
OpenTP1 Version 7

分散トランザクション処理機能

OpenTP1 **プロトコル**

TP1/NET/X25-Extended **編**

解説・手引・文法・操作書

3000-3-D82

マニュアルの購入方法

このマニュアル，および関連するマニュアルをご購入の際は，
巻末の「ソフトウェアマニュアルのサービス ご案内」をご参
照ください。

HITACHI

対象製品

・適用 OS : AIX 5L V5.1 , AIX 5L V5.2 , AIX 5L V5.3

P-1M64-3131 uCosminexus TP1/Message Control 07-00

P-1M64-3231 uCosminexus TP1/NET/Library 07-00

P-F1M64-3231R uCosminexus TP1/NET/X25-Extended 07-00

これらのプログラムプロダクトのほかにも、このマニュアルをご利用になれる場合があります。詳細は「リリースノート」でご確認ください。

これらの製品は、ISO9001 および TickIT の認証を受けた品質マネジメントシステムで開発されました。

輸出時の注意

本製品を輸出される場合には、外国為替および外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認の上、必要な手続きをお取りください。

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

AIX は、米国における米国 International Business Machines Corp. の登録商標です。

UNIX は、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。

発行

2006 年 11 月 (第 1 版) 3000-3-D82

著作権

All Rights Reserved. Copyright (C) 2006, Hitachi, Ltd.

はじめに

このマニュアルは、TP1/NET/X25-Extended を使用した AP 間通信システムの作成方法、および運用方法について説明したものです。

本文中に記載されている製品のうち、このマニュアルの対象製品ではない製品については、OpenTP1 Version 7 対応製品の発行時期をご確認ください。

対象読者

OpenTP1 システムのシステム管理者、システム設計者、およびプログラムを対象としています。また、次のマニュアルを理解されていることを前提としています。

- OpenTP1 解説 (3000-3-D50)
- OpenTP1 プログラム作成の手引 (3000-3-D51)
- OpenTP1 システム定義 (3000-3-D52)
- OpenTP1 運用と操作 (3000-3-D53)
- OpenTP1 プログラム作成リファレンス C 言語編 (3000-3-D54)
- OpenTP1 プログラム作成リファレンス COBOL 言語編 (3000-3-D55)

マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 概要

TP1/NET/X25-Extended を使用した AP 間通信の概要と形態、およびソフトウェア構成について説明しています。

第 2 章 機能

TP1/NET/X25-Extended を使用した場合の、コネクション、コネクショングループ、論理端末の扱い、メッセージ送受信の流れなどについて説明しています。

第 3 章 メッセージ送受信インタフェース

言語別のメッセージ送受信インタフェースについて説明しています。

第 4 章 ユーザOWNコーディング、MCF イベントインタフェース

ユーザOWNコーディング、および MCF イベントインタフェースについて説明しています。

第 5 章 システム定義

TP1/NET/X25-Extended に固有の定義、および通信管理との関係について説明しています。

第 6 章 運用コマンド

TP1/NET/X25-Extended で使用する運用コマンドについて説明しています。

第 7 章 組み込み方法

TP1/NET/X25-Extended を OpenTP1 システムへ組み込む方法について説明しています。

はじめに

第 8 章 障害対策

TP1/NET/X25-Extended の障害時の処理とユーザの対策について説明しています。

付録 A メッセージ送受信の処理の流れ

メッセージを送受信するときのデータの流れ, ジャーナル取得のタイミングについて説明しています。

付録 B 障害発生時の処理の流れ

障害が発生した場合の処理の流れについて説明しています。

付録 C 理由コード一覧

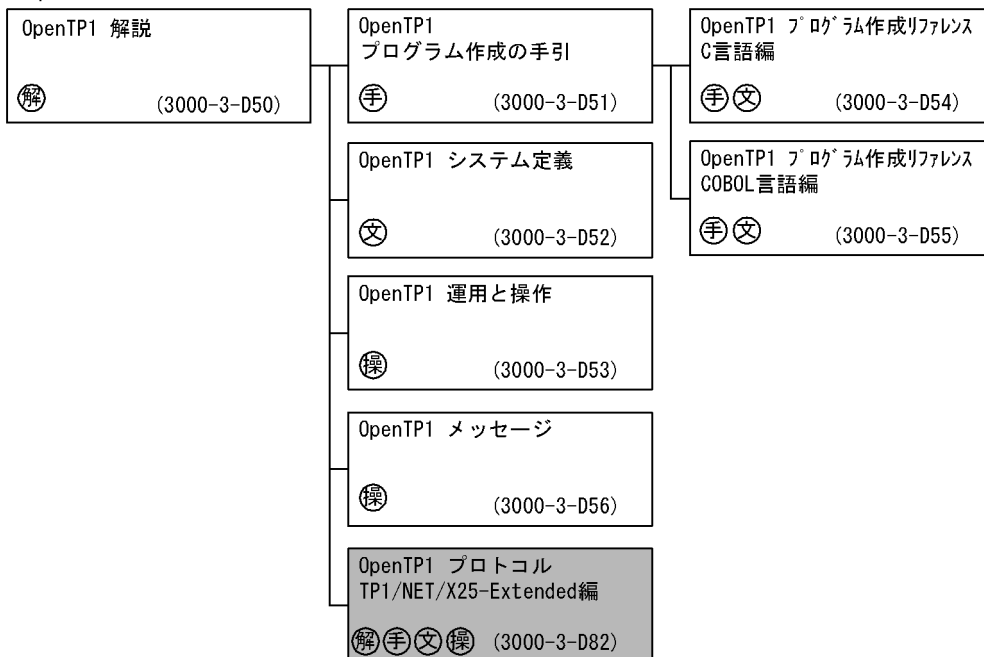
ERREVT2 通知時と障害通知イベント通知時の理由コードについて説明しています。

付録 D 用語解説

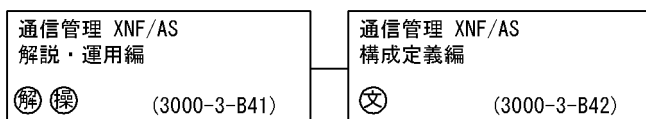
TP1/NET/X25-Extended で使用する用語について説明しています。

関連マニュアル

●OpenTP1 Version 7



●通信管理

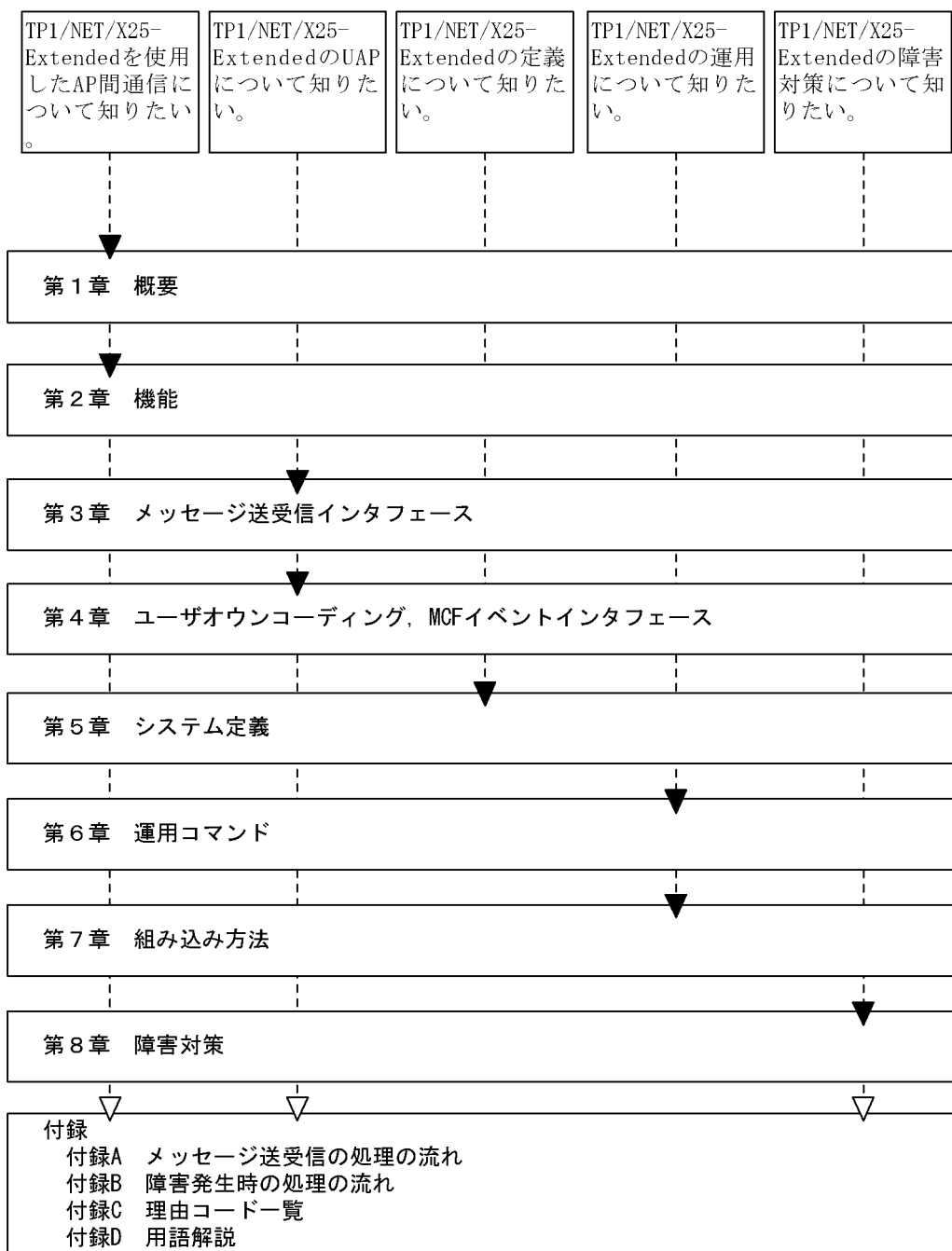


<記号>

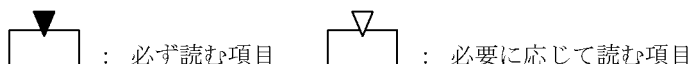
- 解 : 解説書
- 手 : 手引書
- 文 : 文法書
- 操 : 操作書

読書手順

このマニュアルは、利用目的に合わせて章を選択して読むことができます。利用目的別に、次の流れに従ってお読みいただくことをお勧めします。



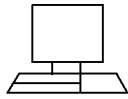
(凡例)



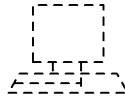
図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。

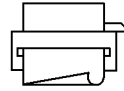
●ワークステーション, 端末



●論理端末



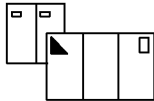
●端末プリンタ



●入出力の動作



●ホストコンピュータ



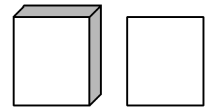
●プログラムの流れ



●ファイル



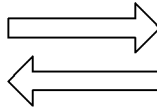
●プログラム



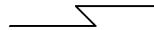
●ネットワーク



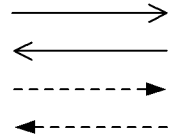
●データの流れ



●通信回線



●制御の流れ



文法の記号

このマニュアルで使用する各種の記号を説明します。

(1) 文法記述記号

文法の記述形式を示す記号です。

文法記述記号	意味
[]	この記号で囲まれている項目は省略できることを示します。 (例) [-s MCF 通信プロセス識別子] -s オプションとそのオペランドを指定するか、何も指定しないことを示します。
 (ストローク)	この記号で区切られた項目は選択できることを示します。 (例) -t int rsp -t オプションに int または rsp を指定できることを示します。 ただし、C 言語のインタフェースの説明でこの記号を使用した場合は、C 言語の文法規則に従います。
{ }	この記号で囲まれている複数の項目のうちから一つを選択できることを示します。 (例) { DCMCFESI DCMCFEMI } DCMCFESI と DCMCFEMI のうち、どちらかを指定できることを示します。

文法記述記号	意味
— (下線)	この記号で示す項目は、オペランド、オプションまたはコマンド引数を省略した場合の省略時解釈値を示します。 (例) <code>-i auto <u>manual</u></code> -i オプションを省略した場合、 <code>manual</code> を省略時解釈値とすることを示します。ただし、データ操作言語の説明の場合、この下線記号で示す予約語は、必要語なので省略できないことを示します。 下線がない予約語は、補助語なので書いても書かなくてもかまいません。
...	この記号で示す直前の一つの項目を繰り返し指定できることを示します。ただし、項目が括弧で囲まれている場合、括弧全体が一つの項目となります。
(白三角)	空白を示します。 (例) <code>コネクション ID1 コネクション ID2</code> コネクション ID1 とコネクション ID2 の間に、空白を 1 個入力することを示します。

(2) 属性表示記号

ユーザ指定値の範囲などを示す記号です。

属性表示記号	意味
~	この記号の後にユーザ指定値の属性を示します。
《 》	ユーザが指定を省略したときの省略時解釈値を示します。
	ユーザ指定値の構文要素を示します。
(())	ユーザ指定値の指定範囲を示します。

(3) 構文要素記号

ユーザ指定値の内容を示す記号です。

構文要素記号	意味
<英字>	アルファベット (A ~ Z, a ~ z) と _ (アンダスコア)
<英字記号>	アルファベット (A ~ Z, a ~ z) と #, @, ¥
<英数字>	英字と数字 (0 ~ 9)
<英数字記号>	英字記号と数字 (0 ~ 9)
<符号なし整数>	数字列 (0 ~ 9)
<10進数>	数字 (0 ~ 9)
<16進数字>	数字 (0 ~ 9) とアルファベット (A ~ F, a ~ f)
<識別子>	先頭がアルファベットの英数字列
<記号名称>	先頭が英字記号の英数字記号列
<文字列>	任意の文字の配列
<パス名>	記号名称, /, および . (ピリオド) (ただし、パス名は使用する OS に依存します)

このマニュアルでの表記

(1) 製品名

このマニュアルで使用する製品名称の略称を次に示します。

製品名称	略称
AIX 5L V5.1	AIX
AIX 5L V5.2	
AIX 5L V5.3	
uCosminexus TP1/Message Control	TP1/Message Control
uCosminexus TP1/NET/Library	TP1/NET/Library
uCosminexus TP1/NET/X25-Extended	TP1/NET/X25-Extended

(2) JIS コード配列のキーボードと ASCII コード配列のキーボードとの違いについて

JIS コード配列と ASCII コード配列では、次に示すコードで入力文字の違いがあります。このマニュアルの文字入力例（コーディング例）の表記は、JIS コード配列（日本語のキーボード）に従った文字に統一しています。

コード	JIS コード配列	ASCII コード配列
(5c) ₁₆	'¥' (円記号)	' \' (バックスラッシュ)
(7e) ₁₆	' ' (オーバーライン)	' ~ ' (チルド)

略語一覧

このマニュアルで使用する英略語の一覧を次に示します。

英略語	英字での表記
ANSER	<u>A</u> utomatic <u>a</u> nswer <u>N</u> etwork <u>S</u> ystem for <u>E</u> lectrical <u>R</u> equest
AP	<u>A</u> pplication <u>P</u> rogram
LE	<u>L</u> ogical <u>E</u> ntity
MCF	<u>M</u> essage <u>C</u> ontrol <u>F</u> acility
MHP	<u>M</u> essage <u>H</u> andling <u>P</u> rogram
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram
UAP	<u>U</u> ser <u>A</u> pplication <u>P</u> rogram
UOC	<u>U</u> ser <u>O</u> wn <u>C</u> oding
VC	<u>V</u> irtual <u>C</u> all
WAN	<u>W</u> ide <u>A</u> rea <u>N</u> etwork

常用漢字以外の漢字の使用について

このマニュアルでは、常用漢字を使用することを基本としていますが、次に示す用語について

はじめに

は常用漢字以外の漢字を使用しています。

迂回(うかい) 個所(かしよ) 閉塞(へいそく)

KB (キロバイト) などの単位表記について

1KB (キロバイト), 1MB (メガバイト), 1GB (ギガバイト), 1TB (テラバイト) はそれぞれ 1,024 バイト, 1,024² バイト, 1,024³ バイト, 1,024⁴ バイトです。

謝辞

COBOL 言語仕様は, CODASYL (the Conference on Data Systems Languages : データシステムズ言語協議会) によって, 開発された。OpenTP1 のユーザアプリケーションプログラムのインタフェース仕様のうち, データ操作言語 (DML Data Manipulation Language) の仕様は, CODASYL COBOL (1981) の通信節, RECEIVE 文, SEND 文, COMMIT 文, 及び ROLLBACK 文を参考にし, それに日立製作所独自の解釈と仕様を追加して開発した。原開発者に対し謝意を表すとともに, CODASYL の要求に従って以下の謝辞を掲げる。なお, この文章は, COBOL の原仕様書「CODASYL COBOL JOURNAL OF DEVELOPMENT 1984」の謝辞の一部を再掲するものである。

いかなる組織であっても, COBOL の原仕様書とその仕様の全体又は一部分を複製すること, マニュアルその他の資料のための土台として原仕様書のアイデアを利用することは自由である。ただし, その場合には, その刊行物のまえがきの一部として, 次の謝辞を掲載しなければならない。書評などに短い文章を引用するときは, "COBOL" という名称を示せば謝辞全体を掲載する必要はない。

COBOL は産業界の言語であり, 特定の団体や組織の所有物ではない。

CODASYL COBOL 委員会又は仕様変更の提案者は, このプログラミングシステムと言語の正確さや機能について, いかなる保証も与えない。さらに, それに関連する責任も負わない。

次に示す著作権表示付資料の著作者及び著作権者

FLOW-MATIC (Sperry Rand Corporation の商標), Programming for the Univac (R) I and II, Data Automation Systems, Sperry Rand Corporation 著作権表示 1958 年, 1959 年;
IBM Commercial Translator Form No.F 28-8013, IBM 著作権表示 1959 年;
FACT, DSI 27A5260-2760, Minneapolis-Honeywell, 著作権表示 1960 年

は, これら全体又は一部分を COBOL の原仕様書中に利用することを許可した。この許可は, COBOL 原仕様書をプログラミングマニュアルや類似の刊行物に複製したり, 利用したりする場合にまで拡張される。

目次

1	概要	1
1.1	AP 間通信の概要	2
1.2	AP 間通信の形態	3
1.2.1	通信形態	3
1.2.2	適用範囲	4
1.3	ソフトウェアの構成	5
1.3.1	前提プログラム	5
1.3.2	ソフトウェア構成の例	5
2	機能	7
2.1	AP 間通信の仕組み	8
2.1.1	コネクションとコネクショングループ	8
2.1.2	コネクショングループと論理端末	14
2.1.3	メッセージ送信時のコネクション選択	15
2.1.4	論理端末とアプリケーションの型との関係	15
2.1.5	メッセージとセグメントとの関係	16
2.2	AP 間通信メッセージの送受信	17
2.2.1	一方送信メッセージの受信	17
2.2.2	一方送信メッセージの送信	18
3	メッセージ送受信インタフェース	19
	メッセージ送受信インタフェースの一覧	20
	dc_mcf_receive - メッセージの受信 (C 言語)	22
	dc_mcf_resend - 一方送信メッセージの再送 (C 言語)	26
	dc_mcf_send - 一方送信メッセージの送信 (C 言語)	30
	CBLDCMCF ('RECEIVE') - メッセージの受信 (COBOL 言語)	34
	CBLDCMCF ('RESEND ') - 一方送信メッセージの再送 (COBOL 言語)	40
	CBLDCMCF ('SEND ') - 一方送信メッセージの送信 (COBOL 言語)	46
	RECEIVE - メッセージの受信 (データ操作言語)	52
	SEND - 一方送信メッセージの送信 (データ操作言語)	55
	ユーザアプリケーションプログラムの作成例	59

4	ユーザOWNコーディング，MCF イベントインタフェース	65
4.1	ユーザOWNコーディングインタフェース	66
4.1.1	入力メッセージの編集とアプリケーション名の決定	66
4.1.2	入力メッセージ編集 UOC インタフェース	69
4.1.3	出力メッセージの編集	73
4.1.4	出力メッセージ編集 UOC インタフェース	74
4.1.5	送信メッセージの通番編集	77
4.1.6	送信メッセージの通番編集 UOC インタフェース	79
4.1.7	UOC 作成上の注意事項	80
4.2	MCF イベントインタフェース	81
4.2.1	MCF イベントの種類	81
4.2.2	MCF イベント通知時のセグメント構成	82
4.2.3	MCF イベント情報の形式 (C 言語)	83
4.2.4	MCF イベント情報の形式 (COBOL 言語)	88
5	システム定義	97
	TP1/NET/X25-Extended の定義の概要	98
	TP1/NET/X25-Extended 固有のシステム定義の種類	100
	mcftalccn (コネクション定義)	103
	mcftalcle (論理端末定義)	107
	mcftgrpcn (コネクショングループ定義の開始)	110
	mcftgrped (コネクショングループ定義の終了)	111
	システムサービス情報定義	112
	システムサービス共通情報定義	113
	MCF 定義オブジェクトの生成	116
	自システムの通信管理プログラムと関係づける内容	117
	定義例	118
6	運用コマンド	123
	TP1/NET/X25-Extended の運用コマンド	124
	mcftactcn (コネクションの確立)	125
	mcftactle (論理端末の閉塞解除)	128
	mcftdctcn (コネクションの解放)	130

mcftdctl (論理端末の閉塞)	133
mcftlscn (コネクションの状態表示)	135
mcftlsle (論理端末の状態表示)	138

7

組み込み方法	141
7.1 TP1/NET/X25-Extended の組み込み方法の流れ	142
7.2 MCF メイン関数の作成	143
7.3 定義オブジェクトファイルの生成	146

8

障害対策	149
8.1 障害の種類と対応処理	150

付録

付録 A メッセージ送受信の処理の流れ	156
付録 B 障害発生時の処理の流れ	160
付録 C 理由コード一覧	165
付録 D 用語解説	167

索引

索引	169
----	-----

目次

図 1-1	TP1/NET/X25-Extended を使用したネットワーク構成の例	2
図 1-2	一方受信形態の AP 間通信の例	3
図 1-3	分岐送信形態の AP 間通信の例	4
図 1-4	TP1/NET/X25-Extended を組み込んだソフトウェア構成の例	6
図 2-1	コネクション，コネクショングループ，および論理端末の関係	8
図 2-2	コネクション確立の流れ（発呼）	10
図 2-3	コネクション確立の流れ（着呼）	11
図 2-4	自システムからのコネクション解放の流れ	12
図 2-5	相手システムからのコネクション解放の流れ	13
図 2-6	コネクション切断の流れ	14
図 2-7	メッセージ受信の流れ	17
図 2-8	メッセージ送信の流れ	18
図 3-1	UAP 作成例の処理の流れ	59
図 4-1	アプリケーション名の決定の流れ	68
図 4-2	UOC インタフェース用のパラメタとバッファの関係	73
図 4-3	MCF イベント通知時のセグメント構成	83
図 5-1	TP1/NET/X25-Extended のプロトコル固有定義コマンドの指定順序	102
図 5-2	TP1/NET/X25-Extended のシステム構成例	118
図 7-1	MCF メイン関数のコーディング概要（ANSI C，C++ の場合）	143
図 7-2	MCF メイン関数のコーディング概要（K&R 版 C の場合）	144
図 7-3	MCF メイン関数のディレクトリへの組み込み方法	145
図 7-4	定義オブジェクトファイルの作成方法の概要	147
図 A-1	メッセージ受信の処理の流れ（パケット分割なし）	156
図 A-2	メッセージ受信の処理の流れ（パケット分割あり）	157
図 A-3	メッセージ送信の処理の流れ（パケット分割なし）	158
図 A-4	メッセージ送信の処理の流れ（パケット分割あり）	159
図 B-1	コネクション確立時の障害	160
図 B-2	メッセージ受信時のコネクション切断	161
図 B-3	メッセージ送信時の障害コネクションの迂回	162
図 B-4	入力メッセージ編集 UOC 障害	163
図 B-5	出力メッセージ編集 UOC 障害	164

表目次

表 2-1	論理端末の端末タイプ，メッセージの種類，アプリケーションの型，通信形態，および UAP インタフェースの関係	16
表 3-1	メッセージ送受信に使用するライブラリ関数（C 言語）	20
表 3-2	メッセージ送受信に使用するプログラム（COBOL 言語）	20
表 3-3	メッセージ送受信に使用する通信文（データ操作言語）	21
表 4-1	MCF イベント一覧	81
表 4-2	COBOL 言語の MCF イベント情報の内容 (ERREVT1)	88
表 4-3	COBOL 言語の MCF イベント情報の内容 (ERREVT2)	89
表 4-4	COBOL 言語の MCF イベント情報の内容 (ERREVT3)	91
表 4-5	COBOL 言語の MCF イベント情報の内容 (ERREVT4)	92
表 4-6	COBOL 言語の MCF イベント情報の内容 (SCMPEVT)	94
表 4-7	COBOL 言語の MCF イベント情報の内容 (CERREVT)	95
表 4-8	COBOL 言語の MCF イベント情報の内容 (COPNEVT，CCLSEVT)	96
表 5-1	MCF で使用する定義ファイル	98
表 5-2	TP1/NET/X25-Extended に固有の定義の一覧	100
表 6-1	TP1/NET/X25-Extended の運用コマンド	124
表 8-1	コネクション障害と対応処理	150
表 8-2	コネクショングループ障害と対応処理	151
表 8-3	受信スケジュール関係の障害と対応処理	152
表 8-4	送信スケジュール関係の障害と対応処理	153
表 8-5	UAP 障害と対応処理	154
表 8-6	MCF 障害と対応処理	154
表 C-1	ERREVT2 の理由コード	165
表 C-2	CERREVT の理由コード	166

1

概要

TP1/NET/X25-Extended は、OpenTP1 システムを構成するプログラムの一つです。TP1/NET/X25-Extended を使用すると、ANSER センタと OpenTP1 システムを X.25 (VC) 手順で接続して、メッセージを送受信できます。この章では、TP1/NET/X25-Extended を使用した AP 間通信の概要を説明します。

1.1 AP 間通信の概要

1.2 AP 間通信の形態

1.3 ソフトウェアの構成

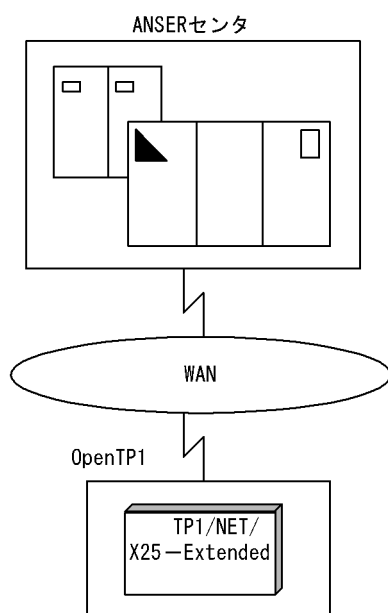
1.1 AP 間通信の概要

AP 間通信とは、異なるシステムにあるアプリケーションプログラム間で、メッセージを送受信することをいいます。

TP1/NET/X25-Extended は、OpenTP1 システムと ANSER センタを X.25 (VC) 手順で接続して、メッセージを送受信するプログラムです。TP1/NET/X25-Extended という相手システムとは、ANSER センタを指します。TP1/NET/X25-Extended を使用して AP 間通信をすると、相手システムで発生したトランザクションを自システムで処理したり、その結果を相手システムへ送り返したりできます。

TP1/NET/X25-Extended を使用したネットワーク構成の例を、図 1-1 に示します。

図 1-1 TP1/NET/X25-Extended を使用したネットワーク構成の例



1.2 AP 間通信の形態

1.2.1 通信形態

TP1/NET/X25-Extended を使用した AP 間通信の形態には、次の二つがあります。

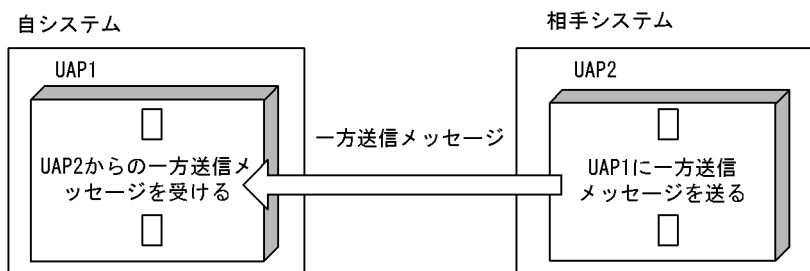
- 一方受信形態
- 分岐送信形態

(1) 一方受信形態

自システムのユーザアプリケーションプログラム (UAP) が、相手システムの UAP からメッセージを受信する形態です。相手システムから受信するメッセージを、一方送信メッセージといいます。

一方受信形態の AP 間通信の例を図 1-2 に示します。

図 1-2 一方受信形態の AP 間通信の例



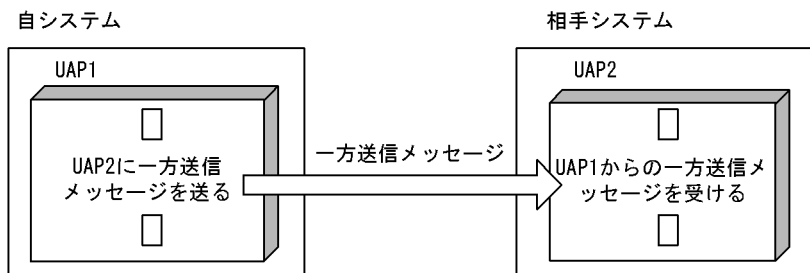
(2) 分岐送信形態

自システムの UAP から別のシステムの UAP へメッセージを送信する形態です。自システムから相手システムへ送信するメッセージも、一方受信形態の場合と同様に、一方送信メッセージといいます。

分岐送信形態の AP 間通信の例を、図 1-3 に示します。

1. 概要

図 1-3 分岐送信形態の AP 間通信の例



1.2.2 適用範囲

TP1/NET/X25-Extended で使用できる X.25 手順の動作モードを次に示します。

- X.25 (80) VC 発呼
- X.25 (80) VC 着呼

1.3 ソフトウェアの構成

TP1/NET/X25-Extended は、OpenTP1 システムに組み込まれて動作するプログラムです。OpenTP1 のメッセージ送受信機能 (TP1/Message Control, TP1/NET/Library) と連携して、メッセージ制御機能 (MCF) を実現します。

この節では、TP1/NET/X25-Extended を使用した MCF を実現するための前提プログラムと、TP1/NET/X25-Extended を組み込んだソフトウェア構成について説明します。

1.3.1 前提プログラム

TP1/NET/X25-Extended を使用した MCF を実現するための前提プログラムは次のとおりです。

適用 OS : AIX

P-1M64-2131 uCosminexus TP1/Server Base 07-00 以降

P-1M14-511 XNF/AS/BASE

P-F1M14-5111 XNF/AS/WAN

R-F1M141-518 XNF/AS/NLI

注

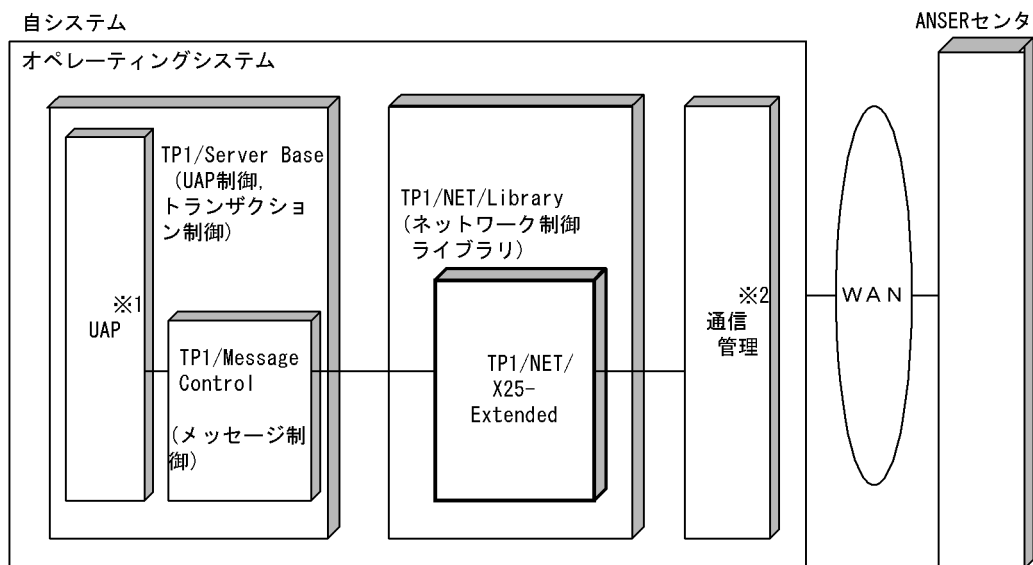
前提となる XNF のバージョンは、AIX V5.1 以降に対応するバージョン以降です。

1.3.2 ソフトウェア構成の例

TP1/NET/X25-Extended を組み込んだソフトウェア構成の例を、図 1-4 に示します。

1. 概要

図 1-4 TP1/NET/X25-Extended を組み込んだソフトウェア構成の例



注 1

TP1/NET/X25-Extended で扱う UAP は、MHP、SPP です。UAP については、マニュアル「OpenTP1 プログラム作成の手引」を参照してください。

注 2

XNF/AS を使用できます。

2

機能

TP1/NET/X25-Extended を使用すると、メッセージの送受信（AP 間通信）ができます。この章では AP 間通信に必要な知識として、コネクション、コネクショングループ、論理端末、メッセージ送受信の流れなどについて説明します。

2.1 AP 間通信の仕組み

2.2 AP 間通信メッセージの送受信

2.1 AP 間通信の仕組み

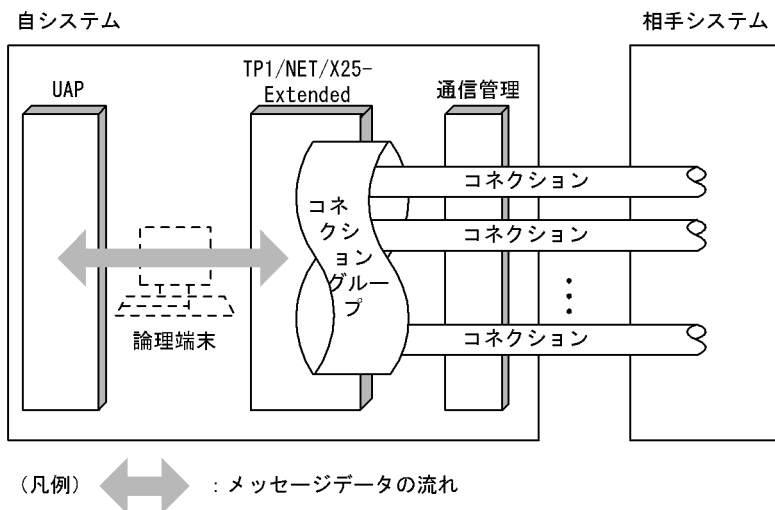
TP1/NET/X25-Extended を使用した AP 間通信では、通信相手システムを表す論理的な概念として、次の三つがあります。

- コネクション
- コネクショングループ
- 論理端末 (LE)

コネクションは、通信管理が相手システムに接続するときの論理的な通信路です。コネクショングループは、幾つかのコネクションをまとめたものです。論理端末は、UAP と TP1/NET/X25-Extended との接点です。

コネクション、コネクショングループ、および論理端末の関係を、図 2-1 に示します。

図 2-1 コネクション、コネクショングループ、および論理端末の関係



2.1.1 コネクションとコネクショングループ

コネクションは、通信相手システムへの論理的な通信路を示します。コネクションを確立・解放することで、相手システムに接続したり、相手システムから切り離したりします。コネクショングループは、同じ目的で使用される幾つかのコネクションをまとめたものです。メッセージを送受信するには、コネクショングループ内のコネクションが一つ以上確立されている必要があります。

コネクションとコネクショングループとの関係は、MCF 通信構成定義で指定します。なお、同じコネクショングループに属するコネクションでは、定義の指定値を一致させる必要があるものがあります。詳細は 5 章の「TP1/NET/X25-Extended 固有のシステム定義の種類」を参照してください。

また、コネクションには、次の二つの種類があります。

発呼コネクション (MCF 通信構成定義 `mcftalccn -t int` を指定)

自システムから相手システムに対して確立するコネクションです。MCF 通信構成定義 (`mcftalccn -q`) で指定した通信相手システムに対して確立します。

着呼コネクション (MCF 通信構成定義 `mcftalccn -t rsp` を指定)

相手システムから自システムに対して確立されるコネクションです。

(1) コネクションおよびコネクショングループの確立

相手システムと通信するには、コネクションを確立する必要があります。コネクショングループ内のコネクションが一つ以上確立されている状態を、そのコネクショングループの確立状態といいます。

