込まれます。これらは、Communications Server for Linux を使用して定義する標準 `AP_*` 値ではありません。システム・ヘッダー・ファイルは、Linux サーバーまたはクライアント上の `/usr/include/linux/socket.h`、および AIX クライアント上の `/usr/include/sys/socket.h` です。

使用する NOF アプリケーションでこれらの値についてテストする必要がある場合は、`#include` を使用してこのシステム・ファイルを `nof_c.h` ヘッダー・ファイルに組み込む必要があります。

rapi_client_info.rapi_client_adj_ip_addr.ip_addr.ipv4_addr

このフィールドは `family` パラメーターが `AF_INET` に設定されている場合の

み使用します。クライアントが Communications Server for Linux に接続するために使用する IPv4 (ドット 10 進) アドレス。

rapi_client_info.rapi_client_adj_ip_addr.ip_addr.ipv6_addr

このフィールドは *family* パラメーターが AF_INET6 に設定されている場合のみ使用します。クライアントが Communications Server for Linux に接続するために使用する IPv6 (コロン 16 進) アドレス。

rapi_client_info.rapi_client_adj_port

クライアントが Communications Server for Linux に接続するために使用する IP ポート番号。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_NODE_NAME

list_options パラメーターが、指定したノード名から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE または AP_LIST_FROM_NEXT に設定されていますが、*sys_name* パラメーターが指定されていないか、または無効でした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_RCF_ACCESS

QUERY_RCF_ACCESS は、Communications Server for Linux リモート・コマンド機能 (RCF) へのアクセス許可に関する情報、つまり、UNIX コマンド機能 (UCF) コマンドを実行するために使用するユーザー ID、およびサービス・ポイント・コマンド機能 (SPCF) を使用して管理コマンドを発行するための制約事項に関する情報を戻します。この情報は、以前は DEFINE_RCF_ACCESS を使用して設定されていました。SPCF および UCF については、「*Communications Server for Linux* 管理ガイド」を参照してください。

この *verb* は、ドメイン構成ファイルに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_rcf_access
{
    AP_UINT16      opcode;                /* Verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;        /* secondary return code */
    unsigned char  ucf_username[32];    /* UCF username */
    AP_UINT32      spcf_permissions;    /* SPCF permissions */
    unsigned char  reserv3[8];          /* Reserved */
} QUERY_RCF_ACCESS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_RCF_ACCESS

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

ucf_username
UCF ユーザーの Linux ユーザー名を指定します。このパラメーターはヌルで終了する ASCII ストリングです。

UCF コマンドはすべて、このユーザーのユーザー ID を使用し、このユーザーのために定義されたデフォルト・シェルとアクセス許可を使用して実行されます。

このパラメーターがヌル・ストリングに設定されている場合は、UCF アクセスが禁止されていることを示しています。

spcf_permissions
SPCF を使用してアクセスできる Communications Server for Linux 管理コマンドのタイプを指定します。このパラメーターは、AP_NONE (SPCF アクセスの禁止を示す) に設定するか、次の値の 1 つ以上 (論理 OR を使用して組み合わせる) に設定します。

AP_ALLOW_QUERY_LOCAL
QUERY_* verb を許可します。

AP_ALLOW_DEFINE_LOCAL
DEFINE_*、SET_*、DELETE_*、ADD_*、REMOVE_* の各 verb と INIT_NODE を許可します。

AP_ALLOW_ACTION_LOCAL
「アクション」 verb、つまり、START_*、STOP_*、ACTIVATE_*、DEACTIVATE_*、および APING、INITIALIZE_SESSION_LIMIT、CHANGE_SESSION_LIMIT、RESET_SESSION_LIMIT も許可します。

AP_ALLOW_QUERY_REMOTE

QUERY_* verb にリモート Communications Server for Linux ノードへのアクセスを許可します。

AP_ALLOW_DEFINE_REMOTE

DEFINE_*, SET_*, DELETE_*, ADD_*, REMOVE_*, および INIT_NODE verb に、リモート Communications Server for Linux ノードへのアクセスを許可します。

AP_ALLOW_ACTION_REMOTE

START_*, STOP_*, ACTIVATE_*, DEACTIVATE_*, APING, INITIALIZE_SESSION_LIMIT, CHANGE_SESSION_LIMIT, および RESET_SESSION_LIMIT verb に、リモート Communications Server for Linux ノードへのアクセスを許可します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_RTP_CONNECTION

QUERY_RTP_CONNECTION verb は、ノードがエンドポイントとなる高速トランスポート・プロトコル (RTP) 接続に関する情報を戻します。この verb は、使用するオプションに応じて、特定の RTP 接続または複数の RTP 接続に関する情報を入手するのに使用できます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_rtp_connection
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  *buf_ptr;        /* pointer to buffer            */
    AP_UINT32      buf_size;        /* buffer size                  */
    AP_UINT32      total_buf_size;  /* total buffer size required   */
    AP_UINT16      num_entries;     /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries      */
    unsigned char  list_options;    /* listing options              */
    unsigned char  reserv3;         /* reserved                     */
    unsigned char  rtp_name[8];     /* name of RTP connection      */
} QUERY_RTP_CONNECTION;

typedef struct rtp_connection_summary
{
    AP_UINT16      overlay_size;    /* size of returned entry       */
    unsigned char  rtp_name[8];     /* RTP connection name         */
    unsigned char  first_hop_ls_name[8]; /* LS name of first hop       */
    unsigned char  dest_node_name[17]; /* fully qualified name of    */
                                        /* destination node            */
}
```

QUERY_RTP_CONNECTION

```
    unsigned char  connection_type; /* LU-LU or CP-CP connection? */
    unsigned char  cos_name[8]; /* class of service name */
    AP_UINT16      num_sess_active; /* number of active sessions */
} RTP_CONNECTION_SUMMARY;

typedef struct rtp_connection_detail
{
    AP_UINT16      overlay_size; /* size of returned entry */
    unsigned char  rtp_name[8]; /* RTP connection name */
    unsigned char  first_hop_ls_name[8]; /* LS name of first hop */
    unsigned char  dest_node_name[17]; /* fully qualified name of
    /* destination node */
    unsigned char  isr_boundary_fn; /* is conn used for Boundary Func? */
    unsigned char  connection_type; /* LU-LU or CP-CP connection? */
    unsigned char  reserv1; /* reserved */
    unsigned char  cos_name[8]; /* class of service name */
    AP_UINT16      max_btu_size; /* maximum BTU size */
    AP_UINT32      liveness_timer; /* liveness timer */
    unsigned char  local_tcid[8]; /* local tcid */
    unsigned char  remote_tcid[8]; /* remote tcid */
    RTP_STATISTICS rtp_stats; /* RTP statistics */
    AP_UINT16      num_sess_active; /* number of active sessions */
    unsigned char  arb_mode; /* ARB-S or ARB-R? */
    unsigned char  reserv2[15]; /* reserved */
    AP_UINT16      rscv_len; /* length of appended RSCV */
} RTP_CONNECTION_DETAIL;
```

セッション詳細構造体のあとに、SNA フォーマットで定義された経路選択制御ベクトル (RSCV) が続く場合があります。この制御ベクトルは、ネットワークを介するセッション経路を定義し、BIND を実行します。この RSCV が組み込まれるのは、RSCV の保管するエンドポイントがノードの構成 (DEFINE_NODE で指定される) で示されている場合のみです。

```
typedef struct rtp_statistics
{
    AP_UINT32      bytes_sent; /* total number of bytes sent */
    AP_UINT32      bytes_received; /* total number of bytes received */
    AP_UINT32      bytes_resent; /* total number of bytes resent */
    AP_UINT32      bytes_discarded; /* total number of bytes discarded */
    AP_UINT32      packets_sent; /* total number of packets sent */
    AP_UINT32      packets_received; /* total number of packets received */
    AP_UINT32      packets_resent; /* total number of packets resent */
    AP_UINT32      packets_discarded; /* total number of packets discarded*/
    AP_UINT32      gaps_detected; /* gaps detected */
    AP_UINT32      send_rate; /* current send rate */
    AP_UINT32      max_send_rate; /* maximum send rate */
    AP_UINT32      min_send_rate; /* minimum send rate */
    AP_UINT32      receive_rate; /* current receive rate */
    AP_UINT32      max_receive_rate; /* maximum receive rate */
    AP_UINT32      min_receive_rate; /* minimum receive rate */
    AP_UINT32      burst_size; /* current burst size */
    AP_UINT32      up_time; /* total uptime of connection */
    AP_UINT32      smooth_rtt; /* smoothed round-trip time */
    AP_UINT32      last_rtt; /* last round-trip time */
    AP_UINT32      short_req_timer; /* SHORT_REQ timer duration */
    AP_UINT32      short_req_timeouts; /* number of SHORT_REQ timeouts */
    AP_UINT32      liveness_timeouts; /* number of liveness timeouts */
    AP_UINT32      in_invalid_sna_frames; /* number of invalid SNA frames */
    AP_UINT32      in_sc_frames; /* number of SC frames received */
    AP_UINT32      out_sc_frames; /* number of SC frames sent */
    AP_INT32      delay_change_sum; /* delay change sum */
    AP_UINT32      current_receiver_threshold; /* current ARB-R receiver threshold */
    AP_UINT32      minimum_receiver_threshold; /* minimum ARB-R receiver threshold */
    AP_UINT32      maximum_receiver_threshold;
```



```

AP_UINT32      sent_normals_count; /* maximum ARB-R receiver threshold */
AP_UINT32      sent_slowdowns_count; /* number of NORMALs sent */
AP_UINT32      rcvd_normals_count; /* number of NORMALs received */
AP_UINT32      rcvd_slowdowns_count; /* number of SLOWDOWNs received */
AP_UINT32      dcs_reset_count_non_heal; /* number of non-healing resets */
AP_UINT16      dcs_reset_count_healing; /* number of self-healing resets */
unsigned char  arb_mode; /* ARB mode (GREEN, YELLOW, RED) */
unsigned char  reserve[1]; /* reserved */
} RTP_STATISTICS;

```

指定パラメーター

提供パラメーターは次のとおりです。

opcode AP_QUERY_RTP_CONNECTION

buf_ptr リスト情報を書き込むことができるバッファーへのポインター。アプリケーションは VCB の終わりにデータを追加することができますが、その場合、*buf_ptr* を NULL に設定する必要があります。

buf_size
指定したバッファーのサイズ。

num_entries
RTP の最大数 (そのデータが戻されます)。ある範囲でなく、特定の接続に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options
各エントリーに必要な情報のレベルおよび Communications Server for Linux がデータを戻すためのリスト内の開始位置。

必要な情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY
要約情報のみ。

AP_DETAIL
詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST
リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE
rtp_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT
rtp_name パラメーターで指定したエントリーの直後のエントリーから開始します。

rtp_name
RTP 接続の名前。この値は、*list_options* パラメーターを AP_FIRST_IN_LIST

に設定した場合には無視されます。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを返します。

primary_rc
AP_OK

buf_size
バッファーに戻された情報の長さ。

total_buf_size
要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。この値は *buf_size* より大きい場合があります。

num_entries
実際に戻されたエントリーの数。

total_num_entries
戻されている可能性があるエントリーの合計数。この値は *num_entries* より大きい場合があります。

rtp_connection_summary.overlay_size
戻された `rtp_connection` 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 `rtp_connection_summary` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

rtp_connection_summary.rtp_name
RTP 接続の名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

rtp_connection_summary.first_hop_ls_name
RTP 接続の最初のホップのリンク・ステーションの名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

rtp_connection_summary.dest_node_name
セッションの RTP 部分に対する宛先制御点の完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

rtp_connection_summary.connection_type

RTP 接続のセッションのタイプを指定します。値は次のいずれかです。

AP_RTP_CP_CP_SESSION

RTP 接続は CP-CP セッションを伝送します。

AP_RTP_LU_LU_SESSION

RTP 接続は LU-LU セッションを伝送します。

AP_RTP_ROUTE_SETUP

RTP 接続は経路セットアップに使用します。

rtp_connection_summary.cos_name

RTP 接続が使用するサービス・クラスの名前。この名前は EBCDIC ストリングで、右側に EBCDIC スペースを入れます。

rtp_connection_summary.num_sess_active

この RTP 接続で活動状態であるセッションの数。

rtp_connection_detail.overlay_size

戻された `rtp_connection` 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 `rtp_connection_detail` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

rtp_connection_detail.rtp_name

RTP 接続の名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

rtp_connection_detail.first_hop_ls_name

RTP 接続の最初のホップのリンク・ステーションの名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

rtp_connection_detail.dest_node_name

セッションの RTP 部分に対する宛先制御点の完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

rtp_connection_detail.isr_boundary_fn

ローカル・ノードが HPR-APPN 境界機能を提供している ISR セッションに RTP 接続を使用しているかどうかを指定します。値は次のいずれかです。

AP_YES ローカル・ノードが HPR-APPN 境界機能を提供している ISR セッションに RTP 接続を使用しています。

QUERY_RTP_CONNECTION

AP_NO ローカル・ノードが HPR-APPN 境界機能を提供している ISR セッションに RTP 接続を使用していません。

rtp_connection_detail.connection_type

RTP 接続のセッションのタイプを指定します。値は次のいずれかです。

AP_RTP_CP_CP_SESSION

RTP 接続は CP-CP セッションを伝送します。

AP_RTP_LU_LU_SESSION

RTP 接続は LU-LU セッションを伝送します。

AP_RTP_ROUTE_SETUP

RTP 接続は経路セットアップに使用します。

rtp_connection_detail.cos_name

RTP 接続が使用するサービス・クラスの名前。この名前は EBCDIC ストリングで、右側に EBCDIC スペースを入れます。

rtp_connection_detail.max_btu_size

RTP 接続で使用される基本伝送単位 (BTU) の最大サイズ (バイト単位)。

rtp_connection_detail.liveness_timer

RTP 接続の活性タイマー (秒単位)。活性タイマーの設定期間内にこの接続にトラフィック・フローがない場合は、RTP は状況交換を開始し、パートナーがまだ存在しているかどうかを検査します。活性タイマー間隔を短くすると、回線障害がすぐに検出され、回線障害の際のパス・スイッチが高速に動作します。しかし、間隔が短すぎると、回線状況の検査が頻繁に行われるため、パフォーマンスが多少低下します。

rtp_connection_detail.local_tcid

RTP 接続のローカル TCID (トランスポート制御 ID)。

rtp_connection_detail.remote_tcid

RTP 接続のリモート TCID。

rtp_connection_detail.rtp_stats.bytes_sent

ローカル・ノードがこの RTP 接続で送信したバイトの合計数。

rtp_connection_detail.rtp_stats.bytes_received

ローカル・ノードがこの RTP 接続で受信したバイトの合計数。

rtp_connection_detail.rtp_stats.bytes_resent

転送中にバイトが消失したため、ローカル・ノードがこの RTP 接続で再送したバイトの合計数。

rtp_connection_detail.rtp_stats.bytes_discarded

RTP 接続の相手側が送信し、既に受信済みのデータの重複として廃棄されたバイトの合計数。

rtp_connection_detail.rtp_stats.packets_sent

ローカル・ノードがこの RTP 接続で送信したパケットの合計数。

rtp_connection_detail.rtp_stats.packets_received

ローカル・ノードがこの RTP 接続で受信したパケットの合計数。

rtp_connection_detail.rtp_stats.packets_resent

転送中にパケットが消失したため、ローカル・ノードがこの RTP 接続で再送したパケットの合計数。

rtp_connection_detail.rtp_stats.packets_discarded

RTP 接続の相手側が送信し、既に受信済みのデータの重複として廃棄されたパケットの合計数。

rtp_connection_detail.rtp_stats.gaps_detected

ローカル・ノードが検出したギャップの合計数。各ギャップは 1 つ以上の消失フレームに相当します。

rtp_connection_detail.rtp_stats.send_rate

この RTP 接続での現在の送信速度 (キロビット/秒)。この速度は、ARB (適応速度ベース) アルゴリズムで計算された最大許容送信速度です。RTP は ARB アルゴリズムを使用し、パートナーへの応答に要する時間の分析に基づいて、RTP によるデータ送信をどれくらい高速にできるかを計算します。

rtp_connection_detail.rtp_stats.max_send_rate

この RTP 接続での最大送信速度 (キロビット/秒)。

rtp_connection_detail.rtp_stats.min_send_rate

この RTP 接続での最小送信速度 (キロビット/秒)。

rtp_connection_detail.rtp_stats.receive_rate

この RTP 接続での現在の受信速度 (キロビット/秒)。この速度は、最後の測定間隔について計算された実際の速度です。

rtp_connection_detail.rtp_stats.max_receive_rate

この RTP 接続での最大受信速度 (キロビット/秒)。

rtp_connection_detail.rtp_stats.min_receive_rate

この RTP 接続での最小受信速度 (キロビット/秒)。

rtp_connection_detail.rtp_stats.burst_size

この RTP 接続での現在のバースト・サイズ (バイト単位)。

rtp_connection_detail.rtp_stats.up_time

この RTP 接続が活動状態であった合計秒数。

rtp_connection_detail.rtp_stats.smooth_rtt

ローカル・ノードとパートナー RTP ノード間の往復の平滑化測定値 (ミリ秒単位)。

rtp_connection_detail.rtp_stats.last_rtt

ローカル・ノードとパートナー RTP ノード間の最終往復の測定値 (ミリ秒単位)。

rtp_connection_detail.rtp_stats.short_req_timer

状況交換要求への応答待ち時間 (ミリ秒単位)。タイマー間隔が短くなると、障害の検出処理は高速になりますが、パフォーマンスが低下します。

rtp_connection_detail.rtp_stats.short_req_timeouts

この RTP 接続について *short_req_timer* が満了した合計回数。

QUERY_RTP_CONNECTION

rtp_connection_detail.rtp_stats.liveness_timeouts

この RTP 接続について活性タイマーが満了した合計回数。接続が *liveness_timer* パラメーターで指定した期間内にアイドルになると、活性タイマーは満了します。

rtp_connection_detail.rtp_stats.in_invalid_sna_frames

この RTP 接続で受信され、無効になったため廃棄された SNA フレームの合計数。

rtp_connection_detail.rtp_stats.in_sc_frames

この RTP 接続で受信されたセッション制御フレームの合計数。

rtp_connection_detail.rtp_stats.out_sc_frames

この RTP 接続で送信されたセッション制御フレームの合計数。

rtp_connection_detail.rtp_stats.delay_change_sum

この RTP 接続で ARB-R アルゴリズムによって現在保持されている遅延変更の合計値。

rtp_connection_detail.rtp_stats.current_receiver_threshold

この RTP 接続で ARB-R アルゴリズムによって現在保持されている受信側しきい値。

rtp_connection_detail.rtp_stats.minimum_receiver_threshold

この RTP 接続で ARB-R アルゴリズムによって現在保持されている受信側しきい値の最小値。

rtp_connection_detail.rtp_stats.maximum_receiver_threshold

この RTP 接続で ARB-R アルゴリズムによって現在保持されている受信側しきい値の最大値。

rtp_connection_detail.rtp_stats.sent_normals_count

この RTP 接続で ARB-R アルゴリズムによって送信された NORMAL フィードバックの ARB-R セグメント数。

rtp_connection_detail.rtp_stats.sent_slowdowns_count

この RTP 接続で ARB-R アルゴリズムによって送信された SLOWDOWN1 および SLOWDOWN2 フィードバックの ARB-R セグメント数。

rtp_connection_detail.rtp_stats.rcvd_normals_count

この RTP 接続で ARB-R アルゴリズムによって受信された NORMAL フィードバックの ARB-R セグメント数。

rtp_connection_detail.rtp_stats.rcvd_slowdowns_count

この RTP 接続で ARB-R アルゴリズムによって受信された SLOWDOWN1 および SLOWDOWN2 フィードバックの ARB-R セグメント数。

rtp_connection_detail.rtp_stats.dcs_reset_count_non_heal

この RTP 接続で 通常の ARB-R 処理の一部として行われた遅延変更リセットの合計回数。

rtp_connection_detail.rtp_stats.dcs_reset_count_healing

この RTP 接続で 通常の ARB-R アルゴリズムの自己回復が行われた遅延変更リセットの合計回数。

rtp_connection_detail.rtp_stats.arb_mode

この RTP 接続の現在の ARB-R 状況モード。値は次のいずれかです。

- 0 GREEN
- 1 YELLOW
- 2 RED

rtp_connection_detail.num_sess_active

この RTP 接続で活動状態であるセッションの数。

rtp_connection_detail.arb_mode

この RTP 接続で使用中の ARB モードを指定します。値は次のいずれかです。

AP_ARB_S

標準モード ARB。

AP_ARB_R

応答モード ARB。

AP_UNKNOWN

RTP 接続がまだ確立されていないため、ARB はまだ決定されていません。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_RTP_CONNECTION

list_options パラメーターを、指定した名前から始まるすべてのエントリーを表示する `AP_LIST_INCLUSIVE` に設定しましたが、*rtp_name* パラメーターが有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_RTP_TUNING

`QUERY_RTP_TUNING` は、今後の RTP 接続に使用されるパラメーターに関する情報を戻します。この情報は `DEFINE_RTP_TUNING` を使用して事前にセットアップされていたものです。

VCB 構造体

```
typedef struct query_rtp_tuning
{
    AP_UINT16      opcode;           /* Verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;          /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  path_switch_attempts; /* number of path switch attempts */
    unsigned char  short_req_retry_limit; /* short request timer retry limit */
    AP_UINT16      path_switch_times[4]; /* path switch times            */
    AP_UINT32      refifo_cap;      /* maximum for refifo timer     */
    AP_UINT32      srt_cap;         /* maximum for short request timer */
    unsigned char  reserved[80];    /* reserved                     */
} QUERY_RTP_TUNING;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_RTP_TUNING

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

path_switch_attempts
新規 RTP 接続で設定を試行されるパス・スイッチの数。

short_req_retry_limit
Communications Server for Linux が RTP 接続の切断およびパス・スイッチ処理の開始を決定する前に、状況要求が送信される回数。

path_switch_times
Communications Server for Linux が切断された RTP 接続のパス・スイッチを試みる時間長 (秒単位)。このパラメーターには、それぞれの有効な伝送優先順位に対する 4 つの個別の時間制限として、AP_LOW、AP_MEDIUM、AP_HIGH、および AP_NETWORK の順序で指定します。

refifo_cap
RTP プロトコルは「Re-FIFO Timer」と呼ばれるタイマーを使用します。このタイマーの値はプロトコルの一部として計算されますが、このパラメーターには最大値 (タイマーはこの値を超えて増やすことができない) をミリ秒で指定します。ある状態では、この最大値の設定によりパフォーマンスを改善できます。0 (ゼロ) 値に設定するという事は、タイマーの制限がないということであり、プロトコルで計算された、どのような値でも取り込むことができます。

srt_cap RTP プロトコルは「Short Request Timer」と呼ばれるタイマーを使用します。このタイマーの値はプロトコルの一部として計算されますが、このパラメーターには最大値 (タイマーはこの値を超えて増やすことができない) をミリ秒で指定します。ある状態では、この最大値の設定によりパフォーマンス

スを改善できます。0 (ゼロ) 値に設定するということは、タイマーの制限がないということであり、プロトコルで計算された、どのような値でも取り込むことができます。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_SECURITY_ACCESS_LIST

QUERY_SECURITY_ACCESS_LIST は、Communications Server for Linux 構成ファイルで定義されたセキュリティー・アクセス・リストに関する情報を戻します。これは、使用するオプションに応じて、単一のリストまたは複数のリストに関する情報を戻すことができます。

VCB 構造体

```
typedef struct query_security_access_list
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;               /* reserved                     */
    unsigned char  format;                /* reserved                     */
    AP_UINT16      primary_rc;            /* primary return code          */
    AP_UINT32      secondary_rc;          /* secondary return code        */
    unsigned char  *buf_ptr;              /* pointer to buffer            */
    AP_UINT32      buf_size;               /* buffer size                  */
    AP_UINT32      total_buf_size;        /* total buffer size required   */
    AP_UINT16      num_entries;            /* number of entries            */
    AP_UINT16      total_num_entries;     /* total number of entries      */
    unsigned char  list_options;          /* listing options              */
    unsigned char  reserv3;               /* reserved                     */
    unsigned char  list_name[14];         /* Security Access List name    */
    unsigned char  user_name[10];         /* user name                    */
    AP_UINT32      num_init_users;        /* number of users for first    */
                                          /* list when starting in middle */
    AP_UINT32      num_last_users;        /* number of users on last     */
                                          /* overlay if last list is     */
                                          /* incomplete                   */
    unsigned char  last_list_incomplete; /* set to AP_YES if user data  */
                                          /* for last list is incomplete */
} QUERY_SECURITY_ACCESS_LIST;

typedef struct security_access_detail
{
    AP_UINT16      overlay_size;           /* size of returned entry       */
    unsigned char  list_name[14];         /* list name                    */
    unsigned char  reserv1[2];            /* reserved                     */
    AP_UINT32      num_filtered_users;     /* number of users returned     */
    SECURITY_LIST_DEF def_data;           /* list definition              */
} SECURITY_ACCESS_DETAIL;

typedef struct security_list_def
{
    unsigned char  description[32];        /* description                   */
    unsigned char  reserv3[16];           /* reserved                     */
    AP_UINT32      num_users;              /* number of users in list      */
    unsigned char  reserv2[16];           /* reserved                     */
} SECURITY_LIST_DEF;
```

QUERY_SECURITY_ACCESS_LIST

```
typedef struct security_user_data
{
    AP_UINT16      sub_overlay_size;    /* reserved          */
    unsigned char  user_name[10];      /* user name         */
} SECURITY_USER_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_SECURITY_ACCESS_LIST

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

セキュリティー・アクセス・リストの最大数 (そのセキュリティー・アクセス・リストのデータが戻されます)。この数には、部分セキュリティー・アクセス・リストのエントリーが含まれます (部分セキュリティー・アクセス・リストにユーザー名を指定できるため、戻りデータにリスト内の最初のユーザー名は含まれません)。

ある範囲でなく、特定のセキュリティー・アクセス・リストに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

最初のセキュリティー・アクセス・リストの最初のユーザー名から開始します。

AP_LIST_INCLUSIVE

提供されたセキュリティー・アクセス・リスト名およびユーザー名で指定したエントリーから開始するか、ユーザー名を指定していない場合は、指定したセキュリティー・アクセス・リストの最初のユーザー名から開始します。

AP_LIST_FROM_NEXT

ユーザー名を指定している場合には、指定されたユーザーのすぐ後のユーザーから開始します。ユーザー名を指定していない場合には、指定されたセキュリティー・アクセス・リストの最初のユーザーから開始します。

リストはセキュリティー・アクセス・リスト名の順に配列され、さらに各セキュリティー・アクセス・リスト内でユーザー名の順で配列されます。リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

list_name

情報を求めるセキュリティー・アクセス・リストの名前、またはセキュリティー・アクセス・リストのリストの索引として使用する名前。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。この名前は、1 ~ 14 文字の ASCII スtring で、名前が 14 文字に満たない場合は、右側にスペースを入れます。

user_name

指定したセキュリティー・アクセス・リストの特定のユーザー名で始まる情報を戻すには、このパラメーターをそのユーザー名に設定します。指定したセキュリティー・アクセス・リストの最初のユーザー名で始まる情報を戻すには、このパラメーターを 10 桁の 2 進ゼロに設定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

バッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。この値は *buf_size* より大きい場合があります。

total_num_entries

戻されている可能性があるエントリーの合計数。この値は *num_entries* より大きい場合があります。

num_entries

実際に戻されたエントリーの数。最後のエントリーが不完全な場合があり、それは *last_list_incomplete* パラメーターで示されます。

num_init_users

user_name パラメーターがゼロ以外の値に設定された結果、戻されたデータ内の最初のセキュリティー・アクセス・リストに関する情報がリスト内の最初のユーザーから始まらない場合、このパラメーターは、戻されたデータに入っているこのリストのユーザー名構造体の数を示します。それ以外の場合は、このパラメーターは使用されません。

num_last_users

last_list_incomplete パラメーターが最後のリストのデータが不完全であることを示している場合、このパラメーターは、戻されたデータに入っているこのリストのユーザー名構造体の数を示します。(このリストに戻された *num_filtered_users* パラメーターは、使用可能なユーザー名構造体の合計数を示します。) それ以外の場合は、このパラメーターは使用されません。

last_list_incomplete

最後のセキュリティー・アクセス・リストに関する情報のデータが不完全であることを示します。値は次のいずれかです。

AP_YES 最後のセキュリティー・アクセス・リストの完全なデータは大き

QUERY_SECURITY_ACCESS_LIST

ざるため、データ・バッファに入りませんでした。少なくとも 1 つのユーザー名構造体が含まれていますが、データ・バッファに入っていないユーザー名構造体はまだあります。 *num_last_users* パラメーターは、ユーザー名構造体に戻された数を示しています。アプリケーションはさらにこの *verb* を発行して残りのデータを取得することができます。

AP_NO 最後のリストのデータは完全です。

データ・バッファ内の各エントリは、次のパラメーターで構成されています。

security_access_detail.list_name

セキュリティー・アクセス・リストの名前。これは 1 ~ 14 文字の ASCII スtring です。

security_access_detail.num_filtered_users

このセキュリティー・アクセス・リスト内のユーザー名の合計数。

security_access_detail.def_data.description

セキュリティー・アクセス・リストを記述したヌルで終了するテキスト・String (そのリストの定義で指定したもの)。

security_access_detail.def_data.num_users

セキュリティー・アクセス・リスト内のユーザーの合計数。

これがデータ・バッファ内の最後のリストで、*last_list_incomplete* パラメーターが **AP_YES** に設定されている場合、このリストに戻されたユーザー名構造体の合計数は、*num_last_users* パラメーターで指定した数になります。したがって、*num_users* より少なくなります。

リスト内のユーザー名ごとに、*security_user_data* 構造体が以下の情報と一緒に戻されます。

user_name

ユーザーの名前。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LIST_NAME

list_options パラメーターを **AP_LIST_INCLUSIVE** に設定しましたが、*list_name* パラメーターで指定した値は、指定されたセキュリティー・アクセス・リストに定義されたユーザー名と一致しませんでした。

AP_INVALID_USER_NAME

list_options パラメーターを **AP_LIST_INCLUSIVE** に設定しました

が、*user_name* パラメーターで指定した値は、指定されたセキュリティ・アクセス・リストに定義されたユーザー名と一致しませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_SESSION

QUERY_SESSION は、特定のローカル LU のセッションに関するリスト情報を戻します。

この verb は、使用するオプションに応じて、特定のセッションまたはある範囲のセッションに関する要約情報または詳細情報を入手するために使用できます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_session
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  *buf_ptr;        /* pointer to buffer            */
    AP_UINT32      buf_size;        /* buffer size                  */
    AP_UINT32      total_buf_size;  /* total buffer size required   */
    AP_UINT16      num_entries;     /* number of entries            */
    AP_UINT16      total_num_entries; /* total number of entries      */
    unsigned char  list_options;    /* listing options              */
    unsigned char  reserv3;         /* reserved                     */
    unsigned char  lu_name[8];      /* LU name                      */
    unsigned char  lu_alias[8];     /* LU alias                     */
    unsigned char  plu_alias[8];    /* partner LU alias             */
    unsigned char  fqplu_name[17];  /* fully qualified partner LU name */
    unsigned char  mode_name[8];    /* mode name                    */
    unsigned char  session_id[8];   /* session ID                   */
} QUERY_SESSION;

typedef struct session_summary
{
    AP_UINT16      overlay_size;    /* size of returned entry       */
    unsigned char  plu_alias[8];    /* partner LU alias             */
    unsigned char  fqplu_name[17];  /* fully qualified partner LU name */
    unsigned char  reserv3[1];      /* reserved                     */
    unsigned char  mode_name[8];    /* mode name                    */
}
```

QUERY_SESSION

```

    unsigned char  session_id[8];      /* session ID */
    FQPCID        fqpcid;             /* fully qualified procedure */
                                        /* correlator ID */
} SESSION_SUMMARY;

typedef struct session_detail
{
    AP_UINT16     overlay_size;       /* size of returned entry */
    unsigned char plu_alias[8];       /* partner LU alias */
    unsigned char fqplu_name[17];     /* fully qualified partner LU name */
    unsigned char reserv3[1];         /* reserved */
    unsigned char mode_name[8];       /* mode name */
    unsigned char session_id[8];      /* session ID */
    FQPCID        fqpcid;             /* fully qualified procedure */
                                        /* correlator ID */

    unsigned char cos_name[8];        /* Class of Service name */
    unsigned char trans_pri;          /* Transmission priority: */
    unsigned char ltd_res;            /* Session spans a limited resource */
    unsigned char polarity;           /* Session polarity */
    unsigned char contention;         /* Session contention */
    SESSION_STATS sess_stats;         /* Session statistics */
    unsigned char reserv3a;           /* reserved */
    unsigned char sscp_id[6];         /* SSCP ID of host */
    unsigned char reserva;            /* reserved */
    AP_UINT32     session_start_time; /* start time of the session */
    AP_UINT16     session_timeout;    /* session timeout */
    unsigned char cryptography;       /* reserved */
    unsigned char reservb[5];         /* reserved */
    unsigned char comp_in_series;     /* reserved */
    unsigned char plu_slu_comp_lvl;   /* PLU to SLU compression level */
    unsigned char slu_plu_comp_lvl;   /* SLU to PLU compression level */
    unsigned char rscv_len;           /* Length of following RSCV */
} SESSION_DETAIL;

```

セッション詳細構造体のあとに、SNA フォーマットで定義された経路選択制御ベクトル (RSCV) が続く場合があります。この制御ベクトルは、ネットワークを介するセッション経路を定義し、BIND を実行します。この RSCV が組み込まれるのは、RSCV の保管するエンドポイントがノードの構成 (DEFINE_NODE で指定される) で示されている場合のみです。

```

typedef struct fqpcid
{
    unsigned char  pcid[8];           /* procedure correlator identifier */
    unsigned char  fqcp_name[17];     /* originator's network qualified */
                                        /* CP name */
    unsigned char  reserve3[3];       /* reserved */
} FQPCID;

typedef struct session_stats
{
    AP_UINT16     rcv_ru_size;        /* session receive RU size */
    AP_UINT16     send_ru_size;       /* session send RU size */
    AP_UINT16     max_send_btu_size;  /* Maximum send BTU size */
    AP_UINT16     max_rcv_btu_size;   /* Maximum rcv BTU size */
    AP_UINT16     max_send_pac_win;   /* Maximum send pacing window size */
    AP_UINT16     cur_send_pac_win;   /* Current send pacing window size */
    AP_UINT16     max_rcv_pac_win;    /* Maximum receive pacing window */
                                        /* size */
    AP_UINT16     cur_rcv_pac_win;    /* Current receive pacing window */
                                        /* size */
    AP_UINT32     send_data_frames;   /* Number of data frames sent */
    AP_UINT32     send_fmd_data_frames; /* Num fmd data frames sent */
    AP_UINT32     send_data_bytes;    /* Number of data bytes sent */
    AP_UINT32     rcv_data_frames;    /* Number of data frames received */
    AP_UINT32     rcv_fmd_data_frames; /* Num fmd data frames received */
    AP_UINT32     rcv_data_bytes;     /* Number of data bytes received */
    unsigned char sidh;               /* Session ID high byte (from LFSID)*/
}

```

```

unsigned char  sid1;           /* Session ID low byte (from LFSID) */
unsigned char  odai;          /* ODAI bit set */
unsigned char  ls_name[8];    /* Link station name (or RTP name) */
unsigned char  pacing_type;   /* type of pacing in use */
} SESSION_STATS;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_SESSION

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

セッションの最大数 (それらのセッションについてのデータが戻されます)。ある範囲でなく、特定のセッションに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

session_id パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

session_id パラメーターで指定したエントリーの直後のエントリーから開始します。

list_options パラメーターを AP_LIST_INCLUSIVE または AP_LIST_FROM_NEXT に設定した場合、指定したローカル LU (*lu_name* または *lu_alias*)、パートナー LU (*plu_alias* または *fqplu_name*)、および *mode_name* の組み合わせは、セッションのリストの索引として使用されます。

リストの順序と、アプリケーションでそのリストから特定のエントリーを手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

lu_name

LU 名。この名前は、8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。LU を LU 名ではなく、別名で識別することを指定するには、このパラメーターを 8 桁の 2 進ゼロに設定し、次のパラメーターで LU 別名を指定します。ローカル CP に関連した LU (デフォルト LU) を指定するには、*lu_name* と *lu_alias* の両方をすべて 2 進ゼロに設定します。

lu_alias

ローカルで定義された LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。このパラメーターは、*lu_name* を 8 桁の 2 進ゼロに設定した場合のみ使用され、それ以外の場合は無視されます。ローカル CP に関連した LU (デフォルト LU) を指定するには、*lu_name* と *lu_alias* の両方をすべて 2 進ゼロに設定します。

plu_alias

パートナー LU 別名。特定のパートナー LU に関連したセッションに関する情報を戻すには、パートナー LU 別名 (このパラメーターで) またはパートナー LU の完全修飾名 (次のパラメーターで) を指定します。パートナー LU で選別しないですべてのセッションに関する情報を戻すには、これらのパラメーターの両方のビットをゼロに設定します。

これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。LU を別名ではなく、LU 名で識別することを指定するには、このパラメーターを 8 桁の 2 進ゼロに設定し、次のパラメーターで LU 名を指定します。

fqplu_name

パートナー LU の完全修飾ネットワーク名。このパラメーターは、*plu_alias* を 8 桁の 2 進ゼロに設定した場合のみ使用され、それ以外の場合は無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

mode_name

モード名フィルター。特定のモードに関連したセッションのみにに関する情報を戻すには、モード名を指定します。パートナー LU 名も指定する必要があります (上記の 2 つのパラメーターのいずれかを使用します)。モード名で選別しないですべてのセッションに関する情報を戻すには、このパラメーターの 8 桁の 2 進ゼロに設定します。

この名前は、8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

session_id

8 バイトからなるセッション ID。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

buf_size
指定したバッファーに戻された情報の長さ。

total_buf_size
要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries
データ・バッファー内に戻されたエントリーの数。

total_num_entries
使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。
データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

session_summary.overlay_size
戻された *session_summary* 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。
アプリケーションが戻されたバッファーを調べて、各 *session_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

session_summary.plu_alias
パートナー LU 別名。これは 8 バイトからなる ASCII 文字ストリングで、右側に ASCII スペースを入れます。

session_summary.fqplu_name
パートナー LU の完全修飾ネットワーク名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

session_summary.mode_name
モード名。これは 8 バイトからなるタイプ A の EBCDIC ストリング (英字で始まる) で、右側に EBCDIC スペースを入れます。

session_summary.session_id
8 バイトからなるセッション ID。

session_summary.fqpcid.pcid

プロシージャ相関関係子 ID。これは 8 バイトからなる 16 進数ストリングです。

session_summary.fqpcid.fqcp_name

CP の完全修飾名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

session_detail.overlay_size

戻された *session_detail* 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *session_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

session_detail.plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII 文字ストリングで、右側に ASCII スペースを入れます。

session_detail.fqplu_name

パートナー LU の完全修飾ネットワーク名。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

session_detail.mode_name

モード名。これは 8 バイトからなるタイプ A の EBCDIC ストリング (英字で始まる) で、右側に EBCDIC スペースを入れます。

session_detail.session_id

8 バイトからなるセッション ID。

session_detail.fqpcid.pcid

プロシージャ相関関係子 ID。これは 8 バイトからなる 16 進数ストリングです。

session_detail.fqpcid.fqcp_name

制御点の完全修飾名。この名前は、17 バイトの EBCDIC ストリングで、この右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A ストリング文字からなるネットワーク名で構成されます。

session_detail.cos_name

サービス・クラス名。この名前は、8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

session_detail.trans_pri

伝送優先順位。値は次のいずれかです。

AP_LOW

AP_MEDIUM

AP_HIGH

AP_NETWORK

session_detail.ltd_res

セッションで限定リソース・リンクを使用するかどうかを指定します。値は次のいずれかです。

AP_YES セッションでは、限定リソース・リンクを使用します。

AP_NO セッションでは、限定リソース・リンクは使用しません。

session_detail.polarity

セッションの極性を指定します。値は次のいずれかです。

AP_PRIMARY

AP_SECONDARY

session_detail.contention

セッションがローカル LU の競合勝者セッションまたは競合敗者セッションのどちらであるかを指定します。値は次のいずれかです。

AP_CONWINNER

競合勝者セッション

AP_CONLOSER

競合敗者セッション

session_detail.sess_stats.rcv_ru_size

受信 RU の最大サイズ。

session_detail.sess_stats.send_ru_size

送信 RU の最大サイズ。

session_detail.sess_stats.max_send_btu_size

送信できる BTU の最大サイズ。

session_detail.sess_stats.max_rcv_btu_size

受信できる BTU の最大サイズ。

session_detail.sess_stats.max_send_pac_win

このセッションの送信ペーシング・ウィンドウの最大サイズ。

session_detail.sess_stats.cur_send_pac_win

このセッションの送信ペーシング・ウィンドウの現行サイズ。

session_detail.sess_stats.max_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの最大サイズ。

QUERY_SESSION

session_detail.sess_stats.cur_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの現行サイズ。

session_detail.sess_stats.send_data_frames

送信された通常フロー・データ・フレームの数。

session_detail.sess_stats.send_fmd_data_frames

送信された通常フロー FMD データ・フレームの数。

session_detail.sess_stats.send_data_bytes

送信された通常フロー・データ・バイトの数。

session_detail.sess_stats.rcv_data_frames

受信された通常フロー・データ・フレームの数。

session_detail.sess_stats.rcv_fmd_data_frames

受信された通常フロー FMD データ・フレームの数。

session_detail.sess_stats.rcv_data_bytes

受信された通常フロー・データ・バイトの数。

session_detail.sess_stats.sidh

セッション ID の上位バイト。

session_detail.sess_stats.sidl

セッション ID の下位バイト。

session_detail.sess_stats.odai

起点宛先アサイナー・インディケータ。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合は、BIND の送信側は、このフィールドをゼロに設定します。BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合は、このフィールドを 1 に設定します。

session_detail.sess_stats.ls_name

統計に関連したリンク・ステーション名。これは 8 バイトからなる ASCII スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。このパラメータは、セッション・データの転送経路となるリンクとセッション統計を関連付けるために使用できます。

session_detail.sess_stats.pacing_type

このセッションで使用している受信ペーシングのタイプ。値は次のいずれかです。

AP_NONE

AP_FIXED

AP_ADAPTIVE

session_detail.duplex_support

BIND で折衝された会話二重サポートを戻します。値は次のいずれかです。

AP_HALF-DUPLEX

半二重会話のみがサポートされます。

AP_FULL_DUPLEX

全二重セッションと半二重セッションの両方がサポートされます。優先データもサポートされます。

session_detail.sscp_id

従属 LU セッションの場合、このパラメーターは、ローカル LU がマップされる PU についてホストからの ACTPU で受信された SSCP ID です。独立 LU セッションの場合、このパラメーターは 0 (ゼロ) に設定されます。

session_detail.session_start_time

CP の起動とセッションの活動化の間の時間 (0.01 秒単位)。照会の処理時にセッションが完全に活動状態でない場合、このパラメーターのビットは 0 (ゼロ) に設定されます。

session_detail.session_timeout

このセッションに関連したタイムアウト。タイムアウトは次の値からとられます。

- ローカル LU に関連した LU 6.2 タイムアウト
- リモート LU に関連した LU 6.2 タイムアウト
- モード・タイムアウト
- グローバル・タイムアウト
- 限定リソース・タイムアウト (このセッションが限定リソース・リンクで実行されている場合)

session_detail.plu_slu_comp_lvl

1 次 LU (PLU) から 2 次 LU (SLU) に送信されるデータの圧縮レベルを指定します。値は次のいずれかです。

AP_NONE

圧縮は使用されません。

AP_RLE_COMPRESSION

実行長さエンコード (RLE) 圧縮が使用されます。

AP_LZ9_COMPRESSION

LZ9 圧縮が使用されます。

AP_LZ10_COMPRESSION

LZ10 圧縮が使用されます。

session_detail.slu_plu_comp_lvl

2 次 LU (SLU) から 1 次 LU (PLU) に送信されるデータの圧縮レベルを指定します。値は次のいずれかです。

AP_NONE

圧縮は使用されません。

AP_RLE_COMPRESSION

実行長さエンコード (RLE) 圧縮が使用されます。

AP_LZ9_COMPRESSION

LZ9 圧縮が使用されます。

AP_LZ10_COMPRESSION

LZ10 圧縮が使用されます。

session_detail.rscv_len

session_detail 構造体に追加される RSCV の長さ。(何も追加しない場合は、長さはゼロです。)

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LU_ALIAS

指定した *lu_alias* パラメーターが、有効ではありませんでした。

AP_INVALID_LU_NAME

指定した *lu_name* パラメーターが、有効ではありませんでした。

AP_INVALID_SESSION_ID

list_options パラメーターが、指定した値から始まるすべてのエンターリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*session_id* パラメーターは無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_SNA_NET

QUERY_SNA_NET は、**sna.net** ファイルで定義されているように、バックアップ・マスター・サーバーとして動作するサーバーに関する情報を戻します。この *verb* は、使用するオプションに応じて、特定のサーバーに関する情報、または複数のサーバーに関する情報を入手するために使用できます。

このファイル内でのサーバー名の配列には、重要な意味があります。このファイルで最初にリストされているサーバーは、活動化されているときには常にマスターになります。2 番目のサーバーは、最初のサーバーが活動化されていない場合にマスターになります。3 番目のサーバーは、最初と 2 番目のサーバーがどちらも活動化されていない場合にマスターになります。このため、QUERY_SNA_NET で戻されるサーバー名のリストは、このファイル内での順序と同じ順序になります。戻される名前は、その他の QUERY_* *verb* と同様、名前の長さやアルファベット順には配列されません。

この verb は、**sna.net** ファイルに対して発行する必要があります。

VCB 構造体

```
typedef struct query_sna_net
{
    AP_UINT16      opcode;           /* Verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* Primary return code      */
    AP_UINT32      secondary_rc;    /* Secondary return code    */
    unsigned char  *buf_ptr;        /* pointer to buffer        */
    AP_UINT32      buf_size;        /* buffer size              */
    AP_UINT32      total_buf_size;  /* total buffer size required */
    AP_UINT16      num_entries;     /* number of entries        */
    AP_UINT16      total_num_entries; /* total number of entries  */
    unsigned char  list_options;    /* listing options          */
    unsigned char  security;        /* reserved                  */
    unsigned char  domain_name[64]; /* domain name              */
    unsigned char  server_name[128]; /* master or backup server name */
    unsigned char  reserv4[4];      /* reserved                  */
} QUERY_SNA_NET;

typedef struct backup_summary
{
    AP_UINT16      overlay_size;    /* size of returned entry   */
    unsigned char  reserv1[2];     /* reserved                  */
    unsigned char  server_name[128]; /* master or backup server name */
    unsigned char  reserv2[4];     /* reserved                  */
} BACKUP_SUMMARY;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_SNA_NET

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size
指定したデータ・バッファのサイズ。

num_entries
サーバー名の最大数 (そのサーバーにデータが戻されます)。ある範囲でなく、特定のエントリーに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options
Communications Server for Linux がデータを戻すためのリスト内の開始位置。
値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

server_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

server_name パラメーターで指定したエントリーの直後のエントリーから開始します。

アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。サーバー名は、このファイル内での順序と同じ順序でリストされます。その他の QUERY_* verb と同様、名前の長さやアルファベット順にはリストされません。

server_name

情報を求めるサーバーの名前、またはサーバーのリストの索引として使用する名前。サーバー名は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

サーバー名に . (ピリオド) 文字が含まれている場合、Communications Server for Linux はこれが完全修飾名であると想定します。ピリオドが含まれていない場合は、DNS ルックアップを実行してサーバー名を判別します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

domain_name

Communications Server for Linux LAN を含む TCP/IP ドメイン名。この名前は、マスター・サーバーをインストールするときに指定します。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

backup_summary.overlay_size

戻された *backup_summary* 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *backup_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造

の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

backup_summary.server_name

サーバー名。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_RECORD_NOT_FOUND

list_options パラメーターが、指定したサーバー名から始まるエントリーをリストするために、AP_LIST_INCLUSIVE または AP_LIST_FROM_NEXT に設定されましたが、*backup_name* パラメーターは、ファイル内のエントリーと一致しません。指定した名前が以前の QUERY_SNA_NET `verb` で戻される名前である場合は、以前の `verb` を使用した後に (別の管理プログラムまたは NOF アプリケーションによって) リストが更新されたことを示しています。アプリケーションで QUERY_SNA_NET を再発行し、完全なリストを取得する必要があります。

AP_INVALID_TARGET

NOF API 呼び出しのターゲット・ハンドルで構成ファイルまたはノードが指定されています。この `verb` は、**sna.net** ファイルに対して発行する必要があります。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_STATISTICS

QUERY_STATISTICS は、LS またはポートの使用状況についての統計を戻します。MPC リンク・タイプはリンク統計をサポートしていないため、MPC LS またはポートに対してはこの verb を発行しないでください。QLLC リンク・タイプはリンク統計をサポートしていないため、QLLC LS またはポートに対してはこの verb を発行しないでください。

戻される情報タイプは、DLC タイプによって異なります。

SDLC の場合、この verb は LS またはポートに関する統計情報 (送受信された特定フレームなどのイベント数) または操作情報 (現在使用されているパラメーターの詳細) を戻します。

トークンリングまたはイーサネットの場合、この verb は LS またはポートに関する統計情報を戻します。

Enterprise Extender の場合、この verb は LS に関する統計情報を戻します。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_statistics
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;               /* reserved                     */
    unsigned char  format;                /* reserved                     */
    AP_UINT16      primary_rc;            /* primary return code          */
    AP_UINT32      secondary_rc;          /* secondary return code        */
    unsigned char  name[8];               /* LS name or port name         */
    unsigned char  stats_type;            /* LS or port statistics?      */
    unsigned char  table_type;            /* statistics table requested   */
    unsigned char  reset_stats;           /* reset the statistics?        */
    unsigned char  dlc_type;              /* type of DLC                  */
    unsigned char  statistics[256];       /* current statistics           */
    unsigned char  reserva[20];           /* reserved                     */
} QUERY_STATISTICS;
```

SDLC に関する LS 統計情報:

```
typedef struct sd1_ls_stats_table
{
    V0_MUX_INFO    mux_info;              /* streams config info          */
    AP_UINT32      index;                 /* index of port that owns LS   */
    unsigned int    address;               /* poll address of secondary link station*/
    unsigned char  reserv[3];             /* reserved                     */
    AP_UINT32      blus_in;               /* frames received from adjacent link */
                                           /* station                      */
    AP_UINT32      blus_out;              /* frames sent to adjacent link station */
    AP_UINT32      octets_in;             /* bytes received from adjacent link */
                                           /* station                      */
    AP_UINT32      octets_out;            /* bytes sent to adjacent link station */
    AP_UINT32      polls_out;             /* polls sent to adjacent link station */
    AP_UINT32      poll_rsps_out;         /* polls responded to by adjacent link */
}
```

