

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 編

解説・手引・文法・操作書

3000-3-D76

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。

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-F1M64-32313 uCosminexus TP1/NET/X25 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。

発行

2009 年 1 月 (第 1 版) 3000-3-D76

著作権

All Rights Reserved. Copyright (C) 2009, Hitachi, Ltd.

はじめに

このマニュアルは、TP1/NET/X25 の概要、機能、操作、および運用について説明したものです。

このマニュアルに記載するプログラムプロダクトは次のとおりです。

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

- P-1M64-3131 uCosminexus TP1/Message Control
- P-1M64-3231 uCosminexus TP1/NET/Library
- P-F1M64-32313 uCosminexus TP1/NET/X25

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 システムの通信に X.25 プロトコルを使用するシステム管理者およびシステム設計者を対象としています。また、オンラインや OpenTP1 システムの基礎的な知識を持っていて、次に示す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 を使用したシステム間の通信 (AP 間通信) の概要について説明しています。

第 2 章 PVC の機能

PVC の場合のコネクションの確立方法などの AP 間通信の仕組み、およびメッセージ送受信の流れについて説明しています。

第 3 章 VC の機能

VC の場合のコネクションの確立方法などの AP 間通信の仕組み、およびメッセージ送受信の流れについて説明しています。

第 4 章 メッセージ送受信インタフェース

TP1/NET/X25 を使用してメッセージを送受信する UAP の作成方法、および作成例について説明しています。

はじめに

第 5 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コーディングのインタフェース，および MCF イベントのインタフェースについて説明しています。

第 6 章 システム定義

X.25 プロトコルを使用するために必要な，OpenTP1 システム定義の中での TP1/NET/X25 固有の定義について説明しています。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ています。

第 7 章 運用コマンド

TP1/NET/X25 で使用する運用コマンドについて説明しています。

第 8 章 組み込み方法

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

第 9 章 PVC の障害対策

PVC を使用した場合の障害時の処理とユーザの対策，および障害発生時の処理の流れについて説明しています。

第 10 章 VC の障害対策

VC を使用した場合の障害時の処理とユーザの対策，および障害発生時の処理の流れ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

一方送信メッセージを送受信するときのデータの流れ，および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

障害が発生したときの処理の流れを PVC と VC に分けて説明しています。

付録 C 理由コード一覧

ERREVT2 通知時の理由コード，および障害通知イベント（CERREVT）通知時の理由コードについて説明しています。

付録 D 旧製品からの移行に関する注意事項

バージョン 6 以前からバージョン 7 に移行する際の注意事項について説明しています。

付録 E バージョンアップ時の変更点

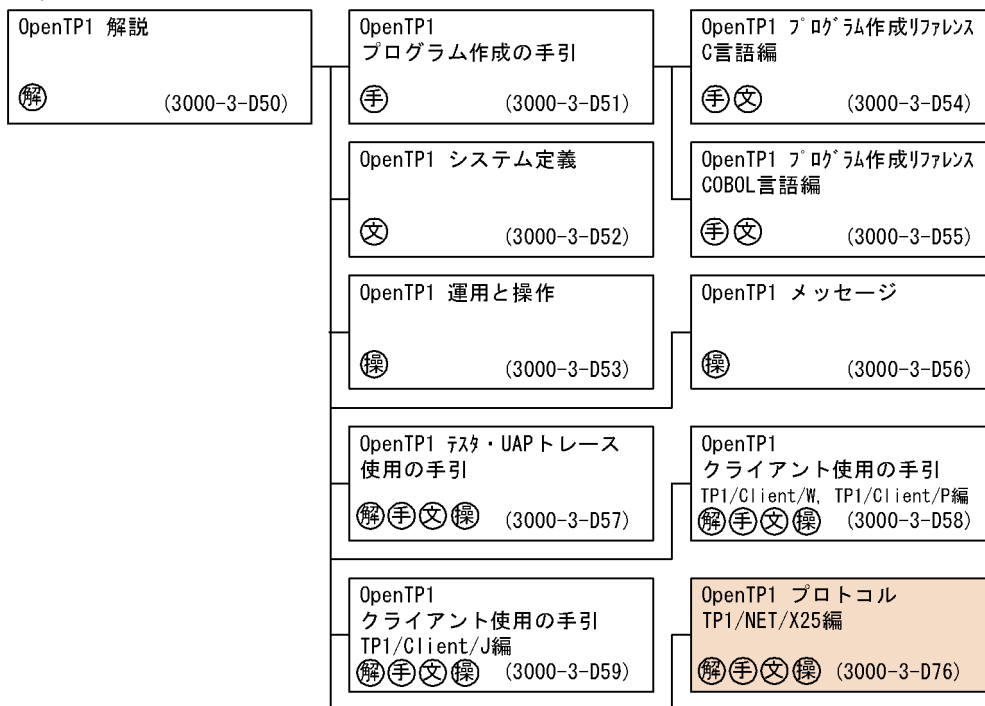
各バージョンでの関数，定義およびコマンドの変更点について説明しています。

付録 F 用語解説

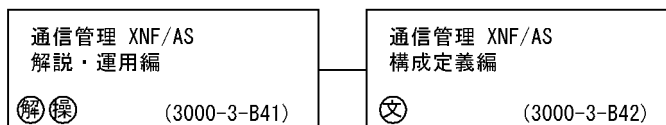
TP1/NET/X25 で使用する用語について説明しています。

関連マニュアル

●OpenTP1 Version 7



●通信管理



<記号>

- ④ : 解説書
- ⑤ : 手引書
- ⑥ : 文法書
- ⑦ : 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)

- ▼ : PVCを使用する場合にお読みください。
- ▽ : VCを使用する場合にお読みください。

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, ●論理端末
端末



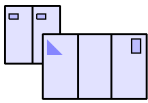
●論理回線



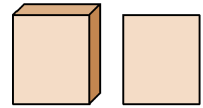
●入出力の動作



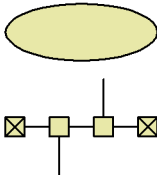
●ホストコンピュータ ●プログラムの流れ ●ファイル



●プログラム



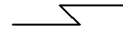
●ネットワーク



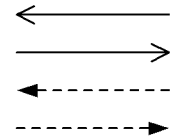
●データの流れ



●通信回線



●制御の流れ



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
(5e) ₁₆	'¥' (円記号)	'\' '(バックスラッシュ)
(7e) ₁₆	' ' (オーバーライン)	'~ '(チルド)

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

文法の記述形式について説明する記号です。

文法記述記号	意味
{ }	この記号で囲まれている項目は省略してもよいことを示します。 (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。

文法記述記号	意味
 (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) -l auto manual -l オプションに auto または manual 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することを示します。 (例) {DCMCFESI DCMCFEMI} DCMCFESI と DCMCFEMI のうち、どちらかを指定することを示します。
<u> </u> (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) -i auto <u>manual</u> -i オプションを省略した場合、manual 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) コネクション ID1 コネクション ID2 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説明する記号です。

属性表示記号	意味
~	この記号のあと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説明する記号です。

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 進数字	数字 (0 ~ 9)
16 進数字	数字 (0 ~ 9) とアルファベット (A ~ F, a ~ f)

構文要素記号	意味
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)

このマニュアルでの表記

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/Message Control/Tester	TP1/Message Control/Tester	
uCosminexus TP1/NET/Library	TP1/NET/Library	
uCosminexus TP1/NET/X25	TP1/NET/X25	
uCosminexus TP1/Server Base	TP1/Server Base	
XNF/AS/BASE	XNF/AS	XNF
XNF/AS/NLI		

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
AP	<u>A</u> pplication <u>P</u> rogram
INS	I <u>n</u> formation <u>N</u> etwork <u>S</u> ystem
ISDN	I <u>n</u> tegrated <u>S</u> ervices <u>D</u> igital <u>N</u> etwork
LAN	L <u>o</u> cal <u>A</u> rea <u>N</u> etwork
LE	L <u>o</u> gical <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
OS	<u>O</u> perating <u>S</u> ystem
PVC	<u>P</u> ermanent <u>V</u> irtual <u>C</u> ircuit
RPC	<u>R</u> emote <u>P</u> rocedure <u>C</u> all
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram

英略語	英字での表記
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、常用漢字以外の漢字を使用しています。

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	4
1.2.1	通信形態	4
1.2.2	適用範囲	5
1.3	ソフトウェアの構成	6
1.3.1	前提プログラム	6
1.3.2	ソフトウェア構成の例	6
2	PVC の機能	9
2.1	PVC の AP 間通信の仕組み	10
2.1.1	コネクションの確立と解放	10
2.1.2	コネクションと論理端末の関係	12
2.1.3	論理端末とアプリケーションの型との関係	12
2.1.4	論理端末の閉塞と閉塞解除	13
2.1.5	メッセージとセグメントの関係	14
2.2	PVC の AP 間通信メッセージの送受信	15
2.2.1	一方送信メッセージの送信	15
2.2.2	一方送信メッセージの受信	15
3	VC の機能	17
3.1	VC の AP 間通信の仕組み	18
3.1.1	コネクションの確立と解放	18
3.1.2	コネクションと論理端末の関係	22
3.1.3	論理端末とアプリケーションの型との関係	23
3.1.4	論理端末の閉塞と閉塞解除	23
3.1.5	メッセージとセグメントの関係	24
3.2	VC の AP 間通信メッセージの送受信	26
3.2.1	一方送信メッセージの送信	26
3.2.2	一方送信メッセージの受信	26

4	メッセージ送受信インタフェース	29
	メッセージ送受信インタフェースの一覧	30
	dc_mcf_receive - 一方送信メッセージの受信 (C 言語)	32
	dc_mcf_resend - メッセージの再送 (C 言語)	38
	dc_mcf_send - 一方送信メッセージの送信 (C 言語)	43
	CBLDCMCF('RECEIVE ') - 一方送信メッセージの受信 (COBOL 言語)	48
	CBLDCMCF('RESEND ') - メッセージの再送 (COBOL 言語)	56
	CBLDCMCF('SEND ') - 一方送信メッセージの送信 (COBOL 言語)	62
	RECEIVE - メッセージの受信 (データ操作言語)	69
	SEND - 一方送信メッセージの送信 (データ操作言語)	73
	ユーザアプリケーションプログラム作成例	77
5	ユーザオウンコーディング, MCF イベントインタフェース	83
5.1	ユーザオウンコーディングインタフェース	84
5.1.1	入力メッセージの編集とアプリケーション名の決定	84
5.1.2	入力メッセージ編集 UOC インタフェース	87
5.1.3	出力メッセージの編集	91
5.1.4	出力メッセージ編集 UOC インタフェース	92
5.1.5	送信メッセージの通番編集	95
5.1.6	送信メッセージの通番編集 UOC インタフェース	97
5.1.7	UOC 作成上の注意事項	98
5.2	MCF イベントインタフェース	100
5.2.1	MCF イベントの種類	100
5.2.2	MCF イベント通知時のセグメント構成	101
5.2.3	MCF イベント情報の形式 (C 言語)	102
5.2.4	MCF イベント情報の形式 (COBOL 言語)	107
6	システム定義	115
	TP1/NET/X25 の定義の概要	116
	TP1/NET/X25 固有のシステム定義の種類	118
	mcftalccn (コネクション定義の開始) - PVC	123
	mcftalccn (コネクション定義の開始) - VC	128
	mcftalced (コネクション定義の終了) - PVC	133

mcftalced (コネクション定義の終了) - VC	134
mcftalcle (論理端末定義) - PVC	135
mcftalcle (論理端末定義) - VC	137
mcftcomn (MCF 通信構成共通定義) - PVC	139
システムサービス情報定義	140
システムサービス共通情報定義	141
MCF 定義オブジェクトの生成	144
自システムの通信管理プログラムと関連づける内容	145
定義例	146

7

運用コマンド	155
--------	-----

TP1/NET/X25 の運用コマンド	156
mcftactcn (コネクションの確立)	157
mcftactle (論理端末の閉塞解除)	159
mcftdctcn (コネクションの解放)	161
mcftdctle (論理端末の閉塞)	163
mcftlscn (コネクションの状態表示)	166
mcftlsle (論理端末の状態表示)	169

8

組み込み方法	173
--------	-----

8.1 TP1/NET/X25 の組み込みの流れ	174
8.2 MCF メイン関数の作成	175
8.3 定義オブジェクトファイルの生成	178

9

PVC の障害対策	181
-----------	-----

9.1 障害の種類と対応処理 (PVC)	182
9.2 コネクション障害 (PVC)	187
9.3 スケジュール関係の障害 (PVC)	190
9.4 UAP 障害 (PVC)	192