コネクションの確立には、次の方法があります。

(a) 発呼コネクションの確立

- オンライン開始時の自動確立

ユーザが MCF 通信構成定義 (`mcftalccn -i`) でコネクションの自動確立を指定しておきます。そのコネクションは、オンライン開始時に、TP1/NET/X25-Extended によって自動的に確立されます。

コネクションが確立されると、状態通知イベント (COPNEVT) が UAP に通知されます。ただし、MCF アプリケーション定義で COPNEVT を定義した場合だけです。

- 運用コマンドによる手動確立

ユーザがオンライン中に運用コマンド (`mcftactcn`) を入力して確立します。運用コマンドでは、コネクションまたはコネクショングループを指定します。コネクショングループを指定すると、コネクショングループ内のすべてのコネクションに対して確立を試みられます。このとき、確立済みのコネクションに対しても確立が試みられることに注意してください。

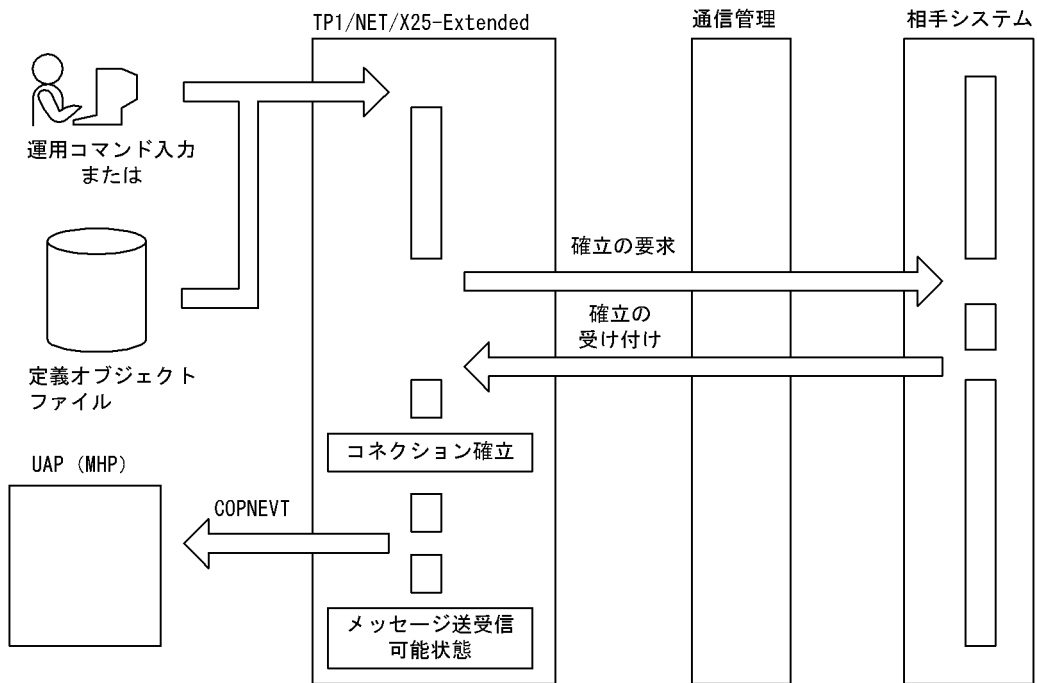
コネクションが確立されると、状態通知イベント (COPNEVT) が UAP に通知されます。ただし、MCF アプリケーション定義で COPNEVT を定義した場合だけです。

コネクショングループを指定して確立した場合は、グループ内のコネクションごとにイベントが通知されます。

発呼コネクションの確立の流れを、図 2-2 に示します。

2. 機能

図 2-2 コネクション確立の流れ（発呼）



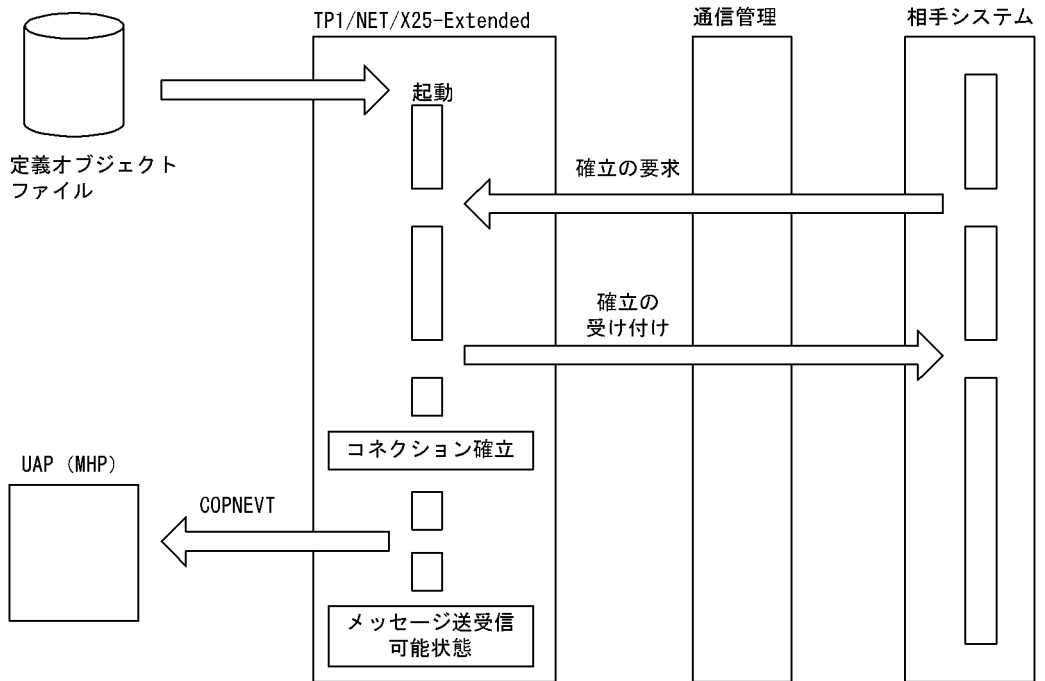
(b) 着呼コネクションの確立

通信相手システムからの確立要求に従って、TP1/NET/X25-Extended がコネクションを確立します。

コネクションが確立されると、状態通知イベント (COPNEVT) が UAP に通知されます。ただし、MCF アプリケーション定義で COPNEVT を定義した場合だけです。

着呼コネクションの確立の流れを、図 2-3 に示します。

図 2-3 コネクション確立の流れ (着呼)



(2) コネクションおよびコネクショングループの解放

相手システムとの通信を終了するために、コネクションを解放します。コネクショングループ内のコネクションが一つも確立されていない状態を、そのコネクショングループの解放状態といいます。コネクショングループが解放状態になったときは、それに対する状態通知イベント (CCLSEVT) は通知されません。このため、ユーザがコネクショングループ内のそれぞれのコネクションの状態を把握して、コネクショングループが解放状態になったかどうかを判断してください。

コネクションが解放されるときに、送受信メッセージが破棄される場合があります。このため、UAP は送受信メッセージを管理する必要があります。なお、このとき、障害通知イベント (CERREVT) は通知されません。

コネクションの解放には、次の方法があります。

(a) 自システムからのコネクションの解放

• 運用コマンドによる手動解放

ユーザがオンライン中に運用コマンド (mcftdetcn) を入力して解放します。運用コマンドでは、コネクションまたはコネクショングループを指定します。コネクショングループを指定すると、コネクショングループ内のすべてのコネクションに対して解放が試みられます。このとき、確立されていないコネクションに対して解放が試みられることに注意してください。

コネクションが解放されると、状態通知イベント (CCLSEVT) が UAP に通知されま

2. 機能

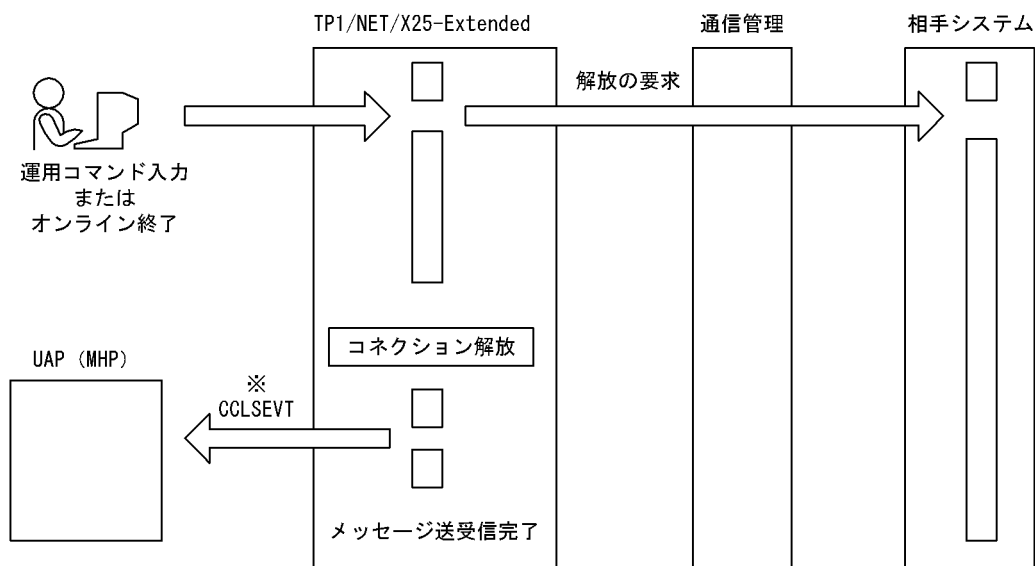
す。ただし、MCF アプリケーション定義で CCLSEVT を定義した場合だけです。コネクショングループを指定して解放した場合は、グループ内のコネクションごとにイベントが通知されます。

- オンライン終了時の自動解放

すべてのコネクションは、オンライン終了時に、TP1/NET/X25-Extended によって自動的に解放されます。このとき、状態通知イベント（CCLSEVT）は通知されません。

自システムからのコネクションの解放の流れを、図 2-4 に示します。

図 2-4 自システムからのコネクション解放の流れ



注 オンライン終了時は、状態通知イベント（CCLSEVT）は通知されません。

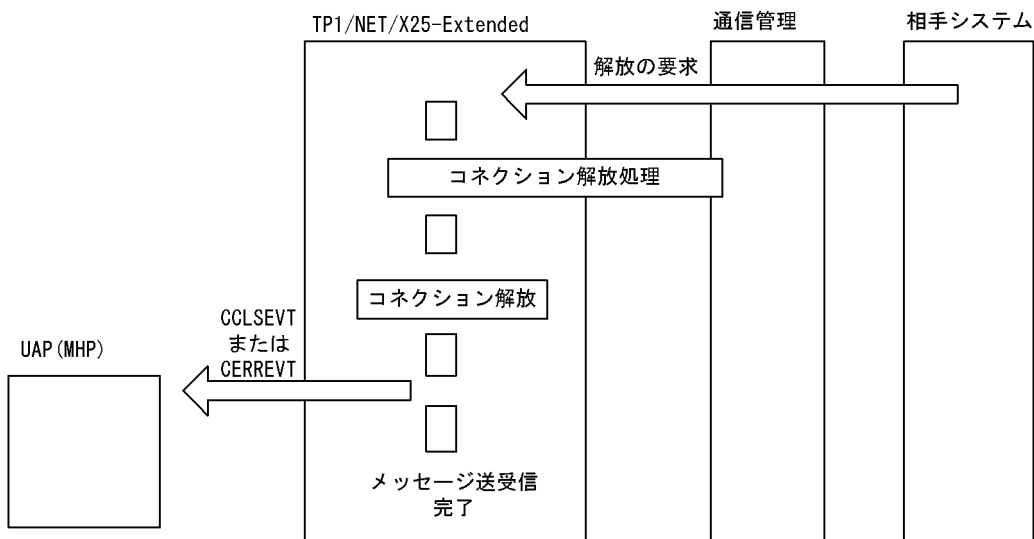
(b) 相手システムからのコネクションの解放

通信相手システムからの解放要求に従って、TP1/NET/X25-Extended がコネクションを解放します。

コネクションが解放されると、状態通知イベント（CCLSEVT）か障害通知イベント（CERREVT）のどちらかが UAP に通知されます。ただし、MCF アプリケーション定義で該当するイベントを定義した場合だけです。どちらのイベントが通知されるかは、MCF 通信構成定義（mcftalccn -f kind）の指定で決まります。

相手システムからのコネクション解放の流れを、図 2-5 に示します。

図 2-5 相手システムからの接続解放の流れ



(3) コネクションの切断

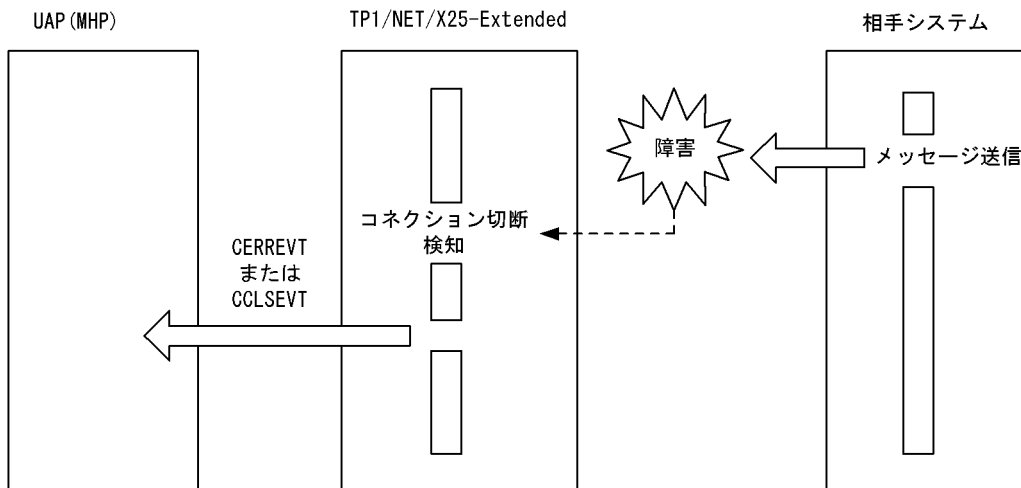
回線障害などによってコネクションが使用できなくなると、TP1/NET/X25-Extended は、コネクションを切断状態にします。

コネクションが切断されると、障害通知イベント (CERREVT) か状態通知イベント (CCLSEVT) のどちらかが UAP に通知されます。ただし、MCF アプリケーション定義で該当するイベントを定義した場合だけです。どちらのイベントが通知されるかは、MCF 通信構成定義 (mcfalcn -f kind) の指定で決まります。

コネクションが切断されるときに、送受信メッセージが破棄される場合があります。このため、UAP は送受信メッセージを管理する必要があります。

コネクション切断の流れを、図 2-6 に示します。

図 2-6 コネクション切断の流れ



(4) コネクション障害後の再確立

回線障害などによってコネクションが使用できない場合、次の方法でコネクションを再確立します。

(a) 自システムからの再確立

• コネクション確立障害時の自動確立再試行

ユーザがMCF通信構成定義 (mcftalcn -b) でコネクションの確立再試行を指定しておきます。コネクションの確立時に障害が発生した場合、そのコネクションは、TP1/NET/X25-Extended によって再び確立が試みられます。指定した条件の再試行でコネクションが確立できなかった場合、エラーを示すメッセージログが出力され、そのコネクションは閉塞 (へいそく) されます。

この確立再試行は、いったん確立されたあとのコネクション障害には適用されないことに注意してください。

• 運用コマンドによる手動確立

障害の要因を取り除いたあと、ユーザが運用コマンド (mcfactcn) を入力して再確立します。再確立の必要があるかどうかは、メッセージログおよび障害通知イベント (CERREVT) の情報で判断してください。

(b) 相手システムからの再確立

通信相手システムからの確立要求に従って、TP1/NET/X25-Extended がコネクションを再確立します。

2.1.2 コネクショングループと論理端末

論理端末 (LE) は、UAP と TP1/NET/X25-Extended との接点です。UAP は、特定の通信相手システムを示す論理端末からメッセージを受信し、同じまたは別の論理端末へ

メッセージを送信します。

論理端末は、コネクシヨングループと1対1に対応します。コネクシヨングループと論理端末の対応は、MCF通信構成定義で指定します。

(1) 論理端末の閉塞

メッセージを送信できない論理端末の状態を、論理端末の閉塞状態といいます。このとき、送信メッセージは、論理端末に対応する出力キューに残ります。

UAPから閉塞状態の論理端末に対して、SENDでメッセージを出力できます。また、相手システムから受信したメッセージも閉塞中の論理端末で受け付けられます。

論理端末に対応するコネクシヨングループが解放状態になると、TP1/NET/X25-Extendedは、その論理端末を自動的に閉塞します。また、ユーザが運用コマンド(mcfctdctl)で論理端末を閉塞することもできます。ただし、使用中の論理端末は閉塞できません。なお、閉塞状態の論理端末を閉塞しようとする、警告を示すメッセージログが出力されます。

(2) 論理端末の閉塞解除

論理端末に対応するコネクシヨングループが確立状態になると、TP1/NET/X25-Extendedは、その論理端末の閉塞状態を自動的に解除します。また、ユーザが運用コマンド(mcfctactl)で論理端末を閉塞解除することもできます。ただし、論理端末に対応するコネクシヨングループが確立状態である必要があります。閉塞されていない論理端末を閉塞解除しようとする、警告を示すメッセージログが出力されます。

2.1.3 メッセージ送信時のコネクシヨン選択

UAPが論理端末を指定してメッセージを送信したときに、実際に使用されるコネクシヨンは次のようになります。

使用されるコネクシヨンは、指定した論理端末に対応するコネクシヨングループ内から、TP1/NET/X25-Extendedによってラウンドロビン方式で選択されます。ただし、確立状態でないもの、ほかのメッセージの送信で使用されているものは除かれます。

コネクシヨ障害などでメッセージが正しく送信されなかった場合、障害コネクシヨンの迂回が試みられます。TP1/NET/X25-Extendedは障害通知イベント(CERREVT)または状態通知イベント(CCLSEVT)を通知したあと、ほかの正常なコネクシヨを使用して、再び送信を試みます。このとき、以前に使用したことのあるコネクシヨかどうかはチェックされません。

2.1.4 論理端末とアプリケーションの型との関係

論理端末やアプリケーションの型に関する情報を、システム定義で指定する必要があります。

2. 機能

TP1/NET/X25-Extended で使用する論理端末の端末タイプを示します。論理端末定義 (mcfaltcle -t) で、この端末タイプを指定してください。

any : 任意型論理端末

次に、TP1/NET/X25-Extended で使用できるアプリケーションの型を示します。アプリケーション属性定義 (mcfalcap -n) で、この型を指定してください。

noans : 非応答型

論理端末定義については「5. システム定義」を、アプリケーション属性定義については、マニュアル「OpenTP1 システム定義」を参照してください。

論理端末の端末タイプ、メッセージの種類、アプリケーションの型、通信形態、および UAP インタフェースの関係を表 2-1 に示します。

表 2-1 論理端末の端末タイプ、メッセージの種類、アプリケーションの型、通信形態、および UAP インタフェースの関係

論理端末の 端末タイプ	メッセージの種類	アプリケーション の型	通信形態	UAP インタ フェース
any (任意型)	一方送信メッセージ	noans (非応答型)	一方受信形態	receive
			分岐送信形態	send
				resend

2.1.5 メッセージとセグメントとの関係

UAP からの 1 回の関数の呼び出しで入出力できるメッセージの分割単位を、セグメントといいます。TP1/NET/X25-Extended を使用して相手システムと送受信する場合、一つのメッセージが一つのセグメントで構成されます。つまり、1 回の関数の呼び出しで、一つのメッセージを送受信できます。ただし、MCF イベント (MCF が通知する特殊なメッセージ) は、二つのセグメントで構成されることがあります。これを受信する場合は、メッセージを受信する関数を 2 回呼び出してください。MCF イベントのセグメント構成については、「4.2.2 MCF イベント通知時のセグメント構成」を参照してください。

メッセージ送受信の関数で処理するセグメントの先頭には、MCF で使用されるヘッダ領域があります。このヘッダ領域の長さによって、バッファ形式 1 とバッファ形式 2 があります。通常、バッファ形式 1 を使用します。

2.2 AP 間通信メッセージの送受信

TP1/NET/X25-Extended を使用すると、次のメッセージの送受信ができます。

- 一方送信メッセージの受信
- 一方送信メッセージの送信

2.2.1 一方送信メッセージの受信

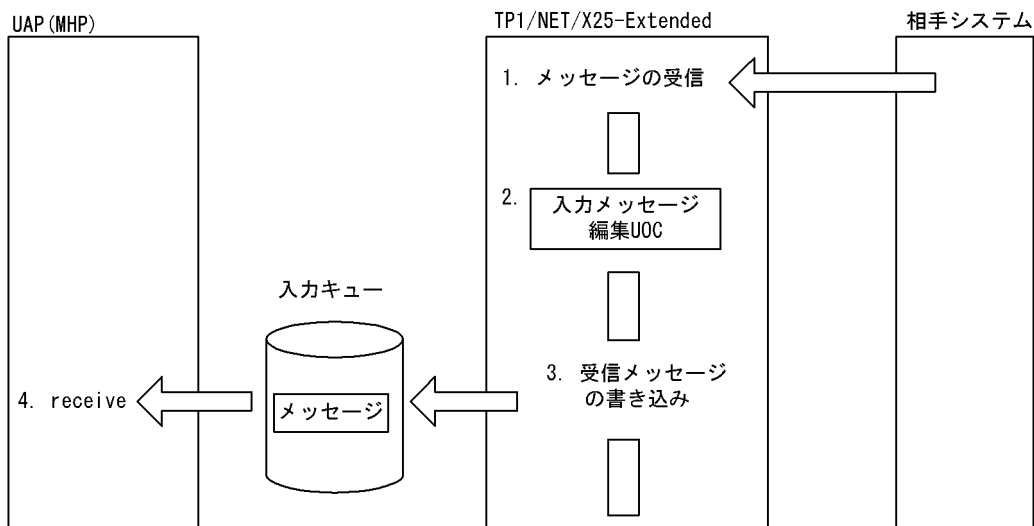
通信相手システムから送られたメッセージを UAP が受信します。TP1/NET/X25-Extended を使用した場合の、一方送信メッセージの受信の流れを次に示します。

1. 相手システムから送られた一方送信メッセージが、TP1/NET/X25-Extended によって受信されます。パケット分割されている場合は、通信管理または TP1/NET/X25-Extended によって組み立てられます。
2. 受信されたメッセージは、入力メッセージ編集 UOC で編集されます。UOC が登録されていない場合、UOC を呼び出さないで次の処理へ続きます。
3. (編集後の)メッセージが入力キューに書き込まれます。
4. RECEIVE 文または `dc_mcf_receive` 関数を呼び出した UAP に、入力キューから取り出されたメッセージが渡されます。

受信するメッセージの長さがパケットサイズ以下になるように設計すると、パケットの組み立てが不要になるため、性能が向上します。

メッセージ受信の流れを図 2-7 に示します。

図 2-7 メッセージ受信の流れ



2.2.2 一方送信メッセージの送信

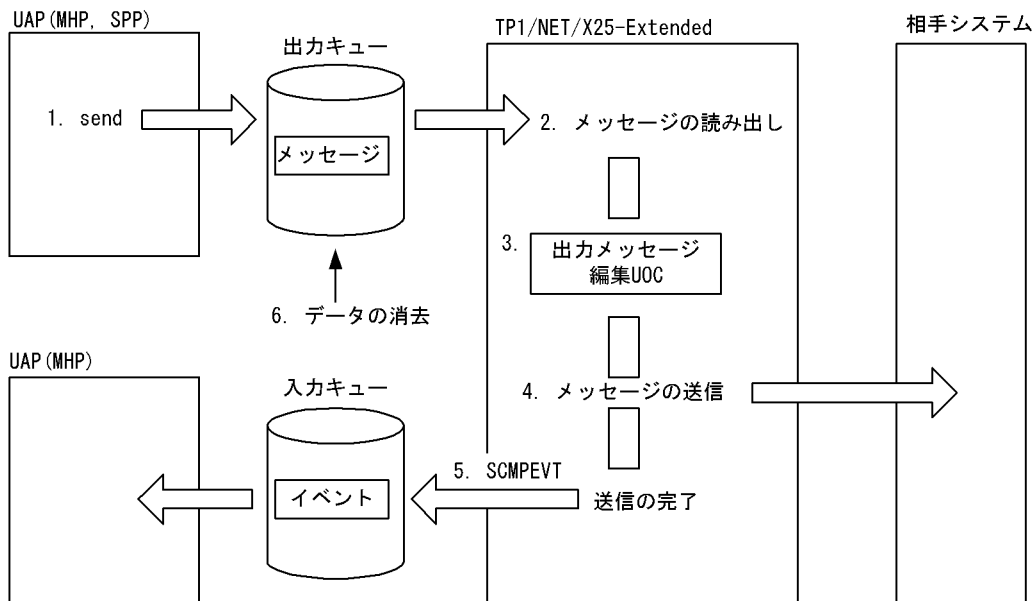
自システムの UAP から通信相手システムへメッセージを送信します。TP1/NET/X25-Extended を使用した場合の、一方送信メッセージの送信の流れを次に示します。

1. SEND 文または `dc_mcf_send` 関数で UAP が渡したメッセージが、出力キューに書き込まれます。
2. TP1/NET/X25-Extended によって、出力キューからメッセージが読み出されます。
3. 読み出されたメッセージは、出力メッセージ編集 UOC で編集されます。UOC が登録されていない場合、UOC を呼び出さずに次の処理へ続きます。
4. (編集後の)メッセージが、TP1/NET/X25-Extended によって相手システムへ送信されます。このとき、通信管理によってパケット分割される場合があります。
5. 送信が完了すると、送信完了通知イベント (SCMPEVT) が通知されます。ただし、MCF アプリケーション定義で SCMPEVT を定義して、かつ、UAP が送信完了通知イベントを要求していた場合だけです。
6. TP1/NET/X25-Extended によって、出力キューにある送信済みのメッセージが消去されます。

送信するメッセージの長さがパケットサイズ以下になるように設計すると、パケット分割が不要になるため、性能が向上します。

メッセージ送信の流れを図 2-8 に示します。

図 2-8 メッセージ送信の流れ



3

メッセージ送受信インタフェース

TP1/NET/X25-Extended を使用してメッセージを送受信する場合、ユーザは個別の業務に対応させるため、UAP を作成します。この章では、UAP の作成方法、および作成例について説明します。

メッセージ送受信インタフェースの一覧

dc_mcf_receive - メッセージの受信 (C 言語)

dc_mcf_resend - 一方送信メッセージの再送 (C 言語)

dc_mcf_send - 一方送信メッセージの送信 (C 言語)

CBLDCMCF ('RECEIVE ') - メッセージの受信 (COBOL 言語)

CBLDCMCF ('RESEND ') - 一方送信メッセージの再送 (COBOL 言語)

CBLDCMCF ('SEND ') - 一方送信メッセージの送信 (COBOL 言語)

RECEIVE - メッセージの受信 (データ操作言語)

SEND - 一方送信メッセージの送信 (データ操作言語)

ユーザアプリケーションプログラムの作成例

メッセージ送受信インタフェースの一覧

TP1/NET/X25-Extended のメッセージ送受信インタフェースについて、C 言語、COBOL 言語、およびデータ操作言語に分けて説明します。

UAP の作成方法の詳細については、マニュアル「OpenTP1 プログラム作成の手引」を参照してください。

C 言語を使用したメッセージ送受信

C 言語を使用してメッセージを送受信する場合は、OpenTP1 システムで提供する関数を使用して UAP を作成します。

TP1/NET/X25-Extended で扱うメッセージ送受信のライブラリ関数を次の表に示します。

表 3-1 メッセージ送受信に使用するライブラリ関数 (C 言語)

関数名	機能
dc_mcf_receive	メッセージの受信
dc_mcf_resend	一方送信メッセージの再送
dc_mcf_send	一方送信メッセージの送信

その他の関数については、マニュアル「OpenTP1 プログラム作成リファレンス C 言語編」を参照してください。

COBOL 言語を使用したメッセージ送受信

COBOL 言語を使用してメッセージを送受信する場合は、OpenTP1 システムで提供する関数に対応しているプログラムを、CALL 文で呼び出して UAP を作成します。

TP1/NET/X25-Extended で扱うメッセージ送受信に使用するプログラムを次の表に示します。

表 3-2 メッセージ送受信に使用するプログラム (COBOL 言語)

プログラム名	データ名	機能
CBLDCMCF	'RECEIVE '	メッセージの受信
	'RESEND '	一方送信メッセージの再送
	'SEND '	一方送信メッセージの送信

その他のプログラムについては、マニュアル「OpenTP1 プログラム作成リファレンス COBOL 言語編」を参照してください。

データ操作言語（COBOL 言語）を使用したメッセージ送受信

データ操作言語（COBOL 言語）を使用した、メッセージ送受信の文について説明します。データ操作言語の形式の詳細については、マニュアル「OpenTP1 プログラム作成リファレンス COBOL 言語編」を参照してください。

TP1/NET/X25-Extended で扱うメッセージ送受信に使用する通信文を次の表に示します。

表 3-3 メッセージ送受信に使用する通信文（データ操作言語）

通信文		機能	対応する CALL インタフェース
データコミュニケーション機能	RECEIVE	メッセージの受信	CBLDCMCF ('RECEIVE ')
	SEND	一方送信メッセージの送信	CBLDCMCF ('SEND ')

注

一方送信メッセージの再送 - CBLDCMCF ('RESEND ') のデータ操作言語のインタフェースはありません。

dc_mcf_receive - メッセージの受信 (C 言語)

形式

ANSI C, C++ の形式

```
#include <dcmcf.h>
int dc_mcf_receive(DCLONG action,DCLONG commform,char *termnam,
                  char *resv01,char *recvdata,DCLONG *rdataleng,
                  DCLONG inbufleng,DCLONG *time)
```

K&R 版 C の形式

```
#include <dcmcf.h>
int
dc_mcf_receive(action,commform,termnam,resv01,recvdata,rdataleng,
               inbufleng,time)

DCLONG    action;
DCLONG    commform;
char      *termnam;
char      *resv01;
char      *recvdata;
DCLONG    *rdataleng;
DCLONG    inbufleng;
DCLONG    *time;
```

機能

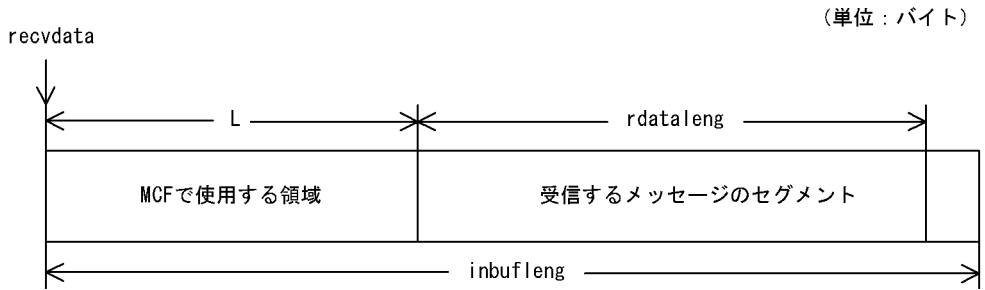
論理端末に届いたメッセージのうち、一つのセグメントを受信します。セグメントの数だけ dc_mcf_receive 関数を呼び出すと、一つの論理メッセージを受信できます。

dc_mcf_receive 関数で受信できるメッセージの種類を次に示します。

- 相手システムから送信されたメッセージ
- MCF イベント
- アプリケーション起動機能で渡されたメッセージ

TP1/NET/X25-Extended を使用して通信する場合、相手システムから送信されるメッセージは、常に単一セグメントで構成されます。

セグメントを受信する領域の形式を次に示します。L は、バッファ形式 1 の場合は 8 バイト、バッファ形式 2 の場合は 4 バイトです。



UAP で値を設定する引数

action

メッセージの先頭セグメントを受信するかどうか、および使用するバッファ形式を、次の形式で設定します。

```
{ DCMCFFRST | DCMCFSEG } [ | { DCMCFBUF1 | DCMCFBUF2 } ]
```

DCMCFFRST

先頭セグメントを受信する場合や、メッセージが単一セグメントの場合に、DCMCFFRST を設定します。

DCMCFSEG

中間セグメントまたは最終セグメントを受信する場合に設定します。

DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

commform

DCNOFLAGS を設定します。

termnam

中間セグメントまたは最終セグメントを受信する場合は、入力元の論理端末名称を設定します。先頭セグメントを受信したときに返された論理端末名称を設定します。

先頭セグメントの処理終了後、termnam には OpenTP1 から値が返されます。

resv01

ヌル文字を設定します。

3. メッセージ送受信インタフェース
 dc_mcf_receive - メッセージの受信 (C 言語)

recvdata

セグメントを受信する領域を設定します。

dc_mcf_receive 関数が終了すると、メッセージのセグメントの一つが返されます。

処理終了後、recvdata には OpenTP1 から値が返されます。

inbufleng

セグメントを受信する領域の長さを設定します。

OpenTP1 から値が返される引数

termnam

先頭セグメントを受信する場合だけ、入力元の論理端末名称が返されます。

中間セグメントまたは最終セグメントを受信する場合は、ここで返された論理端末名称を termnam に設定します。

recvdata

受信したセグメントの内容が返されます。

rdata Leng

受信したセグメントの長さが返されます。

time

メッセージを受信した時刻が、1970 年 1 月 1 日 0 時 0 分 0 秒からの通算の秒数で返されます。

リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71000	先頭セグメントを受信する dc_mcf_receive 関数を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、action に DCMCFSEG を設定して dc_mcf_receive 関数を呼び出してください。
DCMCFRTN_71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する dc_mcf_receive 関数を呼び出しています。直前に呼び出した dc_mcf_receive 関数でメッセージはすべて受信しました。このリターン値が返されたあとに、再び dc_mcf_receive 関数を呼び出した場合は、リターン値 DCMCFRTN_72000 が返されます。
DCMCFRTN_71002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。

リターン値	意味
DCMCFRTN_72000	<p>MHP の実行でリターンした場合</p> <ul style="list-style-type: none"> 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、中間セグメントまたは最終セグメントを受信する dc_mcf_receive 関数を呼び出しています。先頭セグメントを受信する場合は、action に DCMCFRST を設定して dc_mcf_receive 関数を呼び出してください。 リターン値 DCMCFRTN_71001 が返されたあとに、再び dc_mcf_receive 関数を呼び出しています。 <p>SPP の実行でリターンした場合 SPP では dc_mcf_receive 関数を呼び出せません。</p>
DCMCFRTN_72001	termnam に設定した論理端末名称が間違っています。
DCMCFRTN_72013	inbufleng の指定値を超えるセグメントを受信しました。inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRTN_72016	action に設定した値が間違っています。
	resv01 に設定した値が間違っています。
	引数に設定した値に間違いがあります。
DCMCFRTN_72024	commform に設定した値が間違っています。
DCMCFRTN_72025	action に設定したセグメント種別 (DCMCFRST または DCMCFSEG) の値が間違っています。
DCMCFRTN_72036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

dc_mcf_resend - 一方送信メッセージの再送 (C 言語)

形式

ANSI C, C++ の形式

```
#include <dcmcf.h>
int dc_mcf_resend(DCLONG action,DCLONG commform,char *rtermnam,
                 char *resv01,DCLONG oseqid,DCLONG orgseq,
                 char *otermnam,char *resv02,char *resv03,
                 char *resv04,DCLONG opcd)
```

K&R 版 C の形式

```
#include <dcmcf.h>
int dc_mcf_resend(action,commform,rtermnam,resv01,oseqid,orgseq,
                 otermnam,resv02,resv03,resv04,opcd)

DCLONG    action;
DCLONG    commform;
char      *rtermnam;
char      *resv01;
DCLONG    oseqid;
DCLONG    orgseq;
char      *otermnam;
char      *resv02;
char      *resv03;
char      *resv04;
DCLONG    opcd;
```

機能

以前に送信したメッセージを、再び送信します。再送するメッセージは、以前に送信したメッセージとは別の、新しいメッセージとして扱います。どのメッセージを再送するかは、次に示す送信済みメッセージの情報で選択できます。

- 出力先の論理端末名称
- メッセージ通番
- メッセージ種別 (一般の一方送信, 優先の一方送信)

対象としたメッセージが以前に送信されていない場合は、dc_mcf_resend 関数はリターン値 DCMCFRTN_NOMSG を返します。また、メッセージキュー (ディスクキュー) 内に対象のメッセージがない場合も、リターン値 DCMCFRTN_NOMSG を返します。このため、使用するメッセージキューの種別ではディスクキューを指定して、メッセージキューの大きさの定義は余裕を持った値を指定してください。

UAP で値を設定する引数

action

再送するメッセージに出力通番を付け直すかどうか、一般か優先か、および最終出力通番のメッセージを再送するかどうかを、次の形式で設定します。

```
{ DCMCFSEQ | DCMCFNSEQ } [ | { DCMCFNORM | DCMCFPRIO } ] [ | DCMCFLAST ]
```