```

AP_UINT32    local_busies;    /* station */
/* number of times local link station has*/
/* entered busy state (RNR) */
AP_UINT32    remote_busies;  /* number of times remote link station */
/* has entered busy state (RNR) */
AP_UINT32    iframes_in;    /* I-frames rcvd from adjacent link */
/* station */
AP_UINT32    iframes_out;    /* I-frames sent to adjacent link station*/
AP_UINT32    retransmits_in; /* Total number of retransmitted */
/* I-frames received
AP_UINT32    retransmits_out; /* I-frames retransmitted since LS */
/* start-up
AP_UINT32    ioctets_in;    /* bytes in I-frames received */
AP_UINT32    ioctets_out;    /* bytes in I-frames sent */
AP_UINT32    uiframes_in;    /* reserved */
AP_UINT32    uiframes_out;   /* reserved */
AP_UINT32    xids_in;    /* XIDs rcvd from adjacent link station */
AP_UINT32    xids_out;    /* XIDs sent to adjacent link station */
AP_UINT32    tests_in;    /* TEST frames received */
AP_UINT32    tests_out;    /* TEST frames sent */
AP_UINT32    rejs_in;    /* REJ frames received */
AP_UINT32    rejs_out;    /* REJ frames sent */
AP_UINT32    frmrs_in;    /* FRMR frames received */
AP_UINT32    frmrs_out;    /* FRMR frames sent */
AP_UINT32    sims_in;    /* SIM frames received */
AP_UINT32    sims_out;    /* SIM frames sent */
AP_UINT32    rims_in;    /* RIM frames received */
AP_UINT32    rims_out;    /* RIM frames sent */
AP_UINT32    disc_in;    /* reserved */
AP_UINT32    disc_out;    /* reserved */
AP_UINT32    ua_in;    /* reserved */
AP_UINT32    ua_out;    /* reserved */
AP_UINT32    dm_in;    /* reserved */
AP_UINT32    dm_out;    /* reserved */
AP_UINT32    snrm_in;    /* SNRM frames received */
AP_UINT32    snrm_out;    /* SNRM frames sent */
} SDL_LS_STATS_TABLE;

```

SDLC に関する LS 操作情報:

```

typedef struct sdl_ls_oper_table
{
    V0_MUX_INFO    mux_info;    /* streams config info */
    AP_UINT32    index;    /* index of port that owns LS */
    unsigned char    address;    /* poll address of secondary link station */
    unsigned char    reserve;    /* reserved */
    AP_UINT16    role;    /* current role of link station */
    unsigned char    name[8];    /* reserved */
    AP_UINT16    state;    /* operational state of LS */
    AP_UINT16    maxdata;    /* current max PDU size for logical link */
    AP_UINT32    replyto;    /* current reply timeout */
    AP_UINT32    maxin;    /* current max unack'd frames LS can receive*/
    AP_UINT32    maxout;    /* current max unack'd frames LS can send */
    unsigned char    modulo;    /* sequence number modulus */
    unsigned char    reserv2[3];    /* reserved */
    AP_UINT32    retries_m;    /* number of retries in a retry sequence */
    AP_UINT32    retries_t;    /* interval between retry sequences */
    AP_UINT32    retries_n;    /* number of times to repeat retry sequence */
    AP_UINT32    rnrlimit;    /* how long adjacent LS can be in RNR state */
/* before it is considered inoperative */
    unsigned char    datmode;    /* communications mode with adjacent LS */
    unsigned char    last_fail_cause;    /* reserved */
    unsigned char    last_fail_ctrl_in[2];    /* control field of last frame rcvd*/
/* before last failure */
    unsigned char    last_fail_ctrl_out[2];    /* control field of last frame sent*/
/* before last failure */
    unsigned char    last_fail_frmr_info[5];    /* info field of FRMR frame if */

```

QUERY_STATISTICS

```

/* last failure was caused by */
/* invalid frame */
unsigned char sdoppad1; /* reserved */
AP_UINT32 last_fail_replyto_s; /* number of REPLYTO timeouts at */
/* time of last failure */
unsigned char g_poll; /* group poll address */
unsigned char sim_rim; /* are SIM / RIM supported? */
unsigned char xmit_rcv_cap; /* transmit / receive capability */
} SDL_LS_OPER_TABLE;
```

SDLC に関するポート統計情報:

```
typedef struct sdl_port_stats_table
{
    V0_MUX_INFO mux_info; /* streams config info */
    AP_UINT32 index; /* index of port */
    AP_UINT32 dwarf_frames; /* frames received too short to be valid */
    AP_UINT32 polls_out; /* polls sent to adjacent link stations */
    AP_UINT32 poll_rsps_out; /* polls responded to by adjacent link stns*/
    AP_UINT32 local_buies; /* number of times local link station */
    /* has entered busy state (RNR) */
    AP_UINT32 remote_buies; /* number of times remote link stations */
    /* have entered busy state (RNR) */
    AP_UINT32 iframes_in; /* I-frames rcvd from adjacent link */
    /* stations */
    AP_UINT32 iframes_out; /* I-frames sent to adjacent link stations */
    AP_UINT32 octets_in; /* bytes received from adjacent link */
    /* stations */
    AP_UINT32 octets_out; /* bytes sent to adjacent link stations */
    AP_UINT32 protocol_errs; /* link deactivations due to bad rcvd */
    /* frames */
    AP_UINT32 activity_to_s; /* link deactivations due to inactivity */
    AP_UINT32 rnrlimit_s; /* link deacts due to rem busy timer expiry*/
    AP_UINT32 retries_exps; /* link deacts due to end of retry sequence*/
    AP_UINT32 retransmits_in; /* retransmitted I-frames rcvd since */
    /* start-up */
    AP_UINT32 retransmits_out; /* I-frames retransmitted since start-up */
} SDL_PORT_STATS_TABLE;
```

SDLC に関するポート操作情報:

```
typedef struct sdl_port_oper_table
{
    V0_MUX_INFO mux_info; /* streams config info */
    AP_UINT32 index; /* index of port */
    unsigned char name[8]; /* reserved */
    unsigned char role; /* current role of link station(s) */
    /* using port */
    unsigned char type; /* line type - leased or switched */
    unsigned char topology; /* can port be point-to-point or */
    /* multipoint */
    unsigned char reserve; /* reserved */
    AP_UINT32 activto; /* how long switched line can be */
    /* inactive before port disconnects */
    AP_UINT32 pause; /* time between poll cycles */
    unsigned char slow_poll_method; /* slow poll method */
    unsigned char reserv2[3]; /* reserved */
    AP_UINT32 slow_poll_timer; /* slow poll timer */
    unsigned char last_fail_cause; /* reserved */
} SDL_PORT_OPER_TABLE;
```

トークンリングまたはイーサネットに関する LS 統計:

```
typedef struct llc2_ls_stats
{
    V0_MUX_INFO mux_info; /* streams config info */
    unsigned char local_mac[6]; /* MAC address of local port */
}
```

```

unsigned char    local_sap;          /* SAP number of local port */
unsigned char    reserve1;          /* reserved */
unsigned char    remote_mac[6];     /* MAC address of remote port */
unsigned char    remote_sap;        /* SAP number of remote port */
unsigned char    reserve2;          /* reserved */
AP_UINT16       rif_len;            /* length of RIF data for TR */
AP_UINT16       rif[8];            /* Routing Information Field
/* for TR */

unsigned char    ls_fsm;            /* LLC2 FSM state */
unsigned char    reserve3;          /* reserved */
AP_UINT16       mac_type;          /* LS MAC type */
AP_UINT16       max_btu_size;       /* maximum BTU size */
AP_UINT16       send_window;       /* send window */
AP_UINT16       receive_window;    /* receive window */
AP_UINT32       t1_expiry_count;    /* T1 expiry count */
AP_UINT32       t2_expiry_count;    /* T2 expiry count */
AP_UINT32       remote_busy;       /* remote busy state count */
AP_UINT32       local_busy;        /* local busy state count */
AP_UINT32       i_frames_sent;     /* count of I-frames sent */
AP_UINT32       i_bytes_sent;      /* count of bytes in I-frames sent*/
AP_UINT32       i_frames_rcvd;     /* count of I-frames received */
AP_UINT32       i_bytes_rcvd;      /* count of bytes in I-frames
/* received */

AP_UINT32       i_frames_rjctd;     /* count of I-frames rejected */
AP_UINT32       i_bytes_rjctd;     /* count of bytes in I-frames
/* rejected */

AP_UINT32       i_frames_rexmit;    /* count of I-frames retransmitted*/
AP_UINT32       i_bytes_rexmit;    /* count of bytes in I-frms
/* retransmitted */

AP_UINT32       rej_frames_sent;    /* count of REJ frames sent */
AP_UINT32       rej_frames_rcvd;    /* count of REJ frames received */
AP_UINT32       xid_frames_sent;   /* count of XID frames sent */
AP_UINT32       xid_frames_rcvd;   /* count of XID frames received */
AP_UINT16       ack_timeout;        /* acknowledgment timeout */
AP_UINT16       p_bit_timeout;     /* poll bit timeout */
AP_UINT16       t2_timeout;        /* T2 timeout */
AP_UINT16       rej_timeout;       /* REJ timeout */
AP_UINT16       busy_state_timeout; /* busy state timeout */
AP_UINT16       idle_timeout;      /* idle timeout */
AP_UINT16       max_retry;         /* max retry count */
} LLC2_LS_STATS;

```

トークンリングまたはイーサネットに対するポート統計:

```

typedef struct llc2_port_stats
{
    V0_MUX_INFO    mux_info;          /* streams config info */
    AP_UINT32      time_secs;         /* system time when port was
/* activated */

    AP_UINT16      time_ms;          /* system time when port was
/* activated */

    unsigned char  mac_addr[6];      /* MAC address of port */
    AP_UINT16      ack_timeout;       /* reserved */
    AP_UINT16      p_bit_timeout;     /* reserved */
    AP_UINT16      t2_timeout;       /* reserved */
    AP_UINT16      rej_timeout;      /* reserved */
    AP_UINT16      busy_state_timeout; /* reserved */
    AP_UINT16      idle_timeout;     /* reserved */
    AP_UINT16      max_retry;        /* reserved */
    AP_UINT16      max_btu_size;     /* max BTU size for port */
    AP_UINT16      ls_count;         /* count of LSs using port */
    AP_UINT16      reserve1;         /* reserved */
    AP_UINT32      ui_frames_sent;    /* count of UI frames sent */
    AP_UINT32      ui_frames_rcvd;   /* count of UI frames received */
    LLC2_DEV_STATS device_stats;     /* device statistics */
} LLC2_PORT_STATS;

```

QUERY_STATISTICS

```
typedef struct llc2_dev_stats
{
    unsigned char    adapter_number;    /* reserved */
    unsigned char    res1;              /* reserved */
    unsigned char    line_error;        /* line error count */
    unsigned char    internal_error;    /* internal error count */
    unsigned char    burst_error;       /* burst error count */
    unsigned char    ari_fci_error;     /* ARI/FCI error count */
    unsigned char    end_delim;         /* end delimiter */
    unsigned char    res2;              /* reserved */
    unsigned char    lost_frame;        /* lost frame count */
    unsigned char    rcv_cngstn;        /* Receive congestion count */
    unsigned char    frm_cpy_err;       /* Frame Copied error count */
    unsigned char    freq_err;          /* frequency error count */
    unsigned char    token_err;         /* token error count */
    unsigned char    crc_err;           /* CRC error count */
    unsigned char    res3;              /* reserved */
    unsigned char    xmit_err;          /* transmit error count */
    unsigned char    res4;              /* reserved */
    unsigned char    collision_err;     /* collision error count */
    unsigned char    res5[7];           /* reserved */
} LLC2_DEV_STATS;
```

Enterprise Extender に関する LS 統計:

```
typedef struct udp_ls_stats_table
{
    V0_MUX_INFO     mux_info;           /* streams config info */
    AP_UINT32       udp_low_out;        /* Count of UDP datagrams sent
                                         /* containing low priority data */
    AP_UINT32       udp_med_out;        /* Count of UDP datagrams sent
                                         /* containing medium priority data */
    AP_UINT32       udp_high_out;       /* Count of UDP datagrams sent
                                         /* containing high priority data */
    AP_UINT32       udp_network_out;    /* Count of UDP datagrams sent
                                         /* containing network priority data */
    AP_UINT32       udp_llc_out;        /* Count of UDP datagrams sent
                                         /* containing LLC commands */
} UDP_LS_STATS_TABLE;

typedef struct v0_mux_info
{
    AP_UINT16       dlc_type;           /* DLC implementation type */
    unsigned char   need_vrfy_fixup;    /* reserved */
    unsigned char   num_mux_ids;        /* reserved */
    AP_UINT32       card_type;          /* type of adapter card */
    AP_UINT32       adapter_number;     /* DLC adapter number */
    AP_UINT32       oem_data_length;    /* reserved */
    AP_UINT32       mux_ids[5];         /* reserved */
} V0_MUX_INFO;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_STATISTICS

name (*stats_type* パラメーターで指定されている) 統計が必要な LS またはポートの名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。Communications Server for Linux はこの名前を使用して、正確なリンク・ステーションまたはポートへの応答を関連づけます。

stats_type

統計を求めるリソースのタイプ。

トークンリングまたはイーサネットの場合、値は次のいずれかです。

AP_LS LS 統計を戻します。

AP_PORT
ポート統計を戻します。

Enterprise Extender の場合は、この値を AP_LS に設定する必要があります。

table_type

必要な統計情報のタイプ。

SDLC の場合にしようできる値:

AP_STATS_TBL
統計情報

AP_OPER_TBL
操作情報

トークンリングまたはイーサネットの場合は AP_STATS_TBL に設定する必要があります。

Enterprise Extender の場合は AP_STATS_TBL に設定する必要があります。

reset_stats

verb の正常終了時に統計をリセットするかどうかを指定します。このパラメーターは *table_type* が AP_STATS_TBL に設定されている場合のみ適用されます。それ以外の場合は無視されます。値は次のいずれかです。

AP_YES 統計をリセットします。あとに続く QUERY_STATISTICS verb にはこの verb の発行後に収集されたデータのみが入ります。

AP_NO 統計をリセットしません。この verb のデータは、あとに続く QUERY_STATISTICS verb で戻されるデータに組み込まれます。

dlc_type

DLC のタイプ。値は次のいずれかです。

AP_SDLC
同期データ・リンク制御

AP_TR トークンリング

AP_ETHERNET
イーサネット

AP_X25 X.25 パケット交換

AP_IP Enterprise Extender (HPR/IP)

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

dlc_type
統計情報が戻される DLC のタイプ。値は次のいずれかです。

QUERY_STATISTICS

AP_SDLC

SDLC

AP_X25 QLLC

AP_TR トークンリング

AP_ETHERNET

イーサネット

AP_IP Enterprise Extender (HPR/IP)

statistics

リンク・ステーションまたはポートに対する現在の統計。このストリングは、DLC タイプについての適切な構造体で置換されます。構造体のパラメーターについては、以下で説明します。

mux_info.dlc_type, mux_info.card_type, mux_info.adapter_number

DLC に関するストリーム設定情報。これらのパラメーターについては詳しくは、100 ページの『DEFINE_DLC』を参照してください。

SDLC に関する LS 統計情報:

sdl_ls_stats_table.index

この LS を所有するポートを識別するために Communications Server for Linux によって内部使用される索引値。

sdl_ls_stats_table.address

2 次リンク・ステーションのポーリング・アドレス。

sdl_ls_stats_table.blus_in

隣接リンク・ステーションから受信された基本リンク単位 (フレーム) の総数。

sdl_ls_stats_table.blus_out

隣接リンク・ステーションへ伝送された基本リンク単位 (フレーム) の総数。

sdl_ls_stats_table.octets_in

隣接リンク・ステーションから受信されたバイトの総数 (FCS は含まない)。

sdl_ls_stats_table.octets_out

隣接リンク・ステーションへ伝送されたバイト総数 (FCS は含まない)。

sdl_ls_stats_table.polls_out

隣接リンク・ステーションへ送信されたポーリングの総数。

sdl_ls_stats_table.poll_rsps_out

隣接リンク・ステーションが応答したポーリングの総数。

sdl_ls_stats_table.local_busies

ローカル・リンク・ステーションがビジー状態 (RNR) になった合計回数。

sdl_ls_stats_table.remote_busies

リモート・リンク・ステーションがビジー状態 (RNR) になった合計回数。

sdl_ls_stats_table.iframes_in

隣接リンク・ステーションから受信された I フレームの総数 (再試行数および規則違反フレームを含む)。

sdl_ls_stats_table.iframes_out

隣接リンク・ステーションへ伝送された I フレームの総数 (再試行数および規則違反フレームを含む)。

sdl_ls_stats_table.retransmits_in

予約された I フレームの再伝送総数。

sdl_ls_stats_table.retransmits_out

隣接リンク・ステーションへの I フレームの再伝送総数。

sdl_ls_stats_table.ioctets_in

隣接リンク・ステーションから受信された I フレームのバイト総数。

sdl_ls_stats_table.ioctets_out

隣接リンク・ステーションへ伝送された I フレームのバイト総数。

sdl_ls_stats_table.xids_in

隣接リンク・ステーションから受信された XID フレームの総数。

sdl_ls_stats_table.xids_out

隣接リンク・ステーションへ伝送された XID フレームの総数。

sdl_ls_stats_table.tests_in

隣接リンク・ステーションから受信された TEST フレーム、コマンド、または応答の総数。

sdl_ls_stats_table.tests_out

隣接リンク・ステーションへ伝送された TEST フレーム、コマンド、または応答の総数。

sdl_ls_stats_table.rejs_in

隣接リンク・ステーションから受信された REJ フレームの総数。

sdl_ls_stats_table.rejs_out

隣接リンク・ステーションへ伝送された REJ フレームの総数。

sdl_ls_stats_table.fmrs_in

隣接リンク・ステーションから受信された「フレーム・リジェクト」フレームの総数。

sdl_ls_stats_table.fmrs_out

隣接リンク・ステーションへ伝送された「フレーム・リジェクト」フレームの総数。

sdl_ls_stats_table.sims_in

隣接リンク・ステーションから受信された「初期化モードの設定」フレームの総数。

sdl_ls_stats_table.sims_out

隣接リンク・ステーションへ伝送された「初期化モードの設定」フレームの総数。

QUERY_STATISTICS

sdl_ls_stats_table.rims_in

隣接リンク・ステーションから受信された「初期化モードの要求」フレームの総数。

sdl_ls_stats_table.rims_out

隣接リンク・ステーションへ伝送された「初期化モードの要求」フレームの総数。

sdl_ls_stats_table.snrm_in

受信された SNRM フレームの総数。

sdl_ls_stats_table.snrm_out

送信された SNRM フレームの総数。

SDLC に関する LS 操作情報:

sdl_ls_oper_table.index

この LS を所有するポートを識別するために Communications Server for Linux によって内部使用される索引値。

sdl_ls_stats_table.address

2 次リンク・ステーションのポーリング・アドレス。

sdl_ls_stats_table.role

LS のリンク・ロール。値は次のいずれかです。

SDL_MIB_PRIMARY

1 次

SDL_MIB_SECONDARY

2 次

SDL_MIB_NEGOTIABLE

折衝可能

sdl_ls_stats_table.state

LS ソフトウェアの処理状態を示す内部値 (サポート担当者が使用)。

sdl_ls_stats_table.maxdata

論理リンクに許可された現行の最大 PDU サイズ (このサイズには TH および RH が含まれる)。交換回線の場合、この値は XID 交換中に折衝される可能性があります。

sdl_ls_stats_table.replyto

現行の応答タイムアウト (100 分の 1 秒)。このパラメーターは LS ロールが 1 次の場合のみ適用されます。LS ロールが 2 次の場合、この値は未定義となります。

sdl_ls_stats_table.maxin

肯定応答の送信が必要になる前に、LS が受信することのできるフレーム肯定応答の最大数。

sdl_ls_stats_table.maxout

肯定応答の待機が必要になる前に、LS が送信することのできるフレームの最大数。

sdl_ls_stats_table.modulo

LS のシーケンス番号モジュラス。値は次のいずれかです。

SDL_MIB_EIGHT

8

SDL_MIB_ONETWENTYEIGHT

128

sdl_ls_stats_table.retries_m

再試行シーケンス (LS がフレームに対する肯定応答を受信しなかったために、再送するフレームのシーケンス) 内のフレームの最大数。

sdl_ls_stats_table.retries_t

再試行シーケンスの再伝送間のタイムアウト。

sdl_ls_stats_table.retries_n

LS が再試行シーケンスを再送しようとする回数。

sdl_ls_stats_table.rnrlimit

ローカル LS が作動不能と見なされるまでに、隣接 LS が RNR 状態に留まることのできる最大時間長。

sdl_ls_stats_table.datmode

隣接 LS との通信モード。値は次のいずれかです。

SDL_MIB_HALF

両方向交互 (半二重)

SDL_MIB_FULL

両方向同時 (全二重)

sdl_ls_stats_table.last_fail_ctrl_in

最後の障害が起こる前に受信された最終フレームからの制御フィールド。LS に障害が起こらなかった場合、このフィールドはゼロに設定されます。

sdl_ls_stats_table.last_fail_ctrl_out

最後の障害が起こる前に送信された最終フレームからの制御フィールド。LS に障害が起こらなかった場合、このフィールドはゼロに設定されます。

sdl_ls_stats_table.last_fail_frmr_info

最後の障害が無効なフレームに原因している場合、このパラメーターには FRMR フレームからの情報フィールドが含まれています。LS に障害が起こらなかった場合、または障害が無効なフレームに原因していなかった場合、このフィールドはゼロに設定されます。

sdl_ls_stats_table.last_fail_replyto_s

最後の障害の前に応答タイムアウトになった回数。LS に障害が起こらなかった場合、このフィールドはゼロに設定されます。

sdl_ls_stats_table.g_poll

LS のグループ・ポーリング・アドレス。LS がグループ内に存在していない場合、このフィールドはゼロに設定されます。

sdl_ls_stats_table.sim_rim

LS が SIM および RIM 制御フレームの伝送をサポートするかどうかを指定します。値は次のいずれかです。

SDL_MIB_YES

LS は SIM および RIM をサポートします。

QUERY_STATISTICS

SDL_MIB_NOLS

LS は SIM および RIM をサポートしません。

sdl_ls_stats_table.xmit_rcv_cap

LS の伝送/受信機能を指定します。値は次のいずれかです。

SDL_MIB_HALF

半二重

SDL_MIB_FULL

全二重

SDLC に関するポート統計情報:

sdl_port_stats_table.index

ポートを識別するために Communications Server for Linux によって内部使用される索引値。

sdl_port_stats_table.dwarf_frames

ポートが受信した、短すぎて有効にならなかったフレーム数。

sdl_port_stats_table.polls_out

隣接リンク・ステーションへ送信されたポーリングの総数。

sdl_port_stats_table.poll_rsps_out

隣接リンク・ステーションに応答したポーリングの総数。

sdl_port_stats_table.local_busies

ローカル・リンク・ステーションがビジー状態 (RNR) になった合計回数。

sdl_port_stats_table.remote_busies

リモート・リンク・ステーションがビジー状態 (RNR) になった合計回数。

sdl_port_stats_table.iframes_in

隣接リンク・ステーションから受信された I フレームの総数 (再試行数および規則違反フレームを含む)。

sdl_port_stats_table.iframes_out

隣接リンク・ステーションへ伝送された I フレームの総数 (再試行数および規則違反フレームを含む)。

sdl_port_stats_table.octets_in

隣接リンク・ステーションから受信されたバイトの総数 (FCS は含まない)。

sdl_port_stats_table.octets_out

隣接リンク・ステーションへ伝送されたバイト総数 (FCS は含まない)。

sdl_port_stats_table.protocol_errs

隣接リンク・ステーションから受信されたフレームにプロトコル・エラーがあるために、Communications Server for Linux がこのポートを使用して LS を非活動状態にした回数。

sdl_port_stats_table.activity_to_s

リンク上で活動が何もなかったため、Communications Server for Linux がこのポートを使用している LS を非活動状態にした回数。

sdl_port_stats_table.rnrlimit_s

リモート・ビジー・タイマーが時間切れしたために、Communications Server for Linux がこのポートを使用している LS を非活動状態にした回数。

sdl_port_stats_table.retries_exps

再試行シーケンスを使い切ったために、Communications Server for Linux がこのポートを使用している LS を非活動状態にした回数。

sdl_port_stats_table.retransmits_in

隣接リンク・ステーションから受信された再送 I フレームの総数。

sdl_port_stats_table.retransmits_out

隣接リンク・ステーションへ再伝送された I フレームの総数。

SDLC に関するポート操作情報:

sdl_port_oper_table.index

ポートを識別するために Communications Server for Linux によって内部使用される索引値。

sdl_port_oper_table.role

ポートのリンク・ロール。値は次のいずれかです。

SDL_MIB_PRIMARY

1 次

SDL_MIB_SECONDARY

2 次

SDL_MIB_NEGOTIABLE

折衝可能

sdl_port_oper_table.type

ポートの作動状態が専用回線または交換回線に接続されているものとして指定します。値は次のいずれかです。

SDL_MIB_LEASED**SDL_MIB_SWITCHED***sdl_port_oper_table.topology*

ポートがマルチポイント・トポロジーで作動できるかどうかを指定します。値は次のいずれかです。

SDL_MIB_POINT_TO_POINT

ポートは Point-to-Point としてのみ作動できます。

SDL_MIB_MULTIPPOINT

ポートはマルチポイントとして作動できます。

sdl_port_oper_table.activto

ポートの交換回線を切断する前に非活動状態 (転送中の I フレームがない状態) のままにすることができる時間長 (100 分の 1 秒)。ゼロの値はタイムアウトを発生させないことを指示します。すなわち、回線は非活動状態にならなければ、接続されたままの状態が継続します。このパラメーターは交換回線リンクにのみ適用されます。専用回線リンクの場合、この値は未定義となります。

QUERY_STATISTICS

sdl_port_oper_table.pause

1 次ステーションが 2 次ステーションを連続してポーリングするサイクルの間隔をあけるために待機する時間長。このパラメーターは LS ロールが 1 次の場合のみ適用されます。LS ロールが 2 次の場合、この値は未定義となります。

sdl_port_oper_table.slow_poll_method

失敗した 2 次リンク・ステーションの周期的なポーリングに使用された方式。これは SDL_MIB_POLLPAUSE に設定されます。

sdl_port_oper_table.slow_poll_timer

失敗した 2 次リンク・ステーションに対するポーリング間のタイムアウト。このパラメーターはポートが 1 次で、かつ、マルチポイント・トポロジーで作動している場合のみ適用されます。そのほかの場合、この値は未定義となります。

トークンリングまたはイーサネットに関する LS 統計:

llc2_ls_stats.local_mac

ローカル・リンク・ステーションの MAC アドレス。

llc2_ls_stats.local_sap

ローカル・リンク・ステーションの SAP アドレス。

llc2_ls_stats.remote_mac

リモート・リンク・ステーションの MAC アドレス。

llc2_ls_stats.remote_sap

リモート・リンク・ステーションの SAP アドレス。

llc2_ls_stats.rif_len

経路指定情報フィールド・データの長さ。このパラメーターは、トークンリングに対してのみ使用します。これは、その他の DLC タイプ用に予約されています。

llc2_ls_stats.rif

経路指定情報フィールド・データ。このパラメーターは、トークンリングに対してのみ使用します。これは、その他の DLC タイプ用に予約されています。

このデータは、16 ビットの番号の配列としてローカル・フォーマットで戻されます。各番号の最初の 12 ビットはリング番号を指定し、末尾の 4 ビットはブリッジ番号を指定します。

llc2_ls_stats.ls_fsm

LS ソフトウェアの処理状態を示す内部値 (サポート担当者が使用)。

llc2_ls_stats.mac_type

LS を活動化するときに決定されるネットワーク・タイプ。値は次のいずれかです。

LLC_DIX

DIX

LLC2_802_3

802.3

LLC2_802_3_DIX

まだ決定されていません (802.3 または DIX)。この値は、隣接ステーションがこれらのフォーマットのいずれかのフレーム最初に応答する際に、上記のいずれかの値に変更されます。

LLC2_TOKEN_RING

トークンリング

llc2_ls_stats.max_btu_size

LS を活動化するとき決定される BTU の最大サイズ。

llc2_ls_stats.send_window

応答を待つ前にローカル・ステーションが隣接ステーションに送信できる I フレームの数。

llc2_ls_stats.receive_window

応答を待つ前に隣接ステーションがローカル・ステーションに送信できる I フレームの数。

llc2_ls_stats.t1_expiry_count

t1_timeout (受信確認タイムアウト) 期間内に隣接ステーションが応答できなかった回数。

llc2_ls_stats.t2_expiry_count

要求された応答ビットを伝送できるフレームがキューに入れられる前に *t2_timeout* 期間が期限切れになった回数。

llc2_ls_stats.remote_busy

隣接ステーションからの RNR フレームのためにローカル・ステーションがリモート・ビジー状態になった回数。

llc2_ls_stats.local_busy

ローカル・ステーションがローカル・ビジー状態になったときに RNR フレームを隣接ステーションに送信した回数。

llc2_ls_stats.i_frames_sent

送信された I フレームの数。

llc2_ls_stats.i_bytes_sent

送信された I フレーム内のデータ・バイトの数。

llc2_ls_stats.i_frames_rcvd

受信された I フレームの数。

llc2_ls_stats.i_bytes_rcvd

受信された I フレーム内のデータ・バイトの数。

llc2_ls_stats.i_frames_rjctd

リジェクトされた I フレームの数。

llc2_ls_stats.i_bytes_rjctd

リジェクトするされた I フレーム内のデータ・バイトの数。

llc2_ls_stats.i_frames_rexmit

再送された I フレームの数。

llc2_ls_stats.i_bytes_rexmit

再送された I フレーム内のデータ・バイトの数。

QUERY_STATISTICS

llc2_ls_stats.rej_frames_sent

1 つ以上の I フレームの再送を要求するために送信された REJ フレームの数。

llc2_ls_stats.rej_frames_rcvd

1 つ以上の I フレームの再送を要求する、受信された REJ フレームの数。

llc2_ls_stats.xid_frames_sent

送信された XID フレームの数。

llc2_ls_stats.xid_frames_rcvd

受信された XID フレームの数。

llc2_ls_stats.ack_timeout

通知タイムアウト: その時間内に、隣接リンク・ステーションに送信された I フレームに対する応答が受信される必要があるミリ秒単位の時間。

llc2_ls_stats.p_bit_timeout

ポーリング・ビット・タイムアウト: その時間内に、隣接リンク・ステーションにポーリング・ビット・セットとともに送信されたフレームに対する応答が受信される必要があるミリ秒単位の時間。

llc2_ls_stats.t2_timeout

ローカル・ステーションが、受信した I フレームに対する応答を送信するまでに待機できる最大時間 (ミリ秒)。タイムアウトを長くすると、ローカル・ステーションが複数の I フレームに単一の RR で応答できるため、通知トラフィックを削減できます。

llc2_ls_stats.rej_timeout

リジェクト・タイムアウト: その時間内に、隣接リンク・ステーションに送信された REJ フレームに対する応答が受信される必要がある秒単位の時間。

llc2_ls_stats.busy_state_timeout

ローカル・ステーションが、ビジー状態 (RNR) がクリアされたことを示す隣接リンク・ステーションからの通知を待機する時間 (秒)。

llc2_ls_stats.idle_timeout

アイドル・タイムアウト: 完全に非アクティブ・ラインを検出するために使用されるタイマー。この時間内に何も受信しなかった場合、その回線はアイドル状態とみなされます。このタイマーは、秒単位で指定します。

llc2_ls_stats.max_retry

応答やビジー状態のクリアを待機しているときのローカル・ステーションの再試行の最大回数。

トークンリングまたはイーサネットに対するポート統計:

llc2_port_stats.time_secs, time_ms

SNA ソフトウェアが開始されてから LLC2 コンポーネントがポート活動化要求を受信するまでの時間 (秒またはミリ秒)。

llc2_port_stats.mac_addr

ポートを活動化するとき決定されるポートの MAC アドレス。

llc2_port_stats.max_btu_size

ポートを活動化するとき決定される BTU の最大サイズ。

llc2_port_stats.ls_count

現在ポートを使用しているリンク・ステーションの数。この数には、XID が送信されているが SABME がまだ送信されていないステーションの数も含まれます。

llc2_port_stats.ui_frames_sent

このポート上で発行されたタイプ I フレーム (UI、TEST、および XID) の数。

llc2_port_stats.ui_frames_rcvd

このポート上で受信されたタイプ I フレーム (UI、TEST、および XID) の数。

device_stats.line_error

回線エラーの総数。

device_stats.internal_error

内部エラーの総数。

device_stats.burst_error

バースト・エラーの総数。

device_stats.ari_fci_error

アドレス認識エラー/コピー・フレームのビット・エラーの総数。

device_stats.end_delim

フレーム区切り文字エラーの総数。

device_stats.lost_frame

脱落フレーム・エラーの総数。

device_stats.rcv_cngstn

受信側輻輳エラーの総数。

device_stats.frm_cpy_err

コピー・フレーム・エラーの総数。

device_stats.freq_err

繰り返しエラーの総数。

device_stats.token_err

トークン・エラーの総数。

device_stats.crc_err

巡回冗長検査エラーの総数。

device_stats.xmit_err

伝送エラーの総数。

device_stats.collision_err

衝突エラーの総数。

Enterprise Extender に関する LS 統計:

udp_ls_stats_table.udp_low_out

低優先順位 APPN データを含む、送信された UDP データグラムの数。

udp_ls_stats_table.udp_med_out

中優先順位 APPN データを含む、送信された UDP データグラムの数。

QUERY_STATISTICS

udp_ls_stats_table.udp_high_out

高優先順位 APPN データを含む、送信された UDP データグラムの数。

udp_ls_stats_table.udp_network_out

ネットワーク優先順位 APPN データを含む、送信された UDP データグラムの数。

udp_ls_stats_table.udp_llc_out

LLC コマンドを含む、送信された UDP データグラムの数。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_PARAMETER_CHECK`

secondary_rc

値は次のいずれかです。

AP_INVALID_LINK_NAME

指定した名前パラメーターは、有効な LS 名ではありませんでした。

AP_INVALID_PORT_NAME

指定した名前パラメーターは、有効なポート名ではありませんでした。

AP_INVALID_STATS_TYPE

stats_type パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_TABLE_TYPE

table_type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_STATE_CHECK`

secondary_rc

値は次のいずれかです。

AP_LINK_DEACTIVATED

指定したリンクは現在活動状態ではありません。

AP_PORT_DEACTIVATED

指定したポートは現在活動状態ではありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_STATE_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

DLC タイプが戻り統計情報をサポートしていないために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
AP_FUNCTION_NOT_SUPPORTED
```

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TN3270_ACCESS_DEF

QUERY_TN3270_ACCESS_DEF は、TN3270 サーバーを使用して 3270 エミュレーションを行うためのホストにアクセスするために、Communications Server for Linux の TN サーバー機能を使用することができる、別のコンピューター上の TN3270 ユーザーに関する情報を戻します。(TN リダイレクターを使用するホストにアクセスしているユーザーに関する情報を戻すには、QUERY_TN_REDIRECT_DEF を使用します。)

この verb は、使用するオプションに応じて、単一のユーザーまたは複数のユーザーに関する要約情報または詳細情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_tn3270_access_def
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;        /* secondary return code */
    unsigned char  *buf_ptr;             /* pointer to buffer */
    AP_UINT32      buf_size;             /* buffer size */
    AP_UINT32      total_buf_size;       /* total buffer size required */
    AP_UINT16      num_entries;          /* number of entries */
    AP_UINT16      total_num_entries;    /* total number of entries */
    unsigned char  list_options;         /* listing options */
    unsigned char  reserv3;              /* reserved */
    AP_UINT16      default_record;       /* start with DEFAULT record? */
    unsigned char  client_address[256];  /* address of TN3270 user */
    AP_UINT16      port_number;          /* TCP/IP port to access server */
    AP_UINT32      num_init_sessions;    /* number of sessions for first */
                                           /* user when starting in middle */
    AP_UINT32      num_last_sessions;    /* number of sessions on last */
                                           /* detail overlay if last user */
                                           /* is incomplete */
    unsigned char  last_user_incomplete; /* set to AP_YES if session */
                                           /* data for last user incomplete*/
    unsigned char  reserv4[11];          /* Reserved */
} QUERY_TN3270_ACCESS_DEF;

typedef struct tn3270_access_summary
{
    AP_UINT16      overlay_size;         /* overlay size */
    AP_UINT16      default_record;       /* is this the DEFAULT record? */
}
```

QUERY_TN3270_ACCESS_DEF

```
    unsigned char    client_address[256]; /* address of TN3270 user */
    AP_UINT16        address_format;     /* Format of client address */
    unsigned char    reserv3[6];         /* Reserved */
} TN3270_ACCESS_SUMMARY;
typedef struct tn3270_access_detail
{
    AP_UINT16        overlay_size;       /* overlay size */
    AP_UINT16        sub_overlay_offset; /* offset to first sess struct*/
    AP_UINT16        default_record;     /* is this the DEFAULT record?*/
    unsigned char    client_address[256]; /* address of TN3270 user */
    AP_UINT32        num_filtered_sessions; /* num sess returned for user */
    unsigned char    reserv3[4];         /* Reserved */
    TN3270_ACCESS_DEF_DATA def_data;     /* user definition */
} TN3270_ACCESS_DETAIL;
typedef struct tn3270_access_def_data
{
    unsigned char    description[32];    /* Description - null terminated */
    unsigned char    reserv1[16];        /* reserved */
    AP_UINT16        address_format;     /* Format of client address */
    AP_UINT32        num_sessions;       /* Number of sessions being added */
    unsigned char    reserv3[64];        /* reserved */
} TN3270_ACCESS_DEF_DATA;
```

セッションごとに、*num_sessions* パラメーターで指定した数まで、*def_data* 構造体の終わりに次の構造体を入れることができます。

```
typedef struct tn3270_session_def_data
{
    AP_UINT16        sub_overlay_size;   /* reserved */
    unsigned char    description[32];    /* Session description */
    unsigned char    tn3270_support;     /* Level of TN3270 support */
    unsigned char    allow_specific_lu;  /* Allow access to specific LUs */
    unsigned char    printer_lu_name[8]; /* Generic printer LU/pool
                                           /* accessed */
    unsigned char    reserv1[6];         /* reserved */
    AP_UINT16        port_number;        /* TCP/IP port used to access
                                           /* server */
    unsigned char    lu_name[8];         /* Generic display LU/pool
                                           /* accessed */
    unsigned char    session_type;       /* Unused in current version */
    unsigned char    model_override;     /* Unused in current version */
    unsigned char    ssl_enabled;        /* Is this an SSL session? */
    unsigned char    security_level;     /* SSL encryption strength */
    unsigned char    cert_key_label[80]; /* Certificate key label */
    unsigned char    listen_local_address[46];
                                           /* Local addr client connects to */
    unsigned char    allow_ssl_timeout_to_nonssl;
                                           /* Allow non-SSL clients on SSL? */
    unsigned char    reserv3[17];
    AP_UINT32        reserv4;           /* reserved */
} TN3270_SESSION_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TN3270_ACCESS_DEF

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファーへのポインター。

buf_size

指定したデータ・バッファーのサイズ。

num_entries

ユーザーの最大数 (そのユーザーのデータが戻されます)。ユーザー・セッションに関する詳細情報を戻す場合は、この数に部分エントリーが含まれます (部分エントリーにクライアント・アドレスを指定できるため、戻りデータにユーザー定義またはユーザーの最初のセッションは含まれません)。

ある範囲でなく、特定のユーザーに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のユーザーの最初のセッションから開始します。

AP_LIST_INCLUSIVE

指定したクライアント・アドレスおよびポート番号で指定したエントリーから開始するか、ポート番号を指定していない場合は指定したクライアント・アドレスの最初のセッションから開始します。

AP_LIST_FROM_NEXT

ポート番号を指定した場合は、指定したポート番号のあるセッションの直後のセッションから開始します。ポート番号を指定していない場合は、指定したクライアント・アドレスの最初のセッションから開始します。

リストは、最初にクライアント・アドレス、次にユーザーごとのポート番号の順番で並べられます。リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法については、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

default_record

要求したエントリー (またはリストの索引として使用されるエントリー) がデフォルト・レコードであるかどうかを指定します。

TCP/IP アドレスで明示的に識別されない TN3270 ユーザーが使用するデフォルト・レコードを照会するには、AP_YES を指定します。その場合、*client_address* パラメーターは予約済みです。

通常の TN3270 ユーザー・レコードを照会するには、AP_NO を指定します。

client_address

情報を必要とする TN3270 ユーザーの TCP/IP アドレス、またはユーザーのリストへの索引として使用する名前。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。アドレスはヌル終了 ASCII スtring であり、指定できるのは以下のいずれかです。

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

port_number

指定したユーザーの特定のセッションで始まる情報を戻すには、このパラメーターをそのセッションで定義された TCP/IP ポート番号に設定します。指定したユーザーの最初のセッションで始まる情報を戻すには、このパラメーターのビットをゼロに設定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

バッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。この値は *buf_size* より大きい場合があります。

total_num_entries

戻されている可能性があるエントリーの合計数。この値は *num_entries* より大きい場合があります。

num_entries

実際に戻されたエントリーの数。最後のエントリーが不完全な場合があり、それは *last_user_incomplete* パラメーターで示されます。

num_init_sessions

リスト内の最初のユーザーに関する情報がユーザーの最初のセッションから始まらないように *port_number* パラメーターをゼロ以外の値に設定した場合、このパラメーターは、戻されたデータに入っているこのユーザーのセッション構造体の数を示します。それ以外の場合は、このパラメーターは使用されません。

num_last_sessions

last_user_incomplete パラメーターで最後のユーザーのデータが不完全であることを示している場合、このパラメーターは、戻されたデータに入っているこのユーザーのセッション構造体の数を示します。それ以外の場合は、このパラメーターは使用されません。

last_user_incomplete

最後のユーザーに関する情報のデータが不完全であることを示します。値は次のいずれかです。

AP_YES 最後のユーザーのデータは大きすぎて、データ・バッファに入りませんでした。少なくとも 1 つのセッション構造体が含まれていますが、データ・バッファに入っていないセッション構造体があります。*num_last_sessions* パラメーターは、セッション構造体が戻された数を示しています。アプリケーションはさらにこの *verb* を発行して残りのデータを取得することができます。

AP_NO 最後のユーザーのデータは完全です。

データ・バッファ内の各エントリーは、次のパラメーターで構成されています。

tn3270_access_summary.overlay_size

戻された *tn3270_access_summary* 構造体のサイズ。すなわち、データ・バッファ内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *tn3270_access_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の *sizeof()* 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

tn3270_access_summary.default_record

このエントリーがデフォルト・レコードであるかどうかを指定します。値は次のいずれかです。

AP_YES これはデフォルト・レコードです。*client_address* パラメーターは予約済みです。

AP_NO これは通常の TN3270 ユーザー・レコードです。

tn3270_access_summary.client_address

TN3270 ユーザーの TCP/IP アドレス。以下のいずれかの値を指定できます。*address_format* パラメーターには、これが IP アドレスか、または名前のいずれかを指示します。

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

tn3270_access_summary.address_format

client_address パラメーターのフォーマットを指定します。値は次のいずれかです。

QUERY_TN3270_ACCESS_DEF

AP_ADDRESS_IP

IP アドレス (IPv4 または IPv6 のいずれか)

AP_ADDRESS_FQN

別名または完全修飾名

tn3270_access_detail.overlay_size

戻された *tn3270_access_detail* 構造体のサイズ。すなわち、データ・バッファにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *tn3270_access_detail* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

tn3270_access_detail.sub_overlay_offset

データ・バッファ内におけるこの TN3270 アクセス・レコードの最初のセッション・データ構造体の先頭までのオフセット。

tn3270_access_detail.default_record

このエントリーがデフォルト・レコードであるかどうかを指定します。値は次のいずれかです。

AP_YES これはデフォルト・レコードです。 *client_address* パラメーターは予約済みです。

AP_NO これは通常の TN3270 ユーザー・レコードです。

tn3270_access_detail.client_address

TN3270 ユーザーの TCP/IP アドレス。これはヌル終了 ASCII スtring であり、指定できるのは以下のいずれかです。

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

tn3270_access_detail.num_filtered_sessions

このユーザーに戻されるセッションの数。

tn3270_access_detail.def_data

構成情報で定義されているユーザーの詳細。このあとにユーザーのセッションを定義するセッション構造体がいくつか続きます。この情報のフォーマットは `DEFINE_TN3270_ACCESS verb` の場合と同じですが、次の点が異なります。

- *def_data* 構造体の *num_sessions* パラメーターは、ユーザーに定義されているセッションの合計数を定義します。

- *port_number* パラメーターをゼロ以外の値に設定した場合は、最初のユーザーのデータには残りのセッション構造体 (要求したエントリーから始まる) のみが入り、*def_data* 構造体は入りません。
- *last_user_incomplete* パラメーターを AP_YES に設定した場合は、最後のユーザーに戻されるセッション構造体の合計数は *num_last_sessions* パラメーターで指定した数になります。したがって、*num_sessions* より少なくなります。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_CLIENT_ADDRESS

list_options パラメーターを AP_LIST_INCLUSIVE に設定しましたが、*client_address* パラメーターで指定した値は、定義されているどの TN3270 ユーザーのアドレスにも一致しませんでした。

AP_INVALID_PORT_NUMBER

list_options パラメーターを AP_LIST_INCLUSIVE に設定しましたが、*port_number* パラメーターで指定した値は、指定した TN3270 ユーザーに定義されているポート番号に一致しませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TN3270_ASSOCIATION

QUERY_TN3270_ASSOCIATION は、ディスプレイ LU とプリンター LU 間の関連に関する情報を戻します。この関連は、ディスプレイ LU 名で照会し、ディスプレイ LU 名の順に戻されます。

この *verb* は使用するオプションに応じて、特定の関連または複数の関連に関する情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_tn3270_association
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  *buf_ptr;       /* pointer to buffer           */
    AP_UINT32      buf_size;       /* buffer size                 */
    AP_UINT32      total_buf_size; /* total buffer size required  */
    AP_UINT16      num_entries;    /* number of entries           */
    AP_UINT16      total_num_entries; /* total number of entries    */
    unsigned char  list_options;   /* listing options             */
    unsigned char  reserv3;       /* reserved                     */
    unsigned char  display_lu_name[8]; /* Display LU name            */
} QUERY_TN3270_ASSOCIATION;

typedef struct tn3270_association
{
    AP_UINT16      overlay_size;    /* Overlay size                */
    unsigned char  reserv2[2];     /* reserved                    */
    unsigned char  display_lu_name[8]; /* Display LU name            */
    TN3270_ASSOCIATION_DEF_DATA def_data; /* association definition     */
} TN3270_ASSOCIATION;

typedef struct tn3270_association_def_data
{
    unsigned char  description[32]; /* resource description        */
    unsigned char  reserve0[16];   /* reserved                    */
    unsigned char  printer_lu_name[8]; /* name of printer LU/pool    */
    unsigned char  reserv2[8];    /* reserved                    */
} TN3270_ASSOCIATION_DEF_DATA;
```

データは tn3270_association 構造体のフォーマットで戻されます。

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TN3270_ASSOCIATION

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

関連の最大数 (その関連のデータが戻されます)。ある範囲でなく、特定の関連に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すには 0 を指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すための関連リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

display_lu_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

display_lu_name パラメーターで指定したエントリーの直後のエントリーから開始します。

display_lu_name

関連情報を必要とするディスプレイ LU の名前、または関連のリストの索引として使用する名前。ディスプレイ LU 名は EBCDIC スtringで、右側に EBCDIC スペースを入れます。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

バッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。これは、*buf_size* パラメーターに指定した値より大きい場合があります。

num_entries

実際に戻されたエントリーの数。

total_num_entries

戻されている可能性があるエントリーの合計数。これは、*num_entries* パラメーターに指定した値より大きい場合があります。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

tn3270_association.overlay_size

戻された *tn3270_association* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *tn3270_association* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の *sizeof()* 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

QUERY_TN3270_ASSOCIATION

tn3270_association.display_lu_name

association.printer_lu_name パラメーターで指定したプリンター LU に関連したディスプレイ LU の名前。これは、右側に EBCDIC のスペースを入れた EBCDIC ストリングです。

tn3270_association_def_data.description

関連を記述したヌルで終了するテキスト・ストリング (その関連の定義で指定したもの)。

tn3270_association_def_data.printer_lu_name

association.display_lu_name パラメーターで指定したディスプレイ LU に関連したプリンター LU の名前。これは、右側に EBCDIC のスペースを入れた EBCDIC ストリングです。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LU_NAME

次のいずれかを示します。

- *list_options* パラメーターを AP_LIST_FROM_NEXT に設定しましたが、ディスプレイ LU 名が有効な EBCDIC ストリングではありませんでした。
- *list_options* パラメーターを AP_LIST_INCLUSIVE に設定しましたが、ディスプレイ LU 名が有効な EBCDIC ストリングでないか、既存の関連レコードに対応していませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TN3270_DEFAULTS

QUERY_TN3270_DEFAULTS は、すべてのクライアント・セッションで使用する TN3270 パラメーターに関する情報を戻します。

セキュア・ソケット層 (SSL) クライアント認証を使用し、外部の LDAP サーバーで証明書取り消しリストに反するクライアントを検査している場合には、`QUERY_TN3270_SSL_LDAP verb` を使用して、このサーバーへのアクセス方法の詳細を戻します。

VCB 構造体

```
typedef struct query_tn3270_defaults
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    TN3270_DEFAULTS_DEF_DATA def_data; /* TN3270 defaults              */
} QUERY_TN3270_DEFAULTS;

typedef struct tn3270_defaults_def_data
{
    AP_UINT16      force_responses; /* force printer responses?     */
    AP_UINT16      keepalive_method; /* method for sending keep-alives */
    AP_UINT32      keepalive_interval; /* interval between keep-alives */
    unsigned char  reserv2[32];    /* reserved                      */
} TN3270_DEFAULTS_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

`opcode` `AP_QUERY_TN3270_DEFAULTS`

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

`primary_rc`
`AP_OK`

`def_data.force_responses`
プリンター・セッションでのクライアント応答を制御します。値は次のいずれかです。

AP_YES 確定応答を要求します。

AP_NO SNA トラフィックに適合した応答を要求します。

`def_data.keepalive_method`

キープアライブ・メッセージの送信方式。キープアライブ・メッセージとは、接続に他のアクティビティがない場合に TN3270 クライアントへの TCP/IP 接続をアクティブなままにするために、このクライアントに対して送信されるメッセージのことです。これにより、失敗した接続およびクライアントが検出されます。TCP/IP 接続にトラフィックがない場合は、接続やクライアントの失敗は検出されないため、TN サーバーのリソースが浪費され、LU が他のセッションで使用されなくなります。

値は次のいずれかです。

AP_NONE
キープアライブ・メッセージは送信しません。

QUERY_TN3270_DEFAULTS

AP_TN3270_NOP

Telnet NOP メッセージを送信します。

AP_TN3270_TM

Telnet DO TIMING-MARK メッセージを送信します。

def_data.keepalive_interval

連続したキープアライブ・メッセージ間の間隔 (秒単位)。特に多数のアイドル状態のクライアント接続が常にある場合には、この間隔を十分長くして、ネットワーク・トラフィックを最小化します。キープアライブの間隔を短くすると、より素早く障害を検出することができますが、ネットワーク・トラフィックは増大します。キープアライブの間隔が必要以上に短く、また多数のクライアントが存在するとき、このトラフィックが重大なものになることがあります。

TCP/IP の作動上の理由から、構成したキープアライブ間隔は、サーバーがクライアントの消失を認識する際の正確な所要時間にはなりません。正確な時間は、さまざまな要因によって異なりますが、構成されたタイムアウトにさらに数分加えた時間の 2 倍以上にはなりません (正確な数字は、TCP/IP の構成方法によって異なります)。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TN3270_EXPRESS_LOGON

QUERY_TN3270_EXPRESS_LOGON は、TN3270 高速ログオン機能に関する情報を戻します。この機能により、Secure Sockets Layer (SSL) クライアント認証機能を使用して Communications Server for Linux TN サーバーまたは TN リダイレクターに接続する TN3270 クライアント・ユーザーは、TN3270 セキュリティーに対して通常使用するユーザー ID およびパスワードを指定する必要はありません。このユーザーのセキュリティ証明書は、必要なユーザー ID およびパスワードを提供するホストの Digital Certificate Access Server (DCAS) に照らして検査されます。

VCB 構造体

