

**IBM Communications Server for Data Center
Deployment on AIX or Linux**



管理サービス プログラマーズ・ガイド

バージョン 7.0

**IBM Communications Server for Data Center
Deployment on AIX or Linux**



管理サービス プログラマーズ・ガイド

バージョン 7.0

ご注意

本書および本書で紹介する製品をご使用になる前に、57 ページの『付録 B. 特記事項』に記載されている情報をお読みください。

本書は、IBM Communications Server for Data Center Deployment on AIX or Linux バージョン 7.0 (プログラム番号 5725-H32) に適用されるとともに、新しい版またはテクニカル・ニュースレターで特に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も適用されます。

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

原典： SC23-8596-01
IBM Communications Server for Data Center Deployment
on AIX or Linux
Management Services Programmer's Guide
Version 7.0

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2012.12

© Copyright IBM Corporation 1998, 2012.

目次

表	v
本書について	vii
本書の対象読者	viii
本書の使用方法	viii
本書の構成	viii
表記上の規則	viii
このリリースの新機能	ix
詳細について	ix
第 1 章 管理サービスの概要	1
SNA 管理サービス・サポート・レベル	1
Communications Server 管理サービス・サポート	2
管理サービス・アプリケーション・プログラミング・ インターフェース	2
管理サービス・アプリケーション	2
データの送信のみを行う MS アプリケーション	2
データの送受信を行う MS アプリケーション	3
NMVT ルーティング	5
第 2 章 MS アプリケーションの作成	7
MS API エントリー・ポイントの説明	7
同期エントリー・ポイント: ms	8
非同期エントリー・ポイント: ms_async	9
ms_async エントリー・ポイントに指定されたコー ルバック・ルーチン	11
ターゲット・ハンドルの有効範囲	13
MS API ヘッダー・ファイル	14
MS アプリケーションのコンパイルとリンク	14
AIX アプリケーション	14
Linux アプリケーション	14
第 3 章 管理サービス verb	17
CONNECT_MS_NODE	18
VCB 構造体	18
指定パラメーター	18
戻りパラメーター	19
DISCONNECT_MS_NODE	21
VCB 構造体	21
指定パラメーター	21
戻りパラメーター	21
REGISTER_MS_APPLICATION	23
VCB 構造体	23
指定パラメーター	23
戻りパラメーター	24
REGISTER_NMVT_APPLICATION	26
VCB 構造体	27
指定パラメーター	27

戻りパラメーター	28
SEND_MDS_MU	31
VCB 構造体	31
指定パラメーター	31
戻りパラメーター	33
TRANSFER_MS_DATA	35
VCB 構造体	36
指定パラメーター	36
戻りパラメーター	39
UNREGISTER_MS_APPLICATION	40
VCB 構造体	41
指定パラメーター	41
戻りパラメーター	41
UNREGISTER_NMVT_APPLICATION	43
VCB 構造体	43
指定パラメーター	44
戻りパラメーター	44
第 4 章 管理サービス指示	47
FP_NOTIFICATION	47
VCB 構造体	48
パラメーター	48
MDS_MU_RECEIVED	49
VCB 構造体	49
パラメーター	50
MS_STATUS	51
VCB 構造体	51
パラメーター	51
NMVT_RECEIVED	52
VCB 構造体	52
パラメーター	52

付録 A. MS 関数セット	55
基本関数セット	55
オプションの関数セット	55
サポートしていない関数セット	55

付録 B. 特記事項	57
商標	59

参考文献	61
IBM Communications Server for AIX 関連資料	61
IBM Communications Server for Linux 関連資料	62
システム・ネットワーク体系 (SNA) 関連資料	64
APPC 関連資料	64
プログラミング関連資料	65

索引	67
-----------	-----------

表

1. 表記上の規則	viii
---------------------	------

本書について

本書は、IBM® Communications Server for Data Center Deployment on AIX® or Linux MS アプリケーション・プログラミング・インターフェース (API) を使用する管理サービス (MS) アプリケーションを作成するためのガイドです。

本書は、IBM Communications Server for Data Center Deployment (Communications Server) (プログラム製品番号 5725-H32) に適用されます。これは、AIX が稼働しているサーバーまたは Linux が稼働しているコンピューターが SNA ネットワーク上の他のノードと情報交換することを可能にする IBM ソフトウェア製品です。

IBM Communications Server for Data Center Deployment には、その動作環境となるハードウェアに応じて次の 3 種類のインストール形態があります。

IBM Communications Server for Data Center Deployment on AIX (CS/AIX)

IBM Communications Server for Data Center Deployment on AIX は、AIX バージョン 6.1 または 7.1 の基本オペレーティング・システムが稼働しているサーバー上で動作します。

IBM Communications Server for Data Center Deployment on Linux (CS Linux)

IBM Communications Server for Data Center Deployment on Linux は、以下の環境で動作します。

- 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

IBM Communications Server for Data Center Deployment on Linux for System z® (CS Linux for System z)

IBM Communications Server for Data Center Deployment on Linux for System z は、Linux for System z が稼働している System z メインフレーム上で動作します (s390x)。

本書では、相違が明示的に記述されていない限り、Communications Server という名称はこれら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 コンピューター」という用語は、Communications Server が稼働しているすべての種類のコンピューターを示す場合に使用されます。

MS API は、サーバー上または AIX/Linux クライアント上で実行されているアプリケーションで使用できます。Windows クライアントで稼働するアプリケーションでは使用できません。

本書には、MS API を使用してリモート・ネットワーク管理アプリケーションと通信する C 言語アプリケーション・プログラムの開発に必要な情報が記載されています。また、本書では、MS の概念を簡潔に説明し、熟練した MS プログラマー向けの詳細な参照情報を提供します。

本書は Communications Server のバージョン 7.0 に適用されます。

本書の対象読者

本書は、Communications Server がインストールされたシステム用の管理サービス・アプリケーションを作成する、熟練した C プログラマーを対象にしています。ただし、SNA や Communications Server の通信機能に関する実務経験は必ずしも必要ではありません。

アプリケーション・プログラマーは、Communications Server プログラミング・インターフェースを使用して SNA ネットワークでデータを送受信するトランザクション・プログラムとアプリケーション・プログラムを設計し、コーディングします。したがって、アプリケーション・プログラマーは、SNA、トランザクション・プログラムまたはアプリケーション・プログラムの通信相手のリモート・プログラム、および AIX / Linux オペレーティング・システムのプログラミング環境と操作環境に関して十分理解している必要があります。

Communications Server の資料の詳細については、『参考文献』を参照してください。

本書の使用法

この節では、本書の情報の構成と表記法について説明します。

本書の構成

本書は、次のように構成されています。

- 1 ページの『第 1 章 管理サービスの概要』では、Communications Server MS サポートについて説明しています。この章では、さまざまなレベルの SNA ネットワーク管理サポート、Communications Server MS API でサポートされる機能セットとオプションのサブセット、および Communications Server MS verb で提供される機能を説明しています。
- 7 ページの『第 2 章 MS アプリケーションの作成』では、MS アプリケーションの作成、コンパイル、およびリンクに関する情報を記載しています。
- 17 ページの『第 3 章 管理サービス verb』では、パラメーターと戻りコードを含む各 MS verb を詳しく説明しています。
- 47 ページの『第 4 章 管理サービス指示』では、パラメーターと戻りコードを含む Communications Server からアプリケーションに送信される各指示を詳しく説明しています。
- 55 ページの『付録 A. MS 関数セット』では、Communications Server MS API がサポートする SNA MS オプション・セットを掲載しています。

表記上の規則

表 1 に、本書で使用されている書体を示します。

表 1. 表記上の規則

内容	表記例
資料名	<i>IBM Communications Server for Data Center Deployment on AIX or Linux APPC プログラマーズ・ガイド</i>

表 1. 表記上の規則 (続き)

内容	表記例
ファイル名またはパス名	ms_c.h
コマンドまたは AIX / Linux ユーティリティ	cc
オプションまたはフラグ	-L
パラメーター	<i>opcode</i>
ユーザーが入力できるリテラル値または選択項目 (デフォルト値を含む)	0 (ゼロ)
定数	AP_CONNECT_MS_NODE
戻り値	AP_STATE_CHECK
指定値を表す変数	<i>nnnn</i>
環境変数	LD_RUN_PATH
プログラミング verb	CONNECT_MS_NODE
ユーザー入力	cc -L /usr/lib/sna -lms -lsna
関数、コールまたはエントリー・ポイント	ms_async
データ構造	MS_CALLBACK
16 進値	0x20

このリリースの新機能

Communications Server for Data Center Deployment バージョン 7.0 は、現在もサポート対象となっている Distributed Communications Server バージョン 6.4 の後継製品です。

詳細について

Communications Server ライブラリーのその他の資料、および SNA ワークステーションと AIX / Linux ワークステーションに関連する詳しい情報を記載した資料は、『参考文献』に収録されています。

第 1 章 管理サービスの概要

この章では、Communications Server 管理サービス (MS) アプリケーション・プログラミング・インターフェース (API) の概要を説明します。この章には、SNA でのさまざまなタイプの MS サポートに関する情報、および IBM Communications Server for Data Center Deployment on AIX or Linux を介したそれらの MS サポート機能へのアクセスに関する情報が記載されています。

SNA 管理サービス・サポート・レベル

SNA では、次の MS サポート・レベルを定義しています。各レベルは、MS サポート機能をインプリメントしている製品の世代ごとに異なります。

NMVT レベル

NMVT レベルの製品は、NMVT レベルの製品をサポートするノードの物理装置 (PU) とホスト側のシステム・サービス制御点 (SSCP) 間のセッションを介して、ネットワーク管理ベクトル・トランスポート (NMVT) をホスト・フォーカル・ポイントとの間で送受信することによって、管理情報を転送します。このセッションは、PU-SSCP セッションと呼ばれます。

NetView® バージョン 2 リリース 1 以前のバージョンでは、NMVT レベルをサポートしています。

マイグレーション・レベル

マイグレーション・レベルの製品は、独立論理装置 (LU) 6.2 間の LU-LU セッションを介して、CP_MSU (制御点管理サービス単位 GDS 変数) を送受信することによって、管理情報を転送します。CP_MSU は、MS 主ベクトルを含む単純 GDS 変数です。マイグレーション・レベルのフォーカル・ポイントは、アラートを受信できますが、その他の MS カテゴリはサポートされていません。

OS/400® は、マイグレーション・レベル製品の 1 例です。

MDS レベル

MDS (マルチドメイン・サポート) レベルの製品は、LU タイプ 6.2 セッションを介して、MDS_MU (MDS メッセージ単位 GDS 変数) を送受信することによって、管理情報を転送します。MDS_MU は、詳細な MS ルーティングおよび関連情報と共にヘッダーと、これに続く MS 主ベクトルを含む CP_MSU で構成されています。MDS レベルの製品は、同時に複数のフォーカル・ポイントと通信できます。ただし、問題管理などの特定の MS カテゴリには 1 つのフォーカル・ポイントのみを使用します。

OS/2 Communications Server/2 および NetView バージョン 2 リリース 2 (CP ではなく、サブエリア LU として) は、MDS レベルをサポートします。

Communications Server 管理サービス・サポート

Communications Server MS API を使用することにより、アプリケーションは、SNA ネットワーク上の他の MS 製品またはアプリケーションと通信できます。

Communications Server は、NMVT レベルと MDS レベルのアプリケーションをサポートできます。パートナー MS アプリケーションは、1 ページの『SNA 管理サービス・サポート・レベル』で説明されているすべてのレベルを実行できます。

Communications Server は、必要なあらゆるデータ変換をすべて実行します。

管理サービス・アプリケーション・プログラミング・インターフェース

Communications Server MS API は、次のエレメントからなります。

MS verb

verb は、次の項目を実行するために MS アプリケーションにより発行されます。

- MS データおよび状況指示の受信をサポートするため、MS アプリケーションがいつ Communications Server リソースを必要とするかを Communications Server に通知する。
- ネットワーク内の他の場所の MS アプリケーションに MS データを送信する (NMVT 形式か MDS_MU 形式のいずれか)。
- MS アプリケーションを Communications Server に登録して、フォーカル・ポイントから着呼 MS データを受信する (NMVT 形式か MDS_MU 形式のいずれか)。
- MS アプリケーションを Communications Server に登録して、特定の MS カテゴリーを担当するフォーカル・ポイントについての情報を受け取る。この登録により、Communications Server は MDS_MU データを適切なアプリケーションに経路指定できるようになります。

MS verb の詳細については、17 ページの『第 3 章 管理サービス verb』を参照してください。

MS 指示

指示は、Communications Server によってローカルに生成されるか、ネットワークから受信したデータを転送するために使用されます。MS 指示の詳細については、47 ページの『第 4 章 管理サービス指示』を参照してください。

管理サービス・アプリケーション

MS アプリケーションを作成するときに使用する verb とエン트리・ポイントは、MS アプリケーションが次のどちらかを実行するかによって異なります。

- データの送信のみ
- データの送受信

データの送信のみを行う MS アプリケーション

この最も簡単なタイプのアプリケーションは、データを送信するのみで、Communications Server ノードからデータを受信しません。このタイプのアプリケー

ションは、同期エントリー・ポイントか非同期エントリー・ポイントのいずれかを
使用でき、データ送信用に次の verb のいずれか、または両方を使用する必要があります。

- SEND_MDS_MU verb は、MDS_MU 形式でデータを送信し、Communications Server がこのデータをリモート MS アプリケーションに送信します。
- TRANSFER_MS_DATA verb は、NMVT 形式でデータを送信し、Communications Server はこのデータをリモート MS アプリケーションに送信します。このデータは、完全な NMVT またはサブベクトルであり、Communications Server はこのデータに必須 NMVT ヘッダー情報を追加します。

同期および非同期エントリー・ポイントの詳細については、7 ページの『MS API エントリー・ポイントの説明』を参照してください。

データの送受信を行う MS アプリケーション

このタイプのアプリケーションは、データの送信、および Communications Server ノードからのデータと状況指示の受信の両方を実行します。このようなタイプのアプリケーションを作成するためには、次の verb を組み込むことが必要です (特に断りのない限り、同期エントリー・ポイントか非同期エントリー・ポイントのいずれかを使用できます)。

