

IBM Communications Server for Linux or AIX



共通サービス Verb プログラマーズ・ガイド

バージョン 6.4

IBM Communications Server for Linux or AIX



共通サービス Verb プログラマーズ・ガイド

バージョン 6.4

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、51 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 AIX バージョン6.4 (プログラム番号 5765-E51) および新しい版またはテクニカル・ニュースレターで明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

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

原典： SC23-8589-00
IBM Communications Server for Linux or AIX
Common Service Verbs Programmer's Guide
Version 6.4

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 2009.4

© Copyright International Business Machines Corporation 2000, 2009.

目次

表	v	VCB 構造体	24
本書について	vii	指定パラメーター	24
本書の対象読者	vii	戻りパラメーター	24
本書の使用法	viii	DEFINE_TRACE	26
本書の構成	viii	VCB 構造体	26
表記上の規則	viii	指定パラメーター	27
グラフィック規則	viii	戻りパラメーター	28
詳細について	ix	SNACTL 環境変数	29
第 1 章 概念	1	GET_CP_CONVERT_TABLE	29
共通サービス Verb の要約	1	VCB 構造体	30
CSV エントリー・ポイント: AIX または Linux シス テム	2	指定パラメーター	30
CSV エントリー・ポイント: Windows	3	戻りパラメーター	31
ACSSVC_C	4	LOG_MESSAGE	33
WinCSVStartup	5	VCB 構造体	33
WinCSV	6	指定パラメーター	33
WinAsyncCSV	7	戻りパラメーター	35
WinCSVCleanup	9	ログ・メッセージ・ファイルの作成	36
GetCsvReturnCode	9	TRANSFER_MS_DATA	40
Verb の入力	11	VCB 構造体	41
AIX または Linux での考慮事項	12	指定パラメーター	41
CSV ヘッダー・ファイル	12	戻りパラメーター	43
マルチスレッド・アプリケーション	12	付録 A. コード・ページ	47
CSV アプリケーションのコンパイルとリンク	13	ASCII コード・ページ	47
Windows での考慮事項	14	EBCDIC コード・ページ	48
CSV アプリケーションのコンパイルおよびリンク	14	付録 B. 特記事項	51
移植可能なアプリケーションの作成	14	商標	53
第 2 章 共通サービス Verb の解説	17	参考文献	55
CONVERT	18	IBM Communications Server for AIX 資料	55
VCB 構造体	18	IBM Communications Server for Linux 資料	56
指定パラメーター	18	システム・ネットワーク体系 (SNA) 関連資料	58
戻りパラメーター	20	APPC 関連資料	58
タイプ G 変換テーブルの作成	22	プログラミング関連資料	59
COPY_TRACE_TO_FILE	23	索引	61

表

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

本書について

本書は、C 言語アプリケーション・プログラムで IBM Communications Server for Linux or AIX 共通サービス Verbs (CSV) を使用するためのガイドです。

本書は、IBM Communications Server に適用されます。IBM Communications Server は、AIX[®] が稼働するサーバーまたは Linux が稼働するコンピューターで SNA ネットワーク上の他のノードとの情報交換を可能にする IBM[®] ソフトウェア製品です。

IBM Communications Server には、作動するためのハードウェアに応じた 3 つの異なるインストール変種があります。

IBM Communications Server for AIX (CS/AIX)

IBM Communications Server for AIX は、AIX のバージョン 5.2、5.3 または 6.1 基本オペレーティング・システムで作動します。

IBM Communications Server for Linux (Communications Server for Linux)

IBM Communications Server for Linux、プログラム製品番号 5724-i33 は、次のハードウェア上で作動します。

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

IBM Communications Server for Linux on System z (Communications Server for Linux on System z)

IBM Communications Server for Linux on System z (プログラム製品番号 5724-i34) は、Linux for System z が稼働する System z メインフレーム (s390 または s390x) で作動します。

本書では、相違が明示的に記述されていない限り、Communications Server という名称はこれらすべての変種を示すために使用され、「Communications Server computer」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。

本書は、Communications Server バージョン 6.4 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ている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・プログラムを作成する、熟練した C プログラマーを対象としています。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 ネットワークを介してデータを送受信するトランザクションおよびアプリケーション・プログラムの設計およびコーディングを行います。

す。したがって、アプリケーション・プログラマーは、SNA、トランザクションまたはアプリケーション・プログラムが通信するリモート・プログラム、および AIX / Linux のオペレーティング・システムのプログラミング環境と運用環境について、十分に理解していることが必要です。

本書の使用法

この節では、本書の構成と表記法について説明します。

本書の構成

本書は次の章で構成されています。

- 1 ページの『第 1 章 概念』。共通サービス Verb について簡単に説明し、C プログラムにおけるその使用方法を説明します。
- 17 ページの『第 2 章 共通サービス Verb の解説』。各 verb について詳しく説明します。各 verb の説明には、その verb の目的、verb 制御ブロック (VCB)、指定パラメーターと戻りパラメーターなどの説明が記載されています。
- 47 ページの『付録 A. コード・ページ』。GET_CP_CONVERT_TABLE verb がサポートしている ASCII および EBCDIC コード・ページをリストしています。

表記上の規則

本書で使用する表記上の規則を表 1 に示します。

表 1. 表記上の規則

内容	表記例
資料タイトル	<i>IBM Communications Server for Linux or AIX APPC プログラマーズ・ガイド</i>
ファイル名またはパス名	acssvcc.h
プログラムまたはアプリケーション	snamsgf
コマンドまたは AIX / Linux ユーティリティ	kill
オプションまたはフラグ	-I
パラメーターまたは Motif のフィールド	<i>wVersionRequired; primary_rc</i>
リテラル値、ユーザーが入力できる選択項目 (デフォルト値を含む)	0x0001; 0
定数またはシングナル	SIGPOLL; SV_ASCII_TO_EBCDIC
戻り値	WCSVVERNOTSUPPORTED; 0; AP_OK
指定値を表す変数	<i>programname</i>
環境変数	LD_RUN_PATH
プログラミング verb	CONVERT; TRANSFER_MS_DATA
関数、コール、またはエントリー・ポイント	ACSSVC_P
データ構造	WCSVDATA
16 進値	0x20

グラフィック規則

AIX, LINUX

この記号は、AIX または Linux のオペレーティング・システムにのみ適用されるセクションの開始を示します。これは AIX / Linux サーバーと、AIX、Linux、Linux for pSeries、または Linux for System z 上で稼働する IBM Remote API Client に適用されます。

WINDOWS

この記号は、Windows 上の IBM Remote API Client に適用されるセクションの開始を示します。

■

この記号は、オペレーティング・システム固有のセクションの終了を示します。この記号の後の情報は、オペレーティング・システムに関わらず適用されます。

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関連事項についての追加情報は、参考文献を参照してください。

第 1 章 概念

この章には、CSV アプリケーション・プログラムを開発するときに必要な情報を記載しています。次の情報があります。

- 共通サービス Verb の要約
- CSV エントリー・ポイント
- verb の入力

AIX, LINUX

- AIX または Linux での考慮事項

WINDOWS

- Windows での考慮事項

- 移植可能なアプリケーションの作成

共通サービス Verb の要約

この節では、共通サービス Verb について簡単に説明します。各 verb の詳しい説明は、17 ページの『第 2 章 共通サービス Verb の解説』にあります。

CONVERT

文字ストリングを ASCII から EBCDIC、または EBCDIC から ASCII に変換します。

COPY_TRACE_TO_FILE

AIX, LINUX

トレース・ファイル (1 つまたは複数) の現在の内容を別のファイルにコピーして保管します。

DEFINE_TRACE

特定の API について、トレースを使用可能または使用不可にします。

GET_CP_CONVERT_TABLE

ソース・コード・ページからターゲット・コード・ページへ文字列を変換するための、256 バイトの変換テーブルを作成して戻します。

LOG_MESSAGE

AIX, LINUX

メッセージ・ファイルからメッセージを取り出し、指定されたデータを追加し、エラー・ログ・ファイルまたは監査ログ・ファイルにそのメッセージを記録します。

TRANSFER_MS_DATA

WINDOWS

ネットワーク管理ベクトル・トランスポート (NMVT) データを含む、Systems Network Architecture (SNA) の要求単位 (RU) をビルドします。この verb は NMVT データを NetView に送信でき、それによって問題の診断および解明を集中化することができます。このデータは、ローカルのエラー・ログ・ファイルにも記録されます。

CSV エントリー・ポイント: AIX または Linux システム

AIX, LINUX

C プログラムは次のエントリー・ポイントを介して共通サービス Verb を呼び出します。

```
void ACSSVC_C (  
    void * vcbptr  
);
```

関数に渡されるパラメーターは verb 制御ブロック (VCB) のアドレスのみです。VCB は、実行する verb を識別し、verb が使用する情報を提供し、実行の完了時に verb から戻される情報を入れるための変数からなる構造体です。各 verb にはそれぞれ自身の VCB 構造体があり、Communications Server に添付されているヘッダー・ファイル `/usr/include/sna/acssvcc.h` (AIX) または `/opt/ibm/sna/include/acssvcc.h` (Linux) で宣言されます。`#include` を使用して、共通サービス Verb を発行するアプリケーション・プログラムにこのファイルを組み込んでください。

注: CSV 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、予約済みパラメーターにアクセスしてはなりません。また、verb で使用する他のパラメーターを設定する前に、VCB の全内容をゼロに設定して、これらのパラメーターすべてを必ずゼロにしてください。このようにすると、内部的に使用されるパラメーターが Communications Server で誤って解釈されることがなく、将来のバージョンでこれらのパラメーターが新しい関数を提供するために使用されても、アプリケーションは引き続き動作します。

VCB の内容をゼロに設定するには、`memset` を使用します。

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

以前にインプリメントされた CSV との互換性を保つため、Communications Server はエントリー・ポイント `ACSSVC_P` および `ACSSVC` も提供します。これらのエントリー・ポイントは、`ACSSVC_C` と同様に使用できます。

エントリー・ポイントは、CSV ヘッダー・ファイル `acssvcc.h` で定義されています。

CSV エントリー・ポイント: Windows

WINDOWS

Windows アプリケーションは、次の関数を使用して CSV にアクセスします。

ACSSVC_C

`verb` を発行します。 `verb` はブロックを行います。つまり、CSV が `verb` の処理を終了して結果を戻すまで、アプリケーションのスレッドは中断されます。これは、WinCSV と同じ効果を持ちます。

WinCSVStartup

アプリケーションを Windows CSV ユーザーとして登録し、そのアプリケーションが必要とする関数のレベルが CSV ソフトウェアによってサポートされているかどうかを判別します。

WinCSV `verb` を発行します。 `verb` はブロックを行います。つまり、CSV が `verb` の処理を終了して結果を戻すまで、アプリケーションのスレッドは中断されます。これは、`ACSSVC_C` と同じ効果を持ちます。

WinAsyncCSV

`verb` を発行します。 `TRANSFER_MS_DATA` の場合は例外ですが、それ以外では `verb` はブロックを行います。処理は、WinCSV エントリー・ポイントの場合と同じです。 `TRANSFER_MS_DATA` `verb` は通常、非同期で完了し、ブロックを行いません。CSV はアプリケーション・ウィンドウにメッセージを送付することによって、完了したことを示します。

WinCSVCleanup

アプリケーションが CSV の使用を終了したときに、そのアプリケーションを登録抹消します。

GetCsvReturnCode

CSV `verb` で得られる 1 次および 2 次の戻りコードに関する印刷可能文字列を生成します。

エントリー・ポイントは、Windows CSV ヘッダー・ファイル `wincsv.h` で定義されています。このファイルは、Windows Client ソフトウェアをインストールしたディレクトリー内の、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。

CSV エントリー・ポイント: Windows

WinCSV または WinAsyncCSV 呼び出しを使用して verb の発行を試みる前に、アプリケーションで WinCSVStartup を呼び出さなければなりません。それからアプリケーションは、WinAsyncCSV (非同期) または WinCSV (同期) のいずれかを使用して verb を発行します。エラーを示す戻りコードで verb が戻った場合、アプリケーションで GetCsvReturnCode を使用して、これらの戻りコードのテキスト・ストリングを得ることができます。戻りコードを使用して、標準エラー・メッセージを生成することもできます。

WinCSV または WinAsyncCSV 呼び出しを使用した verb の発行をアプリケーションが終了するには、WinCSVCleanup を呼び出してからでなければなりません。WinCSVCleanup を呼び出した後でさらに verb を発行することはできません。

以下の節で、これらの関数について説明します。

ACSSVC_C

アプリケーションはこの関数を使用して verb を発行します。verb はブロックを行います。つまり、CSV が verb の処理を終了して結果を戻すまで、アプリケーションのスレッドは中断されます。