```
typedef struct query_tn3270_express_logon
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;              /* reserved                     */
    unsigned char  format;              /* reserved                     */
    AP_UINT16      primary_rc;          /* primary return code          */
    AP_UINT32      secondary_rc;        /* secondary return code        */
    unsigned char  dcas_server[256];    /* IP hostname of DCAS server   */
    AP_UINT16      dcas_port;           /* port number to access server */
    unsigned char  enabled;             /* is Express Logon enabled?    */
    unsigned char  reserv3[33];        /* reserved                     */
} QUERY_TN3270_EXPRESS_LOGON;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TN3270_EXPRESS_LOGON

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

dcas_server

高速ログオン許可を処理するホスト DCAS サーバーの TCP/IP アドレス。指定できるのは、以下のいずれかです。

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

dcas_port

DCAS サーバーへのアクセスに使用する TCP/IP ポート番号。

enabled

TN3270 高速ログオン機能を使用可能にするかどうかを指定します。値は次のいずれかです。

AP_YES この機能を使用可能にします。TN3270 クライアントは、ホストにアクセスする場合にユーザー ID およびパスワードを指定する必要がありません。

AP_NO この機能を使用不可にします。TN3270 クライアントは、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。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TN3270_SSL_LDAP

QUERY_TN3270_SSL_LDAP は、セキュア・ソケット層 (SSL) クライアント認証機能で使用する証明書取り消しリストへのアクセス方法に関する情報を戻します。この情報は、DEFINE_TN3270_SSL_LDAP verb を使用して指定されました。

VCB 構造体

```
typedef struct query_tn3270_ssl_ldap
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;               /* reserved                     */
    unsigned char  format;               /* reserved                     */
    AP_UINT16      primary_rc;           /* primary return code         */
    AP_UINT32      secondary_rc;        /* secondary return code       */
    unsigned char  *buf_ptr;            /* pointer to buffer           */
    AP_UINT32      buf_size;            /* buffer size                 */
    AP_UINT32      total_buf_size;     /* total buffer size required  */
    AP_UINT16      num_entries;         /* reserved                    */
}
```

QUERY_TN3270_SSL_LDAP

```
    AP_UINT16      total_num_entries;    /* reserved */
    unsigned char  list_options;        /* reserved */
    unsigned char  reserv3;             /* reserved */
} QUERY_TN3270_SSL_LDAP;
typedef struct tn3270_ssl_ldap_def_data
{
    AP_UINT16      overlay_size;        /* reserved */
    unsigned char  auth_type;          /* type of authorization checking */
    unsigned char  reserv1;           /* reserved */
    unsigned char  ldap_addr[256];    /* address of LDAP server */
    AP_UINT16      ldap_port;         /* port number to access server */
    unsigned char  ldap_user[1024];   /* user ID on LDAP server */
    unsigned char  ldap_password[128]; /* password on LDAP server */
    unsigned char  reserv2[256];     /* reserved */
} TN3270_SSL_LDAP_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TN3270_SSL_LDAP

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

バッファに戻された情報の長さ。

total_buf_size

完全な情報を戻すために必要だったバッファのサイズを示す戻り値。これは、*buf_size* パラメーターに指定した値より大きい場合があります。

以下の情報は、データ・バッファに戻されます。

def_data.auth_type

TN サーバーまたは TN リダイレクターが実行する許可検査のタイプを指定します。値は次のいずれかです。

AP_LOCAL_ONLY

サーバーはクライアント証明書をローカルで検査しますが、外部の証明書取り消しリストは使用しません。パラメーター *ldap_addr*—*ldap_password* は予約済みです。

AP_LOCAL_X500

サーバーは証明書をローカルで検査しますが、外部の証明書取り消しリストも検査します。このデータ構造体の残りのパラメーターでは、このリストの場所を指定します。

def_data.ldap_addr

証明書取り消しリストを保持する LDAP サーバーの TCP/IP アドレス。指定できるのは、以下のいずれかです。

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

def_data.ldap_port

LDAP サーバーへのアクセスに使用する TCP/IP ポート番号。

def_data.ldap_user

LDAP サーバー上の証明書取り消しリストへのアクセスに使用するユーザー名。

def_data.ldap_password

LDAP サーバー上の証明書取り消しリストへのアクセスに使用するパスワード。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TN_REDIRECT_DEF

QUERY_TN_REDIRECT_DEF は、ホストへのアクセスに Communications Server for Linux の TN リダイレクター機能を使用できる他のコンピューター上の Telnet クライアントに関する情報を戻します。この verb は、使用するオプションに応じて、単一のユーザーまたは複数のユーザーに関する要約情報または詳細情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_tn_redirect_def
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;                /* reserved                  */
    AP_UINT16      primary_rc;            /* primary return code      */
    AP_UINT32      secondary_rc;          /* secondary return code    */
    unsigned char  *buf_ptr;              /* pointer to buffer        */
    AP_UINT32      buf_size;               /* buffer size              */
    AP_UINT32      total_buf_size;        /* total buffer size required */
    AP_UINT16      num_entries;            /* number of entries        */
    AP_UINT16      total_num_entries;     /* total number of entries  */
    unsigned char  list_options;          /* listing options          */
    unsigned char  reserv3[3];            /* reserved                  */
    TN_REDIRECT_ADDRESS addr;             /* Uniquely defines record  */
} QUERY_TN_REDIRECT_DEF;

typedef struct tn_redirect_data
{
    AP_UINT16      overlay_size;           /* overlay size              */

```

QUERY_TN_REDIRECT_DEF

```
    unsigned char    reserv1[2];        /* Reserved */
    TN_REDIRECT_ADDRESS    addr;        /* addressing information */
    TN_REDIRECT_DEF_DATA    def_data;    /* definitions for the client */
} TN_REDIRECT_DATA;
typedef struct tn_redirect_address
{
    AP_UINT16    default_record;        /* Is this the default record ? */
    unsigned char    address_format;    /* IP address or fully-qualified name */
    unsigned char    client_address[256]; /* Client address */
    AP_UINT16    port_number;          /* Port number that client connects on */
    unsigned char    listen_local_address[46]; /* Local addr client connects to */
    unsigned char    reserved[34];      /* reserved */
} TN_REDIRECT_ADDRESS;
typedef struct tn_redirect_def_data
{
    unsigned char    description[32];    /* Description - null terminated */
    unsigned char    reserve0[16];      /* Reserved */
    unsigned char    cli_conn_ssl_enabled; /* Is the client session SSL? */
    unsigned char    serv_conn_ssl_enabled; /* Is the host session SSL? */
    unsigned char    host_address_format; /* Type of IP address for the host */
    unsigned char    reserv1;          /* Reserved */
    unsigned char    host_address[256]; /* Host address */
    AP_UINT16    host_port_number;      /* Port number to connect to host */
    unsigned char    cli_conn_security_level; /* SSL encryption strength */
    unsigned char    serv_conn_security_level; /* SSL encryption strength */
    unsigned char    cli_conn_cert_key_label[80]; /* Key label for certificate */
    unsigned char    serv_conn_cert_key_label[80]; /* Key label for certificate */
    unsigned char    reserved[46];      /* Reserved */
} TN_REDIRECT_DEF_DATA;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TN_REDIRECT_DEF

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファーへのポインター。

buf_size

指定したデータ・バッファーのサイズ。

num_entries

ユーザーの最大数 (そのユーザーのデータが戻されます)。ある範囲でなく、特定のユーザーに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のユーザーから開始します。

AP_LIST_INCLUSIVE

指定したクライアント・アドレッシング情報で指定したエントリーから開始します。

AP_LIST_FROM_NEXT

指定されたクライアント・アドレッシング情報で指定したエントリーのすぐ後のエントリーから開始します。

リストはクライアント・アドレスの順序に並べられます。リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法の詳細については、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

addr 情報を求める Telnet クライアント、またはユーザーのリストの索引として使用するユーザーに関するアドレッシング情報を指定します。このデータ構造体の内容に関する詳細については、245 ページの『DEFINE_TN_REDIRECT』を参照してください。

list_options が AP_FIRST_IN_LIST に設定されている場合には、この構造体内の情報は無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

バッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。この値は *buf_size* より大きい場合があります。

total_num_entries

戻されている可能性があるエントリーの合計数。この値は *num_entries* より大きい場合があります。

num_entries

実際に戻されたエントリーの数。

tn_redirect_data.overlay_size

戻された *tn_redirect_data* 構造体のサイズ。すなわち、データ・バッファー内における次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *tn_redirect_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

tn_redirect_data.addr

Telnet クライアントに関するアドレッシング情報を指定します。このデータ構造体の内容に関する詳細については、245 ページの『DEFINE_TN_REDIRECT』を参照してください。

QUERY_TN_REDIRECT_DEF

tn_redirect_data.def_data

Telnet クライアントの定義を指定します。このデータ構造体の内容に関する詳細については、245 ページの『DEFINE_TN_REDIRECT』を参照してください。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_CLIENT_ADDRESS

list_options パラメーターを AP_LIST_INCLUSIVE に設定しましたが、指定されたアドレッシング情報は、定義済みのどの TN リダイレクターにも一致しませんでした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TN_SERVER_TRACE

この `verb` は、Communications Server for Linux TN サーバー機能の現行トレース・オプションに関する情報を戻します。

この `verb` は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_tn_server_trace
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;          /* primary return code */
    AP_UINT32      secondary_rc;        /* secondary return code */
    AP_UINT16      trace_flags;         /* trace flags */
    unsigned char  reserv3[6];          /* Reserved */
} QUERY_TN_SERVER_TRACE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TN_SERVER_TRACE

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

trace_flags

現在活動状態であるトレースのタイプ。

活動状態であるトレースがない場合、またはすべてのタイプのトレースが活動状態である場合は、値は次のいずれかです。

AP_TN_SERVER_NO_TRACE

トレースしません。

AP_TN_SERVER_ALL_TRACE

すべてのタイプをトレースします。

特定のインターフェースのトレースを使用している場合、このパラメーターは、次のリストの値の 1 つ以上を論理 OR 演算で組み合わせた値に設定されます。

AP_TN_SERVER_TRC_TCP

TCP/IP インターフェース・トレース: TN サーバーと TN3270 クライアント間のメッセージ

AP_TN_SERVER_TRC_FM

ノード・インターフェース・トレース: 内部制御メッセージ、および TN サーバーと TN3270 クライアント間のメッセージ (内部フォーマット)

AP_TN_SERVER_TRC_CFG

構成メッセージ・トレース: TN サーバーの構成に関連するメッセージ

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TP

QUERY_TP は、現在ローカル LU を使用している TP に関する情報を戻します。この *verb* は、使用するオプションに応じて、特定の TP に関する情報を入手するか、複数の TP に関する情報を入手するために使用できます。この *verb* が戻すのは TP の現在の使用状況に関する情報であって、TP の定義ではありません。TP の定義は、QUERY_TP_DEFINITION を使用して入手してください。

この *verb* は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```

typedef struct query_tp
{
    AP_UINT16      opcode;                /* Verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;                /* reserved                  */
    AP_UINT16      primary_rc;            /* Primary return code      */
    AP_UINT32      secondary_rc;          /* Secondary return code    */
    unsigned char  *buf_ptr;              /* pointer to buffer        */
    AP_UINT32      buf_size;              /* buffer size              */
    AP_UINT32      total_buf_size;        /* total buffer size required */
    AP_UINT16      num_entries;           /* number of entries        */
    AP_UINT16      total_num_entries;     /* total number of entries  */
    unsigned char  list_options;          /* listing options          */
    unsigned char  reserv3;               /* reserved                  */
    unsigned char  lu_name[8];            /* LU name                  */
    unsigned char  lu_alias[8];           /* LU alias                  */
    unsigned char  tp_name[64];           /* TP name                   */
} QUERY_TP;

typedef struct tp_data
{
    AP_UINT16      overlay_size;          /* size of returned entry   */
    unsigned char  tp_name[64];           /* TP name                   */
    unsigned char  description[32];       /* resource description      */
    unsigned char  reserv1[16];           /* reserved                  */
    AP_UINT16      instance_limit;        /* maximum instance count   */
    AP_UINT16      instance_count;        /* current instance count   */
    AP_UINT16      locally_started_count; /* locally started instance */
                                                    /* count                    */
    AP_UINT16      remotely_started_count; /* remotely started instance */
                                                    /* count                    */
    unsigned char  reserva[20];           /* reserved                  */
} TP_DATA;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TP

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファーへのポインター。

buf_size

指定したデータ・バッファーのサイズ。

num_entries

TP の最大数 (それらの TP についてのデータが戻されます)。ある範囲でなく、特定の TP に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すための TP リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

LU 名と TP 名を組み合わせで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

LU 名と TP 名を組み合わせで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

lu_name

LU 名。この名前は、8 バイトからなるタイプ A の EBCDIC スtring で、名前が 8 文字に満たない場合は、右側にスペースを入れます。LU を LU 名ではなく、別名で識別することを指定するには、このパラメーターを 8 桁の 2 進ゼロに設定し、次のパラメーターで LU 別名を指定します。ローカル CP に関連した LU (デフォルト LU) を指定するには、*lu_name* と *lu_alias* の両方をすべて 2 進ゼロに設定します。

lu_alias

ローカルで定義された LU 別名。これは 8 バイトからなる ASCII String で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。このパラメーターは、*lu_name* を 8 桁の 2 進ゼロに設定した場合のみ使用され、それ以外の場合は無視されます。ローカル CP に関連した LU (デフォルト LU) を指定するには、*lu_name* と *lu_alias* の両方をすべて 2 進ゼロに設定します。

tp_name

TP 名。これは 64 バイトの String で、名前が 64 文字に満たない場合は、右側にスペースを入れます。この値は、*list_options* を *AP_FIRST_IN_LIST* に設定した場合には無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファ内の各エントリは、次のパラメーターで構成されています。

tp_data.overlay_size

戻された *tp_data* 構造体のサイズ。すなわち、データ・バッファにおける次のエントリの先頭までのオフセット。

アプリケーションが戻されたバッファを調べて、各 *tp_data* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

tp_data.tp_name

TP 名。これは 64 バイトのストリングで、名前が 64 文字に満たない場合は、右側にスペースを入れます。

tp_data.description

TP を記述したヌルで終了するテキスト・ストリング (その TP の定義で指定したもの)。

tp_data.instance_limit

指定した TP の同時に活動状態であるインスタンスの最大数。

tp_data.instance_count

指定した TP の現在活動状態であるインスタンスの数。

tp_data.locally_started_count

ローカルで (TP で `TP_STARTED verb` を発行して) 開始された TP の数。

tp_data.remotely_started_count

リモートで (受信された `Attach` 要求により) 開始された TP の数。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_PARAMETER_CHECK`

secondary_rc

値は次のいずれかです。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_LU_ALIAS

指定した *lu_alias* パラメーターが、有効ではありませんでした。

AP_INVALID_LU_NAME

指定した *lu_name* パラメーターが、有効ではありませんでした。

AP_INVALID_TP_NAME

list_options パラメーターが、指定した名前から始まるすべてのエン

トリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*tp_name* パラメーターが無効でした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TP_DEFINITION

QUERY_TP_DEFINITION は、Communications Server for Linux システムで定義されている TP に関する情報を戻します。この verb は、使用するオプションに応じて、特定の TP に関する情報を入手するか、複数の TP に関する情報を入手するために使用できます。この verb が戻すのは TP の定義に関する情報であって、その現在の使用状況ではありません。使用状況の情報を入手するには、QUERY_TP を使用してください。

VCB 構造体

```
typedef struct query_tp_definition
{
    AP_UINT16      opcode;                /* Verb operation code          */
    unsigned char  reserv2;               /* reserved                      */
    unsigned char  format;                /* reserved                      */
    AP_UINT16      primary_rc;            /* Primary return code          */
    AP_UINT32      secondary_rc;          /* Secondary return code        */
    unsigned char  *buf_ptr;              /* pointer to buffer            */
    AP_UINT32      buf_size;              /* buffer size                  */
    AP_UINT32      total_buf_size;        /* total buffer size required    */
    AP_UINT16      num_entries;           /* number of entries            */
    AP_UINT16      total_num_entries;     /* total number of entries      */
    unsigned char  list_options;          /* listing options              */
    unsigned char  reserv3;               /* reserved                      */
    unsigned char  tp_name[64];           /* TP name                      */
} QUERY_TP_DEFINITION;

typedef struct tp_def_summary
{
    AP_UINT16      overlay_size;           /* size of returned entry       */
    unsigned char  tp_name[64];           /* TP name                      */
    unsigned char  description[32];        /* resource description          */
    unsigned char  reserv1[16];           /* reserved                      */
} TP_DEF_SUMMARY;

typedef struct tp_def_detail
{
    AP_UINT16      overlay_size;           /* size of returned entry       */
    unsigned char  tp_name[64];           /* TP name                      */
    TP_CHARS       tp_chars;              /* TP characteristics           */
} TP_DEF_DETAIL;

typedef struct tp_chars
{
    unsigned char  description[32];        /* resource description          */
    unsigned char  security_list_name[14]; /* security access list name    */
    unsigned char  reserv1[2];            /* reserved                      */
    unsigned char  conv_type;             /* conversation type            */
    unsigned char  security_rq;           /* security support              */
    unsigned char  sync_level;            /* synchronization level support */
}
```

QUERY_TP_DEFINITION

```
unsigned char    dynamic_load;        /* dynamic load          */
unsigned char    enabled;             /* is the TP enabled?    */
unsigned char    pip_allowed;         /* program initialization */
                                        /* parameters supported   */
unsigned char    reserv3[10];         /* reserved              */
AP_UINT16        tp_instance_limit;   /* limit on currently active TP
                                        /* instances              */
AP_UINT16        incoming_alloc_timeout; /* incoming allocation timeout */
AP_UINT16        rcv_alloc_timeout;   /* receive allocation timeout */
AP_UINT16        tp_data_len;         /* reserved              */
unsigned char    tp_data[120];        /* reserved              */
} TP_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TP_DEFINITION

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファーへのポインター。

buf_size

指定したデータ・バッファーのサイズ。

num_entries

TP の最大数 (それらの TP についてのデータが戻されます)。ある範囲でなく、特定の TP に関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファーに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すためのリスト内の開始位置、および各エントリーに必要な情報レベル。情報レベルは、次のいずれかの値で指定します。

AP_SUMMARY

要約情報のみ。

AP_DETAIL

詳細情報。

論理 OR 演算を使用して、この値を次のいずれかの値と組み合わせてください。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

tp_name パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

tp_name パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

tp_name

TP 名。これは 64 バイトのストリングで、名前が 64 文字に満たない場合は、右側にスペースを入れます。このパラメーターは、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。*buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

tp_def_summary.overlay_size

戻された *tp_def_summary* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *tp_def_summary* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の *sizeof()* 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

tp_def_summary.tp_name

TP 名。これは 64 バイトのストリングで、名前が 64 文字に満たない場合は、右側にスペースを入れます。

tp_def_summary.description

TP を記述したヌルで終了するテキスト・ストリング (その TP の定義で指定したもの)。

tp_def_detail.overlay_size

戻された *tp_def_detail* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

QUERY_TP_DEFINITION

アプリケーションが戻されたバッファを調べて、各 `tp_def_detail` 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは `Communications Server for Linux` の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

tp_def_detail.tp_name

TP 名。これは 64 バイトのストリングで、名前が 64 文字に満たない場合は、右側にスペースを入れます。

tp_def_detail.tp_chars.description

TP を記述したヌルで終了するテキスト・ストリング (その TP の定義で指定したもの)。

tp_def_detail.tp_chars.security_list_name

この TP が使用するセキュリティー・アクセス・リストの名前 (`DEFINE_SECURITY_ACCESS_LIST verb` を使用して定義されたもの)。このパラメーターは、指定リストに名前のあるユーザーのみが TP に会話を割り振れるように TP を制限します。

このパラメーターが 14 桁の 2 進ゼロに設定されている場合は、すべてのユーザーがこの TP を使用できます。

tp_def_detail.tp_chars.conv_type

TP がサポートする会話のタイプ (1 つ以上) を指定します。値は次のいずれかです。

AP_BASIC

TP は基本会話のみをサポートします。

AP_MAPPED

TP はマップ式会話のみをサポートします。

AP_EITHER

TP は基本会話またはマップ式会話をサポートします。

tp_def_detail.tp_chars.security_rqd

TP の開始に必要な会話セキュリティー情報のレベルを指定します。値は次のいずれかです。

AP_YES TP を起動するためにユーザー ID とパスワードが必要です。

AP_NO セキュリティー情報は必要ありません。

tp_def_detail.tp_chars.sync_level

TP がサポートする同期レベルの値を指定します。値は次のいずれかです。

AP_NONE

TP は `sync_level NONE` のみをサポートします。

AP_CONFIRM_SYNC_LEVEL

TP は `sync_level CONFIRM` のみをサポートします。

AP_EITHER

TP は `sync_level NONE` か `CONFIRM` のいずれかをサポートします。

AP_SYNCPT_REQUIRED

TP は *sync_level* SYNCPT のみをサポートします (同期点が必要です)。

AP_SYNCPT_NEGOTIABLE

TP は 3 つの *sync_level* の値 (NONE、CONFIRM、SYNCPT) をサポートします。

tp_def_detail.tp_chars.dynamic_load

TP を動的にロードできるかどうかを指定します。これは AP_YES に設定されます。

tp_def_detail.tp_chars.enabled

TP を正常に接続できるかどうかを指定します。値は次のいずれかです。

AP_YES TP を接続できます。

AP_NO TP を接続できません。

tp_def_detail.tp_chars.pip_allowed

TP がプログラム初期設定パラメーター (PIP) を受信できるかどうかを指定します。値は次のいずれかです。

AP_YES TP は PIP を受信できます。

AP_NO TP は PIP を受信できません。

tp_def_detail.tp_chars.duplex_support

TP がどちらの会話二重タイプをサポートするかを指定します。値は次のいずれかです。

AP_HALF_DUPLEX

TP は半二重会話のみをサポートします。

AP_FULL_DUPLEX

TP は全二重会話をサポートします。

AP_EITHER_DUPLEX

TP は半二重会話と全二重会話をサポートします。

tp_def_detail.tp_chars.tp_instance_limit

同時に活動状態である TP インスタンスの数の制限。

tp_def_detail.tp_chars.incoming_alloc_timeout

着呼 Attach 要求が RECEIVE_ALLOCATE を待つキューに入っている秒数を指定します。値 0 (ゼロ) を指定すると、タイムアウトがないことになり、着呼 Attach 要求のキュー内での滞在期間は無制限になります。

tp_def_detail.tp_chars.rcv_alloc_timeout

RECEIVE_ALLOCATE verb が着呼 Attach 要求を待つキューに入っている秒数。値 0 (ゼロ) を指定すると、タイムアウトがないことになり、RECEIVE_ALLOCATE verb のキュー内での滞在期間は無制限になります。

tp_def_detail.tp_chars.tp_data_len

インプリメンテーション・システムに依存する TP データの長さ。

tp_def_detail.tp_chars.tp_data

Communications Server for Linux はこのパラメーターを使用しません (すべてゼロに設定されます)。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_TP_NAME

list_options パラメーターが、指定した名前から始まるすべてのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*tp_name* パラメーターが無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TP_LOAD_INFO

QUERY_TP_LOAD_INFO は、TP ロード情報エントリーに関する情報を戻します。バッファーにはさまざまなサイズの `tp_load_info` 構造体が入ります。

この `verb` は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_tp_load_info
{
    AP_UINT16          opcode;           /* Verb operation code      */
    unsigned char     reserv2;          /* reserved                 */
    unsigned char     format;           /* reserved                 */
    AP_UINT16         primary_rc;       /* Primary return code     */
    AP_UINT32         secondary_rc;     /* Secondary return code   */
    unsigned char     *buf_ptr;         /* pointer to buffer       */
    AP_UINT32         buf_size;         /* buffer size             */
    AP_UINT32         total_buf_size;   /* total buffer size required */
    AP_UINT16         num_entries;      /* number of entries       */
    AP_UINT16         total_num_entries; /* total number of entries */
    unsigned char     list_options;     /* listing options        */
    unsigned char     reserv3[3];       /* reserved                 */
    unsigned char     tp_name[64];      /* TP name                 */
    unsigned char     lu_alias[8];      /* LU alias                 */
} QUERY_TP_LOAD_INFO;

typedef struct tp_load_info
{
    AP_UINT16          overlay_size;     /* size of returned entry  */
```

```

    unsigned char    tp_name[64];           /* TP name          */
    unsigned char    lu_alias[8];          /* LU alias         */
    TP_LOAD_INFO_DEF_DATA def_data;       /* defined data     */
} TP_LOAD_INFO;

typedef struct tp_load_info_def_data
{
    unsigned char    description[32];      /* Description      */
    unsigned char    reserv1[16];         /* reserved         */
    unsigned char    user_id[64];         /* User ID         */
    unsigned char    group_id[64];        /* Group ID        */
    unsigned short   timeout;             /* Timeout value   */
    unsigned char    type;                /* TP type         */
    unsigned char    reserv2;             /* reserved         */
    AP_UINT16        reserv3;             /* reserved         */
    AP_UINT16        ltv_length;          /* Length of LTV data */
} TP_LOAD_INFO_DEF_DATA;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TP_LOAD_INFO

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size

指定したデータ・バッファのサイズ。

num_entries

追加のデータ制御ブロックの最大数 (そのデータ制御ブロックのデータが戻されます)。ある範囲でなく、特定のリソースに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options

Communications Server for Linux がデータを戻すための TP リスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

TP 名と LU 別名を組み合わせで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

TP 名と LU 別名を組み合わせで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

tp_name

照会する TP 名。これは 64 バイトからなる EBCDIC スtring で、名前が 64 文字に満たない場合は、右側にスペースを入れます。すべての TP に

一致させるには、すべて 2 進ゼロを指定します。この値は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

lu_alias

照会する LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。すべての LU に一致させるには、すべて 2 進ゼロを指定します。

このパラメーターは、TP が APPC アプリケーションである場合にのみ使用できます。TP が CPI-C アプリケーションである場合は予約済みです。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

buf_size

指定したバッファーに戻された情報の長さ。

total_buf_size

要求したリスト情報をすべて戻すために必要だったバッファーのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。 *num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

tp_load_info.overlay_size

このオーバーレイのサイズ (LTV データを含む)。このサイズには、次のオーバーレイが正しく位置合わせされたメモリー位置に配置されるようにスペースが入ります。

アプリケーションが戻されたバッファーを調べて、各 *tp_load_info* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の *sizeof()* 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

tp_load_info.tp_name

TP ロード情報エントリーの TP 名。これは 64 バイトからなる EBCDIC ストリングで、名前が 64 文字に満たない場合は、右側にスペースを入れます。

tp_load_info.lu_alias

TP ロード情報エントリーの LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

このパラメーターを使用するのは、TP が APPC アプリケーションである場合のみです。TP が CPI-C アプリケーションの場合には使用しません。

def_data.description

TP ロード情報の記述。

def_data.user_id

TP にアクセスして実行するために必要なユーザー ID。

def_data.group_id

TP にアクセスして実行するために必要なグループ ID。

def_data.timeout

TP ロード後のタイムアウト (秒単位)。

def_data.type

TP タイプを示します。値は次のいずれかです。

AP_TP_TYPE_QUEUED

AP_TP_TYPE_QUEUED_BROADCAST

AP_TP_TYPE_NON_QUEUED

def_data.ltv_length

この構造体に追加される LTV データ・バッファの長さ。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_TP_NAME

tp_name パラメーターが、定義されている TP の名前に一致しませんでした。

AP_INVALID_LU_ALIAS

lu_alias パラメーターが、定義されているどの LU 別名とも一致しませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TRACE_FILE

この verb は、Communications Server for Linux がトレース・データの記録に使用するファイルに関する情報を戻します。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_trace_file
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;               /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;        /* secondary return code */
    unsigned char  trace_file_type;     /* type of trace file */
    unsigned char  dual_files;          /* dual trace files */
    AP_UINT32      trace_file_size;     /* trace file size */
    unsigned char  reserv3[4];          /* reserved */
    unsigned char  file_name[81];       /* file name */
    unsigned char  file_name_2[81];     /* second file name */
} QUERY_TRACE_FILE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TRACE_FILE

trace_file_type

トレース・ファイルのタイプ。値は次のいずれかです。

AP_CS_TRACE

指定したコンピューターとその他のノードの間で Communications Server for Linux LAN を介して転送されるデータ (SET_CS_TRACE verb により活動化されたもの) のトレースを入れるファイル。

AP_TN_SERVER_TRACE

Communications Server for Linux TN サーバー・コンポーネントのトレースを入れるファイル。

AP_IPS_TRACE

指定したノードのカーネル・コンポーネントのトレースを入れるファイル (SET_TRACE_TYPE verb または ADD_DLC_TRACE verb により活動化されたもの)。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

dual_files

1 つのファイルにトレースするか 2 つのファイルにトレースするかを指定します。値は次のいずれかです。

AP_YES 2 つのファイルへのトレース。最初のファイルが *trace_file_size* で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、2 番目のファイルへのトレースが継続されます。次に、2 番目のファイルが *trace_file_size* で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、最初のファイルへのトレースが継続されます。これにより、余分なディスク・スペースを使用しないで長期にわたってトレースを継続できます。必要な最大スペースは *trace_file_size* の値の約 2 倍です。

AP_NO 1 つのファイルへのトレース。

trace_file_size

トレース・ファイルの最大サイズ。*dual_files* を AP_YES に設定した場合は、現行ファイルがこのサイズに達すると、2 つのファイル間でトレースが切り替わります。*dual_files* を AP_NO に設定した場合は、このパラメーターは無視されます。ファイル・サイズに制限はありません。

file_name

トレース・ファイルの名前。*dual_files* を AP_YES に設定した場合には、最初のトレース・ファイルの名前。このパラメーターは 1 ~ 80 文字の ASCII スtringで、最後にヌル文字 (2 進ゼロ) が続きます。

パスが指定されていない場合、このファイルは診断ファイルのデフォルト・ディレクトリー **/var/opt/ibm/sn** に保管されます。パスが指定されている場合は、絶対パス (/ 文字で始まる) か、デフォルト・ディレクトリーに対する相対パスのいずれかです。

file_name_2

2 番目のトレース・ファイルの名前。このパラメーターは *dual_files* を AP_YES に設定した場合のみ使用されます。このパラメーターは 1 ~ 80 文字の ASCII スtringで、最後にヌル文字 (2 進ゼロ) が続きます。

パスが指定されていない場合、このファイルは診断ファイルのデフォルト・ディレクトリー **/var/opt/ibm/sn** に保管されます。パスが指定されている場合は、絶対パス (/ 文字で始まる) か、デフォルト・ディレクトリーに対する相対パスのいずれかです。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_FILE_TYPE

trace_file_type パラメーターが、有効な値に設定されていませんでした。

QUERY_TRACE_FILE

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_TRACE_TYPE

この verb は、Communications Server for Linux カーネル・コンポーネントの現行トレース・オプションに関する情報を戻します。トレース・オプションについて詳しくは、「*Communications Server for Linux 管理ガイド*」を参照してください。

この verb は DLC 回線トレースに関する情報は戻しません。
QUERY_DLC_TRACE verb を使用して情報を入手してください。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct query_trace_type
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    AP_UINT16      trace_flags;     /* trace flags                  */
    AP_UINT32      truncation_length; /* truncate each msg to this size */
    AP_UINT16      internal_level;  /* reserved                    */
    AP_UINT32      api_flags;       /* reserved                    */
} QUERY_TRACE_TYPE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_TRACE_TYPE

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

trace_flags
現在活動状態であるトレースのタイプ。これらのトレース・タイプについて詳しくは、751 ページの『SET_TRACE_TYPE』を参照してください。

活動状態であるトレースがない場合、またはすべてのタイプのトレースが活動状態である場合は、値は次のいずれかです。

AP_NO_TRACE

トレースしません。

AP_ALL_TRACE

すべてのタイプをトレースします。

特定のインターフェースのトレースを使用している場合、このパラメーターは、次のリストの値の 1 つ以上を論理 OR 演算で組み合わせた値に設定されます。

AP_APPC_MSG

APPC メッセージ

AP_FM_MSG

FM メッセージ

AP_LUA_MSG

LUA メッセージ

AP_NOF_MSG

NOF メッセージ

AP_MS_MSG

MS メッセージ

AP_GSNA_MSG

汎用 SNA メッセージ

AP_PV_MSG

(Communications Server for Linux のこのバージョンでは未使用)

AP_LLC2_MSG

LLC2 メッセージ

AP_LLI_MSG

LLI メッセージ

AP_MAC_MSG

MAC メッセージ

AP_SDLC_MSG

SDLC メッセージ

AP_NLI_MSG

NLI メッセージ

AP_IPDL_MSG

Enterprise Extender (HPR/IP) メッセージ

AP_DLC_MSG

ノードから DLC へのメッセージ

AP_NODE_MSG

ノード・メッセージ

AP_SLIM_MSG

クライアント/サーバー・システムで、マスター・サーバーとバックアップ・サーバーの間で送信されるメッセージ

QUERY_TRACE_TYPE

AP_DATAGRAM

データグラム・メッセージ

truncation_length

それぞれのメッセージについて、トレース・ファイルに書き込まれる情報の最大長 (バイト単位)。メッセージがこの値を超える場合、Communications Server for Linux はメッセージの先頭部分のみをトレース・ファイルに書き込み、*truncation_length* を超えるデータは廃棄されます。これにより、各メッセージの最も重要な情報のみを記録し、長いメッセージでファイルがいっぱいにならないようにします。ゼロの値はトレース・メッセージの切り捨てを行わないことを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

QUERY_USERID_PASSWORD

QUERY_USERID_PASSWORD は、APPC および CPI-C の会話セキュリティー用のユーザー ID/パスワードの対に関する情報、または定義されたユーザー ID とパスワードに関する情報を戻します。この verb は、使用するオプションに応じて、ユーザー ID/パスワードの特定の対または複数の対に関する情報を入手するために使用できます。

VCB 構造体

```
typedef struct query_userid_password
{
    AP_UINT16      opcode;           /* Verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* Primary return code      */
    AP_UINT32      secondary_rc;   /* Secondary return code    */
    unsigned char  *buf_ptr;       /* pointer to buffer        */
    AP_UINT32      buf_size;       /* buffer size              */
    AP_UINT32      total_buf_size; /* total buffer size required */
    AP_UINT16      num_entries;    /* number of entries        */
    AP_UINT16      total_num_entries; /* total number of entries */
    unsigned char  list_options;   /* listing options          */
    unsigned char  reserv3;       /* reserved                  */
    unsigned char  user_id[10];    /* user ID                  */
} QUERY_USERID_PASSWORD;

typedef struct userid_info
{
    AP_UINT16      overlay_size;   /* size of returned entry   */
    unsigned char  user_id[10];    /* user ID                  */
    USERID_PASSWORD_CHARS password_chars; /* password characteristics */
} USERID_INFO;

typedef struct userid_password_chars
{
    unsigned char  description[32]; /* resource description     */
    unsigned char  reserv2[16];    /* reserved                  */
    AP_UINT16      profile_count;  /* number of profiles       */
    AP_UINT16      reserv1;       /* reserved                  */
    unsigned char  password[10];  /* password                  */
    unsigned char  profiles[10][10]; /* profiles                  */
} USERID_PASSWORD_CHARS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_QUERY_USERID_PASSWORD

buf_ptr Communications Server for Linux が要求された情報を戻すために使用するデータ・バッファへのポインター。

buf_size
指定したデータ・バッファのサイズ。

num_entries
ユーザー ID/ パスワードの対の最大数 (そのデータが戻されます)。ある範囲でなく、特定のエントリーに関するデータを要求するには、値 1 を指定します。できるだけ多くのエントリーを戻すにはゼロを指定します。この場合、Communications Server for Linux は指定したデータ・バッファに入る最大数のエントリーを戻します。

list_options
Communications Server for Linux がデータを戻すためのリスト内の開始位置。値は次のいずれかです。

AP_FIRST_IN_LIST

リスト内の最初のエントリーから開始します。

AP_LIST_INCLUSIVE

user_id パラメーターで指定したエントリーから開始します。

AP_LIST_FROM_NEXT

user_id パラメーターで指定したエントリーの直後のエントリーから開始します。

リストの順序と、アプリケーションでそのリストから特定のエントリーを入手する方法について詳しくは、47 ページの『QUERY_* verb のリスト・オプション』を参照してください。

user_id ユーザー ID。これは 10 バイトからなる タイプ AE の EBCDIC スtring で、名前が 10 文字に満たない場合は、右側にスペースを入れます。ユーザー ID は、*list_options* を AP_FIRST_IN_LIST に設定した場合には無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

buf_size
指定したバッファに戻された情報の長さ。

total_buf_size
要求したリスト情報をすべて戻すために必要だったバッファのサイズを示す戻り値。 *buf_size* より大きい値は、使用可能なエントリーがすべては戻されなかったことを示しています。

QUERY_USERID_PASSWORD

num_entries

データ・バッファー内に戻されたエントリーの数。

total_num_entries

使用可能なエントリーの合計数。*num_entries* より大きい値は、使用可能なエントリーがすべて戻されなかったことを示しています。

データ・バッファー内の各エントリーは、次のパラメーターで構成されています。

userid_info.overlay_size

戻された *userid_info* 構造体のサイズ。すなわち、データ・バッファーにおける次のエントリーの先頭までのオフセット。

アプリケーションが戻されたバッファーを調べて、各 *userid_info* 構造体を順番に検出する必要があるときには、この値を使用して次のデータ構造の正しいオフセットに移動しなければなりません。また、C の `sizeof()` 演算子は使用してはいけません。これは、戻されたオーバーレイのサイズは Communications Server for Linux の今後のリリースで大きくなる可能性があるためです。戻されたオーバーレイ・サイズを使用すると、アプリケーションは今後のリリースのもとでも、確実に処理を続けることができます。

userid_info.user_id

ユーザー ID。これは 10 バイトからなるタイプ AE の EBCDIC 文字ストリングで、右側に EBCDIC のスペースを入れます。

userid_info.password_chars.description

ユーザー ID およびパスワードを記述したヌルで終了するテキスト・ストリング (そのユーザー ID およびパスワードの定義で指定されたもの)。

userid_info.password_chars.profile_count

このユーザーに対して定義されたプロファイルの数。

userid_info.password_chars.password

`DEFINE_LU_LU_PASSWORD verb` で指定されたユーザーのパスワードの暗号化されたバージョン。これは 10 バイトからなるタイプ AE の EBCDIC 文字ストリングで、右側に EBCDIC のスペースを入れます。

userid_info.password_chars.profiles

ユーザーに関連したプロファイル。これらはいずれも 10 バイトからなるタイプ AE の EBCDIC ストリングで、右側に EBCDIC スペースを入れます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

`AP_PARAMETER_CHECK`

secondary_rc

値は次のいずれかです。

AP_INVALID_USERID

list_options パラメーターが、指定したユーザー ID から始まるすべ

てのエントリーをリストするために、AP_LIST_INCLUSIVE に設定されていますが、*user_id* パラメーターが無効でした。

AP_INVALID_LIST_OPTION

list_options パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

REGISTER_INDICATION_SINK

REGISTER_INDICATION_SINK は、特定のタイプの指示を受け取る NOF アプリケーションを登録します。Communications Server for Linux NOF 指示について詳しくは、777 ページの『第 4 章 NOF 指示』を参照してください。アプリケーションではその *opcode* パラメーターで必要な指示のタイプを指定します。1 つのアプリケーションで登録を複数回行って、複数の指示タイプを受け取ることができます。アプリケーションが指示を要求しておいたイベント (たとえば、アプリケーションのターゲット・ノードの構成変更または DLC の状況の変更) が発生するたびに、Communications Server for Linux はアプリケーションへ適切な指示メッセージを送信します。

あらゆるタイプの指示を受け取るよう登録されたアプリケーションには、ターゲット・ノードまたはファイルの状況変更を示す NOF_STATUS_INDICATION が戻される場合があります。詳しくは、817 ページの『NOF_STATUS_INDICATION』を参照してください。

この verb は、コールバック・ルーチンを含めた非同期 NOF API エントリー・ポイントを必ず使用して発行しなければなりません (NOF API エントリー・ポイントについて詳しくは、30 ページの『非同期エントリー・ポイント: nof_async』を参照してください)。Communications Server for Linux は、このコールバック・ルーチンを使用して、アプリケーションに要求された指示を戻します。

この verb は、次のように必要な指示のタイプに応じて、異なる宛先に対して発行できます。

- SNA ネットワーク・ファイル指示を登録する場合、**sna.net** ファイルを宛先にする必要があります。
- サーバー指示を受け取るために登録する場合、ターゲットは不要です。アプリケーションはヌル・ターゲット・ハンドルを指定する必要があります。
- ドメイン・リソースに関する構成ディレクティブを受け取るために登録する場合、ドメイン構成ファイルを宛先にする必要があります。
- ノード・リソースに関する構成ディレクティブのために登録する場合、またはその他の指示のために登録する場合、Communications Server for Linux ソフトウェアが稼働しているコンピューター上の実行中のノードまたは非活動ノードのどちらでも、宛先にすることができます。

VCB 構造体

```
typedef struct register_indication_sink
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;          /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    AP_UINT32      proc_id;         /* reserved                     */
    AP_UINT16      queue_id;        /* reserved                     */
    AP_UINT16      indication_opcode; /* opcode of indication to be sunk */
} REGISTER_INDICATION_SINK;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_REGISTER_INDICATION_SINK

indication_opcode

戻される指示の *opcode* パラメーター。指示が生成されるたびに、Communications Server for Linux は、この指示をアプリケーションのコールバック・ルーチンに送信します。

構成ディレクティブを受け取るようにするには、値 AP_CONFIG_INDICATION を指定します。REGISTER_INDICATION_SINK verb で指定したターゲット・ハンドルがドメイン構成ファイルを指定している場合は、この値はそのファイルが更新されるたびに指示を要求します。ターゲット・ハンドルがノードを指定している場合は、この値はノードの構成ファイルが更新されるたびに指示を要求します。

SNA ネットワーク・ファイル指示を受信するには、**sna.net** ファイルを識別するターゲット・ハンドルを使用して verb を発行し、値 AP_SNA_NET_INDICATION を指定します。この値は、ファイルが更新されるたびに指示を要求します。

他のすべての指示の場合は、必要な指示についての *opcode* 値を指定します。詳しくは、777 ページの『第 4 章 NOF 指示』にあるそれぞれの指示の説明を参照してください。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_OP_CODE

次のいずれかが起こったことを示します。

- *indication_opcode* パラメーターが Communications Server for Linux NOF API 指示のどの *opcode* にも一致しませんでした。
- 指定したターゲットには適用されない指示タイプが、*indication_opcode* パラメーターに指定されていました。ターゲット・ハンドルでドメイン構成ファイルを指定している場合は、構成ディレクティブのみが有効です。ターゲット・ハンドルで **sna.net** ファイルを指定している場合は、SNA ネットワーク・ファイル指示のみが有効です。ターゲット・ハンドルで、実行中のノードを指定している場合は、SNA ネットワーク・ファイル指示を除くすべての指示が有効です。

AP_DYNAMIC_LOAD_ALREADY_REGD

indication_opcode パラメーターが予約されている値に設定されました。

AP_SYNC_NOT_ALLOWED

アプリケーションが同期 NOF エントリー・ポイントを使用して REGISTER_INDICATION_SINK を発行しました。この verb では非同期エントリー・ポイントを使用する必要があります。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

指定した指示に関連した機能がローカル・ノードの構成でサポートされていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは指定した指示をサポートしていません。それぞれの指示に必要なサポートについては、777 ページの『第 4 章 NOF 指示』にあるそれぞれの指示の説明を参照してください。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

REMOVE_DLC_TRACE

この verb は、以前に ADD_DLC_TRACE を使用して指定された DLC 回線トレースを除去します。この verb を使用すると、現在トレース中のリソースのすべてのトレースを除去したり、現在トレース中のリソースから特定のメッセージのトレースを除去したり、すべての DLC 回線のトレースを除去したりすることができます。

VCB 構造体

```

typedef struct remove_dlc_trace
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;              /* reserved                  */
    unsigned char  format;               /* reserved                  */
    AP_UINT16      primary_rc;          /* primary return code      */
    AP_UINT32      secondary_rc;        /* secondary return code    */
    DLC_TRACE_FILTER filter;            /* resource to stop tracing */
} REMOVE_DLC_TRACE;

typedef struct dlc_trace_filter
{
    unsigned char  resource_type;        /* type of resource         */
    unsigned char  resource_name[8];    /* name of resource         */
    SNA_LFSID      lfsid;               /* session identifier       */
    unsigned char  message_type;        /* type of messages        */
} DLC_TRACE_FILTER;

typedef struct sna_lfsid
{
    union
    {
        AP_UINT16      session_id;
        struct
        {
            unsigned char  sidh;
            unsigned char  sidl;
        } s;
    } uu;
    AP_UINT16      odai;
} SNA_LFSID;

```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_REMOVE_DLC_TRACE

resource_type

除去または変更するトレース・エントリーのリソース・タイプ。値は次のいずれかです。

AP_ALL_DLC_TRACES

DLC トレース・オプションをすべて除去して、どのリソースもトレースされないようにします。このオプションを指定した場合は、この *verb* の残りのパラメーター (*resource_name* から *message_type* まで) は予約済みです。

AP_ALL_RESOURCES

すべての DLC、ポート、LS のトレースに使用するトレース・オプションを除去または変更します。DLC_TRACE エントリーが明示的に定義されているリソースは、引き続きトレースされます。

AP_DLC *resource_name* で指定した DLC と、この DLC を使用するすべてのポートおよび LS のトレースを除去または変更します。

AP_PORT

resource_name で指定したポートと、このポートを使用するすべての LS のトレースを除去または変更します。

AP_LS *resource_name* で指定した LS のトレースを除去または変更します。

AP_RTP *resource_name* で指定した RTP (高速トランスポート・プロトコル) 接続のトレースを除去または変更します。

AP_PORT_DEFINED_LS
resource_name で指定したポートと、その定義された LS のトレースを変更します。

AP_PORT_IMPLICIT_LS
resource_name で指定したポートと、その暗黙 LS のトレースを変更します。

resource_name

トレースを除去または変更する DLC、ポート、LS、または RTP 接続の名前。このパラメーターは、*resource_type* を AP_ALL_DLC_TRACES または AP_ALL_RESOURCES に設定した場合には予約済みです。

lfsid 指定した LS 上のセッションのローカル・フォーム・セッション ID。これは *resource_type* AP_LS の場合のみ有効で、このセッションについてのメッセージのみを除去することを示します。この構造体には次の 3 つの値が入っており、これらの値は QUERY_SESSION verb の SESSION_STATS セクションに戻されます。

lfsid.uu.s.sidh
セッション ID の上位バイト。

lfsid.uu.s.sidl
セッション ID の下位バイト。

lfsid.odai
起点宛先アサイナー・インディケータ。

message_type
指定したリソースまたはセッションをトレースするメッセージのタイプ。このパラメーターを AP_TRACE_ALL に設定するか、次の値の 1 つ以上 (論理 OR を使用して組み合わせる) を指定します。

AP_TRACE_XID
XID メッセージ

AP_TRACE_SC
セッション制御 RU

AP_TRACE_DFC
データ・フロー制御 RU

AP_TRACE_FMD
FMD メッセージ

AP_TRACE_SEGS
RH を含まない BBIU 以外のセグメント

AP_TRACE_CTL
MU と XID 以外のメッセージ

REMOVE_DLC_TRACE

AP_TRACE_NLP

(このメッセージ・タイプは現在は使用されていません)

AP_TRACE_NC

(このメッセージ・タイプは現在は使用されていません)

RTP 接続に対するトレースでは、値 AP_TRACE_XID、AP_TRACE_NLP、AP_TRACE_CTL は無視されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のいずれかを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_RESOURCE_TYPE

resource_type パラメーターに、無効な値が指定されました。

AP_INVALID_MESSAGE_TYPE

message_type パラメーターに、無効な値が指定されました。

AP_INVALID_DLC_NAME

resource_name で指定した DLC には、トレース・オプションが設定されていません。

AP_INVALID_PORT_NAME

resource_name で指定したポートには、トレース・オプションが設定されていません。

AP_INVALID_LS_NAME

resource_name で指定した LS には、トレース・オプションが設定されていません。

AP_INVALID_RTP_CONNECTION

resource_name パラメーターで指定した RTP 接続には、トレース・オプションが設定されていません。

AP_INVALID_LFSID_SPECIFIED

resource_name で指定した LS には、指定した LFSID のトレース・オプションが設定されていません。

AP_INVALID_FILTER_TYPE

message_type パラメーターが、指定したリソースに、現在トレースされていないメッセージ・タイプを指定しました。

AP_ALL_RESOURCES_NOT_DEFINED

resource_type パラメーターを AP_ALL_RESOURCES に設定しましたが、すべてのリソースのトレース・オプションに対して定義された DLC_TRACE エントリーがありません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

RESET_SESSION_LIMIT

RESET_SESSION_LIMIT verb は、特定の LU-LU モードの組み合わせについて、セッション限度をリセットするよう Communications Server for Linux に要求します。この verb の処理の結果、セッションが非活動化される場合があります。