1. CONNECT_MS_NODE verb を発行することにより Communications Server ノードとの通信を確立する。これにより、アプリケーションはデータ、フォーカル・ポイント指示、またはその両方を受信するように登録することができるようになります。
2. アプリケーションが受信したいデータのタイプを指示するには、Communications Server ノードに登録します。次の verb のいずれか、または両方を使用して Communications Server に登録するために、非同期エントリー・ポイントを使用する必要があります。
 - REGISTER_MS_APPLICATION verb は、MDS_MU を受け付けることができる MDS レベルのアプリケーションとして、アプリケーションを Communications Server に登録します。この verb には、アプリケーションが、特定の MS カテゴリのフォーカル・ポイントに関する情報を要求できるようにするためのオプションがあります。Communications Server は、MDS_MU_RECEIVED 指示または FP_NOTIFICATION 指示、あるいはその両方を使用して、要求データをアプリケーションに渡します。
 - REGISTER_NMVT_APPLICATION verb は、次のいずれかの方法でアプリケーションを Communications Server に登録します。
 - 特定の MS 主ベクトル・キーをもつ NMVT を受け付ける NMVT レベルのアプリケーションとして。Communications Server は、NMVT_RECEIVED 指示を使用して NMVT をアプリケーションに渡します。
 - NMVT が MDS_MU に変換されたあとで、特定の MS 主ベクトル・キーをもつ NMVT を受け付ける MDS レベルのアプリケーションとして。Communications Server は、受信した NMVT を MDS_MU に変換し、MDS_MU_RECEIVED 指示を使用して MDS_MU をアプリケーションに渡します。この使用方法では、MDS レベルのアプリケーションは、NMVT レベルのデータ形式を認識していなくても、NMVT レベルのデータおよび状況指示を受信できます。

アプリケーションは、Communications Server に登録するときにコールバック・ルーチンのアドレスを指定します。Communications Server は、要求したタイプのデータをノードで着信すると、このコールバック・ルーチン呼び出します。Communications Server がコールバック・ルーチンに指定するデータ構造の詳細については、47 ページの『第 4 章 管理サービス指示』を参照してください。

3. 登録されたアプリケーションは、次の処理を実行できます。
 - 次の verb のいずれか、または両方を使用して、Communications Server ノードにデータを送信します。
 - SEND_MDS_MU verb は、MDS_MU 形式でデータを指定し、Communications Server はこのデータをリモート MS アプリケーションに送信します。
 - TRANSFER_MS_DATA verb は、NMVT 形式でデータを指定し、Communications Server はこのデータをリモート MS アプリケーションに送信します。このデータは、完全な NMVT またはサブベクトルであり、Communications Server はこのデータに必須 NMVT ヘッダー情報を追加します。
 - Communications Server が次の状況指示を戻すと、Communications Server ノードから状況情報を受信します。
 - FP_NOTIFICATION 指示は、特定の MS カテゴリのフォーカル・ポイントに関する情報を提供します。Communications Server は、フォーカル・ポイント情報を受信するために登録した MDS レベルのアプリケーションにこの指示を戻します。
 - MS_STATUS 指示は、Communications Server システム状況の変更内容をアプリケーションに通知します (アプリケーションに接続しているノードへのアプリケーションの通信パスが消失したとき、または Communications Server ソフトウェアが停止したとき)。Communications Server は、MDS レベルのアプリケーションと NMVT レベルのアプリケーションの両方にこの指示を戻します。
 - Communications Server が次の受信データ指示を戻すと、Communications Server ノードからデータを受信します。
 - MDS_MU_RECEIVED データ指示は、MDS_MU を MDS レベルのアプリケーションに戻します。戻された MDS_MU は次のいずれかです。
 - MS アプリケーションが REGISTER_MS_APPLICATION verb を使用して登録した場合は、リモート・アプリケーションが送信した MDS_MU。
 - MS アプリケーションが REGISTER_NMVT_APPLICATION verb を使用して登録した場合は、着呼 NMVT から変換された MDS_MU。
 - NMVT_RECEIVED データ指示は、NMVT を受信するために登録した NMVT レベルのアプリケーションに NMVT を戻します。
4. アプリケーションが完了するとき、次の verb のいずれかを発行して、Communications Server 登録を終了する必要があります。
 - UNREGISTER_MS_APPLICATION verb は、アプリケーションの Communications Server 登録を終了します。アプリケーションがこのコールを発行すれば、Communications Server は、MDS_MU をそのアプリケーションに送信しません。

- UNREGISTER_NMVT_APPLICATION verb は、特定の MS 主ベクトル・キーをもつ NMVT をアプリケーションが受け付けないようにアプリケーションの Communications Server 登録を終了します。
5. Communications Server への登録を終了したアプリケーションは、DISCONNECT_MS_NODE verb を発行して Communications Server ノードとの通信を終了し、そのアプリケーションに関連したリソースを解放する必要があります。

同期および非同期エントリー・ポイントの詳細については、7 ページの『MS API エントリー・ポイントの説明』を参照してください。

NMVT ルーティング

Communications Server は、NMVT をリモート・ノードから受信するとき、NMVT の MS 主ベクトル・キーと宛先アプリケーション名サブフィールドを使用して、NMVT の送信先となる MS アプリケーションの優先順位を次のように判断します。

1. Communications Server は、NMVT の宛先名と一致するアプリケーション名で登録した NMVT レベルのアプリケーションを、次の優先順位で見つけ出します。
 - a. 着呼 NMVT で転送された特定の主ベクトル・キーを受け付けるように登録したアプリケーション。
 - b. 主ベクトル・キーが 0x8061~0x8064 の範囲内である場合に、SNA サービス・ポイント・コマンド機能 (SPCF) キーを受け付けるように登録したアプリケーション。
 - c. すべてのキーを受け付けるように登録したアプリケーション。
2. Communications Server は、NMVT レベルの適切なアプリケーションを見つけない場合は、NMVT の宛先名と一致するアプリケーション名で登録し、MDS_MU への変換後に NMVT を受け付けるように登録した MDS レベルのアプリケーションを見つげ出そうとします。適切な主ベクトル・キーを受け付けることができるアプリケーションを選択する場合の優先順位は、NMVT レベルのアプリケーションの場合と同じです。

第 2 章 MS アプリケーションの作成

この章では、MS アプリケーションに関して次の事項を説明します。

- MS API エントリー・ポイントの使用方法
- 非同期イベントのスケジュール方法
- MS API を使用するためのコンパイルおよびリンク方法

MS API エントリー・ポイントの説明

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、MS API にアクセスします。

ms アプリケーションは、MS verb を同期で発行するために、このエントリー・ポイントを使用します。Communications Server は、verb 処理が終了するまで制御をアプリケーションに戻しません。REGISTER_MS_APPLICATION と REGISTER_NMVT_APPLICATION 以外の MS verb は、このエントリー・ポイントを介して発行できます。

次の条件が両方とも該当する場合、アプリケーションはこのエントリー・ポイントのみを使用できます。

- アプリケーションは、TRANSFER_MS_DATA verb または SEND_MDS_MU verb、あるいはその両方を使用して MS データの送信のみを必要とする場合 (アプリケーションは、MS データまたは状況指示を受信する必要がない)。
- Communications Server が verb を完全に処理するまで待っている間、アプリケーションは中断できる。

ms エントリー・ポイントは、MS ヘッダー・ファイル `/usr/include/sna/ms_c.h` (AIX) または `/opt/ibm/sna/include/ms_c.h` (Linux) に定義されています。

ms_async

アプリケーションは、MS verb を非同期で発行するために、このエントリー・ポイントを使用します。Communications Server は、verb がまだ処理中か、完了したかを示す戻り値と共に、アプリケーションに制御をすぐに戻します。verb がまだ処理中であることを戻り値が示した場合、Communications Server は、アプリケーションが指定するコールバック・ルーチンを使用して、verb 処理の結果を戻します。verb 処理が完了したことを戻り値が示した場合、コールバック・ルーチンは呼び出されません。

すべての MS verb は、このエントリー・ポイントを介して発行することができます。REGISTER_MS_APPLICATION verb と REGISTER_NMVT_APPLICATION verb は、このエントリー・ポイントを介して発行する必要があります。

次の条件のいずれかが該当する場合には、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。

- アプリケーションは、MS データおよび状況指示を受信する必要がある。

MS API エントリー・ポイントの説明

- Communications Server が verb の処理の終了を待つ間は、アプリケーションは中断できない。

ms_async エントリー・ポイントは、MS ヘッダー・ファイル `/usr/include/sna/ms_c.h` (AIX) または `/opt/ibm/sna/include/ms_c.h` (Linux) に定義されています。

ms_async 用コールバック・ルーチン

非同期 MS API エントリー・ポイントを使用するとき、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指定する必要があります。Communications Server では、このコールバック・ルーチンを、verb の実行を完了させるために使用するのみでなく、MS データおよび状況指示を戻すためにも使用します。

同期エントリー・ポイント: ms

アプリケーションは、MS verb を同期で発行するために、ms を使用します。Communications Server は、verb 処理が終了するまで制御をアプリケーションに戻しません。

関数コール

```
void ms (
    AP_UINT32 target_handle,
    void * msvcb
);
```

指定パラメーター

アプリケーションは、ms エントリー・ポイントを使用するとき、次のパラメーターを指定します。

target_handle

UNREGISTER_MS_APPLICATION verb、UNREGISTER_NMVT_APPLICATION verb、および DISCONNECT_MS_NODE verb の場合、アプリケーションは CONNECT_MS_NODE verb で戻された値を指定します。このパラメーターは、ターゲット Communications Server ノードを識別するために使用されません。

その他のすべての verb では、このパラメーターを使用しません。0 (ゼロ) に設定してください。

msvcb 発行されている verb のためのパラメーターを含む verb 制御ブロック (VCB) へのポインター。個々の verb の VCB 構造体については、17 ページの『第 3 章 管理サービス verb』で説明しています。これらの構造体は、MS API ヘッダー・ファイル `/usr/include/sna/ms_c.h` (AIX) または `/opt/ibm/sna/include/ms_c.h` (Linux) に定義されています。

注: MS VCB には、「予約済み」とマークされたパラメーターが多数含まれています。予約済みパラメーターには、Communications Server ソフトウェアで内部的に使用されているものや、このバージョンでは使用されていなくても将来のバージョンで使用される可能性があるものがあります。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決してアクセスしないでください。verb によって使用される他のパラメーター

をアプリケーションが設定する前に、VCB の内容全体をゼロに設定して、これらのパラメーターすべてを確実にゼロに設定しておく必要があります。このようにすると、Communications Server がその内部使用パラメーターを誤って解釈することはありません。またこれにより、今後の Communications Server のバージョンで、これらのパラメーターを使って新しい機能を引き続き使用することができるようになります。

VCB の内容をゼロに設定するには、次のように `memset` を使用します。

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

戻り値

ms エントリー・ポイントには、戻り値はありません。このコールが戻ったら、アプリケーションは、VCB 内の戻りコードを調べることによって、該当の verb が正常に完了したかどうかを確認し、後続の verb にどのパラメーターが必要かを判断する必要があります。後続の verb を発行するときにアプリケーション側で必要になるパラメーターの例としては、CONNECT_MS_NODE verb が正常に終了した場合に VCB に設定される `target_handle` があります。

同期エントリー・ポイントの使用

1 つのターゲット・ハンドルで未解決にできる同期 verb は常に 1 つのみです。同じターゲット・ハンドルで別の同期 verb が処理中である場合、同期 verb は、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: ms_async

アプリケーションは、MS verb を非同期で発行するために、`ms_async` を使用します。また、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指定します。Communications Server は、verb がまだ処理中か、完了しているかを示す戻り値と共に、制御をすぐにアプリケーションに戻します。ほとんどの場合、制御がアプリケーションに戻っても、verb はまだ処理中です。その場合、Communications Server はアプリケーションが指定するコールバック・ルーチンを使用して、verb 処理の結果をあとで戻します。Communications Server がアプリケーションに制御を戻すときに verb 処理が完了している場合は、アプリケーションのコールバック・ルーチンを使用しません。

関数コール

```
unsigned short ms_async(
    AP_UINT32      target_handle,
    void *         msvcb,
    VMV_CALLBACK   comp_proc,
    AP_CORR        corr
);

typedef void (*VMV_CALLBACK) (
    AP_UINT32      target_handle,
    void *         msvcb,
    AP_CORR        corr
);

typedef union ap_corr {
```

```

void *      corr_p;
AP_UINT32  corr_l;
AP_INT32   corr_i;
} AP_CORR;

```

VMV_CALLBACK 構造体に含まれるパラメーターの詳細については、11 ページの『ms_async エントリー・ポイントに指定されたコールバック・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、ms_async エントリー・ポイントを使用するとき、次のパラメーターを指定します。

target_handle

ターゲット Communications Server ノードの ID。REGISTER_*、UNREGISTER_*、および DISCONNECT_MS_NODE verb の場合、アプリケーションは、CONNECT_MS_NODE verb で戻された値を指定します。

その他のすべての verb では、このパラメーターを使用しません。0 (ゼロ) に設定してください。

msvcb 発行されている verb のためのパラメーターを含む verb 制御ブロック (VCB) へのポインター。個々の verb の VCB 構造体については、17 ページの『第 3 章 管理サービス verb』で説明しています。これらの構造体は、MS API ヘッダー・ファイル `/usr/include/sna/ms_c.h` (AIX) または `/opt/ibm/sna/include/ms_c.h` (Linux) に定義されています。

注: MS VCB には、「予約済み」とマークされたパラメーターが多数含まれています。予約済みパラメーターには、Communications Server ソフトウェアで内部的に使用されているものや、このバージョンでは使用されていなくても将来のバージョンで使用される可能性があるものがあります。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決してアクセスしないでください。verb によって使用される他のパラメーターをアプリケーションが設定する前に、VCB の内容全体をゼロに設定して、これらのパラメーターすべてを確実にゼロに設定しておく必要があります。このようにすると、Communications Server がその内部使用パラメーターを誤って解釈することはありません。またこれにより、今後の Communications Server のバージョンで、これらのパラメーターを使って新しい機能を引き続き使用することができるようになります。

VCB の内容をゼロに設定するには、次のように `memset` を使用します。

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

comp_proc

verb が完了したときに Communications Server が呼び出すコールバック・ルーチン。コールバック・ルーチンの要件の詳細については、11 ページの『ms_async エントリー・ポイントに指定されたコールバック・ルーチン』を参照してください。

corr アプリケーションが使用するオプションの相関係数。このパラメーターは、

C 共用体として定義され、その結果、アプリケーションは、3 種類のパラメーター (ポインター、符号なし long、整数) のいずれも指定できます。

Communications Server はこの値を使用しません、`verb` が完了するとパラメーターとしてコールバック・ルーチンにこの値を渡します。この値を使用すると、アプリケーションは他の処理で戻された情報と相互に関連付けることができます。

戻り値

非同期エントリー・ポイントは、次の値のいずれかを戻します。

AP_COMPLETED

`verb` はすでに完了しています。アプリケーションは、VCB 内のパラメーターを調べて、`verb` が正常に完了したかどうかを確認できます。

Communications Server は、この `verb` では、指定されたコールバック・ルーチン呼び出しません。

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 MS `verb` の発行など、その他の処理を継続できますが、それらの処理が現在の `verb` の完了に依存していないことが前提です。ただし、アプリケーションは、この `verb` に指定された VCB 内のパラメーターを調べたり、変更したりできません。

Communications Server は、`verb` 処理の完了を指示するために、指定されたコールバック・ルーチン呼び出します。その後、アプリケーションは、VCB パラメーターを調べることができます。