10

VC の障害対策	195
----------	-----

10.1 障害の種類と対応処理 (VC)	196
10.2 コネクション障害 (VC)	202

10.3 スケジュール関係の障害 (VC)	203
10.4 UAP 障害 (VC)	205

付録 207

付録 A メッセージ送受信の処理の流れ	208
付録 B 障害発生時の処理の流れ	212
付録 B.1 PVC の障害	212
付録 B.2 VC の障害	215
付録 C 理由コード一覧	220
付録 C.1 ERREVT2 の理由コード	220
付録 C.2 PVC の理由コード	220
付録 C.3 VC の理由コード	221
付録 D 旧製品からの移行に関する注意事項	223
付録 E バージョンアップ時の変更点	226
付録 E.1 07-00 での変更点	226
付録 F 用語解説	227

索引 229

目次		
図 1-1	TP1/NET/X25 を使用したネットワーク構成の例	2
図 1-2	分岐送信形態の AP 間通信の例	4
図 1-3	一方受信形態の AP 間通信の例	4
図 1-4	TP1/NET/X25 を組み込んだソフトウェア構成の例	6
図 2-1	コネクション確立の流れ	10
図 2-2	コネクション解放の流れ	11
図 2-3	コネクション切断時の流れ	12
図 2-4	PVC のメッセージ送信の流れ	15
図 2-5	PVC のメッセージ受信の流れ	16
図 3-1	VC のコネクション確立の流れ (発呼)	19
図 3-2	VC のコネクション確立の流れ (着呼)	20
図 3-3	自システムからのコネクション解放の流れ	21
図 3-4	相手システムからのコネクション解放の流れ	22
図 3-5	VC のメッセージ送信の流れ	26
図 3-6	VC のメッセージ受信の流れ	27
図 4-1	UAP 作成例の処理の流れ	77
図 5-1	アプリケーション名の決定の処理	86
図 5-2	UOC インタフェース用のパラメタとバッファの関係	91
図 5-3	MCF イベント通知時のセグメント構成	101
図 6-1	プロトコル固有定義コマンドの指定順序	122
図 6-2	TP1/NET/X25 のシステム構成例 (PVC の場合)	146
図 6-3	TP1/NET/X25 のシステム構成例 (VC の場合)	150
図 8-1	MCF メイン関数のコーディング概要 (ANSI C, C++ の場合)	175
図 8-2	MCF メイン関数のコーディング概要 (K & R 版 C の場合)	176
図 8-3	MCF メイン関数のディレクトリへの組み込み方法	177
図 8-4	定義オブジェクトファイルの作成方法の概要	179
図 9-1	一方送信メッセージ送信中のコネクション障害 (PVC)	187
図 9-2	一方送信メッセージ送信後のコネクション障害 (PVC)	188
図 9-3	一方送信メッセージ書き込み前のコネクション障害 (PVC)	189
図 9-4	一方送信メッセージ書き込み後のコネクション障害 (PVC)	189
図 9-5	入力キュー障害 (PVC)	190
図 9-6	出力キュー障害 (PVC)	191
図 9-7	メッセージ受信前の UAP 異常終了 (PVC)	192

図 9-8	メッセージ受信後の UAP 異常終了 (PVC)	193
図 10-1	コネクション障害 (VC)	202
図 10-2	入力キュー障害 (VC)	203
図 10-3	出力キュー障害 (VC)	204
図 10-4	メッセージ受信前の UAP 異常終了 (VC)	205
図 10-5	メッセージ受信後の UAP 異常終了 (VC)	206
図 A-1	メッセージ送信の処理の流れ (パケット分割なし)	208
図 A-2	メッセージ送信の処理の流れ (パケット分割あり)	209
図 A-3	メッセージ受信の処理の流れ (パケット分割なし)	210
図 A-4	メッセージ受信の処理の流れ (パケット分割あり)	211
図 B-1	コネクション確立時の障害 (PVC)	212
図 B-2	コネクション切断 (回復可能障害・手動再確立・論理端末自動起動)(PVC)	213
図 B-3	入力メッセージ編集 UOC 障害 (PVC)	214
図 B-4	出力メッセージ編集 UOC 障害 (PVC)	215
図 B-5	コネクション確立時の障害 (VC)	216
図 B-6	コネクション切断 (VC)	217
図 B-7	入力メッセージ編集 UOC 障害 (VC)	218
図 B-8	出力メッセージ編集 UOC 障害 (VC)	219

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 (PVC)	13
表 3-1	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 (VC)	23
表 4-1	メッセージ送受信の関数 (C 言語)	30
表 4-2	メッセージ送受信の関数に対応するプログラム (COBOL 言語)	30
表 4-3	メッセージ送受信の通信文 (データ操作言語)	31
表 5-1	MCF イベント一覧	100
表 5-2	COBOL 言語の MCF イベント情報の内容 (ERREVT1)	107
表 5-3	COBOL 言語の MCF イベント情報の内容 (ERREVT2)	108
表 5-4	COBOL 言語の MCF イベント情報の内容 (ERREVT3)	109
表 5-5	COBOL 言語の MCF イベント情報の内容 (ERREVTA)	111
表 5-6	COBOL 言語の MCF イベント情報の内容 (CERREVT)	112
表 5-7	COBOL 言語の MCF イベント情報の内容 (COPNEVT, CCLSEVT)	114
表 6-1	MCF で使用する定義ファイル	116
表 6-2	TP1/NET/X25 固有の定義の一覧	118
表 7-1	TP1/NET/X25 で使用する運用コマンド	156
表 9-1	コネクション障害と対応処理 (PVC)	182
表 9-2	受信スケジュール関係の障害と対応処理 (PVC)	184
表 9-3	送信スケジュール関係の障害と対応処理 (PVC)	185
表 9-4	UAP 障害と対応処理 (PVC)	186
表 9-5	MCF 障害と対応処理 (PVC)	186
表 10-1	コネクション障害と対応処理 (VC)	196
表 10-2	受信スケジュール関係の障害と対応処理 (VC)	198
表 10-3	送信スケジュール関係の障害と対応処理 (VC)	199
表 10-4	UAP 障害と対応処理 (VC)	200
表 10-5	MCF 障害と対応処理 (VC)	201
表 C-1	ERREVT2 の理由コード	220
表 C-2	コネクション障害の理由コード (PVC)	221
表 C-3	論理端末障害の理由コード (PVC)	221
表 C-4	コネクション障害の理由コード (VC)	222
表 C-5	論理端末障害の理由コード (VC)	222
表 D-1	バージョン 6 以前で使用していたソースファイルの互換性	223

1

概要

TP1/NET/X25 は、OpenTP1 システムを構成するプログラムの一つです。ホストコンピュータ、端末などを X.25 プロトコルによって論理的に接続し、メッセージを送受信します。この章では、TP1/NET/X25 を使用したシステム間の通信（AP 間通信）の概要について説明します。

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

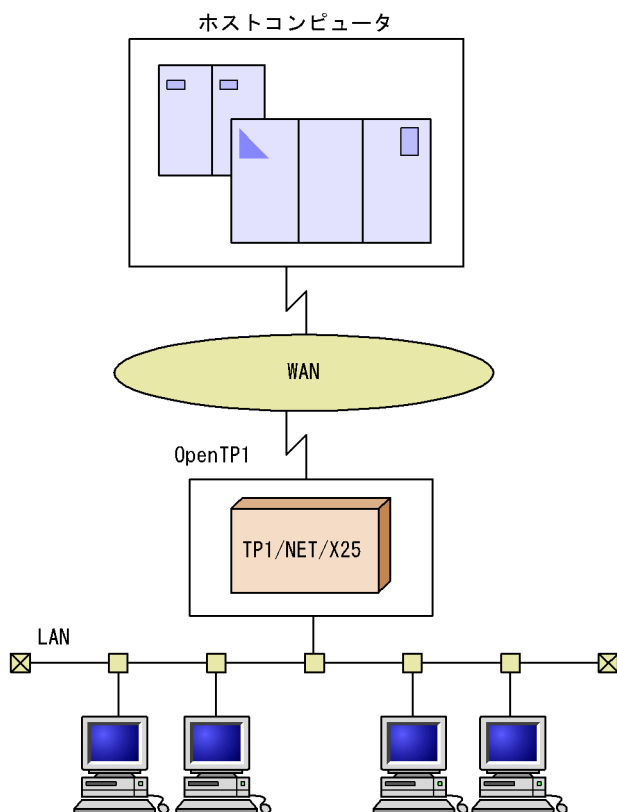
1.1 AP 間通信の概要

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のメッセージの送受信をいいます。

TP1/NET/X25 は、OpenTP1 システムと各種コンピュータシステムを X.25 プロトコルで接続し、AP 間通信をするプログラムです。TP1/NET/X25 を使用した AP 間通信では、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ったりできます。

TP1/NET/X25 を使用したネットワーク構成の例を次の図に示します。

図 1-1 TP1/NET/X25 を使用したネットワーク構成の例



TP1/NET/X25 では、X.25 プロトコルの通信相手の選択方式として、PVC と VC を支援しています。PVC と VC について説明します。

PVC

通信相手を固定して通信する方式です。

VC

通信相手を固定しないで選択して通信する方式です。

このマニュアルでは、PVC と VC で違いがある場合には、分けて説明しているか、どちらの場合に限定して説明しています。特に分けていない場合は、PVC と VC で共通の内容です。

1.2 AP 間通信の形態

1.2.1 通信形態

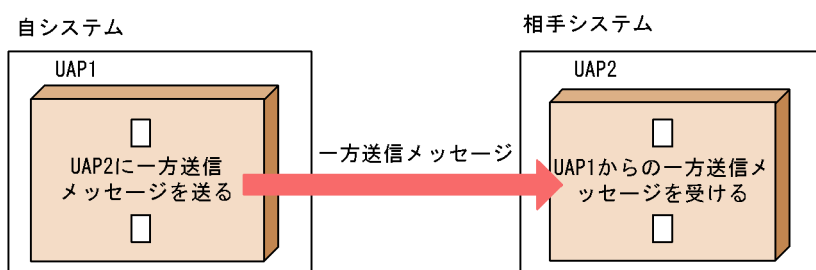
TP1/NET/X25 を使用した AP 間通信の形態には、分岐送信形態と一方受信形態があります。

(1) 分岐送信形態

自システムから別のシステムに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に送信するメッセージを、一方送信メッセージといいます。

分岐送信形態の AP 間通信の例を次の図に示します。

図 1-2 分岐送信形態の AP 間通信の例

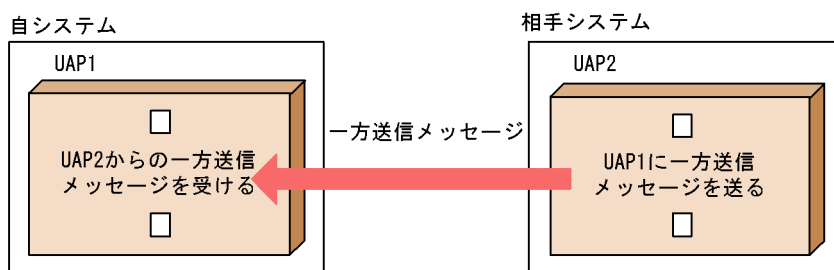


(2) 一方受信形態

相手システム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も、分岐送信形態の場合と同様に一方送信メッセージといいます。

一方受信形態の AP 間通信の例を次の図に示します。

図 1-3 一方受信形態の AP 間通信の例



1.2.2 適用範囲

(1) 伝送動作モード

TP1/NET/X25 で適用する伝送動作モードを次に示します。

- X.25(80)PVC
- X.25(80)VC 発呼
- X.25(80)VC 着呼
- X.25(84)VC 発呼
- X.25(84)VC 着呼

1.3 ソフトウェアの構成

TP1/NET/X25 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 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 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 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 TP1/Server Base 07-00 以降
- P-1M14-511 XNF/AS/BASE
- R-F1M141-518 XNF/AS/NLI