以前にインプリメントされた CSV との互換性を保つため、Communications Server はエントリー・ポイント ACSSVC_P および ACSSVC も提供します。これらのエントリー・ポイントは、ACSSVC_C と同様に使用できます。これらのエントリー・ポイントは、CSV ヘッダー・ファイル `sdk/wincsv.h` で定義されています。

関数呼び出し

```
void ASCCV_C (
    void * vcbptr
)
```

指定パラメーター

関数に渡されるパラメーターは verb 制御ブロック (VCB) のアドレスのみです。VCB は、実行する verb を識別し、verb が使用する情報を提供し、実行の完了時に verb から戻される情報を入れるための変数からなる構造体です。各 verb にはそれぞれ自身の VCB 構造体があり、Windows 上の Remote API Client に含まれるヘッダー・ファイル `sdk/wincsv.h` で宣言されます。#include を使用して、共通サービス Verb を発行するアプリケーション・プログラムにこのファイルを組み込んでください。

注: CSV 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、予約済みパラメーターにアクセスしてはなりません。また、verb で使用する他のパラメーターを設定する前に、VCB の全内容をゼロに設定して、これらのパラメーターすべてを必ずゼロにしてください。このようにすると、内部的に使用されるパラメーターが Communications Server で誤って解釈されることがなく、将来のバージョンでこれらのパラメーターが新しい関数を提供するために使用されても、アプリケーションは引き続き動作します。

VCB の内容をゼロに設定するには、memset を使用します。

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

戻り値

関数は値を戻しません。

WinCSVStartup

アプリケーションはこの関数を使用して、Windows CSV ユーザーとして登録し、そのアプリケーションが必要とする Windows CSV バージョンが CSV ソフトウェアによってサポートされているかどうかを判別します。

関数呼び出し

```
int WINAPI WinCSVStartup (
    WORD wVersionRequired,
    WCSVDATA far * lpData
);
```

```
typedef struct
{
    WORD wVersion;
    char szDescription[128];
} WCSVDATA;
```

指定パラメーター

提供されているパラメーターは、以下のとおりです。

wVersionRequired

アプリケーションに必要な Windows CSV のバージョン。Communications Server は、バージョン 1.0 をサポートします。

このパラメーターの下位バイトはメジャー・バージョン番号を示し、高位バイトはマイナー・バージョン番号を示します。たとえば、以下のとおりです。

バージョン	<i>wVersionRequired</i>
1.0	0x0001
1.1	0x0101
2.0	0x0002

アプリケーションで複数のバージョンを使用できる場合は、使用できるバージョンのうちの最も高いものを指定してください。

戻り値

関数からの戻り値は、以下のいずれかです。

0 (ゼロ)

アプリケーションは正常に登録されました。また、Windows CSV ソフトウェアは、アプリケーションで指定されたバージョン番号またはそれより低いバージョンをサポートしています。アプリケーションで WCSVDATA 構造体内のバージョン番号を検査して、それが十分に高いものであることを確認してください。

CSV エントリー・ポイント: Windows

WCSVVERNOTSUPPORTED

アプリケーションで指定されたバージョン番号は、Windows CSV ソフトウェアでサポートされている最も低いバージョンより低いものでした。アプリケーションは登録されませんでした。

WCSVSYSNOTREADY

Remote API Client ソフトウェアが開始されていなかったか、あるいはローカル・ノードがアクティブになっていません。アプリケーションは登録されませんでした。

WinCSVStartup からの戻り値がゼロの場合、Windows CSV ソフトウェアのサポート関連情報がWCSVDATA 構造体に入ります。戻り値がゼロでない場合は、この構造体の内容は未定義なので、アプリケーションでこの構造体を検査する必要はありません。この構造体内のパラメーターは、以下のとおりです。

wVersion

ソフトウェアでサポートされている Windows CSV のバージョン番号。これは、*wVersionRequired* パラメーター (事前に定義されているもの) と同じフォーマットです。Communications Server は、バージョン 1.0 をサポートします。

アプリケーションで必要とされるバージョン番号をソフトウェアがサポートしている場合、このパラメーターは *wVersionRequired* パラメーターと同じ値に設定されます。アプリケーションで必要とされるバージョン番号をソフトウェアがサポートしていない場合は、このパラメーターは、ソフトウェアがサポートする最も高いバージョンに設定されます。このバージョンは、アプリケーションで提供されるバージョン番号より低いものです。アプリケーションは戻り値を検査して、以下のようなアクションを実行しなければなりません。

- 戻されたバージョン番号が、要求されたバージョン番号と同じである場合、アプリケーションはこの Windows CSV インプリメンテーションを使用できます。
- 戻されたバージョン番号が、要求されたバージョン番号より低い場合、アプリケーションはこの Windows CSV インプリメンテーションを使用できますが、戻されたバージョン番号でサポートされていない機能を使用しないでください。低いバージョンでは使用できない機能を必要としているためにこれを実行できない場合、アプリケーションは初期化に失敗し、CSV verb を発行しようとはしません。

szDescription

Windows CSV ソフトウェアについて記述しているテキスト・ストリング。

WinCSV

アプリケーションはこの関数を使用して verb を発行します。この verb は、verb の処理が完了するまでブロックを行います。

関数呼び出し

```
void WINAPI WinCSV (  
    long vcbptr  
);
```

指定パラメーター

verb 用の VCB 構造体を指すポインターが、関数の唯一のパラメーターです。これは長整数として定義されるので、長整数を指すポインターからキャストされる必要があります。各 verb の VCB 構造体の定義については、17 ページの『第 2 章 共通サービス Verb の解説』を参照してください。

注: CSV 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、予約済みパラメーターにアクセスしてはなりません。また、verb で使用する他のパラメーターを設定する前に、VCB の全内容をゼロに設定して、これらのパラメーターすべてを必ずゼロにしてください。このようにすると、内部的に使用されるパラメーターが Communications Server で誤って解釈されることがなく、将来のバージョンでこれらのパラメーターが新しい関数を提供するために使用されても、アプリケーションは引き続き動作します。

VCB の内容をゼロに設定するには、memset を使用します。

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

戻り値

関数は値を戻しません。呼び出しから戻ったら、アプリケーションで VCB 構造体内の *primary_rc* および *secondary_rc* 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断してください。VCB 構造体で戻されるパラメーターについては、17 ページの『第 2 章 共通サービス Verb の解説』にある個々の verb の説明を参照してください。

WinAsyncCSV

アプリケーションはこの関数を使用して verb を発行します。

TRANSFER_MS_DATA では、verb は非同期で完了する可能性があります。CSV は、アプリケーションのウィンドウ・ハンドルにメッセージを送付して、完了したことを示します。その他の verb はすべて、同期で完了します。

最初に WinAsyncCSV 呼び出しを使用する前に、アプリケーションで RegisterWindowMessage を使用して、非同期の verb の完了を示すメッセージ用に CSV が使用する、メッセージ ID を取得しなければなりません。詳しくは、14 ページの『Windows での考慮事項』を参照してください。

関数呼び出し

```
HANDLE WINAPI WinAsyncCSV (
    HWND hWnd,
    long vcbptr
);
```

指定パラメーター

提供されているパラメーターは、以下のとおりです。

CSV エントリー・ポイント: Windows

hWnd 非同期の *verb* の完了を示すメッセージを送るために CSV が使用するウィンドウ・ハンドル。

vcbptr *verb* 用の VCB 構造体を指すポインター。このパラメーターは長整数として定義されるので、長整数を指すポインターからキャストされる必要があります。VCB 構造体に関する情報、および個々の *verb* でのこの構造体の使用法についての詳細は、17 ページの『第 2 章 共通サービス Verb の解説』を参照してください。

注: CSV 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、予約済みパラメーターにアクセスしてはなりません。また、*verb* で使用する他のパラメーターを設定する前に、VCB の全内容をゼロに設定して、これらのパラメーターすべてを必ずゼロにしてください。このようにすると、内部的に使用されるパラメーターが Communications Server で誤って解釈されることがなく、将来のバージョンでこれらのパラメーターが新しい関数を提供するために使用されても、アプリケーションは引き続き動作します。

VCB の内容をゼロに設定するには、`memset` を使用します。

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

戻り値: TRANSFER_MS_DATA

関数が正常に終了した場合、戻り値はハンドルです。後で *verb* が完了した際に、CSV が、アプリケーションのウィンドウ・プロシージャに渡すメッセージ内で、このハンドルを ID として使用します (詳しくは、『使用法』を参照してください)。

戻り値が 0 である場合、関数呼び出しが受け入れられなかったことを示します。

戻り値: その他の verb

TRANSFER_MS_DATA 以外のすべての *verb* で、関数は WinCSV エントリー・ポイントと同じように作動し (前の節で説明したとおり)、値を戻しません。呼び出しから戻ったら、アプリケーションで VCB 構造体内の *primary_rc* および *secondary_rc* 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断してください。

使用法

最初に WinAsyncCSV を使用する前に、アプリケーションで RegisterWindowMessage 呼び出しを使用して、非同期の *verb* の完了を示すメッセージ用に CSV が使用する、メッセージ ID を取得しなければなりません。RegisterWindowMessage は、標準の Windows 関数呼び出しであって、CSV に固有のものではありません。この関数について詳しくは、ご使用の Windows 資料を参照してください。(後続の *verb* の前に、呼び出しを再発行する必要はありません。アプリケーションが発行するすべての呼び出しで、戻り値は同じです。)

アプリケーションは関数に文字列『WinAsyncCSV』を渡さなければなりません。戻り値は、後述のとおり、メッセージ ID です。

WinAsyncCSV エントリー・ポイントを使用して発行された `verb` が非同期に完了するたびに、CSV は、WinAsyncCSV 呼び出しで指定されたウィンドウ・ハンドルにメッセージを送付します。メッセージのフォーマットは、以下のとおりです。

- メッセージ ID は、RegisterWindowMessage 呼び出しからの戻り値です。
- `lParam` 引数には、オリジナルの WinAsyncCSV 呼び出しに提供された VCB のアドレスが含まれています。アプリケーションはこのアドレスを使用して、VCB 構造体内の戻りパラメーターにアクセスできます。
- `wParam` 引数には、オリジナルの WinAsyncCSV 呼び出しに戻されたハンドルが含まれています。

WinCSVCleanup

`verb` の発行が終了した後、アプリケーションはこの関数を使用して、Windows CSV ユーザーとして登録抹消します。

関数呼び出し

```
BOOL WINAPI WinCSVCleanup (void);
```

指定パラメーター

この関数にはパラメーターがありません。

戻り値

関数からの戻り値は、以下のいずれかです。

TRUE アプリケーションは正常に登録抹消されました。

FALSE 呼び出しの処理中にエラーが発生し、アプリケーションは登録抹消されませんでした。ログ・ファイルで、失敗した原因を示すメッセージを確認してください。

GetCsvReturnCode

この呼び出しは、提供された VCB からの戻りコードを解釈する印刷可能文字列を返します。この文字列を使用して、`AP_OK` 以外の戻りコードのための、アプリケーションのエラー・メッセージを生成できます。

この呼び出しは、アプリケーションのエンド・ユーザーに表示するための文字列を提供するように設計されています。構成上の問題またはユーザー・エラー (たとえば、必要なコンポーネントが構成されていない、あるいは開始されていないなど) を示す戻りコードの場合は、ユーザーに問題を訂正してもらうために、十分な情報を文字列で与える必要があります。アプリケーション・エラー (たとえば、アプリケーションが無効な `verb` を発行した場合、あるいは必要なパラメーターの提供に失敗した場合など) を示す戻りコードの場合は、一般的にユーザーには問題を訂正できません。このような場合は、文字列はアプリケーション開発者にのみ分かるものになります。

関数呼び出し

```
int WINAPI GetCsvReturnCode (  
    struct svc_hdr FAR * vcbbptr,  
    unsigned int      buffer_length,  
    unsigned char FAR * buffer_addr  
);  
  
typedef struct svc_hdr  
{  
    unsigned short  opcode;          /* Verb identifying operation code.    */  
    unsigned char   opext;          /* Verb extension code - reserved.    */  
    unsigned char   reserv2;        /* Reserved.                          */  
    unsigned short  primary_rc;     /* Primary return code from verb.     */  
    unsigned long   secondary_rc;   /* Secondary (qualifying) return code.*/  
};
```