DCMCFSEQ

再送するメッセージに出力通番を付け直す場合に設定します。

DCMCFNSEQ

再送するメッセージに出力通番を付け直さない場合に設定します。

DCMCFNORM

一般の一方送信メッセージとして再送する場合に設定します。

DCMCFPRIO

優先の一方送信メッセージとして再送する場合に設定します。

DCMCFLAST

最終出力通番を持つメッセージを再送する場合に設定します。この値を設定した場合は、orgseq に設定した値は無効となります。

commform

一方送信を示す、DCMCFOUT を設定します。

rtermnam

出力先の論理端末名称を設定します。論理端末名称の長さは最大 8 バイトです。論理端末名称の後ろにはヌル文字を付けてください。

resv01

ヌル文字を設定します。

oseqid

再送するメッセージを検索するキーとして、以前に送信したメッセージの送信種別を設定します。

DCMCFRID_NORM

一般の一方送信メッセージを対象とする場合に設定します。

DCMCFRID_PRIO

優先の一方送信メッセージを対象とする場合に設定します。

3. メッセージ送受信インタフェース

dc_mcf_resend - 一方送信メッセージの再送 (C 言語)

省略した場合は、DCMCFRID_NORM (一般の一方送信メッセージを対象) が設定されます。DCMCFRID_PRIO を設定した場合は、otermnam に出力先の論理端末名称を設定できません。

orgseq

再送するメッセージを検索するキーとして、以前に送信したメッセージの出力通番を設定します。action で DCMCFLAST を設定した場合は、ここに設定した値は無効となります。

otermnam

再送するメッセージを検索するキーとして、以前に送信したメッセージの出力先の論理端末名称を設定します。論理端末名称の長さは最大 8 バイトです。論理端末名称の後ろにはヌル文字を付けてください。

resv02, resv03, resv04

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_NOMSG	該当するメッセージがありません。
DCMCFRTN_BUF_SHORT	再送するメッセージのセグメントの長さが、UAP 共通定義 (mcfmuap -e) で指定した値を超えています。
DCMCFRTN_71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	MCF が終了処理中のため、メッセージの再送を受け付けられません。
DCMCFRTN_71003	メッセージキューが満杯です。
DCMCFRTN_71004	メッセージキューから取り出したメッセージを格納するバッファ (作業領域) を、メモリ上に確保できませんでした。
DCMCFRTN_71108	メッセージを再送しようとしたのですが、再送先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。

リターン値	意味
DCMCFRTN_72000	MHP の実行でリターンした場合 <ul style="list-style-type: none"> 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_resend 関数を呼び出しています。 非トランザクション属性の MHP から、dc_mcf_resend 関数を呼び出しています。
	SPP の実行でリターンした場合 トランザクションでない SPP の処理から、dc_mcf_resend 関数を呼び出しています。
DCMCFRTN_72001	rtermnam または otermnam に設定した論理端末名称が間違っています。
	dc_mcf_resend 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72016	action に設定したメッセージ種別 (DCMCFNORM または DCMCFPRIO) の値が間違っています。
	action に設定した値が間違っています。
	oseqid に設定した値が間違っています。
	opcd に設定した値が間違っています。
	resv01, resv02, resv03, または resv04 に設定した値が間違っています。
	引数に設定した値に間違いがあります。
DCMCFRTN_72017	action に設定した出力通番の要否 (DCMCFSEQ または DCMCFNSEQ) の値が間違っています。
DCMCFRTN_72024	commform に設定した値が間違っています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

注意事項

メッセージの再送時には、MCF マネージャ定義の UAP 共通定義 (mcfmuap) の -e オプションと -l オプションの指定値に注意してください。

-e オプション

-e オプションでは、dc_mcf_resend 関数で使用する作業領域の大きさを指定します。再送するメッセージのセグメントがこの作業領域より大きい場合、dc_mcf_resend 関数はメッセージを再送しないで、リターン値 DCMCFRTN_BUF_SHORT を返します。このため、-e オプションでは、セグメントの最大長よりも大きな値を設定しておいてください。

-l オプション

-l オプションでは、通番に関して指定します。この内容によっては、メッセージキューファイル内に同じ通番を持つメッセージが同時に存在することがあります。この場合は、どのメッセージを再送するか保証できません。

dc_mcf_send - 一方送信メッセージの送信 (C 言語)

形式

ANSI C, C++ の形式

```
#include <dcmcf.h>
int dc_mcf_send(DCLONG action, DCLONG commform, char *termnam,
               char *resv01, char *senddata, DCLONG sdataleng,
               char *resv02, DCLONG opcd)
```

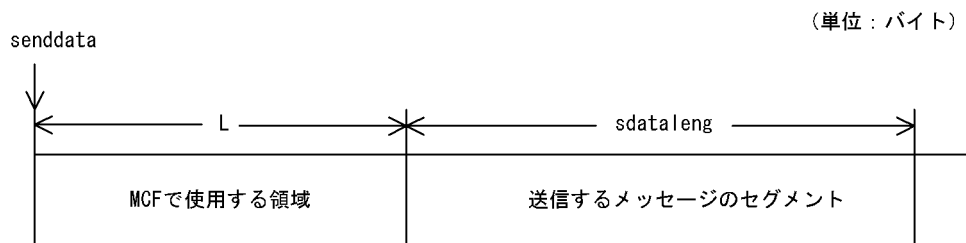
K&R 版 C の形式

```
#include <dcmcf.h>
int dc_mcf_send(action, commform, termnam, resv01, senddata, sdataleng,
               resv02, opcd)
DCLONG action;
DCLONG commform;
char *termnam;
char *resv01;
char *senddata;
DCLONG sdataleng;
char *resv02;
DCLONG opcd;
```

機能

相手システムへ一方送信メッセージを送信します。一方送信メッセージは、一つのセグメントで構成されます。

セグメントを送信する領域の形式を次に示します。L は、バッファ形式 1 の場合は 8 バイト、バッファ形式 2 の場合は 4 バイトです。



UAP で値を設定する引数

action

単一セグメントを送信すること、優先か一般か、出力通番を付けるかどうか、使用する

バッファ形式, および送信完了通知イベントを通知させるかどうかを, 次の形式で設定します。

```
DCMCFEMI [ | { DCMCFNORM | DCMCFPRIO } ] [ | { DCMCFSEQ | DCMCFNSEQ } ] [ | { DCMCFBUF1 | DCMCFBUF2 } ] [ | DCMCFSEVT ]
```

DCMCFEMI

単一セグメントを示す DCMCFEMI を設定します。

DCMCFNORM

一般の一方送信メッセージとして送信する場合に設定します。

DCMCFPRIO

優先の一方送信メッセージとして送信する場合に設定します。

DCMCFSEQ

出力通番が必要な場合に設定します。

DCMCFNSEQ

出力通番が必要ない場合に設定します。

DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

DCMCFSEVT

送信完了通知イベントを通知させる場合に設定します。ただし, 送信完了通知イベント処理用の MHP を MCF アプリケーション定義で指定していない場合は無効です。

commform

一方送信を示す, DCMCFOUT を設定します。

termnam

出力先の論理端末名称を設定します。論理端末名称の長さは最大 8 バイトです。論理端末名称の後ろにはヌル文字を付けてください。

resv01

ヌル文字を設定します。

senddata

送信するセグメントの内容を設定した領域を設定します。一つのセグメントで

3. メッセージ送受信インタフェース

dc_mcf_send - 一方送信メッセージの送信 (C 言語)

32000 バイトまで送信できます。

sdatale`ng`

送信するセグメントの長さを設定します。

resv02

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	sdatale <code>ng</code> に 32000 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
DCMCFRTN_71003	メッセージキューが満杯です。
DCMCFRTN_71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRTN_71108	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	MHP の実行でリターンした場合 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、 dc_mcf_send 関数を呼び出しています。
	SPP の実行でリターンした場合 トランザクションでない SPP の処理から、dc_mcf_send 関数を呼び出しています。
DCMCFRTN_72001	termnam に設定した論理端末名称が間違っています。
	dc_mcf_send 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72016	action に設定したメッセージ種別 (DCMCFNORM または DCMCFPRIO) の値が間違っています。
	action に設定した値が間違っています。
	opcd に設定した値が間違っています。
	resv01 または resv02 に設定した値が間違っています。
	引数に設定した値に間違いがあります。

3. メッセージ送受信インタフェース
dc_mcf_send - 一方送信メッセージの送信 (C 言語)

リターン値	意味
DCMCFRTN_72017	action に設定した出力通番の要否 (DCMCFSEQ または DCMCFNSEQ) の値が間違っています。
DCMCFRTN_72024	commform に設定した値が間違っています。
DCMCFRTN_72026	action に設定したセグメント種別 (DCMCFEMI) の値が間違っています。
DCMCFRTN_72041	sdataleng に 0 以下の値を設定しています。
上記以外	プログラムの破壊などによる, 予期しないエラーが発生しました。

CBLDCMCF ('RECEIVE') - メッセージの受信 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

```
CALL 'CBLDCMCF' USING 一意名1 一意名2 一意名3
```

DATA DIVISION の指定

```
01 一意名1.  
  02 データ名A PIC X(8) VALUE 'RECEIVE'.  
  02 データ名B PIC X(5).  
  02 FILLER PIC X(3).  
  02 データ名C PIC X(4).  
  02 データ名D PIC X(4) VALUE SPACE.  
  02 データ名E PIC 9(8).  
  02 データ名F PIC 9(8).  
  02 データ名G PIC 9(9) COMP.  
  02 データ名H PIC X(4) VALUE SPACE.  
  02 データ名I PIC X(4) VALUE SPACE.  
  02 データ名J PIC X(4) VALUE SPACE.  
  02 データ名K PIC X(4) VALUE SPACE.  
  02 データ名L PIC X(8) VALUE SPACE.  
  02 データ名M1 PIC X(4) VALUE SPACE.  
  02 データ名M2 PIC X(8) VALUE SPACE.  
  02 データ名M3 PIC X(4) VALUE SPACE.  
  02 データ名M4 PIC 9(9) COMP VALUE ZERO.  
  02 データ名M5 PIC 9(9) COMP VALUE ZERO.  
  02 データ名M6 PIC X(1) VALUE SPACE.  
  02 データ名M7 PIC X(1).  
  02 データ名N PIC X(14) VALUE LOW-VALUE.  
01 一意名2.  
  02 データ名O PIC X(4) VALUE SPACE.  
  02 データ名P PIC X(8).  
  02 データ名Q PIC X(8).  
  02 データ名R PIC X(8) VALUE SPACE.  
  02 データ名T PIC X(28) VALUE LOW-VALUE.  
01 一意名3.  
  02 データ名U PIC 9(x) COMP.  
  02 データ名V PIC X(x).  
  02 データ名W PIC X(n).
```

機能

論理端末に届いたメッセージのうち、一つのセグメントを受信します。セグメントの数だけ CBLDCMCF ('RECEIVE ') を呼び出すと、一つの論理メッセージを受信できます。

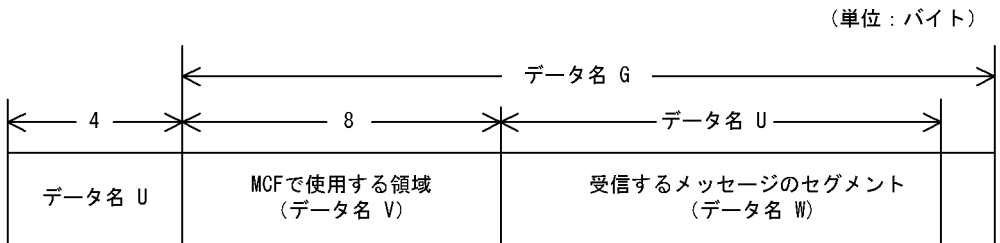
CBLDCMCF ('RECEIVE ') で受信できるメッセージの種類を次に示します。

- 相手システムから送信されたメッセージ
- MCF イベント
- アプリケーション起動機能で渡されたメッセージ

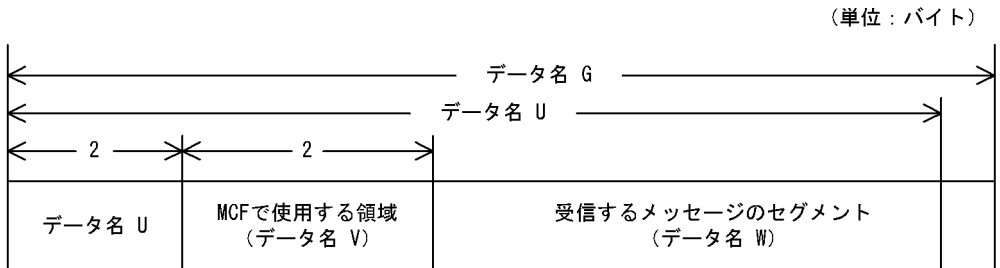
TP1/NET/X25-Extended を使用して通信する場合、相手システムから送信されるメッセージは、常に単一セグメントで構成されます。

セグメントを受信する領域 (一意名 3 で示す領域) の形式を次に示します。

- バッファ形式 1 の場合



- バッファ形式 2 の場合



UAP で値を設定するデータ領域

データ名 A

メッセージの受信を示す要求コードを「VALUE 'RECEIVE '」と設定します。

データ名 C

メッセージの先頭セグメントを受信するかどうかを設定します。次のどちらかを設定します。

VALUE 'FRST'

先頭セグメントを受信する場合や、メッセージが単一セグメントの場合に、「VALUE 'FRST'」を設定します。

VALUE 'SEG '

3. メッセージ送受信インタフェース
CBLDCMCF ('RECEIVE') - メッセージの受信 (COBOL 言語)

中間セグメントまたは最終セグメントを受信する場合に設定します。

データ名 D

空白を設定します。

データ名 G

セグメントを受信する領域の長さを設定します。

データ名 H, データ名 I, データ名 J, データ名 K, データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白を設定

省略されたものとして、「VALUE '1」(バッファ形式 1) が設定されます。

データ名 N

MCF で使用する領域です。

データ名 O

空白を設定します。

データ名 P

中間セグメントまたは最終セグメントを受信する場合、入力元の論理端末名称を設定します。先頭セグメントの受信時に返された論理端末名称を設定してください。

先頭セグメントの受信処理終了後、データ名 P には OpenTP1 から値が返されます。

データ名 Q

MCF で使用する領域です。

データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 V

【バッファ形式 1 の場合】 PIC X (8)

【バッファ形式 2 の場合】 PIC X (2)

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

データ名 B

ステータスコードが、5 けたの数字で返されます。

データ名 E

メッセージを受信した日付が「YYYYMMDD」(YYYY：西暦年 MM：月 DD：日)の形式で返されます。

データ名 F

メッセージを受信した時刻が「HHMMSS00」(HH：時 MM：分 SS：秒 00 は固定)の形式で返されます。

データ名 P

先頭セグメントを受信する場合、入力元の論理端末名称が返されます。

中間セグメントまたは最終セグメントを受信する場合は、ここで返された論理端末名称をデータ名 P に設定します。

データ名 U

【バッファ形式 1 の場合】 PIC 9 (9)

受信したセグメントの長さが返されます。

【バッファ形式 2 の場合】 PIC 9 (4)

受信したセグメントの長さ + 4 が返されます。

3. メッセージ送受信インタフェース
 CBLDCMCF ('RECEIVE') - メッセージの受信 (COBOL 言語)

データ名 W

受信したセグメントの内容が返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する CBLDCMCF ('RECEIVE ') を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、データ名 C に「VALUE 'SEG '」を設定して CBLDCMCF ('RECEIVE ') を呼び出してください。
71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する CBLDCMCF ('RECEIVE ') を呼び出しています。直前に呼び出した CBLDCMCF ('RECEIVE ') でメッセージはすべて受信しました。このステータスコードが返されたあとに、再び CBLDCMCF ('RECEIVE ') を呼び出した場合は、ステータスコード 72000 が返されます。
71002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。
71108	プロセスのローカルメモリが不足しています。
72000	MHP の実行でリターンした場合 <ul style="list-style-type: none"> 先頭セグメントを受信する CBLDCMCF ('RECEIVE ') を呼び出す前に、中間セグメントまたは最終セグメントを受信する CBLDCMCF ('RECEIVE ') を呼び出しています。先頭セグメントを受信する場合は、データ名 C に「VALUE 'FRST」を設定して CBLDCMCF ('RECEIVE ') を呼び出してください。 ステータスコード 71001 が返されたあとに、再び CBLDCMCF ('RECEIVE ') を呼び出しています。 SPP の実行でリターンした場合 SPP では CBLDCMCF ('RECEIVE ') を呼び出せません。
72001	データ名 P に設定した論理端末名称が間違っています。
72013	データ名 G の指定値を超えるセグメントを受信しました。データ名 G の指定値を超えた部分は切り捨てられました。 32763 バイトを超えるセグメントを受信しました。32763 バイトを超えた部分は切り捨てられました。
72016	データ名 D に設定した値が間違っています。 データ名 N またはデータ名 T に設定した値が間違っています。 データ名 M7 に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72025	データ名 C に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。

3. メッセージ送受信インタフェース
CBLDCMCF ('RECEIVE') - メッセージの受信 (COBOL 言語)

ステータスコード	意味
72036	データ名 G の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上, バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる, 予期しないエラーが発生しました。

CBLDCMCF ('RESEND') - 一方送信メッセージの再送 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

```
CALL 'CBLDCMCF' USING 一意名1 一意名2 一意名3
```

DATA DIVISION の指定

```
01 一意名1.  
  02 データ名A PIC X(8) VALUE 'RESEND' .  
  02 データ名B PIC X(5) .  
  02 FILLER PIC X(3) .  
  02 データ名C PIC X(4) VALUE SPACE .  
  02 データ名D PIC X(4) VALUE SPACE .  
  02 データ名E PIC 9(8) .  
  02 データ名F PIC 9(8) .  
  02 データ名G PIC 9(9) COMP VALUE ZERO .  
  02 データ名H PIC X(4) VALUE SPACE .  
  02 データ名I PIC X(4) VALUE SPACE .  
  02 データ名J PIC X(4) .  
  02 データ名K PIC X(4) .  
  02 データ名L PIC X(8) VALUE SPACE .  
  02 データ名M1 PIC X(4) VALUE SPACE .  
  02 データ名M2 PIC X(8) VALUE SPACE .  
  02 データ名M3 PIC X(4) VALUE SPACE .  
  02 データ名M4 PIC 9(9) COMP VALUE ZERO .  
  02 データ名M5 PIC 9(9) COMP VALUE ZERO .  
  02 データ名M6 PIC X(1) VALUE SPACE .  
  02 データ名M7 PIC X(1) VALUE SPACE .  
  02 データ名N PIC X(14) VALUE LOW-VALUE .  
01 一意名2.  
  02 データ名O PIC X(4) VALUE 'OUT' .  
  02 データ名P PIC X(8) .  
  02 データ名Q PIC X(8) VALUE SPACE .  
  02 データ名R PIC X(8) VALUE SPACE .  
  02 データ名S PIC X(28) VALUE LOW-VALUE .  
01 一意名3.  
  02 データ名T PIC X(8) .  
  02 データ名U PIC X(4) .  
  02 データ名V PIC 9(9) COMP .  
  02 データ名W PIC X(4) .  
  02 データ名X PIC X(12) VALUE LOW-VALUE .
```

機能

以前に送信したメッセージを、再び送信します。再送するメッセージは、以前に送信したメッセージとは別の、新しいメッセージとして扱います。どのメッセージを再送する

かは、次に示す送信済みメッセージの情報で選択できます。

- 出力先の論理端末名称
- メッセージ通番
- メッセージ種別 (一般の一方送信, 優先の一方送信)

対象としたメッセージが以前に送信されていない場合は、CBLDCMCF ('RESEND') はステータスコード 70904 を返します。また、メッセージキュー (ディスクキュー) 内に対象のメッセージがない場合も、ステータスコード 70904 を返します。このため、使用するメッセージキューの種別ではディスクキューを指定して、メッセージキューの大きさの定義では余裕を持った値を指定してください。

UAP で値を設定するデータ領域

データ名 A

一方送信メッセージの再送を示す要求コードを「VALUE 'RESEND」」と設定します。

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H, データ名 I

空白を設定します。

データ名 J

一般として再送するか優先として再送するかを設定します。

VALUE 'NORM'

一般の一方送信メッセージとして再送する場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージとして再送する場合に設定します。

空白を設定

省略されたものとして、「VALUE 'NORM'」(一般の一方送信メッセージとして再送) が設定されます。

3. メッセージ送受信インタフェース

CBLDCMCF ('RESEND') - 一方送信メッセージの再送 (COBOL 言語)

データ名 K

再送するメッセージに出力通番を付け直すかどうかを設定します。

VALUE 'SEQ' '

再送するメッセージに出力通番を付け直す場合に設定します。

VALUE 'NSEQ'

再送するメッセージに出力通番を付け直さない場合に設定します。

空白を設定

省略されたものとして、「VALUE 'NSEQ」(出力通番を付け直さない)が設定されます。

データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

データ名 M4, データ名 M5

0を設定します。

データ名 M6, データ名 M7

空白を設定します。

データ名 N

MCF で使用する領域です。

データ名 O

一方送信を示す「VALUE 'OUT' 」を設定します。

データ名 P

出力先の論理端末名称を設定します。論理端末名称の長さは最大 8 バイトです。8 バイトに満たない名称を設定する場合は、後ろを空白で埋めてください。

データ名 Q, データ名 R

空白を設定します。

データ名 S

MCF で使用する領域です。

データ名 T

再送するメッセージを検索するキーとして、以前に送信したメッセージの出力先の論理端末名称を設定します。論理端末名称の長さは最大 8 バイトです。8 バイトに満たない名称を設定する場合は、後ろを空白で埋めてください。

データ名 U

再送するメッセージを検索するキーとして、以前に送信したメッセージの送信種別を設定します。

VALUE 'NORM'

一般の一方送信メッセージを対象とする場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージを対象とする場合に設定します。

空白を設定

省略されたものとして、「VALUE 'NORM'」(一般の一方送信メッセージを対象)が設定されます。

「VALUE 'PRIO'」を設定した場合は、データ名 T に出力先の論理端末名称を設定できません。

データ名 V

再送するメッセージを検索するキーとして、以前に送信したメッセージの出力通番を設定します。データ名 W に「VALUE 'LAST'」を設定した場合は、ここに設定した値は無効となります。

データ名 W

最終出力通番を持つメッセージを再送するかどうかを設定します。

VALUE 'LAST'

最終出力通番を持つメッセージを再送する場合に設定します。この値を設定した場合は、データ名 V に設定した値は無効となります。

空白を設定

データ名 V で設定した出力通番を持つメッセージを再送する場合に設定します。

データ名 X

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

データ名 B

ステータスコードが、5 けたの数字で返されます。

3. メッセージ送受信インタフェース

CBLDCMCF ('RESEND') - 一方送信メッセージの再送 (COBOL 言語)

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
70904	該当するメッセージがありません。
70905	再送するメッセージのセグメントの長さが、UAP 共通定義 (mcfmuap-re) で指定した値を超えています。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	MCF が終了処理中のため、メッセージの再送を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージキューから取り出したメッセージを格納するバッファ (作業領域) を、メモリ上に確保できませんでした。
71108	メッセージを再送しようとしたのですが、再送先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	MHP の実行でリターンした場合 <ul style="list-style-type: none"> 先頭セグメントを受信する CBLDCMCF ('RECEIVE ') を呼び出す前に、CBLDCMCF ('RESEND ') を呼び出しています。 非トランザクション属性の MHP から、CBLDCMCF ('RESEND ') を呼び出しています。
	SPP の実行でリターンした場合 トランザクションでない SPP の処理から、CBLDCMCF ('RESEND ') を呼び出しています。
72001	データ名 P またはデータ名 T に設定した論理端末名称が間違っています。
	CBLDCMCF ('RESEND ') を呼び出せない論理端末を設定しています。
72016	データ名 J に設定した値が間違っています。
	データ名 M4 に設定した値が間違っています。
	データ名 U に設定した値が間違っています。
	データ名 N、データ名 S、またはデータ名 X に設定した値が間違っています。
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

注意事項

メッセージの再送時には、MCF マネージャ定義の UAP 共通定義 (mcfmuap) の -e オプションと -l オプションの指定値に注意してください。

-e オプション

-e オプションでは、CBLDCMCF ('RESEND ') で使用する作業領域の大きさを指定します。再送するメッセージのセグメントがこの作業領域より大きい場合、CBLDCMCF ('RESEND ') はメッセージを再送しないで、ステータスコード 70905 を返します。このため、-e オプションでは、セグメントの最大長よりも大きな値を設定しておいてください。

-l オプション

-l オプションでは、通番に関して指定します。この内容によっては、メッセージキューファイル内に同じ通番を持つメッセージが同時に存在することがあります。この場合は、どのメッセージを再送するか保証できません。

CBLDCMCF('SEND ') - 一方送信メッセージの送信 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

```
CALL 'CBLDCMCF' USING 一意名1 一意名2 一意名3
```

DATA DIVISION の指定

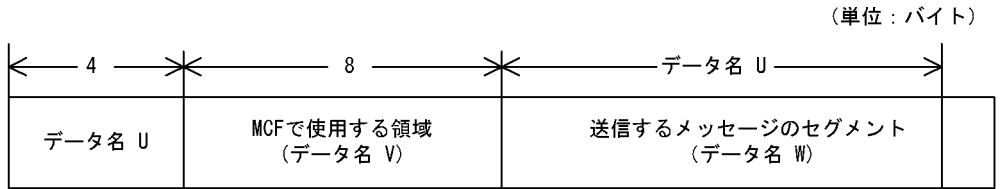
```
01 一意名1.  
  02 データ名A PIC X(8) VALUE 'SEND '  
  02 データ名B PIC X(5).  
  02 FILLER PIC X(3).  
  02 データ名C PIC X(4) VALUE SPACE.  
  02 データ名D PIC X(4) VALUE SPACE.  
  02 データ名E PIC 9(8).  
  02 データ名F PIC 9(8).  
  02 データ名G PIC 9(9) COMP VALUE ZERO.  
  02 データ名H PIC X(4).  
  02 データ名I PIC X(4) VALUE SPACE.  
  02 データ名J PIC X(4).  
  02 データ名K PIC X(4).  
  02 データ名L PIC X(8) VALUE SPACE.  
  02 データ名M1 PIC X(4) VALUE SPACE.  
  02 データ名M2 PIC X(8) VALUE SPACE.  
  02 データ名M3 PIC X(4) VALUE SPACE.  
  02 データ名M4 PIC 9(9) COMP VALUE ZERO.  
  02 データ名M5 PIC 9(9) COMP VALUE ZERO.  
  02 データ名M6 PIC X(1) VALUE SPACE.  
  02 データ名M7 PIC X(1).  
  02 データ名N PIC X(14) VALUE LOW-VALUE.  
01 一意名2.  
  02 データ名O PIC X(4) VALUE 'OUT '  
  02 データ名P PIC X(8).  
  02 データ名Q PIC X(8) VALUE SPACE.  
  02 データ名R PIC X(8) VALUE SPACE.  
  02 データ名T PIC X(28) VALUE LOW-VALUE.  
01 一意名3.  
  02 データ名U PIC 9(x) COMP.  
  02 データ名V PIC X(x).  
  02 データ名W PIC X(n).
```

機能

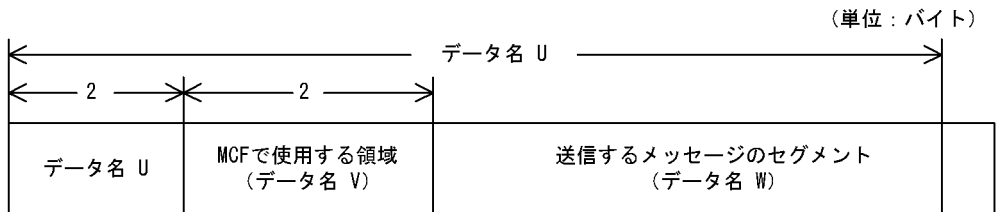
相手システムへ一方送信メッセージを送信します。一方送信メッセージは、一つのセグメントで構成されます。

セグメントを送信する領域 (一意名 3 で示す領域) の形式を次に示します。

• バッファ形式 1 の場合



• バッファ形式 2 の場合



UAP で値を設定するデータ領域

データ名 A

一方送信メッセージの送信を示す要求コードを「VALUE 'SEND '」と設定します。

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H

単一セグメントの送信を示す「VALUE 'EMI '」を設定します。

データ名 I

空白を設定します。

3. メッセージ送受信インタフェース

CBLDCMCF ('SEND') - 一方送信メッセージの送信 (COBOL 言語)

データ名 J

一般として送信するか優先として送信するかを設定します。

VALUE 'NORM'

一般の一方送信メッセージとして送信する場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージとして送信する場合に設定します。

空白を設定

省略されたものとして、「VALUE 'NORM'」(一般の一方送信メッセージとして送信)が設定されます。

データ名 K

出力通番を付けるかどうかを設定します。

VALUE 'SEQ'

出力通番が必要な場合に設定します。

VALUE 'NSEQ'

出力通番が必要ない場合に設定します。

空白を設定

省略されたものとして、「VALUE 'NSEQ'」(出力通番は付けない)が設定されます。

データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白を設定

省略されたものとして、「VALUE '1'」(バッファ形式 1)が設定されます。

データ名 N

MCF で使用する領域です。

データ名 O

一方送信を示す「VALUE 'OUT '」を設定します。

データ名 P

出力先の論理端末名称を設定します。論理端末名称の長さは最大 8 バイトです。8 バイトに満たない名称を設定する場合は、後ろを空白で埋めてください。

データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 U

【バッファ形式 1 の場合】 PIC 9 (9)

送信するセグメントの長さを設定します。

【バッファ形式 2 の場合】 PIC 9 (4)

送信するセグメントの長さ + 4 を設定します。

データ名 V

【バッファ形式 1 の場合】 PIC X (8)

【バッファ形式 2 の場合】 PIC X (2)

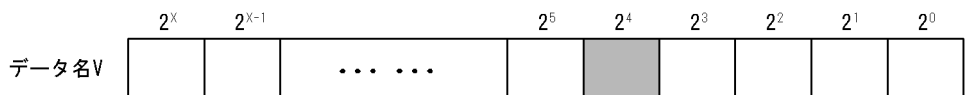
領域の 2^4 ビットに 1 を設定

送信完了通知イベントを通知させます。ただし、送信完了通知イベント処理用の MHP を MCF アプリケーション定義で指定していない場合は無効です。

領域の 2^4 ビットに 0 を設定

送信完了通知イベントを通知させません。

データ名 V とビットの設定位置の関係を次に示します。



(凡例) : 送信完了通知イベントの可否を設定するビット

注

3. メッセージ送受信インタフェース

CBLDCMCF ('SEND') - 一方送信メッセージの送信 (COBOL 言語)

X の値は次のとおりです。

- バッファ形式 1 の場合 X=63
- バッファ形式 2 の場合 X=15

データ名 W

送信するセグメントの内容を設定します。一つのセグメントで 32000 バイトまで送信できます。

OpenTP1 から値が返されるデータ領域

データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	データ名 U に 32000 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	MHP の実行でリターンした場合 先頭セグメントを受信する CBLDCMCF ('RECEIVE ') を呼び出す前に、 CBLDCMCF ('SEND ') を呼び出しています。
	SPP の実行でリターンした場合 トランザクションでない SPP の処理から、CBLDCMCF ('SEND ') を呼び出しています。
72001	データ名 P に設定した論理端末名称が間違っています。
	CBLDCMCF ('SEND ') を呼び出せない論理端末を設定していません。
72016	データ名 N またはデータ名 T に設定した値が間違っています。
	データ名 J に設定した値が間違っています。
	データ名 M1 に設定した値が間違っています。
	データ名 M7 に設定した値が間違っています。

3. メッセージ送受信インタフェース
CBLDCMCF ('SEND') - 一方送信メッセージの送信 (COBOL 言語)

ステータスコード	意味
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
72020	データ名 I に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72026	データ名 H に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72041	バッファ形式 1 の場合はデータ名 U に 0 以下の値を設定しています。バッファ形式 2 の場合はデータ名 U に 4 以下の値を設定しています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

RECEIVE - メッセージの受信 (データ操作言語)

形式

DATA DIVISION (通信記述項) の指定

CD 通信記述名

FOR INPUT

{STATUS KEY IS データ名1}

{SYMBOLIC TERMINAL IS データ名2}

{MESSAGE DATE IS データ名3}

{MESSAGE TIME IS データ名4} .

PROCEDURE DIVISION (通信文) の指定

RECEIVE 通信記述名

{FIRST} SEGMENT

INTO 一意名1 .