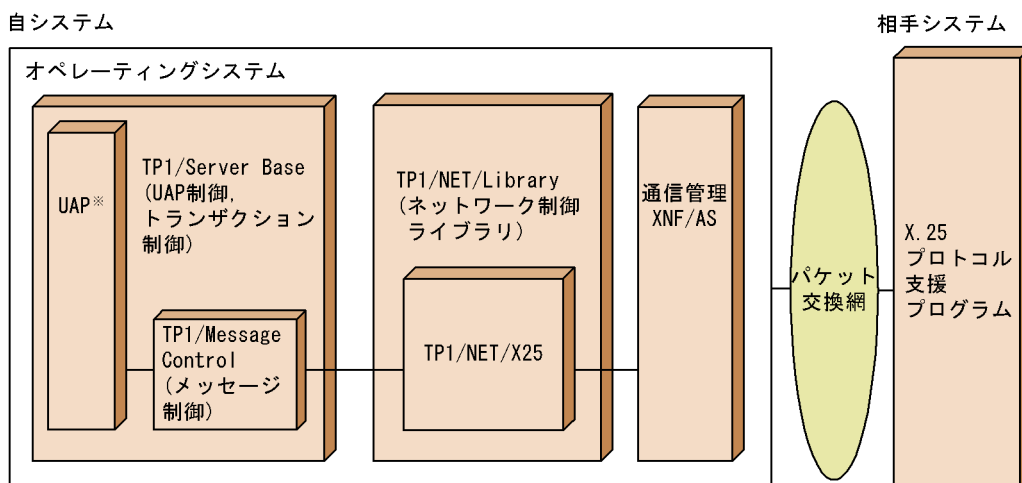
注

前提となる XNF のバージョンは、AIX 5L 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 を組み込んだソフトウェア構成の例を次の図に示します。

図 1-4 TP1/NET/X25 を組み込んだソフトウェア構成の例



注

TP1/NET/X25 で扱う UAP は、MHP および 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。

2

PVC の機能

一般に，AP 間通信をするときには，自システムと相手システムとの間であらかじめ通信上の規約（プロトコル）を決める必要があります。TP1/NET/X25 は，二つのシステムの間にはコネクションという論理的通信路を設定し，メッセージを送受信します。この章では，PVC の場合のコネクションの確立方法などの AP 間通信の仕組み，およびメッセージ送受信の流れについて説明します。

2.1 PVC の AP 間通信の仕組み

2.2 PVC の AP 間通信メッセージの送受信

2.1 PVC の AP 間通信の仕組み

TP1/NET/X25 の PVC を使用する場合は、コネクションや論理端末を使用した AP 間通信の仕組みについて説明します。

2.1.1 コネクションの確立と解放

(1) コネクションの確立

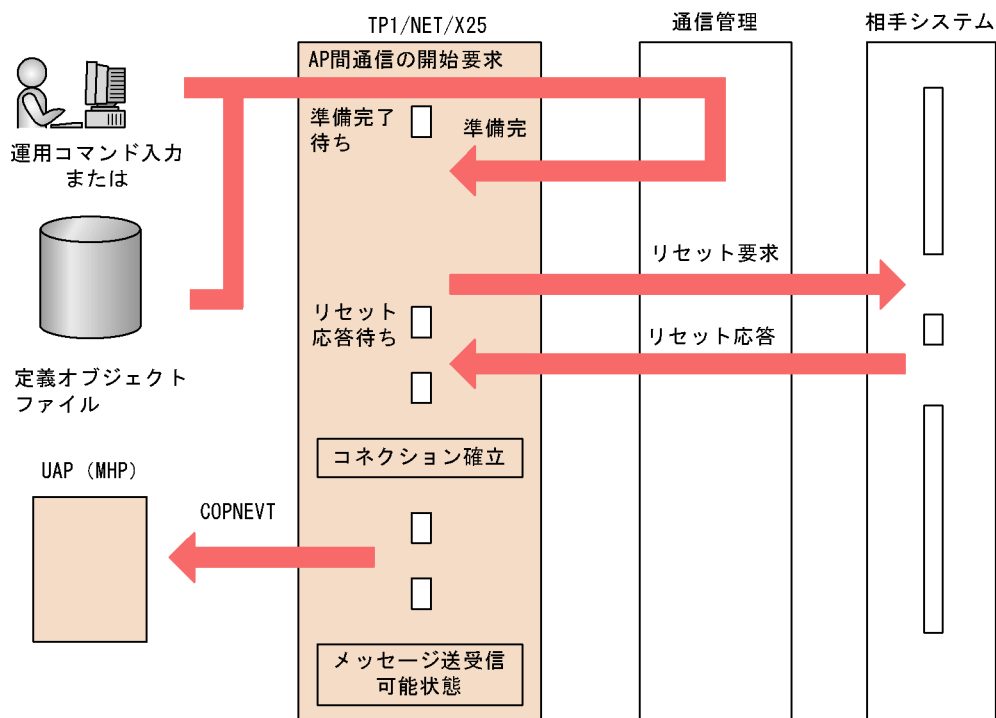
TP1/NET/X25 は、AP 間通信をするために、自システムと相手システムとの間に論理的な通信路（コネクション）を確立します。

オンライン開始時に指定した定義オブジェクトファイルの内容、またはオンライン運用中の運用コマンドの入力によってコネクションを確立し、メッセージの送受信ができる状態になります。

TP1/NET/X25 は、状態通知イベント（COPNEVT）によって、ユーザアプリケーションプログラム（以降、UAP と記述します）にコネクションの確立を通知します。

コネクション確立の流れを次の図に示します。

図 2-1 コネクション確立の流れ

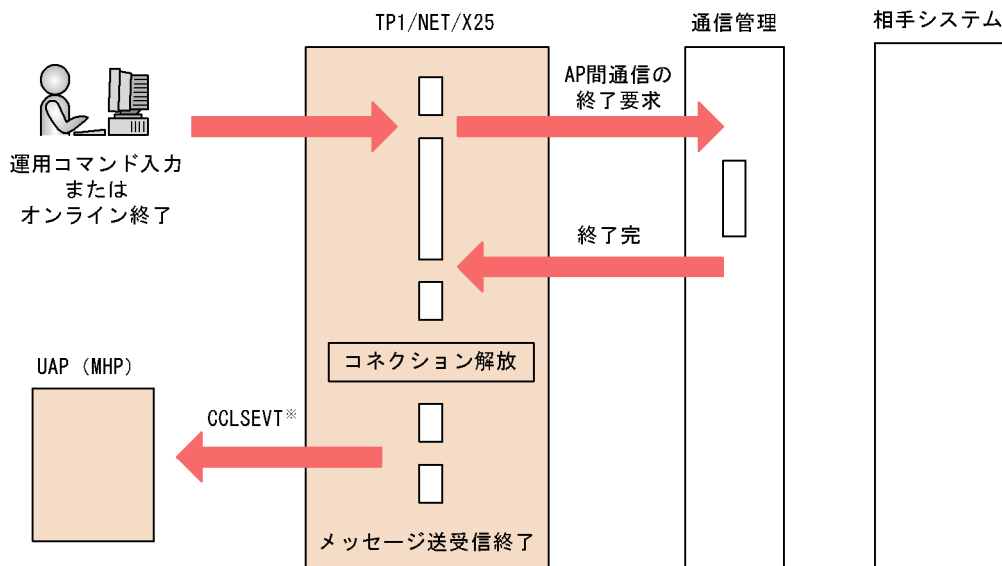


(2) コネクションの解放

オンライン終了時、または運用コマンドの入力によって、TP1/NET/X25 はコネクションを解放します。運用コマンドの入力でコネクションの解放が完了すると、TP1/NET/X25 は、状態通知イベント (CCLSEVT) によって、UAP にコネクションの解放を通知します。ただし、オンライン終了時は、このイベントは通知されません。

コネクション解放の流れを、次の図に示します。

図 2-2 コネクション解放の流れ



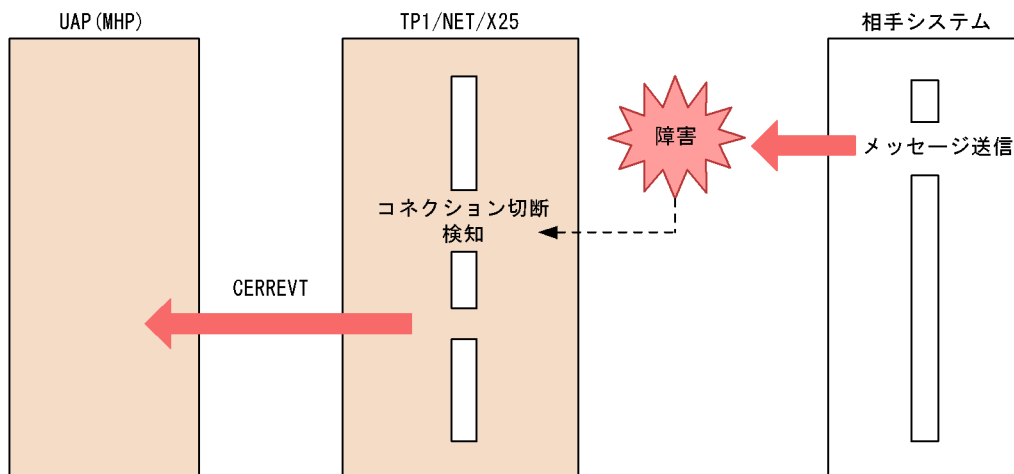
注 オンライン終了時は通知されません。

(3) コネクション切断後の再確立

TP1/NET/X25 は、通信管理から回線障害の情報を受信するとコネクションを切断状態にします。コネクションが切断状態になると、TP1/NET/X25 は、状態通知イベント (CERREVT) によって、障害の発生とコネクションの切断を UAP に通知します。コネクション切断時の障害対策については、「9.2 コネクション障害 (PVC)」を参照してください。

コネクション切断時の流れを、次の図に示します。

図 2-3 コネクション切断時の流れ



コネクション切断後、オンライン開始時に指定した定義オブジェクトファイルの内容、または運用コマンドの入力によってコネクションを再確立できます。

2.1.2 コネクションと論理端末の関係

TP1/NET/X25 は、論理端末 (LE) を通じて、自システムの UAP とメッセージを送受信します。論理端末は、TP1/NET/X25 と UAP の通信接点に当たります。これに対してコネクションとは、TP1/NET/X25 が通信管理を通して、相手システムの UAP とメッセージを送受信するときに確立するものです。コネクションと論理端末の指定を対応させると、自システムとの論理的通信路が確立でき、AP 間通信ができるようになります。コネクションと論理端末は、システム定義時に対応させます。

システム定義では、コネクション一つに対し、論理端末を一つ指定します。

2.1.3 論理端末とアプリケーションの型との関係

TP1/NET/X25 では、メッセージを送受信するために、論理端末の端末タイプとアプリケーションの型をシステム定義時に指定します。TP1/NET/X25 で扱う論理端末の端末タイプは any (任意型) です。論理端末の端末タイプは、MCF 通信構成定義 (mcfaltcle t) で指定します。

アプリケーションは、ユーザが通常、送受信データに指定したアプリケーション名をキーとして一つの UAP (MHP) プロセスで実行されます。アプリケーションには、処理形態によって型があります。TP1/NET/X25 で扱うアプリケーションの型は、非応答型 (noans) です。アプリケーションの型は、MCF アプリケーション定義 (mcfaalcap n) で指定します。MCF アプリケーション定義については、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，MCF 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次の表に示します。

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（PVC）

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	非応答型（noans）	一方受信形態	receive
			分岐送信形態	send
				resend

2.1.4 論理端末の閉塞と閉塞解除

TP1/NET/X25 は，コネクション確立後，または確立と同時に論理端末の閉塞を解除してメッセージの送受信をします。論理端末の状態について次に示します。

1. 閉塞状態

UAP が送信要求したメッセージを出力キューから取り出し，相手システムに送信できない状態です。UAP は該当する論理端末に対して送信要求ができ，相手システムからの受信メッセージのスケジュールもできません。

2. 閉塞解除状態

論理端末が持つ機能を使用できる状態です。

3. 閉塞解除状態で，送信スケジュール抑止状態

1 と同じ内容です。コネクション未確立，または送信完了を待っている間は，この状態になります。

（1）論理端末の閉塞

コネクションの解放または切断と同時に論理端末を自動的に閉塞します。また，オンライン中の任意の時点で，運用コマンド（mcftdctle）を入力して閉塞できます。

（2）論理端末の閉塞解除

コネクション確立直後，論理端末は通常，閉塞状態です。メッセージを送受信するには，論理端末の閉塞解除をする必要があります。

TP1/NET/X25 は，コネクション確立と同時に，自動的に論理端末の閉塞解除をします。

また，コネクション確立状態の任意の時点で，運用コマンド（mcftactle）を入力して論理端末の閉塞解除ができます。論理端末が閉塞解除状態のときに mcftactle コマンドを入力した場合，コマンドはエラーとなります。