指定パラメーター

提供されているパラメーターは、以下のとおりです。

vcbptr verb 用の VCB 構造体を指すポインター。VCB 構造体に関する情報、および個々の verb でのこの構造体の使用法についての詳細は、17 ページの『第 2 章 共通サービス Verb の解説』を参照してください。

buffer_length

戻りデータ・ストリングを保持するために、アプリケーションが提供するバッファの長さ (バイト数)。推奨される長さは 256 バイトです。

buffer_addr

戻りデータ・ストリングを保持するために、アプリケーションが提供するバッファのアドレス。

戻り値

関数からの戻り値は、以下のいずれかです。

0x00000000

関数の処理が正常に完了しました。

0x20000001

CSV は、提供された VCB を読み取れなかったか、提供されたデータ・バッファに書き込めませんでした。

0x20000002

提供されたデータ・バッファが小さすぎて、戻り文字列を保持できません。

0x20000003

この関数の戻り文字列を生成するダイナミック・リンク・ライブラリー (CSVSTR32.DLL) をロードできませんでした。

戻り値が 0x00000000 の場合、*buffer_addr* パラメーターで示されたバッファに戻り文字列が入っています。この文字列はヌル文字 (2 進ゼロ) で終了しますが、末尾に改行 (\n) 文字はありません。



Verb の入力

共通サービス Verb を入力する主な手順は次のとおりです。各ステップの説明には、`CONVERT verb` のサンプル・コードを使っています。この `verb` の詳細については、17 ページの『第 2 章 共通サービス Verb の解説』を参照してください。

1. 入力する `verb` に適用される VCB 構造体から構造変数を作成します。

AIX, LINUX

```
#include <acssvcc.h>
.
.
.
struct convert conv_block;
```

WINDOWS

```
#include <wincsv.h>
.
.
.
struct convert conv_block;
```



VCB 構造体は CSV ヘッダー・ファイルの `acssvcc.h` (AIX / Linux) または `wincsv.h` (Windows) で宣言されています。これらの構造体の 1 つは `convert` という名前です。

2. 構造体内の変数を消去 (ゼロに設定) します。

```
memset(&conv_block, 0, sizeof( conv_block ) );
```

将来の CSV のバージョンを使用するためにアプリケーションをアップグレードできるようにするには、このステップが重要です (将来のバージョンで、現行バージョンで予約されているフィールドが使えるようになります)。このステップは、トレース・データのデバッグおよび解釈にも役立ちます。

3. 必須の VCB 変数に値を割り当てます。

```
conv_block.opcode = SV_CONVERT;
conv_block.direction = SV_ASCII_TO_EBCDIC;
conv_block.char_set = SV_AE;
conv_block.len = sizeof(tpstart_name);
conv_block.source = (unsigned char *) tpstart_name;
conv_block.target = (unsigned char *) tpstart.tp_name;
```

フィールド `SV_CONVERT`、`SV_ASCII_TO_EBCDIC`、および `SV_AE` は、整数を表す記号定数です。これらの定数は、CSV ヘッダー・ファイルで定義します。

文字配列 `tpstart_name` には、EBCDIC に変換して文字配列 `tpstart.tp_name` に入る ASCII スtringが入っています。

4. `verb` を呼び出します。その `verb` 用の VCB が入っている構造体を指すポインターが、唯一のパラメーターです。

AIX, LINUX

```
ACSSVC_C ((char *)&conv_block);
```

他の導入 CSV システムとの互換性のために、エントリー・ポイント ACSSVC_P または ACSSVC を ACSSVC_C の代わりに使用することができます。

WINDOWS

```
WinCSV ( (long) ( (char far *) &conv_block ) );
```

verb から戻される値を使用します。

```
if (conv_block.primary_rc == SV_OK)
{
    /* other statements */
    .
    .
}
}
```

AIX または Linux での考慮事項

AIX, LINUX

この節では、AIX または Linux の環境で使用するアプリケーションを開発するときの考慮事項について説明します。

CSV ヘッダー・ファイル

CSV アプリケーションで使用されるヘッダー・ファイルは、**acssvcc.h** です。このファイルには、CSV エントリー・ポイントおよび verb 制御ブロックの定義が入っています。また、このファイルには共通インターフェース・ヘッダー・ファイル **values.c.h** も入っており、その中には、CSV インターフェース時に、システムが提供する、またはシステムが戻すパラメーター値用として定義された定数が含まれています。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) に入っています。

マルチスレッド・アプリケーション

Communications Server CSV ライブラリーはマルチスレッド・アプリケーションをサポートしています。ただし、次のような制約があります。

- 未解決の verb は常に 1 つしか認められません。実行中の verb が別にあると、verb は失敗し、戻りコード AP_STATE_CHECK および AP_SYNC_PENDING を戻しません。

- アプリケーションは、スレッドが終了する前に、必要な終結処理を実行する必要があります。CSV ライブラリーでは、スレッドと verb の使用との関連付けは行われないため、スレッドの終了時に終結処理が自動的に実行されることはありません。

DCE スレッドをサポートしていないライブラリーのバージョンでは、マルチスレッド・アプリケーションは使用しないでください。

CSV 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/csv_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/csv_r64_5.exp -I  
/usr/include/sna
```

Linux アプリケーション

CSV アプリケーションをコンパイルおよびリンクする前に、共用ライブラリーが入っているディレクトリーを指定して、実行時にアプリケーションがそれらのライブラリーを見付けられるようにしてください。そのためには、環境変数 LD_RUN_PATH を `/opt/ibm/sna/lib` に設定するか、あるいは 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/lib -lcsv -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 -lcsv -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 での考慮事項

WINDOWS

この節では、Windows クライアントでアプリケーションを開発する際に注意すべき処理の考慮事項について説明します。

CSV アプリケーションのコンパイルおよびリンク

この節では、Windows 上での CSV プログラムのコンパイルおよびリンク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・オプション

CSV 用の VCB 構造体は、パックされていません。このパッキング方式を変更するようなコンパイラ・オプションは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 *WORD* 境界、*BYTE* パラメーターは *BYTE* 境界です。

ヘッダー・ファイル

Windows CSV アプリケーションに組み込む CSV ヘッダー・ファイルは、**wincsv.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 TP を CSV にリンクするには、TP を API ライブラリー・ファイル **¥sdk\wincsv32.lib** (32 ビット・アプリケーションの場合)、または **¥sdk64\wincsv32.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 TP を CSV にリンクするには、TP に以下の呼び出しを組み込みます。

- **LoadLibrary**。CSV ダイナミック・リンク・ライブラリー **wincsv32.dll** をロードするために使用します。
- **GetProcAddress**。必要な各 CSV エントリー・ポイントで CSV を指定するために使用します (**WinAsyncCSV**、**WinCSVStartup**、**WinCSVCleanup** など)。
- **FreeLibrary**。ライブラリーがそれ以上必要なくなった場合に使用します。

移植可能なアプリケーションの作成

次に示すのは、他の環境へ移植可能な g Communications Server アプリケーションを作成するためのガイドラインです。

- CSV ヘッダー・ファイルは、パス名を付けずに取り込みます。コンパイラーの組み込みオプションを使用して、ファイルを見付けます (この章で前述の、ご使用のオペレーティング・システムのアプリケーションの節を参照してください)。それによって、ファイル・システムが異なる環境でもアプリケーションを使用することができます。
- パラメーター値と戻りコードには、ヘッダー・ファイルに示されている数値ではなく、記号定数名を使用します。それによって、値がメモリーにどのように保管されているかにかかわらず正しい値が使用されます。
- 現在ご使用のオペレーティング・システムに適用される以外の戻りコードについてチェックし (たとえば、switch ステートメントでの「デフォルト」処理を使用して)、適切な診断を実行します。
- 予約と示されているパラメーターは必ず 0 に設定します。

第 2 章 共通サービス Verb の解説

この章では、各共通サービス Verb について解説します。 verb ごとに、次の情報があります。

- verb の定義。
- verb が使用する verb 制御ブロック (VCB) を定義する構造体。 構造体は CSV ヘッダー・ファイルで宣言します。
- verb に指定されるパラメーター (VCB フィールド) と verb から戻されるパラメーター (VCB フィールド)。パラメーターごとに、次の情報があります。
 - 説明
 - 可能な値
 - 補足情報
- 補足情報。 verb の使用方法について説明します。

共通サービス Verb に指定されるパラメーター、また共通サービス Verb から戻されるパラメーターの大部分は 16 進数です。コーディングを単純にするため、これらの値は、ヘッダー・ファイル **values_c.h** の中で定義されている、意味のある記号定数によって表されます。このヘッダー・ファイルは、CSV ヘッダー・ファイル **acssvcc.h** によって組み込まれます。たとえば、**CONVERT verb** の *opcode* (命令コード) パラメーターは、記号定数 **SV_CONVERT** で表される 16 進値です。ファイル **values_c.h** は、CSV VCB で使用される **AP_UINT16** などのパラメーター・タイプの定義も組み込みます。

指定パラメーターに値を設定するとき、または戻されたパラメーターの値をテストするときは、16 進値ではなく記号定数を使用する必要があります。これは、これらの値をメモリーに保管する方法がシステムにより異なるため、提示した値が、ご使用のシステムで認識できるフォーマットではない場合があるためです。

Communications Server だけでなく他の環境でも使用するアプリケーションを作成する場合は、14 ページの『移植可能なアプリケーションの作成』を参照してください。

注: CSV 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、予約済みパラメーターにアクセスしてはなりません。また、verb で使用する他のパラメーターを設定する前に、VCB の全内容をゼロに設定して、これらのパラメーターすべてを必ずゼロにしてください。このようにすると、内部的に使用されるパラメーターが Communications Server で誤って解釈されることがなく、将来のバージョンでこれらのパラメーターが新しい関数を提供するために使用されても、アプリケーションは引き続き動作します。

VCB の内容をゼロに設定するには、`memset` を使用します。

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

CONVERT

CONVERT verb は、ASCII 文字ストリングを EBCDIC に変換し、また EBCDIC 文字ストリングを ASCII に変換します。

変換するストリングをソース・ストリングと呼びます。変換したストリングをターゲット・ストリングと呼びます。

VCB 構造体

AIX, LINUX

```
typedef struct convert
{
    AP_UINT16      opcode;          /* Verb identifying operation code.    */
    unsigned char  opext;          /* Verb extension code - reserved.    */
    unsigned char  reserv2;        /* Reserved.                            */
    AP_UINT16      primary_rc;     /* Primary return code from verb.      */
    AP_UINT32      secondary_rc;   /* Secondary (qualifying) return code. */
    unsigned char  direction;     /* Direction of conversion - ASCII to  */
                                /* EBCDIC or vice-versa.              */
    unsigned char  char_set;       /* Character set to use for the        */
                                /* conversion A, AE, or user-defined G. */
    AP_UINT16      len;           /* Length of string to be converted.   */
    unsigned char  *source;        /* Pointer to string to be converted.  */
    unsigned char  *target;        /* Address to put converted string at.  */
};
```

WINDOWS