機能

次に示す CALL インタフェースの機能を実現します。

- メッセージの受信 CBLDCMCF ('RECEIVE ')

通信記述項に設定する項目

FOR 句

次の値を指定します。

INPUT

非同期型のメッセージの受信

STATUS KEY 句

ステータスコードを受け取りたい場合に指定します。省略した場合は、ステータスコードを受け取りません。

SYMBOLIC TERMINAL 句

入力元の論理端末名称を参照するデータ項目を指定します。

MESSAGE DATE 句

メッセージを受信した日付を参照するデータ項目を指定します。「YYMMDD」(YY:西暦の下2けた MM:月 DD:日)の形式で参照できます。

MESSAGE TIME 句

メッセージを受信した時刻を参照するデータ項目を指定します。「HHMMSS00」(HH : 時 MM : 分 SS : 秒 00 は固定)の形式で参照できます。

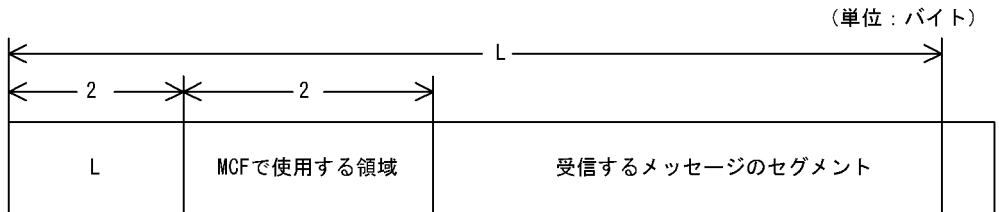
通信文に指定する項目

FIRST

先頭セグメントを受信する場合に指定します。

一意名 1

セグメントを受信するデータ項目を指定します。一意名 1 の形式を次に示します。



ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する RECEIVE 文を 2 回以上実行しています。中間セグメントまたは最終セグメントを受信する場合は、FIRST を指定しないで RECEIVE 文を実行してください。
71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する RECEIVE 文を実行しています。以前に実行した RECEIVE 文でそのメッセージはすべて受信しました。 このステータスコードが返されたあとに、再び RECEIVE 文を実行した場合は、ステータスコード 72000 が返されます。
71002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。 メッセージキューが割り当てられていません。 MCF が終了処理中のため、メッセージの受信を受け付けられません。
71108	メッセージ受信に必要な管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。

3. メッセージ送受信インタフェース

RECEIVE - メッセージの受信 (データ操作言語)

ステータスコード	意味
72000	MHP の実行でリターンした場合 <ul style="list-style-type: none"> 先頭セグメントを受信する RECEIVE 文を実行する前に、中間セグメントおよび最終セグメントを受信する RECEIVE 文を実行しています。先頭セグメントを受信する場合は、FIRST を指定して RECEIVE 文を実行してください。 ステータスコード 71000 が返されたあとで、RECEIVE 文を実行していません。
	SPP の実行でリターンした場合 SPP では RECEIVE 文を実行できません。
72001	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。
	RECEIVE 文を実行できない論理端末を設定しています。
72013	一意名 1 の L の指定値を超えるセグメントを受信しました。一意名 1 の L の指定値を超えた部分は切り捨てられました。
	32763 バイトを超えるセグメントを受信しました。32763 バイトを超えた部分は切り捨てられました。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72036	一意名 1 の L の指定値が不足しています。5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

SEND - 一方送信メッセージの送信 (データ操作言語)

形式

DATA DIVISION (通信記述項) の指定

```
CD 通信記述名  
FOR OUTPUT  
[STATUS KEY IS データ名1]  
[SYMBOLIC TERMINAL IS データ名2]  
[SYNCHRONOUS MODE IS { ASYNC | データ名6 }]  
[SWITCHING MODE IS { NORMAL | PRIOR | データ名7 }]  
[DETAIL MODE IS データ名10].
```

PROCEDURE DIVISION (通信文) の指定

```
SEND 通信記述名 FROM 一意名1  
[WITH { EMI | 一意名2 }].
```

機能

次に示す CALL インタフェースの機能を実現します。単一セグメントだけ扱えます。

- 一方送信メッセージの送信 CBLDCMCF ('SEND ')

通信記述項に設定する項目

FOR 句

次の値を指定します。

OUTPUT

一方送信メッセージの送信

STATUS KEY 句

ステータスコードを受け取りたい場合に指定します。省略した場合は、ステータスコードを受け取りません。

SYMBOLIC TERMINAL 句

論理端末名称を設定したデータ項目を指定します。

SYNCHRONOUS MODE 句

非同同期型でメッセージを送信することを、次のどちらかで指定します。

ASYNC

3. メッセージ送受信インタフェース

SEND - 一方送信メッセージの送信 (データ操作言語)

非同期型のメッセージ送信

一方送信メッセージの送信の場合、指定します。

データ名 6

次の値を設定したデータ項目

'0': 非同期型のメッセージ送信

SYNCHRONOUS MODE 句は、省略してもかまいません。

SWITCHING MODE 句

一般か優先かを指定します。

NORMAL

一般の一方送信メッセージ

PRIOR

優先の一方送信メッセージ

データ名 7

次の値を設定したデータ項目

'0'または' ': 一般の一方送信メッセージ

'1': 優先の一方送信メッセージ

省略した場合は、一般の一方送信メッセージ (NORMAL) が設定されます。

DETAIL MODE 句

出力通番を付けるかどうかを指定します。

データ名 10

次の値を設定したデータ項目

'0'または' ': 出力通番を付けます。

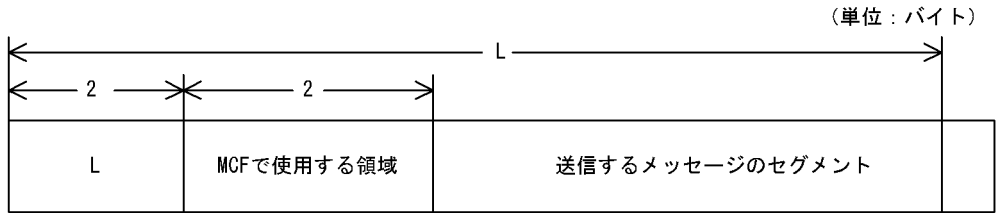
'1': 出力通番を付けません。

省略した場合は、出力通番を付けません。

通信文に指定する項目

一意名 1

セグメントを送信するデータ項目を指定します。一つのセグメントで 32000 バイトまで送信できます。一意名 1 の形式を次に示します。



また, MCF で使用する領域の 2^4 ビット目に, 送信完了通知イベントを通知させるかどうかを指定してください。

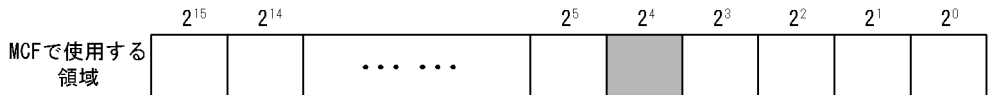
領域の 24 ビットに 1 を設定

送信完了通知イベントを通知させます。ただし, 送信完了通知イベント処理用の MHP を MCF アプリケーション定義で指定していない場合は無効です。

領域の 24 ビットに 0 を設定

送信完了通知イベントを通知させません。

一意名 1 の MCF で使用する領域とビットの設定位置の関係を次に示します。



(凡例) : 送信完了通知イベントの可否を設定するビット

WITH 句

単一セグメントのメッセージを送信することを, 次のどちらかで指定します。

EMI

単一セグメントの送信

一意名 2

次の値を設定したデータ項目

'2': EMI (単一セグメントの送信)

WITH 句は, 省略してもかまいません。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。

3. メッセージ送受信インタフェース

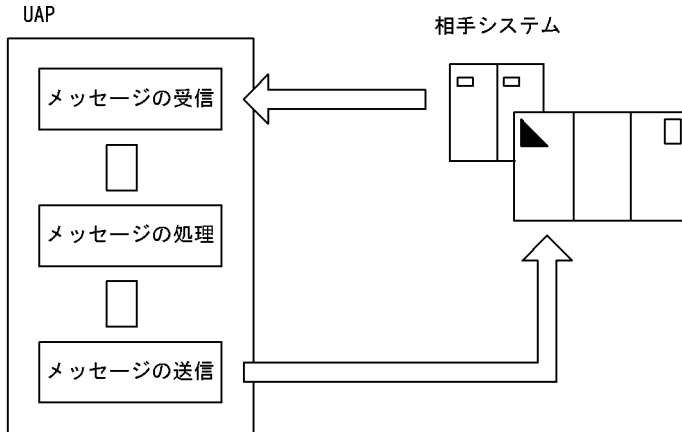
SEND - 一方送信メッセージの送信 (データ操作言語)

ステータスコード	意味
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	一意名 1 の L に 32000 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71008	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	MHP の実行でリターンした場合 先頭セグメントを受信する RECEIVE 文を実行する前に、SEND 文を実行しています。
	SPP の実行でリターンした場合 トランザクションでない SPP の処理から、SEND 文を実行しています。
72001	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。
	SEND 文を実行できない論理端末を設定しています。
72017	DETAIL MODE 句に設定した値が間違っています。
72018	SWITCHING MODE 句に設定した値が間違っています。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72026	WITH 句に設定した値が間違っています。
72041	一意名 1 の L に 4 以下の値を設定しています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

ユーザアプリケーションプログラムの作成例

UAP の作成例の処理の流れを図 3-1 に示します。また、図 3-1 のコーディング例を、C 言語、COBOL 言語、およびデータ操作言語でそれぞれ示します。

図 3-1 UAP 作成例の処理の流れ



コーディング例 (C 言語)

C 言語 (K&R 版 C) を使用した UAP のコーディング例を次に示します。また、このコーディング例を、/BeTRAN/examples/mcf/X25EX/aplib/c/ap.c のファイルで提供しています。

```
#include <dcmcf.h>

void    ex_uapl()
{
    /* 引数定義 */
    char                termnam[16];
    char                mapname[16];
    char                sdataarea[2048];
    char                rdataarea[2048];
    DCLONG              sdataleng;
    DCLONG              rdataleng;
    DCLONG              time;
    char                resv02[9];

    int                rtn;

    memset(mapname, 0, 16);
    memset(sdataarea, 0, 2048);
    memset(rdataarea, 0, 2048);
    memset(resv02, 0, 16);

    /*   メッセージ受信開始   receive関数の呼び出し   */
```

3. メッセージ送受信インタフェース ユーザアプリケーションプログラムの作成例

```

rtn = dc_mcf_receive( DCMCFRST,
                    DCNOFLAGS,
                    termnam,
                    mapname,
                    rdataarea,
                    &rdata Leng,
                    2048,
                    &time );

/
*****
***/
/*
/*          データの処理          */
/*
/*
/
*****
***/

/* メッセージ送信開始      send関数の呼び出し */

rtn = dc_mcf_send   ( DCMCFEMI,
                    DCMCFOUT,
                    termnam,
                    mapname,
                    sdataarea,
                    sdata Leng,
                    resv02,
                    DCNOFLAGS );

}

```

コーディング例 (COBOL 言語)

COBOL 言語を使用した UAP のコーディング例を次に示します。また、このコーディング例を、/BeTRAN/examples/mcf/X25EX/aplib/cobol/ap.cbl のファイルで提供していません。

```

IDENTIFICATION      DIVISION.
PROGRAM-ID.         UAPCBL.

ENVIRONMENT         DIVISION.

DATA                DIVISION.
WORKING-STORAGE    SECTION.

01 RCV.
   02 MSG-REQ       PIC X(8)    VALUE 'RECEIVE'.
   02 RTN           PIC X(5).
   02 FILLER        PIC X(3).
   02 RSV1          PIC X(4)    VALUE 'FRST'.

```

3. メッセージ送受信インタフェース
ユーザアプリケーションプログラムの作成例

```

02 RSV2                PIC X(4)    VALUE  SPACE.
02 MCFUSING1          PIC 9(8).
02 MCFUSING2          PIC 9(8).
02 RSV3                PIC 9(9)    COMP   VALUE  2048.
02 SEGKIND            PIC X(4)    VALUE  SPACE.
02 RSV4                PIC X(4)    VALUE  SPACE.
02 MSGKIND            PIC X(4)    VALUE  SPACE.
02 OUTPUTNO           PIC X(4)    VALUE  SPACE.
02 RSV5                PIC X(8)    VALUE  SPACE.
02 RSV6                PIC X(4)    VALUE  SPACE.
02 RSV7                PIC X(8)    VALUE  SPACE.
02 RSV8                PIC X(4)    VALUE  SPACE.
02 RSV9                PIC 9(9)    COMP   VALUE  ZERO.
02 RSV10              PIC 9(9)    COMP   VALUE  ZERO.
02 RSV11              PIC X(1)    VALUE  SPACE.
02 RSV12              PIC X(1)    VALUE  SPACE.
02 RSV13              PIC X(14)   VALUE  LOW-VALUE.
01 CD1.
02 SEG-CODE1          PIC X(4)    VALUE  SPACE.
02 TERM-CODE          PIC X(8)    VALUE  SPACE.
02 MCFUSE14           PIC X(8)    VALUE  SPACE.
02 MCFUSE15           PIC X(8)    VALUE  SPACE.
02 MCFUSE16           PIC X(28)   VALUE  LOW-VALUE.
01 DATA1.
02 MSGSEG-LENG1      PIC 9(9)    COMP.
02 MCFUSE17           PIC X(4).
02 MCFUSE18           PIC X(2).
02 MCFUSE19           PIC X(1).
02 MCFUSE20           PIC X(1).
02 REC-MSGSEG1       PIC X(2048).
01 SND.
02 MSG-SEND1          PIC X(8)    VALUE  'SEND  '.
02 STATUS-CODE3      PIC X(5).
02 FILLER             PIC X(3).
02 MCFUSE41           PIC X(4)    VALUE  SPACE.
02 MCFUSE42           PIC X(4)    VALUE  SPACE.
02 MCFUSE43           PIC 9(8).
02 MCFUSE44           PIC 9(8).
02 MCFUSE45           PIC 9(9)    COMP   VALUE  ZERO.
02 SEND-SEG1         PIC X(4)    VALUE  'EMI  '.
02 MCFUSE46           PIC X(4)    VALUE  SPACE.
02 SEND-NORM         PIC X(4)    VALUE  'NORM'.
02 SEND-SEQ1         PIC X(4)    VALUE  'NSEQ'.
02 MCFUSE47           PIC X(8)    VALUE  SPACE.
02 SEND-CODE1        PIC X(4)    VALUE  SPACE.
02 MCFUSE48           PIC X(8)    VALUE  SPACE.
02 MCFUSE49           PIC X(4)    VALUE  SPACE.
02 MCFUSE50           PIC 9(9)    COMP   VALUE  ZERO.
02 MCFUSE51           PIC 9(9)    COMP   VALUE  ZERO.
02 MCFUSE52           PIC X(1)    VALUE  SPACE.
02 MCFUSE53           PIC X(1)    VALUE  SPACE.
02 MCFUSE54           PIC X(14)   VALUE  LOW-VALUE.
01 CD2.
02 SENDSEG-CODE2     PIC X(4)    VALUE  'OUT  '.
02 TERM-CODE          PIC X(8)    VALUE  'INTLE01'.
02 MCFUSE55           PIC X(8)    VALUE  SPACE.

```

3. メッセージ送受信インタフェース ユーザアプリケーションプログラムの作成例

```

02 MCFUSE56          PIC X(8)    VALUE    SPACE.
02 MCFUSE57          PIC X(28)   VALUE    LOW-VALUE.
01  DATA2.
02 MSGSEG-LENG2     PIC 9(9)    COMP     VALUE 512.
02 MCFUSE58          PIC X(8)    .
02 REC-MSGSEG2      PIC X(512)  .

PROCEDURE          DIVISION.

*****
*****
*                                     *
*                               メッセージ受信                               *
*                                     *
*****
CALL 'CBLDCMCF'    USING    RCV    CD1    DATA1.
*
*
*****
*****
*                                     *
*                               データの処理                               *
*                                     *
*****
*****
*
*
*****
*****
*                                     *
*                               メッセージ送信                               *
*                                     *
*****
CALL 'CBLDCMCF'    USING    SND    CD2    DATA2.

EXIT PROGRAM.

```

コーディング例（データ操作言語）

データ操作言語を使用した UAP のコーディング例を次に示します。また、このコーディング例を、/BeTRAN/examples/mcf/X25EX/aplib/dml/ap.cbl のファイルで提供しています。

```

*****
*****
*                                     *
*                               MHPサービスプログラム                               *
*                                     *
*****
*
IDENTIFICATION DIVISION.

PROGRAM-ID. UAPDML.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*
*****

```

```

*****
*
*                               ワーク変数                               *
*****
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
*****
*
*                               一方受信領域                               *
*****
*
01   RECV-AREA1.
      02 RE-DATALENG1          PIC 9(4)  COMP VALUE 1028.
      02 RE-RSV1              PIC X(2) .
      02 RE-DATA1             PIC X(1024).
*
*****
*
*                               一方送信領域                               *
*****
*
01   SEND-AREA1.
      02 SE-DATALENG1          PIC 9(4)  COMP VALUE 1028.
      02 SE-RSV1              PIC X(2) .
      02 SE-DATA1             PIC X(1024).
*
*****
*
*                               通信記述項                               *
*****
*
COMMUNICATION SECTION.
*
*****
*
*                               一方受信                               *
*****
*
      CD RECV-IN1
      FOR INPUT
      STATUS KEY IS           RE-STATUS1
      SYMBOLIC TERMINAL IS   RE-TERMNAM1
      MESSAGE DATE IS        RE-DATE1
      MESSAGE TIME IS        RE-TIME1.
*
*****
*
*                               一方送信                               *
*****
*
      CD SEND-IN1

```

3. メッセージ送受信インタフェース ユーザアプリケーションプログラムの作成例

```
FOR OUTPUT
STATUS KEY IS          SE-STATUS1
SYMBOLIC TERMINAL IS  SE-TERMNAM
SYNCHRONOUS MODE IS   ASYNC
SWITCHING MODE IS     NORMAL.

*
*
*****
*****
*
*                               通信文                               *
*****
*****
*
PROCEDURE DIVISION.
*
*****
*****
*
*                               一方受信                               *
*****
*****
*
RECEIVE  RECV-IN1
FIRST   SEGMENT
INTO    RECV-AREA1.

*
*****
*****
*
*                               データの処理                               *
*****
*****
*
*                               一方送信                               *
*****
*****
*
SEND  SEND-IN1
FROM  SEND-AREA1
WITH  EMI.

*
EXIT PROGRAM.
```


4

ユーザOWNコーディング， MCF イベントインタフェース

この章では，TP1/NET/X25-Extendedに関連するユーザOWNコーディング，およびMCF イベントのインタフェースについて説明します。

4.1 ユーザOWNコーディングインタフェース

4.2 MCF イベントインタフェース

4.1 ユーザOWNコーディングインタフェース

メッセージを送受信する UAP を，より多様な業務に対応させるために補助するプログラムを，ユーザOWNコーディング（以降，UOC と略します）といいます。

TP1/NET/X25-Extended で使用できる UOC を次に示します。

- 入力メッセージ編集 UOC
- 出力メッセージ編集 UOC
- 送信メッセージの通番編集 UOC

UOC を使用する場合は，あらかじめ MCF メイン関数または UAP のメイン関数に UOC 関数のアドレスを登録し，UOC 関数のオブジェクトファイルを MCF 通信プロセスまたは UAP の実行形式プログラムに結合（リンケージ）しておく必要があります。また，UOC は C 言語で作成します。

4.1.1 入力メッセージの編集とアプリケーション名の決定

入力メッセージ編集 UOC は，受信したメッセージをユーザ任意の形式に変換します。そして受信した論理メッセージを基に，ユーザは任意のアプリケーション名を決定できます。

UOC は，MHP を起動するメッセージのセグメントを受信すると起動します。ただし，MCF イベント発生時と，UAP からのアプリケーションプログラム起動時はこの UOC は起動しません。

ユーザは，MCF メイン関数で UOC 関数アドレスを設定します。また，必要に応じて MCF 通信構成定義（`mcftalcn -e`）で，メッセージ編集用バッファグループ番号を定義します。

（1）入力メッセージの編集

受信したメッセージが格納されている受信バッファ，および MCF 通信構成定義で指定した編集バッファを引き渡します。UOC では，これらのバッファを使用して，メッセージを編集できます。

また，UAP に通知するメッセージのセグメントは，受信バッファ，または編集バッファのどちらかに格納されたものを使用できます。どちらのセグメントを使用するかは，UOC から返されるリターンコードによって選択できます。

各パラメタに設定した値が不正の場合，TP1/NET/X25-Extended は MCF イベントを通知してコネクションを解放します。

（2）アプリケーション名の決定

受信したメッセージの内容を基に，そのメッセージを処理するアプリケーションの名称

を決定できます。

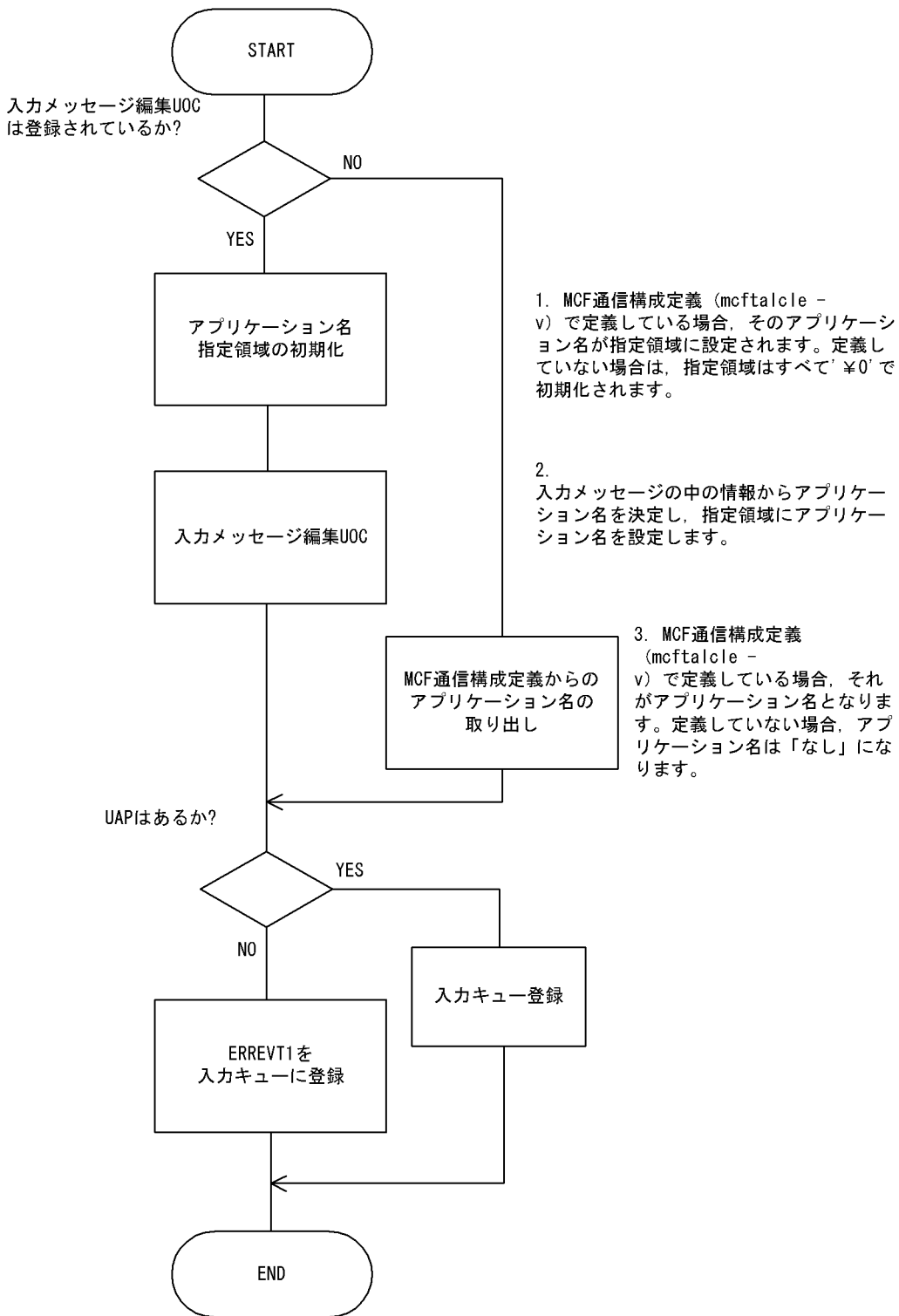
UOCでアプリケーション名を決定しない場合, MCF 通信構成定義 (mcftalcle -v) で指定したアプリケーション名が採用されます。ユーザは UOC か MCF 通信構成定義のどちらかで, 必ずアプリケーション名を決定してください。

アプリケーション名は, '¥0' で終わる 1 ~ 8 バイトの英数字で指定します。決定したアプリケーション名は, アプリケーション名格納領域に設定してください。先頭から 9 バイト目までに '¥0' がないときは, アプリケーション名の不正となります。このとき, 不正アプリケーション名検出通知イベント (ERREVT1) が UAP に通知されます。

アプリケーション名の決定の流れを図 4-1 に示します。

4. ユーザOWNコーディング, MCF イベントインタフェース

図 4-1 アプリケーション名の決定の流れ



(3) UOC エラーリターン処理

UOC から DCMCF_UOC_MSG_NG でリターンした場合，MCF はメッセージログを出力し，障害通知イベント (CERREVT) を通知します。

UOC で障害を検出し，エラー処理 UAP を起動したい場合は，ユーザ任意のエラー処理 UAP のアプリケーション名を設定します。このとき，MCF には DCMCF_UOC_MSG_OK，または DCMCF_UOC_MSG_OK_RCV でリターンします。この場合，MCF は正常なメッセージとして処理するため，受信メッセージの破棄などの障害処理はしません。

(4) UOC パラメタ不正の場合の処理

UOC で設定した値に不正があった場合，MCF はメッセージログを出力し，障害通知イベント (CERREVT) を通知します。

(5) OpenTP1 への組み込み方法

スタート関数 (dc_mcf_svstart) を発行する MCF メイン関数に，作成した UOC の関数アドレスを設定します。入力メッセージ編集 UOC の関数アドレスは任意に決められます。UOC 関数をコンパイルして生成した UOC オブジェクトファイルを，UOC 関数を登録した MCF メイン関数と結合して，TP1/NET/X25-Extended の実行形式プログラムを生成します。MCF メイン関数の詳細については，「7.2 MCF メイン関数の作成」を参照してください。

4.1.2 入力メッセージ編集 UOC インタフェース

入力メッセージ編集 UOC は，次の形式で呼び出します。

(1) 形式

ANSI C，C++ の場合

```
#include <dcmcfuoc.h>
DCLONG uoc_func(dcmcf_uoc_min_n *parm)
```

K&R 版 C の場合

```
#include <dcmcfuoc.h>
DCLONG uoc_func(parm)

dcmcf_uoc_min_n *parm;
```

(2) 説明

uoc_func (入力メッセージ編集 UOC) を呼び出すとき，MCF は次の所定のパラメタを

4. ユーザOWNコーディング, MCF イベントインタフェース

parm に設定します。

(3) パラメタの内容

(a) dcmcf_uoc_min_n の内容

```
typedef struct {
    DCLONG pro_kind;           ... プロトコル種別
    char   le_name[9];         ... 論理端末名称
    char   reserve1[7];        ... 予備
    DCLONG rcv_prim;           ... 受信サービスプリミティブ
    dcmcf_uocbuff_list_n *buflist_adr; ... 受信バッファリストアドレス
    dcmcf_uocbuff_list_n *ebuflist_adr; ... 編集バッファリストアドレス
    char   aplname[9];         ... アプリケーション名
    char   reserve2[7];        ... 予備
    char   *pro_indv_ifa;       ... MCF使用領域
    DCLONG rtn_detail;         ... 詳細リターンコード
    char   reserve3[8];        ... 予備
} dcmcf_uoc_min_n;
```

(b) dcmcf_uocbuff_list_n (バッファリスト) の内容

```
typedef struct {
    DCLONG buf_num;           ... バッファ情報数
    DCLONG used_buf_num;      ... 使用バッファ情報数
    char   reserve1[8];        ... 予備
    dcmcf_uocbufinf_n buf_array[DCMCF_UOC_BUFF_MAX]; ... バッファ情報
} dcmcf_uocbuff_list_n;
```

(c) dcmcf_uocbufinf_n (バッファ情報) の内容

```
typedef struct {
    char   *buf_adr;           ... バッファアドレス
    DCULONG buf_size;          ... バッファ最大長
    DCULONG seg_size;         ... バッファ使用長
    char   reserve1[4];        ... 予備
    dcmcfuoc w type buff_id;   ... MCF内部情報1
    DCMLONG buff_adr;         ... MCF内部情報2
    char   reserve2[4];        ... 予備
} dcmcf_uocbufinf_n;
```

(4) MCF が値を設定する項目

(a) dcmcf_uoc_min_n

pro_kind

プロトコル種別として, 次の値が設定されます。

DCMCF_UOC_PRO_X25EX

X.25 プロトコル (TP1/NET/X25-Extended)

le_name

メッセージを入力した論理端末の名称が設定されます。

rcv_prim

受信サービスプリミティブとして，次の値が設定されます。

DCMCF_UOC_RCV_BRD

一方送信メッセージの受信

buflist_adr

受信用バッファリストのアドレスが設定されます。

ebuflist_adr

編集用バッファリストのアドレスが設定されます。

メッセージ編集バッファが未定義の場合，つまり MCF 通信構成定義の mcftalcn コマンドの -e オプションを省略した場合，ebuflist_adr には NULL が設定されます。

aplname

MCF 通信構成定義の mcftalcle コマンドの -v オプションで指定したアプリケーション名が設定されます。このアプリケーション名を定義していない場合，領域はすべて '¥0' で初期化されます。

pro_indv_ifa

MCF で使用するパラメタです。

(b) dcmcf_uocbuff_list_n (バッファリスト)

buf_num

バッファ情報の数として 1 が設定されます。

buf_array

バッファ情報の配列が設定されます。バッファ情報は，buf_num の数だけ設定されず。

(c) dcmcf_uocbufinf_n (バッファ情報)

buf_adr

バッファのアドレスが設定されます。

buf_size

バッファの最大長が設定されます。

seg_size

送信，または受信用バッファリストの場合だけ，バッファの使用長が設定されます。

buff_id, buff_addr

MCF で使用するパラメタです。

4. ユーザOWNコーディング, MCF イベントインタフェース

(5) ユーザが値を設定する項目

(a) dcmcf_uoc_min_n

aplname

UOC で決定したアプリケーション名を設定します。

rtn_detail

詳細リターンコードを設定します。

このコードは, UOC が DCMCF_UOC_MSG_NG をリターンしたときに, MCF に渡されます。MCF は, 詳細リターンコードをメッセージログファイルに出力します。

詳細リターンコードは, -19999 から -19000 の範囲で設定してください。

(b) dcmcf_uocbuff_list_n (バッファリスト)

used_buf_num

使用したバッファ情報の数を設定します。

使用バッファ情報数は, 1 を設定してください。

(c) dcmcf_uocbufinf_n (バッファ情報)

seg_size

バッファの使用長を設定します。

(6) リターン値

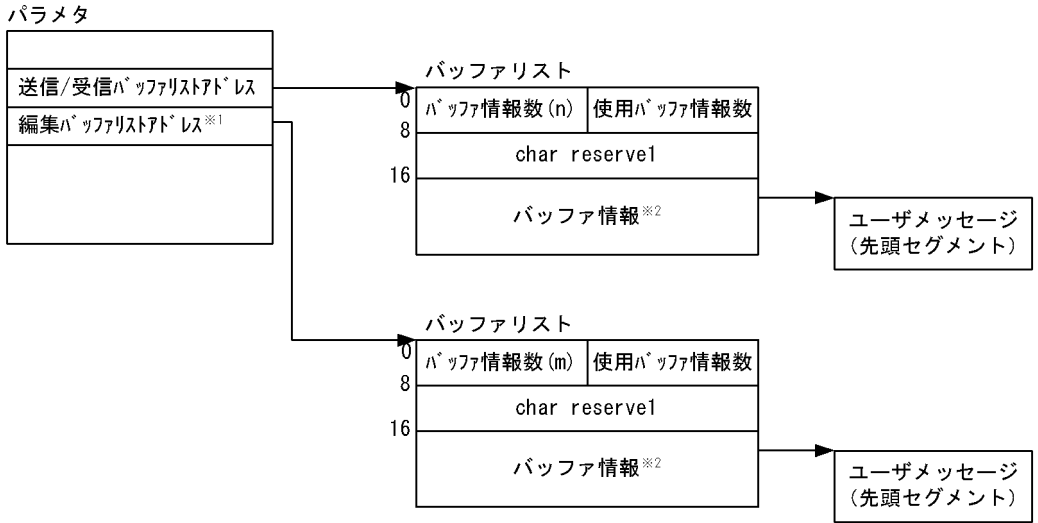
uoc_func () は次のコードでリターンしてください。

リターン値	意味
DCMCF_UOC_MSG_OK	正常リターン (編集バッファでスケジューリング)
DCMCF_UOC_MSG_OK_RCV	正常リターン (受信バッファでスケジューリング)
DCMCF_UOC_MSG_NG	メッセージ編集エラー

(7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係を図 4-2 に示します。

図 4-2 UOC インタフェース用のパラメタとバッファの関係



注 1

mcftalcen -e オプションを指定しない場合，NULL となり，バッファリストとバッファは確保されません。

注 2

バッファ情報は次の形式をしています。



4.1.3 出力メッセージの編集

出力メッセージ編集 UOC は，一方送信メッセージの編集をする UOC です。

出力メッセージの編集 UOC は，UAP が発行した送信メッセージを相手システムに実際に送信する前に処理するように位置させます。出力キューからセグメントを読み出すと起動します。

4. ユーザOWNコーディング，MCF イベントインタフェース

ユーザは，MCF メイン関数で UOC 関数アドレスを設定します。また，必要に応じて MCF 通信構成定義（`mcftalccn -e`）で，メッセージ編集用バッファグループ番号を定義します。

（１）出力メッセージの編集

送信するメッセージが格納されている送信バッファ，および MCF 通信構成定義で指定した編集バッファを引き渡します。UOC では，これらのバッファを使用して，出力メッセージを編集できます。例えば，MCF から渡されるメッセージ出力通番を，メッセージに組み込みます。

また，UOC からのリターンコードによって UAP に通知するメッセージとして送信バッファに格納されたものを使用するか，編集バッファに格納されたものを使用するかを選択できます。

各パラメタに設定した値が不正の場合，TP1/NET/X25-Extended は MCF イベントを通知してコネクションを解放します。

（２）UOC エラーリターン処理

UOC から DCMCF_UOC_MSG_NG でリターンした場合，MCF はメッセージログを出力し，障害通知イベント（CERREVT）を通知します。該当するメッセージは破棄されます。

（３）UOC パラメタ不正の場合の処理

UOC で設定した値に不正があった場合，MCF はメッセージログを出力し，障害通知イベント（CERREVT）を通知します。該当するメッセージは破棄されます。

（４）OpenTP1 への組み込み方法

入力メッセージ編集 UOC の組み込み方法と同じです。「4.1.1(5) OpenTP1 への組み込み方法」を参照してください。

4.1.4 出力メッセージ編集 UOC インタフェース

出力メッセージ編集 UOC は，次に示す形式で呼び出します。

（１）形式

ANSI C，C++ の場合

```
#include <dcmcfuoc.h>
DCLONG uoc_func(dcmcf_uoc_mout_n *parm)
```

K&R 版 C の場合

```
#include <dcmcfuoc.h>
DCLONG uoc_func(parm)

dcmcf_uoc_mout_n *parm ;
```

(2) 説明

uoc_func (出力メッセージ編集 UOC) を呼び出すとき, MCF は次の所定のパラメタを parm に設定します。

(3) パラメタの内容

(a) dcmcf_uoc_mout_n の内容

```
typedef struct {
    DCLONG pro_kind;           ... プロトコル種別
    char   le_name[9];        ... 論理端末名称
    char   reserve1[7];       ... 予備
    dcmcf_uocbuff_list_n *buflist_adr; ... 送信バッファリストアドレス
    dcmcf_uocbuff_list_n *ebuflist_adr; ... 編集バッファリストアドレス
    DCLONG output_no;        ... メッセージ出力通番
    char   msg_type;         ... メッセージ種別
    char   outputno_flag;    ... メッセージ出力通番有効フラグ
    char   resend_flag;     ... 再送フラグ
    char   reserve2[1];      ... 予備
    char   *pro_indv_ifa;    ... MCFが使用
    DCLONG rtn_detail;       ... 詳細リターンコード
    char   reserve3[20];     ... 予備
} dcmcf_uoc_mout_n;
```

(b) dcmcf_uocbuff_list_n, dcmcf_uocbufinf_n の内容

「4.1.2 入力メッセージ編集 UOC インタフェース」を参照してください。

(4) MCF が値を設定する項目

(a) dcmcf_uoc_mout_n

pro_kind

プロトコル種別として, 次の値が設定されます。

DCMCF_UOC_PRO_X25EX

X.25 プロトコル (TP1/NET/X25-Extended)

le_name

メッセージを入力した論理端末の名称が設定されます。

buflist_adr

送信用バッファリストのアドレスが設定されます。

ebuflist_adr

編集用バッファリストのアドレスが設定されます。

4. ユーザOWNコーディング，MCF イベントインタフェース

メッセージ編集バッファが未定義の場合，つまり MCF 通信構成定義の mcftalccn コマンドの -e オプションを省略した場合，eбуlist_adr には NULL が設定されます。

output_no

メッセージ出力通番が設定されます。ただし，outputno_flag が DCMCF_UOC_OUTPUTNO_OK のときだけ有効です。

msg_type

メッセージ種別として，次の値が設定されます。

'n'

一般の一方送信メッセージ

'p'

優先の一方送信メッセージ

outputno_flag

メッセージ出力通番有効フラグとして，次のどちらかの値が設定されます。

DCMCF_UOC_OUTPUTNO_OK

メッセージ出力通番を有効にします。

DCMCF_UOC_OUTPUTNO_NG

メッセージ出力通番を無効にします。

resend_flag

再送メッセージフラグとして，次のどちらかの値が設定されます。

'r'

再送メッセージです。

'n'

再送メッセージではありません。

pro_indv_ifa

MCF で使用するパラメタです。

(b) dcmcf_uocbuff_list_n, dcmcf_uocbufinf_n

「4.1.2 入力メッセージ編集 UOC インタフェース」を参照してください。

(5) ユーザが値を設定する項目

(a) dcmcf_uoc_mout_n

rtn_detail

詳細リターンコードを設定します。

このコードは，UOC が DCMCF_UOC_MSG_NG をリターンしたときに，MCF に渡されます。MCF は，詳細リターンコードをメッセージログファイルに出力します。

詳細リターンコードは，-19999 から -19000 の範囲で設定してください。

(b) dcmcf_uocbuff_list_n, dcmcf_uocbufinf_n

「4.1.2 入力メッセージ編集 UOC インタフェース」を参照してください。

(6) リターン値

uoc_func () は次のコードでリターンしてください。

リターン値	意味
DCMCF_UOC_MSG_OK	正常終了 (編集バッファでスケジューリング)
DCMCF_UOC_MSG_OK_SND	正常終了 (送信バッファでスケジューリング)
DCMCF_UOC_MSG_NG	メッセージ編集エラー

(7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係は, 入力メッセージ編集 UOC の場合と同じです。図 4-2 を参照してください。

4.1.5 送信メッセージの通番編集

(1) 送信メッセージの通番編集

送信メッセージの通番編集 UOC では, 受け取った通番をもとに, ユーザ独自の処理ができます。例えば, 通番を使用して送信メッセージを編集できます。

送信メッセージの通番編集 UOC を起動する場合, メッセージ送信の関数で, 出力通番を付けるように設定してください。UOC は, UAP が send 関数または resend 関数を発行したときに, MCF によって起動されます。

送信メッセージの通番編集 UOC は, 次に示す形式で, send_uoc 関数として作成します。

(2) OpenTP1 への組み込み方法

UAP のメイン関数の中に, 送信メッセージの通番編集 UOC の関数アドレスを登録しておきます。UAP のメイン関数に登録する dc_mcf_register 関数の形式を次に示します。

(a) 形式

ANSI C, C++ の場合

```
#include <dcmcf.h>
int dc_mcf_register(DCLONG flags,DCLONG (*uoc_addr)
(DCLONG flags,char *termname,
DCLONG sendno,DCLONG sendid,
DCLONG dataleng, char *senddata))
```

K&R 版 C の場合

4. ユーザOWNコーディング, MCF イベントインタフェース

```
#include <dcmcf.h>
int dc_mcf_register(flags,uoc_addr)
DCLONG flags;
DCLONG (*uoc_addr)();
```

(b) ユーザが値を設定する引数

flags

DCMCF_SEND_UOC を指定します。

uoc_addr

flags に対応する UOC のアドレスを指定します。

(c) リターン値

リターン値	意味
DC_OK	正常に終了しました。
DCMCFER_INVALID_ARGS	引数の指定が間違っています。
DCMCFER_NOMEM	ローカルメモリが不足しました。