2.1.5 メッセージとセグメントの関係

一つの業務で処理するメッセージを、論理メッセージといいます。

TP1/NET/X25 では、相手システムとのメッセージ送受信の場合、一つの論理メッセージが一つのセグメントで構成されます。

UAP と TP1/NET/X25 との間では、セグメント単位でメッセージを送受信します。TP1/NET/X25 を使用する UAP は、相手システムとのメッセージ送受信の場合、常に単一セグメントのメッセージを扱います。

論理メッセージは、他システムへ送信されるときに、パケット分割をされる場合があります。パケット分割をすると性能が低下するため、送信メッセージ長を、通信管理で指定するパケットサイズ以下にすることをお勧めします。

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 で使用す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。

MCF 通信構成定義 (mcftalccn -o) の `ctlheader` オペランドで、制御ヘッダを付けることを指定した場合、メッセージ送信時にはセグメントに制御ヘッダを付けてください。また、メッセージ受信時にはセグメントに制御ヘッダが付けられます。次の機能を使用する場合、制御ヘッダを付けてください。

- PAD メッセージ (Q ビット付きデータパケット) の送受信
PAD メッセージの送受信をする場合、MCF 通信構成定義 (mcftalccn) の `ctlheader` オペランドで、制御ヘッダを付けることを指定してください。また、PAD メッセージ送信時は、制御ヘッダに PAD メッセージであることを設定してください。PAD メッセージ受信時には、制御ヘッダに PAD メッセージであることが設定されます。

制御ヘッダの内容については、「4. メッセージ送受信インタフェース」を参照してください。

2.2 PVC の AP 間通信メッセージの送受信

TP1/NET/X25 は、AP 間通信メッセージの送受信をします。TP1/NET/X25 で使用するメッセージは一方送信メッセージです。

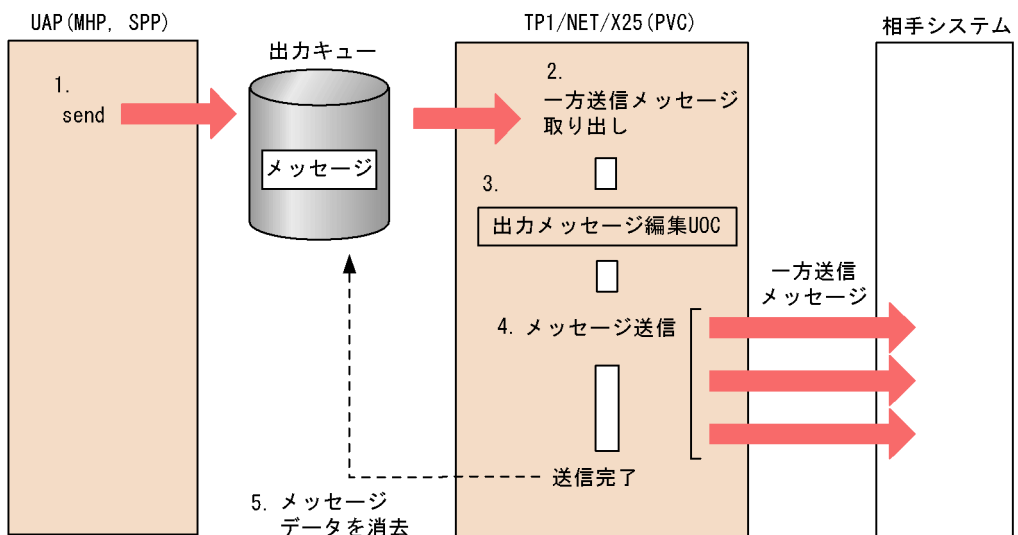
2.2.1 一方送信メッセージの送信

自システムの UAP から相手システムへメッセージを送る形態です。

TP1/NET/X25 が、通信相手システムに一方送信メッセージを送信する手順を説明します。

メッセージ送信の流れを次の図に示します。

図 2-4 PVC のメッセージ送信の流れ



1. UAP は、一方送信メッセージの送信要求をします。
2. TP1/NET/X25 は、出力キューに書き込まれたメッセージを取り出します。
3. 出力メッセージ編集 UOC でメッセージを編集します。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. 相手システムへメッセージを送信します。このとき、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、TP1/NET/X25 は、出力キューにある送信済みのメッセージを消去します。

2.2.2 一方送信メッセージの受信

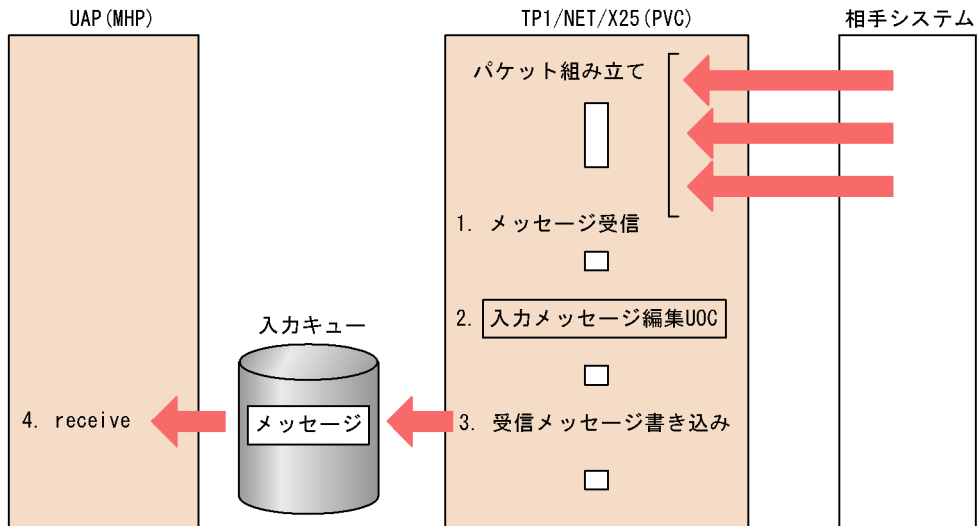
相手システムから一方的に送られたメッセージを受ける形態です。

2. PVC の機能

TP1/NET/X25 が、通信相手システムから一方送信メッセージを受信する手順を説明します。

メッセージ受信の流れを次の図に示します。

図 2-5 PVC のメッセージ受信の流れ



1. TP1/NET/X25 は、相手システムからメッセージを受信します。パケット分割されている場合、最終パケットまで受け取り、一つのメッセージに組み立てます。
2. 入力メッセージ編集 UOC でメッセージを編集します。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。
3. メッセージを入力キューに書き込みます。
4. UAP は receive 関数を呼び出してメッセージを受け取ります。

3

VC の機能

この章では、VC の場合のコネクションの確立方法などの AP 間通信の仕組み、およびメッセージ送受信の流れについて説明します。

3.1 VC の AP 間通信の仕組み

3.2 VC の AP 間通信メッセージの送受信

3.1 VC の AP 間通信の仕組み

TP1/NET/X25 の VC を使用する場合は、コネクションや論理端末を使用した AP 間通信の仕組みについて説明します。

3.1.1 コネクションの確立と解放

TP1/NET/X25 は、AP 間通信をするために、自システムと相手システムとの間に論理的な通信路（コネクション）を確立します。VC では、属性によってコネクションを分類しています。コネクションには、発呼コネクションと着呼コネクションがあります。コネクションの種類は MCF 通信構成定義で指定します。定義の詳細については、「6. システム定義」を参照してください。

コネクションの種類と内容について説明します。

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`）で接続する相手システムを指定し、指定した相手システムとのコネクションを確立します。

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するコネクションです。

MCF 通信構成定義（`mcftalccn -q`）で指定した相手システムからのコネクション確立要求を受信したときだけコネクションを確立します。

（1）コネクションの確立

コネクションの確立方法には、自動確立と手動確立があります。

自動確立

OpenTP1 開始時および再開時に自動的にコネクションを確立します。MCF 通信構成定義（`mcftalccn -i`）で `auto` を指定した場合に確立します。

手動確立

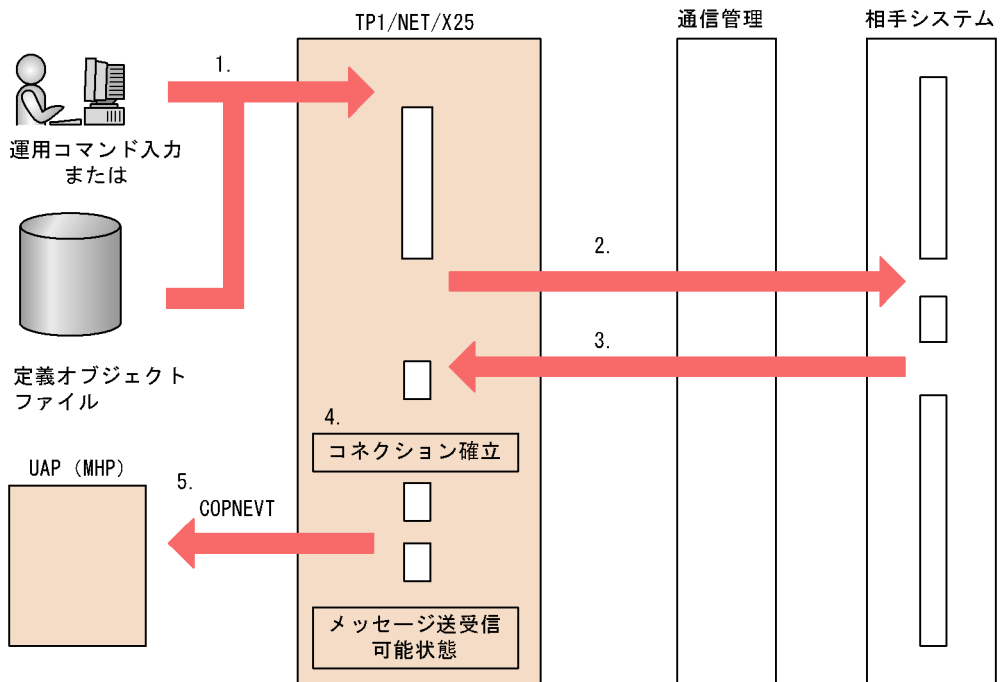
運用コマンド（`mcftactcn`）で確立します。ただし、発呼コネクションの場合に確立します。着呼コネクションの場合は、`mcftactcn` コマンドを入力するとエラーとなります。

VC の場合のコネクション確立の流れを次に示します。

（a）発呼コネクションの確立

発呼コネクションの場合のコネクション確立について、次の図に示します。

図 3-1 VC のコネクション確立の流れ (発呼)



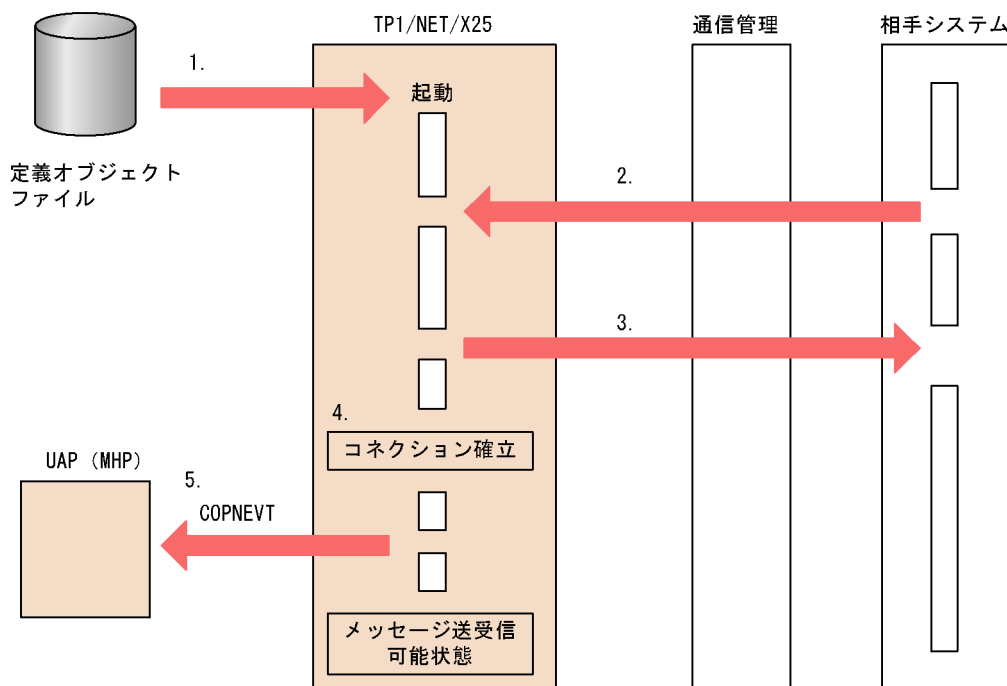
1. OpenTP1 システムの開始および再開始，または運用コマンド (mcftactcn) を入力します。
2. TP1/NET/X25 は，相手システムにコネクションの確立要求をします。
3. 相手システムからコネクションの確立を受け付けたことを通知します。
4. コネクションが確立されます。
5. コネクションが確立されると状態通知イベント (COPNEVT) を通知します。MCF アプリケーション定義に COPNEVT が定義されていない場合は，MHP は起動しません。