```
typedef struct convert
{
    unsigned short opcode;          /* Verb identifying operation code.    */
    unsigned char  opext;          /* Verb extension code - reserved.    */
    unsigned char  reserv2;        /* Reserved.                            */
    unsigned short primary_rc;     /* Primary return code from verb.      */
    unsigned long  secondary_rc;   /* Secondary (qualifying) return code. */
    unsigned char  direction;     /* Direction of conversion - ASCII to  */
                                /* EBCDIC or vice-versa.              */
    unsigned char  char_set;       /* Character set to use for the        */
                                /* conversion A, AE, or user-defined G. */
    unsigned short len;           /* Length of string to be converted.   */
    unsigned char  *source;        /* Pointer to string to be converted.  */
    unsigned char  *target;        /* Address to put converted string at.  */
};
```

指定パラメーター

この verb を使用するプログラムが指定するパラメーターを次に示します。

opcode SV_CONVERT

direction

可能な値は以下のとおりです。

SV_ASCII_TO_EBCDIC

ASCII から EBCDIC 文字への変換

SV_EBCDIC_TO_ASCII

EBCDIC から ASCII 文字への変換

char_set

ソース・ストリングの変換に使用する文字セットを指定します。可能な値は以下のとおりです。

SV_A タイプ A の文字セットは次の文字からなります。

- 大文字
- 数字 0~9
- 特殊文字 \$、#、@、およびスペース

この文字セットは、システム指定のタイプ A 変換テーブルでサポートされます。

ソース・ストリングの最初の文字は、大文字、または特殊文字 \$、#、@ である必要があります。スペースは文字列の末尾にのみ使用できます。小文字は最初の文字以外の位置に使用できますが、大文字に変換されます。

SV_AE タイプ AE の文字セットは次の文字からなります。

- 大文字
- 小文字
- 数字 0~9
- 特殊文字 \$、#、@、およびスペース

この文字セットは、システム指定のタイプ AE 変換テーブルでサポートされます。

ソース・ストリングの最初の文字には、文字セットの任意の文字を使用できます。ストリング全体がスペースの場合を除き、スペースはストリングの末尾にのみ使用できます。大文字小文字の変換は行われません。

SV_G タイプ G の文字セットはユーザー作成の変換テーブルでサポートされます。このテーブルについては、22 ページの『タイプ G 変換テーブルの作成』で詳しく説明します。

AIX, LINUX

このテーブルを入れるファイルは、環境変数 `SNATBLG` で指定する必要があります。この変数をファイルの絶対パス名に設定してください。(環境変数が設定されていない場合、またはファイルが見つからない場合、システムは `SV_TABLE_ERROR` 戻りコードを戻します。)

WINDOWS

Windows クライアントの場合、以下のように、テーブルを含むファイルを *CSVTLG* 値レジストリー・キーで指定する必要があります。

**\\HKEY_LOCAL_MACHINE\SOFTWARE\SNA
Client\SxClient\Parameters\CSV_data**

CSVTLG パラメーターについては、「*IBM Communications Server for Linux 管理ガイド*」、または「*IBM Communications Server for AIX 管理ガイド*」に説明があります。このパラメーターを、ファイルの絶対パス名に設定してください。（ファイルが見付からない場合、システムは戻りコード *SV_TABLE_ERROR* を戻します。）



len 変換する文字の数。

ソース 変換する文字ストリングが入っているバッファのアドレス。

ターゲット

変換後の文字ストリングが入るバッファのアドレス。

このバッファは、ソース・パラメーターが指すバッファと一致したり重なり合ってもかまいません。その場合、変換されたデータ・ストリングはソース・データ・ストリングを上書きします。

戻りパラメーター

verb の実行後、Communications Server は、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。

実行が成功した場合

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

primary_rc
SV_OK

実行が失敗した場合

verb が正常に実行されなかった場合は、Communications Server は、エラーのタイプを示す 1 次戻りコードと、実行が成功しなかった特定の理由を示す 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・エラーのために *verb* が正常に実行されなかった場合は、Communications Server は次のパラメーターを戻します。

primary_rc
SV_PARAMETER_CHECK

secondary_rc
可能な値は以下のとおりです。

SV_CONVERSION_ERROR

ソース・ストリングの中の 1 つ以上の文字が変換テーブルに見つからなかったか、タイプ A または タイプ AE のストリングに組み込

みスペースが見つかりました。これらの文字またはスペースはヌル (0x00) に変換されました。それでも `verb` は実行されました。

SV_INVALID_CHARACTER_SET

`char_set` パラメーターに無効な値が入っていました。

WINDOWS

SV_INVALID_DATA_SEGMENT

提供されたソース・ストリングまたはターゲット・ストリングがデータ・セグメントの境界を越えて拡張したか、あるいはターゲット・データ・セグメントが読み取り/書き込みセグメントではありませんでした。

■■■■

SV_INVALID_DIRECTION

`direction` パラメーターに無効な値が入っていました。

SV_INVALID_FIRST_CHARACTER

タイプ A のソース・ストリングの最初の文字は有効な値ではありません。

SV_TABLE_ERROR

ユーザー作成のタイプ G 変換テーブルが入っているファイルが正しく定義されていないか、アクセスできないか、正しい形式になっていませんでした。

AIX, LINUX

このテーブルを入れるファイルは、環境変数 `SNATBLG` で指定する必要があります。この変数をファイルの絶対パス名に設定してください。

WINDOWS

Windows クライアントの場合、以下のように、テーブルを含むファイルを `CSVTLG` 値レジストリー・キーで指定する必要があります。

**\\HKEY_LOCAL_MACHINE\SOFTWARE\SNA
Client\SxClient\Parameters\CSV_data**

`CSVTLG` パラメーターについては、「*IBM Communications Server for Linux 管理ガイド*」、または「*IBM Communications Server for AIX 管理ガイド*」に説明があります。このパラメーターを、ファイルの絶対パス名に設定してください。

■■■■

その他の状況: 他の状況により、次の 1 次戻りコード (*primary_rc*) が戻される場合があります。

WINDOWS

SV_COMM_SUBSYSTEM_NOT_LOADED

Remote API Client ソフトウェアが始動していません。訂正処置についてシステム管理者に相談してください。

SV_INVALID_VERB_SEGMENT

提供された VCB が、データ・セグメントの境界を越えて拡張しました。

■■■■■

SV_INVALID_VERB

opcode パラメーターがどの *verb* の命令コードとも一致していません。
verb は実行されませんでした。

SV_UNEXPECTED_DOS_ERROR

オペレーティング・システムが *verb* の処理中にエラーを検出しました。オペレーティング・システムからの戻りコードは *secondary_rc* を介して戻されます。問題が解消されない場合は、訂正処置についてシステム管理者に相談してください。

AIX, LINUX

オペレーティング・システムからの戻りコードの意味については、ファイル `/usr/include/errno.h` を参照してください。

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、ご使用のオペレーティング・システムの資料を参照してください。

■■■■■

タイプ G 変換テーブルの作成

GET_CP_CONVERT_TABLE *verb* を使用して、ユーザー作成のタイプ G 変換テーブルを作成することができます。GET_CP_CONVERT_TABLE *verb* については、この章で詳しく説明します。

このテーブルは、32 行分の長さの ASCII ファイルにする必要があります。各行には、16 の文字を表す 32 の 16 進数字を入れる必要があります。最初の 16 行 (256 文字) は、ASCII 文字から変換される EBCDIC 文字を指定します。残りの 16 行は、EBCDIC 文字から変換される ASCII 文字を指定します。

Communications Server では、16 進数字 A~F は大文字でも小文字でもかまいません。ただし、IBMOS/2® Extended Edition が提供する CSV インプリメンテーションと互換性を持たせるために、これらの数字を大文字にする必要がでてくる場合もあります。

Communications Server に添付のファイル `/usr/lib/sna/samples/snatblg.dat` (AIX) または `/opt/ibm/sna/samples/snatblg.dat` (Linux) には、サンプルのタイプ G 変換テーブルが入っています。このテーブルでは、ASCII コード・ページの最初の 127 文字が EBCDIC に変換されます。このファイルの一覧を次に示します。

```
00010203372D2E2F1605250B0C0D0E0F
101112133C3D322618193F27221D351F
405A7F7B5B6C507D4D5D5C4E6B604B61
F0F1F2F3F4F5F6F7F8F97A5E4C7E6E6F
7CC1C2C3C4C5C6C7C8C9D1D2D3D4D5D6
D7D8D9E2E3E4E5E6E7E8E9ADE0BD5F6D
79818283848586878889919293949596
979899A2A3A4A5A6A7A8A9C06AD0A107
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
000102030009007F0000000B0C0D0E0F
10111213000080018190000001D001F
00001C00000A171B0000000000050607
00001600001E0004000000001415001A
20000000000000000000000002E3C282B00
26000000000000000000021242A293B5E
2D2F0000000000000007C2C255F3E3F
0000000000000000000603A2340273D22
00616263646566676869000000000000
006A6B6C6D6E6F707172000000000000
007E737475767778797A00000005B0000
0000000000000000000000000005D0000
7B414243444546474849000000000000
7D4A4B4C4D4E4F505152000000000000
5C00535455565758595A000000000000
30313233343536373839000000000000
```

COPY_TRACE_TO_FILE

AIX, LINUX

`COPY_TRACE_TO_FILE` verb は、API トレース・ファイル (1 つまたは複数) の現行の内容を新規のファイルにコピーし、トレース・ファイルを消去します。これにより、このアプリケーションの現在のトレース・データのコピーを保管することができます。API トレースについて詳しくは、「*IBM Communications Server for Linux 診断ガイド*」、または「*IBM Communications Server for AIX 診断ガイド*」を参照してください。

このアプリケーションで実行中のすべての API トレースは (どの Communications Server API についても) `COPY_TRACE_TO_FILE` を入力する前に停止する必要があります。

COPY_TRACE_TO_FILE

ります。 アクティブなトレースがある場合は、この `verb` を使用する前に `DEFINE_TRACE verb` を使用して停止してください。

VCB 構造体

```
typedef struct copy_trace_to_file
{
    AP_UINT16      opcode;           /* Verb identifying operation code. */
    unsigned char opext;           /* Verb extension code - reserved. */
    unsigned char reserv2;        /* Reserved. */
    AP_UINT16      primary_rc;     /* Primary return code from verb. */
    AP_UINT32      secondary_rc;   /* Secondary (qualifying) return code. */
    unsigned char  reserv3[8];    /* Reserved. */
    unsigned char  file_name[64]; /* File name to write to. */
    unsigned char  file_option;   /* File options. New or overwrite. */
    unsigned char  reserv4[12];   /* Reserved. */
};
```

指定パラメーター

この `verb` を使用するプログラムが指定するパラメーターを次に示します。

opcode SV_COPY_TRACE_TO_FILE

file_name

最大 64 文字のトレース情報を保持するファイルの名前 (およびオプションでパスも)。この名前には、最大 64 文字まで使用することができます。ファイルが現行ディレクトリーにない場合は、絶対パスを指定します。このパスは、この `verb` を入力するいずれのコンピューターでも有効なパスである必要があります。

file_option パラメーターを SV_NEW に設定した場合は、既存のファイルの名前をファイル名として指定することはできません。

file_option

可能な値は以下のとおりです。

SV_NEW *file_name* で指定した名前の新しいファイルを作成します。このファイルがすでに存在している場合は、エラーが戻されます。

SV_OVERWRITE

ファイルが存在する場合は上書きし、存在しない場合はファイルを作成します。

戻りパラメーター

`verb` の実行後、Communications Server は、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。

実行が成功した場合

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

primary_rc
SV_OK

実行が失敗した場合

verb が正常に実行されなかった場合は、Communications Server は、エラーのタイプを示す 1 次戻りコードと、実行が成功しなかった特定の理由を示す 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・エラーのために verb が正常に実行されなかった場合は、Communications Server は次のパラメーターを戻します。

primary_rc

SV_PARAMETER_CHECK

secondary_rc

SV_INVALID_FILE_OPTION

file_option パラメーターに無効な値が入っていました。

状態チェック: 状態エラーのために verb が正常に実行されない場合は、次のパラメーターが戻されます。

primary_rc

SV_STATE_CHECK

secondary_rc

戻される値は次のとおりです。

SV_TRACE_BUFFER_EMPTY

ファイルにコピーするトレース情報はありません。トレース・ファイルが空であるか、SNATRC 環境変数が設定されていませんでした。この環境変数は、アプリケーションの始動前に設定する必要があります。API トレースの制御方法については、「*IBM Communications Server for Linux 診断ガイド*」、または「*IBM Communications Server for AIX 診断ガイド*」を参照してください。

SV_TRACE_NOT_STOPPED

verb を入力したとき、トレースがまだアクティブでした。COPY_TRACE_TO_FILE を入力する前に、CSV、APPC、CPI-C、および RUI インターフェースのトレースをオフにする必要があります。COPY_TRACE_TO_FILE を入力する前に、DEFINE_TRACE を使用して、アクティブなトレースをすべてオフにします。詳しくは、26 ページの『DEFINE_TRACE』を参照してください。

その他の状況: 他の状況により、次の 1 次戻りコード (*primary_rc*) が戻される場合があります。

SV_FILE_ALREADY_EXISTS

file_option パラメーターに (新しい出力ファイルを作成するために) 値 SV_NEW を指定しましたが、指定した名前のファイルがすでに存在します。

SV_INVALID_VERB

opcode パラメーターがどの verb の命令コードとも一致していません。verb は実行されませんでした。

SV_OUTPUT_DEVICE_FULL

出力ファイルのディスクまたはディレクトリーに、トレース情報を保存するための十分なスペースがありません。トレース・ファイルはリセットされま

COPY_TRACE_TO_FILE

せんでした。出力ファイルに使用可能なトレース情報が収められる場合もありますが、一部であって完全な情報ではありません。

SV_UNEXPECTED_DOS_ERROR

オペレーティング・システムが `verb` の処理中にエラーを検出しました。オペレーティング・システムからの戻りコードは `secondary_rc` を介して戻されます。問題が解消されない場合は、訂正処置についてシステム管理者に相談してください。

オペレーティング・システム戻りコードの意味については、`/usr/include/errno.h` ファイルを参照してください。

DEFINE_TRACE

`DEFINE_TRACE verb` は、特定のアプリケーション・プログラミング・インターフェース (API) について、トレースを使用可能または使用不可にします。

この `verb` を発行するアプリケーションを始動する前に、`SNATRC` 環境変数を使用してトレース・ファイルをセットアップする必要があります。API トレースの制御方法について詳しくは、「*IBM Communications Server for Linux 診断ガイド*」、または「*IBM Communications Server for AIX 診断ガイド*」を参照してください。

この `verb` の操作は、`SNACTL` 環境変数により影響されます (詳しくは、29 ページの『*SNACTL 環境変数*』を参照してください)。

VCB 構造体