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ときには、次の点に注意してください。

- アプリケーションがヌル・ポインターを `comp_proc` パラメーターに指定した場合、`verb` は同期で完了します (同期エントリー・ポイントを使用してアプリケーションが `verb` を発行したときと同様)。
- `ms_async` をアプリケーションのコールバック内から呼び出す場合、ヌル・ポインターを `comp_proc` パラメーターに指定することは許可されていません。これを指定すると、Communications Server は、1 次戻りコード値 `AP_PARAMETER_CHECK` と 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は VCB 内のどのパラメーターも使用したり、変更したりできません。
- 複数の `verb` が発行された順に完了するわけではありません。アプリケーションが非同期 `verb` を発行したあとに同期 `verb` を発行した場合には、その同期 `verb` が完了しても、非同期 `verb` がすでに完了しているとは限りません。

ms_async エントリー・ポイントに指定されたコールバック・ルーチン

非同期 MS API エントリー・ポイントを使用するとき、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指定する必要があります。Communications Server では、このコールバック・ルーチンを、`verb` の実行を完了させるために使用するのみでなく、MS データおよび状況指示を戻すためにも使用します。アプリケ

MS API エントリー・ポイントの説明

ーションは、VCB 内の *opcode* パラメーターを調べて、コールバック・ルーチンに含まれているイベントを判別する必要があります。

この節では、コールバック・ルーチン、およびコールバック・ルーチンが実行しなければならない関数を Communications Server がどのように使用するかについて説明します。

コールバック関数

```
typedef void (*VMV_CALLBACK) (  
    AP_UINT32    target_handle,  
    void *       msvcb,  
    AP_CORR     corr  
);  
  
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 は、次のパラメーターでコールバック・ルーチンを呼び出します。

target_handle

MS データおよび状況指示の場合、Communications Server は、REGISTER_MS_APPLICATION verb または REGISTER_NMVT_APPLICATION verb で指定されたターゲット・ハンドルを渡します。verb の完了では、このパラメーターは未定義です。

msvcb 次のいずれかです。

- MS データおよび状況指示の場合、Communications Server が指定する VCB へのポインター。
- verb の完了の場合、アプリケーションが指定する VCB へのポインター。その後、VCB は Communications Server によって設定された戻りパラメーターを組み込みます。

corr アプリケーションが指定する相関係数の値。この値を使用すると、アプリケーションは他の処理で戻された情報と相互に関連付けることができます。

コールバック・ルーチンは、これらのパラメーターをすべて使用する必要はありません（『指示のためのコールバック・ルーチンの使用』で説明されている例は除く）。コールバック・ルーチンは、戻りパラメーターで必要な処理をすべて実行できます。また、変数を設定するのみで、verb が完了したことを MS アプリケーションに通知します。

戻り値

コールバック関数は値を戻しません。

指示のためのコールバック・ルーチンの使用

REGISTER_MS_APPLICATION VCB で指定されたコールバック・ルーチンは、次の指示を受信できます。

- FP_NOTIFICATION (登録時にこの情報をアプリケーションが要求した場合)

- MDS_MU_RECEIVED
- MS_STATUS

REGISTER_NMVT_APPLICATION VCB で指定されたコールバック・ルーチンは、次の指示を受信できます。

- NMVT_RECEIVED (NMVT レベルのデータから MDS レベルのデータへの変換をアプリケーションが要求しなかった場合)
- MDS_MU_RECEIVED (NMVT レベルのデータから MDS レベルのデータへの変換をアプリケーションが要求した場合)
- MS_STATUS

アプリケーションは VCB を MS verb 用に割り振りますが、Communications Server は VCB を指示用に割り振ります。したがって、アプリケーションにはコールバック・ルーチン内からのみ、VCB 情報へのアクセス権があります。Communications Server がコールバック・ルーチンに提供する VCB ポインターは、コールバック・ルーチンの外側では無効となります。アプリケーションは、コールバック・ルーチン内から必要な処理すべてを完了させる必要があります。あるいは、アプリケーションがルーチン外部で使用する必要のあるすべての VCB データのコピーを作成する必要があります。

コールバック・ルーチン内の指示の処理は、次の追加要件を満たす必要があります。

- NMVT レベルのアプリケーションが REGISTER_NMVT_APPLICATION を使用して着呼 NMVT を受信する場合、そのアプリケーションは、512 バイト (最大 NMVT サイズ) のデータ長を受信できること。
- MDS レベルのアプリケーションが REGISTER_NMVT_APPLICATION を使用して、MDS_MU への変換後に着呼 NMVT を受信する場合、そのアプリケーションは、700 バイトのデータ長を受信できること。これは、MDS_MU ヘッダー情報を含めた最大 NMVT サイズを考慮した長さです (この要件は、MDS_MU を受信するために REGISTER_MS_APPLICATION を使用するアプリケーションには適用されません。アプリケーションは受信できる最大データ長を指定でき、Communications Server は、必要に応じてデータをセグメント化するからです)。
- MDS レベルのアプリケーションが REGISTER_MS_APPLICATION を使用して、着呼 MDS_MU を受信する場合、そのアプリケーションは、REGISTER_MS_APPLICATION verb の *max_rcv_size* パラメーターに指定した値のデータ長を受信できること。

ターゲット・ハンドルの有効範囲

MS を使用する各アプリケーションは、CONNECT_MS_NODE verb を使用して各アプリケーション固有のハンドルを取得する必要があります。2 つの MS アプリケーションが同じ MS 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_MS_NODE を発行したアプリケーションが、後で fork して子プロセスを作成した場合、その子プロセスは親プロセスによって取得されたターゲット・ハンドルを使用するなどの MS verb も発行できません。ただし、子プロセスは、別の CONNECT_MS_NODE を発行して子プロセス固有のターゲット・ハンドルを取得できます。

MS API ヘッダー・ファイル

MS アプリケーションで使用されるヘッダー・ファイルは **ms_c.h** です。このファイルには、MS API エントリー・ポイントと MS VCB の定義が含まれています。また、共通インターフェースのヘッダー・ファイル **values_c.h** も含まれています。この 2 つのファイルには、MS API で指定パラメーター値と戻りパラメーター値に定義されたすべての定数が含まれています。ファイル **values_c.h** には、MS VCB で使用される AP_UINT16 のようなパラメーター型の定義が含まれています。両方のファイルはディレクトリー **/usr/include/sna** (AIX) または **/opt/ibm/sna/include** (Linux) に保管されます。

MS 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・アプリケーションのコンパイルとリンクを行うには、次のオプションを使用します。

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

64 ビット・アプリケーションのコンパイルとリンクを行うには、次のオプションを使用します。

```
-bimport:/usr/lib/sna/ms_r64_5.exp -I  
/usr/include/sna
```

Linux アプリケーション

MS アプリケーションのコンパイルとリンクを行う前に、共用ライブラリーが保管されているディレクトリーを指定します。これにより、アプリケーションが実行時に共用ライブラリーを見つけることができます。64 ビット・アプリケーションをコンパイルする場合は、環境変数 **LD_RUN_PATH** を **/opt/ibm/sna/lib** または **/opt/ibm/sna/lib64** に変更してください。

32 ビット・アプリケーションのコンパイルとリンクを行うには、次のオプションを使用します。

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

64 ビット・アプリケーションのコンパイルとリンクを行うには、次のオプションを使用します。

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

オプション **-lpLiS** は、Communications Server 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

MS アプリケーションのコンパイルとリンク

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。

第 3 章 管理サービス verb

この章では、各 MS verb についての次の情報を提供します。

- verb の目的と使用方法。
- verb で使用される verb 制御ブロック (VCB) 構造体。すべての VCB 構造体は、ヘッダー・ファイル `/usr/include/sna/ms_c.h` (AIX) または `/opt/ibm/sna/include/ms_c.h` (Linux) に定義されています。
- 指定パラメーター (verb に指定された VCB フィールド)。パラメーターごとに次の情報を示しています。
 - 説明
 - 有効な値とその意味
 - 追加情報 (必要な場合)
- 戻りパラメーター。verb が完了すると、その verb には次の戻りパラメーターが含まれています。

primary_rc

このパラメーターは、verb が正常に完了したかどうかを示します。verb が正常に完了しなかった場合、このパラメーターは、その理由のカテゴリを示します。

secondary_rc

このパラメーターは、実行が失敗した理由を具体的に示します。

さらに、一部の verb には、追加戻りパラメーターがあります。

指定パラメーター値と戻りパラメーター値の多くは数値で表されます。コーディングを単純にするには、アプリケーションをさらにポータブルにしてからプログラムのソースを読みやすくしてください。これらの値はヘッダー・ファイル `ms_c.h` に定義されている記号定数で示されます。例えば、`SEND_MDS_MU` verb の *opcode* (命令コード) パラメーターは、記号定数 `AP_SEND_MDS_MU` で示される値です。

システムのタイプにより、これらの値をメモリーに保管する方法が異なるため、指定パラメーターに値を設定するとき、または戻りパラメーターの値をテストするときは、数値ではなく記号定数を使用することが重要です。ヘッダー・ファイルに示されている値は、使用しているシステムで認識される形式ではない場合があります。

注: MS VCB には、「予約済み」とマークされたパラメーターが多数含まれています。予約済みパラメーターには、Communications Server ソフトウェアで内部的に使用されているものや、このバージョンでは使用されていなくても将来のバージョンで使用される可能性があるものがあります。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決してアクセスしないでください。verb によって使用される他のパラメーターをアプリケーションが設定する前に、VCB の内容全体をゼロに設定して、これらのパラメーターすべてを確実にゼロに設定しておく必要があります。このようにすると、Communications Server がその内部使用パラメーターを誤って解釈することはありません。またこれにより、今

後の Communications Server のバージョンで、これらのパラメーターを使って新しい機能を引き続き使用することができるようになります。

VCB の内容をゼロに設定するには、次のように `memset` を使用します。

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

CONNECT_MS_NODE

この verb は、アプリケーションを Communications Server ノードに接続します。また、あとに続く MS エントリー・ポイントに対する後続のすべてのコールで使用されるハンドルを戻します。

TRANSFER_MS_DATA verb か SEND_MDS_MU verb のいずれかを使用してデータを送信するのみで、MS データまたは状況指示を受信する必要のないアプリケーションでは、この verb を発行したり、あとに続く MS エントリー・ポイントのコールにハンドルを指定する必要はありません。

VCB 構造体

```
typedef struct connect_ms_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[64];    /* Name of Node to connect to  */
    AP_UINT32         target_handle;    /* Handle to identify Node on  */
    /* subsequent verbs          */
} CONNECT_MS_NODE;
```

指定パラメーター

アプリケーションは、CONNECT_MS verb を発行するときには、次のパラメーターを指定します。

opcode AP_CONNECT_MS_NODE

node_name

接続先の Communications Server ノードの名前。これは ASCII 文字ストリングです。

アプリケーションが NMVT を受信して NMVT レベルのバージョンの NetView プログラムに対するサービス・ポイントとして機能するために登録しようとする場合、NetView ホスト (NMVT を NetView プログラムに伝送するために使用される PU-SSCP セッションのノード) に直接接続するノードの名前を指定してください。NMVT レベルのプログラムの詳細については、1 ページの『第 1 章 管理サービスの概要』を参照してください。

以下の条件のいずれかが成立する場合、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 (ノード名の指定は不要です)。

- Communications Server が単一の AIX / Linux コンピューター上で (LAN 上ではなく) すべてのコンポーネントを使用して実行中である。
- Communications Server LAN で使用されるサーバーが 1 つのみである。

- アプリケーションが MDS レベルで、データの送受信が NMVT フォーマットではなく、MDS_MU フォーマットで行われる。

Communications Server LAN に複数のサーバーがあり、このパラメーターがすべて 2 進ゼロに設定されていると、アプリケーションは、アプリケーションと同じサーバーにあるノード (有効である場合) または他の有効なノードに接続します。

戻りパラメーター

verb の実行後、Communications Server は、その実行が成功したかどうかを示すパラメーターを戻します。verb の実行が成功した場合には、Communications Server は、アプリケーションが後続の MS エントリー・ポイントで使用するターゲット・ハンドルも戻します。verb の実行が成功しなかった場合は、Communications Server は、その理由を示すパラメーターを戻します。

正常に実行された場合

verb が正常に実行された場合は、Communications Server は以下のパラメーターを戻します。

primary_rc
AP_OK

secondary_rc
使用されません。

target_handle
このノードに送信される後続の verb で使用するための戻り値。

正常に実行されなかった場合

verb の実行が失敗した場合は、Communications Server はエラーのタイプを示す 1 次戻りコード、および実行が失敗した理由を具体的に示す 2 次戻りコードを戻します。

パラメーター・チェック: パラメーター・エラーが原因で verb が実行されなかった場合、Communications Server 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 パラメーターは、いずれの Communications Server ノードの名前とも一致しませんでした。

状態チェック: 状態エラーのために verb が実行されない場合、Communications Server は次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_CONNECT_FAILED

指定したノードがアクティブではないため、またはヌル・ノード名を指定した場合にどのノードもアクティブでないために、ノードの接続中にエラーが発生しました。

AP_INVALID_TARGET_STATE

MS コールで使用されたターゲット・ハンドルが、0 (ゼロ) に設定されていませんでした。CONNECT_MS_NODE の場合、ターゲット・ハンドルを 0 (ゼロ) に設定する必要があります。

AP_SYNC_PENDING

アプリケーションは、同期エントリー・ポイントを使用してこの verb を発行しましたが、別の同期 verb がこのターゲット・ハンドルで処理中でした。特定のターゲット・ハンドルで、いつでも処理中にできる同期 verb は 1 つのみです。

AP_SYNC_NOT_ALLOWED

アプリケーションは、同期 MS エントリー・ポイントを使用してこの verb をコールバック・ルーチン内から発行しました。verb をコールバック・ルーチンから発行するには、アプリケーション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。

Communications Server ソフトウェアがアクティブでない場合: Communications Server ソフトウェアがアクティブでないために verb が実行されない場合は、Communications Server は、次のパラメーターを戻します。

primary_rc

次のいずれかです。

AP_COMM_SUBSYSTEM_NOT_LOADED

Communications Server ソフトウェアがまだ始動していないか、停止しています。

AP_COMM_SUBSYSTEM_ABENDED

Communications Server ソフトウェアに障害が起きました。

Communications Server ソフトウェアがアクティブでない場合は、Communications Server から *secondary_rc* は戻されません。

システム・エラー: システム・エラーのために verb が実行されない場合、Communications Server は次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

オペレーティング・システムのコールからの戻りコード。この戻りコードの意味については、ファイル `/usr/include/sys/errno.h` 内の戻り値を調べてください。

DISCONNECT_MS_NODE

この verb は、アプリケーションをノードから切断し、その接続に関連したすべてのリソースを解放します。アプリケーションが切断するノードは、コールでの *target_handle* パラメーターで識別されます。

VCB 構造体