VCB 構造体

```
typedef struct reset_session_limit
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;              /* reserved                      */
    unsigned char  format;               /* reserved                      */
    AP_UINT16      primary_rc;           /* primary return code          */
    AP_UINT32      secondary_rc;        /* secondary return code        */
    unsigned char  lu_name[8];           /* local LU name                */
    unsigned char  lu_alias[8];         /* local LU alias               */
    unsigned char  plu_alias[8];        /* partner LU alias             */
    unsigned char  fqplu_name[17];      /* fully qualified partner LU name*/
    unsigned char  reserv3;              /* reserved                      */
    unsigned char  mode_name[8];         /* mode name                    */
    unsigned char  mode_name_select;    /* select mode name             */
    unsigned char  set_negotiable;      /* set max negotiable limit to  */
                                        /* zero?                        */
    unsigned char  reserv4[8];          /* reserved                      */
    unsigned char  responsible;         /* who is responsible for      */
                                        /* deactivation                 */
    unsigned char  drain_source;        /* drain source                 */
    unsigned char  drain_target;        /* drain target                 */
    unsigned char  force;                /* force                        */
    AP_UINT32      sense_data;          /* sense data                   */
} RESET_SESSION_LIMIT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_RESET_SESSION_LIMIT

lu_name

Communications Server for Linux に対して定義されているローカル LU の LU 名。これは 8 バイトからなるタイプ A の EBCDIC スtringで、名

RESET_SESSION_LIMIT

前が 8 バイトに満たない場合は、右側にスペースを入れます。LU が LU 名でなく LU 別名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

lu_alias

Communications Server for Linux に対して定義されているローカル LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。これは、*lu_name* をゼロに設定した場合にのみ使用します。

CP へ関連付けられている LU (デフォルト LU) を示すには、*lu_name* と *lu_alias* の両方を 8 桁の 2 進ゼロに設定します。

plu_alias

パートナー LU の LU 別名。

これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。パートナー LU が LU 別名でなく完全修飾 LU 名で定義されていることを示すには、このパラメーターを 8 桁の 2 進ゼロに設定します。

fqplu_name

Communications Server for Linux に対して定義されているパートナー LU の完全修飾 LU 名。このパラメーターは、*plu_alias* フィールドをゼロに設定した場合のみ使用し、*plu_alias* を指定した場合には無視されます。

この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

mode_name

セッション限度がリセットされるモードの名前。このパラメーターは、*mode_name_select* を AP_ALL に設定した場合には無視されます。

これは 8 バイトの英数字からなる (英文字で始まる) タイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。

mode_name_select

指定した 1 つのモードでセッション限度をリセットするか、ローカル LU とパートナー LU 間のすべてのモードでセッション限度をリセットするかを選択します。値は次のいずれかです。

AP_ONE *mode_name* で指定したモードでセッション限度をリセットします。

AP_ALL すべてのモードでセッション限度をリセットします。

set_negotiable

この LU-LU モードを組み合わせで折衝可能な最大セッション限度をゼロにリセットするかどうかを指定します。(現行の限度は、モードに指定されている場合と、*initialize_session_limit* または *change_session_limit* により変更されている場合があります)。値は次のいずれかです。

AP_YES この LU-LU モードを組み合わせると折衝可能な最大セッション限度をゼロにリセットし、セッションは INITIALIZE_SESSION_LIMIT で変更するまで活動化できなくなります。

AP_NO 折衝可能な最大セッション限度は変更しません。

responsible

セッション限度をリセットしたあと、ソース (ローカル LU) とターゲット (パートナー LU) のどちらがセッションの非活動化を担当するかどうかを指定します。値は次のいずれかです。

AP_SOURCE

ローカル LU がセッションを非活動化します。

AP_TARGET

パートナー LU がセッションを非活動化します。

drain_source

セッションを非活動化する前にソース LU が待機状態のセッション要求を満たすかどうかを指定します。値は次のいずれかです。

AP_YES 待機状態のセッション要求は満たされます。

AP_NO 待機状態のセッション要求は満たされません。

drain_target

セッションを非活動化する前にターゲット LU が待機状態のセッション要求を満たすかどうかを指定します。値は次のいずれかです。

AP_YES 待機状態のセッション要求は満たされます。

AP_NO 待機状態のセッション要求は満たされません。

force

CNOS 折衝が失敗してもセッション限度をゼロに設定するかどうかを指定します。値は次のいずれかです。

AP_YES セッション限度はゼロに設定されます。

AP_NO CNOS 折衝が失敗した場合、セッション限度はゼロに設定されません。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

値は次のいずれかです。

AP_FORCED

CNOS 折衝が失敗してもセッション限度はゼロに設定されました。

AP_AS_NEGOTIATED

セッション限度が変更されましたが、パートナー LU によって 1 つ以上の値が折衝されました。

AP_AS_SPECIFIED

セッション限度は要求どおりに変更され、パートナー LU による折衝は行われませんでした。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_EXCEEDS_MAX_ALLOWED

Communications Server for Linux 内部エラーが発生しました。

AP_INVALID_LU_ALIAS

lu_alias パラメーターが、定義されているどのローカル LU 別名にも一致しませんでした。

AP_INVALID_LU_NAME

lu_name パラメーターが、定義されているどのローカル LU 名にも一致しませんでした。

AP_INVALID_MODE_NAME

mode_name パラメーターが、定義されているどのモード名にも一致しませんでした。

AP_INVALID_PLU_NAME

fqplu_name パラメーターが、定義されているどのパートナー LU 名にも一致しませんでした。

AP_INVALID_MODE_NAME_SELECT

mode_name_select パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_DRAIN_SOURCE

drain_source パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_DRAIN_TARGET

drain_target パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_FORCE

force パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_RESPONSIBLE

responsible パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_SET_NEGOTIABLE

set_negotiable パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

AP_MODE_RESET

この LU-LU モードの組み合わせについて、現在活動状態であるセッションはありません。RESET_SESSION_LIMIT ではなく INITIALIZE_SESSION_LIMIT を使用して限度を指定してください。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: セッションの割り振りエラー

セッション割り振りエラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_ALLOCATION_ERROR

secondary_rc

AP_ALLOCATION_FAILURE_NO_RETRY

訂正処置が必要な状態のために、セッションを割り振ることができませんでした。sense_data パラメーターとログに記録されたメッセージを検査して障害の理由を判別し、必要な処置を実行してください。エラー状態を訂正してからこの verb を再試行してください。

sense_data

割り振りの障害に関連した SNA センス・データ。

パラメーター: CNOS 処理エラー

エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_CONV_FAILURE_NO_RETRY

処置を必要とする状態 (構成の不一致またはセッション・プロトコル・エラーなど) のために、セッション限度を変更できませんでした。エラー状態について Communications Server for Linux ログ・ファイルを確認し、エラー状態を訂正してから、この verb を再試行してください。

primary_rc

RESET_SESSION_LIMIT

AP_CNOS_PARTNER_LU_REJECT

Communications Server for Linux がパートナーとのセッション限度の折衝に失敗したため、verb は失敗しました。ローカル LU とパートナー LU の両方で構成を検査してください。

secondary_rc

AP_CNOS_COMMAND_RACE_REJECT

セッションを活動化または非活動化したり、セッション限度を処理するために別の管理プログラム (または内部的に Communications Server for Linux ソフトウェア) が指定したモードでアクセスしていたため、verb が失敗しました。アプリケーションで verb を再試行してください。ただし、競争状態が解決されるように、タイムアウトのあとで再試行するようにしてください。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_BUFFER_AVAILABILITY

この verb は、Communications Server for Linux が常時使用できる STREAMS バッファの大きさを指定します。これにより、ノードで使用可能なバッファを有効利用することが可能であり、ユーザーは Linux コンピューター上の他のプロセス用として、バッファが使用可能であることを確認できます。

VCB 構造体

```
typedef struct set_buffer_availability
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
    AP_UINT32      buf_avail;     /* maximum buffer space available */
    unsigned char  reserv3[8];     /* reserved                  */
} SET_BUFFER_AVAILABILITY;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_SET_BUFFER_AVAILABILITY

buf_avail

使用可能な STREAMS バッファ・スペースの最大容量 (バイト単位)。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc
未使用。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_CENTRAL_LOGGING

この verb は、Communications Server for Linux ログ・メッセージをすべてのサーバーから中央ファイルに送信するか、各サーバー上の個々のファイルに送信するかを指定します。詳しくは、738 ページの『SET_LOG_FILE』を参照してください。

この verb は、中央ロガーとして現在機能しているノードに発行する必要があります。このノードへのアクセスについては、70 ページの『CONNECT_NODE』を参照してください。

VCB 構造体

```
typedef struct set_central_logging
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;              /* reserved                  */
    unsigned char  format;               /* reserved                  */
    AP_UINT16      primary_rc;           /* primary return code      */
    AP_UINT32      secondary_rc;         /* secondary return code    */
    unsigned char  enabled;              /* is central logging enabled? */
    unsigned char  reserv3[3];           /* reserved                  */
} SET_CENTRAL_LOGGING;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_SET_CENTRAL_LOGGING

enabled

中央ロギングが使用可能であるか、または使用不可であるかを指定します。値は次のいずれかです。

AP_YES 中央ロギングは使用可能です。すべてのログ・メッセージは、現在の中央ロガーであるノード上の単一のファイルに送信されます。

AP_NO 中央ロギングは使用不可です。各サーバーからのログ・メッセージは、そのサーバー上のファイル (SET_LOG_FILE verb を使用して指定される) に送信されます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

SET_CENTRAL_LOGGING

secondary_rc
未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_NOT_CENTRAL_LOGGER
この *verb* は、中央ロガーではないノードに発行されました。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_CS_TRACE

この *verb* は、Communications Server for Linux LAN 上のコンピューター間で送信されるデータのトレース・オプションを指定します。トレース・オプションについて詳しくは、「*Communications Server for Linux 管理ガイド*」を参照してください。

この *verb* は AIX または Linux クライアント上で実行中の NOF アプリケーションから発行することができます。NOF アプリケーションはユーザー ID *root* を使用するか、または *sys* グループ (AIX) または *sna* グループ (Linux) のメンバーであるユーザー ID を使用して実行しなければなりません。

この *verb* は、クライアントから発行しない場合、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct set_cs_trace
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  dest_sys[128];  /* node to which messages are traced */
    unsigned char  reserv4[4];     /* reserved                     */
    AP_UINT16      trace_flags;    /* trace flags                 */
    AP_UINT16      trace_direction; /* direction (send/rcv/both) to trace */
    unsigned char  reserv3[8];    /* reserved                     */
} SET_CS_TRACE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_SET_CS_TRACE

dest_sys

トレースを必要とするサーバー名。この名前は ASCII ストリングです。名前が 128 文字に満たない場合は、右側にスペースを入れます。

この verb の発行先となるコンピューター (NOF API 呼び出しの *target_handle* パラメーターで識別されます) と、LAN 上の別のサーバーとの間を流れるメッセージのトレースを管理するには、ここでそのサーバーの名前を指定します。LAN 上の他の複数のコンピューター間で流れるメッセージのトレースは変更されません。特に、同じターゲット・コンピューターと、2 つの異なる宛先サーバーとの間でトレースを活動化する場合は、2 つの SET_CS_TRACE verb を発行することができます。

サーバー名に . (ピリオド) 文字が含まれている場合、Communications Server for Linux はこれが完全修飾名であると想定します。ピリオドが含まれていない場合は、DNS ルックアップを実行してサーバー名を判別します。

この verb の発行先となるコンピューター (NOF API 呼び出しの *target_handle* パラメーターで識別されます) と、LAN 上の他のすべてのサーバーおよびクライアントとの間を流れるメッセージのトレースを管理するには、このパラメーターを 128 個の ASCII スペース文字に設定します。この verb で指定したオプションによって、特定のコンピューターへのトレースに関する以前の設定 (以前の verb の *dest_sys* によって指定された設定) が指定変更されます。

trace_flags

必要なトレースのタイプ。トレースをすべてオフにしたり、すべてのタイプのトレースをオンにするには、次の値のいずれかを指定します。

AP_NO_TRACE

トレースしません。

AP_ALL_TRACE

すべてのタイプをトレースします。

特定のメッセージ・タイプのトレースを活動化するには、次のリストの値の 1 つ以上を論理 OR 演算で組み合わせた値を選択します。

AP_CS_ADMIN_MSG

クライアント/サーバー・トポロジーに関する内部メッセージ

AP_CS_DATAGRAM

データグラム・メッセージ

AP_CS_DATA

データ・メッセージ

trace_direction

トレースが要求される方向を指定します。 *trace_flags* が AP_NO_TRACE に設定されている場合、このパラメーターは無視されます。値は次のいずれかです。

AP_CS_SEND

ターゲット・コンピューターから、*dest_sys* で定義したコンピューターに流れるメッセージをトレースします。

AP_CS_RECEIVE

dest_sys で定義したコンピューターからターゲット・コンピューターに流れるメッセージをトレースします。

AP_CS_BOTH

両方向に流れるメッセージをトレースします。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_NAME_NOT_FOUND

dest_sys パラメーターで指定したサーバーは存在していないか、開始されていません。

AP_LOCAL_SYSTEM

dest_sys パラメーターで指定したサーバーは、この *verb* の発行先のターゲット・ノードと同じです。

AP_INVALID_TRC_DIRECTION

trace_direction パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_TARGET

この *verb* はスタンドアロン・サーバーで発行されました。この *verb* はクライアント/サーバー・システムでのみ発行できます。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_GLOBAL_LOG_TYPE

この verb は、Communications Server for Linux がログ・ファイルに記録する情報のタイプを指定します。これにより、すべてのサーバー上で使用されるデフォルト値が指定されます。SET_LOG_TYPE を使用すると、特定のサーバー上でこれらのデフォルトを指定変更することができます。ログ・ファイルについての詳細は、738 ページの『SET_LOG_FILE』を参照してください。

Communications Server for Linux は、以下のタイプのイベントについてメッセージをログに記録します。

問題 ユーザーが感知できる、システムの劣化を引き起こす異常イベント (セッションの異常終了など)。

例外 システムを劣化させることがあっても、ユーザーが直接感知できない異常イベント (リソースの不足など)、またはシステムを劣化させることはなくても、あとで起こる例外または問題の原因を示すことがあるイベント (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 通常のイベント (セッションの開始など)。

問題メッセージおよび例外メッセージは、エラー・ログ・ファイルに記録されます。監査メッセージは監査ログ・ファイルに記録されます。問題メッセージは常に記録され、使用不可にできませんが、他の 2 つのタイプのメッセージは、それぞれ記録するかどうかを指定できます。2 つのファイル (監査とエラー) のそれぞれについて、簡略ログ (メッセージのテキストとメッセージ送信元の要約のみを含む) またはフル・ログ (メッセージの送信元、原因、および必要なすべての処置の詳細すべてを含む) のどちらを使用するかを指定できます。

この verb は、中央ロガーとして現在機能しているノードに発行する必要があります。詳しくは、70 ページの『CONNECT_NODE』を参照してください。

VCB 構造体

```
typedef struct set_global_log_type
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  audit;          /* audit logging on or off      */
    unsigned char  exception;      /* exception logging on or off  */
    unsigned char  succinct_audits; /* use succinct logging in audit file?*/
    unsigned char  succinct_errors; /* use succinct logging in error file?*/
    unsigned char  reserv3[4];     /* reserved                      */
} SET_GLOBAL_LOG_TYPE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_SET_GLOBAL_LOG_TYPE

audit 監査メッセージを記録するかどうかを指定します。値は次のいずれかです。

AP_YES 監査メッセージを記録します。

SET_GLOBAL_LOG_TYPE

AP_NO 監査メッセージを記録しません。

AP_LEAVE_UNCHANGED

監査ログを既存の定義の状態を変更しません。(Communications Server for Linux ソフトウェアの起動時の初期デフォルトでは、監査メッセージは記録されません。)

exception

例外メッセージを記録するかどうかを指定します。値は次のいずれかです。

AP_YES 例外メッセージを記録します。

AP_NO 例外メッセージを記録しません。

AP_LEAVE_UNCHANGED

例外ログを既存の定義の状態を変更しません。(Communications Server for Linux ソフトウェアの起動時の初期デフォルトでは、例外メッセージは記録されます。)

succinct_audits

監査ログ・ファイルで簡略ログまたはフル・ログのどちらを使用するかを指定します。値は次のいずれかです。

AP_YES 簡略ログ: ログ・ファイル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、ログ・タイプ、システム名など)、メッセージ・テキスト・ストリング、およびパラメーターが入ります。ログの原因と必要な処置の詳細を入手するには、`snahelp` コーティリティーを使用してください。

AP_NO フル・ログ: ログ・ファイル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、ログの原因と必要な処置に関する追加情報が入っています。

AP_LEAVE_UNCHANGED

前の `SET_GLOBAL_LOG_TYPE verb` でこのパラメーターに指定した値 (簡略ログまたはフル・ログ) を使用します。

`SET_GLOBAL_LOG_TYPE verb` を発行する前に、初期デフォルトでは、簡略ログを使用します。

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ついて簡略ログまたはフル・ログのいずれを選択するかは、中央ロガーとして機能しているサーバー上でのこのパラメーターの設定によって決定されます。この設定は、デフォルトを指定変更するために、`SET_GLOBAL_LOG_TYPE verb` から、またはそのサーバーに発行された `SET_LOG_TYPE verb` から生じます。

succinct_errors

エラー・ログ・ファイルで簡略ログまたはフル・ログのどちらを使用するかを指定します。これは例外ログと問題ログの両方に適用されます。指定できる値とその意味は、*succinct_audits* パラメーターの場合と同じです。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

```
secondary_rc
    未使用。
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_NOT_CENTRAL_LOGGER
この `verb` は、中央ロガーではないノードに発行されました。

AP_INVALID_SUCCINCT_SETTING
`succinct_audits` パラメーターまたは `succinct_errors` パラメーターが、有効な値に設定されませんでした。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_KERNEL_MEMORY_LIMIT

この `verb` は、Communications Server for Linux が常時使用できるカーネル・メモリーの大きさの限度を指定します。これにより、Linux コンピューター上の他のプロセスにも使用可能なメモリーを確保できるようになります。

Communications Server for Linux ソフトウェアの始動時にカーネル・メモリーの限度を指定することもできます (詳しくは、「*Communications Server for Linux* 管理ガイド」を参照してください)。Communications Server for Linux ソフトウェアの始動時に指定した限度がある場合、この `verb` はその限度を指定変更します。

VCB 構造体

```
typedef struct set_kernel_memory_limit
{
    AP_UINT16    opcode;           /* verb operation code          */
    unsigned char reserv2;        /* reserved                      */
    unsigned char format;        /* reserved                      */
    AP_UINT16    primary_rc;      /* primary return code          */
    AP_UINT32    secondary_rc;    /* secondary return code        */
    AP_UINT32    limit;           /* kernel memory limit, 0 => no limit */
    unsigned char reserv3[8];     /* Reserved                      */
} SET_KERNEL_MEMORY_LIMIT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

SET_KERNEL_MEMORY_LIMIT

opcode AP_SET_KERNEL_MEMORY_LIMIT

limit Communications Server for Linux が任意の時点で使用するカーネル・メモリの最大容量 (バイト単位)。 Communications Server for Linux のコンポーネントの 1 つがカーネル・メモリーを割り振ろうとした結果、現在、 Communications Server for Linux のコンポーネントに割り振られているメモリーの合計容量がこの限度を超えると、その割り振りの試みは失敗します。

前の SET_KERNEL_MEMORY_LIMIT verb で設定した限度を除去するには、ゼロを指定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_LOG_FILE

この verb は、Communications Server for Linux がログ・メッセージの記録に使用するファイルを管理します。この verb により、次のことが可能になります。

- ログ・メッセージ (監査、エラー、使用量のログ) の記録に使用するファイルと、(ログ情報をコピーする) バックアップ・ファイルを指定します。
- ログ・ファイルの最大サイズを指定します (ログ・ファイルがこのサイズに達すると、Communications Server for Linux はログ情報をバックアップ・ファイルにコピーし、ログ・ファイルをリセットします)。
- ログ・ファイルの現在の内容をバックアップ・ファイルにコピーし、オプションで現行ファイルを削除します。

監査ログ・メッセージとエラー・ログ・メッセージを別々のファイルに記録することができ、また両方のタイプのメッセージを同じファイルに記録することもできます。

SET_CENTRAL_LOGGING で定義するとき中央ロギングを使用している場合、この verb は、中央ロガーとして現在機能しているノードに発行する必要があります。それ以外の場合は、異なるログ・ファイルを各ノードで指定するために、この verb を各ノードへ個別に発行することができます。

この verb は AIX または Linux クライアント上で実行中の NOF アプリケーションから発行することができます。NOF アプリケーションはユーザー ID root を使用するか、または sys グループ (AIX) または sna グループ (Linux) のメンバーであるユーザー ID を使用して実行しなければなりません。

VCB 構造体

```
typedef struct set_log_file
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;              /* reserved                  */
    unsigned char  format;              /* reserved                  */
    AP_UINT16      primary_rc;          /* primary return code      */
    AP_UINT32      secondary_rc;        /* secondary return code    */
    unsigned char  log_file_type;       /* type of log file        */
    unsigned char  action;              /* reset and/or backup existing */
    unsigned char  file_name[81];       /* file name                 */
    unsigned char  backup_file_name[81]; /* backup file               */
    AP_UINT32      file_size;           /* log file size            */
    unsigned char  succinct;            /* reserved                  */
    unsigned char  reserv3[3];          /* reserved                  */
} SET_LOG_FILE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_SET_LOG_FILE

log_file_type

管理するログ・ファイルのタイプ。値は次のいずれかです。

AP_AUDIT_FILE

監査ログ・ファイル (監査メッセージのみ)。

AP_ERROR_FILE

エラー・ログ・ファイル (問題メッセージおよび例外メッセージ)。

AP_USAGE_FILE

使用量ログ・ファイル (Communications Server for Linux リソースの現在とピーク時の使用量に関する情報)。

監査とエラーの両方のメッセージを同じファイルに記録するには、2 つの SET_LOG_FILE verb を同じファイル名で発行し、一方の verb に AP_AUDIT_FILE を、もう一方の verb に AP_ERROR_FILE を指定します。

action ログ・ファイルに対する処置。値は次のいずれかです。

AP_NO_FILE_ACTION

file_name パラメーターで指定したファイルをログ・ファイルとして使用し、*backup_file_name* パラメーターで指定したファイルをバックアップ・ファイルとして使用します。verb が正常に完了すると、*log_file_type* で定義されたタイプのログ・メッセージは、すべて新しいログ・ファイルに書き込まれます。この verb を発行する前に使用していたファイルがあっても、そのファイルは変更されません。

AP_DELETE_FILE

現行ログ・ファイルの内容を削除します。

AP_BACKUP_FILE

現行ログ・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、現行ファイルの内容を削除します。

file_name

新しいログ・ファイルの名前。

診断ファイル用のデフォルト・ディレクトリー **/var/opt/ibm/sna** にファイルを作成するには、パスを含めずにファイル名を指定します。別のディレクトリーにファイルを作成するには、絶対パスまたはデフォルト・ディレクトリーに対する相対パスを指定します。パスを含める場合は、この **verb** を入力するどのコンピューターでも有効なパス (アプリケーションの作業ディレクトリーに対する相対パスか、絶対パスのいずれか) である必要があります。

このパラメーターは 1 ~ 80 文字の ASCII スtringで、最後にヌル文字 (2 進ゼロ) が続きます。前の **SET_LOG_FILE verb** で指定したファイルへのログ記録を引き続き行うためには、ヌル・Stringを指定します。

backup_file_name

バックアップ・ログ・ファイルの名前。ログ・ファイルが下記の *file_size* で指定したサイズに達すると、Communications Server for Linux は、現在の内容をバックアップ・ファイルにコピーしてから、ログ・ファイルをクリアします。また、上記のアクション・パラメーターを使用すれば、いつでもバックアップを要求できます。

診断ファイル用のデフォルト・ディレクトリー **/var/opt/ibm/sna** にファイルを作成するには、パスを含めずにファイル名を指定します。別のディレクトリーにファイルを作成するには、絶対パスまたはデフォルト・ディレクトリーに対する相対パスを指定します。パスを含める場合は、この **verb** を入力するどのコンピューターでも有効なパス (アプリケーションの作業ディレクトリーに対する相対パスか、絶対パスのいずれか) である必要があります。

このパラメーターは 1 ~ 80 文字の ASCII Stringで、最後にヌル文字 (2 進ゼロ) が続きます。前の **SET_LOG_FILE verb** で指定したバックアップ・ファイルを引き続き使用するためには、ヌル・Stringを指定します。

file_size

log_file_type で指定したログ・ファイルの最大サイズ。ファイルへのメッセージの書き込みでファイル・サイズがこの限度を超えると、Communications Server for Linux は、ログ・ファイルの現在の内容をバックアップ・ログ・ファイルへコピーしてから、ログ・ファイルをクリアします。したがって、ログ・ファイルが占有するディスク・スペースの最大容量は *file_size* の値の約 2 倍になります。

前の **SET_LOG_FILE verb** で指定したファイル・サイズを引き続き使用するためには、このパラメーターをゼロに設定します。SET_LOG_FILE verb を発行する前の初期デフォルト値は、1,000,000 バイトです。ゼロの値は、「既存のファイル・サイズを引き続き使用する」および「制限なし」を示します。

監査ログ・ファイルとエラー・ログ・ファイルのサイズは、より大きいシステムで生成されるログ情報量に対処するために、Communications Server for Linux クライアント/サーバー・ネットワークのサイズに応じて大きくする必

要がある場合もあります。特に、以下のような状況では、ログ・ファイル・サイズの拡大を考慮する必要があります。

- 多数のクライアントまたはユーザーが存在 (通信リンクで 1 つの障害が起こっても、セッション障害に関するサーバー上では大量のログが生成されることがあります。)
- 監査ログならびに例外ログの活動化
- 分散ロギングではなく中央ロギングの使用
- 簡略ログではなく、フル・ログの使用

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_FILE_ACTION
action パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_FILE_TYPE
log_file_type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_LOG_TYPE

この verb は、Communications Server for Linux が特定サーバーのログ・ファイルに記録する情報のタイプを指定します。この verb を使用すると、SET_GLOBAL_LOG_TYPE で指定したデフォルト設定を指定変更するか、この指定変更を取り消して、このサーバーがデフォルト設定を使用するように設定し直すこ

とができます。ログ・ファイルについての詳細は、738 ページの『SET_LOG_FILE』を参照してください。

この verb は AIX または Linux クライアント上で実行中の NOF アプリケーションから発行することができます。NOF アプリケーションはユーザー ID root を使用するか、または sys グループ (AIX) または sna グループ (Linux) のメンバーであるユーザー ID を使用して実行しなければなりません。

問題 ユーザーが感知できる、システムの劣化を引き起こす異常イベント (セッションの異常終了など)。

例外 システムを劣化させることがあっても、ユーザーが直接感知できない異常イベント (リソースの不足など)、またはシステムを劣化させることはなくても、あとで起こる例外または問題の原因を示すことがあるイベント (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 通常のイベント (セッションの開始など)。

問題メッセージおよび例外メッセージは、エラー・ログ・ファイルに記録されます。監査メッセージは監査ログ・ファイルに記録されます。問題メッセージは常に記録され、使用不可にできませんが、他の 2 つのタイプのメッセージは、それぞれ記録するかどうかを指定できます。2 つのファイル (監査とエラー) のそれぞれについて、簡略ログ (メッセージのテキストとメッセージ送信元の要約のみを含む) またはフル・ログ (メッセージの送信元、原因、および必要なすべての処置の詳細すべてを含む) のどちらを使用するかを指定できます。

VCB 構造体

```
typedef struct set_log_type
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code        */
    unsigned char  override;      /* override global defaults?   */
    unsigned char  audit;         /* audit logging on or off     */
    unsigned char  exception;     /* exception logging on or off */
    unsigned char  succinct_audits; /* use succinct logging in audit file? */
    unsigned char  succinct_errors; /* use succinct logging in error file? */
    unsigned char  reserv3[3];     /* reserved                     */
} SET_LOG_TYPE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_SET_LOG_TYPE

override

この verb を使用して、SET_GLOBAL_LOG_TYPE で指定したグローバル・ログ・タイプを指定変更するか、またはこれらのデフォルトを使用するように設定し直すかを指定します。値は次のいずれかです。

AP_YES グローバル・ログ・タイプを指定変更します。このサーバー上で使用されるログ・タイプは、下記の *audit* および *exception* パラメー

ターで指定します。簡略ログとフル・ログのどちらを選択するかは、下記の *succinct_** パラメーターで指定します。

AP_NO グローバル・ログ・タイプを使用するように設定し直します。以下の *audit*、*exception*、および *succinct_** パラメーターは無視されません。

audit 監査メッセージを記録するかどうかを指定します。値は次のいずれかです。

AP_YES 監査メッセージを記録します。

AP_NO 監査メッセージを記録しません。

AP_LEAVE_UNCHANGED

監査ログを既存の定義の状態に変更しません。

exception

このサーバーに例外メッセージを記録するかどうかを指定します。値は次のいずれかです。

AP_YES 例外メッセージを記録します。

AP_NO 例外メッセージを記録しません。

AP_LEAVE_UNCHANGED

例外ログを既存の定義の状態に変更しません。

succinct_audits

このサーバーの監査ログ・ファイルで簡略ログまたはフル・ログのどちらを使用するかを指定します。値は次のいずれかです。

AP_YES 簡略ログ: ログ・ファイル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、ログ・タイプ、システム名など)、メッセージ・テキスト・ストリング、およびパラメーターが入ります。ログの原因と必要な処置の詳細を入手するには、*snahelp* コーティリティーを使用してください。

AP_NO フル・ログ: ログ・ファイル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、ログの原因と必要な処置に関する追加情報が入っています。

AP_LEAVE_UNCHANGED

簡略ログまたはフル・ログを既存の定義の状態に変更しません。

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ついて簡略ログまたはフル・ログのいずれを選択するかは、中央ロガーとして機能しているサーバー上でのこのパラメーターの設定によって決定されます。この設定は、デフォルトを指定変更するために、

SET_GLOBAL_LOG_TYPE verb から、またはそのサーバーに発行された **SET_LOG_TYPE verb** から生じます。

succinct_errors

このサーバーのエラー・ログ・ファイルで簡略ログまたはフル・ログのどちらを使用するかを指定します。これは例外ログと問題ログの両方に適用されます。指定できる値とその意味は、*succinct_audits* パラメーターの場合と同じです。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
AP_INVALID_SUCCINCT_SETTING
succinct_audits パラメーターまたは *succinct_errors* パラメーターが、有効な値に設定されませんでした。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_PROCESSING_MODE

この verb は、NOF アプリケーションがターゲット・ノード、構成ファイル、または SNA ネットワーク・データ・ファイルとどのように相互作用するか、つまり、アプリケーションが読み取り専用アクセスまたは読み取り/書き込みアクセスのいずれを使用するか、さらにドメイン構成ファイルに対して他のアプリケーションがアクセスできないように排他アクセスを使用するかどうかを指定します。

この verb はサーバーで実行中の NOF アプリケーションにのみ適用されます。クライアントで実行中のアプリケーションに使用可能な処理モードは、読み取り専用モード (デフォルト) のみです。このモードでは、アプリケーションは QUERY_* verb を発行できますが、リソースを定義、開始、または停止することができません。クライアント・アプリケーションは SET_PROCESSING_MODE を使用して他のモードを選択することができません。

ターゲット・ノードまたはファイルは NOF API コールの *target_handle* パラメーターで指定されます。アプリケーションはこのパラメーターを CONNECT_NODE verb (ノードの場合) または OPEN_FILE verb (ファイルの場合) から取得します。これらのパラメーターの使用方法について詳しくは、28 ページの『AIX または Linux の NOF API エントリー・ポイント』を参照してください。

この verb は、ドメイン構成ファイル、**sna.net** ファイル、または実行中のノードに対して発行することができます。この verb で設定できる有効な処理モードは、ターゲット・タイプによって異なります。

VCB 構造体

```
typedef struct set_processing_mode
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code        */
    unsigned char  mode;           /* new mode to be set for this handle */
    AP_UINT16      reserv1;       /* reserved                     */
} SET_PROCESSING_MODE;
```

指定パラメーター

opcode AP_SET_PROCESSING_MODE

mode このターゲット・ハンドルに要求されたモード。このターゲット・ハンドルを使用して前に発行された *verb* がまだ未処理の間は、モードは変更できません。値は次のいずれかです。

AP_MODE_READ_ONLY

読み取り専用モード: アプリケーションは、構成の変更を伴わない `QUERY_* verb` のみを使用します。このオプションは、ターゲットがファイルの場合でもノードの場合でも使用できます。

AP_MODE_READ_WRITE

読み取り/書き込みモード: このアプリケーションは任意の `NOF API verb` で使用できます。このオプションは、ターゲットがファイルの場合でもノードの場合でも使用できます。

AP_MODE_COMMIT

コミット・モード: アプリケーションはターゲット・ファイルに対する排他的な読み取り/書き込みアクセスを使用するため、他のアプリケーションは、このアプリケーションがファイルを解放するまでこのファイルにアクセスできません。このオプションは、ターゲットがドメイン構成ファイルの場合のみ使用できます。

このモードは、ファイルに対して一連の結合された *verb* (関連するコンポーネントに対する一連の `DEFINE verb` など) を発行する場合に使用します。アプリケーションは、ファイルを解放して他の `NOF API` アプリケーションまたは `Communications Server for Linux` のコンポーネントがアクセスできるようにするために、一連の *verb* をできるだけ早く完了し、その処理モードを他のオプションの 1 つにリセットする必要があります。

注: ファイルへの読み取り/書き込みアクセス権限またはコミット・アクセス権限を取得するには、`SNA` 管理者グループ `sna` のメンバーであるユーザー `ID` を使用して (または `root` 使用して)、`NOF` アプリケーションを実行している必要があります。ユーザー `ID` がこのグループのメンバーでないか、または `root` でない場合、有効な処理モードは `AP_MODE_READ_ONLY` のみです。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_PROC_MODE
mode パラメーターが、有効な値に設定されていませんでした。

AP_INVALID_TARGET_MODE
mode パラメーターは、選択したターゲットに対して有効ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_STATE_CHECK

secondary_rc
値は次のいずれかです。

AP_FILE_UNAVAILABLE
アプリケーションはコミット・モードを指定しましたが、必要な構成ファイルへの排他的なアクセスを取得できませんでした。これはおそらく、別のアプリケーションがコミット・モードでそのファイルにアクセスしているためです。

AP_VERB_IN_PROGRESS
指定したターゲット・ハンドルに対して以前に発行された verb が未処理であるために、このターゲット・ハンドルの処理モードを変更することができません。ターゲット・ハンドルに対するすべての verb が正常に終了してから、処理モードの変更を試みる必要があります。

AP_NOT_AUTHORIZED

NOF アプリケーションはクライアントで実行されているため、または SNA 管理者グループ `sna` のメンバーでないユーザー ID を使用して実行されているため、ファイルへの読み取り/書き込みアクセスを行うことができません。ユーザー ID がこのグループのメンバーでない場合、有効な処理モードは `AP_MODE_READ_ONLY` のみです。

AP_NOT_MASTER

ターゲット・ハンドルはマスター・サーバーとして機能しないバックアップ・サーバー上のファイル (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) を指定しているため、処理モードを `AP_MODE_READ_WRITE` または `AP_MODE_COMMIT` に変更することはできません。実行構成ファイルの変更は、マスター上のこのファイルのコピーに対してのみ行うことができます (したがって、変更は他のサーバーに配信されます)。このファイルの他のコピーにアクセスできるのは、読み取り専用モードの場合のみです。アプリケーションで読み取り/書き込みモードまたはコミット・モードを使用する場合は、このターゲット・ハンドルに対して `CLOSE_FILE` を発行してから、`OPEN_FILE` を再発行して、新しいマスター・サーバー上のファイルにアクセスする必要があります。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、`AP_STATE_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_TN_SERVER_TRACE

この verb は、Communications Server for Linux TN サーバー・コンポーネントのトレース・オプションを指定します。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct set_tn_server_trace
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;         /* secondary return code */
    AP_UINT16      trace_flags;          /* trace flags */
    unsigned char  reserv3[6];           /* reserved */
} SET_TN_SERVER_TRACE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

```
opcode AP_SET_TN_SERVER_TRACE
```

SET_TN_SERVER_TRACE

trace_flags

必要なトレースのタイプ。トレースをすべてオフにしたり、すべてのタイプのトレースをオンにするには、次の値のいずれかを指定します。

AP_TN_SERVER_NO_TRACE

トレースしません。

AP_TN_SERVER_ALL_TRACE

すべてのタイプをトレースします。

特定のメッセージ・タイプのトレースを活動化するには、次のリストの値の 1 つ以上を論理 OR 演算で組み合わせた値を選択します。

AP_TN_SERVER_TRC_TCP

TCP/IP インターフェース・トレース: TN サーバーと TN3270 クライアント間のメッセージ

AP_TN_SERVER_TRC_FM

ノード・インターフェース・トレース: 内部制御メッセージ、および TN サーバーと TN3270 クライアント間のメッセージ (内部フォーマット)

AP_TN_SERVER_TRC_CFG

構成メッセージ・トレース: TN サーバーの構成に関連するメッセージ

AP_TN_SERVER_TRC_NOF

内部の Node Operator Facility (NOF) トレース: TN サーバーによる NOF トレース要求。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_TRACE_FILE

この *verb* は、Communications Server for Linux がトレース・データの記録に使用するファイルの名前を指定します。

同じファイル・タイプの新しいファイルを指定して 2 回目の SET_TRACE_FILE *verb* を発行すると、それ以後のトレース情報はすべて新しいファイルに書き込まれます。既存のファイルは除去されませんが、それ以降情報は書き込まれなくなります。同じトレース・ファイルに対して 2 回目の SET_TRACE_FILE *verb* を発行す

ると、そのファイルはリセットされます (2 回目の `verb` より前にファイルに書き込まれていたトレース情報は廃棄されます)。

この `verb` は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct set_trace_file
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;              /* reserved */
    AP_UINT16      primary_rc;          /* primary return code */
    AP_UINT32      secondary_rc;        /* secondary return code */
    unsigned char  trace_file_type;     /* type of trace file */
    unsigned char  dual_files;          /* dual trace files */
    AP_UINT32      trace_file_size;     /* trace file size */
    unsigned char  reserv3[4];          /* reserved */
    unsigned char  file_name[81];       /* file name */
    unsigned char  file_name_2[81];     /* second file name */
} SET_TRACE_FILE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_SET_TRACE_FILE

trace_file_type

トレース・ファイルのタイプ。値は次のいずれかです。

AP_CS_TRACE

指定したコンピューターとその他のノードの間で Communications Server for Linux LAN を介して転送されるデータ (SET_CS_TRACE `verb` により活動化されたもの) のトレースを入れるファイル。

AP_TN_SERVER_TRACE

Communications Server for Linux TN サーバー・コンポーネントのトレースを入れるファイル。

AP_IPS_TRACE

指定したノードのカーネル・コンポーネントのトレースを入れるファイル (SET_TRACE_TYPE `verb` または ADD_DLC_TRACE `verb` により活動化されたもの)。

dual_files

1 つのファイルにトレースするか 2 つのファイルにトレースするかを指定します。値は次のいずれかです。

AP_YES 2 つのファイルへのトレース。最初のファイルが *trace_file_size* で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、2 番目のファイルへトレースが継続されます。次に、2 番目のファイルが *trace_file_size* で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、最初のファイルへトレースが継続されます。これにより、余分なディスク・スペースを使用しないで長期にわたってトレースを継続できます。必要な最大スペースは *trace_file_size* の値の約 2 倍です。

SET_TRACE_FILE

AP_NO 1 つのファイルへのトレース。

AP_LEAVE_UNCHANGED

dual_files の設定は、既存の定義のままにし、変更しません。

(Communications Server for Linux ソフトウェアの起動時の初期デフォルトでは、2 つのファイルを使用します。)

trace_file_size

トレース・ファイルの最大サイズ (バイト単位)。既存のファイル・サイズ定義を引き続き使用するには、ゼロを指定します。

dual_files を **AP_YES** に設定した場合は、現行ファイルがこのサイズに達すると、2 つのファイル間でトレースが切り替わります。*dual_files* を **AP_NO** に設定した場合は、このパラメーターは無視されます。ファイル・サイズに制限はありません。

トレース・ファイルのサイズは、より大きいシステムで生成されるトレース情報量に対処するために、Communications Server for Linux クライアント/サーバー・ネットワークのサイズに応じて大きくする必要がある場合もあります。特に、多数のクライアントまたはユーザーがサーバーにアクセスすることを考慮して、サーバーのトレース・ファイル・サイズを大きくすることを検討してください。

file_name

トレース・ファイルの名前。*dual_files* を **AP_YES** に設定した場合には、最初のトレース・ファイルの名前。以前の **SET_TRACE_FILE verb** で指定したファイル名を引き続き使用するためには、このパラメーターをヌル・ストリングに設定します。

診断ファイル用のデフォルト・ディレクトリー **/var/opt/ibm/sna** にファイルを作成するには、パスを含めずにファイル名を指定します。別のディレクトリーにファイルを作成するには、絶対パスまたはデフォルト・ディレクトリーに対する相対パスを指定します。パスを含める場合は、この **verb** を入力するどのコンピューターでも有効なパス (アプリケーションの作業ディレクトリーに対する相対パスか、絶対パスのいずれか) である必要があります。

このパラメーターは 1 ~ 80 文字の ASCII ストリングで、最後にヌル文字 (2 進ゼロ) が続きます。

file_name_2

2 番目のトレース・ファイルの名前。このパラメーターは *dual_files* を **AP_YES** に設定した場合のみ使用されます。以前の *set_trace_file verb* で指定したファイル名を引き続き使用するためには、このパラメーターをヌル・ストリングに設定します。

診断ファイル用のデフォルト・ディレクトリー **/var/opt/ibm/sna** にファイルを作成するには、パスを含めずにファイル名を指定します。別のディレクトリーにファイルを作成するには、絶対パスまたはデフォルト・ディレクトリーに対する相対パスを指定します。パスを含める場合は、この **verb** を入力するどのコンピューターでも有効なパス (アプリケーションの作業ディレクトリーに対する相対パスか、絶対パスのいずれか) である必要があります。

このパラメーターは 1 ～ 80 文字の ASCII スtring で、最後にヌル文字 (2 進ゼロ) が続きます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_FILE_NAME

file_name パラメーターまたは *file_name_2* パラメーターが有効な Linux ファイル名に設定されていなかったか、単一トレース・ファイルから二重トレース・ファイルへの変更の際に *file_name_2* が指定されていませんでした。

AP_INVALID_FILE_TYPE

trace_file_type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

SET_TRACE_TYPE

この verb は、Communications Server for Linux カーネル・コンポーネントのトレース・オプションを指定します。この verb を使用すると、すべてのインターフェースのトレースの状態 (オンまたはオフ) を指定でき、また特定のインターフェースのトレースをオンまたはオフにする (他のインターフェースでのトレースは変更しない) ことができます。トレース・オプションについて詳しくは、「*Communications Server for Linux* 管理ガイド」を参照してください。

SET_TRACE_TYPE

DLC 回線トレースを制御するには、ADD_DLC_TRACE verb を使用します。この verb で指定された切り捨て長さは DLC トレースにも適用されますが、この verb のトレース・オプションは DLC トレースに適用されません。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct set_trace_type
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
    AP_UINT16      trace_flags;     /* trace flags              */
    AP_UINT32      truncation_length; /* truncate each msg to this size */
    unsigned char  init_flags;      /* TRUE if initializing flags */
    unsigned char  set_flags;       /* TRUE if setting flags    */
                                     /* FALSE if unsetting flags  */
    unsigned char  set_internal;    /* reserved                  */
    AP_UINT16      internal_level;  /* reserved                  */
    AP_UINT32      api_flags;       /* api trace flags          */
} SET_TRACE_TYPE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_SET_TRACE_TYPE

trace_flags

必要なトレースのタイプ。トレースをすべてオフにしたり、すべてのタイプのトレースをオンにするには、次の値のいずれかを指定します。

AP_NO_TRACE

トレースしません。

AP_ALL_TRACE

すべてのタイプをトレースします。

特定のインターフェースのトレースを制御するには、次のリストの値の 1 つ以上を論理 OR 演算で組み合わせた値を選択します。これらのトレース・タイプについて詳しくは、754 ページの『トレース・タイプ』を参照してください。

init_flags を AP_YES に設定した場合は、トレースを活動状態にするインターフェースに対応する値を選択し、トレースを非活動状態にするインターフェースに対応する値は選択しないでください。*init_flags* を AP_NO に設定した場合は、トレースの状態を変更するインターフェースに対応する値を選択してください。

AP_APPC_MSG

APPC メッセージ

AP_LUA_MSG

LUA メッセージ

AP_NOF_MSG

NOF メッセージ

AP_MS_MSG

MS メッセージ

AP_LLC2_MSG

LLC2 メッセージ

AP_LLI_MSG

LLI メッセージ

AP_MAC_MSG

MAC メッセージ

AP_SDLC_MSG

SDLC メッセージ (このオプションにより SDLC 回線トレースの追加の詳細情報も提供されることに注意してください)

AP_NLI_MSG

NLI メッセージ

AP_IPDL_MSG

Enterprise Extender (HPR/IP) メッセージ

AP_DLC_MSG

ノードから DLC へのメッセージ

AP_NODE_MSG

ノード・メッセージ

AP_SLIM_MSG

マスター・サーバーとバックアップ・サーバーの間で送信されるメッセージ。

AP_DATAGRAM

データグラム・メッセージ

truncation_length

それぞれのメッセージについて、トレース・ファイルに書き込まれる情報の最大長 (バイト単位) を指定します。この値は少なくとも 256 でなければなりません。

トレース・メッセージがこのパラメーターで指定した長さを超える場合、Communications Server for Linux はメッセージの先頭部分のみをトレース・ファイルに書き込み、*truncation_length* を超えるデータを廃棄します。これにより、各メッセージの最も重要な情報のみを記録し、長いメッセージでファイルがいっぱいにならないようにします。

切り捨てなし (各メッセージのデータをすべてこのファイルに書き込む) を指定するには、このパラメーターをゼロに設定します。

init_flags

トレースを初期化する (すべてのインターフェースのトレース状態を定義する) か、1 つ以上のインターフェースのトレース状態を変更する (他は変更しない) かを指定します。値は次のいずれかです。

AP_YES トレースを初期化します。*trace_flags* パラメーターは、すべてのインターフェースで必要なトレースの状態を定義します。

SET_TRACE_TYPE

AP_NO トレースを変更します。*trace_flags* パラメーターは、トレースを活動化または非活動化するインターフェースを定義します。他のインターフェースには影響しません。

set_flags

init_flags を AP_NO に設定した場合は、このパラメーターは要求したインターフェースでトレースを活動化するか非活動化するかを指定します。値は次のいずれかです。

AP_YES トレースは *trace_flags* パラメーターで指定したインターフェースで活動化されます。

AP_NO トレースは *trace_flags* パラメーターで指定したインターフェースで非活動化されます。

このパラメーターは、*init_flags* を AP_YES に設定した場合には無視されません。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

secondary_rc

未使用。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_TRUNC_LEN

truncation_length パラメーターが 256 バイトより小さい値を指定しました。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

トレース・タイプ

755 ページの図 2 に Communications Server for Linux の全体の構造を示します。Communications Server for Linux のコンポーネント間の特定のインターフェースを介するデータ転送に関連するカーネル空間の各トレース・タイプは、ダイアグラム

ではトレースが行われるインターフェースに示されています。

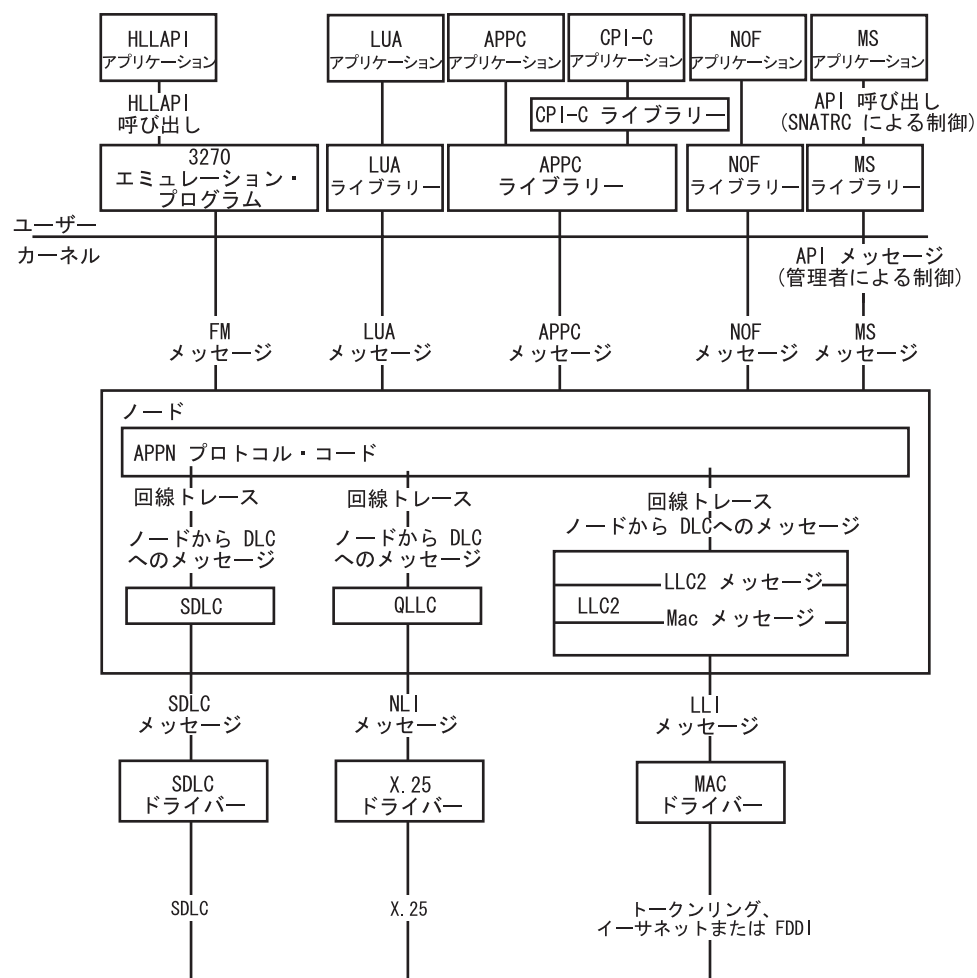


図2. Communications Server for Linux の全体構造

図2には、次のタイプのトレースが示されており、トレースはそれぞれ別々に制御されます。

APPC メッセージ

APPC ライブラリーとノード間のメッセージ

LUA メッセージ

LUA ライブラリーとノード間のメッセージ

NOF メッセージ

NOF ライブラリーとノード間のメッセージ

MS メッセージ

MS ライブラリーとノード間のメッセージ

DLC 回線トレース

DLC で送信される SNA データ (これらのメッセージのトレースは SET_TRACE_TYPE verb ではなく、ADD_DLC_TRACE verb により制御されます)

SET_TRACE_TYPE

ノードから DLC へのメッセージ

APPN ノードと DLC コンポーネント間のメッセージ。

さらに、以下のメッセージ・タイプ (Communications Server for Linux から見て内部的) をトレースすることができます。

ノード・メッセージ

APPN プロトコル・コード内のコンポーネント間のメッセージ

制御メッセージ

システム・コンポーネント間の内部制御メッセージ

START_DLC

START_DLC は、DLC の活動化を要求します。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct start_dlc
{
    AP_UINT16      opcode;           /* verb operation code */
    unsigned char  reserv2;         /* reserved */
    unsigned char  format;         /* reserved */
    AP_UINT16      primary_rc;      /* primary return code */
    AP_UINT32      secondary_rc;    /* secondary return code */
    unsigned char  dlc_name[8];     /* name of DLC */
} START_DLC;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_START_DLC

dlc_name

開始する DLC の名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。この名前は定義された DLC に一致する必要があります。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

この戻りコードは、verb が正常に発行されたことのみを示します。この verb は DLC の初期化を待たないため、DLC の初期化が失敗してもエラー戻りコードを戻しません。DLC 初期化の失敗は、エラー・ログ・ファイルに書き込まれるメッセージを使用して報告されます。