(b) 着呼コネクションの確立

着呼コネクションの場合のコネクション確立について，次の図に示します。

3. VC の機能

図 3-2 VC のコネクション確立の流れ (着呼)



1. OpenTP1 システムの開始時および再開時、定義オブジェクトファイルの内容によって、相手システムからのコネクション確立要求を待ちます。
2. 相手システムからコネクションの確立要求を受け付けます。
3. コネクションの確立を受け付けたことを通知します。
4. コネクションが確立されます。
5. コネクションが確立されると状態通知イベント (COPNEVT) を通知します。MCF アプリケーション定義に COPNEVT が定義されていない場合は、MHP は起動しません。

(c) コネクション確立の再試行

コネクション確立を失敗した場合、次の条件を満たしていれば自動的にコネクション確立を再試行します。

- MCF 通信構成定義 (mcfalccn -a) で "aretry=yes" を指定した場合
- 通信管理から取得する障害回復再試行可否情報が回復可能な場合

(2) コネクションの解放

コネクションの解放方法には、自動解放と手動解放があります。

自動解放

OpenTP1 終了時に自動的にコネクションを解放します。

手動解放

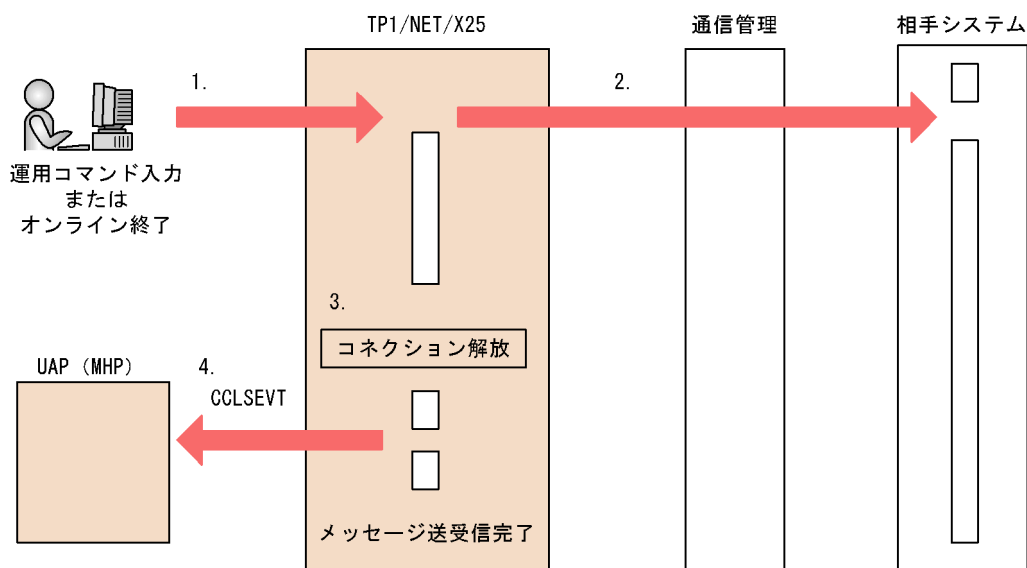
運用コマンド (mcftdcten) を入力してコネクションを解放します。コネクションが解放されると、状態通知イベント (CCLSEVT) を通知します。

コネクションの解放は、コネクションの種類には関係なく、自システムからと相手システムからの解放に分けられます。VC の場合のコネクション解放の流れを次に示します。

(a) 自システムからのコネクションの解放

自システムからのコネクションの解放について、次の図に示します。

図 3-3 自システムからのコネクション解放の流れ

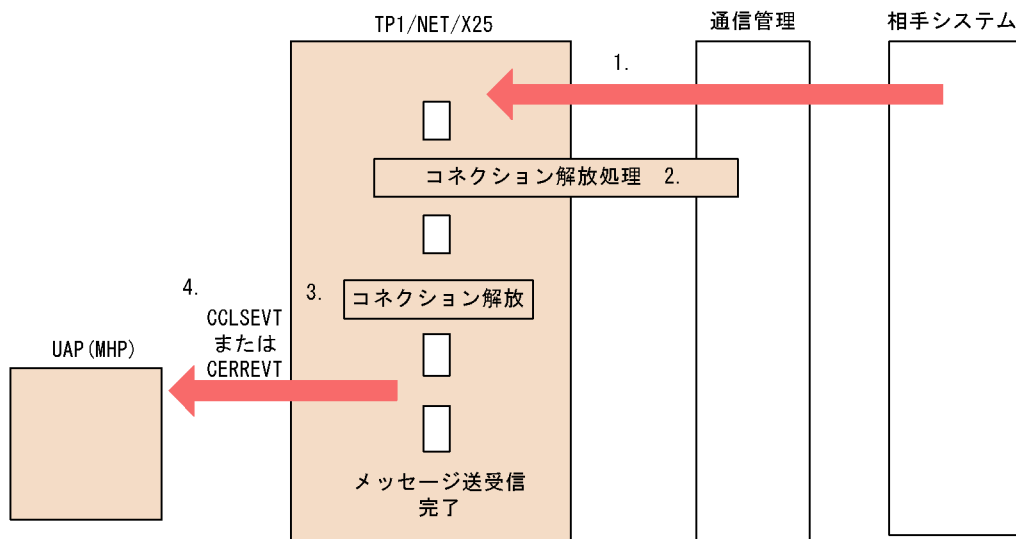


1. OpenTP1 システムの終了、または運用コマンド (mcftdcten) を入力します。
2. TP1/NET/X25 は、相手システムに対してコネクションの解放要求をします。
3. コネクションが解放されます。
4. コネクションが解放されると状態通知イベント (CCLSEVT) を通知します。ただし、OpenTP1 システム終了時と MCF アプリケーション定義に CCLSEVT が定義されていない場合は、MHP は起動しません。

(b) 相手システムからのコネクションの解放

相手システムからのコネクションの解放について、次の図に示します。

図 3-4 相手システムからのコネクション解放の流れ



1. 相手システムからのコネクション解放要求を受け取ります。
2. TP1/NET/X25 は、通信管理との間でコネクションの解放処理をします。
3. コネクションが解放されます。
4. コネクションが解放されると状態通知イベント（CCLSEVT）または障害通知イベント（CERREVT）を通知します。どちらを通知するかは、MCF 通信構成定義（mcftalccn -f）の kind オペランドで指定します。

（3）コネクションの切断

回線の障害などでコネクションが切断されます。コネクションが切断されると、CCLSEVT または CERREVT を通知します。どちらのイベントを通知するかは MCF 通信構成定義（mcftalccn）の kind オペランドで指定します。定義の詳細は「6. システム定義」を参照してください。

3.1.2 コネクションと論理端末の関係

TP1/NET/X25 は、論理端末（LE）を通して、自システムの UAP とメッセージの送受信をします。論理端末は、TP1/NET/X25 と UAP の通信接点に当たります。これに対してコネクションは、TP1/NET/X25 が通信管理を通して、相手システムの UAP とメッセージの送受信をするときに確立されます。コネクションと論理端末の指定を対応させると、自システムとの論理的通信路が確立でき、AP 間通信ができるようになります。コネクションと論理端末はシステム定義時に対応させます。

システム定義では、コネクション一つに対し、論理端末を一つ指定します。

3.1.3 論理端末とアプリケーションの型との関係

TP1/NET/X25 では、メッセージの送受信をするために、論理端末の端末タイプとアプリケーションの型をシステム定義時に指定します。TP1/NET/X25 で扱う論理端末の端末タイプは any (任意型) です。論理端末の端末タイプは MCF 通信構成定義 (mcfaltcle -t) で指定します。

アプリケーションは、ユーザが通常、送受信データに指定したアプリケーション名をキーとして一つの UAP(MHP) プロセスで実行されます。アプリケーションには、処理形態によって型があります。TP1/NET/X25 で扱うアプリケーションの型は、非応答型 (noans) です。アプリケーションの型は、MCF アプリケーション定義 (mcfaalcap -n) で指定します。MCF アプリケーション定義については、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次の表に示します。

表 3-1 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 (VC)

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	非応答型 (noans)	一方受信形態	receive
			分岐送信形態	send
				resend

3.1.4 論理端末の閉塞と閉塞解除

TP1/NET/X25 は、コネクション確立後、または確立と同時に論理端末の閉塞を解除してメッセージの送受信をします。論理端末の状態について次に示します。

1. 閉塞状態

UAP が送信要求したメッセージを出力キューから取り出し、相手システムに送信できない状態です。UAP は該当する論理端末に対して送信要求ができ、相手システムからの受信メッセージのスケジュールもできます。

2. 閉塞解除状態

論理端末が持つ機能を使用できる状態です。

3. 閉塞解除状態で、送信スケジュール抑止状態

1 と同じ内容です。コネクション未確立、または送信完了を待っている間は、この状態になります。

(1) 論理端末の閉塞

VC の場合、コネクションの解放または切断と同時に論理端末を自動的に閉塞します。

また、オンライン中の任意の時点で、運用コマンド (mcftdctle) を入力して閉塞できません。

(2) 論理端末の閉塞解除

VC の場合、コネクション確立と同時に、自動的に論理端末の閉塞解除をします。

また、コネクション確立状態の任意の時点で、運用コマンド (mcftactle) を入力して論理端末の閉塞解除ができます。論理端末が閉塞解除状態のときに mcftactle コマンドを入力した場合、コマンドはエラーとなります。

3.1.5 メッセージとセグメントの関係

一つの業務で処理するメッセージを論理メッセージといいます。TP1/NET/X25 では、相手システムとのメッセージ送受信の場合、一つの論理メッセージが一つのセグメントで構成されます。

UAP と TP1/NET/X25 との間では、セグメント単位でメッセージの送受信をします。TP1/NET/X25 を使用する UAP は、相手システムとのメッセージ送受信の場合、常に単一セグメントのメッセージを扱います。

論理メッセージは、他システムへ送信されるときに、パケット分割をされる場合があります。パケット分割をすると性能が低下するため、送信メッセージ長を、通信管理で指定するパケットサイズ以下にすることをお勧めします。

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 で使用す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。

MCF 通信構成定義 (mcftalccn -o) の ctlheader オペランドで、制御ヘッダを付けることを指定した場合、メッセージ送信時にはセグメントに制御ヘッダを付けてください。また、メッセージ受信時にはセグメントに制御ヘッダが付けられます。次の機能を使用する場合、制御ヘッダを付けてください。

- PAD メッセージ (Q ビット付きデータパケット) の送受信
PAD メッセージの送受信をする場合、MCF 通信構成定義 (mcftalccn -o) の ctlheader オペランドで、制御ヘッダを付けることを指定してください。また、PAD メッセージ送信時には、制御ヘッダに PAD メッセージであることを設定してください。PAD メッセージ受信時には、制御ヘッダに PAD メッセージであることが設定されます。
- 割り込みメッセージ (割り込みパケット) の受信通知
割り込みメッセージの受信通知をする場合、MCF 通信構成定義 (mcftalccn -o) で、制御ヘッダを付けること (ctlheader オペランド)、および割り込みメッセージの受信を通知すること (itdata オペランド) を指定してください。すると、割り込みメッセージ受信時に、制御ヘッダに割り込みメッセージであることが設定されます。

制御ヘッダの内容については、「4. メッセージ送受信インタフェース」を参照してください。

3.2 VC の AP 間通信メッセージの送受信

TP1/NET/X25 は、AP 間通信メッセージの送受信をします。VC で使用するメッセージは一方送信メッセージです。一方送信メッセージの送信と受信について次に示します。