(d) メイン関数への登録例

• MHP の場合

```
main()
{
extern DCLONG send_uoc();

dc_rpc_open();
dc_mcf_open();
dc_mcf_register(DCMCF_SEND_UOC,send_uoc);
dc_mcf_mainloop();
dc_mcf_close();
dc_rpc_close();
}
```

• SPP の場合

```
main()
{
extern DCLONG send_uoc();
dc_rpc_open();
dc_mcf_open();
dc_mcf_register(DCMCF_SEND_UOC,send_uoc);
dc_rpc_mainloop();
dc_mcf_close();
dc_rpc_close();
}
```

メイン関数, サービス関数, スタブ関数に UOC 関数をリンケージして UAP の実行形式プログラムを生成します。

4.1.6 送信メッセージの通番編集 UOC インタフェース

送信メッセージの通番編集 UOC は, 次に示す形式で, send_uoc 関数として作成します。

UOC の関数名はユーザの任意です。

(1) 形式

ANSI C, C++ の場合

```
#include <dcmcf.h>
DCLONG send_uoc(DCLONG flags, char *termname,
                DCLONG sendno, DCLONG sendid,
                DCLONG dataleng, char *senddata)
```

K&R 版 C の場合

```
#include <dcmcf.h>
DCLONG send_uoc(flags, termname, sendno, sendid, dataleng, senddata)
DCLONG          flags;
char            *termname;
DCLONG          sendno;
DCLONG          sendid;
DCLONG          dataleng;
char            *senddata;
```

(2) MCF から値が渡される項目

flags

送信メッセージの通番編集 UOC がいつ呼ばれたかが設定されます。

DCMCF_SEND_DML

メッセージを送信する関数または命令文が呼び出されたときを意味します。

DCMCF_RESEND_DML

メッセージを再送する関数または命令文が呼び出されたときを意味します。

termname

送信先の論理端末名称が設定されます。

sendno

送信メッセージの通番が設定されます。

sendid

送信するメッセージ種別が設定されます。

DCMCF_SEND_PRIO

優先の一方送信メッセージ

DCMCF_SEND_NORM

一般の一方送信メッセージ

4. ユーザOWNコーディング，MCF イベントインタフェース

dataleng

送信メッセージ（セグメント）の長さが設定されます。

senddata

送信メッセージ（セグメント）の内容が設定されます。

（3）リターン値

リターン値	意味
DC_OK	正常に終了しました。

4.1.7 UOC 作成上の注意事項

UOC 作成上の注意事項を次に示します。

（1）UOC の構造

UOC で使用するローカル変数のサイズの合計は，各 UOC で 1024 バイト以内になるように作成してください。また，UOC の中で関数の再帰呼び出しはしないでください。

（2）UOC で使用できる関数

UOC を作成する場合，UOC では次に示す関数だけが使用できます。ほかの関数を使用した場合，正常に動作しないことがあるためご注意ください。

メモリを操作する関数

- データ領域管理（例：malloc，free）
- 共有メモリ管理関数（例：shmctl，shmget，shmop）
- メモリ操作（例：memcpy）
- 文字列操作（例：strcpy）

時間取得関数

（3）UOC の異常処理

TP1/NET/X25-Extended の UOC で異常を検知した場合，MCF の所定のリターンコードを使用して，MCF に異常の発生を通知してください。UOC でプロセス終了となるシグナル，または abort（）を発行すると，MCF が異常終了します。

（4）UOC の実行タイミング

MCF が起動する UOC の実行タイミングは，OpenTP1 システム，および UAP の開始，終了シーケンスと同期しない場合があります。したがって，UAP より先に UOC が実行されたり，UAP がすべて終了してから UOC が呼ばれたりしてもよいように作成してください。

4.2 MCF イベントインタフェース

メッセージの送受信に関する各種のシステム情報が，MCF によってメッセージとして出力されます。このメッセージを MCF イベントといいます。MCF イベントは，メッセージの送受信処理でエラーや障害が発生した場合などに通知されます。

ユーザは MCF イベントを処理するための UAP (MHP) を作成できます。これを MCF イベント処理用 MHP といいます。MCF イベントは入力キューに書き込まれ，MCF イベント処理用 MHP に渡されます。ただし，MCF イベントでは入力メッセージ編集 UOC は呼び出されません。

MCF イベントの詳細については，マニュアル「OpenTP1 プログラム作成の手引」を参照してください。

4.2.1 MCF イベントの種類

TP1/NET/X25-Extended を使用した場合に通知される MCF イベントを，表 4-1 に示します。

表 4-1 MCF イベント一覧

MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP の処理の例
不正アプリケーション名検出通知イベント	ERREVT1	メッセージのアプリケーション名が MCF アプリケーション定義にありません。	該当するアプリケーション名がなかったことを通知します。
メッセージ廃棄通知イベント	ERREVT2	次の理由で受信メッセージを廃棄しました。 <ul style="list-style-type: none"> 入力キューに障害が発生しました。 MHP のサービス，サービスグループ，アプリケーションが閉塞しました。 MHP のセグメント受信回数に，セグメントを渡す前に MHP の異常終了が発生しました。 アプリケーション名に相当する MHP のサービスがありません。 アプリケーションの即時起動時に障害が発生しました。 MHP のアプリケーション，サービスグループがセキュア状態です。 	メッセージを廃棄したことを報告します。

4. ユーザOWNコーディング，MCF イベントインタフェース

MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP の処理の例
UAP 異常終了通知イベント	ERREVT3	MHP のセグメント受信関数に、セグメントを渡したあとに MHP の異常終了が発生しました。	UAP 異常終了時の対処障害メッセージを送信します。
タイマ起動メッセージ廃棄通知イベント	ERREVT4	アプリケーションのタイマ起動時に障害が発生しました。	メッセージを廃棄したことを報告します。
未処理送信メッセージ廃棄通知イベント	ERREVT4	次の理由で未処理送信メッセージを破棄しました。 <ul style="list-style-type: none"> • MCF の正常終了処理時に、未処理送信メッセージの滞留時間監視の時間切れ（タイムアウト）が発生しました。 • 運用コマンド（mcftdlqle）の入力によって、出力キューが削除されました。 	未処理送信メッセージを廃棄したことを報告します。
送信完了通知イベント	SCMPEVT	メッセージの送信を正常に完了しました。	送信の完了を確認して、任意の処理ができます。
障害通知イベント	CERREVT	通信管理プログラムのコネクション障害、または論理端末障害が発生しました。	コネクション、または論理端末に障害が発生したことを報告します。
状態通知イベント	COPNEVT	コネクションが確立しました。	メッセージの送受信ができることを報告します。
	CCLSEVT	コネクションが正常に解放されました。	メッセージの送受信ができないことを報告します。

注

MCF アプリケーション定義（mcfaalcap -g）の recvmsg オペランドに r を指定した場合、または dc_mcf_rollback の action に DCMCFRTRY または DCMCFRRTN を指定した場合は除きます。

4.2.2 MCF イベント通知時のセグメント構成

MCF イベントを MHP に通知される場合、先頭セグメントに MCF イベント情報を設定します。エラーイベント（ERREVTn）の場合は、第 2 セグメントに処理できなかったメッセージセグメントを設定します。

MCF イベント通知時のセグメント構成を次の図に示します。

図 4-3 MCF イベント通知時のセグメント構成



注 エラーイベント (ERREVTn) 通知時だけ渡されます。

MCF イベントは，UAP を作成した言語によって，UAP に通知されるデータの形式が異なります。

エラーイベントの場合はバッファ形式 1 とバッファ形式 2 で先頭の内容が異なります。

このため，それ以降の項目の位置にずれがあります。「4.2.4 MCF イベント情報の形式 (COBOL 言語)」のエラーイベントの表では ERREVT1, ERREVT2, ERREVT3, および ERREVT4 のバッファ形式別に位置 (バイト) を分けて説明しています。なお，ERREVT4 についてはマニュアル「OpenTP1 プログラム作成リファレンス」の該当する言語編を参照してください。

4.2.3 MCF イベント情報の形式 (C 言語)

MCF イベント情報は構造体で，MCF イベント処理用 MHP に渡されます。MHP に渡される構造体の形式は，MCF イベントの種類によって異なります。ただし，MCF イベント情報の先頭部分の形式は，各イベントで共通です。

エラーイベント (ERREVTn) および送信完了通知イベント (SCMPEVT) で使用する構造体は，<dc_mcf.h> で定義してあります。

COPNEVT, CCLSEVT および CERREVT で使用する構造体は，<dcmxem.h> で定義されています。また，dc_mcf_evtheader は <dc_mcf.h> で定義されているので，<dcmxem.h> の前に <dc_mcf.h> を取り込んでおいてください。

MCF イベント情報の先頭部分の形式を次に示します。

(1) MCF イベントの共通ヘッダ

(a) 形式

```
struct dc_mcf_evtheader {
  char   mcfevt_name[9] ;      ... MCFイベント情報コード
  char   le_name[16] ;        ... 入力元論理端末名称
                                   (ERREVT1, ERREVT2, ERREVT3の場合)
                                   出力先論理端末名称
                                   (ERREVT4の場合)
  char   cn_name[9] ;          ... コネクション名またはコネクショングループ名
  unsigned char format_kind ; ... MCF使用領域
  char   reserve01 ;           ... 予備
  DCLONG time ;                ... メッセージ入力時刻
}
```

4. ユーザOWNコーディング, MCF イベントインタフェース

```
} ;
```

(b) MCF イベントとして設定される項目

le_name

ERREVT1, ERREVT2, または ERREVT3 では, メッセージを入力した論理端末名称が設定されます。

ただし, ERREVT2 または ERREVT3 で, 次に示す場合には, '*' が設定されます。

- SPP がアプリケーション起動をした MHP で, 障害が発生した場合
- 上記の障害発生時に MCF イベントとして起動された MHP によって, さらにアプリケーション起動をされた MHP で, 障害が発生した場合

ERREVT4, SCMPEVT では, メッセージを出力する論理端末名称が設定されます。COPNEVT または CCLSEVT の場合は無効です。

cn_name

ERREVT1, ERREVT2, ERREVT3, ERREVT4 および SCMPEVT の場合, コネクショングループ名が設定されます。

ただし, ERREVT2 または ERREVT3 で, 次に示す場合には, '*' が設定されます。

- SPP がアプリケーション起動をした MHP で, 障害が発生した場合
- 上記の障害発生時に MCF イベントとして起動された MHP によって, さらにアプリケーション起動をされた MHP で, 障害が発生した場合

COPNEVT, CCLSEVT, CERREVT の場合, コネクション名が設定されます。

time

メッセージを入力した時刻が, 1970 年 1 月 1 日 0 時 0 分 0 秒からの通算の秒数で設定されます。

(2) ERREVT1

(a) 形式

```
struct dc_mcf_evt1_type {
    struct dc_mcf_evtheader  evtheader ; ... MCFイベント情報共通ヘッダ
    char reserve01[12] ; ... 予備
    char reserve02[10] ; ... 予備
    char reserve03[2] ; ... 予備
    char ap_name[10] ; ... アプリケーション名
                                (メッセージに対応する
                                アプリケーション名)
    char reserve04[2] ; ... 予備
} ;
```

(b) MCF イベントとして設定される項目

ap_name

次に示すどちらかが設定されます。

- 形式不正の場合...不正となったアプリケーション名
- 定義されていない場合...定義されていないアプリケーション名

アプリケーション名は, MHP から送信されたメッセージの場合に設定されます。ただし, MHP 以外から送信された場合はヌル文字が設定されます。

(3) ERREVT2

(a) 形式

```
struct dc_mcf_evt2_type {
  struct dc_mcf_evtheader  evtheader ; ... MCFイベント情報共通ヘッダ
  char reserve01[12] ; ... 予備
  char reserve02[10] ; ... 予備
  char reserve03[2] ; ... 予備
  char ap_name[10] ; ... アプリケーション名
                                (メッセージに対応する
                                アプリケーション名)
  short reason_code ; ... 理由コード
};
```

(b) MCF イベントとして設定される項目

ap_name

エラーになった UAP のアプリケーション名が設定されます。

アプリケーション名は, MHP から送信されたメッセージの場合に設定されます。

MHP 以外から送信された場合は, ヌル文字が設定されます。

reason_code

ERREVT2 の理由コードが設定されます。理由コードの詳細については, 「付録 C 理由コード一覧」を参照してください。

(4) ERREVT3

(a) 形式

```
struct dc_mcf_evt3_type {
  struct dc_mcf_evtheader  evtheader ; ... MCFイベント情報共通ヘッダ
  char reserve01[12] ; ... 予備
  char map_name[10] ; ... MCF使用領域
  char reserve03[2] ; ... 予備
  char ap_name[10] ; ... アプリケーション名
                                (異常が発生したメッセージの
                                アプリケーション名)
  char reserve04[2] ; ... 予備
  char service_name[32] ; ... サービス名
  char serv_grp_name[32] ; ... サービスグループ名
  char bid[36] ; ... トランザクションプランチID領域
};
```

4. ユーザOWNコーディング, MCF イベントインタフェース

(b) MCF イベントとして設定される項目

ap_name

異常が発生した MHP のアプリケーション名が設定されます。

アプリケーション名は, MHP から送信されたメッセージの場合に設定されます。

ただし, MHP 以外から送信された場合はヌル文字が設定されます。

service_name

異常が発生した MHP のアプリケーション名に対応するサービス名が設定されます。

serv_grp_name

異常が発生した MHP のサービスが属するサービスグループ名が設定されます。

bid

トランザクションブランチ ID が設定されます。

(5) ERREVT

(a) 形式

```
struct dc_mcf_evta_type {
    struct dc_mcf_evtheader  evtheader ; ... MCFイベント情報共通ヘッダ
    char reserve01[12] ; ... 予備
    char map_name[10] ; ... MCF使用領域
    char reserve03[2] ; ... 予備
    char ap_name[10] ; ... アプリケーション名
                                (正常終了したメッセージの
                                アプリケーション名)

    char reserve04[2] ; ... 予備
    char reserve05[32] ; ... 予備
    char reserve06[32] ; ... 予備
    DCLONG user_leng ; ... 他プロトコルの場合の使用領域
    char user_data[16] ; ... 他プロトコルの場合の使用領域
    char reserve07[16] ; ... 予備
} ;
```

(b) MCF イベントとして設定される項目

ap_name

正常終了したメッセージのアプリケーション名が設定されます。

アプリケーション名は, MHP から送信されたメッセージの場合に設定されます。た

だし, MHP 以外から送信された場合はヌル文字が設定されます。

(6) SCMPEVT

(a) 形式

```
struct dc_mcf_scmpevt_type {
    struct dc_mcf_evtheader  evtheader ; ... MCFイベント情報共通ヘッダ
    DCLONG output_no ; ... 出力通番
    char map_name[9] ; ... MCF使用領域
}
```

```

char    msg_type ;                ... メッセージ種別
char    reason_code ;            ... 起動理由
char    reserve01[5] ;           ... 予備
DCLONG  user_leng ;              ... 他プロトコルの場合の使用領域
char    pro_indv_inf[16] ;       ... 他プロトコルの場合の使用領域
char    user_data[16] ;          ... 他プロトコルの場合の使用領域
} ;

```

(b) MCF イベントとして設定される項目

output_no

出力通番が設定されます。通番がない場合は、「-1」が設定されます。

msg_type

メッセージ種別として次の値が設定されます。

'n'

一般の一方送信メッセージ

'p'

優先の一方送信メッセージ

reason_code

起動理由として次の値が設定されます。

' , '

メッセージが正常に送信されました

(7) CERREVT

(a) 形式

```

typedef struct {
    struct dc_mcf_evtheader  header ;    ... MCFイベント情報共通ヘッダ
    DCLONG  err_fact ;                ... 障害要因コード
                                        0x30 : コネクション障害発生
    DCLONG  err_reason1 ;            ... 理由コード1
    DCLONG  err_reason2 ;            ... 理由コード2
    DCLONG  reserve ;                ... 予備
    char    group_name[16] ;          ... コネクショングループ名
    char    reserve1[26] ;            ... 予備
} dcmxem_cerrevt ;

```

(b) MCF イベントとして設定される項目

err_reason1 , err_reason2

CERREVT の理由コード 1 , 理由コード 2 が設定されます。「付録 C 理由コード一覧」を参照してください。

group_name

コネクショングループ名が設定されます。

4. ユーザOWNコーディング，MCF イベントインタフェース

(8) COPNEVT，CCLSEVT

(a) 形式

```
typedef struct {
    struct dc_mcf_evtheader header ;    ... MCFイベント情報共通ヘッダ
    char reserve[16] ;                ... 予備
    char group_name[16] ;              ... コネクショングループ名
    char reserve1[26] ;                ... 予備
} dcmxem_statevt ;
```

(b) MCF イベントとして設定される項目

group_name

コネクショングループ名が設定されます。

4.2.4 MCF イベント情報の形式 (COBOL 言語)

COBOL 言語の場合はセグメントの並びとして渡されます。COBOL 言語の UAP の場合の MCF イベントの内容を表 4-2 ~ 表 4-8 に示します。

表 4-2 COBOL 言語の MCF イベント情報の内容 (ERREVT1)

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベント コード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT1 を示す '1' が設定され ます。
入力元論理端末名 称	12	8	8	英数字	メッセージを入力した論理端末名称で す。
予備	20	16	20	-	-
アプリケーション 名	40	36	8	英数字	次に示すどれかが設定されます。 • 形式不正となったアプリケーション 名 • 定義されていないアプリケーション 名
予備	48	44	8	-	-

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備	56	52	8	-	-
予備	64	60	8	-	-
コネクショングループ名	72	68	8	英数字	コネクショングループ名です。
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進	端末入力メッセージを入力した日付です。「yyyymmdd」の形式です。 yyyy: 西暦の年 mm: 月 dd: 日
メッセージが入力された時刻	104	100	8	外部 10 進	端末入力メッセージを入力した時刻です。「hhmmss00」の形式です。 hh: 時 mm: 分 ss: 秒 00 は固定です。
予備	112	108	16	-	-

(凡例)

- : 該当しません。または, 使用されません。

表 4-3 COBOL 言語の MCF イベント情報の内容 (ERREVT2)

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベント コード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT2 を示す '2' が設定されます。

4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
入力元論理端末名称	12	8	8	英数字	メッセージを入力した論理端末名称です。次に示す場合には，'*' が設定されます。 <ul style="list-style-type: none"> • SPP がアプリケーション起動をした MHP で障害が発生した場合 • 上記の障害発生時に MCF イベントとして起動された MHP によって，更にアプリケーション起動された MHP で，障害が発生した場合
予備	20	16	20	-	-
アプリケーション名	40	36	8	英数字	エラーになった UAP のアプリケーション名が設定されます。
予備	48	44	8	-	-
予備	56	52	8	-	-
予備	64	60	8	-	-
コネクショングループ名	72	68	8	英数字	コネクショングループ名です。次に示す場合には，'*' が設定されます。 <ul style="list-style-type: none"> • SPP がアプリケーション起動をした MHP で障害が発生した場合 • 上記の障害発生時に MCF イベントとして起動された MHP によって，更にアプリケーション起動された MHP で，障害が発生した場合
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進	端末入力メッセージを入力した日付です。「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	104	100	8	外部 10 進	端末入力メッセージを入力した時刻です。「hhmmss00」の形式です。 hh：時 mm：分 ss：秒 00 は固定です。
理由コード	112	108	4	外部 10 進	理由コードが設定されます。
予備	116	112	12	-	-

(凡例)

- : 該当しません。または，使用されません。

注

理由コードの内容については、「付録 C 理由コード一覧」を参照してください。

表 4-4 COBOL 言語の MCF イベント情報の内容 (ERREVT3)

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベント コード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT3 を示す '3' が設定されます。
入力元論理端末名称	12	8	8	英数字	メッセージを入力した論理端末名称です。次に示す場合は, '*' が設定されます。 <ul style="list-style-type: none"> • SPP がアプリケーション起動をした MHP で, 障害が発生した場合 • 上記の障害発生時に MCF イベントとして起動された MHP が, 更にアプリケーション起動をした MHP で, 障害が発生した場合
予備	20	16	20	-	-
予備	40	36	8	-	-
マップ名	48	44	8	-	MCF が使用します。
アプリケーション名	56	52	8	英数字	異常が発生したメッセージのアプリケーション名です。
予備	64	60	8	-	-
コネクショングループ名	72	68	8	英数字	コネクショングループ名です。次に示す場合は, '*' が設定されます。 <ul style="list-style-type: none"> • SPP がアプリケーション起動をした MHP で, 障害が発生した場合 • 上記の障害発生時に MCF イベントとして起動された MHP が, 更にアプリケーション起動をした MHP で, 障害が発生した場合
予備	80	76	16	-	-

4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
メッセージが入力された日付	96	92	8	外部 10 進	端末入力メッセージを入力した日付です。「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	104	100	8	外部 10 進	端末入力メッセージを入力した時刻です。「hhmmss00」の形式です。 hh：時 mm：分 ss：秒 00 は固定です。
予備	112	108	16	-	-
サービス名	128	124	31	英数字	異常が発生した UAP のアプリケーション名に対応するサービス名です。
予備	159	155	1	-	-
サービスグループ名	160	156	31	英数字	異常が発生したサービスグループ名です。
予備	191	187	1	-	-
トランザクション ID(BID)	192	188	36	2 進	異常が発生したトランザクションの BID です。
予備	228	224	28	-	-

(凡例)

- : 該当しません。または，使用されません。

表 4-5 COBOL 言語の MCF イベント情報の内容 (ERREVTA)

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベント コード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-

4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
	10	6	2	英数字	ERREVTVA を示す 'A' が設定されます。
出力先論理端末名称	12	8	8	英数字	メッセージを出力する論理端末名称です。
予備	20	16	20	-	-
予備	40	36	8	-	-
マップ名	48	44	8	-	MCF が使用します。
アプリケーション名	56	52	8	英数字	正常終了したメッセージのアプリケーション名 (MHP から送信されたメッセージの場合設定されます。MHP 以外から送信された場合は空白が設定されます。)
予備	64	60	8	-	-
コネクショングループ名	72	68	8	英数字	コネクショングループ名です。
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進	端末入力メッセージを入力した日付です。「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	104	100	8	外部 10 進	端末入力メッセージを入力した時刻です。「hhmmss00」の形式です。 hh：時 mm：分 ss：秒 00 は固定です。
予備	112	108	16	-	-
予備	128	124	31	-	-
予備	159	155	1	-	-
予備	160	156	31	-	-
予備	191	187	1	-	-
予備	192	188	36	-	-
予備	228	224	28	-	-

(凡例)

- : 該当しません。または、使用されません。

4. ユーザOWNコーディング，MCF イベントインタフェース

表 4-6 COBOL 言語の MCF イベント情報の内容 (SCMPEVT)

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
長さ (形式 1 のときだけ)	0	-	2	-	MCF が使用します。
予備 (形式 1 のときだけ)	2	-	2	-	-
イベントコード	4	0	8	英数字	「SCMPEVT」が設定されます。
出力先論理端末名称	12	8	8	英数字	送信が完了した論理端末の名称です。
予備	20	16	8	-	-
コネクショングループ名	28	24	8	英数字	コネクショングループ名です。
メッセージが入力された日付	36	32	8	外部 10 進	SCMPEVT が通知された日付です。 「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	44	40	8	外部 10 進	SCMPEVT が通知された時刻です。 「hhmmss00」の形式です。 hh：時 mm：分 ss：秒 00 は固定です。
出力通番	52	48	4	2 進数	出力通番です。出力通番がない場合は、(FFFFFFF) ₁₆ が設定されます。
出力マップ名	56	52	8	英数字	MCF が使用します。
メッセージ種別	64	60	1	英数字	メッセージ種別が設定されます。 n：一般一方送信メッセージ p：優先一方送信メッセージ
起動理由	65	61	1	英数字	起動要因が設定されます。 ：送信が正常に完了しました。
予備	66	62	9	-	-
プロトコル個別情報	75	71	16	英数字	MCF が使用します。
予備	91	87	19	-	-

(凡例)

-：該当しません。または、使用されません。

表 4-7 COBOL 言語の MCF イベント情報の内容 (CERREVT)

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	「CERREVT」が設定されます。
入力元論理端末名称	8	8	英数字	障害の発生した論理端末名称が設定されます。
予備	16	8	-	-
入力元コネクション名	24	8	英数字	コネクション名が設定されます。
メッセージが入力された日付	32	8	外部 10 進	CERREVT が通知された日付です。 「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	40	8	外部 10 進	CERREVT が通知された時刻です。 「hhmmss00」の形式です。 hh：時 mm：分 ss：秒 00 は固定です。
障害要因コード	48	4	2 進数	障害要因が設定されます。 (00000030) ₁₆ ：コネクション障害
理由コード 1	52	4	2 進数	理由コード 1 が設定されます。
理由コード 2	56	4	2 進数	理由コード 2 が設定されます。
予備	60	4	-	-
コネクショングループ名	64	16	英数字	コネクショングループ名が設定されます。
予備	80	28	-	-

(凡例)

-：該当しません。または、使用されません。

注

理由コード 1 および理由コード 2 については、「付録 C 理由コード一覧」を参照してください。

4. ユーザOWNコーディング，MCF イベントインタフェース

表 4-8 COBOL 言語の MCF イベント情報の内容 (COPNEVT，CCLSEVT)

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	「COPNEVT」，または「CCLSEVT」 が設定されます。
入力元論理端末名称	8	8	英数字	論理端末名称が設定されます。
予備	16	8	-	-
入力元コネクション名	24	8	英数字	コネクション名が設定されます。
メッセージが入力され た日付	32	8	外部 10 進	COPNEVT，CCLSEVT が通知され た日付です。「yyyymmdd」の形式で す。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力され た時刻	40	8	外部 10 進	COPNEVT，CCLSEVT が通知され た時刻です。「hhmmss00」の形式で す。 hh：時 mm：分 ss：秒 00 は固定です。
予備	48	16	-	-
コネクション グループ名	64	16	英数字	コネクショングループ名が設定され ます。
予備	80	28	-	-

(凡例)

-：該当しません。または，使用されません。

5

システム定義

この章では，TP1/NET/X25-Extended を使用するために必要な，OpenTP1 のシステム定義の中での TP1/NET/X25-Extended 固有のシステム定義について説明します。また，自システム内にある通信管理プログラムと関連づける内容，およびシステム定義例について説明します。

TP1/NET/X25-Extended の定義の概要

TP1/NET/X25-Extended 固有のシステム定義の種類

mcftalccn (コネクション定義)

mcftalcle (論理端末定義)

mcftgrpcn (コネクショングループ定義の開始)

mcftgrped (コネクショングループ定義の終了)

システムサービス情報定義

システムサービス共通情報定義

MCF 定義オブジェクトの生成

自システムの通信管理プログラムと関係づける内容

定義例

TP1/NET/X25-Extended の定義の概要

TP1/NET/X25-Extended のシステム定義は、OpenTP1 のネットワークコミュニケーション定義の中で定義します。また、自システムの通信管理プログラムの定義と、システム定義内容を関連づける必要があります。

OpenTP1 のネットワークコミュニケーション定義の中での定義

OpenTP1 のネットワークコミュニケーション定義のうち、TP1/NET/X25-Extended に固有の定義について説明します。

使用する定義ファイル

MCF および TP1/NET/X25-Extended を起動するには、定義ファイルに環境情報を設定する必要があります。MCF で使用する定義ファイルを次の表に示します。

表 5-1 MCF で使用する定義ファイル

定義の種類	定義のソースファイル	定義の内容
MCF マネージャ定義	MCF マネージャ定義ソースファイル	MCF 全体の実行環境
MCF 通信構成定義	共通定義ソースファイル	プロトコルごとの
	プロトコル固有定義ソースファイル	実行環境
MCF アプリケーション定義	MCF アプリケーション定義ソースファイル	アプリケーションの属性

定義のソースファイルは、定義コマンド、オプション、オペランドを使用して作成します。それらの中には、プロトコルで共通のものと、プロトコルに固有のものがあります。表 5-1 の定義の中で、TP1/NET/X25-Extended に固有の定義があるものを次に示します。

- MCF 通信構成定義

この章では、TP1/NET/X25-Extended に固有の定義コマンド、オプション、およびオペランドについて説明します。プロトコルで共通の定義については、マニュアル「OpenTP1 システム定義」を参照してください。ただし、mcftbuf (バッファグループ定義) の length, count オペランドの指定値については、5 章の「mcftalccn (コネクション定義)」の注意事項に記載してあります。

TP1/NET/X25-Extended の組み込み時に必要なファイル

次に示すファイルは、TP1/NET/X25-Extended を OpenTP1 システムに組み込むときに必要なファイルです。

- システムサービス情報定義ファイル
- システムサービス共通情報定義ファイル

- MCF 定義オブジェクトファイル

この章では、システムサービス情報定義ファイルとシステムサービス共通情報定義ファイルの記述内容、および MCF 定義オブジェクトファイルを生成するユーティリティの起動コマンドについて説明します。TP1/NET/X25-Extended を組み込む方法については、「7. 組み込み方法」を参照してください。

通信定義の内容の関連づけ

X25 (VC) 手順で相手システムと通信するためには、TP1/NET/X25-Extended のシステム定義内容を自システムの通信管理プログラムと関連づける必要があります。

この章では、自システムの通信管理プログラム (XNF/AS) と関連づける内容を示します。

TP1/NET/X25-Extended 固有のシステム定義の種類

OpenTP1 のネットワークコミュニケーション定義のうち，TP1/NET/X25-Extended に固有の定義の一覧を次の表に示します。

表 5-2 TP1/NET/X25-Extended に固有の定義の一覧

定義名	コマンド	オプション・オペランド	定義内容	指定値 ((値範囲)) 《省略時解釈値》
M C F 通 信 構 成 定 義	共通定義		プロトコル共通のコマンドだけで指定できます。共通のコマンドについては，マニュアル「OpenTP1 システム定義」を参照してください。	
	プロトコル固有定義	mcftgrpcn (コネクショングループ定義の開始) 指定数：1 ~ 512	-g	-
	mcfalcle (論理端末定義) 指定数：mcftgrpcn と同数	-l	-	論理端末名称 1 ~ 8 文字の識別子
		-t	-	論理端末タイプ any
		-m	mmsgcnt	メモリ出力メッセージ最大格納数 符号なし整数 ((0 ~ 65535)) 《0》
			dmsgcnt	ディスク出力メッセージ最大格納数 符号なし整数 ((0 ~ 65535)) 《0》
		-k	quekind	出力キューの媒体の種類 《memory》 disk
			quegrpID	キューグループ ID 1 ~ 8 文字の識別子
		-o	aj	送信完了時の情報を取得するかどうか 《yes》 no
	-v	-	メッセージ受信時に起動するアプリケーション名 1 ~ 8 文字の識別子	

定義名	コマンド	オプション・オペランド		定義内容	指定値 ((値範囲)) 《省略時解釈値》
mcftalccn (コネクション定義) 指定数: 1 ~ 512		-c	-	コネクション ID	1 ~ 8 文字の識別子
		-p	-	プロトコルの種別	x25ex
		-g	sndbuf	送信用バッファグループ番号	符号なし整数 ((1 ~ 512))
			rcvbuf	受信用バッファグループ番号	符号なし整数 ((1 ~ 512))
		-e	msgbuf	編集用バッファグループ番号	符号なし整数 ((1 ~ 512))
		-i	-	システム開始時および再開時にコネクションを自動的に確立するかどうか	auto 《manual》
		-z	vslot	仮想スロット番号	符号なし整数 ((1 ~ 1000))
		-b	bretry	コネクション確立時に障害が発生した場合にコネクション確立再試行をするかどうか	《yes》 no
			bretrycnt	コネクションの確立再試行回数	符号なし整数 (単位: 回) ((0 ~ 65535)) 《0》
			bretryint	コネクションの確立再試行間隔	符号なし整数 (単位: 秒) ((0 ~ 2550)) 《60》
		-t	-	発呼か着呼か	int 《rsp》
		-q	snpa	相手 SNPA アドレス	1 ~ 30 けたの数字
		-f	kind	MCF イベントの種類	cerr 《ccls》
mcftgrped (コネクショングループ定義の終了) 指定数: mcftgrpen と同数		-	-	コネクショングループ定義の終了	-

(凡例)

- : 該当する内容がないことを表します。

定義の指定順序

TP1/NET/X25-Extended のプロトコル固有定義コマンドの指定順序を次の図に示しま

5. システム定義

TP1/NET/X25-Extended 固有のシステム定義の種類

す。

図 5-1 TP1/NET/X25-Extended のプロトコル固有定義コマンドの指定順序

⎧	mcftgrpcn	(コネクショングループ定義の開始)	⎥	一組として 繰り返し可
	mcftalcle	(論理端末定義)		
	mcftalccn*	(コネクション定義)		
	⋮			
	mcftgrped	(コネクショングループ定義の終了)		

注 コネクション定義は、一つの繰り返し単位の中に複数指定できます。

mcftalccn (コネクション定義)

形式

```
mcftalccn -c コネクションID
          -p x25ex
          -g "sndbuf = メッセージ送信用バッファグループ番号
             rcvbuf = メッセージ受信用バッファグループ番号"
          [-e " { msgbuf = メッセージ編集用バッファグループ番号 } " ]
          [-i auto | manual ]
          [-z " { vslot = 仮想スロット番号 } " ]
          [-b " { bretry = yes | no }
             { bretrycnt = コネクション確立障害時の確立再試行回数 }
             { bretryint = コネクション確立障害時の確立再試行間隔 } " ]
          [-t int | rsp ]
          -q "snpa = 相手SNPAアドレス"
          [-f " { kind = cerr | ccls } " ]
```

機能

コネクションに関する環境を定義します。

オプション

-c コネクション ID ~ 1 ~ 8 文字の識別子

コネクション ID を指定します。

このコネクション ID は、ほかの mcftalccn コマンドの -c オプションで指定するコネクション ID と重複して指定できません。

-p x25ex

プロトコルの種別を指定します。

x25ex

X.25 プロトコル (TP1/NET/X25-Extended)

-g

(オペランド)

sndbuf = メッセージ送信用バッファグループ番号 ~ 符号なし整数 ((1 ~ 512))

メッセージ送信用のバッファグループ番号を指定します。

mcftbuf コマンドの -g オプションの groupno オペランドで指定したバッファグループ番号を指定します。

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

rcvbuf = メッセージ受信用バッファグループ番号 ~ 符号なし整数 ((1 ~ 512))

5. システム定義

mcftalccn (コネクション定義)

メッセージ受信用のバッファグループ番号を指定します。

mcftbuf コマンドの -g オプションの groupno オペランドで指定したバッファグループ番号を指定します。

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

-e

(オペランド)

msgbuf = メッセージ編集用バッファグループ番号 ~ 符号なし整数 ((1 ~ 512))

入力および出力メッセージ編集 UOC で、メッセージ編集用として使用するバッファグループの番号を指定します。

このオペランドを省略した場合、メッセージ編集用バッファは確保されません。

メッセージ編集用バッファグループ番号は、mcftbuf コマンドの -g オプションの groupno オペランドで指定したバッファグループ番号を指定します。

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

-i auto | manual ~ 《manual》

MCF 開始時および再開時に、自動的にコネクションを確立するかどうかを指定します。

auto

コネクションを自動的に確立します。

manual

MCF 起動後、ユーザが運用コマンド (mcftactcn) を入力して、コネクションを確立します。

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

-z

(オペランド)

vslot = 仮想スロット番号 ~ 符号なし整数 ((1 ~ 1000))

仮想スロット番号を指定します。-t オプションで int を指定した場合、このオペランドは省略できません。-t オプションで rsp を指定した場合は、このオペランドの指定は無効となります。

構成定義 link 文の VASS オペランドで指定した仮想スロット番号を指定します。

-b

(オペランド)

bretry = yes | no ~ 《yes》

コネクション確立時に障害が発生した場合、コネクションの確立を再試行するかどうかを指定します。

yes

コネクションの確立を再試行します。

no

コネクションの確立を再試行しません。

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

bretrycnt = コネクション確立障害時の確立再試行回数 ~ 符号なし整数 ((0 ~ 65535)) 《0》 (単位: 回)

コネクション確立時に障害が発生した場合の、コネクションの確立を再試行する回数を指定します。

このオペランドを省略した場合、または0を指定した場合、無限に再試行されます。

bretry オペランドで no を指定した場合、このオペランドの指定は無効です。

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

bretryint = コネクション確立障害時の確立再試行間隔 ~ 符号なし整数 ((0 ~ 2550)) 《60》 (単位: 秒)