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
```

AP_INVALID_DLC

dlc_name パラメーターが定義された DLC の名前ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

AP_DLC_DEACTIVATING

指定した DLC は既に開始され、非活動化中です。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

START_INTERNAL_PU

START_INTERNAL_PU は、以前に定義されたローカル PU について、DLUR のサービスを受ける、SSCP-PU セッションの活動化を始めるように DLUR に要求します。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct start_internal_pu
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;         /* secondary return code */
    unsigned char  pu_name[8];           /* internal PU name */
    unsigned char  dlus_name[17];        /* DLUS name */
    unsigned char  bkup_dlus_name[17];   /* Backup DLUS name */
} START_INTERNAL_PU;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

START_INTERNAL_PU

opcode AP_START_INTERNAL_PU

pu_name

開始する内部 PU の名前 (これは以前に DEFINE_INTERNAL_PU を使用して定義されている必要があります)。この名前は 8 バイトからなるタイプ A の EBCDIC ストリング (英字で始まる) で、右側に EBCDIC スペースを入れます。

dlus_name

該当する PU についての SSCP-PU セッションの活動化を要求するために DLUR が接続する先となる DLUS ノードの名前。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

DEFINE_INTERNAL_PU verb で指定した DLUS、または DEFINE_INTERNAL_PU で何も指定していない場合に、DEFINE_DLUR_DEFAULTS で指定したグローバル・デフォルトを使用するには、このパラメーターの 17 桁の 2 進ゼロに設定します。

bkup_dlus_name

該当する PU の バックアップ DLUS として DLUR が保管する DLUS ノードの名前。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

DEFINE_INTERNAL_PU verb で指定したバックアップ DLUS、または DEFINE_INTERNAL_PU で何も指定していない場合に DEFINE_DLUR_DEFAULTS で指定したグローバル・バックアップ・デフォルトを使用するには、このパラメーターを 17 桁の 2 進ゼロに設定します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc
値は次のいずれかです。

AP_INVALID_DLUS_NAME

dlus_name パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

AP_INVALID_BKUP_DLUS_NAME

bkup_dlus_name パラメーターに、無効な文字またはフォーマットが正しくない文字が指定されました。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_NO_DEFAULT_DLUS_DEFINED

この `verb` または `DEFINE_INTERNAL_PU verb` で DLUS 名が指定されておらず、また `DEFINE_DLUR_DEFAULTS verb` が発行されていないためにデフォルト DLUS が定義されていません。

AP_PU_NOT_DEFINED

指定した PU 名は、`DEFINE_INTERNAL_PU` を使用して定義された内部 PU の名前ではありませんでした。

AP_PU_ALREADY_ACTIVATING

該当の PU は既に開始中です。

AP_PU_ALREADY_ACTIVE

該当の PU は既に開始されています。

戻りパラメーター: 失敗時

この `verb` が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_UNSUCCESSFUL

secondary_rc

値は次のいずれかです。

AP_DLUS_REJECTED

DLUS はセッション開始要求を拒否しました。

AP_DLUS_CAPS_MISMATCH

構成した DLUS 名が DLUS ノードではありませんでした。

AP_PU_FAILED_ACTPU

ローカル・ノードは DLUS からのメッセージを拒否しました。原因としては、内部エラー、リソースの不足、受信メッセージの問題が考えられます。さらに詳細情報を提供するメッセージが記録されている Communications Server for Linux ログ・ファイルを検査してください。

戻りパラメーター: サポートされていない機能

ノードの構成がこの verb をサポートしていないために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

ノードは DLUR をサポートしていません。これは、DEFINE_NODE の *dlur_supported* パラメーターによって定義されています。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

START_LS

START_LS は、通常、活動状態でない LS を開始します。また、LS を活動状態でない状態にして、必要なときに Communications Server for Linux による自動活動化またはリモート・システムによる活動化が可能であることを指定するためにも、この verb を使用できます。

注: LS が専用 SDLC リンクまたは QLLC PVC リンクの場合、LS は Communications Server for Linux の場合と同様にリモート・システムでも活動状態にする必要があります。ノードの始動時、および障害発生後にノードを自動的に再活動化するときに、活動化する LS を定義して、必ず、リンクを常時使用可能にしておくことをお勧めします。詳しくは、135 ページの『DEFINE_LS』を参照してください。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct start_ls
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  ls_name[8];     /* name of link station        */
    unsigned char  enable;         /* start ls or enable auto-activation? */
    unsigned char  react_kicked;   /* retry in progress?         */
    unsigned char  reserv3[2];     /* reserved                    */
} START_LS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_START_LS

ls_name

開始するリンク・ステーションの名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。この名前は DEFINE_LS verb で既に定義されている必要があります。

enable LS に対する処置を指定します。

LS を開始するには、このパラメーターを `AP_ACTIVATE` に設定します。

LS を活動状態でない状態にして、必要なときに (Communications Server for Linux またはリモート・システムにより) 活動化を行えることを指定するためには、以下の値の 1 つを指定するか、以下の値の両方を (論理 OR で組み合わせて) 指定します。

AP_AUTO_ACT

LS は、セッションに必要なときに Communications Server for Linux により自動的に活動化できます。この値は、LS が自動活動化を行えるように定義されている (LS 定義内の `auto_act_supp` が `AP_YES` に設定されている) 場合のみ使用してください。この値は、`STOP_LS` を使用して手動で停止した LS を再び自動活動化を行えるようにします。

AP_REMOTE_ACT

LS は、リモート・システムから活動化できます。この値は、LS 定義にある `disable_remote_act` の定義された値は変更しません。したがって、次にこの LS を停止すると、LS 定義での設定に戻ります。

戻りパラメーター: セッション起動の成功時

`verb` が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_LINK_NAME_SPECIFIED

`ls_name` パラメーターが、定義された LS の名前ではありませんでした。

AP_INVALID_LINK_ENABLE

`enable` パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、`AP_PARAMETER_CHECK` に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_ACTIVATION_LIMITS_REACHED

活動化の限度に達しました。

AP_LINK_DEACT_IN_PROGRESS

指定した LS は非活動化中です。非活動化プロセスが終了するまでは開始できません。

AP_ALREADY_STARTING

指定した LS は既に開始されています。

AP_PARALLEL_TGS_NOT_SUPPORTED

リモート・システムとのリンクは既に活動状態です。隣接ノードは並列伝送グループをサポートしていません。

AP_PORT_INACTIVE

関連したポートが活動状態でないため、LS は開始できません。

react_kicked

Communications Server for Linux が LS の活動化を (LS 定義の *react_timer_retry* パラメーターに基づいて) 再試行するかどうかを指定します。値は次のいずれかです。

AP_YES LS の活動化は (*react_timer_retry* で指定した試行回数まで) 再試行されます。

AP_NO LS の活動化は再試行されません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 失敗時

リモート・コンピューター上の SNA サブシステムへ接続できなかったために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_LS_FAILURE

secondary_rc

値は次のいずれかです。

AP_PARTNER_NOT_FOUND

この LS に関連したポートからの応答はありませんでした。トークンリング、イーサネットの場合は、LS 定義の中の *mac_address* パラメーターが正しいことを確認してください。

AP_ERROR

リモート・コンピューターへの接続が確立されませんでした。この原因としては、リモート・コンピューターの SNA サブシステムが始動していないことが考えられます。LAN タイプ以外のリンク・

タイプ (トークンリング、イーサネットなど) の場合、Communications Server for Linux が、指定されたアドレッシング情報と一致するリモート・コンピュータを見つけることができなかつたことを示している場合もあります。

戻りパラメーター: 取り消し時

別の verb によって取り消されたために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_CANCELLED

secondary_rc

値は次のいずれかです。

AP_NO_SECONDARY_RC

START_LS verb が終了する前に STOP_LS verb が発行されました。START_LS verb は取り消されました。

AP_LINK_DEACTIVATED

START_LS verb が終了する前に、LS が使用する DLC またはポートが停止しました。START_LS verb は取り消されました。

react_kicked

Communications Server for Linux が LS の活動化を (LS 定義の *react_timer_retry* パラメーターに基づいて) 再試行するかどうかを指定します。値は次のいずれかです。

AP_YES LS の活動化は (*react_timer_retry* で指定した試行回数まで) 再試行されます。

AP_NO LS の活動化は再試行されません。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

START_PORT

START_PORT は、ポートの活動化を要求します。ポートに対して指定する DLC は、この verb を発行する前に活動状態にする必要があります。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct start_port
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;          /* primary return code */
    AP_UINT32      secondary_rc;       /* secondary return code */
    unsigned char  port_name[8];       /* name of port */
} START_PORT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_START_PORT

port_name

開始するポートの名前。これは 8 バイトからなる ASCII スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。この名前は DEFINE_PORT verb で既に定義されている必要があります。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_PORT

port_name パラメーターが定義されたポートの名前ではありませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc
AP_STATE_CHECK

secondary_rc

値は次のいずれかです。

AP_DLC_INACTIVE

関連した DLC が活動状態でないため、ポートは開始できません。

AP_DUPLICATE_PORT

指定したポートは既に開始されています。

AP_STOP_PORT_PENDING

指定したポートは現在非活動化中です。非活動化プロセスが終了するまでは開始できません。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 取り消し時

この verb が取り消されたために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_CANCELLED

この verb が終了する前に STOP_PORT verb が発行されました。
START_PORT verb は取り消されました。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

STOP_DLC

STOP_DLC は、DLC を停止するように Communications Server for Linux に要求します。これにより DLC を使用する活動状態のポートおよび LS のいずれも停止します。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct stop_dlc
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;         /* secondary return code */
    unsigned char  stop_type;            /* stop type */
    unsigned char  dlc_name[8];          /* name of DLC */
} STOP_DLC;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_STOP_DLC

stop_type

必要な停止処理のタイプ。値は次のいずれかです。

AP_ORDERLY_STOP

Communications Server for Linux は DLC を停止する前に終結処理操作を実行します。

AP_IMMEDIATE_STOP

Communications Server for Linux は即時に DLC を停止します。

dlc_name

停止する DLC の名前。これは 8 バイトからなる ASCII スtring で、名

前が 8 バイトに満たない場合は、右側にスペースを入れます。この名前は DEFINE_DLC verb で既に定義されている必要があります。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_INVALID_DLC

dlc_name パラメーターは、定義された DLC の名前に一致しませんでした。

AP_UNRECOGNIZED_DEACT_TYPE

stop_type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

AP_STOP_DLC_PENDING

指定した DLC は既に停止中です。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 取り消し時

この verb が取り消されたために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
```


AP_CANCELLED

stop_type パラメーターは通常の停止を指定していましたが、DLC は、即時停止を指定した 2 番目のコマンドまたは障害状態により停止しました。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

STOP_INTERNAL_PU

STOP_INTERNAL_PU は、以前に定義された DLUR からサービスを受けるローカル PU について、SSCP-PU セッションの非活動化を開始するように DLUR に要求します。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct stop_internal_pu
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;         /* secondary return code */
    unsigned char  pu_name[8];           /* internal PU name */
    unsigned char  stop_type;            /* type of stop requested */
} STOP_INTERNAL_PU;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_STOP_INTERNAL_PU

pu_name

SSCP-PU セッションが非活動化されている内部 PU の名前。これは 8 バイトからなるタイプ A の EBCDIC スtring (英文字で始まる) で、右側に EBCDIC スペースを入れます。

stop_type

PU を停止する方法を指定します。値は次のいずれかです。

AP_ORDERLY_STOP

基礎 PLU-SLU セッションおよび基礎 SSCP-LU セッションをすべて非活動化してから、SSCP-PU セッションを非活動化します。

AP_IMMEDIATE_STOP

SSCP-PU セッションをすぐに非活動化します。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

STOP_INTERNAL_PU

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
```

AP_INVALID_STOP_TYPE

`stop_type` パラメーターが、有効な値に設定されていませんでした。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

値は次のいずれかです。

AP_PU_NOT_DEFINED

指定した PU 名は、定義された内部 PU の名前に一致しませんでした。

AP_PU_ALREADY_DEACTIVATING

PU は既に停止中です。

AP_PU_NOT_ACTIVE

PU は活動状態ではありません。

戻りパラメーター: サポートされていない機能

ノードの構成がこの `verb` をサポートしていないために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
```

AP_FUNCTION_NOT_SUPPORTED

ノードは DLUR をサポートしていません。これは、`DEFINE_NODE` の `dlur_supported` パラメーターによって定義されています。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

STOP_LS

STOP_LS は活動状態である LS を停止します。その代わりに、これを活動状態でない LS に発行して、必要なときに Communications Server for Linux により LS を自動活動化できないようにすること、またはリモート・システムにより LS を活動化できないようにすることを指定できます。これらの両方が使用不可の場合、START_LS を発行して LS を活動状態にすることができます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct stop_ls
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  stop_type;     /* stop type                   */
    unsigned char  ls_name[8];    /* name of link station        */
    unsigned char  disable;       /* disable remote or auto activation? */
    unsigned char  reserved[3];   /* reserved                     */
} STOP_LS;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_STOP_LS

stop_type

必要な停止処理のタイプ。値は次のいずれかです。

AP_ORDERLY_STOP

Communications Server for Linux は LS を停止する前に終結処理操作を実行します。

AP_IMMEDIATE_STOP

Communications Server for Linux は即時に LS を停止します。

ls_name

停止する LS の名前。これは 8 バイトからなる ASCII スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。この名前は DEFINE_LS verb で既に定義されている必要があります。

disable LS に対する処置を指定します。

活動状態である LS を停止し、自動活動化およびリモート活動化に関するデフォルト設定に戻すには、このパラメーターを AP_NO に設定します。

活動状態でない LS を Communications Server for Linux、またはリモート・システムによって活動化できないように指定するには、以下の値の 1 つを指定するか、両方を (論理 OR で組み合わせて) 指定します。

AP_AUTO_ACT

Communications Server for Linux により LS を自動的に活動化することはできません。

AP_REMOTE_ACT

LS は、リモート・システムにより活動化することはできません。
この値は、LS 定義内の *disable_remote_act* の定義された値は変更しません。次に LS が開始されて停止されると、LS は定義された設定に戻ります。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    値は次のいずれかです。
```

AP_LINK_NOT_DEFD

ls_name パラメーターは、定義された LS の名前に一致しませんでした。

AP_UNRECOGNIZED_DEACT_TYPE

stop_type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_STATE_CHECK
```

```
secondary_rc
```

AP_LINK_DEACT_IN_PROGRESS

指定した LS は既に非活動化中です。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 取り消し時

この *verb* が取り消されたために *verb* が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
```

AP_CANCELLED

stop_type パラメーターは通常の停止を指定していましたが、LS は、即時停止を指定した 2 番目の *verb* または障害状態により停止しました。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF *verb* に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

STOP_PORT

STOP_PORT は、アプリケーションによるポートの停止を可能にします。また、そのポートを使用している活動状態である LS をすべて停止させます。

この *verb* は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct stop_port
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;              /* reserved */
    AP_UINT16      primary_rc;          /* primary return code */
    AP_UINT32      secondary_rc;        /* secondary return code */
    unsigned char  stop_type;          /* Stop Type */
    unsigned char  port_name[8];       /* name of port */
} STOP_PORT;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_STOP_PORT

stop_type

必要な停止処理のタイプ。値は次のいずれかです。

AP_ORDERLY_STOP

Communications Server for Linux はポートを停止する前に終結処理操作を実行します。

AP_IMMEDIATE_STOP

Communications Server for Linux は即時にポートを停止します。

port_name

停止するポートの名前。これは 8 バイトからなる ASCII スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_OK

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

値は次のいずれかです。

AP_INVALID_PORT_NAME

port_name パラメーターは、定義されたポートの名前に一致しませんでした。

AP_UNRECOGNIZED_DEACT_TYPE

stop_type パラメーターが、有効な値に設定されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 状態の検査

状態エラーのために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

AP_STOP_PORT_PENDING

指定したポートは既に非活動化中です。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通し、AP_STATE_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: 取り消し時

この `verb` が取り消されたために `verb` が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_CANCELLED

stop_type パラメーターは通常の停止を指定していましたが、ポートは、即時停止を指定した 2 番目の `verb` または障害条件により停止しました。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF `verb` に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

TERM_NODE

TERM_NODE は、アプリケーションでノードに指定した緊急度で停止できるようにします。また、そのノードに関連したすべての接続リソースも停止させます。

この verb は、実行中のノードに対して発行しなければなりません。

VCB 構造体

```
typedef struct term_node
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;              /* reserved */
    unsigned char  format;               /* reserved */
    AP_UINT16      primary_rc;           /* primary return code */
    AP_UINT32      secondary_rc;         /* secondary return code */
    unsigned char  stop_type;            /* stop type */
} TERM_NODE;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_TERM_NODE

stop_type

Communications Server for Linux でノードを停止させる方法を指定します。値は次のいずれかです。

AP_ABORT

終結処理をまったく行わずにすぐに停止します。この値は、ノードのリソースを使用している他のプログラムに問題を引き起こす場合があるため、重大エラー状態の場合のみ使用してください。

AP_SHUTDOWN

ノードに関連した LS をすべて非活動化してから、ノードを停止します。

AP QUIESCE

ノードが静止したことをネットワークに指示し、すべてのモードでセッション限度をリセットし、ノードの LU に対するすべてのエンドポイント・セッションをバインド解除してから、AP_SHUTDOWN の場合と同様に停止します。

AP QUIESCE_ISR

AP QUIESCE と同じ機能ですが、ノードはすべての中間セッションの終了を待ちます。この値はネットワーク・ノードのみに適用されます。

AP_DEACT_CLEAN

セッション限度をリセットできないこと、および LS を非活動状態にする前に RTP 接続を正常に終了できることを除いて、AP QUIESCE と同じ機能です。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

UNREGISTER_INDICATION_SINK

UNREGISTER_INDICATION_SINK は、NOF アプリケーションが今後、特定のタイプの指示 (前に REGISTER_INDICATION_SINK を使用して指定された指示) を受け取らないように、NOF アプリケーションの登録を解除します。

アプリケーションが複数の指示タイプを受け取るように複数回登録された場合は、受け取る必要がなくなった指示ごとに別々に登録解除する必要があります。

この verb は、コールバック・ルーチンを含めた非同期 NOF API エントリ・ポイントを必ず使用して発行する必要があります (非同期 NOF API エントリ・ポイントについて詳しくは、30 ページの『非同期エントリ・ポイント: nof_async』を参照してください)。

この verb は、アプリケーションが登録解除する指示のタイプに応じて、ドメイン構成ファイル、実行中のノード、ノードを実行していないサーバー、または **sna.net** ファイルに対して発行できます。

VCB 構造体

```
typedef struct unregister_indication_sink
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    AP_UINT32      proc_id;        /* reserved                    */
    AP_UINT16      queue_id;       /* reserved                    */
    AP_UINT16      indication_opcode; /* opcode of indication to be unsunk */
} UNREGISTER_INDICATION_SINK;
```

指定パラメーター

アプリケーションが提供するパラメーターを次に示します。

opcode AP_UNREGISTER_INDICATION_SINK

indication_opcode

不要になった指示の *opcode* パラメーター。

戻りパラメーター: セッション起動の成功時

verb が正常に実行されると、Communications Server for Linux は、以下のパラメーターを返します。

```
primary_rc
    AP_OK
```

戻りパラメーター: パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを返します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
```

AP_INVALID_OP_CODE

indication_opcode パラメーターが Communications Server for Linux NOF 指示のいずれかの *opcode* に一致しなかったか、指定した指示をこのターゲット・ハンドルで受け取るように、事前にアプリケーションに登録されていませんでした。

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通し、AP_PARAMETER_CHECK に関連した上記以外の 2 次戻りコードのリストを示します。

戻りパラメーター: サポートされていない機能

指定した指示に関連した機能がローカル・ノードの構成でサポートされていないために verb が正常に実行されないと、Communications Server for Linux は、以下のパラメーターを返します。

```
primary_rc
```

AP_FUNCTION_NOT_SUPPORTED

ローカル・ノードは指定した指示をサポートしていません。それぞれの指示に必要なサポートについて詳しくは、777 ページの『第 4 章 NOF 指示』にあるそれぞれの指示の説明を参照してください。

戻りパラメーター: その他の場合

851 ページの『付録 B. 共通戻りコード』に、すべての NOF verb に共通した上記以外の 1 次戻りコードおよび 2 次戻りコードの組み合わせのリストを示します。

第 4 章 NOF 指示

本章では、それぞれの NOF 指示ごとに、次の情報を記載しています。

- 指示の目的と使用方法の説明
- verb 制御ブロック (VCB) 構造体。NOF API ヘッダー・ファイル `/usr/include/sna/nof_c.h` (AIX) または `/opt/ibm/sna/include/nof_c.h` (Linux) に定義されています。
- VCB でアプリケーションに戻されるパラメーターの説明

NOF 指示を受け取るようにアプリケーションを登録する方法については、719 ページの『REGISTER_INDICATION_SINK』を参照してください。

CONFIG_INDICATION

この指示は、別の NOF アプリケーションまたは Communications Server for Linux 管理ツールがターゲットの構成に変更を行った場合、ターゲット・ノードが停止または開始した場合、あるいは、ターゲット・ノードが所有する DLC、ポート、または LS が停止または開始した場合に生成されます。ターゲットは、ドメイン構成ファイル、実行中のノード、または Communications Server for Linux ソフトウェアを実行しているサーバー上の非活動ノードの場合があります。ターゲットは、この指示を受け取るための登録を行った REGISTER_INDICATION_SINK verb の `target_handle` パラメーターにより識別されます。

VCB 構造体

この指示に関連した特定の VCB 構造体はありません。構成ディレクティブのための登録を行うには、アプリケーションで値 `AP_CONFIG_INDICATION` を REGISTER_INDICATION_SINK の `indication_opcode` パラメーターとして指定します。変更が行われると、Communications Server for Linux は、変更を行った NOF verb からの VCB のコピーを送信することによって、アプリケーションのコールバック・ルーチンに変更を報告します。たとえば、DEFINE_DLC verb により構成が変更されると、Communications Server for Linux は DEFINE_DLC VCB のコピーをアプリケーションのコールバック・ルーチンに送信します。

アプリケーションで構成ディレクティブと独自の NOF verb への非同期応答を区別できるようにするために、Communications Server for Linux は、構成ディレクティブのための VCB の `primary_rc` パラメーターを変更します。値 `AP_INDICATION` は、構成ディレクティブに関連した VCB を特定します。値 `AP_OK` またはその他の値は、アプリケーション独自の NOF verb のいずれかへの非同期応答を示します。

以下のイベントは構成ディレクティブとして報告されません。

- SNA ネットワーク・ファイル `sna.net` に対する変更。これらの変更の指示を受け取るために、アプリケーションは指示タイプ `AP_SNA_NET_INDICATION` を登録する必要があります。詳しくは、838 ページの『SNA_NET_INDICATION』を参照してください。

CONFIG_INDICATION

- 他のサーバーでの SNA ソフトウェアの開始および停止。これらの変更の指示を受け取るために、アプリケーションは指示タイプ AP_SERVER_INDICATION を登録する必要があります。詳しくは、832 ページの『SERVER_INDICATION』を参照してください。

構成ディレクティブとして戻すことができる VCB の範囲は、REGISTER_INDICATION_SINK で指定したターゲット・ハンドルのタイプにより異なります。

ドメイン構成ファイル

アプリケーションは、ドメイン・リソースを変更するがノード・リソースは変更しない、どの verb (ドメイン構成ファイルに対して発行できる verb) についても、VCB を受信できます。

ノード構成ファイル

アプリケーションは、ノード・リソースを変更する、どの verb についても、VCB を受信できます。

実行中のノード

アプリケーションは、ノード・リソースを変更するすべての verb についての VCB、TERM_NODE VCB、および DLC、ポート、LS に対する START_* VCB と STOP_* VCB を受信できます。

非活動ノード

アプリケーションは、ノード・リソースを変更するどの verb についても VCB を受信でき、さらに INIT_NODE VCB も受信できます。

DIRECTORY_INDICATION

この指示は、エントリーがローカル・ディレクトリー・データベースに追加されたとき、またはローカル・ディレクトリー・データベースから除去されたときに生成されます。

VCB 構造体

```
typedef struct directory_indication
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code         */
    AP_UINT32      secondary_rc;    /* secondary return code       */
    unsigned char  data_lost;      /* previous indication lost    */
    unsigned char  removed;        /* is entry being removed?    */
    unsigned char  resource_name[17]; /* resource name              */
    unsigned char  invalid;       /* invalid entry being removed? */
    AP_UINT16      resource_type;   /* resource type              */
    unsigned char  parent_name[17]; /* parent resource name       */
    unsigned char  entry_type;     /* type of the directory entry */
    AP_UINT16      parent_type;    /* parent resource type       */
    unsigned char  description[32]; /* resource description        */
    unsigned char  reserv3[16];    /* reserved                    */
    AP_UINT16      real_owing_cp_type; /* CP type of real owner     */
    unsigned char  real_owing_cp_name[17]; /* CP name of real owner    */
    AP_UINT16      supplier_cp_type; /* CP type of supplier       */
    unsigned char  supplier_cp_name[17]; /* CP name of supplier      */
    unsigned char  reserva;       /* reserved                    */
} DIRECTORY_INDICATION;
```

パラメーター

opcode AP_DIRECTORY_INDICATION

primary_rc

AP_OK

data_lost

既存のディレクトリー指示の中に消失したものがどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のディレクトリー指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のディレクトリー指示は消失していません。

removed

指示されたリソースがディレクトリーから除去されたか、ディレクトリーに追加されたかを指定します。値は次のいずれかです。

AP_YES エントリーは除去されました。

AP_NO エントリーは追加されました。

resource_name

リソースの完全修飾名。この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。

invalid

エンド・ノードがそのリソースをネットワーク・ノードに登録すると、ネットワーク・ノードのディレクトリー・データベースに、これらのリソースについての新しいディレクトリー・エントリーが追加されます。これらのリソースの 1 つに対して明示的に定義されたエントリーが既にデータベースにあっても、そのエントリーが登録されたリソースに一致しない場合、Communications Server for Linux はその無効なエントリーを除去し、正しいエントリーに置換します。このパラメーターは、エントリーが除去された理由が、無効であったために登録されたリソースから正しいエントリーに置換されたためか、明示的に削除されたためかを示します。値は次のいずれかです。

AP_YES エントリーは正しくないために除去されました。

AP_NO エントリーは明示的に削除されたために除去されました。

ローカル・ノードがネットワーク・ノードでない場合、または *removed* を AP_NO に設定した場合、このパラメーターは使用されません。

resource_type

リソース・タイプ。値は次のいずれかです。

AP_NNCP_RESOURCE

ネットワーク・ノード。

DIRECTORY_INDICATION

AP_ENCP_RESOURCE

エンド・ノード。

AP_LU_RESOURCE

LU。

parent_name

親リソースの完全修飾名。*resource_type* が AP_NNCP_RESOURCE の場合は、これは 17 桁の 2 進ゼロに設定されます。

この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A String文字からなるネットワーク名で構成されます。

entry_type

ディレクトリー・エントリーのタイプを指定します。次のいずれかです。

AP_HOME

ローカル・リソース。

AP_CACHE

キャッシュに入れられたエントリー。

AP_REGISTER

登録済みリソース (NN のみ)。

parent_type

登録するリソースの親タイプを指定します。このパラメーターは、*resource_type* が AP_NNCP_RESOURCE の場合には使用されません。値は次のいずれかです。

AP_NNCP_RESOURCE

ネットワーク・ノード。

AP_ENCP_RESOURCE

エンド・ノード。

description

リソースを記述したヌルで終了するテキスト・String (そのリソースの定義で指定したもの)。

real_owning_cp_type

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

このディレクトリー・エントリーによって識別されるリソースを所有する実 CP が、親リソースであるのか、別のノードであるのかを指定します。次のいずれかです。

AP_NONE

実際の所有者は親リソースです。

AP_ENCP_RESOURCE

実際の所有者は、親リソースではないエンド・ノードです。たとえば、分岐ネットワーク・ノード (Branch Network Node: BrNN) のドメインでエンド・ノードがリソースを所有する場合、この BrNN の

ネットワーク・ノード・サーバーのディレクトリーには、親リソースとして BrNN を含みますが、実際の所有 CP はエンド・ノードです。

real_owning_cp_name

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

real_owning_cp_type パラメーターがリソースの実際の所有者は親でないことを示す場合には、このパラメーターは、リソースを所有する CP の完全修飾名を指定します。それ以外の場合には、使用しません。

この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String文字からなるネットワーク名で構成されます。

supplier_cp_type

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

リソースの所有 CP ではない別のノードが、このディレクトリー・エントリーを登録したかどうかを指定します。次のいずれかです。

AP_NONE

ディレクトリー・エントリーは、登録されなかったか、その所有側 CP によって登録されました。

AP_ENCP_RESOURCE

ディレクトリー・エントリーは、その所有側 CP ではないノードによって登録されました。たとえば、それ自体ローカル・ノードのドメインにある分岐ネットワーク・ノード (BrNN) のドメインにあるエンド・ノードがリソースを所有している場合、そのリソースをローカル・ノードに登録した BrNN がそのリソースの提供者になりますが、実所有側 CP はエンド・ノードです。

supplier_cp_name

このパラメーターは、ローカル・ノードがネットワーク・ノードまたは分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

supplier_cp_type パラメーターが、所有リソースではないノードによってディレクトリー・エントリーが登録されたことを示す場合には、このパラメーターは、登録を提供した CP の完全修飾名を指定します。それ以外の場合には、使用しません。

この名前は 17 バイトの EBCDIC Stringで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字の A String文字からなるネットワーク名で構成されます。

DLC_INDICATION

この指示は、DLC の状態が活動から非活動へ変わると生成されます。

VCB 構造体

```
typedef struct dlc_indication
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code         */
    AP_UINT32      secondary_rc;    /* secondary return code       */
    unsigned char  data_lost;      /* previous indication lost    */
    unsigned char  deactivated;    /* has session been deactivated? */
    unsigned char  dlc_name[8];    /* link station name          */
    unsigned char  description[32]; /* resource description        */
    unsigned char  reserv1[16];    /* reserved                    */
    unsigned char  reserva[20];   /* reserved                    */
} DLC_INDICATION;
```

パラメーター

opcode AP_DLC_INDICATION

primary_rc
AP_OK

data_lost

既存の指示の中に消失したものがあるかどうかを指定します。

Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の DLC 指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存の DLC 指示は消失していません。

deactivated

DLC が非活動状態になったか、活動状態になったかを指定します。値は次のいずれかです。

AP_YES DLC は非活動状態になりました。

AP_NO DLC は活動状態になりました。

dlc_name

DLC の名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

description

DLC を記述したヌルで終了するテキスト・ストリング (その DLC の定義で指定したもの)。

DLUR_LU_INDICATION

この指示は、DLUR LU が活動化または非活動化されると生成されます。この指示は、登録されたアプリケーションが現在活動状態である DLUR LU のリストを保持するために使用できます。

VCB 構造体

```
typedef struct dlur_lu_indication
{
    AP_UINT16      opcode;           /* Indication operation code      */
    unsigned char  reserv2;         /* reserved                        */
    unsigned char  format;         /* reserved                        */
    AP_UINT16      primary_rc;      /* primary return code            */
    AP_UINT32      secondary_rc;    /* secondary return code          */
    unsigned char  data_lost;       /* Previous indication lost?      */
    unsigned char  reason;         /* reason for this indication     */
    unsigned char  lu_name[8];     /* LU name                        */
    unsigned char  pu_name[8];     /* PU name                        */
    unsigned char  nau_address;     /* NAU address                    */
    unsigned char  reserv5[7];     /* reserved                       */
} DLUR_LU_INDICATION;
```

パラメーター

opcode AP_DLUR_LU_INDICATION

primary_rc
AP_OK

data_lost

既存のディレクトリー指示の中に消失したものがあるかどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のディレクトリー指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のディレクトリー指示は消失していません。

reason この指示の理由。値は次のいずれかです。

AP_ADDED
DLUR は DLUS により活動化されたところです。

AP_REMOVED
DLUR は、DLUS により明示的に、あるいはリンク障害または PU の非活動化により暗黙的に非活動化されました。

lu_name

論理装置 (LU) の名前。これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

DLUR_LU_INDICATION

pu_name

この LU が使用する物理装置 (PU) の名前。これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

nau_address

LU のネットワーク・アクセス可能単位 (NAU) アドレス。この値の範囲は 1 ~ 255 です。

DLUR_PU_INDICATION

この指示は、従属 LU リクエスター (DLUR) 機能をサポートするノードの物理装置 (PU) が活動化を試みたとき、活動化に失敗したとき、活動化したとき、または非活動化したときに生成されます。この指示は、現在活動状態である DLUR PU のリストを保持するために使用できます。

VCB 構造体

```
typedef struct dlur_pu_indication
{
    AP_UINT16      opcode;           /* Indication operation code      */
    unsigned char  reserv2;         /* reserved                        */
    unsigned char  format;         /* reserved                        */
    AP_UINT16      primary_rc;      /* primary return code            */
    AP_UINT32      secondary_rc;    /* secondary return code          */
    unsigned char  data_lost;       /* Previous indication lost?      */
    unsigned char  reason;         /* reason for this indication     */
    unsigned char  pu_name[8];     /* PU name                        */
    unsigned char  pu_id[4];       /* PU identifier                  */
    unsigned char  pu_location;    /* downstream or local PU        */
    unsigned char  pu_status;      /* status of the PU              */
    unsigned char  dlus_name[17];  /* current DLUS name             */
    unsigned char  dlus_session_status; /* status of the DLUS pipe      */
    unsigned char  reserv5[2];     /* reserved                       */
} DLUR_PU_INDICATION;
```

パラメーター

opcode AP_DLUR_PU_INDICATION

primary_rc

AP_OK

data_lost

既存のディレクトリー指示の中に消失したものがどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のディレクトリー指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のディレクトリー指示は消失していません。

reason 指示の原因。値は次のいずれかです。

値は次のいずれかです。

AP_ACTIVATION_STARTED

PU は活動化中です。

AP_ACTIVATING

PU は活動状態になりました。

AP_DEACTIVATING

PU は非活動状態になりました。

AP_FAILED

PU は失敗しました。

AP_ACTIVATION_FAILED

PU は活動化に失敗しました。

pu_name

物理装置 (PU) の名前。これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

pu_id

DEFINE_INTERNAL_PU verb で定義された PU ID またはダウンストリーム PU からの XID で取得された PU ID。この値は 4 バイトの 16 進数 Stringです。ビット 0 ~ 11 はブロック番号に設定され、ビット 12 ~ 31 は PU を固有に識別する ID 番号に設定されます。

pu_location

PU の位置。値は次のいずれかです。

AP_INTERNAL

PU は DEFINE_INTERNAL_PU verb で定義されています。

AP_DOWNSTREAM

PU はダウンストリーム・コンピューターにあります。

pu_status

PU の状況 (DLUR から見た)。値は次のいずれかです。

AP_RESET_NO_RETRY

PU はリセット状態であり、再試行されません。

AP_RESET_RETRY

PU はリセット状態であり、再試行されます。

AP_PEND_ACTPU

PU はホストからの ACTPU を待ちます。

AP_PEND_ACTPU_RSP

DLUR は ACTPU を PU に転送したあと、PU の応答を待ちます。

AP_ACTIVE

PU は活動状態です。

AP_PEND_DACTPU_RSP

DLUR は DACTPU を PU に転送したあと、PU の応答を待ちます。

AP_PEND_INOP

DLUR は PU を非活動化する前に、必要なイベントがすべて完了するのを待ちます。

DLUR_PU_INDICATION

dlus_name

PU が現在使用している (または使用しようとしている) 従属 LU サーバー (DLUS) ノードの名前。この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A スtring文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A スtring文字からなるネットワーク名で構成されます。*reason* パラメーターを AP_FAILED に設定した場合は、*dlus_name* パラメーターはすべてゼロに設定されます。

dlus_session_status

PU が現在使用している DLUS パイプの状況。値は次のいずれかです。

AP_PENDING_ACTIVE

DLUS パイプは現在活動化中です。

AP_ACTIVE

DLUS パイプは活動状態です。

AP_PENDING_INACTIVE

DLUS パイプは現在活動化されていません。

AP_INACTIVE

DLUS パイプは非活動状態です。

DLUS_INDICATION

この指示は、DLUS ノードへのパイプの状態が活動から非活動に変わると生成されます。パイプが活動状態でなくなると、指示にはパイプ統計も組み込まれます。

VCB 構造体

```
typedef struct dlus_indication
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;                /* reserved                  */
    AP_UINT16      primary_rc;            /* primary return code      */
    AP_UINT32      secondary_rc;          /* secondary return code    */
    unsigned char  data_lost;             /* previous indication lost */
    unsigned char  deactivated;           /* has DLUS become inactive? */
    unsigned char  dlus_name[17];         /* DLUS name                 */
    unsigned char  reserv1;               /* reserved                  */
    PIPE_STATS     pipe_stats;             /* pipe statistics          */
    unsigned char  persistent_pipe_support; /* reserved                  */
    unsigned char  persistent_pipe;       /* reserved                  */
    unsigned char  reserva[18];           /* reserved                  */
} DLUS_INDICATION;

typedef struct pipe_stats
{
    AP_UINT32      reqactpu_sent;          /* REQACTPUs sent to DLUS  */
    AP_UINT32      reqactpu_rsp_received; /* RSP(REQACTPU)s received */
                                          /* from DLUS                */
    AP_UINT32      actpu_received;         /* ACTPUs received from DLUS */
    AP_UINT32      actpu_rsp_sent;         /* RSP(ACTPU)s sent to DLUS */
    AP_UINT32      reqdactpu_sent;         /* REQDACTPUs sent to DLUS */
    AP_UINT32      reqdactpu_rsp_received; /* RSP(REQDACTPU)s received */
                                          /* from DLUS                */
    AP_UINT32      dactpu_received;        /* DACTPUs received from DLUS */
    AP_UINT32      dactpu_rsp_sent;        /* RSP(DACTPU)s sent to DLUS */
    AP_UINT32      actlu_received;         /* ACTLUs received from DLUS */
}
```

```

AP_UINT32    actlu_rsp_sent;          /* RSP(ACTLU)s sent to DLUS */
AP_UINT32    dactlu_received;        /* DACTLUS received from DLUS */
AP_UINT32    dactlu_rsp_sent;        /* RSP(DACTLU)s sent to DLUS */
AP_UINT32    sscp_pu_mus_rcvd;       /* MUs for SSCP-PU sessions rcvd */
AP_UINT32    sscp_pu_mus_sent;       /* MUs for SSCP-PU sessions sent */
AP_UINT32    sscp_lu_mus_rcvd;       /* MUs for SSCP-LU sessions rcvd */
AP_UINT32    sscp_lu_mus_sent;       /* MUs for SSCP-LU sessions sent */
} PIPE_STATS;

```

パラメーター

opcode AP_DLUS_INDICATION

primary_rc

AP_OK

data_lost

既存の DLUS 指示の中に消失したものがどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のダウンストリーム LU 指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のダウンストリーム LU 指示は消失していません。

deactivated

パイプが非活動状態になったか活動状態になったかを指定します。値は次のいずれかです。

AP_YES パイプは非活動状態になりました。

AP_NO パイプは活動状態になりました。

dlus_name

DLUS ノードの名前。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

パイプが非活動化された場合は、*pipe_stats* 構造体が組み込まれます。この構造体のフィールドは次のとおりです。

pipe_stats.reqactpu_sent

パイプを介して DLUS に送信された REQACTPU の数。

pipe_stats.reqactpu_rsp_received

パイプを介して DLUS から受信された RSP(REQACTPU) の数。

pipe_stats.actpu_received

パイプを介して DLUS から受信された ACTPU の数。

pipe_stats.actpu_rsp_sent

パイプを介して DLUS に送信された RSP(ACTPU) の数。

DLUS_INDICATION

pipe_stats.reqdactpu_sent

パイプを介して DLUS に送信された REQDACTPU の数。

pipe_stats.reqdactpu_rsp_received

パイプを介して DLUS から受信された RSP(REQDACTPU) の数。

pipe_stats.dactpu_received

パイプを介して DLUS から受信された DACTPU の数。

pipe_stats.dactpu_rsp_sent

パイプを介して DLUS に送信された RSP(DACTPU) の数。

pipe_stats.actlu_received

パイプを介して DLUS から受信された ACTLU の数。

pipe_stats.actlu_rsp_sent

パイプを介して DLUS に送信された RSP(ACTLU) の数。

pipe_stats.dactlu_received

パイプを介して DLUS から受信された DACTLU の数。

pipe_stats.dactlu_rsp_sent

パイプを介して DLUS に送信された RSP(DACTLU) の数。

pipe_stats.sscp_pu_mus_rcvd

パイプを介して DLUS から受信された SSCP-PU MU の数。

pipe_stats.sscp_pu_mus_sent

パイプを介して DLUS に送信された SSCP-PU MU の数。

pipe_stats.sscp_lu_mus_rcvd

パイプを介して DLUS から受信された SSCP-LU MU の数。

pipe_stats.sscp_lu_mus_sent

パイプを介して DLUS に送信された SSCP-LU MU の数。

DOWNSTREAM_LU_INDICATION

この指示は、ダウンストリーム LU とホスト間の LU-SSCP セッションまたは PLU-SLU セッションの状態が活動と非活動の間で変わると生成されます。これらのセッションの 1 つが非活動状態になると、指示にはそのセッションのセッション統計も組み込まれます。

VCB 構造体

```
typedef struct downstream_lu_indication
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code         */
    AP_UINT32      secondary_rc;    /* secondary return code       */
    unsigned char  data_lost;       /* previous indication lost    */
    unsigned char  dspu_name[8];    /* PU name                    */
    unsigned char  ls_name[8];     /* Link station name          */
    unsigned char  dsLu_name[8];   /* LU name                    */
    unsigned char  description[32]; /* resource description        */
    unsigned char  reserv3[16];    /* reserved                   */
    unsigned char  nau_address;    /* NAU address                 */
    unsigned char  lu_sscp_sess_active; /* Is LU-SSCP session active */
}
```

```

unsigned char   plu_sess_active;    /* Is PLU-SLU session active    */
unsigned char   dspu_services;     /* DSPU services                 */
unsigned char   reserv1;          /* reserved                      */
SESSION_STATS  lu_sscp_stats;     /* LU-SSCP session statistics    */
SESSION_STATS  ds_plu_stats;      /* Downstream PLU-SLU session stats */
SESSION_STATS  us_plu_stats;      /* Upstream PLU-SLU session stats */
} DOWNSTREAM_LU_INDICATION;
typedef struct session_stats
{
  AP_UINT16     rcv_ru_size;       /* session receive RU size      */
  AP_UINT16     send_ru_size;     /* session send RU size         */
  AP_UINT16     max_send_btu_size; /* maximum send BTU size       */
  AP_UINT16     max_rcv_btu_size; /* maximum rcv BTU size        */
  AP_UINT16     max_send_pac_win; /* maximum send pacing window size */
  AP_UINT16     cur_send_pac_win; /* current send pacing window size */
  AP_UINT16     max_rcv_pac_win; /* maximum receive pacing window size*/
  AP_UINT16     cur_rcv_pac_win; /* current receive pacing window size*/
  AP_UINT32     send_data_frames; /* number of data frames sent   */
  AP_UINT32     send_fmd_data_frames; /* num fmd data frames sent   */
  AP_UINT32     send_data_bytes; /* number of data bytes sent    */
  AP_UINT32     rcv_data_frames; /* number of data frames received */
  AP_UINT32     rcv_fmd_data_frames; /* num fmd data frames received */
  AP_UINT32     rcv_data_bytes; /* number of data bytes received */
  unsigned char sidh;            /* session ID high byte (from LFSID) */
  unsigned char sidl;            /* session ID low byte (from LFSID) */
  unsigned char odai;           /* ODAI bit set                  */
  unsigned char ls_name[8];     /* Link station name             */
  unsigned char reserve;        /* reserved                      */
} SESSION_STATS;

```

パラメーター

opcode AP_DOWNSTREAM_LU_INDICATION

primary_rc

AP_OK

data_lost

既存のダウンストリーム LU 指示の中に消失したものがあるかどうかを指定します。 Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のダウンストリーム LU 指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のダウンストリーム LU 指示は消失していません。

dspu_name

ダウンストリーム LU に関連したダウンストリーム PU の名前。これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC スtring で、右側に EBCDIC のスペースを入れます。

ls_name

ダウンストリーム LU に関連したリンク・ステーションの名前。これは 8 バイトからなる ASCII スtring で、名前が 8 文字に満たない場合、右側にスペースを入れます。

DOWNSTREAM_LU_INDICATION

dslu_name

ダウンストリーム LU の名前。これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC スtringで、右側に EBCDIC のスペースを入れます。

description

ダウンストリーム LU を記述したヌルで終了するテキスト・String (その LU の定義で指定したもの)。

nau_address

LU のネットワーク・アクセス可能単位アドレス。

lu_sscp_sess_active

LU-SSCP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

plu_sess_active

PLU-SLU セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

dspu_services

ローカル・ノードがダウンストリーム LU に提供するサービスを指定します。

値は次のいずれかです。

AP_PU_CONCENTRATION

ダウンストリーム LU は SNA ゲートウェイからサービスを受けません。

AP_DLUR

ダウンストリーム LU は DLUR からサービスを受けません。

LU-SSCP セッションが非活動化された場合は、このセッションについて `session_stats` 構造体が組み込まれます。PLU-SLU セッションが非活動化された場合は、ダウンストリームおよびアップストリームの PLU-SLU セッションについて `session_stats` 構造体が組み込まれます。この構造体のフィールドは次のとおりです。

rcv_ru_size

受信 RU の最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

send_ru_size

送信 RU の最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

max_send_btu_size

送信できる BTU の最大サイズ。

max_rcv_btu_size

受信できる BTU の最大サイズ。

max_send_pac_win

このセッションの送信ペーシング・ウィンドウの最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

cur_send_pac_win

このセッションの送信ペーシング・ウィンドウの現行サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

max_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

cur_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの現行サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

send_data_frames

送信された通常フロー・データ・フレームの数。

send_fmd_data_frames

送信された通常フロー FMD データ・フレームの数。

send_data_bytes

送信された通常フロー・データ・バイトの数。

rcv_data_frames

受信された通常フロー・データ・フレームの数。

rcv_fmd_data_frames

受信された通常フロー FMD データ・フレームの数。

rcv_data_bytes

受信された通常フロー・データ・バイトの数。

sidh セッション ID の上位バイト。(アップストリーム PLU-SLU セッション統計では、このパラメーターは予約済みです。)

sidl セッション ID の下位バイト。(アップストリーム PLU-SLU セッション統計では、このパラメーターは予約済みです。)

odai 起点宛先アサイナー・インディケーター。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合、BIND の送信側は、このフィールドをゼロに設定し、BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合は、このフィールドを 1 に設定します。(アップストリーム PLU-SLU セッション統計では、このパラメーターは予約済みです。)

ls_name

統計に関連したリンク・ステーション名。この名前は、8 バイトからなる ASCII 文字ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。(アップストリーム PLU-SLU セッション統計では、*dspu_services* を AP_PU_CONCENTRATION に設定した場合はこのパラメーターは予約済みです。)

DOWNSTREAM_PU_INDICATION

この指示は、ダウンストリーム PU とホスト間の PU-SSCP セッションの状態が活動と非活動の間で変わると生成されます。セッションが非活動状態になると、指示には PU-SSCP セッション統計も組み込まれます。

VCB 構造体

```
typedef struct downstream_pu_indication
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;               /* reserved                      */
    unsigned char  format;                /* reserved                      */
    AP_UINT16      primary_rc;            /* primary return code          */
    AP_UINT32      secondary_rc;         /* secondary return code        */
    unsigned char  data_lost;             /* previous indication lost     */
    unsigned char  dspu_name[8];         /* PU name                      */
    unsigned char  description[32];      /* resource description          */
    unsigned char  reserv3[16];          /* reserved                      */
    unsigned char  ls_name[8];           /* Link station name            */
    unsigned char  pu_sscp_sess_active;  /* Is PU-SSCP session active    */
    unsigned char  dspu_services;        /* DSPU services                */
    unsigned char  reserv1[2];           /* reserved                      */
    SESSION_STATS  pu_sscp_stats;        /* PU-SSCP session statistics   */
} DOWNSTREAM_PU_INDICATION;

typedef struct session_stats
{
    AP_UINT16      rcv_ru_size;           /* session receive RU size      */
    AP_UINT16      send_ru_size;         /* session send RU size         */
    AP_UINT16      max_send_btu_size;    /* maximum send BTU size        */
    AP_UINT16      max_rcv_btu_size;     /* maximum rcv BTU size         */
    AP_UINT16      max_send_pac_win;     /* maximum send pacing window  */
    AP_UINT16      cur_send_pac_win;     /* current send pacing window   */
    AP_UINT16      max_rcv_pac_win;      /* maximum receive pacing window */
    AP_UINT16      cur_rcv_pac_win;      /* current receive pacing window */
    AP_UINT32      send_data_frames;     /* number of data frames sent   */
    AP_UINT32      send_fmd_data_frames; /* num fmd data frames sent    */
    AP_UINT32      send_data_bytes;      /* number of data bytes sent    */
    AP_UINT32      rcv_data_frames;      /* number of data frames received */
    AP_UINT32      rcv_fmd_data_frames;  /* num fmd data frames received */
    AP_UINT32      rcv_data_bytes;       /* number of data bytes received */
    unsigned char  sidh;                 /* session ID high byte (from LFSID) */
    unsigned char  sidl;                 /* session ID low byte (from LFSID) */
    unsigned char  odai;                 /* ODAI bit set                 */
    unsigned char  ls_name[8];           /* Link station name            */
    unsigned char  reserve;              /* reserved                      */
} SESSION_STATS;
```

パラメーター

opcode AP_DOWNSTREAM_PU_INDICATION

primary_rc
AP_OK

data_lost

既存のダウンストリーム PU 指示の中に消失したのがあるかどうかを指定します。 Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のダウンストリーム PU 指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のダウンストリーム PU 指示は消失していません。

dspu_name

ダウンストリーム PU の名前。この名前は、8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

description

ダウンストリーム PU を記述するヌルで終了したテキスト・String (PU に関連した LS の定義で指定したもの)。

ls_name

ダウンストリーム PU に関連したリンク・ステーションの名前。これは 8 バイトからなる ASCII Stringで、名前が 8 文字に満たない場合、右側にスペースを入れます。

pu_sscp_sess_active

ダウンストリーム PU との PU-SSCP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

dspu_services

ローカル・ノードがダウンストリーム PU に提供するサービスを指定します。

値は次のいずれかです。

AP_PU_CONCENTRATION

ダウンストリーム LU は SNA ゲートウェイからサービスを受けます。

AP_DLUR

ダウンストリーム LU は DLUR からサービスを受けます。

pu_sscp_stats.rcv_ru_size

予約済み (常にゼロに設定される)。

pu_sscp_stats.send_ru_size

予約済み (常にゼロに設定される)。

pu_sscp_stats.max_send_btu_size

送信できる BTU の最大サイズ。

pu_sscp_stats.max_rcv_btu_size

受信できる BTU の最大サイズ。

pu_sscp_stats.max_send_pac_win

予約済み (常にゼロに設定される)。

pu_sscp_stats.cur_send_pac_win

予約済み (常にゼロに設定される)。

DOWNSTREAM_PU_INDICATION

pu_sscp_stats.max_rcv_pac_win
予約済み (常にゼロに設定される)。

pu_sscp_stats.cur_rcv_pac_win
予約済み (常にゼロに設定される)。

pu_sscp_stats.send_data_frames
送信された通常フロー・データ・フレームの数。

pu_sscp_stats.send_fmd_data_frames
送信された通常フロー FMD データ・フレームの数。

pu_sscp_stats.send_data_bytes
送信された通常フロー・データ・バイトの数。

pu_sscp_stats.rcv_data_frames
受信された通常フロー・データ・フレームの数。

pu_sscp_stats.rcv_fmd_data_frames
受信された通常フロー FMD データ・フレームの数。

pu_sscp_stats.rcv_data_bytes
受信された通常フロー・データ・バイトの数。

pu_sscp_stats.sidh
セッション ID の上位バイト。

pu_sscp_stats.sidl
セッション ID の下位バイト。

pu_sscp_stats.odai
起点宛先アサイナー・インディケータ。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合、BIND の送信側は、このフィールドをゼロに設定し、BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合は、このフィールドを 1 に設定します。

pu_sscp_stats.ls_name
統計に関連したリンク・ステーション名。この名前は、8 バイトからなる ASCII 文字ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

FOCAL_POINT_INDICATION

この指示は、フォーカル・ポイントを獲得したとき、変更したとき、または取り消したときに必ず生成されます。

VCB 構造体