```
typedef struct disconnect_ms_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_MS_NODE;
```

指定パラメーター

アプリケーションは、DISCONNECT_MS_NODE を発行するときに、次のパラメーターを指定します。

opcode AP_DISCONNECT_MS_NODE

戻りパラメーター

verb の実行後は、Communications Server は、その実行が成功したかどうかを示すパラメーターを戻し、成功しなかった場合は、その理由を示すパラメーターを戻します。

正常に実行された場合

verb が正常に実行された場合は、Communications Server は以下のパラメーターを戻します。

primary_rc
AP_OK

verb の実行が成功した場合は、Communications Server から *secondary_rc* は戻されません。

正常に実行されなかった場合

verb の実行が失敗した場合は、Communications Server はエラーのタイプを示す 1 次戻りコード、および実行が失敗した理由を具体的に示す 2 次戻りコードを戻します。

パラメーター・チェック: パラメーター・エラーが原因で verb が実行されなかった場合、Communications Server は、以下のパラメーターを戻します。

primary_rc
AP_PARAMETER_CHECK

secondary_rc

AP_INVALID_TARGET_HANDLE

指定したターゲット・ハンドルは、直前の CONNECT_MS_NODE verb で戻された有効な値ではありませんでした。

DISCONNECT_MS_NODE

状態チェック: 状態エラーのために verb が実行されない場合、Communications Server は次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_INVALID_TARGET_STATE

アプリケーションは、CONNECT_MS_NODE または直前の DISCONNECT_MS_NODE がまだ未解決のときに、DISCONNECT_MS_NODE を発行しました。

AP_SYNC_PENDING

アプリケーションは、同期エン트리・ポイントを使用してこの verb を発行しましたが、別の同期 verb がこのターゲット・ハンドルで処理中でした。特定のターゲット・ハンドルで、いつでも処理中にできる同期 verb は 1 つのみです。

AP_VERB_IN_PROGRESS

アプリケーションは、直前の非同期 MS verb がまだ未解決のときに、DISCONNECT_MS_NODE を発行しました。

AP_SYNC_NOT_ALLOWED

アプリケーションは、同期 MS エントリー・ポイントを使用してこの verb をコールバック・ルーチン内から発行しました。verb をコールバック・ルーチンから発行するには、アプリケーションは非同期エン트리・ポイントを使用する必要があります。

Communications Server ソフトウェアがアクティブでない場合: Communications Server ソフトウェアがアクティブでないために verb が正常に実行されない場合、Communications Server は次のパラメーターを戻します。

primary_rc

AP_COMM_SUBSYSTEM_ABENDED

Communications Server ソフトウェアに障害が起きました。

Communications Server ソフトウェアがアクティブでない場合は、Communications Server から *secondary_rc* は戻されません。

システム・エラー: システム・エラーのために verb が実行されない場合、Communications Server は次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

オペレーティング・システムのコールからの戻りコード。この戻りコードの意味については、ファイル `/usr/include/sys/errno.h` 内の戻り値を調べてください。

REGISTER_MS_APPLICATION

REGISTER_MS_APPLICATION verb は、MDS_MU を受信できる MDS レベルのアプリケーションとして、MS アプリケーションを Communications Server に登録します。この verb を発行する前に、アプリケーションは CONNECT_MS_NODE を発行して Communications Server ノードのターゲット・ハンドルを取得する必要があります。このハンドルは、REGISTER_MS_APPLICATION の MS エントリー・ポイントに対する必須パラメーターです。

アプリケーションは、非同期 MS エントリー・ポイントを使用して、この verb を常に発行し、コールバック・ルーチンを指定する必要があります。Communications Server はこのコールバック・ルーチンを使用して、受信した MDS_MU をアプリケーションに戻します (MS エントリー・ポイントの詳細については、7 ページの『第 2 章 MS アプリケーションの作成』を参照してください)。

VCB 構造体

```
typedef struct register_ms_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  ms_appl_name[8]; /* MS application name          */
    unsigned char  ms_category[8]; /* MS category                  */
    AP_UINT16      max_rcv_size;    /* Maximum size that can be received */
} REGISTER_MS_APPLICATION;
```

指定パラメーター

アプリケーションは、REGISTER_MS_APPLICATION verb を発行するときには、次のパラメーターを指定します。

opcode AP_REGISTER_MS_APPLICATION

ms_appl_name

このアプリケーションを識別する名前。1 つのアプリケーションを、異なるアプリケーション名を使用して複数回、登録できます。アプリケーション名には、次の条件があります。

- MS アプリケーションとして現在登録されている他のアプリケーションで使用している名前と一致してはなりません。
- NODE または UNIX のいずれかであってはなりません。これらは、Communications Server コンポーネントで使用するために予約されています。
- 8 文字の長さにしなければなりません。必要に応じて、右側を EBCDIC スペース文字 (0x40) で埋めてください。
- 次のいずれかにしてください。
 - タイプ 1134 文字 (大文字の A~Z と数字の 0~9) を使用する EBCDIC ストリング。
 - 「IBM Systems Network Architecture: Management Services Reference」の付録で指定されている MS Discipline-Specific Application Program のいずれか。

REGISTER_MS_APPLICATION

ms_category

特定の MS カテゴリについて、アプリケーションがそのフォーカル・ポイントの名前を取得する必要がある場合は、カテゴリ名を指定してください。アプリケーションがフォーカル・ポイント情報を取得する必要がない場合は、このパラメーターを 8 つの 2 進ゼロに設定してください。アプリケーションは、異なる MS カテゴリ名について複数回、登録できます。

MS カテゴリ名は、次のいずれかにしてください。

- ユーザーが定義したカテゴリ名で、タイプ 1134 文字 (大文字の A~Z と数字の 0~9) を使用する 8 バイトの EBCDIC スtring。
- 「*IBM Systems Network Architecture: Management Services Reference*」の付録の『MS Discipline-Specific Application Programs』の表に指定されているカテゴリ名のいずれか。

どのタイプの名前も、必要に応じてスStringの後に 8 バイトまでスペース文字 (0x40) で埋める必要があります。

Communications Server は、REGISTER_MS_APPLICATION で指定されたコールバック・ルーチンの FP_NOTIFICATION 指示を使用して、フォーカル・ポイントの詳細情報を戻します。その後、フォーカル・ポイントが変更されると、Communications Server は新しい情報を含む別の FP_NOTIFICATION を送信します。

max_rcv_size

1 つのメッセージでアプリケーションが受信できる最大バイト数。着呼 MDS_MU が最大バイト数より長い場合、Communications Server はそれをセグメント化し、各セグメントを個別の MDS_MU_RECEIVED シグナルで配信します。

戻りパラメーター

verb の実行後は、Communications Server は、その実行が成功したかどうかを示すパラメーターを戻し、成功しなかった場合は、その理由を示すパラメーターを戻します。

正常に実行された場合

verb が正常に実行された場合は、Communications Server は以下のパラメーターを戻します。

primary_rc

AP_OK

verb の実行が成功した場合は、Communications Server から *secondary_rc* は戻されません。

正常に実行されなかった場合

verb の実行が失敗した場合は、Communications Server はエラーのタイプを示す 1 次戻りコード、および実行が失敗した理由を具体的に示す 2 次戻りコードを戻します。

パラメーター・チェック: パラメーター・エラーが原因で verb が実行されなかった場合、Communications Server 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_MS_APPL_NAME_ALREADY_REGD

指定した名前での別のアプリケーションが現在登録されているか、アプリケーションが、2つの予約名 NODE と UNIX のいずれかを指定しました。

AP_INVALID_APPLICATION_NAME

指定したアプリケーション名に EBCDIC タイプ 1134 文字セットに入っていない文字が含まれ、そのアプリケーション名が MS Discipline-Specific Application Program 名の 1 つではありません。

AP_INVALID_CATEGORY_NAME

EBCDIC タイプ 1134 文字セットに入っていない文字が、指定したカテゴリ名に含まれ、そのカテゴリ名が MS Discipline-Specific Application Program カテゴリ名の 1 つではありません。

AP_INVALID_TARGET_HANDLE

verb が使用するエン트리・ポイントで指定されたターゲット・ハンドルは、直前の CONNECT_MS_NODE verb で戻された有効な値ではありません。

AP_SYNC_NOT_ALLOWED

アプリケーションは、同期 MS エン트리・ポイントを使用してこの verb をコールバック・ルーチン内から発行しました。verb をコールバック・ルーチンから発行するには、アプリケーションは非同期エン트리・ポイントを使用する必要があります。

状態チェック: 状態エラーのために verb が実行されない場合、Communications Server は次のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

AP_INVALID_TARGET_STATE

アプリケーションは、CONNECT_MS_NODE または DISCONNECT_MS_NODE が未解決のときにこの verb を発行しました。

Communications Server ソフトウェアがアクティブでない場合: Communications Server ソフトウェアがアクティブでないために verb が実行されない場合は、Communications Server は、次のパラメーターを戻します。

primary_rc

次のいずれかです。

AP_COMM_SUBSYSTEM_NOT_LOADED

Communications Server ソフトウェアがロードされていません。

REGISTER_MS_APPLICATION

AP_COMM_SUBSYSTEM_ABENDED

Communications Server ソフトウェアに障害が起きました。

Communications Server ソフトウェアがアクティブでない場合は、Communications Server から *secondary_rc* は戻されません。

MDS サポートが構成されていない場合: Communications Server の構成のために *verb* が実行できない場合は、Communications Server は次のパラメーターを戻しません。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

Communications Server ローカル・ノードは、MDS レベルのネットワーク管理アプリケーションをサポートするように構成されていません。NMVT レベルのアプリケーションのみを使用できます。

Communications Server が MDS レベルをサポートするように構成されていない場合は、CS/AIX から *secondary_rc* は戻されません。

システム・エラー: システム・エラーのために *verb* が実行されない場合、Communications Server は次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

オペレーティング・システムのコールからの戻りコード。この戻りコードの意味については、ファイル `/usr/include/sys/errno.h` 内の戻り値を調べてください。

REGISTER_NMVT_APPLICATION

REGISTER_NMVT_APPLICATION *verb* は、NMVT を受信できる NMVT レベルのアプリケーションとして、MS アプリケーションを Communications Server に登録します。この *verb* は、通常、NMVT レベルのアプリケーションで使用しますが、MDS_MU に変換されたあとで NMVT を受信できる MDS レベルのアプリケーションで使用することもできます。この *verb* を発行する前に、アプリケーションは CONNECT_MS_NODE を発行して Communications Server ノードのターゲット・ハンドルを取得する必要があります。このハンドルは、REGISTER_NMVT_APPLICATION の MS エントリー・ポイントに対する必須パラメーターです。

アプリケーションは、非同期 MS エントリー・ポイントを使用して、この *verb* を常に発行し、コールバック・ルーチンを指定する必要があります。Communications Server は、このコールバック・ルーチンを使用して、受信した NMVT をアプリケーションに戻します。MS エントリー・ポイントの詳細については、7 ページの『第 2 章 MS アプリケーションの作成』を参照してください。

NMVT 内の宛先名と MS 主ベクトル・キーの両方が、このコールに指定した値と一致する場合にのみ、Communications Server は NMVT をこのアプリケーションに経路指定します。詳細については、5 ページの『NMVT ルーティング』を参照してください。

VCB 構造体

```
typedef struct register_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  ms_appl_name[8];     /* MS application name         */
    AP_UINT16      ms_vector_key_type;  /* MS vector key accepted by appl */
    unsigned char  conversion_required; /* MDS level application requesting */
                                     /* MDS_MUs                     */
} REGISTER_NMVT_APPLICATION;
```

指定パラメーター

アプリケーションは、REGISTER_NMVT_APPLICATION verb を発行するときに、次のパラメーターを指定します。

opcode AP_REGISTER_NMVT_APPLICATION

ms_appl_name

このアプリケーションを識別する名前。1 つのアプリケーションを、異なるアプリケーション名を使用して複数回、登録できます。アプリケーション名には、次の条件があります。

- *ms_vector_key_type* パラメーターで指定されたキーの範囲と同じ範囲を受信するように現在登録されている、他のアプリケーションで使用している名前と一致してはなりません。
- NODE または UNIX のいずれかであってはなりません。これらは、Communications Server コンポーネントで使用するために予約されています。
- 8 文字の長さにしなければなりません。必要に応じて、右側を EBCDIC スペース文字 (0x40) で埋めてください。
- 次のいずれかにしてください。
 - タイプ 1134 文字 (大文字の A~Z と数字の 0~9) を使用する EBCDIC ストリング。
 - 「*IBM Systems Network Architecture: Management Services Reference*」の付録で指定されている MS Discipline-Specific Application Program のいずれか。

NMVT 内の MS 主ベクトルの宛先アプリケーション名 (0x50) サブフィールドと、このパラメーターに指定した値が一致する場合のみ、着呼 NMVT がこのアプリケーションに経路指定されます。

ms_vector_key_type

アプリケーションが受け付ける 1 つ以上の MS 主ベクトル・キー。ここに

REGISTER_NMVT_APPLICATION

指定した 1 つ以上の値と、NMVT 内の MS 主ベクトル・キーが一致する場合のみ、Communications Server は、着呼 NMVT をこの verb を発行したアプリケーションに経路指定します。

次のいずれかを指定してください。

0xnnnn 特定の主ベクトル・キーの 2 バイトの 16 進値。

AP_SPCF_KEYS

0x8061~0x8064 の範囲内のすべての主ベクトル・キーを受け付けます。この値は、SNA サービス・ポイント・コマンド機能 (SPCF) をインプリメントしているアプリケーションで使用します。アプリケーションがこの機能をインプリメントしていない場合は、この値を使用しないでください。 *ms_appl_name* パラメーターは、SPCF キーを受け付けるように登録されている他のアプリケーションの名前と一致してはなりません。

AP_ALL_KEYS

すべての主ベクトル・キーを受け付けます。 *ms_appl_name* パラメーターは、すべてのキーを受け付けるように登録されている他のアプリケーションの名前と一致してはなりません。

アプリケーションは、複数の REGISTER_NMVT_APPLICATION verb を発行し (同じまたは異なるアプリケーション名で)、複数のキーまたは複数の範囲のキーの NMVT を受け付けます。

Communications Server は、名前とキーの両方を使用して、NMVT を受信するアプリケーションを特定します。したがって、2 つ以上のアプリケーションが異なるアプリケーション名を使用する場合、同じ範囲のキーの NMVT (AP_SPCF_KEYS または AP_ALL_KEYS) を受け付けるように登録できません。ただし、1 つのキーに対応する NMVT を受け付けられるアプリケーションは 1 つのみです。主ベクトル・キーを指定した場合、その指定したキーに対応する NMVT の受け付けを別のアプリケーションがすでに登録していると、verb はエラーを戻します。

conversion_required

登録アプリケーションが MDS レベルかどうか、NMVT を MDS_MU に変換する必要があるかどうかを指定します。次のいずれかを指定してください。

AP_YES アプリケーションは MDS レベルです。NMVT を MDS_MU に変換する必要があります。

AP_NO アプリケーションは NMVT レベルです。NMVT を変換する必要はありません。

戻りパラメーター

verb の実行後は、Communications Server は、その実行が成功したかどうかを示すパラメーターを戻し、成功しなかった場合は、その理由を示すパラメーターを戻します。

正常に実行された場合

verb が正常に実行された場合は、Communications Server は以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

verb の実行が成功した場合は、Communications Server から *secondary_rc* は戻されません。

正常に実行されなかった場合

verb の実行が失敗した場合は、Communications Server はエラーのタイプを示す 1 次戻りコード、および実行が失敗した理由を具体的に示す 2 次戻りコードを戻します。

パラメーター・チェック: パラメーター・エラーが原因で verb が実行されなかった場合、Communications Server は、以下のパラメーターを戻します。