コネクション確立時に障害が発生した場合の、コネクションの確立を再試行する時間間隔を指定します。

0を指定した場合、障害後直ちに再試行されます。

bretry オペランドで no を指定した場合、このオペランドの指定は無効です。

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

-t int | rsp ~ 《rsp》

コネクションの種類を指定します。

int

発呼コネクションとして使用します。

rsp

着呼コネクションとして使用します。

-q

(オペランド)

snpa = 相手 SNPA アドレス ~ 1 ~ 30 けたの数字

相手 SNPA アドレスを指定します。

5. システム定義

mcftalccn (コネクション定義)

-f

(オペランド)

kind = cerr | ccls ~ 《ccls》

コネクションの切断時(自システムからのコネクション解放は除く)に通知させる MCF イベントの種別を指定します。

cerr

CERREVT を通知させます。

ccls

CCLSEVT を通知させます。

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

注意事項

同じコネクショングループに属するコネクションでは、次のオプションおよびオペランドの指定値がすべて一致する必要があります。

- -g オプションの sndbuf, および revbuf オペランド
- -e オプションの msgbuf オペランド
- -i オプション
- -b オプションの bretry, bretrycnt, および bretryint オペランド
- -f オプションの kind オペランド

-g オプションおよび -e オプションで指定するバッファグループ番号は、バッファグループ定義の mcftbuf コマンドに対応しています。mcftbuf コマンドでは、1 コネクショングループ単位に次の表に示す資源が必要です。バッファグループ定義については、マニュアル「OpenTP1 システム定義」を参照してください。

バッファ種別	mcftbuf コマンド	
	length オペランド	count オペランド
sndbuf	送信最大メッセージ長	コネクション数以上
revbuf	受信最大メッセージ長	同時受信メッセージ数
msgbuf	メッセージ編集 UOC で編集後の最大メッセージ長	コネクション数 × 2

注 UOC でメッセージを編集しない場合は、指定の必要はありません。

mcftalcle (論理端末定義)

形式

```
mcftalcle -l 論理端末名称
-t any
[-m " { mmsgcnt = メモリ出力メッセージ最大格納数
        { dmsgcnt = ディスク出力メッセージ最大格納数 } " ]
[-k " { quekind = memory | disk }
        { quegrpid = キューグループID } " ]
[-o " { aj = yes | no } " ]
[-v アプリケーション名]
```

機能

論理端末に関する環境を定義します。

オプション

-l 論理端末名称 ~ 1 ~ 8 文字の識別子

論理端末名称を指定します。

この論理端末名称は、ほかの mcftalcle コマンドの -l オプションで指定する論理端末名称と重複して指定できません。

-t any

論理端末の端末タイプを指定します。

any

任意型論理端末

-m

(オペランド)

mmsgcnt = メモリ出力メッセージ最大格納数 ~ 符号なし整数 ((0 ~ 65535)) 《0》
メモリキューで待ち合わせをする出力メッセージの最大格納数を指定します。
出力メッセージの待ち合わせ数が指定した最大数になると、それ以降の UAP からの送信要求 (de_mcf_send 関数または SEND 文) はエラーとなります。
0 を指定した場合、メモリキューで待ち合わせをする出力メッセージの数は無制限になります。ただし、実際に待ち合わせをできる出力メッセージ数は動的共用メモリの容量に依存します。

dmsgcnt = ディスク出力メッセージ最大格納数 ~ 符号なし整数 ((0 ~ 65535))
《0》

ディスクキューで待ち合わせをする出力メッセージの最大格納数を指定します。

5. システム定義
mcftalcle (論理端末定義)

出力メッセージの待ち合わせ数が指定した最大数になると、それ以降の UAP からの送信要求 (dc_mcf_send 関数または SEND 文) はエラーとなります。

0 を指定した場合、ディスクキューで待ち合わせをする出力メッセージの数は無制限になります。ただし、実際に待ち合わせをできる出力メッセージ数はメッセージキューファイルの容量に依存します。

-k

(オペランド)

quekind = memory | disk ~ 《memory》

出力メッセージの割り当て先 (メモリキューまたはディスクキュー) を指定します。

memory

メモリキューだけに割り当てます。

disk

ディスクキュー、およびメモリキューに割り当てます。disk を指定した場合、必ず quegrpид オペランドを指定してください。

quegrpид = キューグループ ID ~ 1 ~ 8 文字の識別子

ディスクキューで待ち合わせをする出力メッセージに使用するキューグループ ID を指定します。MCF マネージャ定義の mcfmqgid コマンドで指定したキューグループ ID (キューグループ種別は otq) のどれかを指定します。

この quegrpид オペランドは、quekind オペランドで disk を指定した場合だけ指定します。

-o

(オペランド)

aj = yes | no ~ 《yes》

メッセージ送信が完了した場合に、メッセージ送信完了ジャーナル (AJ) を取得するかどうかを指定します。

yes

メッセージ送信完了ジャーナルを取得します。

no

メッセージ送信完了ジャーナルを取得しません。

-v アプリケーション名 ~ 1 ~ 8 文字の識別子

この論理端末で受信したメッセージを処理するために起動するアプリケーションの名称を指定します。MCF アプリケーション定義 (mcfaalcap コマンドの -n オプションの name オペランド) で指定した名称を指定してください。MCF アプリケーション属性定義の詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

このオプションは、入力メッセージ編集 UOC でアプリケーション名を決定する場合は不

要です。なお、UOC とこのオプションの両方でアプリケーション名を指定した場合、UOC のアプリケーション名が優先されます。

mcftgrpcn (コネクショングループ定義の開始)

形式

mcftgrpcn -g コネクショングループ名

機能

コネクショングループ定義の開始を示します。

オプション

-g コネクショングループ名 ~ 1 ~ 8 の文字の識別子

コネクショングループ名を指定します。

このコネクショングループ名は、ほかの mcftgrpcn コマンドの -g オプションで指定するコネクショングループ名と重複して指定できません。また、mcftalcle コマンドで指定する論理端末名称、mcftalcen コマンドで指定するコネクション ID とも重複して指定できません。

mcftgrped (コネクショングループ定義の終了)

形式

mcftgrped

機能

コネクショングループの定義の終了を示します。

オプション

なし。

システムサービス情報定義

MCF サービスはユーザが作るシステムサービスで、OpenTP1 のシステムサービスと同じ位置付けになります。

システムサービス情報定義では、MCF 通信サービスを起動するための環境を定義します。ユーザが MCF サービスを作成するときに定義する必要があります。

システムサービス情報定義は、テキストエディタを使用して作成します。

システムサービス情報定義の完全パス名を次に示します。

完全パス名：\$DCDIR/lib/sysconf/定義ファイル名

定義ファイル名には、システムサービス情報定義の module オペランドで指定する実行形式プログラム名を指定します。この定義ファイル名を MCF マネージャ定義の mcfmname コマンドに指定します。

形式

```
set module = "TP1/NET/X25-Extendedの実行形式プログラム名"
```

機能

プロセスサービスが MCF 通信サービスを起動するための環境を定義します。

各 MCF 通信サービスに対して一つ、システムサービス情報定義を作成できます。また、複数の MCF 通信サービスで一つのシステムサービス情報定義を共用することもできます。

説明

set 形式のオペランド

```
module = "TP1/NET/X25-Extended の実行形式プログラム名" ~ 1 ~ 8 文字の識別子
```

MCF 通信サービスを起動するための実行形式プログラム名を指定します。

MCF 実行形式プログラムには、MCF 通信プロセス用のものとアプリケーション起動プロセス用のものがあります。

MCF 実行形式プログラムは、MCF 通信プロセス同士、およびアプリケーション起動プロセス同士で共有できます。

TP1/NET/X25-Extended の実行形式プログラム名には、先頭 4 文字が mcfu で始まる最大 8 文字の名称を指定します。

システムサービス共通情報定義

TP1/NET/X25-Extended で定義したシステム構成の内容によっては、OpenTP1 のシステムサービス共通情報定義を指定する必要があります。

システムサービス共通情報定義の完全パス名を次に示します。

```
$DCDIR/lib/sysconf/mcf
```

形式

set 形式

```
set max_socket_descriptors=ソケット用ファイル記述子の最大数  
set max_open_fds=MCF通信プロセスでアクセスするファイルの最大数
```

機能

システムサービス共通情報定義では、複数の MCF 通信サービスに共通する情報を定義します。この定義ファイルは、標準値を定義した状態で製品に含まれています。次に示すオペランドについては、必要に応じて、テキストエディタを使用して定義値を変更してください。ほかのオペランドについては、変更しないでください。

説明

set 形式のオペランド

max_socket_descriptors= ソケット用ファイル記述子の最大数 ~ 符号なし整数
((64 ~ 2047))

各 MCF 通信プロセスでソケット用に使用するファイル記述子数の中の最大数を指定します。

ソケット用ファイル記述子の最大数を求める計算式を次に示します。 は、小数点以下を切り上げることを意味します。

(このMCF通信プロセスに対してメッセージ送信要求を行うUAPプロセス数¹ + システムサービスプロセス数² + このMCF通信プロセスに対して同時に処理要求を行う運用コマンド数) / 0.8

注 1

アプリケーション起動サーバに対するアプリケーション起動要求を行う UAP プロセス数も含まれます。

注 2

システムサービスプロセス数とは、自 OpenTP1 内のシステムサービスプロセス数

です。

自 OpenTP1 内の各 MCF 通信プロセスごとに計算し、その結果の中で最大値が 64 より大きい場合は、その値を指定します。64 以下の場合は、64 を指定します。

max_open_fds=MCF 通信プロセスでアクセスするファイルの最大数 ~ 符号なし整数 ((100 ~ 2016))

各 MCF 通信プロセスでアクセスするファイルの数の中の最大値を指定します。

ファイル記述子の最大数を求める計算式を次に示します。

(プロトコル制御で使用するファイル記述子数¹⁾
+ MCFメイン関数でユーザが使用するファイル記述子数
+ 30²)

注 1

TP1/NET/X25-Extended の場合、MCF 通信構成定義に定義したコネクションの総数を 2 倍した値になります。実際に通信を行うコネクションの総数ではありませんので、注意してください。

注 2

MCF 通信プロセスが扱う定義ファイルなどの数の最大値です。

自 OpenTP1 内の MCF 通信プロセスごとに計算し、その結果の中で最大値が 500 より大きい場合は、その値を指定します。500 以下の場合は、500 を指定します。指定値を超えてファイルのアクセスが発生した場合、その超過分はソケット用ファイル記述子使用数として扱われます。この場合、max_socket_descriptors オペランドの指定値から max_open_fds オペランドの指定値を減算した超過分が、実際のソケット用ファイル記述子の最大数になりますので、注意してください。

なお、このオペランドには、次の条件を満たす値を指定してください。

(「このオペランドの指定値」+ 同定義内の「max_socket_descriptors オペランドの指定値」) 2048

条件を満たさない指定をした場合、このオペランドの指定値は次に示すように強制的に補正されます。

2048 - (同定義内の「max_socket_descriptors オペランドの指定値」)

注意事項

max_socket_descriptors オペランドの指定値と max_open_fds オペランドの指定値の合計は、OS のシステムパラメタで指定する「1 プロセスでオープンできるファイル数」を

超えないようにする必要があります。超える場合は、OS のシステムパラメタの指定を変更してください。

MCF 定義オブジェクトの生成

MCF 定義オブジェクト生成ユーティリティでは、MCF の定義ファイルの構文のチェックと定義オブジェクトファイルへの変換をします。ここでは、MCF 定義オブジェクト生成ユーティリティの起動コマンドについて説明します。

形式

```
mcfx25ex -i 〔パス名〕入力ファイル名  
          -o 〔パス名〕出力オブジェクトファイル名
```

機能

MCF 通信構成定義の TP1/NET/X25-Extended プロトコル固有定義ファイルの構文をチェックし、定義オブジェクトファイルを生成します。

ただし、開始から再開始の間に定義オブジェクトファイルを変更してはいけません。変更した場合、再開始時に正常に動作しないことがありますのでご注意ください。

TP1/NET/X25-Extended のプロトコル固有定義以外のユーティリティについては、マニュアル「OpenTP1 システム定義」を参照してください。

オプション

-i〔パス名〕入力ファイル名 ~ パス名 1 ~ 8 文字の識別子

定義ソースが格納されているファイル名を指定します。

-o〔パス名〕出力オブジェクトファイル名 ~ パス名 1 ~ 8 文字の英数字

定義オブジェクトを格納するファイル名を指定します。

自システムの通信管理プログラムと関係づける内容

TP1/NET/X25-Extended を使用して AP 間通信をする場合、TP1/NET/X25-Extended の定義と自システム内の通信管理の定義とを一致させる必要があります。

通信管理が XNF/AS の場合

TP1/NET/X25-Extended の MCF 通信構成定義 (mcftalccn の `-z` オプション) で指定する仮想スロット番号を、XNF/AS 構成定義のリンク定義文 (link 文の VASS オペランド) の指定値と合わせます。XNF/AS 構成定義については、マニュアル「通信管理 XNF/AS 構成定義編」を参照してください。

定義例

ここでは、定義のコーディング例を示します。

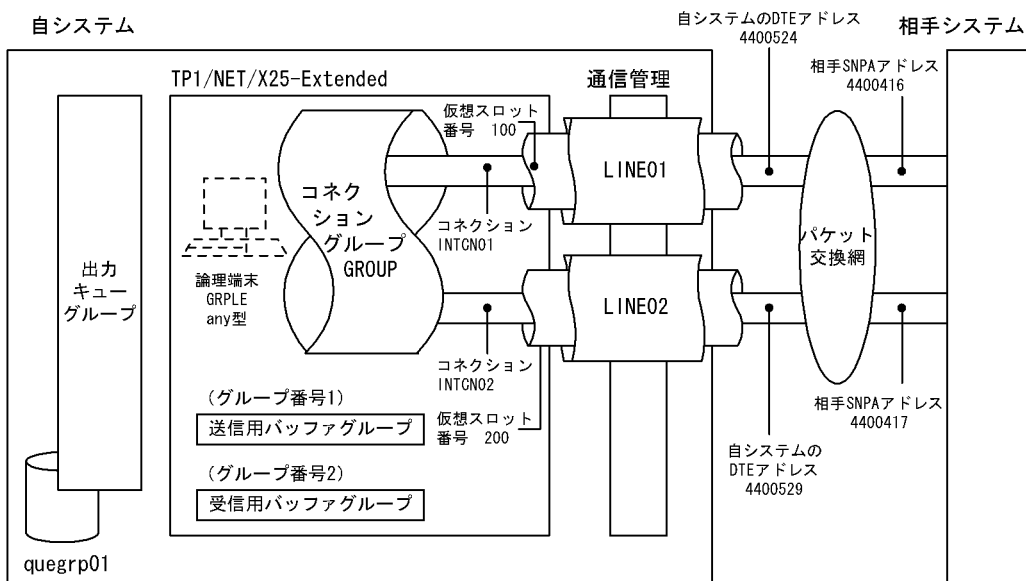
システム構成の例を図 5-2 に示します。また、この構成に対する XNF/AS 構成定義のコーディング例と、TP1/NET/X25-Extended の定義のコーディング例を示します。

TP1/NET/X25-Extended の定義のコーディング例を次のファイルで提供しています。

共通定義：/BeTRAN/examples/mcf/X25EX/conf/com_c

プロトコル固有定義：/BeTRAN/examples/mcf/X25EX/conf/com_d

図 5-2 TP1/NET/X25-Extended のシステム構成例



XNF/AS 構成定義のコーディング例

```

/
*****
*****/
/*                X25(VC)model                */
/
*****
*****/
configuration
  version 01
  max_NLI_VC 32
  max_VC_network_connection 32
  max_X25_link 150
  max_X25_VASS 150
  max_line 11

```

```

max_Line_adapter 1
max_link 11
;

Line_adapter
location_code 37-08
name LADP01
adapter_type HDLC
initial_status active
auto_start yes
;
group
name PASSVC01
type HDLC
;

line
name LINE01
number 00
line_type private_PS
line_mode 84VC
speed 9.6K
modem_clock synch_ST2
modem_type full
RS_control on_fixed
CDcheck no
NRZI no
;

link
name LINK01
VASS 100
data_link_address 1
data_link_address2 3
;

NL
DTE_address 4400524
min_VC_LCGN 13
max_VC_LCGN 15
min_VC_LCN 1
max_VC_LCN 20
send_window_size 2
receive_window_size 2
;

line
name LINE02
number 01
line_type private_PS
line_mode 84VC
speed 9.6K
modem_clock synch_ST2
modem_type full
RS_control on_fixed
CDcheck no
NRZI no

```

5. システム定義 定義例

```
;  
  
link  
    name LINK02  
    VASS 200  
    data_link_address 1  
    data_link_address2 3  
    ;  
  
NL  
    DTE_address 4400529  
    min_VC_LCGN 13  
    max_VC_LCGN 15  
    min_VC_LCN 1  
    max_VC_LCN 20  
    send_window_size 2  
    receive_window_size 2  
    ;
```

注

数字は MCF 通信構成定義の仮想スロット番号に対応します。

TP1/NET/X25-Extended の定義のコーディング例 (共通定義)

```
# MCF通信構成定義 (共通定義)  
#  
#  
#--- MCF環境定義  
#  
    mcftenv      -s 01                ¥  
                -a XEXapl  
#  
#--- MCF共通定義  
#  
    mcftcomn  
#  
#--- 最大処理多重度定義  
#  
    mcfttred  
#  
#--- タイマ定義  
#  
    mcfttim      -t "rmtim =60        ¥  
                  mtim = 600"  
#  
#--- トレース環境定義  
#  
    mcfttrc      -t "disk = yes"  
#  
#--- 送受信バッファ定義  
#  
#--- メッセージ送受信バッファ定義 (送信用)  
#  
    mcftbuf      -g "groupno = 1      ¥  
                  length = 10240     ¥
```



```

                                count = 100"
#
#--- メッセージ送受信バッファ定義(受信用)
#
    mcftbuf      -g "groupno = 2           ¥
                   length = 10240        ¥
                   count = 100"
#
#--- 入力編集UOC用バッファ定義
#
    mcftbuf      -g "groupno = 4           ¥
                   length = 32768        ¥
                   count = 60"
#
#--- 終わり
#

```

TP1/NET/X25-Extended の定義のコーディング例 (プロトコル固有定義)

```

#   MCF通信構成定義(プロトコル固有定義)
#
#
#--- コネクショングループ定義の開始
#
    mcftgrpcn    -g GROUP
#
#--- 論理端未定義
#
    mcftalcle    -l GRPLE                 ¥
                   -t any                 ¥
                   -m "mmsgcnt = 5        ¥
                       dmsgcnt = 5"       ¥
                   -k "quekind = disk     ¥
                       quegrp01"         ¥
                   -o "aj = yes"         ¥
                   -v aptpr001
#
#--- コネクション定義
#
    mcftalccn    -c INTCN01               ¥
                   -p x25ex               ¥
                   -g "sndbuf = 1         ¥
                       rcvbuf = 2"       ¥
                   -e "msgbuf = 4"       ¥
                   -i auto                 ¥
                   -z "vslot = 100"      ¥
                   -b "bretry = yes      ¥
                       bretrycnt = 5     ¥
                       bretryint = 10"   ¥
                   -t int                 ¥
                   -q "snpa = 4400416"   ¥
                   -f "kind = ccls"
    mcftalccn    -c INTCN02               ¥
                   -p x25ex               ¥
                   -g "sndbuf = 1         ¥
                       rcvbuf = 2"       ¥

```

5. システム定義 定義例

```
-e "msgbuf = 4"           ¥
-i auto                   ¥
-z "vslot = 200"         ¥
-b "bretry = yes         ¥
    bretrycnt = 5        ¥
    bretryint = 10"      ¥
-t int                     ¥
-q "snpa = 4400417"      ¥
-f "kind = ccls"         ¥

#
#--- コネクショングループ定義の終了
#
#    mcftgrpед
#
#--- 終わり
#
```

6

運用コマンド

OpenTP1 には、システムを運用するための幾つかの運用コマンドがあります。この章では、TP1/NET/X25-Extended で使用する運用コマンドについて説明します。

TP1/NET/X25-Extended の運用コマンド

mcftactcn (コネクションの確立)

mcftactle (論理端末の閉塞解除)

mcftdctcn (コネクションの解放)

mcftdctle (論理端末の閉塞)

mcftlscn (コネクションの状態表示)

mcftlsle (論理端末の状態表示)

TP1/NET/X25-Extended の運用コマンド

TP1/NET/X25-Extended のオンライン中に、運用コマンドを入力して種々の操作ができます。TP1/NET/X25-Extended で使用する運用コマンドを次の表に示し、それぞれについて説明します。

なお、ここでは、TP1/NET/X25-Extended に関係のあるオプションについてだけ説明しています。ほかのオプション、運用コマンドの入力方法、およびその他の運用コマンドについては、マニュアル「OpenTP1 運用と操作」を参照してください。

表 6-1 TP1/NET/X25-Extended の運用コマンド

機能		コマンド名称	定義中 組み込み	オフライン中 に実行	オンライン中 に実行	UAP から 実行
コネクション 管理	コネクションの確立	mcftactcn	×	×		
	コネクションの解放	mcftdctcn	×	×		
	コネクションの状態表示	mcftlscn	×	×		
論理端末管理	論理端末の閉塞解除	mcftactle	×	×		
	論理端末の閉塞	mcftdctle	×	×		
	論理端末の状態表示	mcftlsle	×	×		

(凡例)

○ : 組み込み, または実行ができます。

× : 組み込み, または実行ができません。

mcfactcn (コネクションの確立)

形式

```
mcfactcn [-s MCF通信プロセス識別子]
          {-c コネクションID | -g コネクショングループ名}
```

機能

コネクションを確立します。

オプション

`-s` MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信プロセスに対して、mcfactcn コマンドを実行します。

`-c` コネクション ID ~ 1 ~ 8 文字の識別子

確立するコネクションのコネクション ID を指定します。

コネクション ID は、一度に 8 個まで指定できます。複数指定するときは、引用符 (") で囲んで、コネクション ID とコネクション ID との間を空白で区切ります。同一コネクション ID は、重複して指定できません。

また、コネクション ID は、* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定をするときは、引用符 (") で囲んで指定します。

* : すべてのコネクションを確立します。

先行文字列 * : 先行文字列で始まるすべてのコネクションを確立します。

複数指定の例 cnn1 , cnn2 , cnn3 を指定する場合

```
-c "cnn1  cnn2  cnn3"
```

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

```
-c "cnn*"
```

`-g` コネクショングループ名 ~ 1 ~ 8 文字の識別子

確立するコネクショングループの名称を指定します。コネクショングループ内のすべてのコネクションを確立します。

コネクショングループ名は、一度に 8 個まで指定できます。複数指定するときは、引用

6. 運用コマンド

mcftacten (コネクションの確立)

符 (") で囲んで、コネクショングループ名とコネクショングループ名との間を空白で区切ります。同一コネクショングループ名は、重複して指定できません。

また、コネクショングループ名は、* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクショングループ名を混在して指定できません。一括指定をするときは、引用符 (") で囲んで指定します。

* : すべてのコネクショングループを確立します。

先行文字列 * : 先行文字列で始まるすべてのコネクショングループを確立します。

複数指定の例 gnn1, gnn2, gnn3 を指定する場合

```
-g "gnn1 gnn2 gnn3"
```

一括指定の例 gnn で始まるすべてのコネクショングループを指定する場合

```
-g "gnn*"
```

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftacten コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftacten コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftacten コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftacten コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10500-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	mcftacten コマンドで指定したコネクショングループ名は登録されていません。	標準エラー出力
KFCA17730-W	MCF 運用コマンド処理中に異常が発生しました。	標準エラー出力

出力メッセージ ID	内容	出力先
KFCA17732-W	コネクションが確立済みのため運用コマンドは受け付けられません。	標準エラー出力
KFCA17733-W	コネクション確立処理中のため運用コマンドは受け付けられません。	標準エラー出力
KFCA17734-W	コネクション解放処理中のため運用コマンドは受け付けられません。	標準エラー出力
KFCA17739-W	着呼モードコネクションのため運用コマンドは受け付けられません。	標準エラー出力

mcftactle (論理端末の閉塞解除)

形式

```
mcftactle [-s MCF通信プロセス識別子] -l 論理端末名称
```

機能

論理端末の閉塞を解除します。

オプション

```
-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))
```

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信プロセスに対して、mcftactle コマンドを実行します。

```
-l 論理端末名称 ~ 1 ~ 8 文字の識別子
```

閉塞解除する論理端末の名称を指定します。

論理端末名称は、一度に 8 個まで指定できます。複数指定するときは、引用符 (") で囲んで、論理端末名称と論理端末名称との間を空白で区切ります。同一論理端末名称は、重複して指定できません。

また、論理端末名称は、* を使って一括指定ができます。一括指定は一つだけ指定できません。一括指定と一括指定以外の論理端末名称を混在して指定できません。一括指定をするときは、(") で囲んで指定します。

* : すべての論理端末の閉塞を解除します。

先行文字列 * : 先行文字列で始まるすべての論理端末の閉塞を解除します。

複数指定の例 len1, len2, len3 を指定する場合

```
-l "len1 len2 len3"
```

一括指定の例 len で始まるすべての論理端末を指定する場合

```
-l "len*"
```

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftactle コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力

6. 運用コマンド
mcfactle (論理端末の閉塞解除)

出力メッセージ ID	内容	出力先
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcfactle コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcfactle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcfactle コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定したコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10391-E	mcfactle コマンドはサポートされていません。	標準エラー出力
KFCA10395-E	指定したコネクションには未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10503-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA17731-W	コネクションが未確立のため運用コマンドは受け付けられません。	標準エラー出力
KFCA17736-W	論理端末が閉塞解除済みのため運用コマンドは受け付けられません。	標準エラー出力
KFCA17750-I	論理端末を閉塞解除しました。	メッセージログファイル

mcftdctcn (コネクションの解放)

形式

```
mcftdctcn [-s MCF通信プロセス識別子]
           {-c コネクションID | -g コネクショングループ名} [-f]
```

機能

コネクションを解放します。

指定したコネクションに仕掛り中のメッセージがある場合でも、コネクションは強制的に解放されます。

オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信プロセスに対して、mcftdctcn コマンドを実行します。

-c コネクション ID ~ 1 ~ 8 文字の識別子

解放するコネクションのコネクション ID を指定します。

コネクション ID は、一度に 8 個まで指定できます。複数指定するときは、引用符 (") で囲んで、コネクション ID とコネクション ID との間を空白で区切ります。

同一コネクション ID は、重複して指定できません。

また、コネクション ID は、* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定をするときは、引用符 (") で囲んで指定します。

* : すべてのコネクションを解放します。

先行文字列 * : 先行文字列で始まるすべてのコネクションを解放します。

複数指定の例 cnn1, cnn2, cnn3 を指定する場合

```
-c "cnn1    cnn2    cnn3"
```

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

```
-c "cnn*"
```

-g コネクショングループ名 ~ 1 ~ 8 文字の識別子

解放するコネクショングループの名称を指定します。コネクショングループ内のすべて

のコネクションを解放します。

コネクショングループ名は、一度に 8 個まで指定できます。複数指定するときは、引用符 (") で囲んで、コネクショングループ名とコネクショングループ名との間を空白で区切ります。同一コネクショングループ名は、重複して指定できません。

また、コネクショングループ名は、* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクショングループ名を混在して指定できません。一括指定をするときは、引用符 (") で囲んで指定します。

* : すべてのコネクショングループを解放します。

先行文字列 * : 先行文字列で始まるすべてのコネクショングループを解放します。

複数指定の例 gnn1 , gnn2 , gnn3 を指定する場合

```
-g "gnn1 gnn2 gnn3"
```

一括指定の例 gnn で始まるすべてのコネクショングループを指定する場合

```
-g "gnn*"
```

```
-f
```

該当するコネクションを強制的に解放します。

TP1/NET/X25-Extended では、このオプションを指定しても指定しなくても、動作に影響はありません。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcfstdctcn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcfstdctcn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcfstdctcn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcfstdctcn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力

6. 運用コマンド

mcftdctn (コネクションの解放)

出力メッセージ ID	内容	出力先
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10501-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	mcftdctn コマンドで指定したコネクショングループ名は登録されていません。	標準エラー出力
KFCA17731-W	コネクションが未確立のため運用コマンドは受け付けられません。	標準エラー出力
KFCA17733-W	コネクション確立処理中のため運用コマンドは受け付けられません。	標準エラー出力
KFCA17734-W	コネクション解放処理中のため運用コマンドは受け付けられません。	標準エラー出力

mcftdctl (論理端末の閉塞)

形式

```
mcftdctl [-s MCF通信プロセス識別子] -l 論理端末名称
```

機能

論理端末を閉塞します。

オプション

```
-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))
```

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信プロセスに対して、mcftdctl コマンドを実行します。

```
-l 論理端末名称 ~ 1 ~ 8 文字の識別子
```

閉塞する論理端末の名称を指定します。

論理端末名称は、一度に 8 個まで指定できます。複数指定するときは、引用符 (") で囲んで、論理端末名称と論理端末名称との間を空白で区切ります。同一論理端末名称は、重複して指定できません。

また、論理端末名称は、* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外の論理端末名称を混在して指定できません。一括指定をするときは、引用符 (") で囲んで指定します。

* : すべての論理端末を閉塞します。

先行文字列 * : 先行文字列で始まるすべての論理端末を閉塞します。

複数指定の例 len1, len2, len3 を指定する場合

```
-l "len1 len2 len3"
```

一括指定の例 len で始まるすべての論理端末を指定する場合

```
-l "len*"
```

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftdctl コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力

6. 運用コマンド
mcfddctle (論理端末の閉塞)

出力メッセージ ID	内容	出力先
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcfddctle コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcfddctle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcfddctle コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定されたコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10395-E	指定したコネクションには未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10504-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA17735-W	論理端末が閉塞済みのため運用コマンドは受け付けられません。	標準エラー出力
KFCA17737-W	論理端末が使用中のため運用コマンドは受け付けられません。	標準エラー出力
KFCA17751-I	論理端末を閉塞しました。	メッセージログファイル

注意事項

- 受信仕掛り中に運用コマンド (mcfddctle) を入力した場合は、論理端末が閉塞状態でも、メッセージを受信します。論理端末閉塞によるメッセージ受信処理への影響はありません。
- 送信仕掛り中に運用コマンド (mcfddctle) を入力した場合は、運用コマンドがエラーリターンします。送信仕掛り中でない場合は、論理端末は閉塞されます。

mcftlscn (コネクションの状態表示)

形式

```
mcftlscn [-s MCF通信プロセス識別子]
          [-c コネクションID | -g コネクショングループ名] [-d]
```

機能

コネクションの状態を標準出力に出力します。

オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信プロセスに対して、mcftlscn コマンドを実行します。

-c コネクション ID ~ 1 ~ 8 文字の識別子

状態を表示するコネクションのコネクション ID を指定します。

コネクション ID は、一度に 8 個まで指定できます。複数指定するときは、引用符 (") で囲んで、コネクション ID とコネクション ID との間を空白で区切ります。同一コネクション ID は、重複して指定できません。

また、コネクション ID は、* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定をするときは、引用符 (") で囲んで指定します。

* : すべてのコネクションの状態を表示します。

先行文字列 * : 先行文字列で始まるすべてのコネクションの状態を表示します。

複数指定の例 cnn1 , cnn2 , cnn3 を指定する場合

```
-c "cnn1  cnn2  cnn3"
```

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

```
-c "cnn*"
```

-g コネクショングループ名 ~ 1 ~ 8 文字の識別子

状態を表示するコネクショングループの名称を指定します。コネクショングループ内のすべてのコネクションの状態を表示します。

コネクショングループ名は、一度に 8 個まで指定できます。複数指定するときは、引用

6. 運用コマンド

mcftlscn (コネクションの状態表示)

符 (") で囲んで、コネクショングループ名とコネクショングループ名との間を空白で区切ります。同一コネクショングループ名は、重複して指定できません。

また、コネクショングループ名は、* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクショングループ名を混在して指定できません。一括指定をするときは、引用符 (") で囲んで指定します。

* : すべてのコネクショングループの状態を表示します。

先行文字列 * : 先行文字列で始まるすべてのコネクショングループの状態を表示します。

複数指定の例 gnn1, gnn2, gnn3 を指定する場合

```
-g "gnn1 gnn2 gnn3"
```

一括指定の例 gnn で始まるすべてのコネクショングループを指定する場合

```
-g "gnn*"
```

```
-d
```

コネクションの状態と該当するコネクションに対応する論理端末の情報を表示します。

このオプションの指定を省略すると、コネクションの状態だけを表示します。

出力形式

```
mmm cccccccc ppp sssss gggggggg  
  lllllllll ttt ss vv
```

注

-d オプションを指定しないで mcftlscn コマンドを実行した場合は、「mmm cccccccc ppp sssss gggggggg」の行だけ出力されます。

- mmm : MCF 識別子
- cccccccc : コネクション ID
- ppp : プロトコル種別
X2E...X.25 プロトコル (TP1/NET/X25-Extended)
- sssss : コネクション状態
ACT...確立
ACT/B...確立処理中
DCT...解放
DCT/B...解放処理中
- gggggggg : コネクショングループ名
- llllllll : 論理端末名称
- ttt : 論理端末の端末タイプ
ANY...any 型
- ss : 保守情報 1

- vv : 保守情報 2

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftlscn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftlscn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10360-I	状態表示を開始します。	標準出力
KFCA10361-I	標準情報を表示します。	標準出力
KFCA10361-I	詳細情報を表示します。	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10371-I	mcftlscn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftlscn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10502-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	mcftlscn コマンドで指定したコネクショングループ名は登録されていません。	標準エラー出力

mcftlsle (論理端末の状態表示)

形式

```
mcftlsle [-s MCF通信プロセス識別子] -l 論理端末名称 [-q] [-t]
```

機能

論理端末の状態を標準出力に出力します。

オプション

`-s` MCF 通信プロセス識別子 `~` 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信プロセスに対して、mcftlsle コマンドを実行します。

`-l` 論理端末名称 `~` 1 ~ 8 文字の識別子

状態を表示する論理端末の名称を指定します。

論理端末名称は、一度に 8 個まで指定できます。複数指定するときは、引用符 (") で囲んで、論理端末名称と論理端末名称との間を空白で区切ります。同一論理端末名称は、重複して指定できません。

また、論理端末名称は、* を使って一括指定ができます。一括指定は一つだけ指定できません。一括指定と一括指定以外の論理端末名称を混在して指定できません。一括指定をするときは、引用符 (") で囲んで指定します。

* : すべての論理端末の状態を表示します。

先行文字列 * : 先行文字列で始まるすべての論理端末の状態を表示します。

複数指定の例 len1, len2, len3 を指定する場合

```
-l "len1 len2 len3"
```

一括指定の例 len で始まるすべての論理端末を指定する場合

```
-l "len*"
```

`-q`

指定した論理端末に対応する出力キューの保留状態を表示します。

このオプションの指定を省略すると、論理端末に対応する出力キューの保留状態は表示しません。

-t

指定した論理端末がセキュア状態かどうかを表示します。

出力形式

```
mmm 11111111 sss [ggg] [tttt]
  SYNC xxxxxxxxxxxx yyyyyyyyyy zzzzzzzzzz
    IO      :           :           :
  PRIO     :           :           :
  NORM     :           :           :
  iii ooo
```

- mmm : MCF 識別子
- 11111111 : 論理端末名称
- sss : 論理端末状態
ACT...閉塞解除状態
DCT...閉塞状態
- ggg : 論理端末のセキュア状態 (-t オプション指定時だけ表示)
NOS...非セキュア状態
SEC...セキュア状態
- tttt : 論理端末のテストモード状態 (TP1/Message Control/Tester 使用時だけ表示)
TEST...テストモード
空白...非テストモード
- SYNC : 同期型メッセージ
- IO : 非同期型問い合わせ応答メッセージ
- PRIO : 非同期型一方送信メッセージ (優先)
- NORM : 非同期型一方送信メッセージ (一般)
- xxxxxxxxxxxx : 未送信メッセージ数
- yyyyyyyyyy : 未送信メッセージ最小通番
- zzzzzzzzzz : 未送信メッセージ最大通番
- iii : 出力キューの入力の保留状態 (-q オプション指定時だけ表示)
NOH...保留解除
HLD...保留
- ooo : 出力キューの出力の保留状態 (-q オプション指定時だけ表示)
NOH...保留解除
HLD...保留

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcfllsle コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力

6. 運用コマンド

mcftlsle (論理端末の状態表示)

出力メッセージ ID	内容	出力先
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftlsle コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10360-I	状態表示を開始します。	標準出力
KFCA10364-I	表示情報	標準出力
KFCA10365-I	表示情報	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10371-I	mcftlsle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftlsle コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定したコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10391-E	mcftlsle コマンドはサポートされていません。	標準エラー出力
KFCA10395-E	指定したコネクションには未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10505-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