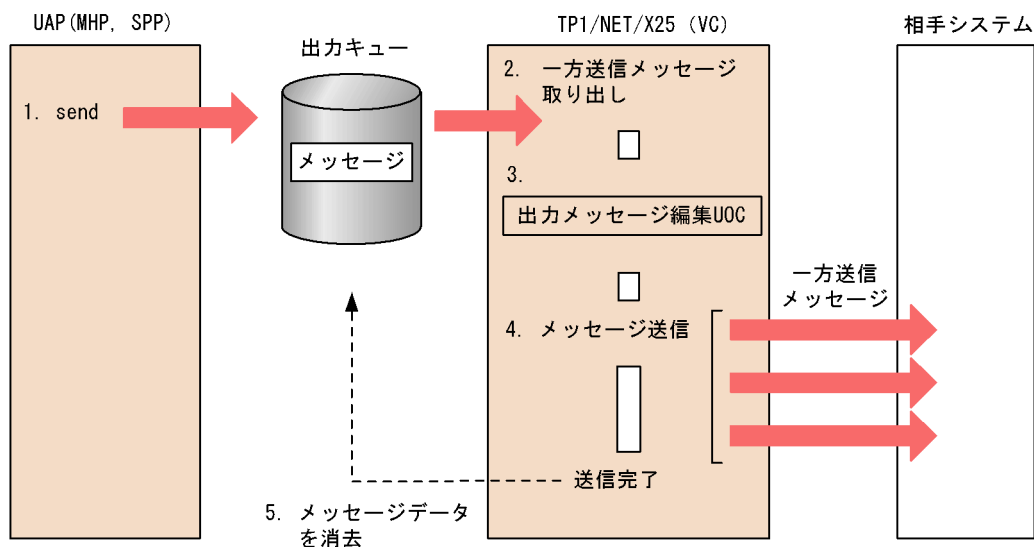
3.2.1 一方送信メッセージの送信

自システムの UAP から相手システムへメッセージを送る形態です。

TP1/NET/X25 が相手システムに一方送信メッセージを送信する手順を説明します。

VC のメッセージ送信の流れを次の図に示します。

図 3-5 VC のメッセージ送信の流れ



1. UAP は、一方送信メッセージの送信要求をします。
2. TP1/NET/X25 は、出力キューに書き込まれたメッセージを取り出します。
3. 出力メッセージ編集 UOC でメッセージを編集します。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. 相手システムへメッセージを送信します。このとき、パケット分割される場合もあります。
5. 送信が完了すると、TP1/NET/X25 は、出力キューにある送信済みのメッセージを消去します。

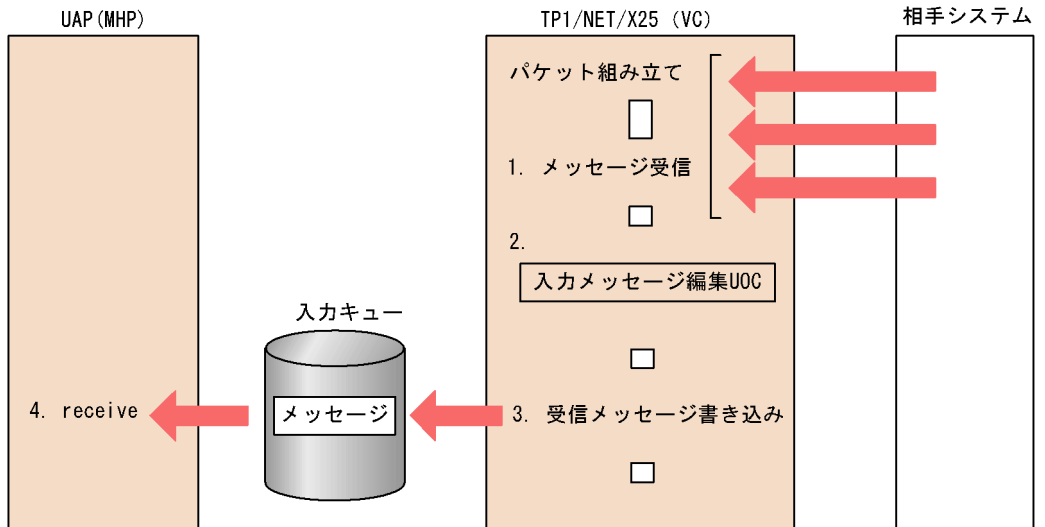
3.2.2 一方送信メッセージの受信

相手システムからのメッセージを受ける形態です。

TP1/NET/X25 が相手システムから一方送信メッセージを受信する手順を説明します。

VC のメッセージ受信の流れを次の図に示します。

図 3-6 VC のメッセージ受信の流れ



1. TP1/NET/X25 は、相手システムからメッセージを受信します。パケット分割されている場合、最終パケットまで受け取り、一つのメッセージに組み立てます。
2. 入力メッセージ編集 UOC でメッセージを編集します。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. メッセージを入力キューに書き込みます。
4. UAP は receive 関数を呼び出してメッセージを受け取ります。

4

メッセージ送受信インタフェース

TP1/NET/X25 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 - 一方送信メッセージの受信 (C 言語)

dc_mcf_resend - 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 ') - 一方送信メッセージの受信 (COBOL 言語)

CBLDCMCF('RESEND ') - 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作成例

メッセージ送受信インタフェースの一覧

TP1/NET/X25 で使用する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 作成の詳細については、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 言語のメッセージ送受信

C 言語で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。

メッセージ送受信の関数を次の表に示します。

表 4-1 メッセージ送受信の関数 (C 言語)

関数名	機能
dc_mcf_receive	一方送信メッセージの受信
dc_mcf_resend	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 言語のメッセージ送受信

COBOL 言語で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。

メッセージ送受信の関数に対応するプログラムを次の表に示します。

表 4-2 メッセージ送受信の関数に対応するプログラム (COBOL 言語)

プログラム名	データ名	機能
CBLDCMCF	'RECEIVE '	一方送信メッセージの受信
	'RESEND '	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) のメッセージ送受信

データ操作言語 (COBOL 言語) を使用した、メッセージ送受信の通信文について説明

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 固有の通信文について、次の表に示します。

表 4-3 メッセージ送受信の通信文（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	一方送信メッセージの受信 CBLDCMCF('RECEIVE')
	SEND	一方送信メッセージの送信 CBLDCMCF('SEND')

注

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 - 一方送信メッセージの受信 (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

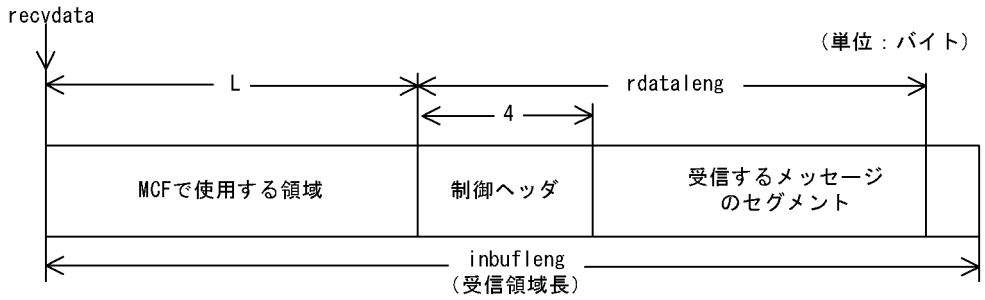
論理端末に届いたメッセージのうち、一つのセグメントを受信します。セグメントの数だけ dc_mcf_receive 関数を呼び出すと、一つの論理メッセージを受信できます。

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。

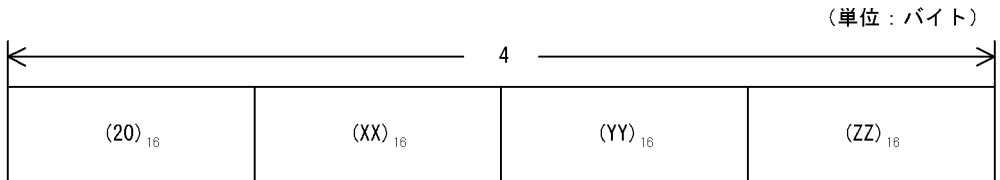
- 相手システムから送信されたメッセージ
- MCF イベント
- アプリケーション起動で渡されたメッセージ

TP1/NET/X25 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。



制御ヘッダは、MCF 通信構成定義で mcftalcn -o ctlheader=use を指定したときだけ付けられます。制御ヘッダの内容を次に示します。



XX

【PVC の場合】

リセットパケットの原因コード。この値は相手システムに依存します。

【VC の場合】

予備 (0)

YY

【PVC の場合】

リセットパケットの診断コード。この値は相手システムに依存します。

【VC の場合】

予備 (0)

ZZ

【PVC の場合】

2^7 ビット: 相手システムから受信したデータパケットの Q ビット

$2^6 \sim 2^5$ ビット: メッセージ種別

- $(10)_2$: リセットメッセージ (テキスト部は存在しません)
- $(00)_2$: 通常メッセージ

$2^4 \sim 2^0$ ビット: 予備 (0)

【VC の場合】

4. メッセージ送受信インタフェース

dc_mcf_receive - 一方送信メッセージの受信 (C 言語)

2^7 ビット : 相手システムから受信したデータパケットの Q ビット

$2^6 \sim 2^5$ ビット : メッセージ種別

- $(11)_2$: 割り込みメッセージ
- $(00)_2$: 通常メッセージ

$2^4 \sim 2^0$ ビット : 予備 (0)

制御ヘッダの構造体は、PVC の場合は <dcmx25c.h> に、VC の場合は <dcmx2m.h> に定義してあります。

各構造体の形式を次に示します。

【PVC の場合】 dcmx25c.h に定義

```
typedef struct {
    unsigned char    id;                ...制御ヘッダID (20)16
    unsigned char    cause;             ...リセットパケットの原因コード
    unsigned char    diag;              ...リセットパケットの診断コード
    unsigned char    dcmx25_ctlbit;
} dcmx25_ctl_type;
```

【VC の場合】 dcmx2m.h に定義

```
typedef struct {
    unsigned char    id;                ...制御ヘッダID (20)16
    unsigned char    rsv1;              ...予備
    unsigned char    rsv2;              ...予備
    unsigned char    dcmx2m_ctlbit;
} dcmx2m_ctl_type;
```

制御ヘッダに対してビット単位でアクセスできません。制御ヘッダ内への設定・参照は、ビット演算で実行するようにしてください。

設定例

データパケットの Q ビット設定時の例

```
dcmx25_ctlbit |= 0x80 ;
```

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、およびメッセージが単一セグメントの場合に設定します。

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。論理端末名称には、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 から値が返ります。

resv01

ヌル文字を設定します。

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 から値が返ります。

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 に設定してください。

recvdata

受信したセグメントの内容が返されます。VC の場合、0 バイトセグメントは受信しないで破棄します。

4. メッセージ送受信インタフェース

dc_mcf_receive - 一方送信メッセージの受信 (C 言語)

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	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71000	-12000	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	-12001	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	-12002	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。
DCMCFRTN_72000	-13000	< 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、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。
		< 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。
DCMCFRTN_72001	-13001	termnam に設定した論理端末名称が間違っています。
DCMCFRTN_72013	-13013	inbufleng の指定値を超えるメッセージ (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	-13016	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	-13024	commform に設定した値が間違っています。

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

リターン値	リターン値 (数値)	意味
DCMCFRTN _72025	-13025	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	-13036	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。

(凡例)

- : 該当しません。

dc_mcf_resend - 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

以前に送信した送信メッセージを、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するとともに、メッセージキューファイルの容量および保持メッセージ数 (メッセージキューサービス定義の quegrp コマンドの -m オプションで指定) に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、一般か優先かどうか、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になります。

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。論理端末名称は最大 8 バイトの長さです。論理端末名称の最後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。

4. メッセージ送受信インタフェース
 dc_mcf_resend - 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になります。

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。論理端末名称は最大 8 バイトの長さです。論理端末名称の最後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	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_NOMSG	-11904	該当するメッセージがありません。
DCMCFRTN_BUF_SHORT	-11905	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。
DCMCFRTN_71002	-12002	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	-12003	メッセージキューが満杯です。
DCMCFRTN_71004	-12004	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を、メモリ上に確保できませんでした。
DCMCFRTN_71108	-12108	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	リターン値 (数値)	意味
DCMCFRTN_72000	-13000	< 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	-13001	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	-13016	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。
		opcd 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。
		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	-13017	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	-13024	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。