```
primary_rc
    AP_PARAMETER_CHECK
```

```
secondary_rc
    次の値のいずれかです。
```

AP_ALL_APPL_ALREADY_REGISTERED

次のエラー状態のいずれかを表します。

- このアプリケーションは、すべてのキーを受け付けるようにすでに登録しています。
- 同じアプリケーション名を使用してすべてのキーを受け付けるように、別のアプリケーションがすでに登録しています。
- アプリケーションは、2 つの予約名 NODE と UNIX のいずれかを 사용하여すべてのキーを受け付けるように登録しました。

AP_INVALID_APPLICATION_NAME

指定したアプリケーション名に EBCDIC タイプ 1134 文字セットに入っていない文字が含まれ、そのアプリケーション名が MS Discipline-Specific Application Program 名の 1 つではありません。

AP_INVALID_TARGET_HANDLE

verb が使用するエントリー・ポイントで指定されたターゲット・ハンドルは、直前の CONNECT_MS_NODE verb で戻された有効な値ではありません。

AP_KEY_APPL_ALREADY_REGISTERED

次のエラー状態のいずれかを表します。

- この特定のキーの NMVT を受け付けるように、別のアプリケーションがすでに登録しています。各キーに登録できるアプリケーションは 1 つのみです。
- アプリケーションは、2 つの予約名 NODE と UNIX のいずれかを 사용하여特定のキーを受け付けるように登録しました。

REGISTER_NMVT_APPLICATION

AP_SPCF_APPL_ALREADY_REGD

次のエラー状態のいずれかを表します。

- このアプリケーションは、SPCF キーを受け付けるようにすでに登録しています。
- 同じアプリケーション名を使用して SPCF キーを受け付けるように、別のアプリケーションがすでに登録しています。
- アプリケーションは、2 つの予約名 NODE と UNIX のいずれかを使用して SPCF キーを受け付けるように登録しました。

AP_SYNC_NOT_ALLOWED

アプリケーションは、同期 MS エントリー・ポイントを使用してこの verb をコールバック・ルーチン内から発行しました。verb をコールバック・ルーチンから発行するには、アプリケーション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。

状態チェック: 状態エラーのために verb が実行されない場合、Communications Server は次のパラメーターを戻します。

primary_rc

AP_STATE_CHECK

secondary_rc

AP_INVALID_TARGET_STATE

アプリケーションは、CONNECT_MS_NODE または DISCONNECT_MS_NODE が未解決のときにこの verb を発行しました。

Communications Server ソフトウェアがアクティブでない場合: Communications Server ソフトウェアがアクティブでないために verb が実行されない場合は、Communications Server は、次のパラメーターを戻します。

primary_rc

次のいずれかです。

AP_COMM_SUBSYSTEM_NOT_LOADED

Communications Server ソフトウェアがロードされていません。

AP_COMM_SUBSYSTEM_ABENDED

Communications Server ソフトウェアに障害が起こりました。

Communications Server ソフトウェアがアクティブでない場合は、Communications Server から *secondary_rc* は戻されません。

システム・エラー: システム・エラーのために verb が実行されない場合、Communications Server は次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

オペレーティング・システムのコールからの戻りコード。この戻りコードの意味については、ファイル `/usr/include/sys/errno.h` 内の戻り値を調べてください。

SEND_MDS_MU

MDS レベルのアプリケーションは、この verb を使用して MDS_MU 形式のネットワーク管理データを送信します。また、MDS レベルのアプリケーションは、TRANSFER_MS_DATA を使用して NMVT 形式のデータを送信することもできます。アラート情報を送信するときには、必ず SEND_MDS_MU を使用せずに、TRANSFER_MS_DATA を使用してください。

アプリケーションは、送信のための完全な MDS_MU を指定できます。また、アプリケーションは必要なサブベクトルの一部を指定し、Communications Server に要求して別のサブベクトルを追加することもできます。Communications Server が追加するサブベクトルの形式などの MDS_MU の形式の詳細については、「*IBM Systems Network Architecture: Formats*」を参照してください。

宛先アプリケーションが NMVT レベルである場合、Communications Server は指定された MDS_MU を NMVT に自動的に変換します。

宛先アプリケーションに MDS_MU を送信している間に発生したエラーは、そのエラーが検出された場所別に、それぞれの方法でアプリケーションに報告されます。

- Communications Server ローカル・ノードはエラーを検出した場合、エラー戻りコードを SEND_MDS_MU verb に戻します。
- リモート・ノードがエラーを検出した場合、エラー MDS_MU を送信します。アプリケーションが MDS_MU を受信するように登録している場合、Communications Server は、エラー MDS_MU を MDS_MU_RECEIVED 指示でアプリケーションに戻します。

VCB 構造体

```
typedef struct send_mds_mu
{
    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  options;        /* Verb options                 */
    unsigned char  reserv3;        /* reserved                     */
    unsigned char  originator_id[8]; /* Originator ID               */
    unsigned char  pu_name[8];     /* Physical unit name          */
    unsigned char  reserv4[4];     /* reserved                     */
    AP_UINT16      dlen;           /* Length of data              */
    unsigned char  *dptr;         /* Data                         */
} SEND_MDS_MU;
```

指定パラメーター

アプリケーションは SEND_MDS_MU を発行するときに、次のパラメーターを指定します。

opcode AP_SEND_MDS_MU

options このパラメーターは、1 バイトの値であり、選択したオプションを指示するために、個々のビットを次のように使用します。ビット 0 は最上位ビットで、ビット 7 は最下位ビットです。他のインプリメンテーションとの互換性を確保するために、ビット 0 ~ 3 のビット値は、1 の値がアクションなしを指示し、0 の値がアクションを指示するように定義されています

ビット 0

「Date/Time (日付 / 時刻)」サブベクトルをデータに追加します。このビットは、次の値のいずれかに設定してください。

- 0 Communications Server にサブベクトルの追加を要求します。
- 1 Communications Server にサブベクトルを追加しないことを要求します。

ビット 1

製品セット ID サブベクトルをデータに追加します。このビットは、次のいずれかに設定してください。

- 0 Communications Server にサブベクトルの追加を要求します。アプリケーションが製品セット ID サブベクトルをすでに含むデータを指定する場合、Communications Server 自身の製品セット ID サブベクトルを既存のサブベクトルの直前に追加します。
- 1 Communications Server にサブベクトルを追加しないことを要求します。

ビット 2

予約済みです。0 に設定する必要があります。

ビット 3

Communications Server エラー・ログ・ファイルにデータを記録します。このビットは、次のいずれかに設定してください。

- 0 Communications Server にデータの記録を要求します。
- 1 Communications Server にデータを記録しないことを要求します。

ビット 4

MS が MS データを宛先アプリケーションに送るときに、デフォルトの経路を使うか、直接経路を使うかについて指定します。このビットは、次のいずれかに設定してください。

- 0 Communications Server がデフォルトの経路を使用するように要求します。アプリケーションが宛先アプリケーションを記述する FP_NOTIFICATION 指示を受信しておらず、*fp_routing* パラメーターを AP_DIRECT に設定していない限り、デフォルトの経路に指定してください。詳細については、47 ページの『FP_NOTIFICATION』を参照してください。

- 1 Communications Server が直接経路を使用するように要求します。

ビット 5 ~ 7

予約済みです。0 に設定する必要があります。

originator_id

verb を発行したコンポーネントの名前。Communications Server エラー・ログ・ファイルにデータが記録されている場合、この名前はログ・メッセージの発信元を識別するために使用されます。データが記録されていない場合は、この名前は使用されません。

このオプション・パラメーターは、ローカルで表示できる文字を使用した 8 文字までの ASCII ストリングです。先頭文字を入れたくない場合は、その文字を 0x00 に設定してください。

pu_name

この MDS-MU の宛先物理装置。次のいずれかに設定してください。

単一の PU 名

右側を EBCDIC スペース文字 (0x40) で埋めた 8 バイトのタイプ A EBCDIC ストリングを指定します。着呼 NMVT から変換された MDS_MU_RECEIVED 指示に応答する SEND_MDS_MU を使用するアプリケーションは、MDS_MU_RECEIVED 指示で受信した *pu_name* を指定します。

この場合、*pu_name* は、リンク・ステーション (LS) の定義に指定されている *pu_name* と一致しなければなりません。MDS_MU は、このリンク・ステーションを介して送信されます。LS の定義の詳細については、「*IBM Communications Server for Data Center Deployment on AIX or Linux 管理ガイド*」を参照してください。

すべて 2 進ゼロ

通常の MDS ルーティング・プロトコルを使用して移送される MDS_MU にこの値を使用します。

dlen アプリケーションで指定されるデータ・ストリングの長さ。

dptr アプリケーションで指定されるデータ・ストリングへのポインター。次の場合を除いて、このデータ・ストリングに完全な MDS_MU を組み込むことが必要です。

- アプリケーションが *options* パラメーターを使用して 1 つ以上のサブベクトルを追加した場合は、指定したデータからこれらのサブベクトルを省略できます。
- 「*Origin Net ID (起点 Net ID)*」および「*Origin NAU Name (起点 NAU 名)*」フィールドは、すべて EBCDIC のスペース (0x40) に設定できます。この場合、Communications Server はデータの送信前に適切な情報を入れます。

戻りパラメーター

verb の実行後は、Communications Server は、その実行が成功したかどうかを示すパラメーターを戻し、成功しなかった場合は、その理由を示すパラメーターを戻します。

正常に実行された場合

verb が正常に実行された場合は、Communications Server は以下のパラメーターを戻します。

```
primary_rc
    AP_OK
```

verb の実行が成功した場合は、Communications Server から *secondary_rc* は戻されません。

正常に実行されなかった場合

verb の実行が失敗した場合は、Communications Server はエラーのタイプを示す 1 次戻りコード、および実行が失敗した理由を具体的に示す 2 次戻りコードを戻します。

パラメーター・チェック: パラメーター・エラーが原因で verb が実行されなかった場合、Communications Server 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_DATA_SIZE

指定した MDS_MU の「length (長さ)」フィールドが *dlen* パラメーターの値に対応していません。

AP_INVALID_MDS_MU_FORMAT

指定したデータ・ストリングに、有効な MDS_MU が入っていません。

AP_INVALID_PU_NAME

Communications Server は、*pu_name* 指定パラメーターで指定した名前をもつ活動 PU を検出できません。

状態チェック: 状態エラーのために verb が実行されない場合、Communications Server は次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_SSCP_PU_SESSION_NOT_ACTIVE

アプリケーションは PU 名を指定しましたが、この PU と SSCP の間にセッションが存在していません。

AP_SYNC_PENDING

この verb は同期エントリー・ポイントを使用して発行されましたが、このターゲット・ハンドルで別の同期 verb が処理中でした。特定のターゲット・ハンドルで、いつでも処理中にできる同期 verb は 1 つのみです。

AP_SYNC_NOT_ALLOWED

アプリケーションは、同期 MS エントリー・ポイントを使用してこの verb をコールバック・ルーチン内から発行しました。verb をコールバック・ルーチンから発行するには、アプリケーション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。

Communications Server ソフトウェアがアクティブでない場合: Communications Server ソフトウェアがアクティブでないために verb が実行されない場合は、Communications Server は、次のパラメーターを戻します。

primary_rc

AP_COMM_SUBSYSTEM_ABENDED

Communications Server ソフトウェアに障害が起きました。

Communications Server ソフトウェアがアクティブでない場合は、Communications Server から *secondary_rc* は戻されません。

MDS サポートが構成されていない場合: Communications Server の構成のために verb が実行できない場合は、Communications Server は次のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

Communications Server ローカル・ノードは、MDS レベルのネットワーク管理アプリケーションをサポートするように構成されていません。NMVT レベルのアプリケーションのみを使用できます。

Communications Server が MDS レベルをサポートするように構成されていない場合は、CS/AIX から *secondary_rc* は戻されません。

システム・エラー: システム・エラーのために verb が実行されない場合、Communications Server は次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

オペレーティング・システムのコールからの戻りコード。この戻りコードの意味については、ファイル `/usr/include/sys/errno.h` 内の戻り値を調べてください。

TRANSFER_MS_DATA

この verb は次のアプリケーションで使用します。

- 直前に受信した NMVT 要求に応答し、非送信請求 NMVT を送信する NMVT レベルのアプリケーション。
- 非送信請求 NMVT (アラート情報など) を送信する NMVT レベルのアプリケーションと MDS レベルのアプリケーション。

TRANSFER_MS_DATA

アプリケーションは、送信のための完全な NMVT を指定できます。また、アプリケーションは必要なサブベクトルの一部を指定し、Communications Server に要求してヘッダー情報や別のサブベクトルを追加することもできます。Communications Server が追加するヘッダーやサブベクトルの形式など、NMVT の形式の詳細については、「*IBM Systems Network Architecture: Formats*」を参照してください。

VCB 構造体