```
typedef struct focal_point_indication
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;          /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  data_lost;       /* previous indication lost     */
    unsigned char  ms_category[8];  /* Focal point category         */
    unsigned char  fp_fqcp_name[17]; /* Fully qualified focal point cp name*/
    unsigned char  ms_appl_name[8]; /* Focal point application name */
}
```

```

unsigned char  fp_type;           /* type of current focal point */
unsigned char  fp_status;        /* status of focal point */
unsigned char  fp_routing;       /* type of MDS routing to reach FP */
unsigned char  reserva[20];     /* reserved */
} FOCAL_POINT_INDICATION;

```

パラメーター

opcode AP_FOCAL_POINT_INDICATION

primary_rc
AP_OK

data_lost

既存のフォーカル・ポイント指示の中に消失したものがどうかを指定します。 Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のフォーカル・ポイント指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のフォーカル・ポイント指示は消失していません。

ms_category

フォーカル・ポイントが変更された Management Services カテゴリー。これは、「*SNA Management Services Reference*」に記載された『MS Discipline-Specific Application Programs』表に指定されたカテゴリー名の 1 つ (名前が 8 バイトに満たない場合は、右側にスペースを入れます) か、ユーザー定義のカテゴリーです。ユーザー定義のカテゴリー名は 8 バイトからなるタイプ 1134 の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

fp_fqcp_name

指定された MS カテゴリーの場合の現行フォーカル・ポイントの完全修飾名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。このパラメーターの 17 桁の 2 進ゼロに設定されている場合は、指定された MS カテゴリーには現在フォーカル・ポイントが定義されていないことを示します。既存のフォーカル・ポイントは削除されたのであって、置換されてはいません。

ms_appl_name

現行フォーカル・ポイント・アプリケーションの名前。これは、「*Systems Network Architecture: Management Services Reference SC30-3346*」に記載された『MS Discipline-Specific Application Programs』表に指定されたアプリケーション名の 1 つ (名前が 8 バイトに満たない場合は、右側にスペースを入れます) か、ユーザー定義のアプリケーション名です (xxiii ページの『関連資料』を参照)。ユーザー定義名は 8 バイトからなるタイプ 1134 の EBCDIC ストリングで、名前が 8 バイトに満たない場合、右側にスペース

FOCAL_POINT_INDICATION

を入れます。このパラメーターがゼロに設定されている場合は、指定された MS カテゴリには現在フォーカル・ポイントが定義されていないことを示します。既存のフォーカル・ポイントは削除されたのであって、置換されてはいません。

fp_type フォーカル・ポイントのタイプ。詳しくは、「SNA Management Services」を参照してください。値は次のいずれかです。

AP_EXPLICIT_PRIMARY_FP

AP_IMPLICIT_PRIMARY_FP

AP_BACKUP_FP

AP_DEFAULT_PRIMARY_FP

AP_DOMAIN_FP

AP_HOST_FP

AP_NO_FP

fp_status

フォーカル・ポイントの状況。値は次のいずれかです。

AP_NOT_ACTIVE

フォーカル・ポイントは活動状態から非活動状態に変わりました。

AP_ACTIVE

フォーカル・ポイントは非活動状態または保留活動状態から活動状態に変わりました。

fp_routing

フォーカル・ポイントにデータを送信するときにアプリケーションが指定する経路指定のタイプ。このパラメーターはフォーカル・ポイント状況が AP_ACTIVE の場合のみ使用されます。値は次のいずれかです。

AP_DEFAULT

デフォルト経路指定を使用してデータを送信します。

AP_DIRECT

直接経路指定を使用してデータを送信します。

ISR_INDICATION

この指示は、中間セッション経路指定 (ISR) セッションが活動化または非活動化されると生成されます。セッションが非活動化された場合は、セッションの使用状況に関する統計情報が戻りデータに組み込まれます。

VCB 構造体

```
typedef struct isr_indication
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code        */
    unsigned char  data_lost;     /* previous indication lost     */
    unsigned char  deactivated;    /* has ISR session been         */
}
```

```

/* deactivated? */
FQPCID      fqpcid;          /* FQPCID for ISR session */
unsigned char fqplu_name[17]; /* fully-qualified primary LU name */
unsigned char fqslu_name[17]; /* fully-qualified secondary LU name */
/* mode name */
unsigned char mode_name[8];   /* mode name */
/* COS name */
unsigned char cos_name[8];    /* COS name */
/* transmission priority */
unsigned char transmission_priority;
AP_UINT32   sense_data;      /* sense data */
unsigned char reserv2a[2];    /* reserved */
SESSION_STATS pri_sess_stats; /* Primary hop session statistics */
SESSION_STATS sec_sess_stats; /* Secondary hop session statistics */
unsigned char reserva[20];    /* reserved */
} ISR_INDICATION;

typedef struct fqpcid
{
  unsigned char pcid[8];      /* procedure correlator identifier */
  unsigned char fqcp_name[17]; /* originator's network qualified CP name */
  unsigned char reserve3[3];  /* reserved */
} FQPCID;

typedef struct session_stats
{
  AP_UINT16   rcv_ru_size;    /* session receive RU size */
  AP_UINT16   send_ru_size;   /* session send RU size */
  AP_UINT16   max_send_btu_size; /* maximum send BTU size */
  AP_UINT16   max_rcv_btu_size; /* maximum rcv BTU size */
  AP_UINT16   max_send_pac_win; /* maximum send pacing window size */
  AP_UINT16   cur_send_pac_win; /* current send pacing window size */
  AP_UINT16   max_rcv_pac_win; /* maximum receive pacing window size */
  AP_UINT16   cur_rcv_pac_win; /* current receive pacing window size */
  AP_UINT32   send_data_frames; /* number of data frames sent */
  AP_UINT32   send_fmd_data_frames; /* num fmd data frames sent */
  AP_UINT32   send_data_bytes; /* number of data bytes sent */
  AP_UINT32   rcv_data_frames; /* number of data frames received */
  AP_UINT32   rcv_fmd_data_frames; /* num fmd data frames received */
  AP_UINT32   rcv_data_bytes; /* number of data bytes received */
  unsigned char sidh;          /* session ID high byte (from LFSID) */
  unsigned char sidl;          /* session ID low byte (from LFSID) */
  unsigned char oda;          /* ODAI bit set */
  unsigned char ls_name[8];    /* Link station name */
  unsigned char reserve;      /* reserved */
} SESSION_STATS;

```

パラメーター

opcode AP_ISR_INDICATION

primary_rc

AP_OK

data_lost

既存の ISR 指示の中に消失したものがどうかを指定します。

Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の ISR 指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存の ISR 指示は消失していません。

deactivated

ISR セッションが非活動化されたか活動化されたかを指定します。値は次のいずれかです。

AP_YES セッションは非活動化されました。

AP_NO セッションは活動化されました。

fqpcid.pcid

セッションのプロシージャ相互関係子 ID。これは 8 バイトからなる 16 進数ストリングです。

fqpcid.fqcp_name

制御点の完全修飾名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

fqplu_name

このセッションの 1 次 LU の完全修飾名。*deactivated* を AP_YES に設定した場合、このパラメーターは予約済みです。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

fqslu_name

このセッションの 2 次 LU の完全修飾名。*deactivated* を AP_YES に設定した場合、このパラメーターは予約済みです。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

mode_name

このセッションのモード名。*deactivated* を AP_YES に設定した場合、このパラメーターは予約済みです。これは 8 バイトからなるタイプ A の EBCDIC ストリング (英文字で始まる) で、右側に EBCDIC スペースを入れます。

cos_name

このセッションの COS 名。*deactivated* を AP_YES に設定した場合、このパラメーターは予約済みです。これは 8 バイトからなるタイプ A の EBCDIC ストリング (英文字で始まる) で、右側に EBCDIC スペースを入れます。

transmission_priority

セッションに関連した伝送優先順位。*deactivated* を AP_YES に設定した場合、このパラメーターは予約済みです。

sense_data

UNBIND 要求で送信または受信されたセンス・データ。*deactivated* を AP_NO に設定した場合、このパラメーターは予約済みです。

ISR セッションが非活動化された場合は、1 次セッションおよび 2 次セッションについて `session_stats` 構造体が組み込まれます。この構造体のフィールドは次のとおりです。

rcv_ru_size

受信 RU の最大サイズ。

send_ru_size

送信 RU の最大サイズ。

max_send_btu_size

送信できる BTU の最大サイズ。

max_rcv_btu_size

受信できる BTU の最大サイズ。

max_send_pac_win

このセッションの送信ペーシング・ウィンドウの最大サイズ。

cur_send_pac_win

このセッションの送信ペーシング・ウィンドウの現行サイズ。

max_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの最大サイズ。

cur_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの現行サイズ。

send_data_frames

送信された通常フロー・データ・フレームの数。

send_fmd_data_frames

送信された通常フロー FMD データ・フレームの数。

send_data_bytes

送信された通常フロー・データ・バイトの数。

rcv_data_frames

受信された通常フロー・データ・フレームの数。

rcv_fmd_data_frames

受信された通常フロー FMD データ・フレームの数。

rcv_data_bytes

受信された通常フロー・データ・バイトの数。

sidh セッション ID の上位バイト。

sidl セッション ID の下位バイト。

odai 起点宛先アサイナー・インディケータ。セッションの起動時に、BIND の送信側は、ローカル・ノードに 1 次リンク・ステーションが含まれる場合はこのフィールドをゼロに設定し、ローカル・ノードに 2 次リンク・ステーションが含まれる場合はこのフィールドを 1 に設定します。

ls_name

統計に関連したリンク・ステーション名。これは 8 バイトからなる ASCII スtringで、名前が 8 文字に満たない場合、右側にスペースを入れます。この名前は、セッションのトラフィックの転送経路となるリンクとセッション統計を関連付けるために使用できます。

LOCAL_LU_INDICATION

この指示は、ローカル LU が定義されたり削除されると生成されます。この指示は、現在定義されているすべてのローカル LU のリストを維持するために、登録されたアプリケーションで使用できます。

VCB 構造体

```
typedef struct local_lu_indication
{
    AP_UINT16      opcode;           /* Indication operation code      */
    unsigned char  reserv2;         /* reserved                        */
    unsigned char  format;         /* reserved                        */
    AP_UINT16      primary_rc;      /* primary return code            */
    AP_UINT32      secondary_rc;    /* secondary return code          */
    unsigned char  data_lost;       /* Previous indication lost?      */
    unsigned char  reason;         /* reason for this indication     */
    unsigned char  lu_name[8];     /* LU name                        */
    DESCRIPTION    description;     /* resource description           */
    unsigned char  lu_alias[8];    /* LU alias                       */
    unsigned char  nau_address;     /* NAU address                    */
    unsigned char  reserv4;         /* reserved                        */
    unsigned char  pu_name[8];     /* PU name                        */
    unsigned char  lu_sscp_active;  /* Is LU-SSCP session active     */
    unsigned char  reserv5;         /* reserved                        */
    SESSION_STATS  lu_sscp_stats;   /* LU-SSCP session statistics     */
    unsigned char  sscp_id[6];     /* SSCP ID                        */
} LOCAL_LU_INDICATION;

typedef struct session_stats
{
    AP_UINT16      rcv_ru_size;     /* session receive RU size        */
    AP_UINT16      send_ru_size;    /* session send RU size           */
    AP_UINT16      max_send_btu_size; /* max send BTU size             */
    AP_UINT16      max_rcv_btu_size; /* max receive BTU size          */
    AP_UINT16      max_send_pac_win; /* max send pacing window size   */
    AP_UINT16      cur_send_pac_win; /* current send pacing window size */
    AP_UINT16      max_rcv_pac_win; /* max receive pacing window size */
    AP_UINT16      cur_rcv_pac_win; /* current rcv pacing window size */
    AP_UINT32      send_data_frames; /* number of data frames sent     */
    AP_UINT32      send_fmd_data_frames; /* num of fmd data frames sent */
    AP_UINT32      send_data_bytes; /* number of data bytes sent      */
    AP_UINT32      rcv_data_frames; /* number of data frames received */
    AP_UINT32      rcv_fmd_data_frames; /* num of fmd data frames received */
    AP_UINT32      rcv_data_bytes; /* number of data bytes received  */
    unsigned char  sidh;           /* session ID high byte           */
    unsigned char  sidl;           /* session ID low byte            */
    unsigned char  odai;          /* *origin-destination assignor bit set*/
    unsigned char  ls_name[8];     /* link station name              */
    unsigned char  pacing_type;    /* type of pacing in use         */
} SESSION_STATS;
```

session_stats 構造体に含まれる LU-SSCP 統計は、*nau_address* パラメーターがゼロ以外の値に設定され、なおかつ *lu_sscp_active* パラメーターが AP_YES に設定されている場合のみ有効です。それ以外の場合は、session_stats 構造体のパラメーターは予約済みです。

パラメーター

```
opcode AP_LOCAL_LU_INDICATION

primary_rc
    AP_OK
```

data_lost

既存のディレクトリ指示の中に消失したものがあるかどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のディレクトリ指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のディレクトリ指示は消失していません。

reason 指示の理由コード。値は次のいずれかです。

AP_ADDED

LU は定義されました。

AP_REMOVED

LU は、DELETE_LOCAL_LU を使用して明示的に、あるいは DELETE_LS、DELETE_PORT、または DELETE_DLC を使用して暗黙的に削除されました。

AP_SSCP_ACTIVE

ノードが ACTLU を正常に処理したあと、LU-SSCP セッションは活動状態になりました。

AP_SSCP_INACTIVE

正常な DACTLU またはリンク障害のあと、LU-SSCP セッションは非活動状態になりました。

lu_name

状態が変更されたローカル論理装置 (LU) の名前。これは 8 バイトの英数字からなる (英字で始まる) タイプ A の EBCDIC スtring で、右側に EBCDIC のスペースを入れます。

description

DEFINE_LOCAL_LU で指定したリソース記述。

lu_alias

ローカルで定義された LU 別名。これは 8 バイトからなるローカル・システム上で表示可能な文字セットの String です。8 バイトはすべて有効です。

nau_address

LU のネットワーク・アクセス可能単位 (NAU) アドレス。この値の範囲は 1 ~ 255 です。ゼロ以外の値は LU が従属 LU であることを暗黙指定します。値 0 (ゼロ) は LU が独立 LU であることを暗黙指定します。

pu_name

この LU が使用する物理装置 (PU) の名前。この名前は、8 バイトからなるタイプ A の EBCDIC String で、右側に EBCDIC スペースを入れます。このパラメーターは、*nau_address* パラメーターを 0 (ゼロ) に設定しない場合のみ有効です。*nau_address* パラメーターを 0 に設定した場合は、*pu_name* パラメーターはすべてゼロに設定されます。

LOCAL_LU_INDICATION

lu_sscp_sess_active

LU-SSCP セッションが活動状態であるかどうかを指定します。 *nau_address* パラメーターを 0 (ゼロ) に設定した場合は、このパラメーターは予約済みです。値は次のいずれかです。

AP_YES LU-SSCP セッションは活動状態です。

AP_NO LU-SSCP セッションは活動状態ではありません。

lu_sscp_stats.rcv_ru_size

このパラメーターは常に予約済みです。

lu_sscp_stats.send_ru_size

このパラメーターは常に予約済みです。

lu_sscp_stats.max_send_btu_size

送信できる基本伝送単位 (BTU) の最大サイズ。

lu_sscp_stats.max_rcv_btu_size

受信できる基本伝送単位 (BTU) の最大サイズ。

lu_sscp_stats.max_send_pac_win

このパラメーターは常にゼロに設定されます。

lu_sscp_stats.cur_send_pac_win

このパラメーターは常にゼロに設定されます。

lu_sscp_stats.max_rcv_pac_win

このパラメーターは常にゼロに設定されます。

lu_sscp_stats.cur_rcv_pac_win

このパラメーターは常にゼロに設定されます。

lu_sscp_stats.send_data_frames

送信された通常フロー・データ・フレームの数。

lu_sscp_stats.send_fmd_data_frames

送信された通常フロー機能管理データ (FMD) フレームの数。

lu_sscp_stats.send_data_bytes

送信された通常フロー・データ・バイトの数。

lu_sscp_stats.rcv_data_frames

受信された通常フロー・データ・フレームの数。

lu_sscp_stats.rcv_fmd_data_frames

受信された通常フロー機能管理データ (FMD) フレームの数。

lu_sscp_stats.rcv_data_bytes

受信された通常フロー・データ・バイトの数。

lu_sscp_stats.sidh

セッション ID の上位バイト。

lu_sscp_stats.sidl

セッション ID の下位バイト。

lu_sscp_stats.odai

起点宛先アサイナー・インディケーター。 ACTLU の送信側は、セッションを活動化するとき、ローカル・ノードに 1 次リンク・ステーションが含ま

まれる場合はこのパラメーターをゼロに設定し、ACTLU 送信側のノードに 2 次リンク・ステーションが含まれる場合はこのパラメーターを 1 に設定します。

lu_sscp_stats.ls_name

統計に関連したリンク・ステーション名。これは 8 バイトからなるローカル・システム上で表示可能な文字セットのストリングです。8 バイトはすべて有効です。このパラメーターは、セッションの転送経路となるリンクとこのセッションを関連付けるために使用できます。

lu_sscp_stats.pacing_type

LU-SSCP セッションで使用している受信ペーシングのタイプ。値は AP_NONE です。

sscp_id この LU が使用する PU の ACTPU で受信された SSCP の ID。このパラメーターは 6 バイトからなり、従属 LU のみが使用します。独立 LU の場合、このパラメーターはすべてゼロに設定されます。また、*lu_sscp_sess_active* パラメーターが AP_YES に設定されていない場合も、このパラメーターはすべてゼロに設定されます。

LOCAL_TOPOLOGY_INDICATION

この指示は、次のいずれかが起こったときに生成されます。

- ローカル・ノードのトポロジー・データベースにある TG の状態が、活動から非活動、あるいはその逆へと変更された。
- ローカル・ノードのトポロジー・データベースにある TG の状態が、静止から非静止、あるいはその逆へと変更された。
- ローカル・ノードのトポロジー・データベースにある TG 上のコンテンション勝者 CP-CP セッションが、活動化または非活動化された。

VCB 構造体

```
typedef struct local_topology_indication
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  data_lost;     /* previous indication lost    */
    unsigned char  status;        /* TG status                   */
    unsigned char  dest[17];      /* name of TG destination node */
    unsigned char  dest_type;     /* TG destination node type    */
    unsigned char  tg_num;        /* TG number                   */
    unsigned char  cp_cp_session_active; /* CP-CP sessions active? */
    unsigned char  branch_link_type; /* Up or down link?          */
    unsigned char  branch_tg;     /* Branch TG?                  */
    unsigned char  reserva[17];   /* reserved                     */
} LOCAL_TOPOLOGY_INDICATION;
```

パラメーター

opcode AP_LOCAL_TOPOLOGY_INDICATION

primary_rc
AP_OK

LOCAL_TOPOLOGY_INDICATION

data_lost

既存のローカル・トポロジー指示の中に消失したのがあるかどうかを指定します。 Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のローカル・トポロジー指示 (1 つ以上) が消失しました。

AP_NO 既存のローカル・トポロジー指示は消失していません。

status TG の状況を指定します。 AP_NONE に設定するか、次の値の 1 つ以上 (論理 OR を使用して組み合わせる) に設定してください。

AP_TG_OPERATIVE

AP_TG_CP_CP_SESSIONS

AP_TG QUIESCING

dest TG の宛先ノードの完全修飾名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

dest_type

宛先ノードのタイプ。値は次のいずれかです。

AP_END_NODE

エンド・ノード。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_VRN 仮想経路指定ノード。

tg_num TG に関連した伝送グループ番号。

cp_cp_session_active

ローカル・ノードのコンテンツ勝者 CP-CP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES CP-CP セッションは活動状態です。

AP_NO CP-CP セッションは活動状態ではありません。

AP_UNKNOWN

CP-CP セッションの状況は不明です。

branch_link_type

このパラメーターは、ノードが分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

この TG の分岐リンク・タイプを指定します。値は次のいずれかです。

AP_UPLINK

TG はアップリンクです

AP_DOWNLINK

TG はエンド・ノードへのダウンリンクです。

AP_DOWNLINK_TO_BRNN

TG は、ローカル・ノードの全体像のエンド・ノードとして表示される分岐ネットワーク・ノードへのダウンリンクです。

AP_OTHERLINK

TG は VRN へのリンクです。

branch_tg

このパラメーターは、ノードがネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

TG が分岐 TG であるかどうかを指定します。値は次のいずれかです。

AP_YES TG は分岐 TG です。

AP_NO TG は分岐 TG ではありません。

AP_UNKNOWN

TG のタイプは不明です。

LS_INDICATION

この指示は、リンク・ステーションが活動化または非活動化されると生成されます。リンク・ステーションが非活動化された場合は、リンク・ステーションの使用状況の統計が戻りデータに組み込まれます。

VCB 構造体

```
typedef struct ls_indication
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;              /* reserved                      */
    unsigned char  format;               /* reserved                      */
    AP_UINT16      primary_rc;           /* primary return code          */
    AP_UINT32      secondary_rc;         /* secondary return code        */
    unsigned char  data_lost;            /* previous indication lost     */
    unsigned char  deactivated;          /* has LS been deactivated?     */
    unsigned char  ls_name[8];           /* link station name            */
    unsigned char  description[32];      /* resource description         */
    unsigned char  reserv1[16];          /* reserved                      */
    unsigned char  adj_cp_name[17];      /* network qualified Adjacent CP name*/
    unsigned char  adj_node_type;        /* adjacent node type          */
    AP_UINT16      act_sess_count;       /* active session count on link  */
    unsigned char  indication_cause;     /* cause of indication         */
    LS_STATS       ls_stats;             /* link station statistics      */
    unsigned char  tg_num;               /* tg number                    */
    AP_UINT32      sense_data;           /* sense data                   */
    unsigned char  brnn_link_type;       /* type of branch link         */
    unsigned char  adj_cp_is_brnn;       /* is adjacent node a BrNN?    */
    unsigned char  mltg_member;          /* reserved                      */
    unsigned char  tg_sharing;           /* reserved                      */
    unsigned char  ls_type;              /* how LS was defined or discovered */
    unsigned char  reserva[14];          /* reserved                      */
} LS_INDICATION;

typedef struct ls_stats
{
    AP_UINT32      in_xid_bytes;          /* number of XID bytes received */
    AP_UINT32      in_msg_bytes;         /* number of message bytes received */
    AP_UINT32      in_xid_frames;        /* number of XID frames received */
    AP_UINT32      in_msg_frames;        /* number of message frames received*/
    AP_UINT32      out_xid_bytes;        /* number of XID bytes sent     */
    AP_UINT32      out_msg_bytes;        /* number of message bytes sent  */
}
```

LS_INDICATION

```
AP_UINT32    out_xid_frames;        /* number of XID frames sent */
AP_UINT32    out_msg_frames;        /* number of message frames sent */
AP_UINT32    in_invalid_sna_frames; /* number of invalid
                                     /* frames received */
AP_UINT32    in_session_control_frames; /* number of control
                                     /* frames received */
AP_UINT32    out_session_control_frames; /* number of control
                                     /* frames sent */
AP_UINT32    echo_rsps;            /* reserved */
AP_UINT32    current_delay;        /* reserved */
AP_UINT32    max_delay;            /* reserved */
AP_UINT32    min_delay;            /* reserved */
AP_UINT32    max_delay_time;       /* reserved */
AP_UINT32    good_xids;            /* successful XID on LS count */
AP_UINT32    bad_xids;             /* unsuccessful XID on LS count */
} LS_STATS;
```

パラメーター

opcode AP_LS_INDICATION

primary_rc

AP_OK

data_lost

既存の LS 指示の中に消失したものがどうかを指定します。

Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の LS 指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存の LS 指示は消失していません。

deactivated

LS が非活動化されたか活動化されたかを指定します。値は次のいずれかです。

AP_YES LS は非活動化されました。

AP_NO LS は活動化されました。

ls_name

リンク・ステーションの名前。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

description

LS を記述したヌルで終了するテキスト・ストリング (その LS の定義で指定したもの)。

adj_cp_name

隣接ノードの完全修飾 CP 名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

adj_node_type

隣接ノードのタイプ。値は次のいずれかです。

AP_END_NODE

エンド・ノード。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_LEN_NODE

LEN ノード。

AP_VRN 仮想経路指定ノード。*act_sess_count*

リンクを使用している活動セッション (エンドポイント・セッションと中間セッションの両方) の合計数。

indication_cause

指示の原因。値は次のいずれかです。

AP_ACTIVATING

LS は活動化されました。

AP_DEACTIVATION_STARTED

LS の非活動化処理が開始されました。

AP_DEACTIVATING

LS は非活動化されました。

AP_SESS_COUNT_CHANGING

LS を使用している活動セッションの数が変わりました。

AP_CP_NAME_CHANGING

隣接ノードの CP 名が変わりました。

AP_DATA_LOST

既存の指示を送信できませんでした。

AP_FAILED

LS に障害が起きました。

AP_ACTIVATION_STARTED

LS は自動活動化をサポートしており、セッションで必要になったときに自動的に始動されました。

AP_ACTIVATION_FAILED

LS は自動活動化をサポートしていますが、必要になったときに自動的に始動しようとして失敗しました。

AP_LR_ACTIVATING

LS に障害が起こり (または失敗した LS を活動化しようとして)、Communications Server for Linux が LS を再活動化しようとしています。

次のパラメーターは、`deactivated` が `AP_YES` に設定されている場合のみ戻され、LS が非活動化されたことを示します。

LS_INDICATION

ls_stats.in_xid_bytes

このリンク・ステーションで受信した XID (交換識別) バイトの合計数。

ls_stats.in_msg_bytes

このリンク・ステーションで受信したデータ・バイトの合計数。

ls_stats.in_xid_frames

このリンク・ステーションで受信した XID (交換識別) フレームの合計数。

ls_stats.in_msg_frames

このリンク・ステーションで受信したデータ・フレームの合計数。

ls_stats.out_xid_bytes

このリンク・ステーションで送信した XID (交換識別) バイトの合計数。

ls_stats.out_msg_bytes

このリンク・ステーションで送信したデータ・バイトの合計数。

ls_stats.out_xid_frames

このリンク・ステーションで送信した XID (交換識別) フレームの合計数。

ls_stats.out_msg_frames

このリンク・ステーションで送信したデータ・フレームの合計数。

ls_stats.in_invalid_sna_frames

このリンク・ステーションで受信して、有効でなかった SNA フレームの合計数。

ls_stats.in_session_control_frames

このリンク・ステーションで受信したセッション制御フレームの合計数。

ls_stats.out_session_control_frames

このリンク・ステーションで送信したセッション制御フレームの合計数。

ls_stats.good_xids

このリンク・ステーションで始動したあとに発生して成功した XID 交換の合計数。

ls_stats.bad_xids

このリンク・ステーションで始動したあとに発生して失敗した XID 交換の合計数。

tg_num LS に関連した伝送グループ番号。

sense_data

LS が XID プロトコル・エラーのために失敗した場合は、このパラメーターにはエラーに関連したセンス・データが入ります。 *indication_cause* が AP_FAILED 以外の値に設定されている場合は、このパラメーターは予約済みです。

brnn_link_type

このパラメーターは、ローカル・ノードが分岐ネットワーク・ノードである場合にのみ適用されます。それ以外の場合には、予約済みです。

このリンクの分岐リンク・タイプを指定します。値は次のいずれかです。

AP_UPLINK

リンクはアップリンクです。

AP_DOWNLINK

リンクはダウンリンクです。

AP_OTHERLINK

VRN へのリンクです。

AP_UNKNOWN_LINK_TYPE

分岐リンク・タイプは不明です。

AP_BRNN_NOT_SUPPORTED

リンクは PU 2.0 トラフィックのみをサポートします。

adj_cp_is_brnn

隣接ノードが分岐ネットワーク・ノードであるかどうかを指定します。値は次のいずれかです。

AP_YES 隣接ノードは分岐ネットワーク・ノードです。

AP_NO 隣接ノードは分岐ネットワーク・ノードではありません。

AP_UNKNOWN

隣接ノードのタイプは不明です。

ls_type このリンクが定義された方法または検出された方法を指定します。値は次のいずれかです。

AP_LS_DEFINED

リンク・ステーションは、Communications Server for Linux 管理プログラムによって、明示的に定義されました。

AP_LS_DYNAMIC

リンク・ステーションは、ローカル・ノードが接続ネットワークを通じて他のノードへ接続したときに作成されました。

AP_LS_TEMPORARY

リンク・ステーションは着呼を処理するために一時的に作成されましたが、まだ活動状態になっていません。

AP_LS_IMPLICIT

リンク・ステーションは、定義済みのリンク・ステーションと一致しなかった着呼を Communications Server for Linux が受け取ったときに、暗黙的に定義されました。

AP_LS_DLUS_DEFINED

リンク・ステーションは、DLUR サービスを受けるダウンストリーム PU へのダイナミック・リンク・ステーションであり、ローカル・ノードが DLUS から ACTPU を受信したときに定義されました。

LU_0_TO_3_INDICATION

この指示は、タイプ 0-3 LU のセッション状況が変わると生成されます。

VCB 構造体

```
typedef struct lu_0_to_3_indication
{
    AP_UINT16          opcode;          /* verb operation code */

```

LU_0_TO_3_INDICATION

```
unsigned char    reserv2;           /* reserved */
unsigned char    format;           /* reserved */
AP_UINT16       primary_rc;        /* primary return code */
AP_UINT32       secondary_rc;      /* secondary return code */
unsigned char    data_lost;        /* previous indication lost */
unsigned char    pu_name[8];       /* PU Name */
unsigned char    lu_name[8];       /* LU Name */
unsigned char    description[32];   /* resource description */
unsigned char    reserv1[16];      /* reserved */
unsigned char    nau_address;       /* NAU address */
unsigned char    lu_sscp_sess_active; /* Is SSCP session active? */
unsigned char    appl_conn_active; /* Is application using LU? */
unsigned char    plu_sess_active;  /* Is PLU-SLU session active? */
unsigned char    host_attachment;  /* Host attachment */
SESSION_STATS   lu_sscp_stats;     /* LU-SSCP session statistics */
SESSION_STATS   plu_stats;         /* PLU session statistics */
unsigned char    sscp_id[6];       /* SSCP id */
} LU_0_TO_3_INDICATION;

typedef struct session_stats
{
    AP_UINT16     rcv_ru_size;       /* session receive RU size */
    AP_UINT16     send_ru_size;      /* session send RU size */
    AP_UINT16     max_send_btu_size; /* maximum send BTU size */
    AP_UINT16     max_rcv_btu_size;  /* maximum rcv BTU size */
    AP_UINT16     max_send_pac_win;  /* maximum send pacing window size */
    AP_UINT16     cur_send_pac_win;  /* current send pacing window size */
    AP_UINT16     max_rcv_pac_win;   /* maximum receive pacing window
    /* size */
    AP_UINT16     cur_rcv_pac_win;   /* current receive pacing window
    /* size */

    AP_UINT32     send_data_frames;  /* number of data frames sent */
    AP_UINT32     send_fmd_data_frames; /* num fmd data frames sent */
    AP_UINT32     send_data_bytes;   /* number of data bytes sent */
    AP_UINT32     rcv_data_frames;   /* number of data frames received */
    AP_UINT32     rcv_fmd_data_frames; /* num fmd data frames received */
    AP_UINT32     rcv_data_bytes;    /* number of data bytes received */
    unsigned char sidh;              /* session ID high byte (from LFSID) */
    unsigned char sidl;              /* session ID low byte (from LFSID) */
    unsigned char odai;              /* ODAI bit set */
    unsigned char ls_name[8];        /* Link station name */
    unsigned char reserve;           /* reserved */
} SESSION_STATS;
```

パラメーター

opcode AP_LU_0_TO_3_INDICATION

primary_rc

AP_OK

data_lost

既存の LU 0-3 指示の中に消失したものがどうかを指定します。

Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の LU 0-3 指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存の LU 0-3 指示は消失していません。

pu_name

LU が使用するローカル PU の名前。この名前は、8 バイトからなる (英字で始まる) タイプ A の EBCDIC スtringで、8 文字に満たない場合は、右側に EBCDIC スペースを入れます。

lu_name

セッション状況が変わった LU の名前。この名前は、8 バイトからなるタイプ A の EBCDIC スtringで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

description

LU を記述したヌルで終了するテキスト・String (その LU の定義で指定したもの)。

nau_address

LU のネットワーク・アクセス可能単位アドレス。

lu_sscp_sess_active

SSCP セッションが活動状態であるかどうか (つまり、ACTLU が正常に処理されたかどうか) を指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

appl_conn_active

この LU を使用しているアプリケーションがあるかどうかを指定します。値は次のいずれかです。

AP_YES LU を使用しているアプリケーションがあります。

AP_NO LU を使用しているアプリケーションはありません。

plu_sess_active

PLU-SLU セッションが活動化されたかどうかを指定します。値は次のいずれかです。

AP_YES セッションは活動状態です。

AP_NO セッションは活動状態ではありません。

host_attachment

LU ホスト接続タイプ。

値は次のいずれかです。

AP_DIRECT_ATTACHED

LU はホスト・システムに直接接続されています。

AP_DLUR_ATTACHED

LU は DLUR を使用してホスト・システムに接続されています。

sscp_id 従属 LU セッションの場合、このパラメーターは、ローカル LU がマップされる PU についてホストからの ACTPU で受信された SSCP ID です。独立 LU セッションの場合、このパラメーターは 0 (ゼロ) に設定されます。この値は、16 進値で表示される 6 バイトの配列です。

2 つのセッション (LU-SSCP セッションと PLU-SLU セッション) のそれぞれについて、*session_stats* 構造体が組み込まれます。セッションが活動状態から非活動

LU_0_TO_3_INDICATION

状態が変わった場合は、この構造体には次のパラメーターが入ります。それ以外の場合は、このパラメーターは予約されません。

rcv_ru_size

受信 RU の最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

send_ru_size

送信 RU の最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

max_send_btu_size

送信できる BTU の最大サイズ。

max_rcv_btu_size

受信できる BTU の最大サイズ。

max_send_pac_win

このセッションの送信ペーシング・ウィンドウの最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

cur_send_pac_win

このセッションの送信ペーシング・ウィンドウの現行サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

max_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの最大サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

cur_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの現行サイズ。(LU-SSCP セッション統計では、このパラメーターは予約済みです。)

send_data_frames

送信された通常フロー・データ・フレームの数。

send_fmd_data_frames

送信された通常フロー FMD データ・フレームの数。

send_data_bytes

送信された通常フロー・データ・バイトの数。

rcv_data_frames

受信された通常フロー・データ・フレームの数。

rcv_fmd_data_frames

受信された通常フロー FMD データ・フレームの数。

rcv_data_bytes

受信された通常フロー・データ・バイトの数。

sidh セッション ID の上位バイト。

sidl セッション ID の下位バイト。

odai 起点宛先アサイナー・インディケーター。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合、BIND の送信側は、このフィールドをゼロに設定し、BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合は、このフィールドを 1 に設定します。

ls_name

統計に関連したリンク・ステーション名。この名前は、8 バイトからなる ASCII 文字ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

MODE_INDICATION

この指示は、ローカル LU とパートナー LU が特定のモードを使用して通信を開始したとき、または LU-LU モードの組み合わせについて活動セッション・カウントが変わったときに送信されます。

VCB 構造体

```
typedef struct mode_indication
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;     /* primary return code         */
    AP_UINT32      secondary_rc;   /* secondary return code       */
    unsigned char  data_lost;      /* previous indication lost    */
    unsigned char  removed;       /* is entry being removed?    */
    unsigned char  lu_alias[8];    /* LU alias                    */
    unsigned char  plu_alias[8];   /* partner LU alias           */
    unsigned char  fqplu_name[17]; /* fully qualified partner LU name */
    unsigned char  mode_name[8];  /* mode name                  */
    unsigned char  description[32]; /* resource description       */
    unsigned char  reserv1[16];   /* reserved                   */
    AP_UINT16      curr_sess_count; /* current session count      */
    unsigned char  reserva[20];   /* reserved                   */
} MODE_INDICATION;
```

パラメーター

opcode AP_MODE_INDICATION

primary_rc
AP_OK

data_lost

既存のモード指示の中に消失したものがどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のモード指示 (1 つ以上) が消失しました。

AP_NO 既存のモード指示は消失していません。

removed

このパラメーターは現在使用されていません。モード指示は LU がモードの使用を開始したときのみ生成され、LU がモードの使用を停止したときには生成されません。

lu_alias

ローカルで定義された LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

MODE_INDICATION

plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

fqplu_name

パートナー LU の完全修飾名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

mode_name

1 グループのセッションのネットワーク特性を指定するモード名。これは 8 バイトからなる (英文字で始まる) タイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

description

モードを記述したヌルで終了するテキスト・ストリング (そのモードの定義で指定したもの)。

curr_sess_count

この LU-LU モードの組み合わせについて現在活動状態であるセッションの数。

NN_TOPOLOGY_NODE_INDICATION

この指示は、ネットワーク・ノードのトポロジー・データベースが活動化または非活動化されると生成されます。

VCB 構造体

```
typedef struct nn_topology_node_indication
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code        */
    unsigned char  data_lost;      /* previous indication lost     */
    unsigned char  deactivated;    /* has the node become inactive? */
    unsigned char  node_name[17]; /* node name                    */
    unsigned char  node_type;     /* node type                    */
    unsigned char  branch_aware;  /* is the node branch aware?   */
    unsigned char  reserva[19];   /* reserved                      */
} NN_TOPOLOGY_NODE_INDICATION;
```

パラメーター

opcode AP>NN_TOPOLOGY_NODE_INDICATION

primary_rc
AP_OK

data_lost

既存の NN トポロジー・ノード指示の中に消失したものがあるかどうかを指定します。 Communications Server for Linux は、指示の送信を妨げる条

件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の NN トポロジー・ノード指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存の NN トポロジー・ノード指示は消失していません。

deactivated

ノードが非活動化されたか活動化されたかを指定します。値は次のいずれかです。

AP_YES ノードは非活動化されました。

AP_NO ノードは活動化されました。

node_name

ネットワーク・トポロジー・データベースからのネットワーク修飾ノード名。この名前は 17 バイトの EBCDIC スtringで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A String文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A String文字からなるネットワーク名で構成されます。

node_type

ノードのタイプ。値は次のいずれかです。

AP_NETWORK_NODE

ネットワーク・ノード。

AP_VRN 仮想経路指定ノード。

branch_aware

ノードが分岐認識 (APPN オプション・セット 1120) をサポートするかどうかを指定します。

AP_NO ノードはオプション・セット 1120 をサポートしません。

AP_YES ノードはオプション・セット 1120 をサポートします。

NN_TOPOLOGY_TG_INDICATION

この指示は、ネットワーク・ノードのトポロジー・データベースで TG エントリーが活動化または非活動化されると生成されます。

VCB 構造体

```
typedef struct nn_topology_tg_indication
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;     /* primary return code          */
    AP_UINT32      secondary_rc;   /* secondary return code        */
    unsigned char  data_lost;     /* previous indication lost     */
    unsigned char  status;        /* TG status                    */
    unsigned char  owner[17];     /* name of TG owner node        */
    unsigned char  dest[17];     /* name of TG destination node  */
    unsigned char  tg_num;       /* TG number                    */
}
```

NN_TOPOLOGY_TG_INDICATION

```
    unsigned char    owner_type;           /* type of node that owns TG */
    unsigned char    dest_type;           /* TG destination node type */
    unsigned char    cp_cp_session_active; /* are CP-CP sessions active? */
    unsigned char    branch_tg;          /* is this a branch link? */
    unsigned char    multilink_tg;        /* reserved */
    unsigned char    reserva[15];         /* reserved */
} NN_TOPOLOGY_TG_INDICATION;
```

パラメーター

opcode AP_NN_TOPOLOGY_TG_INDICATION

primary_rc

AP_OK

data_lost

既存の NN トポロジー TG 指示の中に消失したものがあがるかどうかを指定します。 Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の NN トポロジー TG 指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存の NN トポロジー TG 指示は消失していません。

status TG の状況を指定します。 AP_NONE に設定するか、次の値の 1 つ以上 (論理 OR を使用して組み合わせる) に設定してください。

AP_TG_OPERATIVE

AP_TG_CP_CP_SESSIONS

AP_TG_QUIESCING

owner TG の起点ノードの名前 (Communications Server for Linux ローカル・ノード名)。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

dest TG の宛先ノードの完全修飾名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

tg_num TG に関連した伝送グループ番号。

owner_type

TG を所有するノードのタイプ。値は次のいずれかです。

AP_NETWORK_NODE

AP_VRN

dest_type

TG の宛先ノードのタイプ。値は次のいずれかです。

AP_NETWORK_NODE

AP_VRN

cp_cp_session_active

所有側ノードのコンテンション勝者 CP-CP セッションが活動状態であるかどうかを指定します。値は次のいずれかです。

AP_YES CP-CP セッションは活動状態です。

AP_NO CP-CP セッションは活動状態ではありません。

AP_UNKNOWN

CP-CP セッションの状況は不明です。

branch_tg

TG が分岐 TG であるかどうかを指定します。値は次のいずれかです。

AP_YES TG は分岐 TG です。

AP_NO TG は分岐 TG ではありません。

AP_UNKNOWN

TG のタイプは不明です。

NOF_STATUS_INDICATION

この指示は、接続されたターゲットにアプリケーションがアクセスできなくなったときに (ターゲット・コンピューター上の Communications Server for Linux ソフトウェアが停止したか、ターゲット・コンピューターへの通信パスが失われたため) 生成されます。ターゲットがドメイン構成ファイルである場合、別のサーバーがマスターの動作を引き継いだときにもこの指示は生成されます (したがって、接続されたターゲット・ファイルは、ファイルのマスター・コピーではなくなります)。

アプリケーションは、この指示の受け取り先として明示的に登録されている必要はありません。Communications Server for Linux は、いずれのタイプの指示についても、指定ターゲット・ハンドルに登録されている、いずれのアプリケーションにもこの指示を戻します。複数のコールバック・ルーチンを使用して指示を受け取るように、現在、登録されているアプリケーションの場合、Communications Server for Linux は最初に登録されたルーチンに対してこの指示を戻します。

ターゲットにアクセスできなくなったという指示をアプリケーションが受信したあと、該当のターゲット・ハンドルを使用する後続の verb は、DISCONNECT_NODE または CLOSE_FILE (ターゲットとのアプリケーションの接続を終了する) を除き、すべて拒否されます。さらに、このターゲット・ハンドルについての指示を受信するための登録はすべて解除されます。ターゲットがアクセス可能になった時点で指示の受信を継続するには、アプリケーション側ではターゲットへの再接続と、必要な指示の受け取り先としての再登録が必要です。

VCB 構造体

```
typedef struct nof_status_indication
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;         /* reserved                  */
    AP_UINT16      primary_rc;     /* primary return code      */
    AP_UINT32      secondary_rc;   /* secondary return code    */
}
```

NOF_STATUS_INDICATION

```
    AP_UINT32      status;                /* status being reported */
    AP_UINT32      dead_target_handle;    /* Handle of dead connection */
                                           /* NULL for system termination */
    unsigned char  reserv1[32];          /* reserved */
} NOF_STATUS_INDICATION;
```

パラメーター

opcode AP_NOF_STATUS_INDICATION

primary_rc
AP_OK

status 報告される状況変更を指定します。値は次のいずれかです。

AP_LOCAL_ABENDED

ローカル・コンピューター上の Communications Server for Linux ソフトウェアが停止しました。ソフトウェアが再始動するまで、アプリケーション側ではこれ以上 NOF verb を発行しないようにする必要があります。

AP_TARGET_ABENDED

ターゲット・コンピューター上の Communications Server for Linux ソフトウェアが停止したか、ターゲット・コンピューターとの通信パスに障害が起きました。

AP_MASTER_TAKEOVER

この値は、アプリケーションがマスター構成ファイル (OPEN_FILE の *requested_role* パラメーターにより指定される) に接続されている場合にのみ戻されます。別のサーバーがマスターの動作を引き継いだ場合、ターゲット・ファイルは、マスター構成ファイルではなくなります。アプリケーションにより、実行中の構成をさらに変更する必要がある場合、ファイルとの接続を閉じるために CLOSE_FILE を使用し、再度 OPEN_FILE を発行して、新規マスター構成ファイルにアクセスする必要があります。

dead_target_handle

障害が起こったターゲットのターゲット・ハンドル、またはマスター構成ファイルではなくなったファイルのターゲット・ハンドル。アプリケーション側では、このターゲット・ハンドルに対し、DISCONNECT_NODE と CLOSE_FILE 以外の verb を発行するのを控える必要があります。

status を AP_LOCAL_ABENDED に設定した場合は、このパラメーターは予約済みです。

PLU_INDICATION

この指示は、ローカル LU がパートナー LU との通信を開始すると生成されます。これは、この PLU に対する最初の ALLOCATE が処理されるとき、またはこの PLU から最初の BIND を受信するとき起こります。また、パートナー LU の CP 名が変更されたときにも、この指示が生成されます。

VCB 構造体

```
typedef struct plu_indication
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;              /* reserved                  */
    unsigned char  format;               /* reserved                  */
    AP_UINT16      primary_rc;           /* primary return code      */
    AP_UINT32      secondary_rc;        /* secondary return code    */
    unsigned char  data_lost;           /* has previous indication  */
                                           /* been lost?               */
    unsigned char  removed;             /* is entry being removed? */
    unsigned char  lu_alias[8];         /* LU alias                  */
    unsigned char  plu_alias[8];        /* partner LU alias         */
    unsigned char  fqplu_name[17];      /* fully qualified partner  */
                                           /* LU name                   */
    unsigned char  description[32];     /* resource description     */
    unsigned char  reserv1[16];         /* reserved                  */
    unsigned char  partner_cp_name[17]; /* partner CP name         */
    unsigned char  partner_lu_located;  /* partner CP name resolved? */
    unsigned char  reserva[20];        /* reserved                  */
} PLU_INDICATION;
```

パラメーター

opcode AP_PLU_INDICATION

primary_rc
AP_OK

data_lost

既存の PLU 指示の中に消失したものがどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の PLU 指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存の PLU 指示は消失していません。

removed

このパラメーターは現在使用されていません。PLU 指示は LU が通信を開始したときのみ生成され、通信を停止したときには生成されません。

lu_alias

ローカル LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

plu_alias

パートナー LU 別名。これは 8 バイトからなる ASCII ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

fqplu_name

17 バイトからなるパートナー LU の完全修飾ネットワーク名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

PLU_INDICATION

description

パートナー LU を記述したヌルで終了するテキスト・ストリング (そのパートナー LU の定義で指定したもの)。

partner_cp_name

パートナー LU に関連した CP の 17 バイトからなる完全修飾ネットワーク名。次の *partner_lu_located* を AP_NO に設定した場合は、このパラメーターは使用されません。

この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

partner_lu_located

ローカル・ノードがパートナー LU を見つけた場所で CP を見つけたかどうかを指定します。値は次のいずれかです。

AP_YES パートナー LU が見つかりました。 *partner_cp_name* パラメーターにはパートナー LU の CP 名が入っています。

AP_NO パートナー LU はまだ見つかっていません。 *partner_cp_name* パラメーターは検査しないでください。

PORT_INDICATION

この指示は、ポートが活動化または非活動化されると生成されます。

VCB 構造体

```
typedef struct port_indication
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                      */
    unsigned char  format;         /* reserved                      */
    AP_UINT16      primary_rc;      /* primary return code          */
    AP_UINT32      secondary_rc;    /* secondary return code        */
    unsigned char  data_lost;       /* previous indication lost     */
    unsigned char  deactivated;     /* has session been deactivated? */
    unsigned char  port_name[8];    /* port name                     */
    unsigned char  description[32]; /* resource description          */
    unsigned char  reserv1[16];     /* reserved                      */
    unsigned char  reserva[20];     /* reserved                      */
} PORT_INDICATION;
```

パラメーター

opcode AP_PORT_INDICATION

primary_rc
AP_OK

data_lost

既存のポート指示の中に消失したものがあるかどうかを指定します。
Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、

内部リソースの不足)を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメータを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のポート指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のポート指示は消失していません。

deactivated

ポートが非活動化されたか活動化されたかを指定します。値は次のいずれかです。

AP_YES ポートは非活動化されました。

AP_NO ポートは活動化されました。

port_name

ポートの名前。これは 8 バイトからなる ASCII スtring で、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

description

ポートを記述したヌルで終了するテキスト・String (そのポートの定義で指定したもの)。

PU_INDICATION

この指示は、ローカル PU の PU-SSCP セッション状況が変わると生成されます。

VCB 構造体

```
typedef struct pu_indication
{
    AP_UINT16      opcode;                /* verb operation code      */
    unsigned char  reserv2;               /* reserved                  */
    unsigned char  format;               /* reserved                  */
    AP_UINT16      primary_rc;           /* primary return code      */
    AP_UINT32      secondary_rc;        /* secondary return code    */
    unsigned char  data_lost;           /* previous indication lost */
    unsigned char  pu_name[8];          /* PU Name                   */
    unsigned char  description[32];     /* resource description     */
    unsigned char  reserv3[16];         /* reserved                  */
    unsigned char  pu_sscp_sess_active; /* Is SSCP session active? */
    unsigned char  host_attachment;     /* Host attachment          */
    unsigned char  reserv1[2];          /* reserved                  */
    SESSION_STATS  pu_sscp_stats;       /* PU-SSCP session statistics */
    unsigned char  sscp_id[6];          /* SSCP id                   */
} PU_INDICATION;

typedef struct session_stats
{
    AP_UINT16      rcv_ru_size;          /* session receive RU size  */
    AP_UINT16      send_ru_size;        /* session send RU size     */
    AP_UINT16      max_send_btu_size;   /* maximum send BTU size    */
    AP_UINT16      max_rcv_btu_size;    /* maximum rcv BTU size     */
    AP_UINT16      max_send_pac_win;    /* maximum send pacing window size */
    AP_UINT16      cur_send_pac_win;    /* current send pacing window size */
    AP_UINT16      max_rcv_pac_win;     /* maximum receive pacing window size */
    AP_UINT16      cur_rcv_pac_win;     /* current receive pacing window size */
    AP_UINT32      send_data_frames;    /* number of data frames sent */
}
```

PU_INDICATION