```
typedef struct define_trace
{
    AP_UINT16      opcode;           /* Verb identifying operation code. */
    unsigned char opext;           /* Verb extension code - reserved. */
    unsigned char reserv2;        /* Reserved. */
    AP_UINT16      primary_rc;      /* Primary return code from verb. */
    AP_UINT32      secondary_rc;    /* Secondary (qualifying) return code.*/
    unsigned char reserv3[8];     /* Reserved. */
    unsigned char dt_set;         /* Trace state to be set (on/off). */
    unsigned char appc;          /* Tracing for APPC. */
    unsigned char nof;           /* Tracing for NOF. */
    unsigned char srpi;          /* Reserved. */
    unsigned char sdlc;          /* Reserved. */
    unsigned char tkn_rng_dlc;    /* Reserved. */
    unsigned char pcnet_dlc;     /* Reserved. */
    unsigned char dft;           /* Reserved. */
    unsigned char acdi;          /* Reserved. */
    unsigned char reserv5;       /* Reserved. */
    unsigned char comm_serv;     /* Tracing for Comm_Serv_API. */
    unsigned char ehllapi;       /* Reserved. */
    unsigned char x25_api;       /* Reserved. */
    unsigned char x25_dlc;       /* Reserved. */
    unsigned char twinax;        /* Reserved. */
    unsigned char ms;            /* Tracing for MS. */
    unsigned char rui;           /* Tracing for RUI interface of LUA. */
    unsigned char etherand;      /* Reserved. */
    unsigned char subsym;        /* Reserved. */
    unsigned char reserv7[8];    /* Reserved. */
    unsigned char reset_trc;     /* Flag to reset the trace files. */
    AP_UINT16      trunc;         /* Truncation size for trace records. */
    AP_UINT16      strg_size;     /* Reserved. */
};
```

```

unsigned char   reserv8[1];           /* Reserved.          */
unsigned char   phys_link[8];        /* Reserved.          */
unsigned char   reserv9[56];        /* Reserved.          */
};

```

指定パラメーター

この `verb` を使用するプログラムが指定するパラメーターを次に示します。

opcode SV_DEFINE_TRACE

dt_set DEFINE_TRACE `verb` を使用してトレースをオンにするかオフにするかを指定します。

可能な値は以下のとおりです。

SV_ON 特定の API (*appc*、*nof*、*comm_serv*、*ms* または *rui*) のパラメーターのビット 0 が 1 に設定されている場合は、その API のトレースを使用可能にします。パラメーターのビット 0 が 0 に設定されている場合は、API のトレースを変更しないでください。

SV_OFF 特定の API のパラメーターのビット 0 が 1 に設定されている場合は、その API のトレースを使用不可にします。パラメーターのビット 0 が 0 に設定されている場合は、API のトレースを変更しないでください。

appc APPC および CPI-C のトレースの状態 (オンまたはオフ) を変更するかどうか指定します。このオプションは、APPC トレースと CPI-C トレースの両方を制御します。これらを個別に制御することはできません。

Communications Server は、このバイトの最上位ビット (ビット 0) のみをチェックします。他のビットは無視されます。

dt_set パラメーターの内容によって APPC および CPI-C のトレースを使用可能または使用不可にするには、このバイトの最上位ビットを 1 に設定します。

APPC および CPI-C のトレースの状態を現在のまま保持するには、このバイトの最上位ビットを 0 に設定します。

nof NOF のトレースの状態 (オンまたはオフ) を変更するかどうか指定します。

Communications Server は、このバイトの最上位ビット (ビット 0) のみをチェックします。他のビットは無視されます。

dt_set パラメーターの内容によって NOF トレースを使用可能または使用不可にするには、このバイトの最上位ビットを 1 に設定します。

NOF のトレースの状態を現在のまま保持するには、このバイトの最上位ビットを 0 に設定します。

comm_serv

共通サービス Verb のトレースの状態 (オンまたはオフ) を変更するかどうか指定します。

Communications Server は、このバイトの最上位ビット (ビット 0) のみをチェックします。他のビットは無視されます。

dt_set パラメーターによって共通サービス Verb のトレースを使用可能または使用不可にするには、このバイトの最上位ビットを 1 に設定します。

共通サービス Verb のトレースの状態を現在のまま保持するには、このバイトの最上位ビットを 0 に設定します。

ms MS トレースの状態 (オンまたはオフ) を変更するかどうか指定します。

Communications Server は、このバイトの最上位ビット (ビット 0) のみをチェックします。他のビットは無視されます。

dt_set パラメーターの内容によって MS トレースを使用可能または使用不可にするには、このバイトの最上位ビットを 1 に設定します。

MS のトレースの状態を現在のまま保持するには、このバイトの最上位ビットを 0 に設定します。

ru LUA の RUI インターフェースのトレースの状態 (オンまたはオフ) を変更するかどうか指定します。

Communications Server は、このバイトの最上位ビット (ビット 0) のみをチェックします。他のビットは無視されます。

dt_set パラメーターの内容によって RUI インターフェースのトレースを使用可能または使用不可にするには、このバイトの最上位ビットを 1 に設定します。

RUI インターフェースのトレースの状態を現在のまま保持するには、このバイトの最上位ビットを 0 に設定します。

reset_trc

トレース・ファイル (1 つまたは複数) をリセットするかどうかを指定します。可能な値は以下のとおりです。

SV_YES トレース・ファイル (1 つまたは複数) をリセットします。ファイルを空にし、現在の内容を廃棄します。

SV_NO トレース・ファイルをリセットしません。

trunc 各トレース・レコードを切り捨てる長さ。切り捨てるを望まない場合は、ゼロを指定します。

戻りパラメーター

verb の実行後、Communications Server は、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。

実行が成功した場合

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

primary_rc

SV_OK

実行が失敗した場合

verb が正常に実行されなかった場合は、Communications Server は、エラーのタイプを示す 1 次戻りコードと、実行が成功しなかった特定の理由を示す 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・エラーのために `verb` が正常に実行されなかった場合は、`Communications Server` は次のパラメーターを戻します。

primary_rc

`SV_PARAMETER_CHECK`

secondary_rc

可能な値は以下のとおりです。

SV_INVALID_SET

dt_set パラメーターに無効な値が入っていました。

SV_INVALID_RESET_TRACE

reset_trc パラメーターに無効な値が入っていました。

その他の状況: 他の状況により、次の 1 次戻りコード (*primary_rc*) が戻される場合があります。

SV_INVALID_VERB

opcode パラメーターがどの `verb` の命令コードとも一致していません。
`verb` は実行されませんでした。

SV_UNEXPECTED_DOS_ERROR

オペレーティング・システムが `verb` の処理中にエラーを検出しました。オペレーティング・システムからの戻りコードは *secondary_rc* を介して戻されます。問題が解消されない場合は、訂正処置についてシステム管理者に相談してください。

オペレーティング・システム戻りコードの意味については、`/usr/include/errno.h` ファイルを参照してください。

SNACTL 環境変数

SNACTL 環境変数は、`DEFINE_TRACE verb` を使用するアプリケーション・プログラムのデバッグ用として `Communications Server` により提供されています。この変数を設定した場合は、プログラムが発行する `DEFINE_TRACE verb` はトレースに作用しません (しかし、エラーが起こらない限り、`SV_OK` が戻ります)。この変数を使用すると、通常はトレースをオフにしているプログラムでトレースを行ったり、通常はトレースをオンにしているプログラムでトレースを行わないようにしたりすることができます。トレース情報およびこの環境変数に関する情報については、「*IBM Communications Server for Linux 診断ガイド*」、または「*IBM Communications Server for AIX 診断ガイド*」を参照してください。

GET_CP_CONVERT_TABLE

`GET_CP_CONVERT_TABLE verb` は、ソース・コード・ページからターゲット・コード・ページへ文字ストリングを変換するために 256 バイトの変換テーブルを作成して戻します。ソース・コード・ページにある文字がターゲット・コード・ページに存在しない場合は、変換された (ターゲット) ストリングは元の (ソース) ストリングと異なります。

GET_CP_CONVERT_TABLE

コード・ページとは、特定の ASCII 値または EBCDIC 値を特定の文字に関連付けるテーブルです。コード・ページは、特定の言語に固有の文字をサポートする各国語版の ASCII または EBCDIC を提供するために使用されます。Communications Server がサポートするコード・ページと、それらを使用する各国語の一覧については、47 ページの『付録 A. コード・ページ』を参照してください。

VCB 構造体

AIX, LINUX

```
typedef struct get_cp_convert_table
{
    AP_UINT16      opcode;          /* Verb identifying operation code. */
    unsigned char  opext;          /* Verb extension code - reserved. */
    unsigned char  reserv2;        /* Reserved. */
    AP_UINT16      primary_rc;     /* Primary return code from verb. */
    AP_UINT32      secondary_rc;   /* Secondary (qualifying) return code. */
    AP_UINT16      source_cp;      /* Source code page for conversion table.*/
    AP_UINT16      target_cp;      /* Target code page for conversion table.*/
    unsigned char  *conv_tbl_addr; /* Address to put conversion table at. */
    unsigned char  char_not_fnd;   /* Character not found option: either */
                                /* substitute character or round trip. */
    unsigned char  substitute_char; /* Substitute character to use. */
};
```

WINDOWS

```
typedef struct get_cp_convert_table
{
    unsigned short opcode;          /* Verb identifying operation code. */
    unsigned char  opext;          /* Verb extension code - reserved. */
    unsigned char  reserv2;        /* Reserved. */
    unsigned short primary_rc;     /* Primary return code from verb. */
    unsigned long  secondary_rc;   /* Secondary (qualifying) return code. */
    unsigned short source_cp;      /* Source code page for conversion table.*/
    unsigned short target_cp;      /* Target code page for conversion table.*/
    unsigned char  *conv_tbl_addr; /* Address to put conversion table at. */
    unsigned char  char_not_fnd;   /* Character not found option: either */
                                /* substitute character or round trip. */
    unsigned char  substitute_char; /* Substitute character to use. */
};
```

指定パラメーター

この verb を使用するプログラムが指定するパラメーターを次に示します。

opcode SV_GET_CP_CONVERT_TABLE

source_cp

ソース・コード・ページ (文字の変換元)。

使用するコード・ページを識別する 10 進数。有効なコード・ページ番号の一覧は、47 ページの『付録 A. コード・ページ』を参照してください。

target_cp

ターゲット・コード・ページ (文字の変換先)。

使用するコード・ページを識別する 10 進数。有効なコード・ページ番号の一覧は、47 ページの『付録 A. コード・ページ』を参照してください。

conv_tbl_addr

256 バイトの変換テーブルが入るバッファのアドレス。

char_not_fnd

ソース・コード・ページにある文字がターゲット・コード・ページに存在しない場合に行うアクションを指定します。

可能な値は以下のとおりです。

SV_ROUND_TRIP

ソース・コード・ページの各文字について、固有値を変換テーブルに保管します。この値は、2 つの同じコード・ページ間で逆方向の変換を行うために 2 つ目の変換テーブルを作成する場合にのみ有効です。両方の変換テーブルの作成で SV_ROUND_TRIP 値を指定した場合は、1 つのコード・ページから別のコード・ページへ変換したあとで元のコード・ページに再変換した文字は、いずれも変更されません。

SV_SUBSTITUTE

変換テーブルに置換文字 (*substitute_char* パラメーターで指定する) を保管します。変換された文字ストリングを元のコード・ページに再変換しても、元の文字ストリングが再作成されるとは限りません。

substitute_char

ソース・コード・ページの文字に対応する文字がターゲット・コード・ページに存在しない場合に、変換テーブルに保管する文字を指定します。

このパラメーターは、*char_not_fnd* パラメーターを SV_SUBSTITUTE に設定した場合にのみ使用します。

ターゲット・コード・ページが EBCDIC コード・ページの場合は、このパラメーターは、実際の文字ではなく、使用したい文字の EBCDIC 値に設定する必要があります。たとえば、ASCII から EBCDIC への変換テーブルで - 文字を置換文字として使用するには、実際の文字 - ではなく、値 60 (EBCDIC で文字 - に関連付けられている値) を指定します。ターゲット・コード・ページが ASCII コード・ページの場合は、文字とその ASCII 値のどちらでも指定できます。

戻りパラメーター

verb の実行後、Communications Server は、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。

実行が成功した場合

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

primary_rc

SV_OK

実行が失敗した場合

verb が正常に実行されなかった場合は、Communications Server は、エラーのタイプを示す 1 次戻りコードと、実行が成功しなかった特定の理由を示す 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・エラーのために verb が正常に実行されなかった場合は、Communications Server は次のパラメーターを返します。

primary_rc

SV_PARAMETER_CHECK

secondary_rc

可能な値は以下のとおりです。

SV_INVALID_CHAR_NOT_FOUND

char_not_fnd パラメーターに無効な値が入っていました。

SV_INVALID_SOURCE_CODE_PAGE

source_cp パラメーターで指定したコード・ページはサポートされていません。

SV_INVALID_TARGET_CODE_PAGE

target_cp パラメーターで指定したコード・ページはサポートされていません。

その他の状況: 他の状況により、次の 1 次戻りコード (*primary_rc*) が戻される場合があります。

WINDOWS

SV_COMM_SUBSYSTEM_NOT_LOADED

Remote API Client ソフトウェアが始動していません。訂正処置についてシステム管理者に相談してください。

SV_INVALID_VERB_SEGMENT

提供された VCB が、データ・セグメントの境界を越えて拡張しました。

■■■■■

SV_INVALID_VERB

opcode パラメーターがどの verb の命令コードとも一致していません。verb は実行されませんでした。

SV_UNEXPECTED_DOS_ERROR

オペレーティング・システムが verb の処理中にエラーを検出しました。オペレーティング・システムからの戻りコードは *secondary_rc* を介して返されます。問題が解消されない場合は、訂正処置についてシステム管理者に相談してください。

AIX, LINUX

オペレーティング・システム戻りコードの意味については、`/usr/include/errno.h` ファイルを参照してください。

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、ご使用のオペレーティング・システムの資料を参照してください。

LOG_MESSAGE

AIX, LINUX

`LOG_MESSAGE verb` は、Communications Server のエラー・ログ・ファイルまたは監査ログ・ファイルにメッセージを記録します。メッセージのテキストはユーザー定義のメッセージ・ファイルから取り出されます。この `verb` を使用して、メッセージに挿入するパラメーターを指定することもできます。

この `verb` を使用する場合は、アプリケーションで使用する適切なメッセージ・ファイルを用意する必要があります。詳しくは、36 ページの『ログ・メッセージ・ファイルの作成』を参照してください。

Communications Server 監査、エラー・ログ・ファイル、ログ・メッセージの形式に関する詳細は、「*IBM Communications Server for Linux 診断ガイド*」、または「*IBM Communications Server for AIX 診断ガイド*」を参照してください。

VCB 構造体