7

組み込み方法

この章では、TP1/NET/X25-Extended を OpenTP1 システムへ組み込む方法について説明します。

7.1 TP1/NET/X25-Extended の組み込み方法の流れ

7.2 MCF メイン関数の作成

7.3 定義オブジェクトファイルの生成

7.1 TP1/NET/X25-Extended の組み込み方法の流れ

TP1/NET/TCP/IP を OpenTP1 システムに組み込むときの作業の流れを示します。

(1) MCF メイン関数の作成

TP1/NET/X25-Extended を起動するためには、MCF メイン関数をコーディングし、コンパイル、およびリンクージュしておく必要があります。詳細は、「7.2 MCF メイン関数の作成」を参照してください。

(2) MCF サービス名の登録

TP1/NET/X25-Extended を実行するために、MCF サービス名をシステムサービス構成定義で定義しておく必要があります。

MCF サービス名は MCF マネージャ定義オブジェクトファイル名と一致させてください。

詳細は、マニュアル「OpenTP1 システム定義」を参照してください。

(3) システムサービス情報定義ファイルの作成

システムサービス情報定義ファイルをテキストエディタで作成します。作成するファイルのパス名は、「\$DCDIR/lib/sysconf/ システムサービス情報定義ファイル名」です。

ファイルの定義形式については、5 章の「システムサービス情報定義」を参照してください。

(4) 定義オブジェクトファイルの生成

OpenTP1 のネットワークコミュニケーション定義の各ソースファイルから定義オブジェクトファイルを生成します。詳細は、「7.3 定義オブジェクトファイルの生成」を参照してください。

7.2 MCF メイン関数の作成

TP1/NET/X25-Extended は、OpenTP1 プロセスサービスによって起動されます。

TP1/NET/X25-Extended を起動するには、ユーザが MCF メイン関数をコーディングし、コンパイル、およびリンケージを行って TP1/NET/X25-Extended の実行形式プログラムを作成する必要があります。リンケージには、mcfplx25ex コマンドを使用します。

MCF メイン関数では、スタート関数 (dc_mcf_svstart) を呼び出します。UOC を使用する場合は、MCF メイン関数で UOC 関数アドレスを指定してください。

UOC は、MCF メイン関数と同じ言語 (ANSI C, C++ または K&R 版 C) で作成してください。

MCF メイン関数のコーディング概要を図 7-1、図 7-2 に示します。また、ディレクトリへの組み込み方法を図 7-3 に示します。

なお、これらのコーディング例は、次のファイルで提供されています。

- /BeTRAN/examples/mcf/X25EX/cmlib/ansi/com.c
- /BeTRAN/examples/mcf/X25EX/cmlib/c/com.c

図 7-1 MCF メイン関数のコーディング概要 (ANSI C, C++ の場合)

```

#include <dcxep.h> /*TP1/NET/X25-Extended用ヘッダファイル */ 1
extern DCLONG msgrcv01(dcmcf_uoc_min_n *); /*入力メッセージ編集UOC関数extern宣言*/ 2
extern DCLONG msgsend01(dcmcf_uoc_mout_n *); /*出力メッセージ編集UOC関数extern宣言*/ 2
extern dcmcf_uoc_t dcmcf_uoctbl; /*UOCテーブルextern宣言 */ 3
int main()
{
    dcmcf_uoctbl.msgrcv =(dcmcf_uocfunc)msgrcv01; /*入力メッセージ編集UOCアドレス設定*/ 4
    dcmcf_uoctbl.msgsend =(dcmcf_uocfunc)msgsend01; /*出力メッセージ編集UOCアドレス設定*/ 4
    dc_mcf_svstart(); /*スタート関数の呼び出し */ 5
    return 0;
}

```

7. 組み込み方法

図 7-2 MCF メイン関数のコーディング概要 (K&R 版 C の場合)

```
#include <dcmxep.h> /*TP1/NET/X25-Extended用ヘッダファイル*/ 1
extern DCLONG msgrcv01(); /*入力メッセージ編集UOC関数extern宣言*/ 2
extern DCLONG msgsend01(); /*出力メッセージ編集UOC関数extern宣言*/ 2
extern dcmcf_uoc_t dcmcf_uoctbl; /*UOCテーブルextern宣言 */ 3
main()
{
    dcmcf_uoctbl.msgrcv = msgrcv01; /*入力メッセージ編集UOCアドレス設定 */ 4
    dcmcf_uoctbl.msgsend = msgsend01; /*出力メッセージ編集UOCアドレス設定 */ 4
    dc_mcf_svstart(); /*スタート関数の呼び出し */ 5
}
```

1. TP1/NET/X25-Extended で提供するヘッダファイルを取り込みます。
2. 使用する UOC 関数を extern 宣言します。UOC のリターン値は DCLONG 型にしてください。

UOC をまったく使用しない場合、このコーディングは必要ありません。

3. UOC テーブルを extern 宣言します。UOC を使用する場合、必ずこのとおりにコーディングしてください。

UOC をまったく使用しない場合、このコーディングは必要ありません。

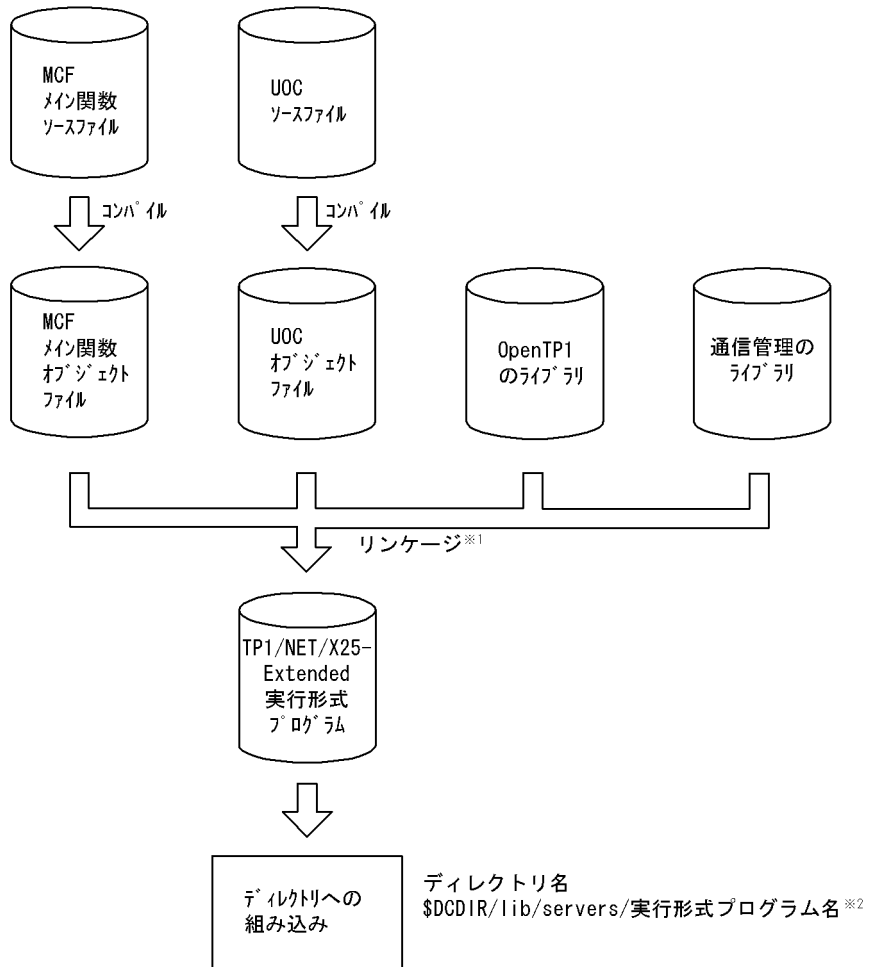
4. 各 UOC 関数のアドレスを、次に示すシステム提供変数に設定します。使用する UOC に対するコーディングだけが必要です。

```
dcmcf_uoctbl.msgrcv /*入力メッセージ編集UOCアドレス*/
dcmcf_uoctbl.msgsend /*出力メッセージ編集UOCアドレス*/
```

UOC をまったく使用しない場合、このコーディングは必要ありません。

5. スタート関数を呼び出します。MCF メイン関数には必ずコーディングしてください。

図 7-3 MCF メイン関数のディレクトリへの組み込み方法



注 1
mcfplx25ex コマンドでリンケージします。

注 2
TP1/NET/X25-Extended の実行形式プログラム名は、先頭が mcfu で始まる 8 文字以内の名称にしてください。

7.3 定義オブジェクトファイルの生成

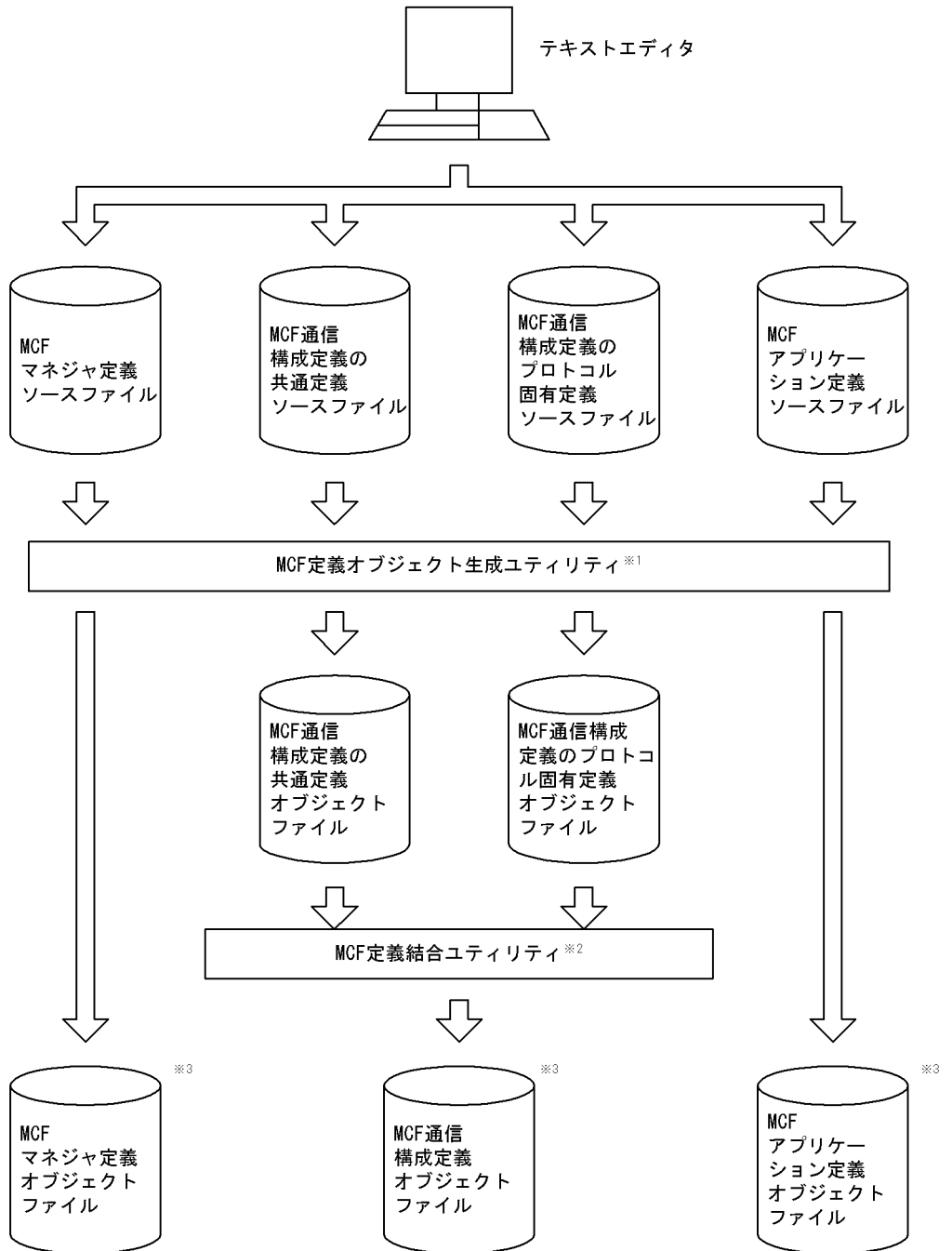
定義オブジェクトファイルを次の手順で作成します。

ただし、開始から再開始の間に定義オブジェクトファイルを変更しないでください。変更した場合、再開始時に正常に動作しない場合がありますのでご注意ください。

1. テキストエディタを使用して、MCF の定義ファイルから、次に示す定義ソースファイルを作成します。
 - MCF マネージャ定義ソースファイル
 - MCF 通信構成定義の共通定義ソースファイル
 - MCF 通信構成定義の TP1/NET/X25-Extended のプロトコル固有定義ソースファイル
 - MCF アプリケーション定義ソースファイル
2. MCF 定義オブジェクト生成ユーティリティを使用して、定義ソースファイルから、次に示すオブジェクトファイルを作成します。
 - MCF マネージャ定義オブジェクトファイル
 - MCF 通信構成定義の共通定義オブジェクトファイル
 - MCF 通信構成定義の TP1/NET/X25-Extended のプロトコル固有定義オブジェクトファイル
 - MCF アプリケーション定義オブジェクトファイル
3. MCF 定義結合ユーティリティを使用して、MCF 通信構成定義の共通定義とプロトコル固有定義のオブジェクトファイルを結合し、次に示すオブジェクトファイルを作成します。
 - MCF 通信構成定義オブジェクトファイル

定義オブジェクトファイルの作成方法の概要を次の図に示します。

図 7-4 定義オブジェクトファイルの作成方法の概要



注 1

次に示すコマンドで生成します。

7. 組み込み方法

```
mcf x x x x -i 〔パス名〕入力ファイル名  
             -o 〔パス名〕出力オブジェクトファイル名
```

mcf x x x x は、ソースファイルごとに異なります。

- mcfmngrr : MCF マネージャ定義ソースファイル
- mcfcomn : MCF 通信構成定義の共通定義ソースファイル
- mcfx25ex : MCF 通信構成定義のプロトコル (TP1/NET/X25-Extended) 固有定義ソースファイル
- mcfapli : MCF アプリケーション定義ソースファイル

MCF 定義オブジェクト生成ユーティリティの mcfx25ex コマンドについては 5 章の「MCF 定義オブジェクトの生成」を、その他のコマンドについてはマニュアル「OpenTP1 システム定義」を参照してください。

注 2

次に示すコマンドで、MCF 通信構成定義の二つのオブジェクトファイルを結合します。

```
mcf link -i 共通定義オブジェクトファイル名  
          TP1/NET/X25-Extended定義オブジェクトファイル名  
          -o 出力オブジェクトファイル名
```

注 3

定義オブジェクトファイルは、システム環境定義の DCCONFPATH で指定したディレクトリに格納してください。システム環境定義については、マニュアル「OpenTP1 システム定義」を参照してください。

8

障害対策

この章では、運用中に発生する障害と TP1/NET/X25-Extended の対応処理について説明します。

8.1 障害の種類と対応処理

8.1 障害の種類と対応処理

システム運用中に障害が発生すると、TP1/NET/X25-Extended は、エラーメッセージを出力したり、MCF イベントを通知したりします。ユーザが MCF イベント処理用 MHP を作成してシステム定義で登録しておくこと、障害に応じたユーザ任意の回復処理ができます。

TP1/NET/X25-Extended 運用中の障害と対応処理について、障害の種類ごとに示します。

(1) コネクション障害

表 8-1 コネクション障害と対応処理

障害内容	TP1/NET/X25-Extended の処理	ユーザの処理
コネクション確立時回復不可能障害	<ol style="list-style-type: none"> 1. コネクション障害を通知するメッセージログ (KFCA16902-E) を出力します。 2. コネクション確立失敗を通知するメッセージログ (KFCA16903-E) を出力します。 3. CERREVT (コネクション確立不可能) を起動します。 	障害の要因を取り除いたあと、mcftacten コマンドを入力します。
コネクション確立時回復可能障害	<ol style="list-style-type: none"> 1. コネクション障害を通知するメッセージログ (KFCA16902-E) を出力します。 2. リトライタイマの指定によって、一定時間経過後、確立を再試行します。 	ありません。
	リトライアウトした場合 <ol style="list-style-type: none"> 1. コネクション障害を通知するメッセージログ (KFCA16902-E) を出力します。 2. コネクション確立失敗を通知するメッセージログ (KFCA16903-E) を出力します。 3. CERREVT (コネクション確立不可能) を起動します。 	障害の要因を取り除いたあと、mcftacten コマンドを入力します。
コネクション切断	<ol style="list-style-type: none"> 1. コネクション解放を通知するメッセージログ (KFCA16904-I) を出力します。 2. CERREVT (コネクション切断), または CCLSEVT を起動します。 	<ul style="list-style-type: none"> • mcftacten コマンドを入力します。 • 相手からの確立要求を待ちます。
メッセージ送信時障害 (回復不可能)	<ol style="list-style-type: none"> 1. コネクション障害を通知するメッセージログ (KFCA16902-E) を出力します。 2. メッセージ送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 3. CERREVT (コネクション切断) を起動します。 4. ほかの正常なコネクションを使用して再送します。正常なコネクションがない場合、メッセージは出力キューに残し、再開時に再送されます。 	<ul style="list-style-type: none"> • mcftacten コマンドを入力します。 • 相手からの確立要求を待ちます。

障害内容	TP1/NET/X25-Extended の処理	ユーザの処理
メッセージ送信時リセットパケット受信	<ol style="list-style-type: none"> リセット受信を通知するメッセージログ (KFCA16916-I) を出力します。 コネクション障害を通知するメッセージログ (KFCA16902-E) を出力します。 メッセージ送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 CERREVT (リセットパケット受信) を起動します。 ほかの正常なコネクションを使用して再送します。正常なコネクションがない場合、メッセージは出力キューに残し、再開時に再送されます。 	<ul style="list-style-type: none"> mcftactcn コマンドを入力します。 相手からの確立要求を待ちます。
受信バッファオーバーフロー	<ol style="list-style-type: none"> バッファオーバーフローを通知するメッセージログ (KFCA16911-E) を出力します。 コネクション障害を通知するメッセージログ (KFCA16902-E) を出力します。 CERREVT (受信バッファオーバーフロー) を起動します。 受信メッセージを破棄します。 	システム定義を修正します。
受信バッファ不足	<ol style="list-style-type: none"> バッファ不足を通知するメッセージログ (KFCA13213-E) を出力します。 コネクション障害を通知するメッセージログ (KFCA16902-E) を出力します。 CERREVT (受信バッファ不足) を起動します。 	システム定義を修正します。
メッセージ受信時リセットパケット受信	<ol style="list-style-type: none"> リセット受信を通知するメッセージログ (KFCA16916-I) を出力します。 コネクション障害を通知するメッセージログ (KFCA16902-E) を出力します。 メッセージ送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 CERREVT (リセットパケット受信) を起動します。 受信メッセージを破棄します。 	<ul style="list-style-type: none"> mcftactcn コマンドを入力します。 相手からの確立要求を待ちます。

(2) コネクショングループ障害

表 8-2 コネクショングループ障害と対応処理

障害内容	TP1/NET/X25-Extended の処理	ユーザの処理
グループ内のコネクションが一つも使用できない	論理端末閉塞を通知するメッセージログ (KFCA17751-I) を出力します。	グループ内のコネクションを再確立します。

8. 障害対策

(3) 受信スケジュール関係の障害(入力キュー, 入力メッセージ編集 UOC)

表 8-3 受信スケジュール関係の障害と対応処理

障害内容	TP1/NET/X25-Extended の処理	ユーザの処理
ユーザアプリケーション名未定義	<ol style="list-style-type: none"> ERREVT1 を起動します。 入力キュー障害を通知するメッセージログ (KFCA10604-E) を出力します。 コネクション解放を通知するメッセージログ (KFCA16901-I) を出力します。 CERREVT (コネクション切断) を起動します。 	システム定義を修正します。
MCF イベント未定義 • ERREVT x • C x x x EVT	メッセージおよびイベントを破棄します。	ありません。
MHP サービス, サービスグループ閉塞	<ol style="list-style-type: none"> サービス閉塞を通知するメッセージログ (KFCA11051-E) を出力します。 ERREVT2 を起動します。 入力キュー障害を通知するメッセージログ (KFCA10604-E) を出力します。 コネクション解放を通知するメッセージログ (KFCA16901-I) を出力します。 CERREVT (コネクション切断) を起動します。 	<ul style="list-style-type: none"> サービス, サービスグループの閉塞解除後, mcfacten コマンドを入力します。 サービス, サービスグループの閉塞解除後, 相手からの確立要求を待ちます。
入力キュー書き込み障害	<ol style="list-style-type: none"> ERREVT2 を起動します。 入力キュー障害を通知するメッセージログ (KFCA10604-E) を出力します。 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。 受信メッセージを破棄します。 コネクション解放を通知するメッセージログ (KFCA16901-I) を出力します。 CERREVT (コネクション切断) を起動します。 	<ul style="list-style-type: none"> 障害の要因を取り除いたあと, mcfacten コマンドを入力します。 障害の要因を取り除いたあと, 相手からの確立要求を待ちます。
入力メッセージ編集 UOC エラーリターン, またはパラメタ不正	<ol style="list-style-type: none"> UOC エラーリターンを通知するメッセージログ (KFCA10611-E), または UOC パラメタ不正を通知するメッセージログ (KFCA10620-E) を出力します。 CERREVT (UOC 障害) を起動します。 受信メッセージを破棄します。 コネクション解放を通知するメッセージログ (KFCA16901-I) を出力します。 	入力メッセージ編集 UOC を見直してください。
アプリケーション名形式不正	<ol style="list-style-type: none"> アプリケーション名取得失敗を通知するメッセージログ (KFCA10610-E) を出力します。 ERREVT1 を起動します。 コネクション解放を通知するメッセージログ (KFCA16901-I) を出力します。 CERREVT (コネクション切断) を起動します。 	アプリケーション名取得失敗の要因を取り除いてください。

(4) 送信スケジュール関係の障害（出力キュー，出力メッセージ編集 UOC）

表 8-4 送信スケジュール関係の障害と対応処理

障害内容	TP1/NET/X25-Extended の処理	ユーザの処理
出力キュー読み込み障害	<ol style="list-style-type: none"> 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 コネクション解放を通知するメッセージログ (KFCA16901-I) を出力します。 メッセージ送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 CERREVT (出力キュー読み込み障害) を起動します。 ほかの正常なコネクションを使用して再送します。正常なコネクションがない場合、メッセージは出力キューに残し、再開時に再送されます。 	<ul style="list-style-type: none"> 障害の要因を取り除いたあと、mftactcn コマンドを入力します。 障害の要因を取り除いたあと、相手からの確立要求を待ちます。
出力メッセージ編集 UOC エラーリターン，またはパラメタ不正	<ol style="list-style-type: none"> UOC エラーリターンを通知するメッセージログ (KFCA10611-E)，または UOC パラメタ不正を通知するメッセージログ (KFCA10620-E) を出力します。 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 メッセージ送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 コネクション解放を通知するメッセージログ (KFCA16901-I) を出力します。 CERREVT (UOC 障害) を起動します。 ほかの正常なコネクションを使用して再送します。正常なコネクションがない場合、メッセージは出力キューに残し、再開時に再送されます。 	出力メッセージ編集 UOC を見直してください。
メッセージ消し込み障害	<ol style="list-style-type: none"> 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 処理を続行します。 	ありません。
送信バッファ不足	<ol style="list-style-type: none"> バッファ不足を通知するメッセージログ (KFCA10618-E) を出力します。 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 メッセージ送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 コネクション解放を通知するメッセージログ (KFCA16901-I) を出力します。 CERREVT (送信バッファ不足) を起動します。 ほかの正常なコネクションを使用して再送します。正常なコネクションがない場合、メッセージは出力キューに残し、再開時に再送されます。 	システム定義を修正します。

8. 障害対策

(5) UAP 障害

表 8-5 UAP 障害と対応処理

障害内容	TP1/NET/X25-Extended の処理	ユーザの処理
セグメント受信前の UAP 異常終了	1. ERREVT2 を起動します。	UAP を見直してください。
セグメント受信後の UAP 異常終了	1. ERREVT3 を起動します。	UAP を見直してください。

(6) MCF 障害

表 8-6 MCF 障害と対応処理

障害内容	TP1/NET/X25-Extended の処理	ユーザの処理
<ul style="list-style-type: none"> • 内部論理矛盾 • 他アプリケーションプログラムとのインタフェースエラー 	1. 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。 2. 内部論理矛盾を通知するメッセージログ (KFCA16997-E, KFCA16999-E, KFCA17797-E, または KFCA17799-E) を出力します。	保守情報 (\$DCDIR/spool ディレクトリ以下) を回避します。
	KFCA16997-E または KFCA17797-E の場合 3. 処理を続行します。	
	KFCA16999-E または KFCA17799-E の場合 3. MCF を異常終了します。	

付録

付録 A メッセージ送受信の処理の流れ

付録 B 障害発生時の処理の流れ

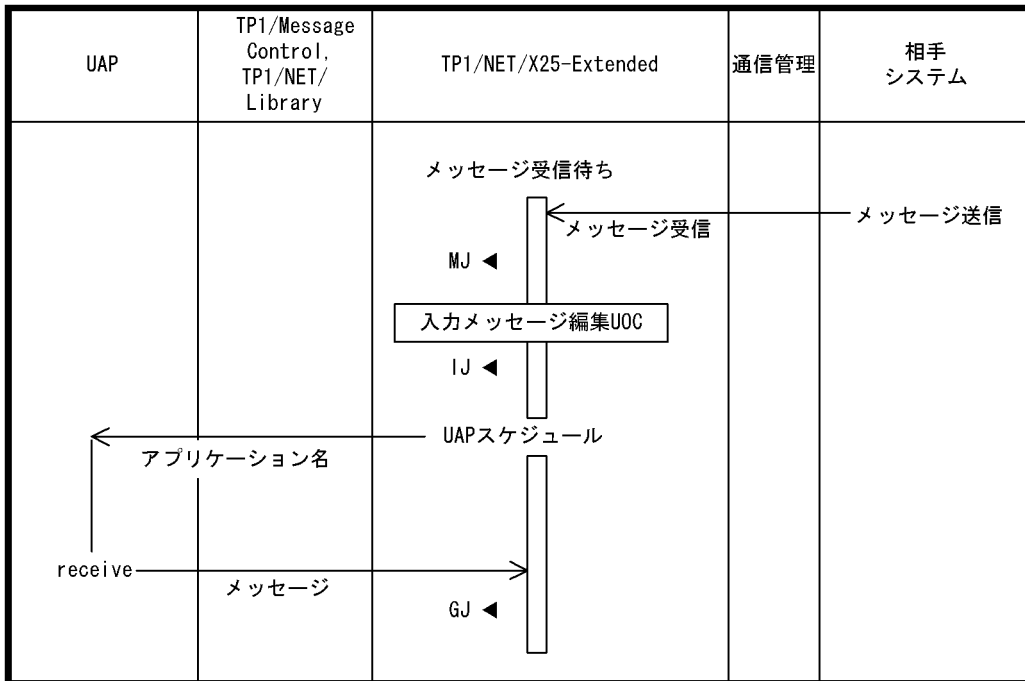
付録 C 理由コード一覧

付録 D 用語解説

付録 A メッセージ送受信の処理の流れ

メッセージを送受信するときのデータの流れ，ジャーナルの取得タイミングなどを示します。受信の場合と送信の場合，パケット分割がある場合とない場合に分けて，図 A-1 ~ 図 A-4 に示します。

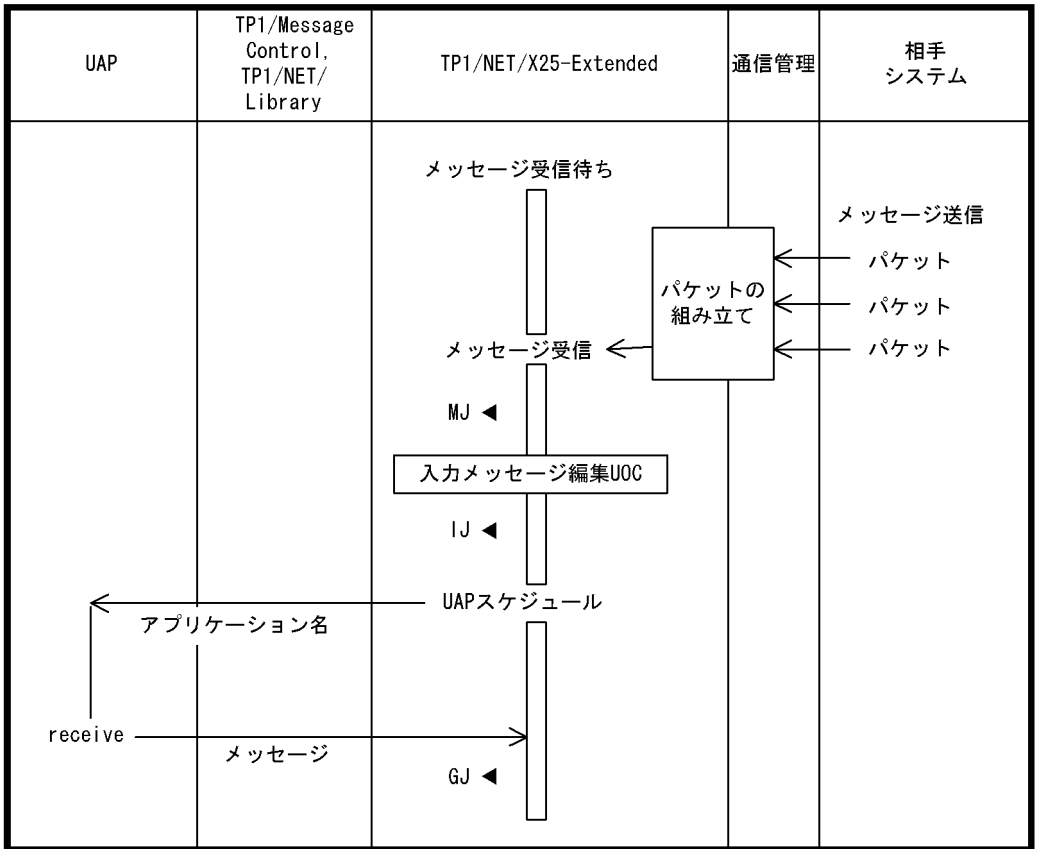
図 A-1 メッセージ受信の処理の流れ（パケット分割なし）



(凡例)

- MJ ◀: メッセージジャーナル取得
- IJ ◀: メッセージ入力ジャーナル取得
- GJ ◀: メッセージ受信ジャーナル取得

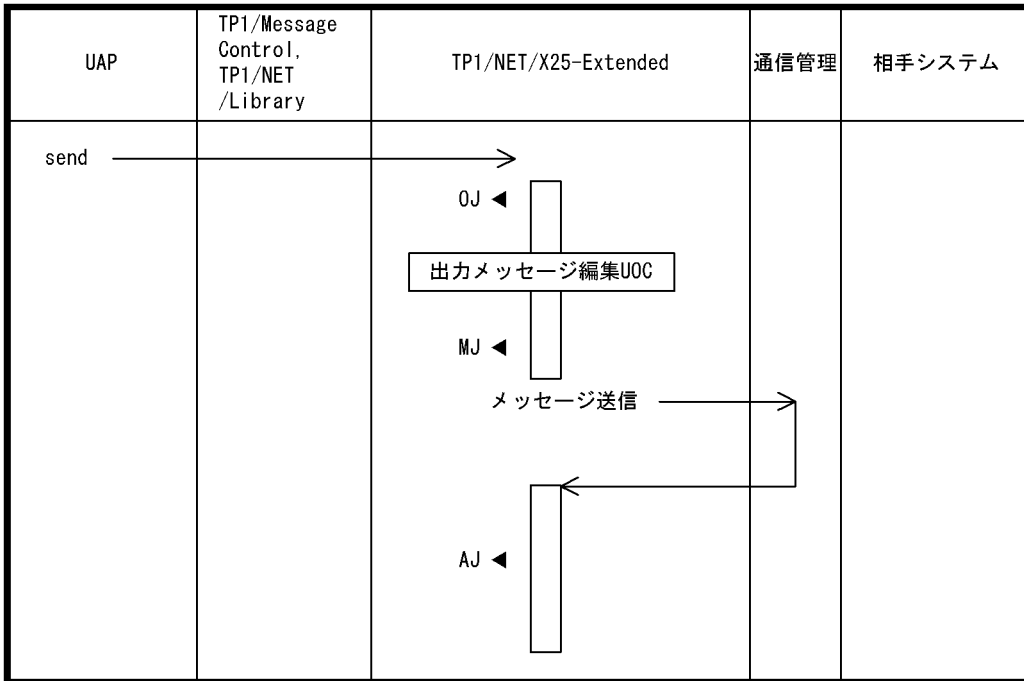
図 A-2 メッセージ受信の処理の流れ (パケット分割あり)



(凡例)

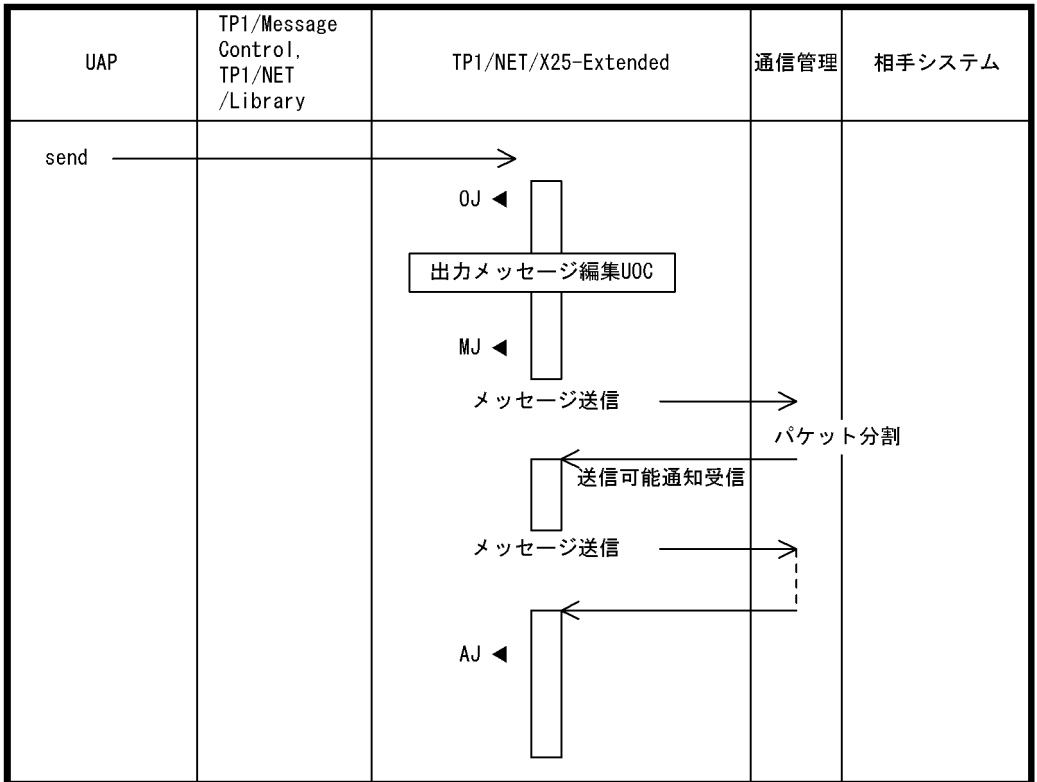
- MJ ◀ : メッセージジャーナル取得
- IJ ◀ : メッセージ入力ジャーナル取得
- GJ ◀ : メッセージ受信ジャーナル取得

図 A-3 メッセージ送信の処理の流れ (パケット分割なし)



- (凡例)
- OJ ◀ : メッセージ出力ジャーナル取得
 - MJ ◀ : メッセージジャーナル取得
 - AJ ◀ : メッセージ送信完了ジャーナル取得

図 A-4 メッセージ送信の処理の流れ (パケット分割あり)



(凡例)