(凡例)

- : 該当しません。

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 オプションおよび -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。

4. メッセージ送受信インタフェース

dc_mcf_resend - メッセージの再送 (C 言語)

-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。指定内容によっては、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場合があります。同じ通番を持つメッセージが存在する場合、どのメッセージを再送するかは保証できません。

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)
```

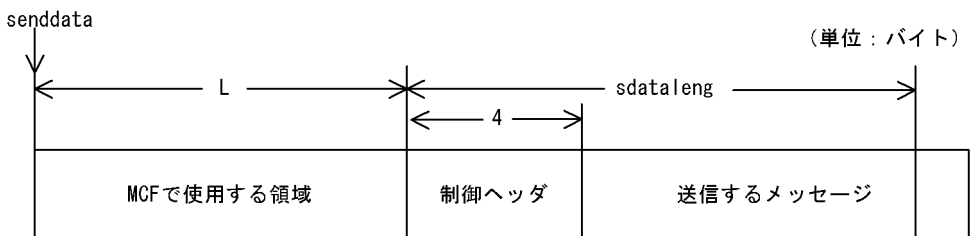
K&R 版 C の形式

```
#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。

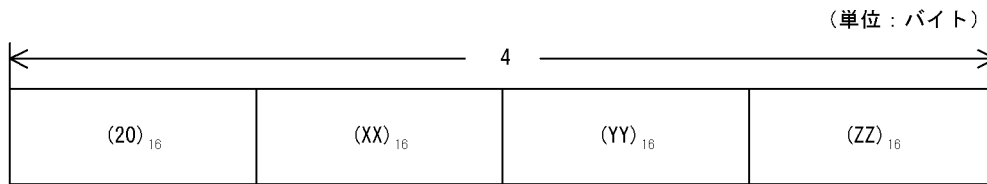
メッセージを送信する領域の形式を次に示します。L は、バッファ形式 1 の場合は 8 バイト、バッファ形式 2 の場合は 4 バイトです。



制御ヘッダは、MCF 通信構成定義で mcftalccn の ctlheader=use を指定したときだけ付けます。制御ヘッダの内容を次に示します。

4. 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)



XX

予備 (0)

YY

予備 (0)

ZZ

2^7 ビット: 相手システムへ送信されるデータパケットの Q ビット

$2^6 \sim 2^0$ ビット: 予備 (0)

制御ヘッダの構造体は、PVC の場合は <dcmx25c.h> に、VC の場合は <dcmx2m.h> に定義してあります。

各構造体の形式を次に示します。

【PVC の場合】dcmx25c.h に定義

```
typedef struct {
    unsigned char    id;                ...制御ヘッダID (20)16
    unsigned char    cause;             ...リセットパケットの原因コード
    unsigned char    diag;              ...リセットパケットの診断コード
    unsigned char    dcmx25_ctlbit;
} dcmx25_ctl_type;
```

【VC の場合】dcmx2m.h に定義

```
typedef struct {
    unsigned char    id;                ...制御ヘッダID (20)16
    unsigned char    rsv1;              ...予備
    unsigned char    rsv2;              ...予備
    unsigned char    dcmx2m_ctlbit;
} dcmx2m_ctl_type;
```

制御ヘッダに対してビット単位でアクセスできません。制御ヘッダ内への設定・参照は、ビット演算で実行するようにしてください。

設定例

データパケットの Q ビット設定時の例

```
dcmx25_ctlbit |= 0x80 ;
```


UAP で値を設定する引数

action

単一セグメントを送信するかどうか、一般か優先かどうか、出力通番を付けるかどうか、および使用するバッファ形式を、次の形式で設定します。

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

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。

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。論理端末名称は最大 8 バイトの長さです。論理端末名称の最後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

送信するメッセージのセグメントの内容を設定した領域を設定します。32000 バイト (制御ヘッダを付けた場合、制御ヘッダの 4 バイトを含みます) まで送信できます。なお、制御ヘッダを使用する場合、4 バイト以下のメッセージは送信しないで、出力キュー

4. 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)

から削除されます。

sdata Leng

送信するメッセージの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	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71002	-12002	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。
		sdata Leng 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	-12003	メッセージキューが満杯です。
DCMCFRTN_71004	-12004	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	-12108	メッセージを送信しようとしたが、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	-13000	< 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	-13001	termnam に設定した論理端末名称が間違っています。
		termnam に設定した出力先の論理端末名称は、MCF で定義していません。
		dc_mcf_send 関数を呼び出せない論理端末を設定しています。

4. 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	リターン値 (数値)	意味
DCMCFRTN _72016	-13016	action に設定したセグメント種別 (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。
		oped 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。
DCMCFRTN _72017	-13017	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	-13024	commform に設定した値が間違っています。
DCMCFRTN _72026	-13026	action に設定したメッセージ種別 (DCMCFEMI) の値が間違っています。
DCMCFRTN _72041	-13041	sdataleng に 0, またはマイナス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。

(凡例)

- : 該当しません。

CBLDCMCF('RECEIVE') - 一方送信メッセージの受信 (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) 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Y.  
03 データ名Y1 PIC X(n).  
03 データ名Y2 REDEFINES データ名Y1.  
04 データ名Y3.  
05 データ名Y4 PIC 9(1).  
05 データ名Y5 PIC 9(1).  
05 データ名Y6 PIC 9(1).  
05 データ名Y7.  
06 データ名Y8 PIC 1(1).  
06 データ名Y9 PIC 1(2).
```

```
06 データ名Y10 PIC 1(5).  
04 データ名Y11 PIC X(n-4).
```

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で渡されたメッセージ

TP1/NET/X25 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。

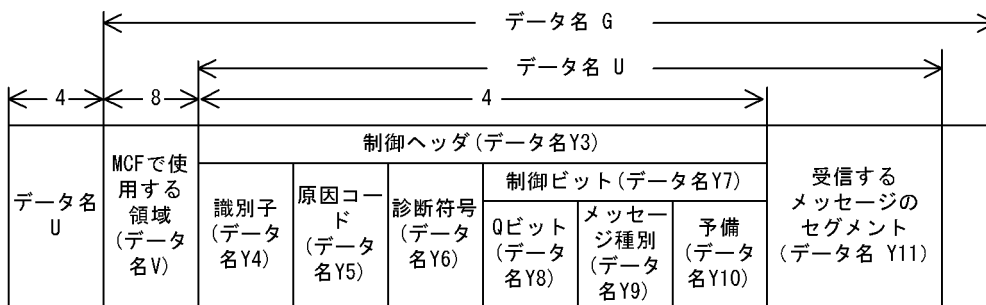
4. メッセージ送受信インタフェース

CBLDCMCF('RECEIVE') - 一方送信メッセージの受信 (COBOL 言語)

制御ヘッダを使用する場合 (MCF 通信構成定義 mcfalccn -o ctlheader=use を指定)

●バッファ形式1の場合

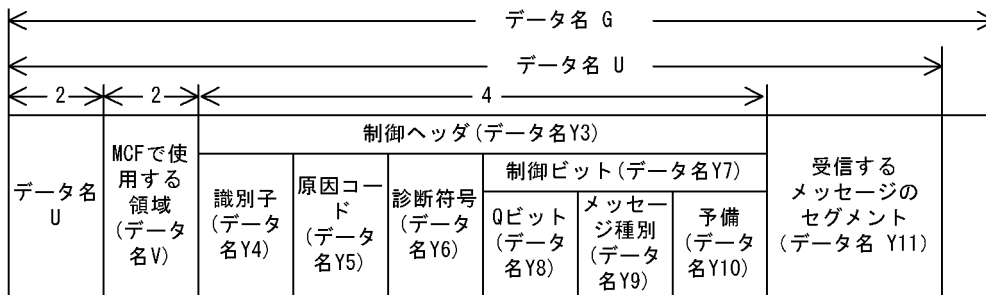
(単位: バイト)



データ名 Y5 およびデータ名 Y6 は、VC の場合予備の領域となります。

●バッファ形式2の場合

(単位: バイト)

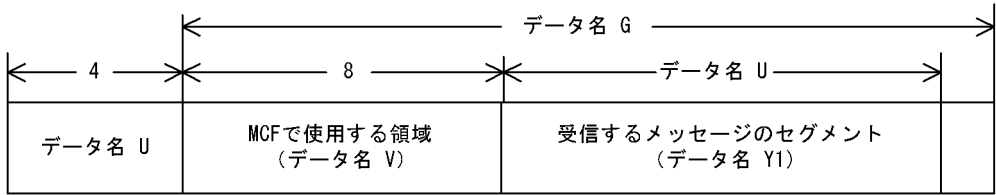


データ名 Y5 およびデータ名 Y6 は、VC の場合予備の領域となります。

制御ヘッダを使用しない場合 (MCF 通信構成定義 mcftalccn -o ctlheader=nouse を指定, または -o オプションを省略)

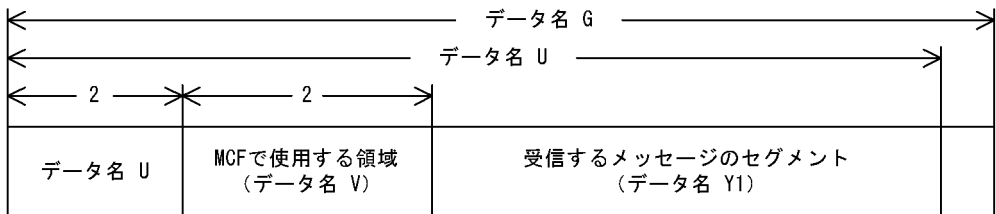
●バッファ形式1の場合

(単位: バイト)



●バッファ形式2の場合

(単位: バイト)



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「VALUE'RECEIVE」を設定します。

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。次のどちらかの値を設定します。

VALUE 'FRST'

先頭セグメントを受信する場合, およびメッセージが単一セグメントの場合に設定します。

VALUE 'SEG'

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。

4. メッセージ送受信インタフェース

CBLDCMCF('RECEIVE') - 一方送信メッセージの受信 (COBOL 言語)

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

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 から値が返ります。

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。

データ名 Y1【制御ヘッダを付けなかった場合】

受信したセグメントの内容が返されます。32000 バイトまで受信できます。

データ名 Y4【制御ヘッダを付けた場合】

制御ヘッダ識別子 (X'20') が返されます。

データ名 Y5【制御ヘッダを付けた場合】

【PVC の場合】

リセットパケットの原因コードが返されます。

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

【VC の場合】

予備の領域です。

データ名 Y6【制御ヘッダを付けた場合】

【PVC の場合】

リセットパケットの診断符号が返されます。

【VC の場合】

予備の領域です。

データ名 Y8【制御ヘッダを付けた場合】

受信したデータパケットの Q ビットが返されます。

データ名 Y9【制御ヘッダを付けた場合】

【PVC の場合】

メッセージ種別が次の値で返されます。

(10)₂: リセットメッセージ

(00)₂: 通常メッセージ

【VC の場合】

メッセージ種別が次の値で返されます。

(11)₂: 割り込みメッセージ

(00)₂: 通常メッセージ

データ名 Y10【制御ヘッダを付けた場合】

予備の領域です。

データ名 Y11【制御ヘッダを付けた場合】

受信したセグメントの内容が返されます。31996 バイトまで受信できます。VC の場合、0 バイトセグメントは受信しないで破棄します。

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 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。

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

ステータスコード	意味
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	<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 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。 <p>< 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。</p>
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	<p>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。</p> <p>< バッファ形式 2 の場合 > 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。</p>
72016	<p>データ名 D に設定した値が間違っています。</p> <p>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。</p> <p>データ名 M7 に設定した値が間違っています。</p>
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。
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 ') - 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 (一般一方送信, 優先一方送信)

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するとともに, メッセージキューファイルの容量および保持メッセージ数 (メッセージキューサービス定義の quegrp コマンドの -m オプションで指定) に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

メッセージの再送を示す要求コード「VALUE 'RESEND 」を設定します。

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

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。

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。

4. メッセージ送受信インタフェース

CBLDCMCF('RESEND') - メッセージの再送 (COBOL 言語)

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

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。論理端末名称は最大 8 バイトの長さです。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。論理端末名称は最大 8 バイトの長さです。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

定めます。

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

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になります。

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、データ名 V に設定した値は無効になります。

空白

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。

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 -e) で指定した値を超えています。

4. メッセージ送受信インタフェース

CBLDPCMCF('RESEND') - メッセージの再送 (COBOL 言語)

ステータスコード	意味
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	メッセージを再送しようとしたますが、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 > 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 オプションおよび -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、CBLDCMCF('RESEND ')

で使用する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。

-I オプション

-I 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、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場合があります。同じ通番を持つメッセージが存在する場合、どのメッセージを再送するかは保証できません。

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.  
03 データ名W1 PIC X(n).  
03 データ名W2 REDEFINES データ名W1.  
04 データ名W3.  
05 データ名W4 PIC 9(1).  
05 データ名W5 PIC 9(2).  
05 データ名W6.  
06 データ名W7 PIC 1(1).  
06 データ名W8 PIC 1(7).  
04 データ名W9 PIC X(n-4).
```