```
typedef struct log_message
{
    AP_UINT16      opcode;           /* Verb identifying operation code. */
    unsigned char opext;           /* Verb extension code - reserved. */
    unsigned char reserv2;        /* Reserved. */
    AP_UINT16      primary_rc;     /* Primary return code from verb. */
    AP_UINT32      secondary_rc;   /* Secondary (qualifying) return code. */
    AP_UINT16      msg_num;        /* Number of message to log. */
    unsigned char  origintr_id[8]; /* ID of the originator of the message. */
    unsigned char  msg_file_name[3]; /* Message file to search for the
                                     /* required message number.
    unsigned char  msg_act;        /* Message action - how to log the msg. */
    AP_UINT16      msg_ins_len;    /* Length of data for insertion in msg. */
    unsigned char  *msg_ins_ptr;   /* Address of data for insertion in msg. */
};
```

指定パラメーター

この `verb` を使用するプログラムが指定するパラメーターを次に示します。

```
opcode SV_LOG_MESSAGE
```

msg_num

msg_file_name で指定したメッセージ・ファイルの中のメッセージの数。

Communications Server ログ・ファイルに示されるメッセージ ID は、Communications Server コンポーネント ID とメッセージ番号の 2 つの部分で構成されます。メッセージ番号は *msg_num* パラメーターで指定されます。この *verb* でログに記録されるメッセージのコンポーネント ID は常に 32,767 です。

orignttr_id

LOG_MESSAGE *verb* を発行するコンポーネントの名前。最大 8 文字のストリングです。このパラメーターはオプションです。組み込まない場合は、最初のバイトを 0x00 に設定してください。

この名前を指定した場合は、Communications Server はそれをメッセージ・テキストに挿入する最初のパラメーターとして使用します。つまり、メッセージ・テキストの中の「%1」がこの名前です。詳しくは、36 ページの『ログ・メッセージ・ファイルの作成』を参照してください。

msg_file_name

ログに記録するメッセージのテキストが入っているファイルの名前。このメッセージ・ファイルの作成方法については、36 ページの『ログ・メッセージ・ファイルの作成』を参照してください。

メッセージ・ファイルには、3 文字のあとに **.msg** という拡張子が付いた名前が必要です。このパラメーターは基本ファイル名のみを指定します。**.msg** 拡張子は自動的に追加されます。

メッセージ・ファイルは、アプリケーションを実行するコンピューター上の **/usr/lib/sna** (AIX) または **/opt/ibm/sna/lib** (Linux) ディレクトリーに保管する必要があります。単一サーバーでの集中ロギングを使用するように Communications Server がセットアップされている場合は、ログ・ファイルを保持するサーバー上の **/usr/lib/sna** にも同じメッセージ・ファイルがなければなりません。

msg_act

メッセージを処理するときに実行されるアクション。これにより、ログ・メッセージのログ・カテゴリー (問題、例外、または監査) が定義されます。ログ・カテゴリーについて詳しくは、「*IBM Communications Server for Linux 診断ガイド*」、または「*IBM Communications Server for AIX 診断ガイド*」を参照してください。可能な値は以下のとおりです。

SV_PROBLEM

問題メッセージとしてのログ

SV_EXCEPTION

例外メッセージとしてのログ

SV_AUDIT

監査メッセージとしてのログ

他の CSV インプリメンテーションとの互換性のために、次の値もサポートされます。これらの値は移行のみを目的としています。これらの値と Communications Server ログ・カテゴリー間のマッピングは近似的なものであるにすぎず、最も適切なカテゴリーが常に得られるとは限らないからで

す。新しいアプリケーションを作成する場合は、値 SV_PROBLEM、SV_EXCEPTION または SV_AUDIT を使用してください。

SV_INTRV, SV_INTRV_16, SV_NO_INTRV_16

SV_PROBLEM と同等

SV_NO_INTRV, SV_NO_INTRV_10

SV_EXCEPTION と同等

SV_NO_INTRV_8, SV_NO_INTRV_6

SV_AUDIT と同等

メッセージ・タイプ SV_EXCEPTION または SV_AUDIT、あるいはそれと同等のものは、現在 Communications Server が該当タイプ (例外または監査) のメッセージを記録するように構成されている場合のみログに記録されます。それ以外の場合は、メッセージは無視されます (しかし、無視された場合でも verb は SV_OK を戻します)。SV_INTRV および SV_NO_INTRV 以外の値は、他の導入 CSV システムではサポートされない場合があります。

msg_ins_len

メッセージに挿入するデータの長さ (0~1000 文字)。データを挿入しない場合は、長さ 0 (ゼロ) を指定してください。

msg_ins_ptr

メッセージに挿入するデータのアドレス。*msg_ins_len* が 0 (ゼロ) の場合は、このパラメーターは無視されます。

データは 1~19 のヌル終止ストリングで構成されます。挿入するデータの合計長は 1000 文字を超えないようにする必要があります。

ログ・メッセージ・ファイルを作成するときは、これらのデータ・ストリングを挿入するメッセージ・テキスト内の位置を指定します。詳しくは、36 ページの『ログ・メッセージ・ファイルの作成』を参照してください。この verb に指定するデータには、メッセージ・テキストに必要なパラメーターごとにストリングを組み込む必要があります。最初のストリングは、このデータ・ストリングではなく *origintr_id* パラメーターで指定することもできます。

戻りパラメーター

verb の実行後、Communications Server は、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戻します。

実行が成功した場合

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

primary_rc

SV_OK メッセージが正常にログに記録されたか、あるいは現在 Communications Server が指定のタイプ (例外または監査) のメッセージを記録するように構成されていないため、メッセージが無視されました。

実行が失敗した場合

verb が正常に実行されなかった場合は、Communications Server は、エラーのタイプを示す 1 次戻りコードと、実行が成功しなかった特定の理由を示す 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・エラーのために verb が正常に実行されなかった場合は、Communications Server は次のパラメーターを戻します。

primary_rc

SV_PARAMETER_CHECK

secondary_rc

次のいずれかです。

SV_INVALID_FIRST_CHARACTER

msg_file_name パラメーターの最初の文字がゼロまたはスペース文字でした。

SV_INVALID_MESSAGE_ACTION

msg_act パラメーターに無効な値が入っていました。

指定されたメッセージ・ファイルが見つからないかオープンできないことを示す SV_PARAMETER_CHECK 2 次戻りコードがありません。このエラーの結果、戻りコード SV_UNEXPECTED_DOS_ERROR が戻される場合があります。

その他の状況: 他の状況により、次の 1 次戻りコード (*primary_rc*) が戻される場合があります。

SV_COMM_SUBSYSTEM_NOT_LOADED

Remote API Client ソフトウェアが起動していません。訂正処置についてシステム管理者に相談してください。

SV_INVALID_VERB

opcode パラメーターがどの verb の命令コードとも一致していません。verb は実行されませんでした。

SV_UNEXPECTED_DOS_ERROR

オペレーティング・システムが verb の処理中にエラーを検出しました。オペレーティング・システムからの戻りコードは *secondary_rc* を介して戻されます。問題が解消されない場合は、訂正処置についてシステム管理者に相談してください。

オペレーティング・システム戻りコードの意味については、`/usr/include/errno.h` ファイルを参照してください。

ログ・メッセージ・ファイルの作成

Communications Server に添付されている `snamsgf` プログラムを使用すると、独自のメッセージ・ファイルを作成して LOG_MESSAGE verb と共に使用することができます。

この機能を使用するには、メッセージ番号とテキストが入ったテキスト・ファイルを作成してから、`snamsgf` を使用してそのファイルをメッセージ・ファイルに変換する必要があります。

メッセージ・ソース・ファイルの形式

メッセージ・ソース・ファイルは、単純な ASCII テキスト・ファイルです。行の最初の文字としてアスタリスク (*) を使用することにより、ファイルのどこにでもコメント行を挿入することができます。Communications Server はこの行の残りのテキストをすべて無視します。

ソース・ファイルの最初の行は「ID:」とし、そのあとにメッセージ・ログに記録するコンポーネントを識別する 1~8 文字の文字ストリングが続く必要があります。このストリングは、ログ・ファイル中の各メッセージの最初に印刷されます。このメッセージ・ファイルを使用するプログラムを識別するストリングを指定してください。たとえば、1 つのアプリケーションのみがこのメッセージ・ファイルを使用する場合はそのアプリケーションの名前を指定します。あるいは、このメッセージ・ファイルを使用する 1 グループのアプリケーションを識別するストリングを指定します。

メッセージ・ソース・ファイルの残りの部分は、個別のメッセージのエントリです。各メッセージは、次の例に示すように、一連のフィールドとして定義されます。