```
typedef struct transfer_ms_data
{
    AP_UINT16      opcode;                /* Verb operation code          */
    unsigned char  data_type;             /* Type of data supplied by appl */
    unsigned char  format;                /* reserved                      */
    AP_UINT16      primary_rc;            /* Primary return code          */
    AP_UINT32      secondary_rc;          /* Secondary return code        */
    unsigned char  options;               /* Verb options                  */
    unsigned char  reserv3;               /* reserved                      */
    unsigned char  originator_id[8];      /* Originator ID                 */
    unsigned char  pu_name[8];            /* Physical unit name            */
    unsigned char  reserv4[4];            /* reserved                      */
    AP_UINT16      dlen;                   /* Length of data                */
    unsigned char  *dptr;                  /* Data                          */
} TRANSFER_MS_DATA;
```

指定パラメーター

アプリケーションは、TRANSFER_MS_DATA verb を発行するときに、次のパラメーターを指定します。

opcode SV_TRANSFER_MS_DATA

data_type

次の値のいずれかを指定してください。

SV_NMVT

完全な NMVT がデータに入っています。データにアラートが含まれ、そのアラートが MDS レベルかマイグレーション・レベルのいずれかのフォーカル・ポイントに送信される場合は、Communications Server はデータを MDS_MU 形式または CP_MSU 形式に変換します。

NMVT_RECEIVED 指示に回答しているアプリケーションは、完全な NMVT を指定して、これを指示する値 SV_NMVT を使用する必要があります。

SV_ALERT_SUBVECTORS

アラート主ベクトルに対する SNA 定義形式の MS サブベクトルがデータに入っています。Communications Server は NMVT ヘッダーとアラート主ベクトル・ヘッダーを追加します。アラートが MDS レベルかマイグレーション・レベルのいずれかのフォーカル・ポイントに送信される場合、Communications Server はデータを MDS_MU 形式または CP_MSU 形式に変換します。

SV_USER_DEFINED

完全な NMVT 要求単位がデータに入っています。Communications Server はデータを常に記録しますが、これをどのフォーカル・ポイントにも送信しません。

SV_PDSTATS_SUBVECTORS

問題判別統計がデータに入っています。Communications Server はデータを常に記録しますが、これをどのフォーカル・ポイントにも送信しません。

options これは 1 バイトの値で、個々のビットは選択したオプションを指示します。ビット 0 は最上位ビットで、ビット 7 は最下位ビットです。他のインプリメンテーションとの互換性を確保するために、ビット 0 ~ 3 のビット値は、1 の値がアクションなしを指示し、0 の値がアクションを指示するように定義されています (*data_type* パラメーターを SV_USER_DEFINED に設定すると、ビット 1 ~ 3 は無視されます)。

ビット 0

「Date/Time (日付 / 時刻)」サブベクトルをデータに追加します。このビットは、次の値のいずれかに設定してください。

- 0 Communications Server にサブベクトルの追加を要求します。
- 1 Communications Server にサブベクトルを追加しないことを要求します。

ビット 1

製品セット ID サブベクトルをデータに追加します。このビットは、次のいずれかに設定してください。

- 0 Communications Server にサブベクトルの追加を要求します。アプリケーションが製品セット ID サブベクトルをすでに含むデータを指定する場合、Communications Server 自身の製品セット ID サブベクトルを既存のサブベクトルの直前に追加します。
- 1 Communications Server にサブベクトルを追加しないことを要求します。

ビット 2

直前に受信した NMVT に対する応答を送信するためにこの verb を使用する場合は、フォーカル・ポイント、または *pu_name* パラメーターで指定した PU にデータを送信します。このビットは、次のいずれかに設定してください。

- 0 Communications Server にデータの送信を要求します。
- 1 Communications Server にデータを送信しないことを要求します。

ビット 3

Communications Server エラー・ログ・ファイルにデータを記録します。このビットは、次のいずれかに設定してください。

- 0 Communications Server にデータの記録を要求します。
- 1 Communications Server にデータを記録しないことを要求します。

ビット 4 から 7

予約済みです。0 に設定する必要があります。

originator_id

verb を発行したコンポーネントの名前。Communications Server エラー・ログ・ファイルにデータが記録されている場合、この名前はログ・メッセージの発信元を識別するために使用されます。データが記録されていない場合は、この名前は使用されません。

このオプション・パラメーターは、ローカルで表示できる文字を使用した 8 文字までの ASCII ストリングです。先頭文字を入れたくない場合は、その文字を 0x00 に設定してください。

pu_name

この NMVT の宛先物理装置。次のいずれかに設定してください。

単一の PU 名

右側を EBCDIC スペース文字 (0x40) で埋めた 8 バイトのタイプ A EBCDIC ストリングを指定します。

TRANSFER_MS_DATA を使用して NMVT_RECEIVED 指示に応答するアプリケーションは、NMVT_RECEIVED 指示で受信した *pu_name* を指定する必要があります。

非送信請求アラートを送信するアプリケーションでは、特定の物理装置へのアラート・データの送信がどうしても必要な場合を除き、*pu_name* パラメーターには値を指定しないでください (このパラメーターのフィールドはすべて 2 進ゼロにしてください)。この場合、*pu_name* は、リンク・ステーション (LS) の定義に指定されている *pu_name* と一致する必要があります。NMVT はこのリンク・ステーションを介して送信されます。LS の定義の詳細については、「*IBM Communications Server for Data Center Deployment on AIX or Linux 管理ガイド*」を参照してください。

すべて 2 進ゼロ

pu_name を指定しません。*data_type* パラメーターに SV_NMVT を設定し、*pu_name* パラメーターにすべて 2 進ゼロを指定している TRANSFER_MS_DATA *verb* に入っているデータは、デフォルトの PU セッションが使用可能な場合にこのセッションを介して送信されます。

dlen アプリケーションで指定されるデータの長さ。

NMVT の最大長は 512 バイトです。アプリケーションが完全な NMVT を指定する場合、データ長は 512 バイトを超えてはなりません。アプリケーションがアラート・サブベクトルを指定する場合、または指定したデータに 1 つ以上のサブベクトルを追加するように Communications Server に要求する場合、必要なヘッダーとサブベクトルを追加した合計の長さが 512 バイトを超えてはなりません。

dptr アプリケーションで指定されるデータ・ストリングへのポインター。データは、*data_type* パラメーターで指定した、NMVT、アラート・サブベクトル、または問題判別統計に有効な形式でなければなりません。

戻りパラメーター

verb の実行後は、Communications Server は、その実行が成功したかどうかを示すパラメーターを返し、成功しなかった場合は、その理由を示すパラメーターを返しません。

正常に実行された場合

verb が正常に実行された場合は、Communications Server は以下のパラメーターを返します。

```
primary_rc
    SV_OK
```

verb の実行が成功した場合は、Communications Server から *secondary_rc* は返されません。

正常に実行されなかった場合

verb の実行が失敗した場合は、Communications Server はエラーのタイプを示す 1 次戻りコード、および実行が失敗した理由を具体的に示す 2 次戻りコードを返します。

パラメーター・チェック: パラメーター・エラーが原因で verb が実行されなかった場合、Communications Server は、以下のパラメーターを返します。

```
primary_rc
    SV_PARAMETER_CHECK
```

```
secondary_rc
    次のいずれかです。
```

SV_INVALID_DATA_TYPE

指定した *data_type* パラメーターが有効な値ではありません。

AP_INVALID_DATA_SIZE

次のいずれかの状況が発生しました。

- アプリケーションは、512 バイトの最大 NMVT サイズを超えるデータ・ストリングを指定しました。
- アプリケーションは、データをアラート・サブベクトルとして指定しました。または、アプリケーションは Communications Server が 1 つ以上のサブベクトルをデータに追加するように指定しましたが、ヘッダーとサブベクトルが追加されたためにデータ・サイズが大きくなり、512 バイトを超えました。

AP_INVALID_PU_NAME

Communications Server は、*pu_name* 指定パラメーターで指定した名前をもつ活動 PU を検出できませんでした。

状態チェック: 状態エラーのために verb が実行されない場合、Communications Server は次のパラメーターを返します。

```
primary_rc
    SV_STATE_CHECK
```

TRANSFER_MS_DATA

secondary_rc

次のいずれかです。

AP_SYNC_PENDING

この verb は同期エントリー・ポイントを使用して発行されましたが、このターゲット・ハンドルで別の同期 verb が処理中でした。特定のターゲット・ハンドルで、いつでも処理中にできる同期 verb は 1 つのみです。

SV_SSCP_PU_SESSION_NOT_ACTIVE

アプリケーションは、*options* パラメーターのビット 2 を 0 に設定してデータを送信するように Communications Server に要求しましたが、適切な PU に対するセッションがアクティブではありませんでした。

AP_SYNC_NOT_ALLOWED

アプリケーションは、同期 MS エントリー・ポイントを使用してこの verb をコールバック・ルーチン内から発行しました。verb をコールバック・ルーチンから発行するには、アプリケーション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。

Communications Server ソフトウェアがアクティブでない場合: Communications Server ソフトウェアがアクティブでないために verb が正常に実行されない場合は、Communications Server は次のパラメーターを戻します。

primary_rc

AP_COMM_SUBSYSTEM_ABENDED

Communications Server ソフトウェアに障害が起きました。

Communications Server ソフトウェアがアクティブでない場合は、Communications Server から *secondary_rc* は戻されません。

システム・エラー: システム・エラーのために verb が実行されない場合、Communications Server は次のパラメーターを戻します。

primary_rc

SV_UNEXPECTED_DOS_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

オペレーティング・システムのコールからの戻りコード。この戻りコードの意味については、ファイル `/usr/include/sys/errno.h` 内の戻り値を調べてください。

UNREGISTER_MS_APPLICATION

UNREGISTER_MS_APPLICATION verb は、MDS_MU を受信するために以前に登録されていたこのアプリケーションが、これ以降は MDS_MU の受信を必要としなくなったことを Communications Server に通知します。この verb が正常に完了すると、以後 Communications Server は受信した MDS_MU をアプリケーションに送信しません。

終了する前に、アプリケーションは必ず登録されたアプリケーション名のすべてに UNREGISTER_MS_APPLICATION を発行してから DISCONNECT_MS_NODE を発行する必要があります。

VCB 構造体

```
typedef struct unregister_ms_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  ms_appl_name[8]; /* MS application name          */
} UNREGISTER_MS_APPLICATION;
```

指定パラメーター

アプリケーションは、UNREGISTER_MS_APPLICATION を発行するときに、次のパラメーターを指定します。

opcode AP_UNREGISTER_MS_APPLICATION

ms_appl_name

登録解除を行っているアプリケーションを識別する名前。これは、REGISTER_MS_APPLICATION を使用してアプリケーションが以前に指定した名前であればなりません。ストリングは 8 文字の長さにして、必要に応じて右側を EBCDIC スペース文字 (0x40) で埋める必要があります。

戻りパラメーター

verb の実行後は、Communications Server は、その実行が成功したかどうかを示すパラメーターを戻し、成功しなかった場合は、その理由を示すパラメーターを戻します。

正常に実行された場合

verb が正常に実行された場合は、Communications Server は以下のパラメーターを戻します。

primary_rc

AP_OK

verb の実行が成功した場合は、Communications Server から *secondary_rc* は戻されません。

正常に実行されなかった場合

verb の実行が失敗した場合は、Communications Server はエラーのタイプを示す 1 次戻りコード、および実行が失敗した理由を具体的に示す 2 次戻りコードを戻します。

パラメーター・チェック: パラメーター・エラーが原因で *verb* が実行されなかった場合、Communications Server は、以下のパラメーターを戻します。

primary_rc

AP_PARAMETER_CHECK

UNREGISTER_MS_APPLICATION

secondary_rc

次のいずれかです。

AP_INVALID_TARGET_HANDLE

指定したターゲット・ハンドルは、直前の `CONNECT_MS_NODE` verb で戻された有効な値ではありませんでした。

AP_MS_APPL_NAME_NOT_REGD

アプリケーションは、この verb に指定したアプリケーション名を使用して `REGISTER_MS_APPLICATION` を以前に発行していません。

状態チェック: 状態エラーのために verb が実行されない場合、Communications Server は次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_INVALID_TARGET_STATE

アプリケーションは、`CONNECT_MS_NODE` または `DISCONNECT_MS_NODE` が未解決のときにこの verb を発行しました。

AP_SYNC_PENDING

この verb は同期エントリー・ポイントを使用して発行されましたが、このターゲット・ハンドルで別の同期 verb が処理中でした。特定のターゲット・ハンドルで、いつでも処理中にできる同期 verb は 1 つのみです。

AP_SYNC_NOT_ALLOWED

アプリケーションは、同期 MS エントリー・ポイントを使用してこの verb をコールバック・ルーチン内から発行しました。verb をコールバック・ルーチンから発行するには、アプリケーション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。

Communications Server ソフトウェアがアクティブでない場合: Communications Server ソフトウェアがアクティブでないために verb が実行されない場合は、Communications Server は、次のパラメーターを戻します。

primary_rc

AP_COMM_SUBSYSTEM_ABENDED

Communications Server ソフトウェアに障害が起きました。

Communications Server ソフトウェアがアクティブでない場合は、Communications Server から *secondary_rc* は戻されません。

MDS サポートが構成されていない場合: Communications Server の構成のために verb が実行できない場合は、Communications Server は次のパラメーターを戻します。

primary_rc

AP_FUNCTION_NOT_SUPPORTED

Communications Server ローカル・ノードは、MDS レベルのネットワーク管理アプリケーションをサポートするように構成されていません。NMVT レベルのアプリケーションのみを使用できます。

Communications Server が MDS レベルをサポートするように構成されていない場合は、CS/AIX から *secondary_rc* は戻されません。

システム・エラー: システム・エラーのために *verb* が実行されない場合、Communications Server は次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

オペレーティング・システムのコールからの戻りコード。この戻りコードの意味については、ファイル `/usr/include/sys/errno.h` 内の戻り値を調べてください。

UNREGISTER_NMVT_APPLICATION

UNREGISTER_NMVT_APPLICATION *verb* は、特定のアプリケーション名に対する NMVT を受信するため、以前に登録されていたこのアプリケーションが、そのアプリケーション名に対する NMVT の受信を必要としなくなったことを Communications Server に通知します。

アプリケーションが同じアプリケーション名を複数の

REGISTER_NMVT_APPLICATION *verb* に使用して、さまざまなタイプの NMVT を受け付けた場合、このアプリケーション名を登録解除すると、そのアプリケーションは、これ以降にはどのタイプの NMVT も受信しません。ただし、アプリケーションが複数の名前を使用して登録した場合は、そのアプリケーションは、残りのアプリケーション名で指定したタイプの NMVT を続けて受け付けます。

終了する前に、アプリケーションは必ず登録されたアプリケーション名のすべてに UNREGISTER_NMVT_APPLICATION を発行してから DISCONNECT_MS_NODE を発行する必要があります。

VCB 構造体