- 0J ◀: メッセージ出力ジャーナル取得
- MJ ◀: メッセージジャーナル取得
- AJ ◀: メッセージ送信完了ジャーナル取得

付録 B 障害発生時の処理の流れ

障害発生時の処理の流れを図 B-1 から図 B-5 に示します。

図 B-1 コネクション確立時の障害

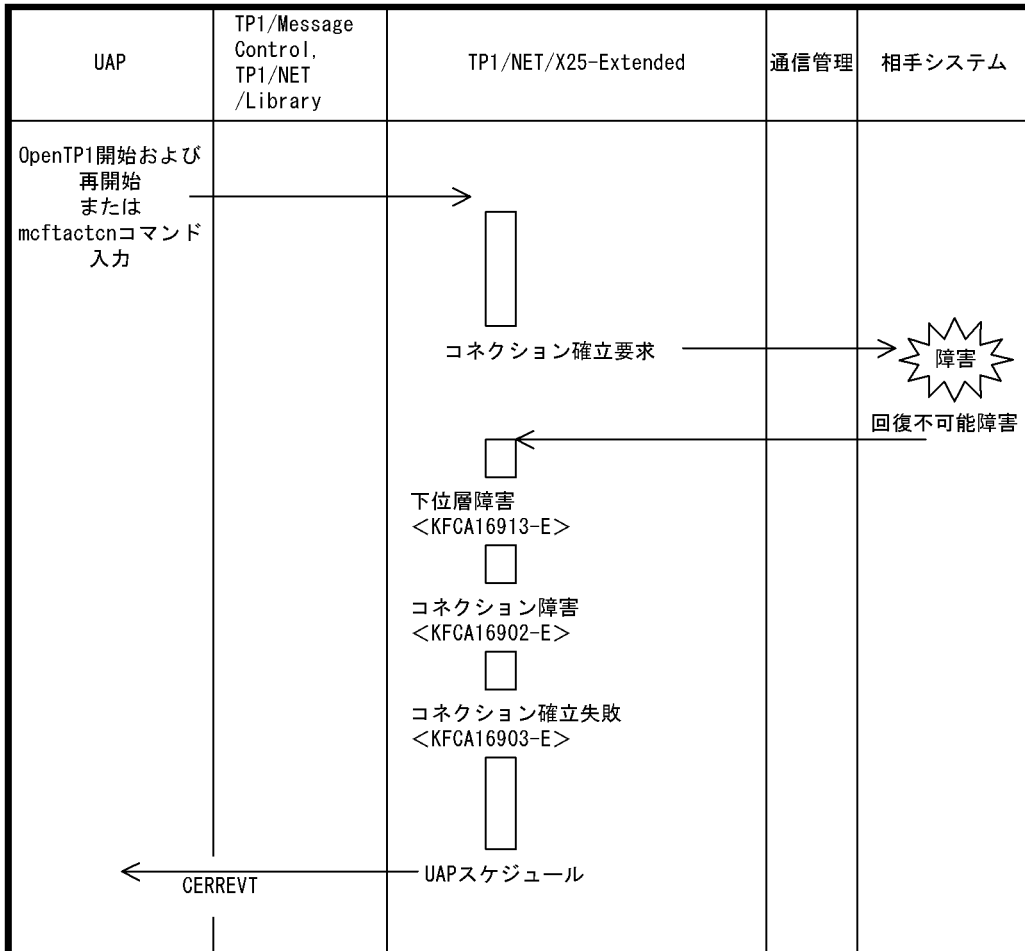
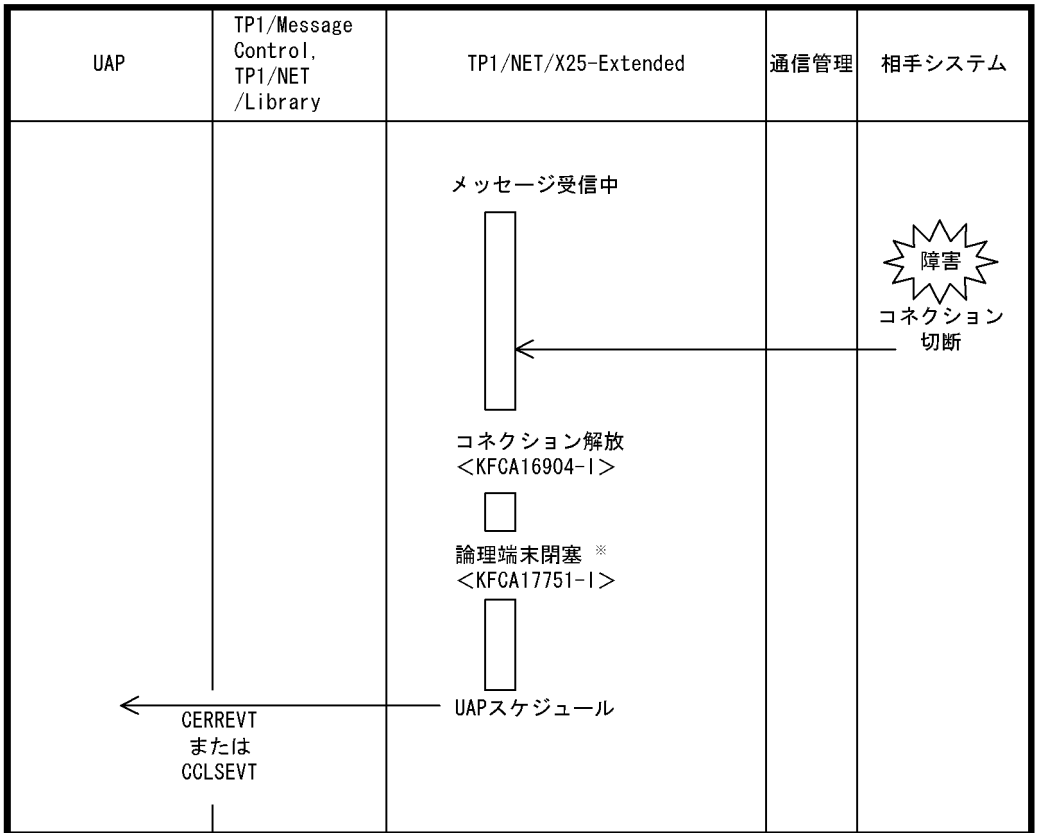


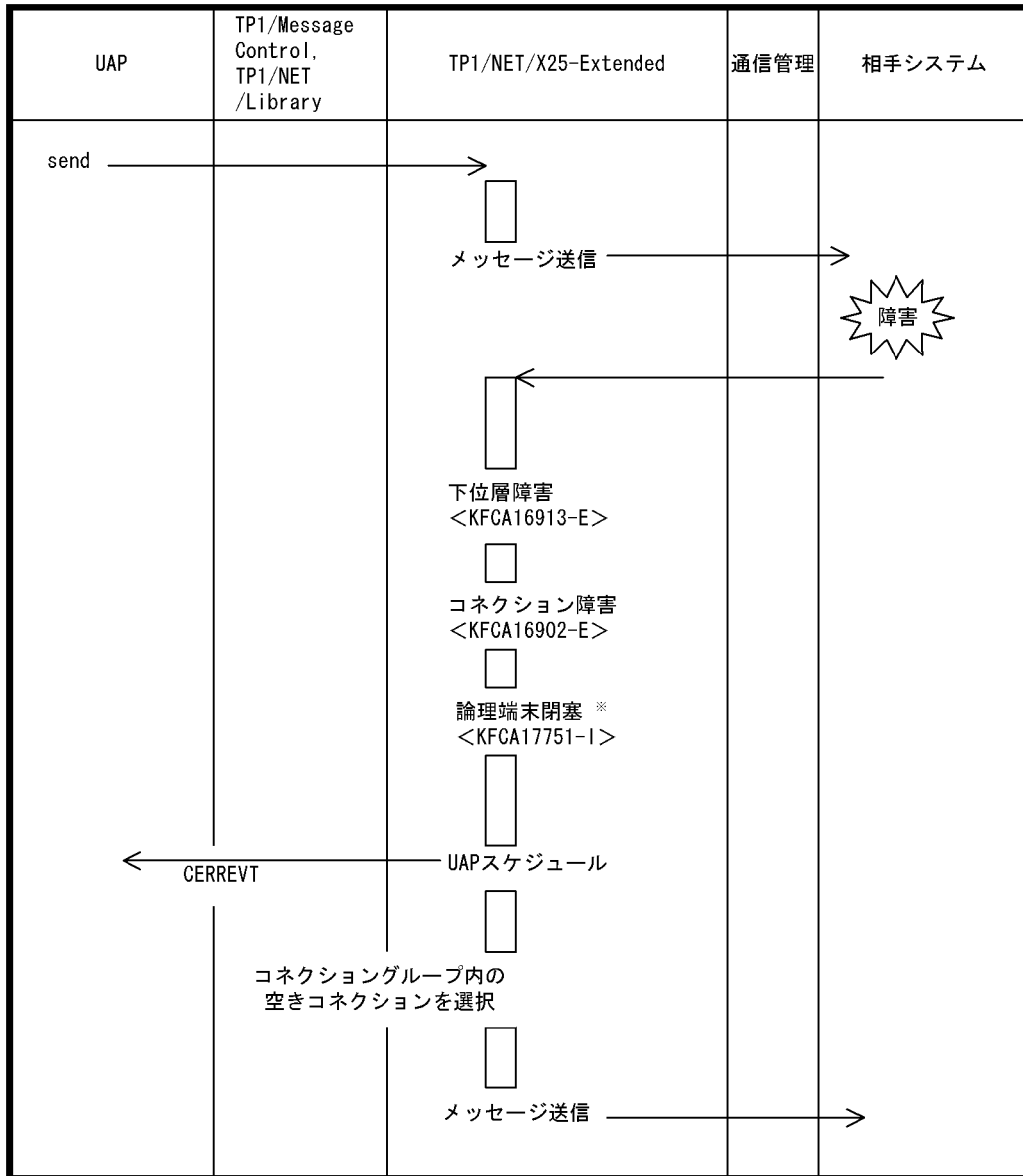
図 B-2 メッセージ受信時のコネクション切断



注

コネクショングループ内のコネクションが一つも使用できないときだけ閉塞されます。

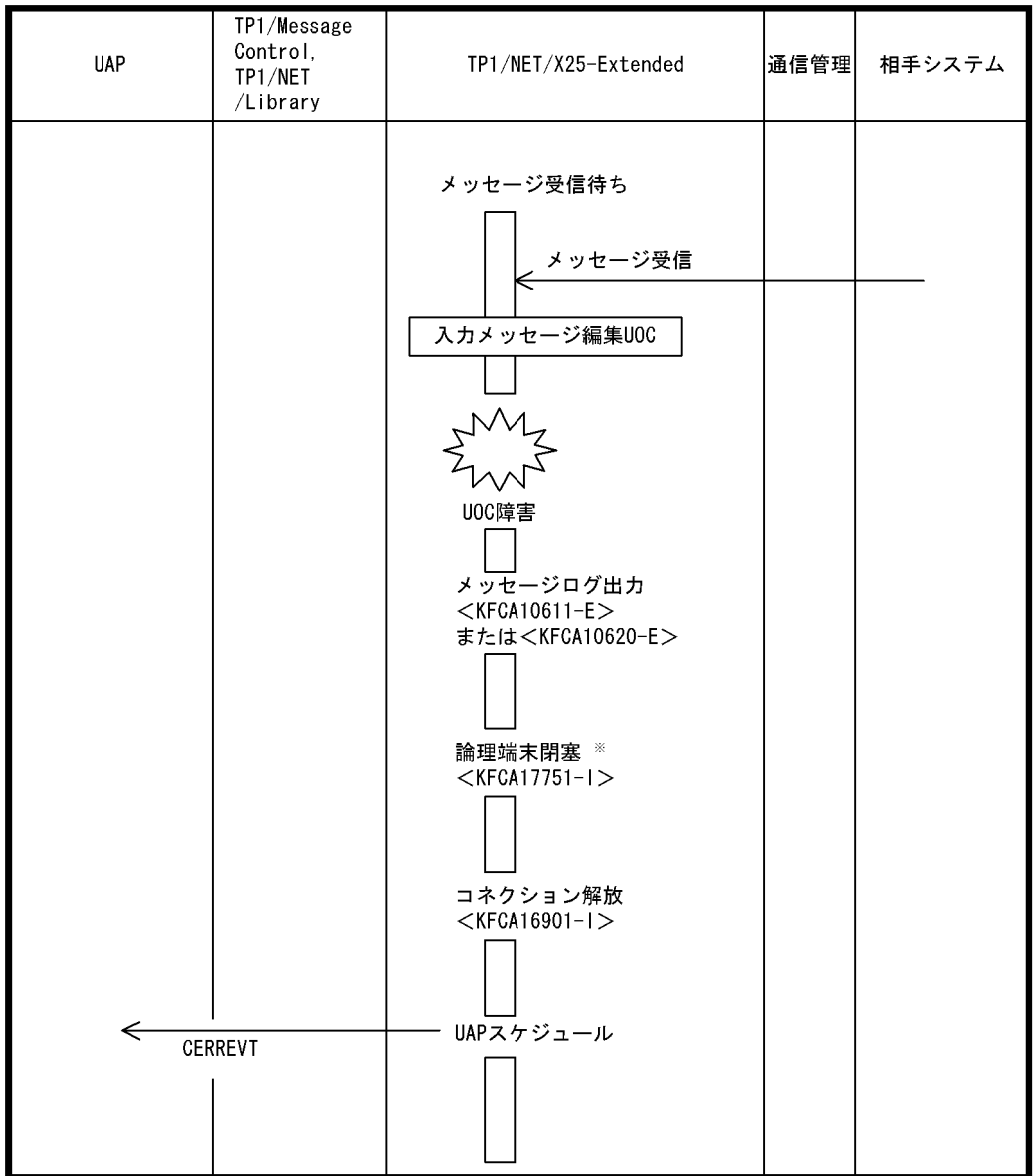
図 B-3 メッセージ送信時の障害コネクションの迂回



注

コネクショングループ内のコネクションが一つも使用できないときだけ閉塞されます。

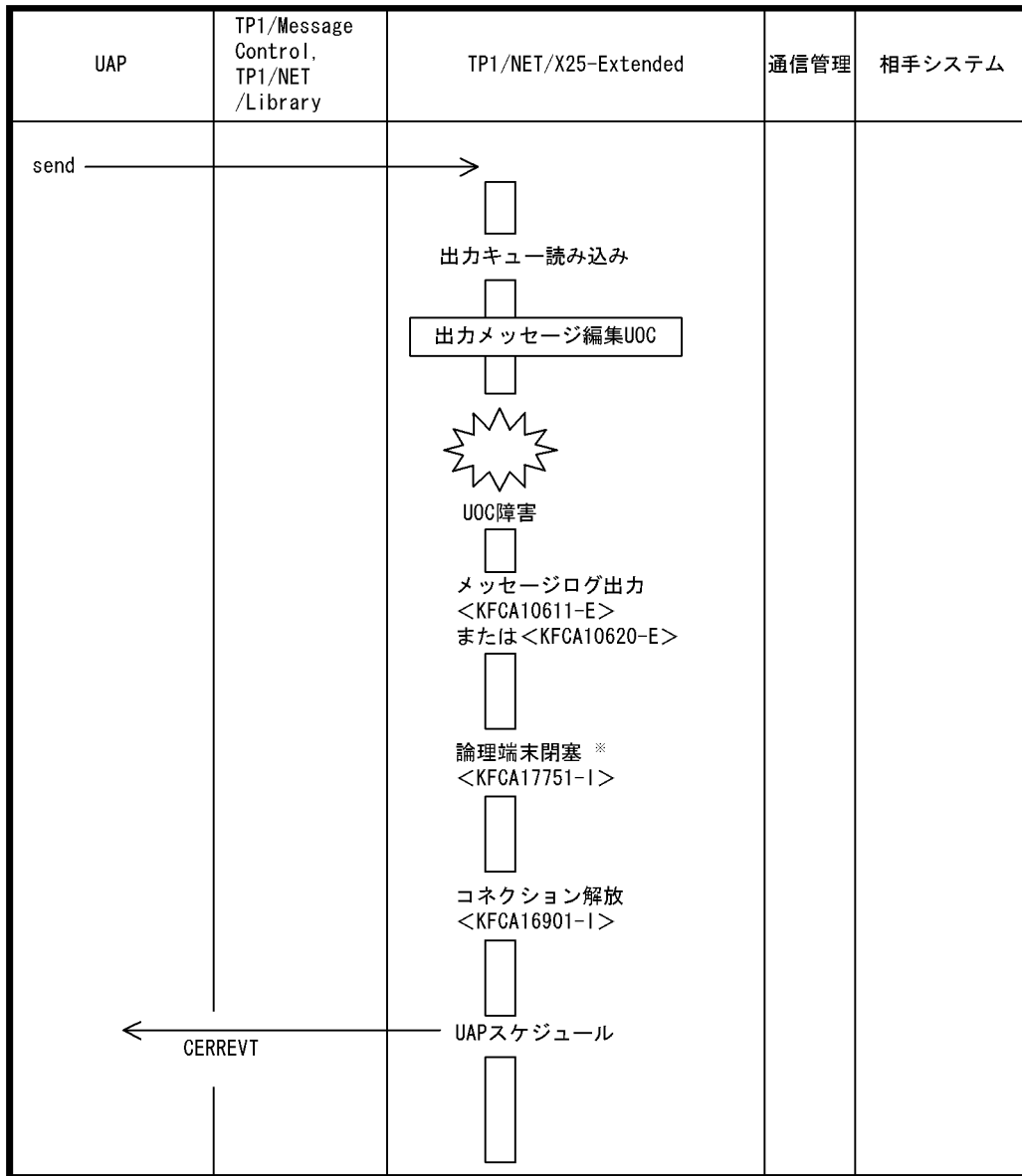
図 B-4 入力メッセージ編集 UOC 障害



注

コネクショングループ内のコネクションが一つも使用できないときだけ閉塞されます。

図 B-5 出力メッセージ編集 UOC 障害



注

コネクショングループ内のコネクションが一つも使用できないときだけ閉塞されます。

付録 C 理由コード一覧

ERREVT2 発生時の理由コードを表 C-1 に、CERREVT 発生時の理由コードを表 C-2 に示します。

表 C-1 ERREVT2 の理由コード

C 言語の理由コード (16 進数)	COBOL 言語の理由コード (外部 10 進)	ERREVT2 の通知理由
DCMCF_NO_SERV (0010)	0010	アプリケーション名に相当する MHP のサービスがありません。
DCMCF_SCD_ERR (0020)	0020	RPC 障害、サーバ未起動などによって MHP または SPP の起動に失敗しました。
DCMCF_QUE_BUF_ERR (0030)	0030	メモリ不足のため、入力キューへの書き込みに失敗しました。
DCMCF_QUE_FIL_OVER (0031)	0031	キューファイルが満杯のため、入力キューへの書き込みに失敗しました。
DCMCF_QUE_LIMIT_OVER (0032)	0032	入力メッセージ最大格納数の定義指定値を超えたため、入力キューに書き込みませんでした。
DCMCF_QUE_IO_ERR (0033)	0033	入力キューへの書き込み時に障害が発生しました。
DCMCF_AP_CLOSE (0040)	0040	MHP のアプリケーションが閉塞中です。
DCMCF_AP_SECURE (0041)	0041	MHP のアプリケーションがセキュア状態です。
DCMCF_SERV_CLOSE (0042)	0042	MHP のサービスまたはサービスグループが閉塞中です。
DCMCF_SERV_SECURE (0043)	0043	MHP のサービスグループがセキュア状態です。
DCMCF_ABNORMAL_END (0050)	0050	MHP のセグメント受信関数にセグメントを渡す前に、MHP の異常が発生しました。

表 C-2 CERREVT の理由コード

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件	発生個所
DCMXEM_RSN1_MCF (00000001) MCF 障害	DCMXEM_RSN2_CNN C (00000000)	コネクション確立不可能	コネクション
	DCMXEM_RSN2_CNE R (00000001)	コネクション切断	
	DCMXEM_RSN2_RBF OF (00000002)	受信バッファオーバーフロー	
	DCMXEM_RSN2_OTG ET (00000005)	出力キュー読み込み障害	
	DCMXEM_RSN2_SBF ST (00000006)	送信バッファ不足	
	DCMXEM_RSN2_RBF ST (00000010)	受信バッファ不足	
	DCMXEM_RSN2_RES ET (00000011)	リセットパケット受信	
	DCMXEM_RSN2_BAD AD (00000015)	プロトコルアドレス不正	
	DCMXEM_RSN2_EVT NG (00000016)	MCF イベント起動失敗	
DCMXEM_RSN1_UOC (00000003) UOC 障害	UOC からの詳細リターンコード	UOC エラーリターン	
	DCMXEM_RSN2_BCN T (00000001)	使用バッファ数不正	
	DCMXEM_RSN2_SEG (00000002)	有効セグメント不正	
	DCMXEM_RSN2_BAD R (00000003)	編集バッファアドレス不正	

付録 D 用語解説

(英字)

AJ (メッセージ送信完了ジャーナル)

TP1/NET/X25-Extended で取得する履歴情報の一つです。メッセージの送信完了情報である、送信通番と送信先論理端末名称とで構成されます。

AJ の取得タイミングは相手システムにメッセージを送信し、その送信完了を受信した直後です。

ANSER (Automatic answer Network System for Electrical Request)

NTT によって提供される通信サービスの一つで、金融機関向けに開発された共同利用型音声照会通知システムです。電話器、ファクシミリ機、キャプテン端末が対象端末で、銀行が持つ会計情報や証券情報を照会できます。

GJ (メッセージ受信ジャーナル)

TP1/NET/X25-Extended で取得する履歴情報の一つです。論理端末名称、入力メッセージなどの情報で構成されます。

GJ の取得タイミングは、receive を発行して、入力キューから取り出したメッセージを UAP に渡す直前です。

IJ (メッセージ入力ジャーナル)

入力キューに入力された情報です。論理端末名称、メッセージ種別、入力メッセージなどの情報で構成されます。

IJ の取得タイミングは相手システムから受信したメッセージを入力キューに入力する直前です。

MJ (メッセージジャーナル)

TP1/NET/X25-Extended で取得する情報の一つです。論理端末名称、メッセージ種別、入力メッセージ編集前のデータ、出力メッセージ編集後のデータなどの情報で構成されます。

MJ の取得タイミングはメッセージ送信前、出力メッセージ編集 UOC 処理後、および入力メッセージ編集 UOC 処理前です。

OJ (メッセージ出力ジャーナル)

出力キューに入力された情報です。メッセージの送信通番、論理端末名称、メッセージ種別、出力メッセージ、セグメント種別などの情報で構成されます。

OJ の取得タイミングは UAP から send を受け付けたときです。

(ア行)

迂回

コネクション障害が発生したとき、同じコネクショングループ内のほかの正常なコネクションを使用して、再び通信を試みる機能です。

(カ行)

コネクション

通信相手システムとの論理的な通信路です。MCFと通信管理との通信接点であり、MCFはコネクションを通して相手システムとメッセージを送受信します。

コネクショングループ

同じ目的で使用されるコネクションの集まりです。コネクショングループは、論理端末と1対1に対応します。UAPは論理端末(コネクショングループ)を通してメッセージを送受信するため、一つのコネクションが使用中または障害中の場合でも、特に意識することなく別のコネクションを使用してメッセージを送受信できます。

(タ行)

着呼コネクション

相手システムからの要求に従って確立されるコネクションです。

(ハ行)

発呼コネクション

自システムから相手システムに対して確立するコネクションです。

(ラ行)

ラウンドロビン方式

実行順序を決めるスケジューリング方式の一つです。待ち行列の最後尾の次が先頭になる円状にして、常に一定方向で実行させることによって対象を平等な優先順位で扱います。

論理端末 (LE)

通信相手を示す論理的概念です。MCFとUAPとの通信接点であり、UAPは論理端末を通して相手システムとメッセージを送受信します。

索引

A

AJ 167
aj 108
ANSER 167
ANSER センタ 2
any 16
aplname 71,72
AP 間通信 2
AP 間通信の概要 2
AP 間通信の形態 3
AP 間通信の仕組み 8
AP 間通信メッセージの送受信 17

B

bretry 105
bretrycnt 105
bretryint 105
buf_adr 71
buf_array 71
buf_num 71
buf_size 71
buff_addr 71
buff_id 71
buflist_adr 71,75

C

CBLDCMCF ('RECEIVE') 34
CBLDCMCF ('RESEND') 40
CBLDCMCF ('SEND') 46
CCLSEVT 82
CCLSEVT (COBOL 言語) 96
CCLSEVT (C 言語) 88
CERREVT 82
CERREVT (COBOL 言語) 95
CERREVT (C 言語) 87
CERREVT の理由コード 166
COBOL 言語を使用したメッセージ送受信
20
COPNEVT 82

COPNEVT(COBOL 言語) 96
COPNEVT (C 言語) 88
C 言語を使用したメッセージ送受信 20

D

dataleng 80
dc_mcf_receive 22
dc_mcf_resend 26
dc_mcf_send 30
dc_mcf_svstart 143
dcmcf_uoc_min_n 70
dcmcf_uoc_mout_n 75
dcmcf_uocbuff_list_n 70,75
dcmcf_uocbufinf_n 70,75
dmsgcnt 107

E

ebuflist_adr 71,75
ERREVT1 81
ERREVT1 (COBOL 言語) 88
ERREVT1 (C 言語) 84
ERREVT2 81
ERREVT2 (COBOL 言語) 89
ERREVT2 (C 言語) 85
ERREVT2 の理由コード 165
ERREVT3 82
ERREVT3 (COBOL 言語) 91
ERREVT3 (C 言語) 85
ERREVT4 82
ERREVT4 82
ERREVT4 (COBOL 言語) 92
ERREVT4 (C 言語) 86

F

flags 78,79

G

GJ 167

I

IJ 167

K

kind 106

L

LE 14, 168

le_name 71, 75

M

mcftactcn 125

mcftactle 128

mcftalcen 103

mcftalcle 107

mcftdctcn 130

mcftdctle 133

mcftgrpccn 110

mcftgrpccle 111

mcftlscn 135

mcftlsle 138

MCF アプリケーション定義 98

MCF イベント 81

MCF イベント一覧 81

MCF イベントインタフェース 81

MCF イベント情報 82

MCF イベントの共通ヘッダ 83

MCF イベントの種類 81

MCF 通信構成定義 98

MCF で使用する定義ファイル 98

MCF マネージャ定義 98

MCF メイン関数のコーディング概要 (ANSI C, C++ の場合) 143

MCF メイン関数のコーディング概要 (K&R 版 C の場合) 144

MCF メイン関数の作成 143

MCF メイン関数のディレクトリへの組み込み方法 145

MHP 6

MJ 167

mmsgcnt 107

module 112

msg_type 76

msgbuf 104

N

noans 16

O

OJ 167

OpenTP1 プロセスサービス 143

OpenTP1 への組み込み方法 (出力メッセージ編集 UOC) 74

OpenTP1 への組み込み方法 (送信メッセージの通番編集 UOC) 77

OpenTP1 への組み込み方法 (入力メッセージ編集 UOC) 69

output_no 76

outputno_flag 76

P

pro_indv_ifa 71, 76

pro_kind 70, 75

Q

quegrpid 108

quekind 108

R

rev_prim 71

revbuf 103

RECEIVE 52

resend_flag 76

rtn_detail 72, 76

S

SCMPEVT 82

SCMPEVT (COBOL 言語) 94

SCMPEVT (C 言語) 86

seg_size 71, 72

SEND 55

senddata 80
 sendid 79
 sendno 79
 sndbuf 103
 snpa 105
 SPP 6

T

termname 79
 TP1/NET/X25-Extended 固有のシステム定義の種類 100
 TP1/NET/X25-Extended に固有の定義の一覧 100
 TP1/NET/X25-Extended の運用コマンド 124
 TP1/NET/X25-Extended の組み込み方法の流れ 142
 TP1/NET/X25-Extended のシステム構成例 118
 TP1/NET/X25-Extended の実行形式プログラム名 112
 TP1/NET/X25-Extended の定義の概要 98
 TP1/NET/X25-Extended を組み込んだソフトウェア構成の例 6
 TP1/NET/X25-Extended を使用したネットワーク構成の例 2

U

UAP 3
 UAP 異常終了通知イベント 82
 UAP 作成例の処理の流れ 59
 uoc_addr 78
 UOC インタフェース用のパラメタとバッファの関係 73
 UOC 作成上の注意事項 80
 UOC で使用できる関数 80
 UOC の異常処理 80
 UOC の構造 80
 UOC の実行タイミング 80
 used_buf_num 72

V

vslot 104

X

X.25 4
 XNF/S-E2 構成定義 117

あ

相手 SNPA アドレス 105
 相手システムからのコネクション解放の流れ 13
 相手システムからのコネクションの解放 12
 相手システムからの再確立 14
 アプリケーションの型 15
 アプリケーション名 108
 アプリケーション名の決定 68
 アプリケーション名の決定の流れ 68

い

一方受信形態 3
 一方受信形態の AP 間通信の例 3
 一方送信メッセージ 3
 一方送信メッセージの再送 (COBOL 言語) 40
 一方送信メッセージの再送 (C 言語) 26
 一方送信メッセージの受信 17
 一方送信メッセージの送信 18
 一方送信メッセージの送信 (COBOL 言語) 46
 一方送信メッセージの送信 (C 言語) 30
 一方送信メッセージの送信 (データ操作言語) 55

う

迂回 167
 運用コマンド 123

か

解放 11
 概要 1

確立 9
仮想スロット番号 104

き

機能 7
キューグループ ID 108
共通定義 98

く

組み込み方法 141

こ

コーディング例 (COBOL 言語) 60
コーディング例 (C 言語) 59
コーディング例 (TP1/NET/X25-Extended ,
共通定義) 120
コーディング例 (TP1/NET/X25-Extended ,
プロトコル固有定義) 121
コーディング例 (XNF/AS 構成定義) 118
コーディング例 (データ操作言語) 62
コネクション 8, 168
コネクション, コネクショングループ, およ
び論理端末の関係 8
コネクション ID 103
コネクションおよびコネクショングループの
解放 11
コネクションおよびコネクショングループの
確立 9
コネクション確立時の障害 160
コネクション確立障害時の確立再試行回数
105
コネクション確立障害時の確立再試行間隔
105
コネクション確立の流れ (着呼) 11
コネクション確立の流れ (発呼) 10
コネクショングループ 8, 168
コネクショングループ定義の開始 110
コネクショングループ定義の終了 111
コネクショングループと論理端末 14
コネクショングループ名 110
コネクション障害後の再確立 14
コネクション切断の流れ 14

コネクション定義 103
コネクションとコネクショングループ 8
コネクションの解放 130
コネクションの確立 125
コネクションの確立を再試行するかどうか
105
コネクションの種類 105
コネクションの状態表示 135
コネクションの切断 13

さ

再確立 14

し

自システムからのコネクション解放の流れ
12
自システムからのコネクションの解放 11
自システムからの再確立 14
自システムの通信管理プログラムと関係づけ
る内容 117
システムサービス共通情報定義 113
システムサービス情報定義 112
システム定義 97
自動的にコネクションを確立するかどうか
104
出力メッセージの編集 73
出力メッセージの割り当て先 108
出力メッセージ編集 UOC 73
出力メッセージ編集 UOC 障害 164
障害コネクションの迂回 15
障害対策 149
障害通知イベント 82
障害の種類と対応処理 150
障害発生時の処理の流れ 160
状態通知イベント 82

す

スタート関数 143

せ

セグメント 16

切断 13
前提プログラム 5

そ

送信完了通知イベント 82
送信メッセージの通番編集 77
送信メッセージの通番編集 UOC 77
ソフトウェア構成の例 5
ソフトウェアの構成 5

た

タイマ起動メッセージ廃棄通知イベント 82
端末タイプ 16, 107

ち

着呼コネクション 9, 168
着呼コネクションの確立 10

つ

通信形態 3

て

定義オブジェクトファイルの作成 146
定義オブジェクトファイルの作成方法の概要 147
定義の指定順序 101
定義例 118
ディスク出力メッセージ最大格納数 107
データ操作言語 (COBOL 言語) を使用したメッセージ送受信 21
適用範囲 4

に

入力メッセージの編集 66
入力メッセージの編集とアプリケーション名の決定 66
入力メッセージ編集 UOC 障害 163
任意型論理端末 16

は

発呼コネクション 9, 168
発呼コネクションの確立 9
バッファ形式 1 16
バッファ形式 2 16
バッファ情報 70
バッファリスト 70

ひ

非応答型 16

ふ

不正アプリケーション名検出通知イベント 81
プロトコル固有定義 98
プロトコルの種別 103
分岐送信形態 3
分岐送信形態の AP 間通信の例 4

へ

閉塞 15
閉塞解除 15

み

未処理送信メッセージ廃棄通知イベント 82

め

メッセージジャーナル 167
メッセージ受信時のコネクション切断 161
メッセージ受信ジャーナル 167
メッセージ受信の処理の流れ (パケット分割あり) 157
メッセージ受信の処理の流れ (パケット分割なし) 156
メッセージ受信の流れ 17
メッセージ受信用バッファグループ番号 103
メッセージ出力ジャーナル 167
メッセージ送受信インタフェース 19
メッセージ送受信インタフェースの一覧 20
メッセージ送受信に使用する通信文 21

メッセージ送受信に使用するプログラム 20
 メッセージ送受信に使用するライブラリ関数
 20
 メッセージ送受信の処理の流れ 156
 メッセージ送信完了ジャーナル 167
 メッセージ送信時のコネクション選択 15
 メッセージ送信時の障害コネクションの迂回
 162
 メッセージ送信の処理の流れ（パケット分割
 あり）159
 メッセージ送信の処理の流れ（パケット分割
 なし）158
 メッセージ送信の流れ 18
 メッセージ送信用バッファグループ番号 103
 メッセージとセグメントとの関係 16
 メッセージ入力ジャーナル 167
 メッセージの受信（COBOL 言語）34
 メッセージの受信（C 言語）22
 メッセージの受信（データ操作言語）52
 メッセージ廃棄通知イベント 81
 メッセージ編集用バッファグループ番号 104
 メモリ出力メッセージ最大格納数 107

ゆ

ユーザアプリケーションプログラム 3
 ユーザアプリケーションプログラムの作成例
 59
 ユーザオウンコーディング，MCF イベント
 インタフェース 65
 ユーザオウンコーディングインタフェース
 66

よ

用語解説 167

ら

ラウンドロビン方式 15, 168

り

理由コード一覧 165
 リンク定義文 117

ろ

論理端末 14, 168
 論理端末定義 107
 論理端末とアプリケーションの型との関係
 15
 論理端末の状態表示 138
 論理端末の端末タイプ，メッセージの種類，
 アプリケーションの型，通信形態，および
 UAP インタフェースの関係 16
 論理端末の閉塞 15, 133
 論理端末の閉塞解除 15, 128
 論理端末名称 107

ソフトウェアマニュアルのサービス ご案内

ソフトウェアマニュアルについて、3種類のサービスをご案内します。ご活用ください。

1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しております。

URL <http://www.hitachi.co.jp/soft/manual/>

ホームページのメニューは次のとおりです。

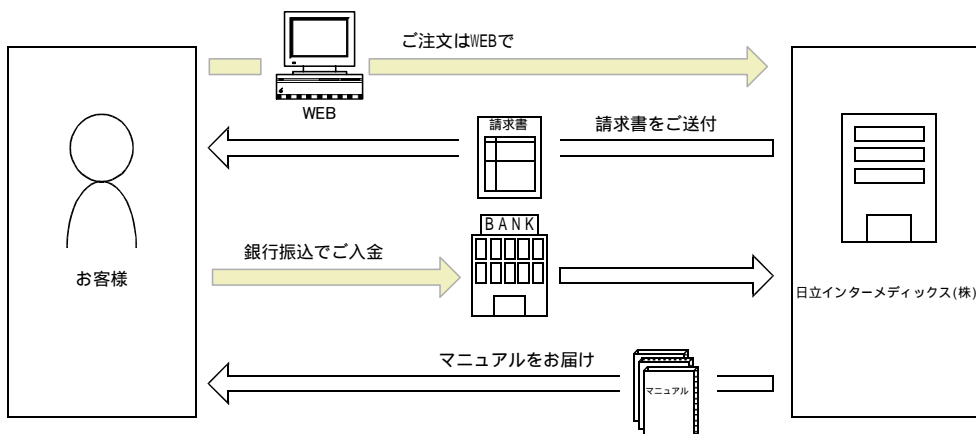
マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル情報	複数マニュアルを格納したCD-ROMマニュアルを提供しています。どの製品に対応したCD-ROMマニュアルがあるか、を参照できます。
マニュアルのご購入	日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からお申し込みできます。 (詳細は「3. マニュアルのご注文」を参照してください。)
Web提供マニュアル一覧	インターネットで参照できるマニュアルの一覧を提供しています。 (詳細は「2. インターネットからのマニュアル参照」を参照してください。)
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

2. インターネットからのマニュアル参照(ソフトウェアサポートサービス)

ソフトウェアサポートサービスの契約をしていただくと、インターネットでマニュアルを参照できます。本サービスの対象となる契約の種別、及び参照できるマニュアルは、マニュアル情報ホームページでご確認ください。なお、ソフトウェアサポートサービスは、マニュアル参照だけでなく、対象製品に対するご質問への回答、問題解決支援、バージョン更新版の提供など、お客様のシステムの安定的な稼働のためのサービスをご提供しています。まだご契約いただいていない場合は、ぜひご契約いただくことをお勧めします。

3. マニュアルのご注文

日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からご注文ください。



下記 URL にアクセスして必要事項を入力してください。

URL http://www2.himdx.net/manual/privacy.asp?purchase_flag=1

ご注文いただいたマニュアルについて、請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。なお、送料は弊社で負担します。

入金確認後、7日以内にお届けします。在庫切れの場合は、納期を別途ご案内いたします。