```
ID:MYAPPL
Message:      1
Type:         PROBLEM
Cause Type:   CSV
Cause:        The specified file could not be opened.
Action:       Check the reason shown on this message for more information.
Flags:        NONE
String:       Could not open the file.$
Filename = %1\nReason = %2
```

各フィールドは次のとおりです。

Message (メッセージ)

メッセージの固有の ID (1~65,535 の範囲の 10 進数)。ファイル中のメッセージは、メッセージ番号の昇順にリストする必要があります。番号を連続させる必要はありませんが、未使用のメッセージ番号が多くなると、メッセージ・ファイルのサイズが増大します。

Type (タイプ)

ログ・メッセージのカテゴリ。PROBLEM、EXCEPTION、または AUDIT を指定します。メッセージをログに記録するときに Communications Server が実際に使用するカテゴリは、LOG_MESSAGE verb の *msg_act* パラメーターで決まります。ソース・ファイルでは、この情報は参考用で、Communications Server は無視します。

Cause type (原因のタイプ)

メッセージの原因の要約。CSV を指定するか (メッセージが CSV LOG_MESSAGE verb を使用してログに記録されたことを示す場合)、次のいずれかの値を指定します。

Internal

アプリケーションの内部エラー。

Resource

リソースの不足 (たとえば、AIX / Linux コンピューターのメモリー不足)。

- User** ユーザー・エラー (たとえば、コマンド行でアプリケーションに指定したパラメーターが無効な場合)。
- SNA** リモート・システムによる SNA プロトコル違反。
- Config** 構成の不一致。
- Audit** 正常なイベント。通知のためにのみ報告されます。

Cause (原因)

ログに記録された状況の原因。

Action (アクション)

メッセージの結果としてローカル・システムの管理者がとるべき処置。監査メッセージの場合は、エラー状態の報告ではなくアカウント情報と進行情報が提供されるため、通常は対処する必要はありません。

Flags (フラグ)

メッセージをログ・ファイルだけでなく AIX / Linux コンピューターのシステム・コンソールにも書き出す場合は、**CONSOLE** を指定します。メッセージをログ・ファイルのみに書き出す場合は、**NONE** を指定します。

String (ストリング)

メッセージのテキスト (1~256 文字)。LOG_MESSAGE verb に指定されたパラメーターを組み込むには、%1、%2 などを使用して、各パラメーターの位置を示します。Communications Server は、メッセージをログに記録するときに、%1 を LOG_MESSAGE に指定された最初のパラメーターに、%2 を 2 番目のパラメーターに、以下同様にそれぞれ置き換えます。

LOG_MESSAGE に指定された *origintr_id* パラメーターがある場合は、このパラメーターが %1 に置き換わります。LOG_MESSAGE に指定されたデータ・ストリング中の最初のパラメーターは %2 (*origintr_id* が使用されている場合) または %1 (*origintr_id* が使用されていない場合) に置き換わります。データ・ストリング中の 2 番目のパラメーターは %3 または %2 に置き換わります。以下同様に続きます。

次のこともこれらのフィールドに適用されます。

- 各フィールド名 (「Message (メッセージ)」など) は行の始めに置き、そのあとにコロンを付ける必要があります。コロンのあとのスペースまたはタブは無視されます。フィールド名に関連付けられたテキストは、すべて 1 行に収める必要があります (後述の \$ 文字を使用して行を連結する場合を除きます)。行の長さには制限はありません。
- 「Cause (原因)」、「Action (アクション)」、「String (ストリング)」の各フィールドでは、次の文字を使用して、ログ・ファイルに書き出すテキストの形式を制御することができます。

\t 出力テキストにタブ文字を挿入します。

\$ (ソース・テキストで改行文字の前に置かれ)

出力テキストに改行文字を挿入し、ソース・ファイルの次の行を続けます。これにより、1 行に収まらないテキスト・フィールドを指定できます。テキスト・フィールドの最後の行の末尾に \$ 文字を置くことはできません。

\n 出力テキストに改行文字を挿入し、ソース・ファイルの次の文字を続けま

す。これにより、ソース・ファイルでテキスト・フィールドを単一の行として指定することができ、出力で行が分割される位置を指定できます。ただし、読みやすくするためには、上述のように \$ 文字を使用して長いテキスト・フィールドを複数の行に分割することを推奨します。

\\$ 出力テキストに \$ 文字を挿入します。

%n (「String (ストリング)」 パラメーターの場合のみ)

出力テキストに、ログ・コールに指定された n 番目のパラメーターを挿入します。

ログ・コードでは、ソース・テキストで n 文字または \$ 文字が組み込まれている場合を除いて、テキスト・ストリングに改行文字を挿入しません。80 桁の画面で出力テキストを容易に読めるようにするためには、これらの文字を使用して行を分割させます。

- フィールド「Message (メッセージ)」、「Type (タイプ)」、「Cause Type (原因のタイプ)」、「Flags (フラグ)」、「String (ストリング)」は必ず指定する必要があります。フィールド「Cause (原因)」および「Action (アクション)」はオプションです。これらのフィールドのいずれかを使用しない指示をするには、頭文字に大文字を使用した次のストリングを指定します。

@!* Not Used

たとえば、メッセージが監査メッセージであって、訂正処置が不要な場合は、次を使用します。

```
Action: @!* Not Used
```

この場合、ログ・ファイルにメッセージを書き込むとき、Communications Server は「Action (アクション)」フィールドを組み込みません。

- 「Cause (原因)」フィールドおよび「Action (アクション)」フィールドの長さは、2048 文字を超えてはいけません。

ログ・メッセージ出力のサンプル

前節では、メッセージ・ソース・ファイルのエントリーのサンプルを示しました。このサンプル・エントリーが組み込まれたソース・ファイルからメッセージ・ファイルを作成した場合は、このメッセージ・ファイル中でメッセージ番号 1 を指定された LOG_MESSAGE をアプリケーションから呼び出せます。アプリケーションの指定データには 2 つのヌル終止ストリングを入れる必要があります。1 つはファイル名 (たとえば、`/usr/jim/myfile`) を指定し、もう 1 つは失敗の理由 (たとえば、「File not found」) を指定します。その場合、出力は次のようになります。

```
-----12:17:28 BST 05/13/1994 -----
MYAPPL      Message 32767-1, Subcode: 0
Log category: PROBLEM Cause Type: CSV
System:     jimsbox
Process ID: 12345
Could not open the file.
Filename = /usr/jim/myfile
Reason  = File not found
Cause:    The specified file could not be opened.
Action:   Check the reason shown on this message for more information.
```

このサンプル出力では、verbose ログ (簡略ログではなく) を使用するものと想定しています。簡略ログの詳細および、簡略ログを使用する場合のログ・ファイル中の

エントリーの形式については、「*IBM Communications Server for Linux 診断ガイド*」、または「*IBM Communications Server for AIX 診断ガイド*」のログ・メッセージについて説明している章を参照してください。

テキスト・ファイルからのメッセージ・ファイルの作成

テキスト・ファイルをメッセージ・ファイルに変換するには、次のように **snamsgf** プログラムを使用します。

```
snamsgf infile outfile
```

入力テキスト・ファイルの名前は *infile* です。現行ディレクトリー以外の場所にある場合はパス名を組み込みます。

出力メッセージ・ファイルの名前は *outfile* です。これは LOG_MESSAGE の *msg_file_name* パラメーターで指定されているものです。出力ファイルには、1~3文字と拡張子 **.msg** から成る名前が必要です。コマンド行では拡張子を指定する必要はありません。

出力ファイルは現行ディレクトリー内に作成されます。LOG_MESSAGE verb で指定されたときに Communications Server が見付けられるように、このファイルはディレクトリー */usr/lib/sna* (AIX) または */opt/ibm/sna* (Linux) に保管する必要があります。

たとえば、次のコマンドでは、ソース・テキスト・ファイル */usr/fred/myfile.text* からメッセージ・ファイル **new.msg** が作成されます。

```
snamsgf /usr/fred/myfile.text new
```

snamsgf プログラムは、入力ファイル形式にエラーを検出すると、標準エラーとしてエラー・メッセージを書き出します。



TRANSFER_MS_DATA

WINDOWS

TRANSFER_MS_DATA verb は、ネットワーク管理ベクトル・トランスポート (NMVT) データを含む要求単位 (RU) をビルドします。この verb は NMVT データを NetView に送信でき、それによって問題の診断および解明を集中化することができます。このデータは、ローカルのエラー・ログ・ファイルにも記録されます。

アプリケーションは、送信用の完全な NMVT を提供できます。あるいは、必要なサブベクトルをいくつか提供して、ヘッダー情報または追加サブベクトルを追加するように、Communications Server に要求することもできます。Communications Server が追加するヘッダーおよびサブベクトルのフォーマットを含めて、NMVT のフォーマットについて詳しくは、「*IBM Systems Network Architecture: Formats*」を参照してください。

VCB 構造体