```
typedef struct unregister_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  ms_app_name[8];  /* MS application name          */
} UNREGISTER_NMVT_APPLICATION;
```


指定パラメーター

アプリケーションは、UNREGISTER_NMVT_APPLICATION を発行するとき、次のパラメーターを指定します。

opcode AP_UNREGISTER_NMVT_APPLICATION

ms_appl_name

登録解除を行っているアプリケーションを識別する名前。これは、REGISTER_NMVT_APPLICATION を使用してアプリケーションが以前に指定した名前であればなりません。文字列は 8 文字の長さにして、必要に応じて右側を EBCDIC スペース文字 (0x40) で埋める必要があります。

戻りパラメーター

verb の実行後は、Communications Server は、その実行が成功したかどうかを示すパラメーターを返し、成功しなかった場合は、その理由を示すパラメーターを返します。

正常に実行された場合

verb が正常に実行された場合は、Communications Server は以下のパラメーターを返します。

primary_rc

AP_OK

verb の実行が成功した場合は、Communications Server から *secondary_rc* は戻されません。

パラメーター・チェック: パラメーター・エラーが原因で *verb* が実行されなかった場合、Communications Server は、以下のパラメーターを返します。

primary_rc

AP_PARAMETER_CHECK

secondary_rc

次のいずれかです。

AP_APPL_NOT_REGISTERED

アプリケーションは、この *verb* に指定したアプリケーション名を使用して REGISTER_NMVT_APPLICATION を以前に発行していません。

AP_INVALID_TARGET_HANDLE

指定したターゲット・ハンドルは、直前の CONNECT_MS_NODE *verb* で戻された有効な値ではありませんでした。

状態チェック: 状態エラーのために *verb* が実行されない場合、Communications Server は次のパラメーターを返します。

primary_rc

AP_STATE_CHECK

secondary_rc

次のいずれかです。

AP_INVALID_TARGET_STATE

アプリケーションは、CONNECT_MS_NODE または DISCONNECT_MS_NODE が未解決のときにこの verb を発行しました。

AP_SYNC_PENDING

この verb は同期エントリー・ポイントを使用して発行されましたが、このターゲット・ハンドルで別の同期 verb が処理中でした。特定のターゲット・ハンドルで、いつでも処理中にできる同期 verb は 1 つのみです。

AP_SYNC_NOT_ALLOWED

アプリケーションは、同期 MS エントリー・ポイントを使用してこの verb をコールバック・ルーチン内から発行しました。verb をコールバック・ルーチンから発行するには、アプリケーション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。

Communications Server ソフトウェアがアクティブでない場合: Communications Server ソフトウェアがアクティブでないために verb が実行されない場合は、Communications Server は、次のパラメーターを戻します。

primary_rc

AP_COMM_SUBSYSTEM_ABENDED

Communications Server ソフトウェアに障害が起きました。

Communications Server ソフトウェアがアクティブでない場合は、Communications Server から *secondary_rc* は戻されません。

システム・エラー: システム・エラーのために verb が実行されない場合、Communications Server は次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

オペレーティング・システムのコールからの戻りコード。この戻りコードの意味については、ファイル `/usr/include/sys/errno.h` 内の戻り値を調べてください。

UNREGISTER_NMVT_APPLICATION

第 4 章 管理サービス指示

この章では、それぞれの指示についての次の情報を提供します。

- 指示の目的、および Communications Server が指示をアプリケーションに戻す条件。
- 指示の説明。整合性を保つために、verb 制御ブロック (VCB) という用語を使用して指示を説明しますが、VCB 構造体は、アプリケーションが発行する verb には関連していません。すべての VCB 構造体は、ヘッダー・ファイル `/usr/include/sna/ms_c.h` (AIX) または `/opt/ibm/sna/include/ms_c.h` (Linux) に定義されています。
- VCB 構造体のパラメーター別に、次の情報を示しています。
 - 説明
 - 戻される可能性のある値とその意味
 - 追加情報 (必要な場合)

指定パラメーター値と戻りパラメーター値の多くは数値で表されます。コーディングを単純にするには、アプリケーションをさらにポータブルにしてからプログラムのソースを読みやすくしてください。これらの値はヘッダー・ファイル `ms_c.h` に定義されている記号定数で示されます。例えば、FP_NOTIFICATION 指示の `opcode` (命令コード) パラメーターは、記号定数 `AP_FP_NOTIFICATION` で示される値です。

システムのタイプにより、これらの値をメモリーに保管する方法が異なるため、指定パラメーターに値を設定するとき、または戻りパラメーターの値をテストするときは、数値ではなく記号定数を使用することが重要です。ヘッダー・ファイルに示されている値は、使用しているシステムで認識される形式ではない場合があります。

注: アプリケーションは VCB を MS verb 用に割り振りますが、Communications Server は VCB を指示用に割り振ります。したがって、アプリケーションにはコールバック・ルーチン内からのみ、VCB 情報へのアクセス権があります。Communications Server がコールバック・ルーチンに提供する VCB ポインターは、コールバック・ルーチンの外側では無効となります。アプリケーションは、コールバック・ルーチン内から必要な処理すべてを完了させる必要があります。あるいは、アプリケーションがルーチン外部で使用する必要のあるすべての VCB データのコピーを作成する必要があります。

FP_NOTIFICATION

Communications Server は、特定の MS カテゴリーのフォーカル・ポイントに関する情報を要求した MDS レベルのアプリケーションに、この状況情報を送信します。アプリケーションがこの情報を要求するには、フォーカル・ポイント・データ・ストリングの一部として指定した特定の MS カテゴリーの名前を指定して `REGISTER_MS_APPLICATION` を発行します。Communications Server は、`FP_NOTIFICATION` を送信してそのカテゴリーに対するアプリケーションの現在の

フォーカル・ポイントを、アプリケーションに通知します。フォーカル・ポイントが変更されるたびに、Communications Server は、別の FP_NOTIFICATION を送信します。

この指示は、アプリケーションが REGISTER_MS_APPLICATION verb で指定したコールバック・ルーチンを使用して戻されます。このコールバック・ルーチンの要件の詳細については、11 ページの『ms_async エントリー・ポイントに指定されたコールバック・ルーチン』を参照してください。

VCB 構造体

```
typedef struct fp_notification
{
    AP_UINT16      opcode;
    unsigned char  reserv2;           /* reserved */
    unsigned char  format;           /* reserved */
    AP_UINT16      primary_rc;       /* Primary return code */
    AP_UINT32      secondary_rc;     /* Secondary return code */
    unsigned char  fp_routing;       /* routing to use with this focal point */
    unsigned char  reserv1;         /* reserved */
    AP_UINT16      fp_data_length;   /* Length of incoming focal point data */
    unsigned char  *fp_data;        /* Focal point data */
} FP_NOTIFICATION;
```

パラメーター

Communications Server は、FP_NOTIFICATION を MDS レベルのアプリケーションに送信するときに、次のパラメーターを設定します。

opcode AP_FP_NOTIFICATION

fp_routing

アプリケーションが MDS_MU をこのフォーカル・ポイントに送信するときに、デフォルトの経路を使うか、直接経路を使うかについて指定します。値は次のとおりです。

AP_DEFAULT

デフォルトの経路を使用して MDS_MU をフォーカル・ポイントに送信します。

AP_DIRECT

MDS_MU をセッションでフォーカル・ポイントに直接経路を指定します。

fp_data_length

フォーカル・ポイント・データの長さ。長さは 78 バイトまでです。

fp_data

フォーカル・ポイント・データ。次のサブベクトルで構成されます。

- 「Focal Point™ Notification (フォーカル・ポイント通知)」(0xE1) サブベクトル。
- 「Focal Point Identification (フォーカル・ポイント識別)」(0x21) サブベクトル。これには、「MS Category (MS カテゴリー)」サブフィールドが含まれます。「MS Category (MS カテゴリー)」サブフィールドは、アプリケーションがフォーカル・ポイント情報を要求したカテゴリーを識別し、次のサブフィールドを含んでいます。

- 「Focal point network identifier (NETID)(フォーカル・ポイント・ネットワーク ID)」
- 「Focal point network accessible unit (NAU)(フォーカル・ポイント・ネットワーク・アクセス可能単位名)」
- 「Application Id (アプリケーション ID)」

このフォーカル・ポイントに関連した MS カテゴリの MDS_MU を送信するとき、アプリケーションは、適切なフォーカル・ポイントに MDS_MU が確実に経路指定されるようにこれらのサブフィールドの情報を MDS_MU に入れる必要があります。これらのサブベクトルに含まれる情報の詳細については、IBM 資料の「*System Network Architecture: Formats*」を参照してください。

MDS_MU_RECEIVED

次の場合に、Communications Server は、このデータ指示を使用して MDS_MU を MDS レベルのアプリケーションに経路指定します。

- MDS レベルのリモート・アプリケーションが MDS_MU を送信し、このアプリケーションが MDS_MU を受け付けるために REGISTER_MS_APPLICATION を使用した場合。
- リモート・アプリケーションが NMVT を送信し、このアプリケーションが MDS_MU への変換後に NMVT を受け付けるために REGISTER_NMVT_APPLICATION を使用した場合。どの MS アプリケーションが着呼 NMVT を受信するかを Communications Server が判別する方法の詳細については、5 ページの『NMVT ルーティング』を参照してください。

MDS_MU_RECEIVED 指示を戻すために、Communications Server はアプリケーションが REGISTER_MS_APPLICATION verb または REGISTER_NMVT_APPLICATION verb で指定したコールバック・ルーチンを使用します。このコールバック・ルーチンの要件の詳細については、11 ページの『ms_async エントリ・ポイントに指定されたコールバック・ルーチン』を参照してください。

VCB 構造体

```
typedef struct mds_mu_received
{
    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  first_message; /* First message for current MDS_MU */
    unsigned char  last_message; /* Last message for current MDS_MU */
    unsigned char  pu_name[8];    /* Physical unit name          */
    unsigned char  reserv3[8];    /* reserved                     */
    AP_UINT16      mds_mu_length; /* Length of incoming MDS_MU    */
    unsigned char  *mds_mu;      /* MDS_MU data                 */
} MDS_MU_RECEIVED;
```

パラメーター

Communications Server は、MDS_MU_RECEIVED 指示を MS アプリケーションに送信するときに、次のパラメーターを設定します。

opcode AP_MDS_MU_RECEIVED

first_message

このメッセージはこの MDS_MU に対する最初の、または唯一のメッセージであるかどうかを指示します。MDS_MU は、通常、アプリケーションに単一メッセージとして送信されます (*first_message* と *last_message* の両方が AP_YES)。ただし、MDS_MU が、アプリケーションが REGISTER_MS_APPLICATION を発行したときに指定される *max_rcv_size* より大きいと、Communications Server は MDS_MU を分割し、これをアプリケーションに複数メッセージとして送信します。値は次のとおりです。

AP_YES この MDS_MU に対して最初の、または唯一のメッセージ。

AP_NO この MDS_MU に対して 2 番目の、または後続のメッセージ。

last_message

このメッセージはこの MDS_MU に対して最後の、または唯一のメッセージであるかどうかを指示します。MDS_MU は、通常、アプリケーションに単一メッセージとして送信されます (*first_message* と *last_message* の両方が AP_YES)。ただし、MDS_MU が、アプリケーションが REGISTER_MS_APPLICATION を発行したときに指定される *max_rcv_size* より大きいと、Communications Server は MDS_MU を分割し、これをアプリケーションに複数メッセージとして送信します。値は次のとおりです。

AP_YES この MDS_MU に対して最後の、または唯一のメッセージ。

AP_NO セグメント化された MDS_MU に対して最初の、または後続のメッセージ。少なくとも 1 つのメッセージがあとに続きます。

pu_name

MDS_MU が着呼 NMVT から変換された場合は、このパラメーターはその NMVT の受信元の物理装置の名前です。NMVT が応答を必要とする場合は、アプリケーションは SEND_MDS_MU verb を使用して応答を送信し、SEND_MDS_MU の *pu_name* パラメーターをこの名前に設定する必要があります。

アプリケーションが、REGISTER_NMVT_APPLICATION verb を使用して、MDS_MU への変換後に NMVT を受け付ける MDS レベルのアプリケーションとして登録した場合にのみ、MDS_MU は着呼 NMVT から変換されます。MDS_MU が MDS レベルのトランスポート・メカニズムから受信された場合、このパラメーターは 2 進ゼロに設定されます。

mds_mu_length

このメッセージに含まれる MDS_MU データの長さ。このメッセージは、*first_message* パラメーターと *last_message* パラメーターによって、完全な MDS_MU、または完全な MDS_MU のセグメントになります。

mds_mu

MDS_MU データ・ストリングへのポインター。

MS_STATUS

Communications Server は、登録されたアプリケーション (MDS レベルか NMVT レベルのいずれか) にこの状況指示を送信し、Communications Server システムの状況に関する次の変更のいずれかをアプリケーションに通知します。

- 接続ノードまたは関連するコンポーネントがアクティブでなくなったために、Communications Server ローカル・ノードへのアプリケーションの通信パスが失われました。
- Communications Server ソフトウェアは停止済みです。

Communications Server は、アプリケーションが REGISTER_MS_APPLICATION verb または REGISTER_NMVT_APPLICATION verb に指定したコールバック・ルーチンで MS_STATUS 指示を戻します。このコールバック・ルーチンの要件の詳細については、11 ページの『ms_async エントリ・ポイントに指定されたコールバック・ルーチン』を参照してください。

アプリケーションが MS_STATUS 指示を受信すると、Communications Server は DISCONNECT_MS_NODE を除いて、関連するターゲット・ハンドルを使用する後続の verb をすべてリジェクトします。

VCB 構造体

```
typedef struct ms_status
{
    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      status;        /* status being reported   */
    AP_UINT32      dead_target_handle; /* Handle of dead connection */
    unsigned char  reserv1[32];    /* reserved                 */
} MS_STATUS;
```

パラメーター

Communications Server は、MS_STATUS 指示を MS アプリケーションに送信するときに、次のパラメーターを設定します。

opcode AP_MS_STATUS

status

AP_TARGET_HAS_DIED

この値は、接続ノードまたは Communications Server ソフトウェアが実行状態ではないことを示します。

dead_target_handle

このパラメーターのヌル値は、ローカル・コンピューター (アプリケーションが実行されているコンピューター) の Communications Server ソフトウェアが停止していることを示します。アプリケーションが使用していたすべてのターゲット・ハンドルは切断され、無効になっています。

MS_STATUS

このパラメーターの非ヌル値は、障害が起きたノードのターゲット・ハンドルを示します。アプリケーションは、このターゲット・ハンドルに DISCONNECT_MS_NODE を発行し、このハンドルに関連したリソースを解放する必要があります。

アプリケーションは、定期的に CONNECT_MS_NODE を発行して、ターゲット・ノードへの再接続を試行できます。ターゲット・ノードまたはローカル Communications Server ソフトウェアが再始動するまで、このコールは正常終了しません。

NMVT_RECEIVED

NMVT を受信するように登録した NMVT レベルのアプリケーションに、リモート・ノードから受信した NMVT を経路指定するために、Communications Server はこのデータ指示を使用します。どの MS アプリケーションが着呼 NMVT を受信するかを Communications Server が判別する方法の詳細については、5 ページの『NMVT ルーティング』を参照してください。

この指示は、アプリケーションが REGISTER_NMVT_APPLICATION verb で指定したコールバック・ルーチンを使用して戻されます。このコールバック・ルーチンの要件の詳細については、11 ページの『ms_async エントリ・ポイントに指定されたコールバック・ルーチン』を参照してください。

VCB 構造体