```
AP_UINT32    send_fmd_data_frames; /* num fmd data frames sent */
AP_UINT32    send_data_bytes;     /* number of data bytes sent */
AP_UINT32    rcv_data_frames;     /* number of data frames received */
AP_UINT32    rcv_fmd_data_frames; /* num fmd data frames received */
AP_UINT32    rcv_data_bytes;     /* number of data bytes received */
unsigned char sidh;               /* session ID high byte
                                  /* (from LFSID) */
unsigned char sidl;               /* session ID low byte (from LFSID)*/
unsigned char odai;               /* ODAI bit set */
unsigned char ls_name[8];         /* Link station name */
unsigned char reserve;            /* reserved */
} SESSION_STATS;
```

パラメーター

opcode AP_PU_INDICATION

primary_rc
AP_OK

data_lost

既存の PU 指示の中に消失したものがどうかを指定します。

Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の PU 指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存の PU 指示は消失していません。

pu_name

PU の名前 (DEFINE_LS verb で指定したもの)。この名前は、8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

description

PU を記述したヌルで終了するテキスト・ストリング (その PU の定義で指定したもの)。

pu_sscp_sess_active

PU-SSCP セッションが活動状態であるかどうか (ACTPU が正常に処理されたかどうか) を指定します。値は次のいずれかです。

AP_YES PU-SSCP セッションは活動状態です。

AP_NO PU-SSCP セッションは活動状態ではありません。

host_attachment

ローカル PU ホスト接続タイプ。

値は次のいずれかです。

AP_DIRECT_ATTACHED

PU はホスト・システムに直接接続されます。

AP_DLUR_ATTACHED

PU は DLUR によりサポートされています。

sscp_id 従属 LU セッションの場合、このパラメーターは、ローカル LU がマップ

される PU についてホストからの ACTPU で受信された SSCP ID です。独立 LU セッションの場合、このパラメーターは 0 (ゼロ) に設定されます。この値は、16 進値で表示される 6 バイトの配列です。

次のパラメーターは、セッション状態が活動から非活動に変わった場合のみ使用されます。

pu_sscp_stats.rcv_ru_size

予約済み (常にゼロに設定される)。

pu_sscp_stats.send_ru_size

予約済み (常にゼロに設定される)。

pu_sscp_stats.max_send_btu_size

送信できる BTU の最大サイズ。

pu_sscp_stats.max_rcv_btu_size

受信できる BTU の最大サイズ。

pu_sscp_stats.max_send_pac_win

予約済み (常にゼロに設定される)。

pu_sscp_stats.cur_send_pac_win

予約済み (常にゼロに設定される)。

pu_sscp_stats.max_rcv_pac_win

予約済み (常にゼロに設定される)。

pu_sscp_stats.cur_rcv_pac_win

予約済み (常にゼロに設定される)。

pu_sscp_stats.send_data_frames

送信された通常フロー・データ・フレームの数。

pu_sscp_stats.send_fmd_data_frames

送信された通常フロー FMD データ・フレームの数。

pu_sscp_stats.send_data_bytes

送信された通常フロー・データ・バイトの数。

pu_sscp_stats.rcv_data_frames

受信された通常フロー・データ・フレームの数。

pu_sscp_stats.rcv_fmd_data_frames

受信された通常フロー FMD データ・フレームの数。

pu_sscp_stats.rcv_data_bytes

受信された通常フロー・データ・バイトの数。

pu_sscp_stats.sidh

セッション ID の上位バイト。

pu_sscp_stats.sidl

セッション ID の下位バイト。

pu_sscp_stats.odai

起点宛先アサイナー・インディケーター。セッションの起動時に、BIND の送信側は、ローカル・ノードに 1 次リンク・ステーションが含まれる場

PU_INDICATION

合はこのフィールドをゼロに設定し、BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合はこのフィールドを 1 に設定します。

pu_sscp_stats.ls_name

統計に関連したリンク・ステーション名。この名前は、8 バイトからなる ASCII 文字ストリングで、名前が 8 文字に満たない場合は、右側にスペースを入れます。

RAPI_CLIENT_INDICATION

この指示は Remote API Client が Communications Server for Linux サーバーに接続する場合、またはそのサーバーから切断する場合に生成されます。NOF アプリケーションでは、これらの指示を使用して、現在クライアントのマスター・サーバーとしてサーバーを使用しているクライアントの追跡を続けることができます。

VCB 構造体

```
typedef struct rapi_client_indication
{
    AP_UINT16      opcode;                /* verb operation code */
    unsigned char  reserv2;               /* reserved */
    unsigned char  format;                /* reserved */
    AP_UINT16      primary_rc;            /* primary return code */
    AP_UINT32      secondary_rc;          /* secondary return code */
    unsigned char  data_lost;             /* previous indication lost */
    unsigned char  reason;                /* reason for indication */
    unsigned char  sys_name[128];         /* system name client sends us */
    SNA_IP_ADDR    rapi_client_origin_ip_addr; /* IP addr client sends us */
    SNA_IP_ADDR    rapi_client_adj_ip_addr; /* IP addr client comes in on */
    AP_UINT16      rapi_client_adj_port; /* port IP client comes in on */
    unsigned char  reserva[16];           /* reserved */
} RAPI_CLIENT_INDICATION;

typedef struct sna_ip_addr
{
    AP_UINT16      family;                 /* IPv4 or IPv6 */
    union
    {
        unsigned char  ipv4_addr[4];
        unsigned char  ipv6_addr[16];
    } ip_addr;
} SNA_IP_ADDR;
```

パラメーター

opcode AP_RAPI_CLIENT_INDICATION

primary_rc
AP_OK

data_lost

既存のクライアント指示の中に消失したものがあるかどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の 1 つ以上のクライアント指示が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のクライアント指示は消失していません。

reason このクライアントに起こった状況変更を指定します。値は次のいずれかです。

AP_RAPI_CLIENT_CONNECTED

クライアントは始動され、マスター・サーバーとして、この Communications Server for Linux サーバーに接続されました。

AP_RAPI_CLIENT_DISCONNECTED

クライアントは停止され、Communications Server for Linux サーバーから切断されました。

sys_name

クライアントの完全修飾システム名 (たとえば、newbox.this.co.uk)。

rapi_client_origin_ip_addr

クライアントの IP アドレス。

rapi_client_origin_ip_addr.family

クライアントに指定された TCP/IP アドレスのタイプ。可能な値は以下のとおりです。(これらは Communications Server for Linux を使用して定義された AP_* 値ではなく標準の TCP/IP 値です。)

AF_INET

IPv4 アドレス。ドット 10 進アドレスで指定します (たとえば、193.1.11.100)。

AF_INET6

IPv6 アドレス。コロン 16 進アドレスで指定します (たとえば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)。

注: 値 AF_INET および AF_INET6 はシステム・ヘッダー・ファイルから取り込まれます。これらは、Communications Server for Linux を使用して定義する標準 AP_* 値ではありません。システム・ヘッダー・ファイルは、Linux サーバーまたはクライアント上の **/usr/include/linux/socket.h**、および AIX クライアント上の **/usr/include/sys/socket.h** です。

使用する NOF アプリケーションでこれらの値についてテストする必要がある場合は、**#include** を使用してこのシステム・ファイルを **nof_c.h** ヘッダー・ファイルに組み込む必要があります。

rapi_client_origin_ip_addr.ip_addr.ipv4_addr

このフィールドは *family* パラメーターが AF_INET に設定されている場合のみ使用します。クライアント・コンピューターの IPv4 (ドット 10 進) アドレス。

rapi_client_origin_ip_addr.ip_addr.ipv6_addr

このフィールドは *family* パラメーターが AF_INET6 に設定されている場合のみ使用します。クライアント・コンピューターの IPv6 (コロン 16 進) アドレス。

rapi_client_adj_ip_addr

クライアントが Communications Server for Linux に接続するために使用する

RAPI_CLIENT_INDICATION

る IP アドレス。以下の項目の 1 つでも真の場合、*rapi_client_origin_ip_addr* と同じにすることはできません。

- クライアントは Web サーバー経由で接続される。
- クライアントは TCP/IP プロキシまたは NAT ルーターを経由して接続される。たとえば、Linux iptables ツール。
- クライアントに複数の IP アドレスがある。

rapi_client_adj_ip_addr.family

クライアントが Communications Server for Linux に接続するために使用する TCP/IP アドレスのタイプ。可能な値は以下のとおりです。(これらは Communications Server for Linux を使用して定義された AP_* 値ではなく標準の TCP/IP 値です。)

AF_INET

IPv4 アドレス。ドット 10 進アドレスで指定します (たとえば、193.1.11.100)。

AF_INET6

IPv6 アドレス。コロン 16 進アドレスで指定します (たとえば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)。

注: 値 AF_INET および AF_INET6 はシステム・ヘッダー・ファイルから取り込まれます。これらは、Communications Server for Linux を使用して定義する標準 AP_* 値ではありません。システム・ヘッダー・ファイルは、Linux サーバーまたはクライアント上の */usr/include/linux/socket.h*、および AIX クライアント上の */usr/include/sys/socket.h* です。

使用する NOF アプリケーションでこれらの値についてテストする必要がある場合は、*#include* を使用してこのシステム・ファイルを *nof_c.h* ヘッダー・ファイルに組み込む必要があります。

rapi_client_adj_ip_addr.ip_addr.ipv4_addr

このフィールドは *family* パラメーターが AF_INET に設定されている場合のみ使用します。クライアントが Communications Server for Linux に接続するために使用する IPv4 (ドット 10 進) アドレス。

rapi_client_adj_ip_addr.ip_addr.ipv6_addr

このフィールドは *family* パラメーターが AF_INET6 に設定されている場合のみ使用します。クライアントが Communications Server for Linux に接続するために使用する IPv6 (コロン 16 進) アドレス。

rapi_client_adj_port

クライアントが Communications Server for Linux に接続するために使用する IP ポート番号。

REGISTRATION_FAILURE

REGISTRATION_FAILURE は、ネットワーク・ノード・サーバーにリソースを登録する試みが失敗したことを示します。

VCB 構造体

```
typedef struct registration_failure
{
    AP_UINT16      opcode;           /* verb operation code          */
    unsigned char  reserv2;         /* reserved                     */
    unsigned char  format;         /* reserved                     */
    AP_UINT16      primary_rc;      /* primary return code         */
    AP_UINT32      secondary_rc;    /* secondary return code       */
    unsigned char  data_lost;       /* previous indication lost    */
    unsigned char  resource_name[17]; /* network qualified resource name */
    AP_UINT16      resource_type;   /* resource type               */
    unsigned char  description[32]; /* resource description        */
    unsigned char  reserv1[16];    /* reserved                    */
    unsigned char  reserv2b[2];    /* reserved                    */
    AP_UINT32      sense_data;     /* sense data                  */
    unsigned char  reserva[20];    /* reserved                    */
} REGISTRATION_FAILURE;
```

パラメーター

opcode AP_REGISTRATION_FAILURE

primary_rc
AP_OK

data_lost

既存の登録失敗指示の中に消失したものがあるかどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の登録失敗指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存の登録失敗指示は消失していません。

resource_name

登録に失敗したリソースの名前。この名前は、右側に EBCDIC のスペースを入れた 17 バイトの EBCDIC ストリングです。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

resource_type

登録に失敗したリソースのタイプ。次のいずれかです。

AP_NNCP_RESOURCE
ネットワーク・ノード。

AP_ENCP_RESOURCE
エンド・ノード。

AP_LU_RESOURCE
LU。

description

リソースを記述したヌルで終了するテキスト・ストリング (そのリソースの定義で指定したもの)。

REGISTRATION_FAILURE

sense_data

センス・データ (SNA フォーマットで指定)。

RTP_INDICATION

この指示は、次のいずれかが起こったときに生成されます。

- RTP 接続が接続または切断された。
- 活動セッション・カウントが変更した。
- 接続でパス・スイッチを実行している。

接続が切断されると、RTP の最終統計が戻されます。それ以外の場合は、*rtp_stats* パラメーターは予約済みです。

VCB 構造体

```
typedef struct rtp_indication
{
    AP_UINT16    opcode;           /* Indication operation code */
    unsigned char reserv2;        /* reserved */
    unsigned char format;        /* reserved */
    AP_UINT16    primary_rc;      /* primary return code */
    AP_UINT32    secondary_rc;    /* secondary return code */
    unsigned char data_lost;     /* Previous indication lost? */
    unsigned char connection_state; /* current state of the RTP connection */

    unsigned char rtp_name[8];    /* name of the RTP connection */
    AP_UINT16    num_sess_active; /* number of active sessions */
    unsigned char indication_cause; /* reason for this indication */
    unsigned char connection_type; /* usage of RTP connection */
    unsigned char reserv3[2];     /* reserved */
    RTP_STATISTICS rtp_stats;     /* RTP statistics */
} RTP_INDICATION;

typedef struct rtp_statistics
{
    AP_UINT32    bytes_sent;      /* total number of bytes sent */
    AP_UINT32    bytes_received; /* total number of bytes received */
    AP_UINT32    bytes_resent;    /* total number of bytes resent */
    AP_UINT32    bytes_discarded; /* total number of bytes discarded */
    AP_UINT32    packets_sent;    /* total number of packets sent */
    AP_UINT32    packets_received; /* total number of packets received */
    AP_UINT32    packets_resent; /* total number of packets resent */
    AP_UINT32    packets_discarded; /* total number of packets discarded */
    AP_UINT32    gaps_detected;   /* gaps detected */
    AP_UINT32    send_rate;       /* current send rate */
    AP_UINT32    max_send_rate;   /* maximum send rate */
    AP_UINT32    min_send_rate;   /* minimum send rate */
    AP_UINT32    receive_rate;    /* current receive rate */
    AP_UINT32    max_receive_rate; /* maximum receive rate */
    AP_UINT32    min_receive_rate; /* minimum receive rate */
    AP_UINT32    burst_size;      /* current burst size */
    AP_UINT32    up_time;         /* total uptime of connection */
    AP_UINT32    smooth_rtt;      /* smoothed round-trip time */
    AP_UINT32    last_rtt;       /* last round-trip time */
    AP_UINT32    short_req_timer; /* SHORT_REQ timer duration */
    AP_UINT32    short_req_timeouts; /* number of SHORT_REQ timeouts */
    AP_UINT32    liveness_timeouts; /* number of liveness timeouts */
    AP_UINT32    in_invalid_sna_frames; /* number of invalid SNA frames received */
    AP_UINT32    in_sc_frames;    /* number of SC frames received */
    AP_UINT32    out_sc_frames;   /* number of SC frames sent */
    AP_INT32     delay_change_sum; /* delay change sum */
}
```

```

AP_UINT32    current_receiver_threshold;
              /* current ARB-R receiver threshold */
AP_UINT32    minimum_receiver_threshold;
              /* minimum ARB-R receiver threshold */
AP_UINT32    maximum_receiver_threshold;
              /* maximum ARB-R receiver threshold */
AP_UINT32    sent_normals_count; /* number of NORMALs sent */
AP_UINT32    sent_slowdowns_count; /* number of SLOWDOWNs sent */
AP_UINT32    rcvd_normals_count; /* number of NORMALs received */
AP_UINT32    rcvd_slowdowns_count; /* number of SLOWDOWNs received */
AP_UINT32    dcs_reset_count_non_heal;
              /* number of non-healing resets */
AP_UINT16    dcs_reset_count_healing;
              /* number of self-healing resets */
unsigned char arb_mode; /* ARB mode (GREEN, YELLOW, RED) */
unsigned char reserve[1]; /* reserved */
} RTP_STATISTICS;

```

パラメーター

opcode AP_RTP_INDICATION

primary_rc
AP_OK

data_lost

既存のディレクトリー指示の中に消失したものがどうかを指定します。 Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のディレクトリー指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のディレクトリー指示は消失していません。

connection_state

RTP 接続の現在の状態。値は次のいずれかです。

AP_CONNECTING

接続のセットアップが開始されましたが、まだ完了していません。

AP_CONNECTED

接続は完全に活動状態です。

AP_DISCONNECTED

接続は活動状態ではなくなりました。

rtp_name

RTP 接続名。この名前は、8 バイトからなるローカル・システム上で表示可能な文字セットのストリングです。8 バイトはすべて有効です。

num_sess_active

接続上で現在活動状態であるセッションの数。

indication_cause

指示の原因。値は次のいずれかです。

AP_ACTIVATED

接続は活動状態になりました。

RTP_INDICATION

AP_DEACTIVATED

接続は非活動状態になりました。

AP_PATH_SWITCHED

接続はパス・スイッチを正常に完了しました。

AP_SESS_COUNT_CHANGING

接続を使用している活動セッションの数が変わりました。

AP_SETUP_FAILED

接続は完全に活動状態になる前に失敗しました。

connection_type

RTP 接続のセッションのタイプを指定します。値は次のいずれかです。

AP_RTP_CP_CP_SESSION

RTP 接続は CP-CP セッションを伝送します。

AP_RTP_LU_LU_SESSION

RTP 接続は LU-LU セッションを伝送します。

AP_RTP_ROUTE_SETUP

RTP 接続は経路セットアップに使用します。

次のパラメーターは、接続が活動状態でなくなったとき (*indication_cause* パラメーターが **AP_DEACTIVATED** または **AP_SETUP_FAILED** に設定されたとき) のみ提供されます。それ以外のすべての場合には、次のパラメーターは予約済みです。

rtp_stats.bytes_sent

ローカル・ノードがこの RTP 接続で送信したバイトの合計数。

rtp_stats.bytes_received

ローカル・ノードがこの RTP 接続で受信したバイトの合計数。

rtp_stats.bytes_resent

転送中のバイト消失のため、ローカル・ノードがこの RTP 接続で再送したバイトの合計数。

rtp_stats.bytes_discarded

RTP 接続の相手側が送信し、既に受信済みのデータの重複として廃棄されたバイトの合計数。

rtp_stats.packets_sent

ローカル・ノードがこの RTP 接続で送信したパケットの合計数。

rtp_stats.packets_received

ローカル・ノードがこの RTP 接続で受信したパケットの合計数。

rtp_stats.packets_resent

転送中のパケット消失のため、ローカル・ノードがこの RTP 接続で再送したパケットの合計数。

rtp_stats.packets_discarded

RTP 接続の相手側が送信し、既に受信済みのデータの重複として廃棄されたパケットの合計数。

rtp_stats.gaps_detected

ローカル・ノードが検出したギャップの合計数。各ギャップは 1 つ以上の消失フレームに相当します。

rtp_stats.send_rate

この RTP 接続での現在の送信速度 (キロビット/秒)。この速度は、ARB (適応速度ベース) アルゴリズムで計算された最大許容送信速度です。

rtp_stats.max_send_rate

この RTP 接続での最大送信速度 (キロビット/秒)。

rtp_stats.min_send_rate

この RTP 接続での最小送信速度 (キロビット/秒)。

rtp_stats.receive_rate

この RTP 接続での現在の受信速度 (キロビット/秒)。この速度は、最後の測定間隔について計算された実際の受信速度です。

rtp_stats.max_receive_rate

この RTP 接続での最大受信速度 (キロビット/秒)。

rtp_stats.min_receive_rate

この RTP 接続での最小受信速度 (キロビット/秒)。

rtp_stats.burst_size

この RTP 接続での現在のバースト・サイズ (バイト単位)。

rtp_stats.up_time

この RTP 接続が活動状態であった合計時間 (秒単位)。

rtp_stats.smooth_rtt

ローカル・ノードとパートナー RTP ノード間の往復の平滑化測定値 (ミリ秒単位)。

rtp_stats.last_rtt

ローカル・ノードとパートナー RTP ノード間の最終往復の測定値 (ミリ秒単位)。

rtp_stats.short_req_timer

SHORT_REQ タイマーに使用する現行所要時間 (ミリ秒単位)。

rtp_stats.short_req_timeouts

SHORT_REQ のタイムアウトの回数。

rtp_stats.liveness_timeouts

この RTP 接続について活性タイマーが満了した合計回数。活性タイマーは、接続のアイドル期間が *rtp_connection_detail.liveness_timer* で指定した値になると満了します。

rtp_stats.in_invalid_sna_frames

この RTP 接続で受信され、有効でないため廃棄にされた SNA フレームの合計数。

rtp_stats.in_sc_frames

この RTP 接続で受信されたセッション制御フレームの合計数。

rtp_stats.out_sc_frames

この RTP 接続で送信されたセッション制御フレームの合計数。

rtp_stats.delay_change_sum

この RTP 接続で ARB-R アルゴリズムによって現在保持されている遅延変更の合計値。

RTP_INDICATION

rtp_stats.current_receiver_threshold

この RTP 接続で ARB-R アルゴリズムによって現在保持されている受信側しきい値。

rtp_stats.minimum_receiver_threshold

この RTP 接続で ARB-R アルゴリズムによって現在保持されている受信側しきい値の最小値。

rtp_stats.maximum_receiver_threshold

この RTP 接続で ARB-R アルゴリズムによって現在保持されている受信側しきい値の最大値。

rtp_stats.sent_normals_count

この RTP 接続で ARB-R アルゴリズムによって送信された NORMAL フィードバックの ARB-R セグメント数。

rtp_stats.sent_slowdowns_count

この RTP 接続で ARB-R アルゴリズムによって送信された SLOWDOWN1 および SLOWDOWN2 フィードバックの ARB-R セグメント数。

rtp_stats.rcvd_normals_count

この RTP 接続で ARB-R アルゴリズムによって受信された NORMAL フィードバックの ARB-R セグメント数。

rtp_stats.rcvd_slowdowns_count

この RTP 接続で ARB-R アルゴリズムによって受信された SLOWDOWN1 および SLOWDOWN2 フィードバックの ARB-R セグメント数。

rtp_stats.dcs_reset_count_non_heal

この RTP 接続で 通常の ARB-R 処理の一部として行われた遅延変更リセットの合計回数。

rtp_stats.dcs_reset_count_healing

この RTP 接続で 通常の ARB-R アルゴリズムの自己回復が行われた遅延変更リセットの合計回数。

rtp_stats.arb_mode

この RTP 接続の現在の ARB-R 状況モード。値は次のいずれかです。

- 0 GREEN
- 1 YELLOW
- 2 RED

SERVER_INDICATION

この指示は、LAN 内の別のコンピューター上で Communications Server for Linux ソフトウェアが開始または停止されたときか、マスターまたはバックアップ・サーバーとしてのサーバーの役割が変更したときに生成されます。NOF アプリケーションでは、これらの指示を使用して、現在アクティブになっているサーバーを追跡したり、新規サーバーがマスターの役割を正常に引き継いだかどうかを判別したりすることができます。

サーバー指示は、サーバー上の他の Communications Server for Linux のコンポーネントの状況が変更されたときにも (Communications Server for Linux 内部での使用

のために) 生成されます。アプリケーションで、前記のようにサーバー指示を使用する必要がある場合には、*status* および *flags* パラメーターが変更されたかどうかを検査する必要があります。これらのパラメーターが変更を示していない場合は、サーバー指示を無視して構いません。

サーバー指示を登録するために使用される REGISTER_INDICATION_SINK verb は、ヌル・ターゲット・ハンドルで発行される必要があります。つまり、特定のターゲットとは関連付けられません。

VCB 構造体

```
typedef struct server_indication
{
    AP_UINT16      opcode;           /* verb operation code      */
    unsigned char  reserv2;         /* reserved                  */
    unsigned char  format;          /* reserved                  */
    AP_UINT16      primary_rc;      /* primary return code      */
    AP_UINT32      secondary_rc;    /* secondary return code    */
    unsigned char  data_lost;       /* previous indication lost */
    AP_UINT32      status;          /* node status              */
    AP_UINT32      flags;           /* is server master or backup? */
    unsigned char  server_name[128]; /* name of server           */
} SERVER_INDICATION;
```

パラメーター

opcode AP_SERVER_INDICATION

primary_rc
AP_OK

data_lost

既存のサーバー指示の中に消失したものがあるかどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のサーバー指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のサーバー指示は消失していません。

status 指示されたサーバー上の SNA ソフトウェアの状況を指定します。値は次のいずれかです。

AP_ACTIVE
SNA ソフトウェアは開始済みです。

AP_NOT_ACTIVE
SNA ソフトウェアは停止済みです。

flags 指示されるサーバーがマスター・サーバーであるかバックアップ・サーバーであるかを指定します。アプリケーションでは、論理 AND 演算を使用して、以下のように適切な値を検査する必要があります。

- 式「*flags* AND AP_MASTER_FLAG」がゼロ以外の値の場合、指示されるサーバーはマスター・サーバーです。

SERVER_INDICATION

- 式「*flags AND AP_BACKUP_FLAG*」ゼロ以外の値の場合、指示されるサーバーはバックアップ・サーバーです。

server_name

SNA ソフトウェアが開始済みまたは停止済みになっているサーバーの名前。

SESSION_INDICATION

この指示は、セッションが活動化または非活動化されると生成されます。セッションが非活動化された場合は、この *verb* はセッションの使用状況の統計を戻します。

VCB 構造体

```
typedef struct session_indication
{
    AP_UINT16      opcode;           /* verb operation code */
    unsigned char  reserv2;          /* reserved */
    unsigned char  format;           /* reserved */
    AP_UINT16      primary_rc;       /* primary return code */
    AP_UINT32      secondary_rc;     /* secondary return code */
    unsigned char  data_lost;        /* previous indication lost */
    unsigned char  deactivated;      /* has session been deactivated? */
    unsigned char  lu_name[8];       /* LU name */
    unsigned char  lu_alias[8];      /* LU alias */
    unsigned char  plu_alias[8];     /* partner LU alias */
    unsigned char  fqplu_name[17];   /* fully qualified partner LU name */
    unsigned char  mode_name[8];     /* mode name */
    unsigned char  session_id[8];    /* session ID */
    FQPCID         fqpcid;           /* fully qualified procedure */
                                     /* correlator ID */
    AP_UINT32      sense_data;       /* sense data */
    unsigned char  reserv1;          /* reserved */
    SESSION_STATS  sess_stats;       /* session statistics */
    unsigned char  sscp_id[6];       /* SSCP ID */
    unsigned char  plu_slu_comp_lvl; /* compression level PLU->SLU */
    unsigned char  slu_plu_comp_lvl; /* compression level SLU->PLU */
    unsigned char  comp_in_series;   /* reserved */
    unsigned char  reserva[11];     /* reserved */
} SESSION_INDICATION;

typedef struct fqpcid
{
    unsigned char  pcid[8];          /* procedure correlator identifier */
    unsigned char  fqcp_name[17];   /* originator's network qualified */
                                     /* CP name */
    unsigned char  reserve3[3];     /* reserved */
} FQPCID;

typedef struct session_stats
{
    AP_UINT16      rcv_ru_size;      /* session receive RU size */
    AP_UINT16      send_ru_size;     /* session send RU size */
    AP_UINT16      max_send_btu_size; /* maximum send BTU size */
    AP_UINT16      max_rcv_btu_size; /* maximum rcv BTU size */
    AP_UINT16      max_send_pac_win; /* maximum send pacing window size */
    AP_UINT16      cur_send_pac_win; /* current send pacing window size */
    AP_UINT16      max_rcv_pac_win;  /* maximum receive pacing window */
                                     /* size */
    AP_UINT16      cur_rcv_pac_win;  /* current receive pacing window */
                                     /* size */
    AP_UINT32      send_data_frames; /* number of data frames sent */
    AP_UINT32      send_fmd_data_frames; /* num fmd data frames sent */
    AP_UINT32      send_data_bytes;  /* number of data bytes sent */
}
```

```

AP_UINT32      rcv_data_frames;      /* number of data frames received */
AP_UINT32      rcv_fmd_data_frames; /* num fmd data frames received */
AP_UINT32      rcv_data_bytes;      /* number of data bytes received */
unsigned char  sidh;                 /* session ID high byte           */
                                           /* (from LFSID)                   */
unsigned char  sidl;                 /* session ID low byte (from LFSID)*/
unsigned char  odai;                 /* ODAI bit set                   */
unsigned char  ls_name[8];          /* Link station name              */
unsigned char  pacing_type;         /* Pacing type                    */
} SESSION_STATS;

```

パラメーター

opcode AP_SESSION_INDICATION

primary_rc
AP_OK

data_lost

既存のセッション指示の中に消失したものがあるかどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件 (たとえば、内部リソースの不足) を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存のセッション指示 (1 つ以上) が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存のセッション指示は消失していません。

deactivated

セッションが非活動化されたか活動化されたかを指定します。値は次のいずれかです。

AP_YES セッションは非活動化されました。

AP_NO セッションは活動化されました。

lu_name

Communications Server for Linux に対して定義されているローカル LU の LU 名。これは 8 バイトからなるタイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

lu_alias

Communications Server for Linux に対して定義されているローカル LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

plu_alias

パートナー LU の LU 別名。これは 8 バイトからなる ASCII ストリングで、ローカル・システム上で表示可能な文字を使用し、名前が 8 バイトに満たない場合は、右側にスペースを入れます。

fqplu_name

Communications Server for Linux に対して定義されているパートナー LU の完全修飾 LU 名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、1 ~ 8 文字までの A スト

SESSION_INDICATION

リング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および 1 ~ 8 文字までの A ストリング文字からなる LU 名で構成されます。

mode_name

LU が使用するモードの名前。これは 8 バイトの英数字からなる (英文字で始まる) タイプ A の EBCDIC ストリングで、名前が 8 バイトに満たない場合は、右側に EBCDIC のスペースを入れます。

session_id

8 バイトからなるセッション ID。

fqpcid.pcid

プロシージャー相関関係子 ID。これは 8 バイトからなる 16 進数ストリングです。

fqpcid.fqcp_name

CP の完全修飾名。この名前は 17 バイトの EBCDIC ストリングで、右側に EBCDIC スペースを入れます。この名前は、最大 8 文字の A ストリング文字からなるネットワーク ID、EBCDIC ドット (ピリオド) 文字、および最大 8 文字の A ストリング文字からなるネットワーク名で構成されます。

次のパラメーターは、*deactivated* が AP_YES に設定されている場合のみ使用されます。

sense_data

セッションを終了した UNBIND メッセージで送信または受信されたセンス・データ。

duplex_support

BIND で折衝された会話二重サポート。値は次のいずれかです。

AP_HALF_DUPLEX

半二重会話のみがサポートされます。

AP_FULL_DUPLEX

半二重会話と全二重会話の両方がサポートされます。優先データもサポートされます。

AP_UNKNOWN

セッションが非活動化されたため、二重サポートは不明です。

sess_stats.rcv_ru_size

受信 RU の最大サイズ。

sess_stats.send_ru_size

送信 RU の最大サイズ。

sess_stats.max_send_btu_size

送信できる BTU の最大サイズ。

sess_stats.max_rcv_btu_size

受信できる BTU の最大サイズ。

sess_stats.max_send_pac_win

このセッションの送信ペーシング・ウィンドウの最大サイズ。

sess_stats.cur_send_pac_win

このセッションの送信ペーシング・ウィンドウの現行サイズ。

sess_stats.max_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの最大サイズ。

sess_stats.cur_rcv_pac_win

このセッションの受信ペーシング・ウィンドウの現行サイズ。

sess_stats.send_data_frames

送信された通常フロー・データ・フレームの数。

sess_stats.send_fmd_data_frames

送信された通常フロー FMD データ・フレームの数。

sess_stats.send_data_bytes

送信された通常フロー・データ・バイトの数。

sess_stats.rcv_data_frames

受信された通常フロー・データ・フレームの数。

sess_stats.rcv_fmd_data_frames

受信された通常フロー FMD データ・フレームの数。

sess_stats.rcv_data_bytes

受信された通常フロー・データ・バイトの数。

sess_stats.sidh

セッション ID の上位バイト。

sess_stats.sidl

セッション ID の下位バイト。

sess_stats.odai

起点宛先アサイナー・インディケーター。セッションの起動時に、ローカル・ノードに 1 次リンク・ステーションが含まれる場合、BIND の送信側は、このフィールドをゼロに設定し、BIND 送信側のノードに 2 次リンク・ステーションが含まれる場合は、このフィールドを 1 に設定します。

sess_stats.ls_name

統計に関連したリンク・ステーション名。これは 8 バイトからなるローカル・システム上で表示可能な文字セットのストリングです。8 バイトはすべて有効です。このフィールドは、セッション・トラフィックの転送経路となるリンクとこのセッションを関連付けるために使用できます。

sess_stats.pacing_type

このセッションで使用している受信ペーシングのタイプ。

sscp_id 従属 LU セッションの場合、この LU が使用する PU の ACTPU で受信された SSCP の ID。このパラメーターは 6 バイトからなり、従属 LU のみで使用します。独立 LU の場合は、このパラメーターのすべてのビットはゼロに設定されます。

session_detail.plu_slu_comp_lvl

1 次 LU (PLU) から 2 次 LU (SLU) に送信されるデータの圧縮レベルを指定します。値は次のいずれかです。

SESSION_INDICATION

AP_NONE

圧縮は使用されません。

AP_RLE_COMPRESSION

実行長さエンコード (RLE) 圧縮が使用されます。

AP_LZ9_COMPRESSION

LZ9 圧縮が使用されます。

AP_LZ10_COMPRESSION

LZ10 圧縮が使用されます。

session_detail.slu_plu_comp_lvl

2 次 LU (SLU) から 1 次 LU (PLU) に送信されるデータの圧縮レベルを指定します。値は次のいずれかです。

AP_NONE

圧縮は使用されません。

AP_RLE_COMPRESSION

実行長さエンコード (RLE) 圧縮が使用されます。

AP_LZ9_COMPRESSION

LZ9 圧縮が使用されます。

AP_LZ10_COMPRESSION

LZ10 圧縮が使用されます。

SNA_NET_INDICATION

この指示は、別の NOF アプリケーションまたは Communications Server for Linux 管理ツールにより SNA ネットワーク・ファイル **sna.net** が変更されたときに生成されます。この verb のターゲットは、この指示を受け取るための登録を行った REGISTER_INDICATION_SINK verb の *target_handle* パラメーターにより識別され、ファイル **sna.net** である必要があります。

VCB 構造体

この指示に関連した特定の VCB 構造体はありません。SNA ネットワーク指示を受け取るための登録を行うには、アプリケーションで、値 AP_SNA_NET_INDICATION を REGISTER_INDICATION_SINK の *indication_opcode* パラメーターとして指定します。SNA ネットワーク・ファイルに変更が行われると、Communications Server for Linux は、変更を行った NOF verb (ADD_BACKUP または DELETE_BACKUP) からの VCB のコピーを送信することにより、アプリケーションのコールバック・ルーチンに変更を報告します。

アプリケーションが SNA ネットワーク指示と、SNA ネットワーク・ファイルに発行される独自の NOF verb への非同期応答を区別できるように、Communications Server for Linux は、指示についての VCB の *primary_rc* パラメーターを変更します。値 AP_INDICATION は、SNA ネットワーク・ファイル指示に関連した VCB を識別します。つまり値 AP_OK またはその他の値は、アプリケーション独自の NOF verb のいずれかへの非同期応答を示します。

TN_REDIRECTION_INDICATION

この指示は Telnet クライアントが TN リダイレクターを使用して、セッションを開始または終了するときに生成されます。また、TN サーバー機能を提供している SNA ノードが停止するときにも生成され、TN リダイレクト指示の再登録が必要になるアプリケーションに通知されます。これを行うのは、このノードの停止と再始動のときに、これらの指示の登録が維持されないためです。

VCB 構造体

```
typedef struct tn_redirection_indication
{
    AP_UINT16      opcode;                /* verb operation code          */
    unsigned char  reserv2;              /* reserved                     */
    unsigned char  format;               /* reserved                     */
    AP_UINT16      primary_rc;          /* primary return code          */
    AP_UINT32      secondary_rc;        /* secondary return code        */
    unsigned char  data_lost;           /* previous indication lost     */
    unsigned char  reason;              /* reason for indication        */
    SNA_IP_ADDR    client_ip_addr;      /* client IP address            */
    AP_UINT16      client_port;         /* client port number           */
    SNA_IP_ADDR    host_ip_addr;        /* host IP address              */
    AP_UINT16      host_port;           /* host port number             */
    unsigned char  client_number;       /* client number                */
    unsigned char  listen_local_address[46]; /* Local addr client connects to */
    unsigned char  reserva[16];         /* reserved                     */
} TN_REDIRECTION_INDICATION;