```
typedef struct transfer_ms_data
{
    unsigned short    opcode;           /* Verb operation code          */
    unsigned char     data_type;        /* Type of data supplied by appl */
    unsigned char     reserv2;         /* reserved                     */
    unsigned short    primary_rc;      /* Primary return code          */
    unsigned long     secondary_rc;    /* Secondary return code        */
    unsigned char     options;         /* Verb options                 */
    unsigned char     reserv3;         /* reserved                     */
    unsigned char     originator_id[8]; /* Originator ID                */
    unsigned short    dlen;            /* Length of data               */
    unsigned char     *dptr;           /* Data                         */
} TRANSFER_MS_DATA;
```

指定パラメーター

この verb を使用するプログラムが指定するパラメーターを次に示します。

opcode SV_TRANSFER_MS_DATA

data_type

可能な値は以下のとおりです。

SV_NMVT

データには、完全な NMVT が含まれます。

SV_ALERT_SUBVECTORS

データには、アラート主ベクトルのための MS サブベクトルが SNA 定義のフォーマットで含まれます。Communications Server は、NMVT ヘッダーおよびアラート主ベクトル・ヘッダーを追加します。

SV_USER_DEFINED

データには、完全な NMVT 要求単位が含まれます。Communications Server は常に、そのデータをログに記録し、NetView には送信しません。

SV_PDSTATS_SUBVECTORS

データには、問題判別統計が含まれます。Communications Server は常に、そのデータをログに記録し、NetView には送信しません。

options このパラメーターは 1 バイトの値です。個々のビットが、選択されたオプションを示します。ビット 0 が最上位ビットで、ビット 7 が最下位ビットです。他のインプリメンテーションと互換性を持たせるため、ビット 0 から 3 のビット値は、値 1 が「アクションなし」を、値 0 が「アクション」を示すように定義されています。(data_type が SV_USER_DEFINED に設定されている場合、ビット 1 から 3 は無視されます。)

ビット 0 — 日付/時刻 (0x01) サブベクトルをデータに追加します。

- サブベクトルを追加するように Communications Server に要求する場合は、このビットを 0 に設定する。
- サブベクトルを追加しないように Communications Server に要求する場合は、このビットを 1 に設定する。

ビット 1 — 製品セット ID (0x10) サブベクトルをデータに追加します。すでに製品セット ID サブベクトルを含むデータをアプリケーションが提供

している場合、Communications Server は、既存のサブベクトルの直前に先行させて、自分の製品セット ID サブベクトルを追加します。

- サブベクトルを追加するように Communications Server に要求する場合は、このビットを 0 に設定する。
- サブベクトルを追加しないように Communications Server に要求する場合は、このビットを 1 に設定する。

ビット 2 — データを NetView に送信します。

- データを送信するように Communications Server に要求する場合は、このビットを 0 に設定する。
- データを送信しないように Communications Server に要求する場合は、このビットを 1 に設定する。

data_type が SV_USER_DEFINED または SV_PDSTATS_SUBVECTORS に設定されている場合、このビットは無視されます。データを NetView に送信することはできません。

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。

- データをログに記録するように Communications Server に要求する場合は、このビットを 0 に設定する。
- データをログに記録しないように Communications Server に要求する場合は、このビットを 1 に設定する。

data_type が SV_USER_DEFINED または SV_PDSTATS_SUBVECTORS に設定されている場合、このビットは無視されます。データは常にログに記録されます。

ビット 4 から 7 は予約済みで、0 に設定しなければなりません。

originator_id

verb を発行したコンポーネントの名前。データが Communications Server エラー・ログ・ファイルにログ記録されている場合、この名前を使用してログ・メッセージの発生元を示します。データがログに記録されていない場合は、この名前は無視されます。

これは最大 8 文字の ASCII 文字列で、ローカルでの表示可能文字を使用しています。パラメーターはオプションです。組み込まない場合は、最初の文字を 0x00 に設定してください。

dlen アプリケーションが提供するデータの長さ。

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 を提供する場合は、データ長が 512 バイトを超えてはいけません。アプリケーションがアラート・サブベクトルを提供している場合、あるいは提供データに 1 つ以上のサブベクトルを追加するように Communications Server に要求している場合は、必要なヘッダーまたはサブベクトルあるいはその両方を追加した後の合計長が 512 バイトを超えてはいけません。

dptr アプリケーションが提供するデータ・ストリングを指すポインター。データは、*data_type* パラメーターで指定されているとおり、NMVT、アラート・サブベクトル、または問題判別統計で有効なフォーマットでなければなりません。

戻りパラメーター

verb の実行後、Communications Server は、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。

実行が成功した場合

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

primary_rc
SV_OK

secondary_rc
使用しない。

実行が失敗した場合

verb が正常に実行されなかった場合は、Communications Server は、エラーのタイプを示す 1 次戻りコードと、実行が成功しなかった特定の理由を示す 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・エラーのために verb が正常に実行されなかった場合は、Communications Server は次のパラメーターを戻します。

primary_rc
SV_PARAMETER_CHECK

secondary_rc
可能な値は以下のとおりです。

SV_INVALID_DATA_TYPE
提供された *data_type* パラメーターが、無効な値でした。

SV_INVALID_DATA_SEGMENT
提供されたデータ・ストリングが、データ・セグメントの境界を越えて拡張しました。

SV_DATA_EXCEEDS_RU_SIZE
以下のいずれかが発生しました。

- アプリケーションが、NMVT の最大長である 512 バイトより長いデータ・ストリングを提供しました。
- アプリケーションがアラート・サブベクトルとしてデータを提供したか、あるいは Communications Server によって 1 つ以上のサブベクトルを追加するように指定しましたが、追加されたヘッダーまたはサブベクトルあるいはその両方のデータ・サイズが 512 バイトを超えてしまいました。

状態チェック: 状態エラーのために verb が正常に実行されなかった場合は、Communications Server は次のパラメーターを戻します。

primary_rc
SV_STATE_CHECK

secondary_rc

SV_SSCP_PU_SESSION_NOT_ACTIVE

アプリケーションが *options* パラメーターで SV_SEND を指定しましたが、該当する PU とのセッションがアクティブではありませんでした。

その他の状況: 他の状況により、次の 1 次戻りコード (*primary_rc*) が戻される場合があります。

primary_rc

SV_CANCELLED

この verb (非同期エントリー・ポイントを使用して発行されたもの) がまだアウトスタンディングである間に、WinCSVCleanup 呼び出しが発行されました。この verb は取り消されました。データは送信されていない可能性があります。

primary_rc

SV_COMM_SUBSYSTEM_NOT_LOADED

Remote API Client ソフトウェアが開始されていなかったか、あるいは停止されていました。

primary_rc

SV_INVALID_VERB

opcode パラメーターがどの verb の命令コードとも一致していません。verb は実行されませんでした。

primary_rc

SV_INVALID_VERB_SEGMENT

提供された VCB が、データ・セグメントの境界を越えて拡張しました。

primary_rc

SV_SERVER_RESOURCE_NOT_FOUND

必要な Communications Server コンポーネントがアクティブになっていませんでした。データを送信できません。

primary_rc

SV_SERVER_RESOURCES_LOST

必要な Communications Server リソースが使用可能になっていませんでした。

secondary_rc

SV_SERVER_COMM_FAILURE

必要な Communications Server コンポーネントへの通信パスに障害がありました。データを送信できませんでした。

primary_rc

SV_THREAD_BLOCKING

同期 CSV エントリー・ポイントを使用して verb が発行されましたが、このアプリケーションに対して同期 verb がすでに進行中です。一時点で進行できる同期 verb は 1 つだけです。

primary_rc

SV_UNEXPECTED_DOS_ERROR

オペレーティング・システムが *verb* の処理中にエラーを検出しました。オペレーティング・システムからの戻りコードは *secondary_rc* を介して戻されます。問題が解消されない場合は、訂正処置についてシステム管理者に相談してください。

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、ご使用のオペレーティング・システムの資料を参照してください。

この戻りコードは、*verb* を発行したアプリケーションが Windows 機能の `PostMessage` ではなく `SendMessage` を使用して起動されたことを示している場合もあります。この状態では、アプリケーションはいかなる *verb* も発行できません。詳しくは、14 ページの『Windows での考慮事項』を参照してください。



付録 A. コード・ページ

この付録では、GET_CP_CONVERT_TABLE verb 用に、Communications Server でサポートされるコード・ページと、それらのコード・ページを使用する各国語版の ASCII または EBCDIC のリストを示します。

ASCII コード・ページ

AIX, LINUX

8859 ISO 8859 で定義された汎用 ASCII コード・ページ。すべての言語版の ASCII のサポートに使用されます。

-
- 437** 米国英語
 - 737** ギリシャ
 - 813** ギリシャ
 - 819** ANSI (米国規格協会)
 - 850** インターナショナル・コード・ページ (米国英語、英国英語、フランス語、ドイツ語、イタリア語、スペイン語、フィンランド語、オランダ語、スウェーデン語のほか、スイス、ベルギー、ラテン・アメリカで使用される各言語)
 - 852** ポーランド、ハンガリー、ルーマニア、スロバキア、チェコ、クロアチア、スロベニア
 - 855** ブルガリア、セルビア・モンテネグロ、マケドニア共和国
 - 857** トルコ
 - 858** マルチリンガル
 - 860** ポルトガル語
 - 861** アイスランド
 - 862** ヘブライ語
 - 863** カナダ・フランス語
 - 864** アラビア語
 - 865** デンマーク語、ノルウェー語
 - 866** ロシア
 - 869** ギリシャ
 - 874** タイ
 - 897** 日本

ASCII コード・ページ

903	中華人民共和国
912	ポーランド、ハンガリー、ルーマニア、スロバキア、チェコ、クロアチア、スロベニア
915	ロシア、ブルガリア、セルビア・モンテネグロ、マケドニア共和国
916	ヘブライ語
920	トルコ
921	ラトビア、リトアニア
922	エストニア
923	ANSI (米国規格協会)
1008	アラビア語
1041	日本
1088	韓国
1089	アラビア語
1114	中華人民共和国 (台湾)
1115	中華人民共和国
1124	ウクライナ
1125	ウクライナ
1126	韓国
1127	アラビア語
1129	ベトナム
1131	ベラルーシ
1133	ラオス
1250	ポーランド、ハンガリー、ルーマニア、スロバキア、チェコ、クロアチア、スロベニア
1251	ロシア、ブルガリア、セルビア・モンテネグロ、マケドニア共和国
1252	米国 / マルチリンガル
1253	ギリシャ
1254	トルコ
1255	ヘブライ語
1256	アラビア語
1257	バルト語
1258	ベトナム

EBCDIC コード・ページ

037	米国英語、カナダ・バイリンガル、オランダ語、ポルトガル語
273	ドイツ語

275	ブラジル
277	デンマーク語、ノルウェー語
278	フィンランド語、スウェーデン語
280	イタリア語
284	スペイン語、ラテン・アメリカ
285	英国英語
290	日本
297	フランス語
420	アラビア語
424	ヘブライ語
500	ベルギー (新)、スイス・フランス語、スイス・ドイツ語
803	アラビア語
833	韓国
836	中華人民共和国
838	タイ
870	ポーランド、ハンガリー、ルーマニア、スロバキア、チェコ、クロアチア、スロベニア
871	アイスランド
875	ギリシャ
924	米国、カナダ (フランス語圏)、オランダ、ポルトガル、フランス、フィンランド
1025	ロシア、ブルガリア、セルビア・モンテネグロ、マケドニア共和国
1026	トルコ
1027	日本
1047	米国、カナダ (フランス語圏)、オランダ、ポルトガル
1112	ラトビア、リトアニア
1122	エストニア
1123	バルト語
1130	ベトナム
1132	ラオス
1140	米国、カナダ (フランス語圏)、オランダ、ポルトガル
1141	ドイツ、オーストリア
1142	デンマーク、ノルウェー
1143	フィンランド、スウェーデン
1144	イタリア
1145	ラテンアメリカ、スペイン

EBCDIC コード・ページ

- 1146 英国
- 1147 フランス
- 1148 ベルギー、スイス (フランス語圏)、スイス (ドイツ語圏)

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。

〒106-8711
東京都港区六本木 3-2-12
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。

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せんが、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にお願いします。

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。

COPYRIGHT LICENSE: 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。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

お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。® Copyright IBM Corp. 2000, 2005, 2006, 2007, 2008, 2009. All rights reserved.

商標

IBM、IBM ロゴ、および ibm.com は、International Business Machines Corp. の商標または登録商標であり、世界中の準拠法に登録されています。他の製品名およびサービス名は、IBM または他の会社の商標の可能性があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtml にある「Copyright and trademark information」をご覧ください。

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

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

Java およびすべての Java 関連の商標およびロゴは Sun Microsystems, Inc. の米国およびその他の国における商標です。

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

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。

- 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。その他の資料については、タイトルおよび資料番号のみをここに記しています。

IBM Communications Server for AIX 資料

IBM Communications Server for AIX ライブラリーは、以下の資料により構成されています。なお、これらの資料のソフトコピー版が CD-ROM で提供されています。CD-ROM 上のソフトコピー・ファイルにアクセスする方法については、「*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 CS/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情報が記載されています。

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

この資料には、SNA および IBM Communications Server for AIX のコマンドに関する情報が記載されています。

- *IBM Communications Server for Linux or AIX 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 Linux or AIX 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 Linux or AIX 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 Linux or AIX 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、共通サービス 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 or AIX 管理サービス プログラマーズ・ガイド* (SC88-5829)

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

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

この資料には、IBM Communications Server for AIX で使用される APPC アプリケーションに関する情報が記載されています。

- *IBM Communications Server for 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 資料

IBM Communications Server for Linux ライブラリーは、以下の資料により構成されています。なお、これらの資料のソフトコピー版が CD-ROM で提供されています。CD-ROM 上のソフトコピー・ファイルにアクセスする方法については、「*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

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、以下の 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情報が記載されています。

- *IBM Communications Server for Linux 管理コマンド解説書* (SC88-9998)

この資料には、SNA および IBM Communications Server for Linux のコマンドに関する情報が記載されています。

- *IBM Communications Server for Linux or AIX 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 Linux or AIX 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 Linux or AIX 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 Linux or AIX 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、共通サービス 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 or AIX 管理サービス プログラマーズ・ガイド* (SC88-5829)

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

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

この資料には、IBM 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が記載されています。

- *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* (邦文番号 N:GA23-0059; 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* (邦文番号 SC88-7217; 英文番号 SC26-4399)
- *Communications Server for OS/2 Version 4 Application Programming Guide* (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]

エラー・ログ・ファイルへのメッセージの記録 33
エンタリー・ポイント, CSV 用の 2

[カ行]

監査ログ・ファイルへのメッセージの記録 33
関数コール, CSV 用の 2
記号定数, 16 進値の 17
コード・ページの変換 29
コンパイルおよびリンク 14

[サ行]

サンプルのタイプ G 変換テーブル 23
サンプル・コード 11
消去, トレース・ファイルの 23

[タ行]

トレース
共通サービス Verb 27
ファイル 23
ファイルの消去 23
ファイルのリセット 28
APPC 27
CPI-C 27
LUA 28
MS 28
NOF 27
RUI 28
トレース・ファイル 26

[ハ行]

ブロッキング, verb, ウィンドウ 6
変換テーブル
タイプ G, 作成 22
A 19
AE 19
G 19, 23

変換テーブルの作成 29
ホスト NetView プログラムへのデータの送信 40

[マ行]

マルチスレッド・アプリケーション 12
メッセージ・ファイル, LOG_MESSAGE verb 34
文字の変換, ASCII から EBCDIC へ 18

[ラ行]

ログ・メッセージ・ファイルの作成 36

A

acssvcc.h ヘッダー・ファイル 2
ACSSVC, ACSSVC_C, ACSSVC_P エントリー・ポイント 2
ACSSVC_C 呼び出し 4
AIX アプリケーション
コンパイルおよびリンク 13
AIX アプリケーションのコンパイル 13
AIX アプリケーションのリンク 13
API トレース 23
ASCII から EBCDIC への文字変換 18

C

CONVERT
指定パラメーター 18
タイプ G 変換テーブルの作成 22
変換エラー 21
文字セット (A, AE, または G) 19
戻りパラメーター 20
ASCII から EBCDIC へ 19
EBCDIC から ASCII へ 19
VCB 18
verb 18
COPY_TRACE_TO_FILE
指定パラメーター 24
状態チェック 25
ファイルの上書き 24
戻りパラメーター 24, 25
VCB 24
verb 23
CSV エントリー・ポイント
ウィンドウ 3, 6

D

DCE スレッド 12
DEFINE_TRACE
共通サービス Verb 27
切り捨て 28
指定パラメーター 27
使用可能化または使用不可化 27
トレース・ファイルのリセット 28
戻りパラメーター 28
APPC 27
CPI-C 27
LUA 28
MS 28
NOF 27
RUI 28
VCB 27
verb 26

E

EBCDIC から ASCII への文字変換 18

G

GetCsvReturnCode 呼び出し 9
GET_CP_CONVERT_TABLE
指定パラメーター 30
タイプ G 変換テーブルの作成 22
戻りパラメーター 31, 32
VCB 30
verb 29

L

Linux アプリケーション
コンパイルおよびリンク 13
Linux アプリケーションのコンパイル 13
Linux アプリケーションのリンク 13
LOG_MESSAGE
指定パラメーター 33
メッセージへのテキストの挿入 35
メッセージ・ファイル名 34
戻りパラメーター 35, 36
ログ・カテゴリ 34
VCB 33
verb 33

S

snamsgf ユーティリティ 36

T

TRANSFER_MS_DATA 40

指定パラメーター 41

戻りパラメーター 43

VCB 41

V

verb 制御ブロック 2

W

WinAsyncCSV 呼び出し 7

WinCSVCleanup 呼び出し 9

WinCSVStartup 呼び出し 5

Windows での考慮事項 14



プログラム番号: 5765-E51

Printed in Japan

SC88-5824-00



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