```
typedef struct nmvt_received
{
    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];     /* Physical unit name           */
    unsigned char  reserv3[6];    /* reserved                      */
    AP_UINT16      nmvt_length;    /* Length of incoming NMVT     */
    unsigned char  *nmvt;         /* NMVT data                    */
} NMVT_RECEIVED;
```

パラメーター

Communications Server は、NMVT_RECEIVED 指示を MS アプリケーションに送信するときに、次のパラメーターを設定します。

opcode AP_NMVT_RECEIVED

pu_name

NMVT の発信元の物理装置の名前。これは 8 バイトの EBCDIC タイプ A スtringで、この名前が 8 バイト未満の場合は、右側が EBCDIC スペース文字で埋められます。

着呼 NMVT が応答を必要とする場合は、アプリケーションは TRANSFER_MS_DATA を使用して応答を送信する必要があります。TRANSFER_MS_DATA の *pu_name* パラメーターを、ここに戻された *pu_name* に設定する必要があります。

nmvt_length

NMVT データの長さ。長さは 512 バイトまでです。

nmvt

REGISTER_NMVT_APPLICATION に指定された 1 つ以上のタイプの MS 主ベクトルを含む完全な NMVT。

付録 A. MS 関数セット

付録 A では、Communications Server MS API がサポートしている SNA MS 関数セットについての情報を提供します。これらの関数セットについての詳細は、IBM 資料の「*Systems Network Architecture: APPN Architecture Reference*」を参照してください。

基本関数セット

Communications Server MS API は、次の基本関数セットをサポートしています。

- 管理サービス - マルチドメイン・サポート (MDS)
 - 150 SNA/MS MDS 共通ベース
 - 151 SNA/MS MDS エンド・ノード・サポート
 - 152 SNA/MS MDS ネットワーク・ノード・サポート
- 管理サービス - MS 機能関数セット
 - 160 SNA/MS MS_CAPS 基本エンド・ノード・サポート
 - 161 SNA/MS MS_CAPS バックアップ/暗黙フォーカル・ポイント
 - 163 SNA/MS MS_CAPS 基本ネットワーク・ノード・サポート
- 管理サービス - エントリー・ポイント・アラート関数セット
 - 170 SNA/MS MS EP アラート基本サブセット

オプションの関数セット

Communications Server MS API は、次のオプションの関数セットをサポートしています。

- 管理サービス - MS 機能関数セット
 - 162 SNA/MS MS_CAPS 制御範囲 (SOC) エンド・ノード
 - 164 SNA/MS MS_CAPS サブエリア FP
- 管理サービス - エントリー・ポイント・アラート関数セット
 - 171 SNA/MS アラート問題診断データ
 - 174 SNA/MS オペレーター開始アラート
 - 175 SNA/MS アラート修飾メッセージ・データ
 - 176 SNA/MS アラート自己定義メッセージ・テキスト・サブベクトル
 - 177 SNA/MS LAN アラート
 - 178 SNA/MS SDLC/LAN LLC アラート
 - 179 SNA/MS X.21 アラート
 - 181 SNA/MS X.25 アラート
 - 182 SNA/MS CPMS 保留アラート

サポートしていない関数セット

Communications Server MS は、次の関数セットはサポートしていません。

- 管理サービス - ファイル・サービス (オプション・セット 1500、1501)
- 管理サービス - 変更管理 (オプション・セット 1510 ~ 1518)

サポートしていない関数セット

- 管理サービス - 操作管理 (オプション・セット 1520、1521) オプション・セット 1520、SNA/MS 共通操作サービスは、Communications Server サービス・ポイント・コマンド機能によってインプリメントされます。

付録 B. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものであり、本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町 19 番 21 号

日本アイ・ピー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

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

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, North Carolina 27709-2195
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。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。

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

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態を提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生した創作物には、次のように、著作権表示を入れていただく必要があります。

® (お客様の会社名) (西暦年).このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。® Copyright IBM Corp. _年を入れる_.

商標

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

インテル、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。

Java™ およびすべての Java 関連の商標は Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

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

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

Adobe および PostScript は、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

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。

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

参考文献

以下の IBM 資料では、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。

- IBM Communications Server for AIX
- IBM Communications Server for Linux
- システム・ネットワーク体系 (SNA)
- 拡張プログラム間通信機能 (APPC)
- プログラミング

IBM Communications Server for AIX および IBM Communications Server for Linux の資料については、その要旨が説明されています。他の資料については、ここではタイトルおよび資料番号のみが示されています。

IBM Communications Server for AIX 関連資料

IBM Communications Server for AIX 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 のソフトコピーへのアクセスの方法については、「*IBM Communications Server for AIX 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 ~ 15 MB のハード・ディスク・スペースが必要になります (このスペースは、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 AIX 移行ガイド* (SC88-6949)

この資料は、Communications Server for AIX バージョン 4.2 以前のバージョンから IBM Communications Server for AIX バージョン 6 への移行方法を説明しています。

- *IBM Communications Server for AIX 入門* (GC88-6947)

この資料は IBM Communications Server for AIX の概要を示すもので、サポートされているネットワークの特性、インストール、構成、および操作について説明しています。

- *IBM Communications Server for AIX 管理ガイド* (SC88-6950)

この資料は、SNA および IBM Communications Server for AIX の概要、および IBM Communications Server for AIX の構成と操作に関する解説です。

- *IBM Communications Server for AIX 管理コマンド解説書* (SD88-6675)

この資料では、SNA および IBM Communications Server for AIX のコマンドについて説明しています。

- *IBM Communications Server for AIX or Linux CPI-C プログラマーズ・ガイド* (SC88-5826)

この資料では、「C」または Java の熟練したプログラマーを対象として、IBM Communications Server 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ています。

- *IBM Communications Server for AIX or Linux APPC プログラマーズ・ガイド* (SC88-5825)

この資料では、拡張プログラム間通信機能 (APPC) を使用するアプリケーション・プログラムを作成するために必要な情報を記載しています。

- *IBM Communications Server for AIX or Linux LUA プログラマーズ・ガイド* (SC88-5827)

この資料では、従来型 LU アプリケーション・プログラミング・インターフェース (LUA) を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。

- *IBM Communications Server for AIX or Linux 共通サービス Verb プログラマーズ・ガイド* (SC88-5824)

この資料では、Common Service Verbs (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 AIX or Linux MS プログラマーズ・ガイド* (SC88-5829)

この資料では、管理サービス (MS) API を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。

- *IBM Communications Server for AIX NOF プログラマーズ・ガイド* (SC88-6958)

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 AIX 診断ガイド* (SC88-6951)

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 AIX or Linux APPC アプリケーション・スイート* (SC88-5828)

この資料では、IBM Communications Server for AIX で使用される APPC アプリケーションについて説明しています。

- *IBM Communications Server for AIX 用語集* (SC88-6952)

この資料は、IBM Communications Server for AIX 関連資料で頻繁に使用される用語とその定義を包括的に収録しています。

IBM Communications Server for Linux 関連資料

IBM 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 ～ 15 MB のハード・ディスク・スペースが必要になります (このスペースは、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 および GC88-9997)

この資料は IBM 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、および操作について説明しています。この資料には、次の 2 つのバージョンがあります。

IBM Communications Server for Linux 入門 (GC88-9996)

IBM Communications Server for Linux on System z 入門 (GC88-9997)

- *IBM Communications Server for Linux 管理ガイド* (SC88-9999)

この資料は、SNA および IBM Communications Server for Linux の概要、および IBM 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)

この資料では、SNA および IBM Communications Server for Linux のコマンドについて説明しています。

- *IBM Communications Server for AIX or Linux CPI-C プログラマーズ・ガイド* (SC88-5826)

この資料では、「C」または Java の熟練したプログラマーを対象として、IBM Communications Server 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ています。

- *IBM Communications Server for AIX or Linux APPC プログラマーズ・ガイド* (SC88-5825)

この資料では、拡張プログラム間通信機能 (APPC) を使用するアプリケーション・プログラムを作成するために必要な情報を記載しています。

- *IBM Communications Server for AIX or Linux LUA プログラマーズ・ガイド* (SC88-5827)

この資料では、従来型 LU アプリケーション・プログラミング・インターフェース (LUA) を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。

- *IBM Communications Server for AIX or Linux 共通サービス Verb プログラマーズ・ガイド* (SC88-5824)

この資料では、Common Service Verbs (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 AIX or Linux MS プログラマーズ・ガイド* (SC88-5829)

この資料では、管理サービス (MS) API を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。

- *IBM Communications Server for Linux NOF プログラマーズ・ガイド* (SC88-8591)

この資料では、Node Operator Facility (NOF) API を使用してアプリケーション・プログラムを作成するために必要な情報を記載しています。

- *IBM Communications Server for Linux 診断ガイド* (GC88-8601)

この資料では、SNA ネットワークの問題解決について説明しています。

- *IBM Communications Server for AIX or Linux APPC アプリケーション・スイート* (SC88-5828)

この資料では、IBM Communications Server for Linux で使用される APPC アプリケーションについて説明しています。

- *IBM Communications Server for Linux 用語集* (GC88-8602)

この資料は、IBM Communications Server for Linux 関連資料で頻繁に使用される用語とその定義を包括的に収録しています。

システム・ネットワーク体系 (SNA) 関連資料

次の資料では、SNA ネットワークについての情報を記載しています。

- *Systems Network Architecture: Format and Protocol Reference Manual—Architecture Logic for LU Type 6.2* (英文番号 SC30-3269)
- *Systems Network Architecture: Formats* (英文番号 GA27-3136)
- *Systems Network Architecture: Guide to SNA Publications* (英文番号 GC30-3438)
- *Systems Network Architecture: Network Product Formats* (英文番号 LY43-0081)
- *Systems Network Architecture: Technical Overview* (英文番号 GC30-3073)
- *Systems Network Architecture: APPN Architecture Reference* (英文番号 SC30-3422)
- *Systems Network Architecture: Sessions between Logical Units* (英文番号 GC20-1868)
- *Systems Network Architecture: LU 6.2 Reference—Peer Protocols* (英文番号 SC31-6808)
- *Systems Network Architecture: Transaction Programmer's Reference Manual for LU Type 6.2* (英文番号 GC30-3084)
- *Systems Network Architecture: 3270 Datastream プログラマー用解説書* (英文番号 GA23-0059)
- *Networking Blueprint Executive Overview* (英文番号 GC31-7057)
- *Systems Network Architecture: Management Services Reference* (英文番号 SC30-3346)

APPC 関連資料

次の資料では、拡張プログラム間通信機能 (APPC) についての情報を記載しています。

- *APPC Application Suite V1 User's Guide* (英文番号 SC31-6532)
- *APPC Application Suite V1 Administration* (英文番号 SC31-6533)
- *APPC Application Suite V1 Programming* (英文番号 SC31-6534)
- *APPC Application Suite V1 Online Product Library* (英文番号 SK2T-2680)

- *APPC Application Suite Licensed Program Specifications* (英文番号 GC31-6535)
- *z/OS V1R2.0 Communications Server: APPC Application Suite User's Guide* (英文番号 SC31-8809)

プログラミング関連資料

次の資料では、プログラミングについての情報を記載しています。

- *共通プログラミング・インターフェース コミュニケーション・インターフェース CPI-C 解説書* (SC88-7217)
- *Communications Server for OS/2 Warp 日本語版 32ビット アプリケーション・プログラミングの手引き* (邦文番号 SC88-5585: 英文番号 SC31-8152)

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]

エントリー・ポイント 7

[カ行]

記号定数 17, 47

コールバック・ルーチン

概要 12

要件 12

comp_proc パラメーター 10

REGISTER_* verb に指定された 12

子プロセス 13

[サ行]

指示 2, 47

受信データ指示 2, 47

MDS_MU 49

NMVT 52

[タ行]

ターゲット・ハンドル 8, 10, 12

データ構造

MDS_MU 49

NMVT 52

データの送信

MDS_MU 形式 31

NMVT 形式 35

同期エントリー・ポイント 7, 8

[ナ行]

ノードとの通信

開始 18

終了 21

障害 51

[ハ行]

非同期エントリー・ポイント 7

フォーカル・ポイントに関する情報の取得

48

ヘッダー・ファイル 17

[マ行]

マイグレーション・レベルの製品 1

マルチプロセス 13

[ラ行]

ローカル・ノードからの登録解除

MDS レベルのアプリケーション 40

NMVT レベルのアプリケーション 43

ローカル・ノードへの登録

MDS レベルのアプリケーション 23, 26

NMVT レベルのアプリケーション 26

A

AIX アプリケーション

コンパイルとリンク 14

AIX アプリケーションのコンパイル 14

AIX アプリケーションのリンク 14

C

Communications Server MS サポート 2

comp_proc (コールバック・ルーチン) 10

CONNECT_MS_NODE

概要 18

指定パラメーター 18

戻りパラメーター 19

VCB 構造体 18

corr (相関係数) 11, 12

CP_MSU 1

D

DISCONNECT_MS_NODE

概要 21

指定パラメーター 21

戻りパラメーター 21

VCB 構造体 21

F

FP_NOTIFICATION

概要 48

使用方法 4

FP_NOTIFICATION (続き)

パラメーター 48

VCB 構造体 48

L

Linux アプリケーション

コンパイルとリンク 14

Linux アプリケーションのコンパイル 14

Linux アプリケーションのリンク 14

M

MDS サポートが構成されていない場合 26, 35, 42

MDS レベルの製品 1

MDS_MU

受信データ指示 49

送信中のエラー 31

MDS レベルの製品による使用 1

NMVT からの変換 5, 49

MDS_MU_RECEIVED

概要 49

使用方法 4

パラメーター 50

VCB 構造体 50

MS verb の要約 2

ms エントリー・ポイント

概要 7

指定パラメーター 8

戻り値 9

MS 関数セット

オプションの 55

基本 55

MS データの受信 4

MS データの送信 3, 4

MS のカテゴリーのフォーカル・ポイント 48

ms_async エントリー・ポイント

概要 7

関数コール 9

コールバック・ルーチン 12

指定パラメーター 10

戻り値 11

ms_c.h ヘッダー・ファイル 17

MS_STATUS

使用方法 4

説明 51

パラメーター 51

VCB 構造体 51

N

NMVT

- 受信データ指示 52
- 主ベクトル・キー 5
- 転送先名 5
- ルーティング 5
- MDS_MU への変換 5, 49

NMVT の送信 3, 4

NMVT レベルの製品 1

NMVT_RECEIVED

- 使用方法 4
- 説明 52
- パラメーター 52
- VCB 構造体 52

R

REGISTER_MS_APPLICATION

- 指定パラメーター 23
- 使用時期 3
- 説明 23
- 戻りパラメーター 24
- VCB 構造体 23

REGISTER_NMVT_APPLICATION

- 指定パラメーター 27
- 使用時期 3
- 説明 26
- 戻りパラメーター 28
- VCB 構造体 27

S

SEND_MDS_MU

- 指定パラメーター 31
- 使用方法 3, 4
- 説明 31
- 戻りパラメーター 33
- VCB 構造体 31

SNA MS サポート 1, 55

T

TRANSFER_MS_DATA

- 指定パラメーター 36
- 使用方法 3, 4
- 説明 35
- 戻りパラメーター 39
- VCB 構造体 36

U

UNREGISTER_MS_APPLICATION

- 指定パラメーター 41
- 使用方法 4

UNREGISTER_MS_APPLICATION (続き)

- 説明 40
- 戻りパラメーター 41
- VCB 構造体 41

UNREGISTER_NMVT_APPLICATION

- 指定パラメーター 44
- 使用方法 5
- 説明 43
- 戻りパラメーター 44
- VCB 構造体 44

V

VCB 構造体、ヘッダー・ファイルに定義されている 17

VCB 構造体へのポインター 8, 10, 12

verb の参照情報 17

verb の要約 2



プログラム番号: 5725-H32

Printed in Japan

SC88-5829-01



日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21