typedef struct sna_ip_addr
{
    AP_UINT16      family;               /* IPv4 or IPv6                */
    union
    {
        unsigned char  ipv4_addr[4];
        unsigned char  ipv6_addr[16];
    } ip_addr;
} SNA_IP_ADDR;
```

パラメーター

opcode AP_TN_REDIRECTION_INDICATION

primary_rc
AP_OK

data_lost

既存の TN リダイレクト指示の中に消失したものがどうかを指定します。Communications Server for Linux は、指示の送信を妨げる条件（たとえば、内部リソースの不足）を検出すると、条件をクリアしたあと、次の指示で *data_lost* パラメーターを設定してその条件を示します。値は次のいずれかです。

AP_YES 既存の 1 つ以上の TN リダイレクト指示が消失しました。この VCB 内の以降のフィールドはゼロに設定される場合があります。

AP_NO 既存の TN リダイレクト指示は消失していません。

reason この指示の送信理由を指定します。値は次のいずれかです。

TN_REDIRECTION_INDICATION

AP_CONNECTION_ACTIVATED

Telnet クライアントは TN リダイレクターを使用するセッションを開始済みです。

AP_CONNECTION_DEACTIVATED

TN リダイレクター・セッションは終了されました。

AP_TN_SERVER_TERMINATED

TN サーバー機能を提供するノードは停止されました。このノードを使用する活動状態の TN リダイレクターが存在していると、アプリケーションは AP_CONNECTION_DEACTIVATED と設定された *reason* が付いた、それぞれのセッションに対する指示も受け取ります。

アプリケーションは、TN リダイレクト指示の受け取りを続行する必要がある場合、ノードの再始動の際にこれらの指示の再登録が必要になります。

以下のフィールドは *reason* が AP_TN_SERVER_TERMINATED に設定されている場合は無効です。

client_ip_addr.family

Telnet クライアントを実行するコンピューター用に指定された TCP/IP アドレスのタイプ。可能な値は以下のとおりです。(これらは Communications Server for Linux を使用して定義された AP_* 値ではなく、標準の TCP/IP 値です。)

AF_INET

IPv4 アドレス。ドット 10 進アドレスで指定します (たとえば、193.1.11.100)。

AF_INET6

IPv6 アドレス。コロン 16 進アドレスで指定します (たとえば、2001:0db8:0000:0000:0000:0000:1428:57ab または 2001:db8::1428:57ab)。

注: 値 AF_INET および AF_INET6 はシステム・ヘッダー・ファイルから取り込まれます。これらは、Communications Server for Linux を使用して定義する標準 AP_* 値ではありません。システム・ヘッダー・ファイルは、Linux サーバーまたはクライアント上の `/usr/include/linux/socket.h`、および AIX クライアント上の `/usr/include/sys/socket.h` です。

使用する NOF アプリケーションでこれらの値についてテストする必要がある場合は、`#include` を使用してこのシステム・ファイルを `nof_c.h` ヘッダー・ファイルに組み込む必要があります。

client_ip_addr.ip_addr.ipv4_addr

このフィールドは *client_ip_addr.family* パラメーターが AF_INET に設定されている場合のみ使用します。Telnet クライアントを実行するコンピューターの IPv4 (ドット 10 進) アドレス。

client_ip_addr.ip_addr.ipv6_addr

このフィールドは *client_ip_addr.family* が AF_INET6 に設定されている場合のみ使用します。Telnet クライアントを実行するコンピューターの IPv6 (コロン 16 進) アドレス。

client_port

Telnet クライアントが TN リダイレクター・ノードにアクセスするために使用するサーバーの TCP/IP ポート番号。

host_ip_addr

クライアントが通信するホスト・コンピューターの TCP/IP アドレス。指定できるのは、以下のいずれかです。

- IPv4 ドット 10 進アドレス (たとえば、193.1.11.100)
- IPv6 コロン 16 進アドレス (たとえば、
2001:0db8:0000:0000:0000:0000:1428:57ab または
2001:db8::1428:57ab)

host_port

TN リダイレクター・ノードがホストにアクセスするために使用する TCP/IP ポート番号。

client_number

それぞれのクライアントに固有の番号。これは、タイプ AP_CONNECTION_ACTIVATED とタイプ AP_CONNECTION_DEACTIVATED の正常なリダイレクト指示を関連づけるために使用できます。

listen_local_address

TN3270 クライアントの接続先であるローカル TN サーバー・コンピューター上のアドレス。

付録 A. 戻りコード値

本書の付録で、NOF インターフェースに関するすべての可能性のある戻りコードの番号順リストを提供します。値はヘッダー・ファイルの **values_c.h** に定義されています。

この付録は使用するアプリケーションで受け取った戻りコードを確認する場合に使用できます。

1 次戻りコード

以下の 1 次戻りコードは NOF アプリケーションで使用されます。

AP_OK	0x0000
AP_PARAMETER_CHECK	0x0100
AP_STATE_CHECK	0x0200
AP_INDICATION	0x0210
AP_TP_BUSY	0x02F0
AP_ALLOCATION_ERROR	0x0300
AP_ACTIVATION_FAIL_RETRY	0x0310
AP_COMM_SUBSYSTEM_ABENDED	0x03F0
AP_ACTIVATION_FAIL_NO_RETRY	0x0410
AP_COMM_SUBSYSTEM_NOT_LOADED	0x04F0
AP_DEALLOC_ABEND	0x0500
AP_LU_SESS_LIMIT_EXCEEDED	0x0510
AP_DEALLOC_ABEND_PROG	0x0600
AP_FUNCTION_NOT_SUPPORTED	0x0610
AP_THREAD_BLOCKING	0x06F0
AP_DEALLOC_ABEND_SVC	0x0700
AP_DEALLOC_ABEND_TIMER	0x0800
AP_DATA_POSTING_BLOCKED	0x0810
AP_INVALID_VERB_SEGMENT	0x08F0
AP_DEALLOC_NORMAL	0x0900
AP_PATH_SWITCH_NOT_ALLOWED	0x0910
AP_CP_CP_SESS_ACT_FAILURE	0x0A10
AP_PROG_ERROR_NO_TRUNC	0x0C00
AP_PROG_ERROR_TRUNC	0x0D00
AP_PROG_ERROR_PURGING	0x0E00
AP_CONV_FAILURE_RETRY	0x0F00
AP_CONV_FAILURE_NO_RETRY	0x1000
AP_SVC_ERROR_NO_TRUNC	0x1100
AP_UNEXPECTED_DOS_ERROR	0x11F0
AP_SVC_ERROR_TRUNC	0x1200
AP_SVC_ERROR_PURGING	0x1300
AP_UNSUCCESSFUL	0x1400
AP_STACK_TOO_SMALL	0x15F0
AP_MIXED_API_USED	0x16F0
AP_IN_PROGRESS	0x17F0
AP_CNOS_PARTNER_LU_REJECT	0x1800
AP_COMPLETED	0x18F0
AP_CONVERSATION_TYPE_MIXED	0x1900
AP_NODE_STOPPING	0x1A00
AP_NODE_NOT_STARTED	0x1B00
AP_CANCELLED	0x2100
AP_BACKED_OUT	0x2200
AP_DUPLEX_TYPE_MIXED	0x2300
AP_LS_FAILURE	0x2300
AP_OPERATION_INCOMPLETE	0x4000
AP_OPERATION_NOT_ACCEPTED	0x4100

1 次戻りコード

AP_CONVERSATION_ENDED	0x4200
AP_ERROR_INDICATION	0x4300
AP_EXPD_NOT_SUPPORTED_BY_LU	0x4400
AP_BUFFER_TOO_SMALL	0x4500
AP_MEMORY_ALLOCATION_FAILURE	0x4600
AP_INVALID_VERB	0xFFFF

2 次戻りコード

以下の 2 次戻りコードは NOF アプリケーションで使用されます。

AP_AS_SPECIFIED	0x00000000
AP_ALLOCATION_ERROR_PENDING	0x00000300
AP_DEALLOC_ABEND_PROG_PENDING	0x00000600
AP_DEALLOC_ABEND_SVC_PENDING	0x00000700
AP_DEALLOC_ABEND_TIMER_PENDING	0x00000800
AP_UNKNOWN_ERROR_TYPE_PENDING	0x00001100
AP_BO_NO_RESYNC	0x00002408
AP_TRANS_PGM_NOT_AVAIL_NO_RETRY	0x00004C08
AP_INVALID_SET_PROT	0x00070000
AP_INVALID_DLU_NAME	0x00900000
AP_SEC_BAD_PASSWORD_EXPIRED	0x00FF0F08
AP_BAD_TP_ID	0x01000000
AP_BO_RESYNC	0x01002408
AP_INVALID_NEW_PROT	0x01070000
AP_DLC_ACTIVE	0x01100000
AP_NO_DEFAULT_DLU_DEFINED	0x01900000
AP_BAD_TPSID	0x01FF0000
AP_SEC_BAD_PASSWORD_INVALID	0x01FF0F08
AP_BAD_CONV_ID	0x02000000
AP_SEND_ERROR_LOG_LL_WRONG	0x02010000
AP_INVALID_SET_UNPROT	0x02070000
AP_INVALID_NUMBER_OF_NODE_ROWS	0x02080000
AP_DUPLICATE_CP_NAME	0x02100000
AP_INVALID_PU_ID	0x02900000
AP_NOT_OWNER	0x02FF0000
AP_SEC_BAD_USERID_REVOKED	0x02FF0F08
AP_BAD_LU_ALIAS	0x03000000
AP_BAD_DLOAD_ID	0x03000001
AP_BAD_REMOTE_LU_ALIAS	0x03000002
AP_SEND_ERROR_BAD_TYPE	0x03010000
AP_INVALID_NEW_UNPROT	0x03070000
AP_DUPLICATE_DEST_ADDR	0x03100000
AP_PU_ALREADY_ACTIVATING	0x03900000
AP_INSUFFICIENT_PRIVILEGES	0x03FF0000
AP_SEC_BAD_USERID_INVALID	0x03FF0F08
AP_ALLOCATION_FAILURE_NO_RETRY	0x04000000
AP_SEND_ERROR_BAD_STATE	0x04010000
AP_INVALID_SET_USER	0x04070000
AP_NODE_ROW_WGT_LESS_THAN_LAST	0x04080000
AP_CANT_MODIFY_PORT_NAME	0x04100000
AP_PU_ALREADY_DEACTIVATING	0x04900000
AP_INVALID_CALLBACK	0x04FF0000
AP_SEC_BAD_USERID_MISSING	0x04FF0F08
AP_ALLOCATION_FAILURE_RETRY	0x05000000
AP_BAD_ERROR_DIRECTION	0x05010000
AP_INVALID_DATA_TYPE	0x05070000
AP_TG_ROW_WGT_LESS_THAN_LAST	0x05080000
AP_DUPLICATE_PORT_NUMBER	0x05100000
AP_PU_ALREADY_ACTIVE	0x05900000
AP_BAD_TP_TYPE	0x05FF0000
AP_SEC_BAD_PASSWORD_MISSING	0x05FF0F08
AP_INVALID_STATS_TYPE	0x06070000
AP_DUPLICATE_PORT_NAME	0x06100000
AP_PU_NOT_ACTIVE	0x06900000
AP_ALREADY_REGISTERED	0x06FF0000

AP_SEC_BAD_GROUP_INVALID	0x06FF0F08
AP_AS_NEGOTIATED	0x07000000
AP_INVALID_TABLE_TYPE	0x07070000
AP_INVALID_DLC_NAME	0x07100000
AP_DLUS_REJECTED	0x07900000
AP_SEC_BAD_UID_REVOKED_IN_GRP	0x07FF0F08
AP_PORT_DEACTIVATED	0x08070000
AP_INVALID_DLC_TYPE	0x08100000
AP_DLUS_CAPS_MISMATCH	0x08900000
AP_SEC_BAD_UID_NOT_DEFD_TO_GRP	0x08FF0F08
AP_ALLOCATE_NOT_PENDING	0x09050000
AP_INVALID_SET_PASSWORD	0x09070000
AP_INVALID_NUMBER_OF_TG_ROWS	0x09080000
AP_INVALID_LINK_ACTIVE_LIMIT	0x09100000
AP_PU_FAILED_ACTPU	0x09900000
AP_SEC_BAD_UNAUTHRZD_AT_RLU	0x09FF0F08
AP_SNA_DEFD_COS_CANT_BE_CHANGE	0x0A080000
AP_SNA_DEFD_COS_CANT_BE_CHANGED	0x0A080000
AP_PU_NOT_RESET	0x0A900000
AP_SEC_BAD_UNAUTHRZD_FROM_LLU	0x0AFF0F08
AP_INVALID_NUM_PORTS_SPECIFIED	0x0B100000
AP_PU_OWNS_LUS	0x0B900000
AP_SEC_BAD_UNAUTHRZD_TO_TP	0x0BFF0F08
AP_INVALID_PORT_NAME	0x0C100000
AP_INVALID_FILTER_OPTION	0x0C900000
AP_SEC_BAD_INSTALL_EXIT_FAILED	0x0CFF0F08
AP_INVALID_PORT_TYPE	0x0D100000
AP_INVALID_STOP_TYPE	0x0D900000
AP_SEC_BAD_PROCESSING_FAILURE	0x0DFF0F08
AP_UNRECOGNIZED_DEACT_TYPE	0x0E050000
AP_PORT_ACTIVE	0x0E100000
AP_PU_ALREADY_DEFINED	0x0E900000
AP_NO_PORTS_DEFINED_ON_DLC	0x0F100000
AP_DEPENDENT_LU_NOT_SUPPORTED	0x0F900000
AP_INVALID_DLC	0x10050000
AP_COS_NAME_NOT_DEFD	0x10080000
AP_DUPLICATE_PORT	0x10100000
AP_INVALID_DSPU_SERVICES	0x10900000
AP_BAD_CONV_TYPE	0x11000000
AP_SNA_DEFD_COS_CANT_BE_DELETE	0x11080000
AP_SNA_DEFD_COS_CANT_BE_DELETED	0x11080000
AP_STOP_PORT_PENDING	0x11100000
AP_DSPU_SERVICES_NOT_SUPPORTED	0x11900000
AP_BAD_SYNC_LEVEL	0x12000000
AP_LU_NAU_ADDR_ALREADY_DEFD	0x12020000
AP_INVALID_SESSION_ID	0x12050000
AP_LINK_DEACT_IN_PROGRESS	0x12100000
AP_INVALID_DSPU_NAME	0x12900000
AP_BAD_SECURITY	0x13000000
AP_INVALID_NN_SESSION_TYPE	0x13050000
AP_LINK_DEACTIVATED	0x13100000
AP_PARTNER_NOT_FOUND	0x13200000
AP_PARTNER_NOT_RESPONDING	0x13300000
AP_ERROR	0x13400000
AP_DSPU_ALREADY_DEFINED	0x13900000
AP_BAD_RETURN_CONTROL	0x14000000
AP_INVALID_MAX_NEGOT_SESS_LIM	0x14020000
AP_INVALID_SET_COLLECT_STATS	0x14050000
AP_LINK_ACT_BY_REMOTE	0x14100000
AP_INVALID_SOLICIT_SSCP_SESS	0x14900000
AP_INVALID_BACK_LEVEL_SUPPORT	0x15000000
AP_INVALID_MODE_NAME	0x15020000
AP_INVALID_SET_COLLECT_NAMES	0x15050000
AP_LINK_ACT_BY_LOCAL	0x15100000
AP_INVALID_TG_NUMBER	0x15500000
AP_MISSING_CP_NAME	0x15510000
AP_MISSING_CP_TYPE	0x15520000

2 次戻りコード

AP_INVALID_CP_TYPE	0x15520000
AP_DUPLICATE_TG_NUMBER	0x15530000
AP_TG_NUMBER_IN_USE	0x15540000
AP_MISSING_TG_NUMBER	0x15550000
AP_PARALLEL_TGS_NOT_ALLOWED	0x15570000
AP_INVALID_BKUP_DLUS_NAME	0x15900000
AP_PIP_LEN_INCORRECT	0x16000000
AP_INVALID_RECV_PACING_WINDOW	0x16020000
AP_INVALID_SET_COLLECT_RSCVS	0x16050000
AP_SEC_REQUESTED_NOT_SUPPORTED	0x16900000
AP_NO_USE_OF_SNASVCMG	0x17000000
AP_INVALID_CNOS_SLIM	0x17020000
AP_LINK_NOT_DEFD	0x17100000
AP_INVALID_DUPLEX_SUPPORT	0x17900000
AP_UNKNOWN_PARTNER_MODE	0x18000000
AP_INVALID_TARGET_PACING_CNT	0x18020000
AP_PS_CREATION_FAILURE	0x18100000
AP_QUEUE_PROHIBITED	0x18900000
AP_INVALID_MAX_RU_SIZE_UPPER	0x19020000
AP_TP_ACTIVE	0x19100000
AP_INVALID_TEMPLATE_NAME	0x19900000
AP_INVALID_SNASVCMG_MODE_LIMIT	0x1A020000
AP_MODE_ACTIVE	0x1A100000
AP_CLASHING_NAU_RANGE	0x1A900000
AP_PLU_ACTIVE	0x1B100000
AP_INVALID_NAU_RANGE	0x1B900000
AP_INVALID_COS_SNASVCMG_MODE	0x1C020000
AP_INVALID_PLU_NAME	0x1C100000
AP_INVALID_NUM_DSLU_TEMPLATES	0x1C900000
AP_INVALID_DEFAULT_RU_SIZE	0x1D020000
AP_INVALID_SET_NEGOTIABLE	0x1D100000
AP_GLOBAL_TIMEOUT_NOT_DEFINED	0x1D900000
AP_INVALID_MIN_CONWINNERS	0x1E020000
AP_INVALID_MODE_NAME_SELECT	0x1E100000
AP_INVALID_RESOURCE_NAME	0x1E900000
AP_INVALID_RESPONSIBLE	0x1F100000
AP_INVALID_DLUS_RETRY_TIMEOUT	0x1F900000
AP_MODE_SESS_LIM_EXCEEDS_NEG	0x20020000
AP_INVALID_DRAIN_SOURCE	0x20100000
AP_INVALID_DLUS_RETRY_LIMIT	0x20900000
AP_CPSVCMG_ALREADY_DEFD	0x21020000
AP_INVALID_CN_NAME	0x21080000
AP_INVALID_DRAIN_TARGET	0x21100000
AP_TP_NAME_NOT_RECOGNIZED	0x21600810
AP_INVALID_MIN_CONLOSERS	0x21900000
AP_BAD_DUPLEX_TYPE	0x22000000
AP_INVALID_BYPASS_SECURITY	0x22020000
AP_DEF_LINK_INVALID_SECURITY	0x22080000
AP_INVALID_FORCE	0x22100000
AP_SYSTEM_TP_CANT_BE_CHANGED	0x22600810
AP_INVALID_MAX_RU_SIZE_LOW	0x22900000
AP_FDX_NOT_SUPPORTED_BY_LU	0x23000000
AP_TEST_INVALID_FOR_FDX	0x23010000
AP_INVALID_IMPLICIT_PLU_FORBID	0x23020000
AP_INVALID_PROPAGATION_DELAY	0x23080000
AP_SYSTEM_TP_CANT_BE_DELETED	0x23600810
AP_INVALID_MAX_RECV_PACING_WIN	0x23900000
AP_SEND_EXPD_INVALID_LENGTH	0x24010000
AP_INVALID_SPECIFIC_SECURITY	0x24020000
AP_INVALID_EFFECTIVE_CAPACITY	0x24080000
AP_INVALID_CLEANUP_TYPE	0x24100000
AP_INVALID_DYNAMIC_LOAD	0x24600810
AP_RU_SIZE_LOW_UPPER_MISMATCH	0x24900000
AP_RCV_EXPD_INVALID_LENGTH	0x25010000
AP_INVALID_DELAYED_LOGON	0x25020000
AP_INVALID_COS_NAME	0x25100000
AP_INVALID_ENABLED	0x25600810

AP_LU_ALREADY_ACTIVATING	0x25900000
AP_EXPD_BAD_RETURN_CONTROL	0x26010000
AP_INVALID_CNOS_PERMITTED	0x26020000
AP_PW_SUB_NOT_SUPP_ON_SESS	0x26050000
AP_INVALID_SESSION_LIMIT	0x26100000
AP_INVALID_PIP_ALLOWED	0x26600810
AP_LU_DEACTIVATING	0x26900000
AP_EXPD_DATA_BAD_CONV_STATE	0x27010000
AP_INVALID_DRAIN	0x27100000
AP_LU_ALREADY_ACTIVE	0x27900000
AP_INVALID_PRL_SESS_SUPP	0x28100000
AP_INVALID_MIN_CONTENTION_SUM	0x28900000
AP_INVALID_LU_NAME	0x29100000
AP_COMPRESSION_NOT_SUPPORTED	0x29900000
AP_MODE_NOT_RESET	0x2A100000
AP_INVALID_MAX_COMPRESS_LVL	0x2A900000
AP_MODE_RESET	0x2B100000
AP_INVALID_COMPRESSION	0x2B900000
AP_CNOS_REJECT	0x2C100000
AP_INVALID_EXCEPTION_INDEX	0x2C900000
AP_INVALID_OP_CODE	0x2D100000
AP_INVALID_MAX_LS_EXCEPTION	0x2D900000
AP_INVALID_DISABLE	0x2E900000
AP_INVALID_MODIFY_TEMPLATE	0x2F900000
AP_INVALID_ALLOW_TIMEOUT	0x30900000
AP_CONFIRM_ON_SYNC_LEVEL_NONE	0x31000000
AP_PIP_NOT_ALLOWED	0x31600810
AP_TRANS_PGM_NOT_AVAIL_RETRY	0x31604B08
AP_POST_ON_RECEIPT_BAD_FILL	0x31900000
AP_CONFIRM_BAD_STATE	0x32000000
AP_UNKNOWN_USER	0x32100000
AP_POST_ON_RECEIPT_BAD_STATE	0x32900000
AP_CONFIRM_NOT_LL_BDY	0x33000000
AP_NO_PROFILES	0x33100000
AP_INVALID_HPR_SUPPORT	0x33900000
AP_CONFIRM_INVALID_FOR_FDX	0x34000000
AP_CONVERSATION_TYPE_MISMATCH	0x34600810
AP_INVALID_LU_MODEL	0x34900000
AP_INVALID_MODEL_NAME	0x35900000
AP_TOO_MANY_PROFILES	0x36100000
AP_INVALID_CRYPTOGRAPHY	0x36900000
AP_INVALID_UPDATE_TYPE	0x37100000
AP_INVALID_CLU_CRYPTOGRAPHY	0x37900000
AP_DIR_ENTRY_PARENT	0x38100000
AP_INVALID_RESOURCE_TYPES	0x38900000
AP_NODE_ALREADY_STARTED	0x39100000
AP_CHECKSUM_FAILED	0x39900000
AP_NODE_FAILED_TO_START	0x3A100000
AP_DATA_CORRUPT	0x3A900000
AP_LU_ALREADY_DEFINED	0x3B100000
AP_INVALID_RETRY_FLAGS	0x3B900000
AP_IMPLICIT_LU_DEFINED	0x3C100000
AP_DELAYED_VERB_PENDING	0x3C900000
AP_PORT_INACTIVE	0x3D100000
AP_DSLU_ACTIVE	0x3D900000
AP_ACTIVATION_LIMITS_REACHED	0x3E100000
AP_ACTIVATION_LIMITS_REACHED	0x3E100000
AP_INVALID_BRANCH_LINK_TYPE	0x3E900000
AP_PARALLEL_TGS_NOT_SUPPORTED	0x3F100000
AP_INVALID_BRNN_SUPPORT	0x3F900000
AP_DLC_INACTIVE	0x40100000
AP_BRNN_SUPPORT_MISSING	0x40900000
AP_CONFIRMED_BAD_STATE	0x41000000
AP_NO_LINKS_DEFINED	0x41100000
AP_SYNC_LEVEL_NOT_SUPPORTED	0x41600810
AP_INVALID_UPLINK	0x41900000
AP_CONFIRMED_INVALID_FOR_FDX	0x42000000

2 次戻りコード

AP_STOP_DLC_PENDING	0x42100000
AP_INVALID_DOWNLINK	0x42900000
AP_INVALID_LS_ROLE	0x43100000
AP_INVALID_IMPLICIT_UPLINK	0x43900000
AP_INVALID_BTU_SIZE	0x44100000
AP_INVALID_ROCP_NAME	0x44900000
AP_LAST_LINK_ON_ACTIVE_PORT	0x45100000
AP_INVALID_REG_WITH_NN	0x45900000
AP_DYNAMIC_LOAD_ALREADY_REGD	0x46100000
AP_LS_PENDING_RETRY	0x46900000
AP_INVALID_LIST_OPTION	0x47100000
AP_INVALID_COS_TABLE_VERSION	0x47900000
AP_INVALID_RES_NAME	0x48100000
AP_CFRTP_REQUIRED_FOR_MLTG	0x48900000
AP_INVALID_RES_TYPE	0x49100000
AP_INVALID_MLTG_PAC_ALGORITHM	0x49900000
AP_INVALID_ADJ_NNCP_NAME	0x4A100000
AP_LIM_RESRCE_INVALID_FOR_MLTG	0x4A900000
AP_INVALID_NODE	0x4B100000
AP_AUTO_ACT_INVALID_FOR_MLTG	0x4B900000
AP_INVALID_ORIGIN_NODE	0x4C100000
AP_MLTG_LS_VISIBILITY_MISMATCH	0x4C900000
AP_INVALID_TG	0x4D100000
AP_SLTG_LINK_ACTIVE	0x4D900000
AP_INVALID_FQPCID	0x4E100000
AP_MLTG_LINK_PROPERTIES_DIFFER	0x4E900000
AP_INVALID_POOL_NAME	0x4F100000
AP_INVALID_ADJ_CP_NAME	0x4F900000
AP_BAD_TYPE	0x50020000
AP_INVALID_NAU_ADDRESS	0x50100000
AP_INVALID_ENABLE_POOL	0x50300000
AP_INVALID_SEND_TERM_SELF	0x50900000
AP_DEALLOC_BAD_TYPE	0x51000000
AP_LU_NAME_POOL_NAME_CLASH	0x51100000
AP_SECURITY_NOT_VALID	0x51600F08
AP_INVALID_TERM_METHOD	0x51900000
AP_DEALLOC_FLUSH_BAD_STATE	0x52000000
AP_INVALID_PRIORITY	0x52100000
AP_INVALID_DISABLE_BRANCH_AWRN	0x52900000
AP_DEALLOC_CONFIRM_BAD_STATE	0x53000000
AP_INVALID_DNST_LU_NAME	0x53100000
AP_INVALID_SHARING_PROHIBITED	0x53900000
AP_INVALID_HOST_LU_NAME	0x54100000
AP_INVALID_LINK_SPEC_FORMAT	0x54900000
AP_DEALLOC_NOT_LL_BDY	0x55000000
AP_PU_NOT_DEFINED	0x55100000
AP_INVALID_CN_TYPE	0x55900000
AP_INVALID_PU_NAME	0x56100000
AP_INVALID_PU_TYPE	0x56600000
AP_INCONSISTENT_BEST_EFFORT	0x56900000
AP_DEALLOC_LOG_LL_WRONG	0x57000000
AP_CNOS_MODE_NAME_REJECT	0x57010000
AP_INVALID_MAX_IFRM_RCVD	0x57100000
AP_INVALID_CN_TG	0x57900000
AP_INVALID_SYM_DEST_NAME	0x58100000
AP_SEC_BAD_PROTOCOL_VIOLATION	0x58600F08
AP_INVALID_LINK_SPEC_DATA	0x58900000
AP_INVALID_LENGTH	0x59100000
AP_DLC_UI_ONLY	0x59900000
AP_INVALID_ISR_THRESHOLDS	0x5A100000
AP_ADJ_CP_WRONG_TYPE	0x5A900000
AP_BAD_PARTNER_LU_ALIAS	0x5B010000
AP_INVALID_NUM_LUS	0x5B100000
AP_CP_CP_SESS_ALREADY_ACTIVE	0x5B900000
AP_EXCEEDS_MAX_ALLOWED	0x5C010000
AP_CANT_DELETE_ADJ_ENDNODE	0x5C100000
AP_NO_ACTIVE_CP_CP_LINK	0x5C900000

AP_LU_MODE_SESSION_LIMIT_ZERO	0x5D010000
AP_INVALID_RESOURCE_TYPE	0x5D100000
AP_PU_CONC_NOT_SUPPORTED	0x5E100000
AP_INVALID_IMPL_APPN_LINKS_LEN	0x5E900000
AP_CNOS_COMMAND_RACE_REJECT	0x5F010000
AP_DLUR_NOT_SUPPORTED	0x5F100000
AP_INVALID_LIMIT_ENABLE	0x5F900000
AP_INVALID_SVCMG_LIMITS	0x60010000
AP_INVALID_RTP_CONNECTION	0x60100000
AP_INVALID_LS_ATTRIBUTE	0x60900000
AP_FLUSH_NOT_SEND_STATE	0x61000000
AP_PATH_SWITCH_IN_PROGRESS	0x61100000
AP_HPR_NOT_SUPPORTED	0x62100000
AP_SOME_ENABLED	0x62900000
AP_RTP_NOT_SUPPORTED	0x63100000
AP_NONE_ENABLED	0x63900000
AP_COS_TABLE_FULL	0x64100000
AP_INCONSISTENT_IMPLICIT	0x64900000
AP_INVALID_DAYS_LEFT	0x65100000
AP_INVALID_PREFER_ACTIVE_DLUS	0x65900000
AP_ANYNET_NOT_SUPPORTED	0x66100000
AP_INVALID_PERSIST_PIPE_SUPP	0x66900000
AP_INVALID_DISCOVERY_SUPPORT	0x67100000
AP_ACTIVATION_PROHIBITED	0x67900000
AP_SESSION_FAIL_ALREADY_REGD	0x68100000
AP_INVALID_NULL_ADDR_MEANING	0x68900000
AP_CANT_MODIFY_VISIBILITY	0x69100000
AP_INVALID_CPLU_SYNCPT_SUPPORT	0x69900000
AP_CANT_MODIFY_WHEN_ACTIVE	0x6A100000
AP_INVALID_CPLU_ATTRIBUTES	0x6A900000
AP_INVALID_BASE_NUMBER	0x6B100000
AP_INVALID_REG_LEN_SUPPORT	0x6B900000
AP_DEACT_CG_INVALID_CGID	0x6C020000
AP_INVALID_NAME_ATTRIBUTES	0x6C100000
AP_LUNAME_CGID_MISMATCH	0x6C900000
AP_NAU_ADDRESS_MISMATCH	0x6D100000
AP_INVALID_DDDL_U_OFFLINE	0x6D900000
AP_POSTED_DATA	0x6E100000
AP_POSTED_NO_DATA	0x6F100000
AP_DEF_PLU_INVALID_FQ_NAME	0x74020000
AP_DLC_DEACTIVATING	0x86020000
AP_INVALID_WILDCARD_NAME	0x8C020000
AP_DUPLICATE	0x8D020000
AP_LU_NAME_WILDCARD_NAME_CLASH	0x8E020000
AP_INVALID_USERID	0x90020000
AP_INVALID_PASSWORD	0x91020000
AP_INVALID_PROFILE	0x93020000
AP_INVALID_TP_NAME	0xA0020000
AP_P_TO_R_INVALID_TYPE	0xA1000000
AP_INVALID_CONV_TYPE	0xA1020000
AP_P_TO_R_NOT_LL_BDY	0xA2000000
AP_P_TO_R_NOT_SEND_STATE	0xA3000000
AP_INVALID_SYNC_LEVEL	0xA3020000
AP_P_TO_R_INVALID_FOR_FDX	0xA5000000
AP_INVALID_LINK_NAME_SPECIFIED	0xB0020000
AP_RCV_AND_WAIT_BAD_STATE	0xB1000000
AP_INVALID_LU_ALIAS	0xB1020000
AP_RCV_AND_WAIT_NOT_LL_BDY	0xB2000000
AP_INVALID_NUM_LS_SPECIFIED	0xB2020000
AP_PLU_ALIAS_CANT_BE_CHANGED	0xB3020000
AP_PLU_ALIAS_ALREADY_USED	0xB4020000
AP_RCV_AND_WAIT_BAD_FILL	0xB5000000
AP_INVALID_AUTO_ACT_SUPP	0xB5020000
AP_CANT_DELETE_IMPLICIT_LU	0xB6020000
AP_FORCED	0xB7020000
AP_INVALID_LS_NAME	0xB7030000
AP_INVALID_LFSID_SPECIFIED	0xB7040000

2 次戻りコード

AP_INVALID_FILTER_TYPE	0xB7050000
AP_INVALID_MESSAGE_TYPE	0xB7060000
AP_CANT_DELETE_CP_LU	0xB7070000
AP_ALL_RESOURCES_NOT_DEFINED	0xB7090000
AP_INVALID_LIST_TYPE	0xB70A0000
AP_RESOURCE_NAME_NOT_ALLOWED	0xB70B0000
AP_LU_ALIAS_CANT_BE_CHANGED	0xB8020000
AP_LU_ALIAS_ALREADY_USED	0xB9020000
AP_INVALID_LINK_ENABLE	0xBA020000
AP_INVALID_CLU_COMPRESSION	0xBB020000
AP_INVALID_DLUR_SUPPORT	0xBC020000
AP_ALREADY_STARTING	0xC0010000
AP_RCV_IMMEDIATE_BAD_STATE	0xC1000000
AP_INVALID_LINK_NAME	0xC1010000
AP_INVALID_USER_DEF_1	0xC3010000
AP_RCV_IMMEDIATE_BAD_FILL	0xC4000000
AP_INVALID_USER_DEF_2	0xC4010000
AP_INVALID_NODE_TYPE	0xC4020000
AP_INVALID_USER_DEF_3	0xC5010000
AP_INVALID_NAME_LEN	0xC5020000
AP_INVALID_NETID_LEN	0xC6020000
AP_INVALID_NODE_TYPE_FOR_HPR	0xC8020000
AP_INVALID_MAX_DECOMPRESS_LVL	0xC9020000
AP_INVALID_CP_NAME	0xCA010000
AP_INVALID_COMP_IN_SERIES	0xCA020000
AP_INVALID_LIMITED_RESOURCE	0xCE010000
AP_RCV_AND_POST_BAD_STATE	0xD1000000
AP_INVALID_BYTE_COST	0xD1010000
AP_RCV_AND_POST_NOT_LL_BDY	0xD2000000
AP_RCV_AND_POST_BAD_FILL	0xD5000000
AP_INVALID_TIME_COST	0xD6010000
AP_BAD_RETURN_STATUS_WITH_DATA	0xD7000000
AP_LOCAL_CP_NAME	0xD7010000
AP_LS_ACTIVE	0xDA010000
AP_INVALID_FQ_OWNING_CP_NAME	0xDB020000
AP_R_T_S_BAD_STATE	0xE1000000
AP_R_T_S_INVALID_FOR_FDX	0xE2000000
AP_BAD_LL	0xF1000000
AP_SEND_DATA_NOT_SEND_STATE	0xF2000000
AP_CP_OR_SNA_SVCMG_UNDELETABLE	0xF3010000
AP_SEND_DATA_INVALID_TYPE	0xF4000000
AP_DEL_MODE_DEFAULT_SPCD	0xF4010000
AP_SEND_DATA_CONFIRM_SYNC_NONE	0xF5000000
AP_MODE_NAME_NOT_DEFD	0xF5010000
AP_SEND_DATA_NOT_LL_BDY	0xF6000000
AP_MODE_UNDELETABLE	0xF6010000
AP_SEND_TYPE_INVALID_FOR_FDX	0xF7000000
AP_INVALID_FQ_LU_NAME	0xFD010000
AP_INVALID_PARTNER_LU	0xFE010000
AP_INVALID_LOCAL_LU	0xFF010000

付録 B. 共通戻りコード

この付録では、すべての NOF verb に共通な 1 次戻りコードおよび 2 次戻りコードについて説明します。

特定の verb または 1 グループの verb に固有の戻りコードについては、49 ページの『第 3 章 NOF API verb』の各 verb の項で説明しています。

通信サブシステムが活動状態でない場合

必要なコンポーネントがアクティブでないために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_COMM_SUBSYSTEM_ABENDED

secondary_rc

次のいずれかです。

AP_LOCAL_ABENDED

Communications Server for Linux ソフトウェアが停止しました。

AP_TARGET_ABENDED

ターゲット・ノードが停止したか、ターゲット・ノードへの通信パスに障害が起きました。

primary_rc

AP_COMM_SUBSYSTEM_NOT_LOADED

Communications Server for Linux ソフトウェアは活動状態ではありません。

secondary_rc

未使用。

primary_rc

AP_NODE_NOT_STARTED

ターゲット・ノードが開始されていません。

secondary_rc

未使用。

primary_rc

AP_NODE_STOPPING

ターゲット・ノードは (TERM_NODE verb の結果として) 停止処理中です。

secondary_rc

未使用。

指示

この戻りコードはエラーを示しません。

アプリケーションが REGISTER_INDICATION_SINK を使用して構成ディレクティブまたは SNA ネットワーク・ファイル指示を受け取るように登録されている場合、別の NOF API アプリケーションまたは Communications Server for Linux のコンポーネントがターゲット・ファイルまたはターゲット・ノードの構成を変更するごとに、Communications Server for Linux は指示を送信します。この指示のフォーマットは、構成を変更した NOF verb に対して戻される VCB のフォーマットと同じです。Communications Server for Linux は、この 1 次戻りコードを設定することによって、戻される VCB が構成ディレクティブまたは SNA ネットワーク・ファイル指示であって、アプリケーションが発行した verb への応答ではないことを示します。これによって、アプリケーションはそれ自体の verb 戻り値と、他のアプリケーションで発行された verb の結果の指示とを区別することができます。

primary_rc

AP_INDICATION

secondary_rc

値は次のいずれかです。

AP_EXTRA_DATA_LOST

Communications Server for Linux は、この指示について完全な VCB を戻すための十分なストレージを割り振ることができませんでした。戻された情報は不完全です。変更されたコンポーネントに関する詳しい情報を入手するには、アプリケーション側で適切な QUERY_* verb を発行してください。

(ゼロ) この指示についての完全な VCB が戻ります。

無効な機能

ノードがこの verb を有効な verb として認識していないために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_INVALID_VERB

opcode パラメーターがどの NOF verb の命令コードにも設定されていないか、またはこのパラメーターで識別される verb が Communications Server for Linux のこのバージョンでサポートされていないため、使用できません。

secondary_rc

未使用。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

指定された *opcode* パラメーターで識別される NOF verb は、ターゲット・ノードの構成でサポートされていないため使用できません。

secondary_rc
未使用。

無効な verb セグメント

WINDOWS

VCB がデータ・セグメント内に含まれていないために verb の実行が失敗すると、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_INVALID_VERB_SEGMENT

verb 制御ブロックがデータ・セグメントの範囲を超えています。
verb は実行されませんでした。

2 次戻りコードが戻されません。



パラメーターの検査

パラメーター・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

次のいずれかです。

AP_INVALID_FORMAT

予約済みのパラメーターの *format* がゼロに設定されていませんでした。

AP_INVALID_TARGET_HANDLE

指定されたターゲット・ハンドルが有効ではありません。

AP_INVALID_TARGET

この verb は、指定したターゲットに対しては発行できません。たとえば、QUERY_PARTNER_LU は、LU の現在の使用状況についての情報を戻すものであり、実行中のノードに対してのみ発行できます。ファイルに対して発行した場合は有効ではありません。

AP_INVALID_TARGET_MODE

この verb は、現行モードでは発行できません。たとえば、読み取り専用モードで発行できるのは QUERY_* verb のみです。DEFINE_*、DELETE_*、START_*、STOP_* の各 verb は、このモードでは有効ではありません。

AP_NOT_SERVER

この戻りコードはクライアントで NOF アプリケーション・プログ

パラメーターの検査

ラムを実行している場合のみ発生します。発行した verb はクライアントでは無効です。この verb はサーバー上でのみ発行できません。

AP_SYNC_NOT_ENABLED

アプリケーションは同期 NOF エントリー・ポイントを使用して、コールバック・ルーチン内でこの verb を発行しました。コールバック・ルーチンから発行する verb では、非同期エントリー・ポイントを使用する必要があります。

状態の検査

状態エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

次のいずれかです。

AP_CANT_MODIFY_VISIBILITY

リソースの定義に使用しようとした名前は、Communications Server for Linux でプログラム内部で使用するために予約済みです。別の名前を選んでください。

AP_FILE_LOCK_FAILED

アプリケーションはコミット・モードへ変更するために SET_PROCESSING_MODE を発行しましたが、Communications Server for Linux は、構成ファイルのロックの取得に失敗しました。別の NOF API アプリケーションまたは Communications Server for Linux のコンポーネントが、既にそのファイルにアクセスしている可能性があります。

AP_FILE_UNLOCK_FAILED

アプリケーションはコミット・モードから他のモードの 1 つに変更するために SET_PROCESSING_MODE を発行しましたが、Communications Server for Linux は、構成ファイルのロックの解除に失敗しました。このエラーが起これば、Communications Server for Linux はファイルを解放するために、ファイルへのアプリケーションのハンドルを閉じます。アプリケーションでは、OPEN_FILE を再発行して新しいファイル・ハンドルを入手してから、このファイルに対する他の verb の発行を試みてください。

AP_FILE_UNAVAILABLE

ターゲット・ファイルとの接続が失われました。

AP_NOT_MASTER

ターゲット・ファイルは、マスター・サーバーではないサーバー上のドメイン構成ファイルまたは **sna.net** ファイルです。これらのファイルを変更する verb は、マスター・サーバー上のファイルに対して発行される必要があります。

AP_SYNC_PENDING

この verb は同期 NOF API エントリ・ポイントを使用して発行されましたが、別の同期 verb が進行中でした。同期 verb は常に 1 つしか進行できません。

システム・エラー

オペレーティング・システム・エラーのために verb が実行されないと、Communications Server for Linux は、以下のパラメーターを戻します。

primary_rc

AP_UNEXPECTED_SYSTEM_ERROR

verb の処理中にオペレーティング・システム・コールが異常終了しました。

secondary_rc

この場合での 2 次戻りコードは、オペレーティング・システム・コールからの戻りコードです。

UNIX

オペレーティング・システムの戻りコードの意味については、エラーが起こったコンピューター上のファイル `/usr/include/errno.h` を参照してください。通常、戻りコードには、メモリー不足などの状態が示されています。

WINDOWS

オペレーティング・システムの戻りコードの意味については、オペレーティング・システムの資料を参照してください。

問題が解決しない場合は、システム管理者に相談してください。

この verb がターゲットの構成の変更 (DEFINE_*、DELETE_* など) またはアクションの実行 (START_* など) のために発行された場合は、アプリケーションで適切な QUERY_* verb を発行して、変更またはアクションが成功しているかどうか調べてください。特に、複数のデータ構造体を含む DEFINE_* verb または DELETE_* verb の処理中にこのエラーが起こった場合は、変更が完了していない可能性があります。

付録 C. 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。

〒106-8711
東京都港区六本木 3-2-12
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

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

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾: 本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。お客様は、IBM のアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物にも、次のように、著作権表示を入れていただく必要があります。® (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。® Copyright International Business Machines Corporation. 1998, 2007. All rights reserved.

商標

以下は、International Business Machines Corporation の米国およびその他の国における商標です。

Advanced Peer-to-Peer Networking®	MVS/ESA™
AIX	NetView
Application System/400®	Operating System/2®
APPN	Operating System/400®
AS/400	OS/2®
CICS®	OS/400®
DB2®	PowerPC®
Enterprise System/3090™	PowerPC Architecture™
Enterprise System/4381™	pSeries
Enterprise System/9000®	S/390®
ES/3090™	System p5
ES/9000®	System/390®
eServer™	VSE/ESA™
IBM	VTAM
IBMLink™	WebSphere®
IMS™	z/OS
MVS™	z9

Java™ およびすべての Java ベースの商標およびロゴは、Sun Microsystems, Inc. の米国およびその他の国における商標または登録商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Intel 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商標です。

Microsoft、Windows、Windows 2003、Windows XP、Windows Vista、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。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

参考文献

以下の IBM 資料では、本書で説明しているトピックについての情報を記載しています。資料は、次のトピック別に大きく分けてあります。

- Communications Server for Linux バージョン 6.2.3
- システム・ネットワーク体系 (SNA)
- ホスト構成
- z/OS® Communications Server
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- X.25
- 拡張プログラム間通信機能 (APPC)
- プログラミング
- その他の IBM ネットワーキング・トピック

Communications Server for Linux ライブラリー関連の資料については、簡単な説明を付記してあります。その他の資料については、タイトル、資料番号を記し、一部の資料については本書で使用している略称タイトルを記しています。

Communications Server for Linux バージョン 6.2.3 関連資料

Communications Server for Linux ライブラリー関連資料として、以下のものがあります。なお、これらの資料のソフトコピー版が CD-ROM で提供されています。CD-ROM のソフトコピー・ファイルへのアクセスの方法については、「*IBM Communications Server for Linux 入門*」を参照してください。これらのソフトコピー・ブックをシステムにインストールするには、9 ~ 15MB のハードウェア・ディスク・スペースが必要になります (このスペースは、どの各国語バージョンをインストールするかによって異なります)。

- *IBM Communications Server for Linux 入門* (GC88-9996-02)

この資料は Communications Server for 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、および操作に関する解説です。

- *IBM Communications Server for Linux 管理ガイド* (SC88-9999-02)

この資料は SNA および Communications Server for Linux の概要、および Communications Server for Linux の構成と操作に関する解説です。

- *IBM Communications Server for Linux 管理コマンド解説書* (SC88-9998-01)

この資料は SNA および Communications Server for Linux のコマンドに関する情報を提供します。

- *IBM Communications Server for Linux CPI-C Programmer's Guide* ()

この資料は、“C” または Java の経験があるプログラマーを対象として、Communications Server for Linux CPI 通信 API を使用する SNA トランザクション・プログラムの作成に関する情報を提供します。

- *IBM Communications Server for Linux APPC Programmer's Guide ()*

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 Linux LUA Programmer's Guide ()*

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 Linux CSV Programmer's Guide ()*

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 Linux MS Programmer's Guide ()*

この資料では、Management Services (MS) API を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。

- *IBM Communications Server for Linux NOF プログラマーズ・ガイド (SC88-8591-01)*

この資料では、Node Operator Facility (NOF) API を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。

- *IBM Communications Server for Linux 診断ガイド (GC88-8601-02)*

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 Linux APPC Application Suite User's Guide ()*

この資料は Communications Server for Linux で使用される APPC アプリケーションに関する情報を提供します。

- *IBM Communications Server for Linux Glossary ()*

この資料は、IBM Communications Server for Linux ライブラリー資料全体にわたって使用される用語とその定義を包括的に収録しています。

システム・ネットワーク体系 (SNA) 関連資料

以下の資料には、SNA ネットワークに関する情報について記載されています。

- *Systems Network Architecture: Format and Protocol Reference Manual—Architecture Logic for LU Type 6.2 (SC30-3269)*
- *Systems Network Architecture: Formats (GA27-3136)*
- *Systems Network Architecture: Guide to SNA Publications (GC30-3438)*
- *Systems Network Architecture: Network Product Formats (LY43-0081)*
- *Systems Network Architecture: Technical Overview (GC30-3073)*

- *Systems Network Architecture: APPN Architecture Reference* (SC30-3422)
- *Systems Network Architecture: Sessions between Logical Units* (GC20-1868)
- *Systems Network Architecture: LU 6.2 Reference—Peer Protocols* (SC31-6808)
- *Systems Network Architecture: Transaction Programmer's Reference Manual for LU Type 6.2* (GC30-3084)
- *Systems Network Architecture: 3270 Datastream Programmer's Reference* (GA23-0059)
- *Networking Blueprint Executive Overview* (GC31-7057)
- *Systems Network Architecture: Management Services Reference* (SC30-3346)

ホスト構成関連資料

以下の資料には、ホスト構成に関する情報について記載されています。

- *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 (HCD) ユーザーズ・ガイド* (SC88-6630)

z/OS Communications Server 関連資料

以下の資料には、z/OS Communications Server に関する情報について記載されています。

- *z/OS V1R7 Communications Server: SNA ネットワーク導入の手引き (MVS/ESA)* (SC88-8928)
- *z/OS V1R7 Communications Server: SNA Diagnostics* (Vol 1: GC31-6850、Vol 2: GC31-6851)
- *z/OS V1R6 Communications Server: リソース定義解説書* (SC88-8929)

TCP/IP 関連資料

以下の資料には、Transmission Control Protocol/Internet Protocol (TCP/IP) ネットワーク・プロトコルに関する情報について記載されています。

- *z/OS V1R7 Communications Server: IP 構成ガイド* (SC88-8926)
- *z/OS V1R7 Communications Server: IP 構成解説書* (SC88-8927)
- *z/VM V5R1 TCP/IP 計画およびカスタマイズ* (SD88-6453)

X.25 関連資料

以下の資料には、X.25 ネットワーク・プロトコルに関する情報について記載されています。

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

APPC 関連資料

以下の資料には、拡張プログラム間通信機能 (APPC) に関する情報について記載されています。

- *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

以下の資料には、プログラミングに関する情報について記載されています。

- *Common Programming Interface Communications CPI-C Reference* (SC26-4399)
- *Communications Server for OS/2 バージョン 4 アプリケーション・プログラミングの手引き* (SC88-5585)

その他の IBM ネットワーキング関連資料

以下の資料には、Communications Server for Linux に関連するその他のトピックに関する情報について記載されています。

- *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]

アクセス・リスト, 会話セキュリティ 227
エラー・ログ・ファイル 474, 738
エントリー・ポイント
 AIX または Linux 28
 Windows 35
エンド・ノード 45
オープンする, 構成ファイル 318

[カ行]

カーネル・コンポーネント, メモリーの使用 457, 737
活動化, セッションの 50
監査ログ・ファイル 474, 738
管理サービス
 活動状態であるトランザクション 323
 デフォルト PU 94
 フォーカル・ポイント 123
関連資料 xxiii
クライアント
 照会 631
クライアント/サーバー操作 5
クローズ, 構成ファイルの 69
コールバック・ルーチン
 概要 33
 概要, Windows 40
 要件 34
 REGISTER_* verb に提供する 34
構成, ノード 2
構成の指示 24
構成ファイル
 オープン 318
 クローズ 69
 ドメイン・リソース 2
 ノード 2
 ヘッダー情報 110, 421
高速ログオン 241
子プロセス 34
コンパイルとリンク
 Windows 41

[サ行]

サーバー 5
サイド情報, CPI-C 91, 365

指示
 概要 23, 777
 登録 719
 登録解除 774
使用可能バッファの照会 334
状況の指示 24
使用量ログ・ファイル 474, 738
処理モード 43, 744
セキュリティ・アクセス・リストの削除 299
セッション限度
 初期化 314
 リセット 725

[タ行]

ターゲット, NOF verb の 42
ターゲット・ハンドル 29, 31
 Windows 36, 38
ダウストリーム LU 111, 115, 422
ダウストリーム LU 範囲の削除 274
ダウストリーム LU 範囲の定義 115
ダウストリーム PU 432
中央ロギング 337, 338, 731
データ・ファイル
 呼び出し可能 TP 3
 TP 定義 3
ディレクトリー統計 389
ディレクトリー・エントリー
 削除 269
 情報の入手 376
 定義 97
 LU 384
同期エントリー・ポイント 29
 AIX または Linux 28
 Windows 35, 36
統計
 ネットワーク・トポロジー 567
 ポート 664
 LS 664
登録, 指示を受け取るための 719
ドメイン構成 5
ドメイン構成ファイル 2
 複数のサーバー上 5
ドメイン・リソース, 構成ファイル 2
トレース・タイプ
 照会 714
 設定 751
 ノード DLC トレース 54
 CS トレース 370, 732
 TN サーバー・トレース 698, 747
トレース・ファイル 712, 748

[ナ行]

ネットワーク・トポロジー
照会 468, 561, 571
統計 567
ネットワーク・ノード
制約事項 45
トポロジー 561, 571
ノード
インプリメンテーション 3
オプション 598
開始 312
照会 580, 595
制限 598
接続 70
定義 189
停止 773
リソース使用量 598
ノード構成ファイル 2
ノード・タイプ、APPN 45

[ハ行]

パートナー LU
位置決めの方法 164, 284, 505
情報の入手 602, 610
定義 203
パートナー LU 定義の照会 610
パスワード
会話セキュリティ 259, 716
セッション・レベル・セキュリティ 524
LU-LU 179, 524
バックアップ・サーバー 5, 660
削除 263
追加 53
非活動化、セッションの
LU タイプ 0-3 74
LU タイプ 6.2 75
非同期エントリー・ポイント
概要 30
コールバック・ルーチン 33
コールバック・ルーチン、Windows 40
AIX または Linux 28
Windows 36, 37
プール、LU 181, 528
フォーカル・ポイント 123, 441
複数のプロセス 34
変更、セッション限度の 64
ポート
開始 763
照会 615
定義 206
停止 771
統計 664

[マ行]

マスター・サーバー 5
メモリー使用量、カーネル・コンポーネント 457, 737
モード 542, 549
定義 183
COS へのマッピング 555
戻りコード
1 次 843
2 次 844
戻りコード、共通 851

[ヤ行]

ユーザー ID、会話セキュリティ 259, 716
呼び出し可能 TP
情報の入手 331
データ・ファイル 3
定義 252
呼び出し可能 TP データ・ファイル 252

[ラ行]

ライセンス制限 598
リスト・オプション、QUERY_* verb の 47
リモート LU への通信パス 58
リンク・ステーションの経路指定
削除 284
照会 505
定義 164
ローカル LU
会話 348
照会 459
セッション 651
定義 130
ローカル・トポロジー 468
ログ・ファイル 474, 738
ログ・メッセージ
中央ロギング 337, 338
ログ・メッセージ、中央ロギング 731
ログ・メッセージ・タイプ 446, 476, 735, 742

[数字]

1 次戻りコード 843
16 進値、NOF パラメーターの 49
2 次戻りコード 844

A

ACTIVATE_SESSION 50
ADD_BACKUP 53
ADD_DLC_TRACE 54
AIX アプリケーション
コンパイルとリンク 34

AIX アプリケーションのコンパイル 34
AIX アプリケーションのリンク 34
APING 58
APPN ノード 3

C

CHANGE_SESSION_LIMIT 64
CLOSE_FILE 69
CN 81, 340
CN ポート 344
comp_proc (コールバック・ルーチン) 32
 Windows 39
CONFIG_INDICATION 24, 777
CONNECT_NODE 70
corr (相関関係子) 32, 34
 Windows 39, 40
COS
 情報の入手 352
 定義 85
 ノード行 355
 TG 行 360
COS マッピングへの照会モード 555
CPI-C、サイド情報 91, 365

D

DEACTIVATE_CONV_GROUP 72
DEACTIVATE_LU_0_TO_3 74
DEACTIVATE_SESSION 75
DEFINE_ADJACENT_LEN_NODE 78
DEFINE_CN 81
DEFINE_COS 85
DEFINE_CPIC_SIDE_INFO 91
DEFINE_DEFAULTS 95
DEFINE_DEFAULT_PU 94
DEFINE_DIRECTORY_ENTRY 97
DEFINE_DLC 100
DEFINE_DLUR_DEFAULTS 108
DEFINE_DOMAIN_CONFIG_FILE 110
DEFINE_DOWNSTREAM_LU 111
DEFINE_DSPU_TEMPLATE 119
DEFINE_FOCAL_POINT 123
DEFINE_INTERNAL_PU 126
DEFINE_LOCAL_LU 130
DEFINE_LS 135
DEFINE_LS_ROUTING verb 164
DEFINE_LU62_TIMEOUT 167
DEFINE_LU_0_TO_3 169
DEFINE_LU_0_TO_3_RANGE 173
DEFINE_LU_LU_PASSWORD 179
DEFINE_LU_POOL 181
DEFINE_MODE 183
DEFINE_PARTNER_LU 203
DEFINE_PORT 206

DEFINE_RCF_ACCESS 223
DEFINE_RTP_TUNING 225
DEFINE_SECURITY_ACCESS_LIST 227
DEFINE_TN3270_ACCESS 230
DEFINE_TN3270_ASSOCIATION 238
DEFINE_TN3270_DEFAULTS 239
DEFINE_TN3270_EXPRESS_LOGON 241
DEFINE_TN3270_SSL_LDAP 243
DEFINE_TN_REDIRECT 245
DEFINE_TP 252
DEFINE_TP_LOAD_INFO 256
DEFINE_USERID_PASSWORD 259
DELETE_ADJACENT_LEN_NODE 261
DELETE_BACKUP 263
DELETE_CN 265
DELETE_COS 267
DELETE_CPIC_SIDE_INFO 268
DELETE_DIRECTORY_ENTRY 269
DELETE_DLC 271
DELETE_DOWNSTREAM_LU 272
DELETE_DSPU_TEMPLATE 276
DELETE_FOCAL_POINT 278
DELETE_INTERNAL_PU 280
DELETE_LOCAL_LU 281
DELETE_LS 283
DELETE_LS_ROUTING 284
DELETE_LU62_TIMEOUT 286
DELETE_LU_0_TO_3 288
DELETE_LU_0_TO_3_RANGE 289
DELETE_LU_LU_PASSWORD 291
DELETE_LU_POOL 293
DELETE_MODE 294
DELETE_PARTNER_LU 296
DELETE_PORT 297
DELETE_RCF_ACCESS 298
DELETE_TN3270_ACCESS 301
DELETE_TN3270_ASSOCIATION 303
DELETE_TN_REDIRECT 305
DELETE_TP 307
DELETE_TP_LOAD_INFO 308
DELETE_USERID_PASSWORD 309
DIRECTORY_INDICATION 778
DISCONNECT_NODE 311
DLC
 開始 756
 照会 391
 定義 100
 停止 765
DLC_INDICATION 782
DLUR
 サポート 46
 デフォルト DLUS 108
 LU 404
 PU 409
DLUR_LU_INDICATION 783
DLUR_PU_INDICATION 784

DLUS 416
DLUS_INDICATION 786
DOWNSTREAM_LU_INDICATION 788
DOWNSTREAM_PU_INDICATION 792
DSPU テンプレート 437

F

FNA 150
FOCAL_POINT_INDICATION 794

H

HNA 150

I

INITIALIZE_SESSION_LIMIT 314
INIT_NODE 312
ISR セッション 449
ISR_INDICATION 796

L

LAN 上の複数のサーバー 5
LEN ノード 45
Linux アプリケーション
コンパイルとリンク 34
Linux アプリケーションのコンパイル 34
Linux アプリケーションのリンク 34
LOCAL_LU_INDICATION 800
LOCAL_TOPOLOGY_INDICATION 803
LS
開始 760
照会 478
定義 135
停止 769
統計 664
LS_INDICATION 805
LU タイプ 0-3 169, 173
LU タイプ 6.2 タイムアウト
削除 286
照会 532
定義 167
LU プール
照会 528
定義 181
LU-LU パスワード 179, 524
LU_0_TO_3_INDICATION 809

M

MAC アドレス、トークンリング/イーサネット 163

Management Services
活動状態であるアプリケーション 536
デフォルト PU 373
統計 539
フォーカル・ポイント 441
MDS アプリケーション 536
MDS サポート 46
MDS 統計 539
MODE_INDICATION 813

N

NN トポロジー・ノード指示 814
NN_TOPOLOGY_TG_INDICATION 815
NOF API の概要 1
NOF verb
概要 49
共通戻りコード 851
ノード構成に基づいた制約事項 45
発行する順序 44
nof エントリー・ポイント
指定パラメーター 29
説明 29
戻り値 30
AIX または Linux 28
Windows 35, 36
nofvcb 構造体 30, 31, 33, 40
Windows 37, 38
nof_async エントリー・ポイント
コールバック・ルーチン 33
コールバック・ルーチン、Windows 40
指定パラメーター 31
指定パラメーター、Windows 38
説明 30
戻り値 32
戻り値、Windows 39
AIX または Linux 28
Windows 36, 37
NOF_STATUS_INDICATION 24, 817

O

OPEN_FILE 318

P

PATH_SWITCH 321
PLU_INDICATION 818
PORT_INDICATION 820
PU 625
PU_INDICATION 821

Q

QUERY_ACTIVE_TRANSACTION 323
QUERY_ADJACENT_NN 328
QUERY_AVAILABLE_TP 331
QUERY_CENTRAL_LOGGER 337
QUERY_CENTRAL_LOGGING 338
QUERY_CN 340
QUERY_CN_PORT 344
QUERY_CONVERSATION 348
QUERY_COS 352
QUERY_COS_NODE_ROW 355
QUERY_COS_TG_ROW 360
QUERY_CPIC_SIDE_INFO 365
QUERY_CS_TRACE 370
QUERY_DEFAULTS 374
QUERY_DEFAULT_PU 373
QUERY_DIRECTORY_ENTRY 376
QUERY_DIRECTORY_LU 384
QUERY_DIRECTORY_STATS 389
QUERY_DLC 391
QUERY_DLC_TRACE 396
QUERY_DLUR_DEFAULTS 402
QUERY_DLUR_LU 404
QUERY_DLUR_PU 409
QUERY_DLUS 416
QUERY_DOMAIN_CONFIG_FILE 421
QUERY_DOWNSTREAM_LU 422
QUERY_DOWNSTREAM_PU 432
QUERY_DSPU_TEMPLATE 437
QUERY_FOCAL_POINT 441
QUERY_GLOBAL_LOG_TYPE 446
QUERY_ISR_SESSION 449
QUERY_KERNEL_MEMORY_LIMIT 457
QUERY_LOCAL_LU 459
QUERY_LOCAL_TOPOLOGY 468
QUERY_LOG_FILE 474
QUERY_LOG_TYPE 476
QUERY_LS 478
QUERY_LS_ROUTING 505
QUERY_LU62_TIMEOUT 532
QUERY_LU_0_TO_3 508
QUERY_LU_LU_PASSWORD 524
QUERY_LU_POOL 528
QUERY_MDS_APPLICATION 536
QUERY_MDS_STATISTICS 539
QUERY_MODE 542
QUERY_MODE_DEFINITION 549
QUERY_NN_TOPOLOGY_NODE 561
QUERY_NN_TOPOLOGY_STATS 567
QUERY_NN_TOPOLOGY_TG 571
QUERY_NODE 580
QUERY_NODE_ALL 595
QUERY_NODE_LIMITS 598
QUERY_PARTNER_LU 602
QUERY_PORT 615

QUERY_PU 625
QUERY_RAPI_CLIENTS 631
QUERY_RCF_ACCESS 635
QUERY_RTP_CONNECTION 637
QUERY_RTP_TUNING 645
QUERY_SECURITY_ACCESS_LIST 647
QUERY_SESSION 651
QUERY_SNA_NET 660
QUERY_STATISTICS 664
QUERY_TN3270_ACCESS_DEF 681
QUERY_TN3270_ASSOCIATION 687
QUERY_TN3270_DEFAULTS 690
QUERY_TN3270_EXPRESS_LOGON 692
QUERY_TN3270_SSL_LDAP 693
QUERY_TN_REDIRECT_DEF 695
QUERY_TN_SERVER_TRACE 698
QUERY_TP 699
QUERY_TP_DEFINITION 703
QUERY_TP_LOAD_INFO 708
QUERY_TRACE_FILE 712
QUERY_TRACE_TYPE 714
QUERY_USERID_PASSWORD 716
QUERY_* verb
 詳細情報 48
 複数のリソースに関する情報を戻す 47
 要約情報 48
 リスト・オプション 47

R

RAPI_CLIENT_INDICATION 824
RCF
 アクセス 635
 アクセスの防止 298
 定義 223
REGISTER_INDICATION_SINK 719
REGISTRATION_FAILURE 826
Remote API Client
 照会 631
REMOVE_DLC_TRACE 721
RESET_SESSION_LIMIT 725
RTP 接続
 照会 637
 バスの切り替え 321
 パラメーター 225, 645
RTP_INDICATION 828

S

SERVER_INDICATION 832
SESSION_INDICATION 834
SET_BUFFER_AVAILABILITY 730
SET_CENTRAL_LOGGING 731
SET_CS_TRACE 732
SET_GLOBAL_LOG_TYPE 735

SET_KERNEL_MEMORY_LIMIT 737
SET_LOG_FILE 738
SET_LOG_TYPE 742
SET_PROCESSING_MODE 43, 744
SET_TN_SERVER_TRACE 747
SET_TRACE_FILE 748
SET_TRACE_TYPE 751
SNA ゲートウェイ・サポート 46
SNA ネットワーク・ファイル指示 24
sna.net ファイル
 オープン 318
 クローズ 69
 バックアップ・サーバーの追加 53
 バックアップ・サーバーの削除 263
 バックアップ・サーバーの照会 660
sna.net ファイルのクローズ 69
SNA_NET_INDICATION 24, 838
SPCF
 アクセス 635
 定義 223
START_DLC 756
START_INTERNAL_PU 757
START_LS 760
START_PORT 763
STOP_DLC 765
STOP_INTERNAL_PU 767
STOP_LS 769
STOP_PORT 771
STREAMS コンポーネント 4
STREAMS バッファー 334, 730

T

Telnet クライアント
 許可検査 243
 高速ログオン 241
 TN リダイレクターの使用 245, 695
TERM_NODE 773
thesna.net ファイルのオープン 318
TN リダイレクト指示 839
TN3270 高速ログオン 241
TN3270 ユーザー
 TN3270 サーバーの使用 230, 681
TP 252, 699, 703, 708

U

UCF
 アクセス 635
 定義 223
UNREGISTER_INDICATION_SINK 774

V

VCB 構造体、ポインター、Windows 40
VCB 構造体へのポインター 30, 31, 33
 Windows 37, 38



プログラム番号:

Printed in Japan

SC88-8591-01



日本アイ・ビー・エム株式会社
〒106-8711 東京都港区六本木3-2-12