機能

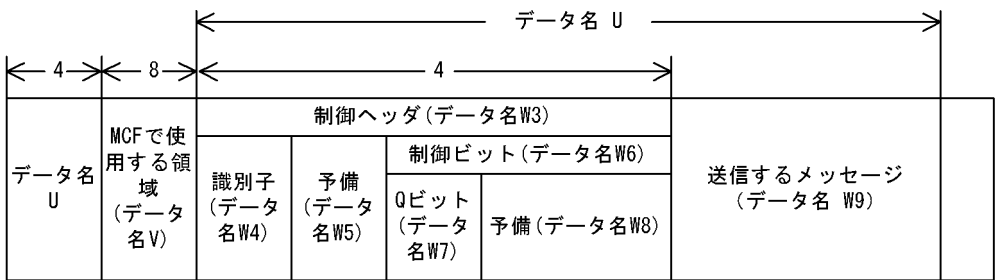
相手システムへ一方送信メッセージを送信します。一方送信メッセージは、一つのセグメントで構成されます。

メッセージを送信する領域 (一意名 3 で示す領域) の形式を次に示します。

制御ヘッダを使用する場合 (MCF 通信構成定義 mcftalccn -o ctlheader=use を指定)

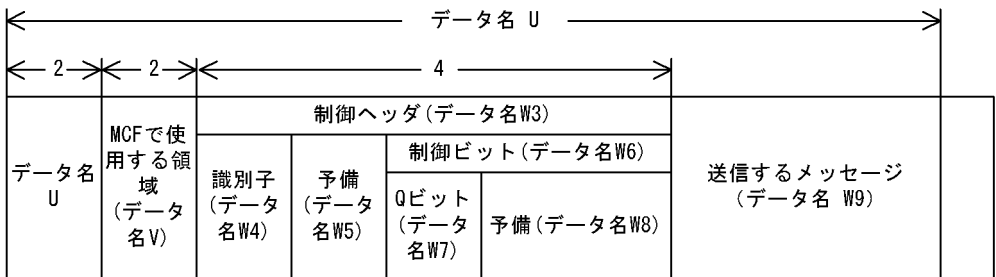
●バッファ形式1の場合

(単位: バイト)



●バッファ形式2の場合

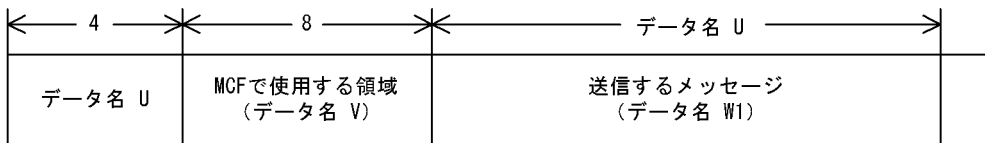
(単位: バイト)



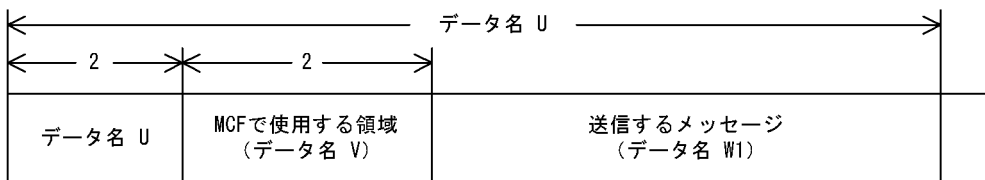
4. メッセージ送受信インタフェース
 CBLDCMCF('SEND') - 一方送信メッセージの送信 (COBOL 言語)

制御ヘッダを使用しない場合 (MCF 通信構成定義 `mcftalccn -o ctlheader=nouse` を指定, または `-o` オプションを省略)

●バッファ形式1の場合 (単位: バイト)



●バッファ形式2の場合 (単位: バイト)



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「VALUE 'SEND」を設定します。

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。

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

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

省略されたものとして、「VALUE 'NSEQ'」(出力通番は必要ない)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

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。

4. 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 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。論理端末名称は最大 8 バイトの長さです。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)

MCF で使用する領域です。

データ名 W1【制御ヘッダを付けなかった場合】

送信するメッセージの内容を設定します。32000 バイトまで送信できます。

データ名 W4【制御ヘッダを付けた場合】

制御ヘッダ識別子 (20)₁₆ を設定します。

データ名 W5【制御ヘッダを付けた場合】

0 を設定します。

データ名 W7【制御ヘッダを付けた場合】

送信するデータパケットの Q ビットを設定します。

データ名 W8【制御ヘッダを付けた場合】

0 を設定します。

データ名 W9【制御ヘッダを付けた場合】

送信するメッセージの内容を設定します。31996 バイトまで送信できます。

なお、制御ヘッダを使用する場合、4 バイト以下のメッセージは送信しないで、出力キューから削除されます。

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 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。
	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。
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。

4. 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
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, またはマイナスの値を設定していません。バッファ形式 2 の場合はデータ名 U に 4 以下, またはマイナスの値を設定していません。
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。

- 一方送信メッセージの受信 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

一方送信メッセージの受信

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。

4. 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)

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。

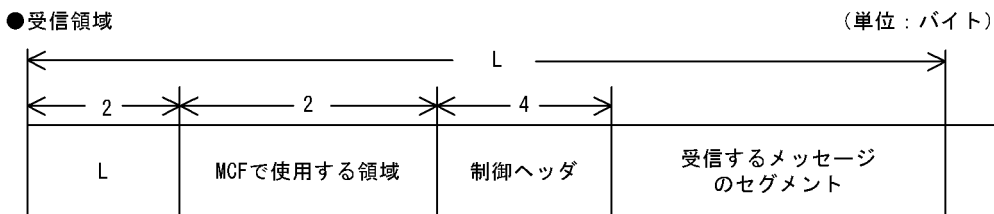
通信文に指定する項目

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。



制御ヘッダは、MCF 通信構成定義で mcftalccn -o ctlheader=use を指定したときだけ付けられます。制御ヘッダの内容を次に示します。



XX

【PVC の場合】

リセットパケットの原因コード。この値は相手システムに依存します。

【VC の場合】

予備 (0)

YY

【PVC の場合】

リセットパケットの診断コード。この値は相手システムに依存します。

【VC の場合】

予備 (0)

ZZ

【PVC の場合】

2⁷ ビット: 相手システムから受信したデータパケットの Q ビット

2⁶ ~ 2⁵ ビット: メッセージ種別

- (10)₂ リセットメッセージ (テキスト部は存在しません)
- (00)₂ 通常メッセージ

2⁴ ~ 2⁰ ビット: 予備 (0)

【VC の場合】

2⁷ ビット: 相手システムから受信したデータパケットの Q ビット

2⁶ ~ 2⁵ ビット: メッセージ種別

- (11)₂: 割り込みメッセージ
- (00)₂: 通常メッセージ

2⁴ ~ 2⁰ ビット: 予備 (0)

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。直前に実行した RECEIVE 文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 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。
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、中間セグメントまたは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 71001 が返されたあとで、RECEIVE 文を実行しています。 < 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。

4. 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
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 の指定値を超える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 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。

- 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

メッセージ送信の型を指定します。

ASYNC

一方送信メッセージの送信

4. 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)

一方送信メッセージの送信の場合に設定します。

データ名 6

次の値を設定したデータ項目

'0': 一方送信メッセージの送信

省略した場合は、一方送信メッセージの送信 (ASYNC) が設定されます。

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。

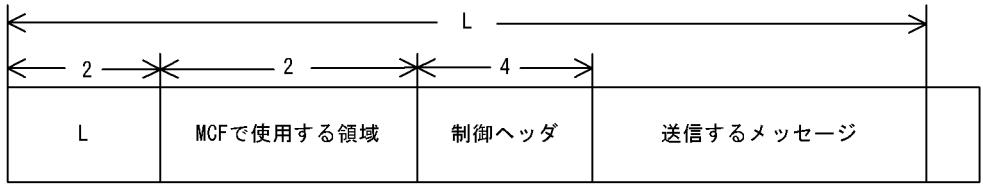
通信文に指定する項目

一意名 1

メッセージを送信するデータ項目を指定します。32000 バイト (制御ヘッダを付けた場合、制御ヘッダの 4 バイトを含みます) まで送信できます。一意名 1 の形式を次に示します。

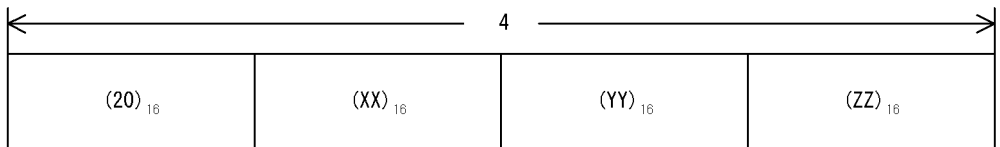
●送信領域

(単位: バイト)



制御ヘッダは、MCF 通信構成定義で `mcftalccn -o ctlheader=use` を指定したときだけ付けます。制御ヘッダの内容を次に示します。なお、制御ヘッダを使用する場合、4 バイト以下のメッセージは送信しないで、出力キューから削除されます。

(単位: バイト)



XX

予備 (0)

YY

予備 (0)

ZZ

2^7 ビット: 相手システムへ送信されるデータパケットの Q ビット

$2^6 \sim 2^0$ ビット: 予備 (0)

WITH 句

次の値で、単一セグメントのメッセージを送信するかどうかを指定します。

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)

省略した場合は、EMI (単一セグメントの送信) が設定されます。

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。

4. メッセージ送受信インタフェース

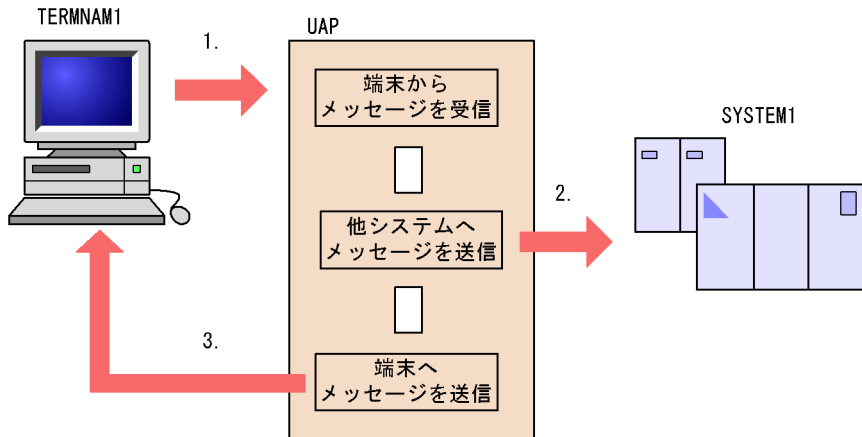
SEND - 一方送信メッセージの送信 (データ操作言語)

ステータスコード	意味
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。
	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。
71108	メッセージを送信しようとしたが、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。
72005	先頭セグメントまたは中間セグメントを送信する SEND 文で、長さが 0 バイトのセグメントを送信しています。
72008	応答メッセージを送信する SEND 文を実行したあとに、再び応答メッセージを送信する SEND 文を実行しています。
	SEND 文を実行して応答型のアプリケーションを起動させたあとに、応答メッセージを送信する SEND 文を実行しています。
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。
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	メッセージを送信する SEND 文で、長さが 4 バイト以下のメッセージを送信しています。
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作成例

UAP 作成例の処理の流れを次の図に示し、その流れに沿ったコーディング例を、C 言語、COBOL 言語、およびデータ操作言語別に示します。なお、コーディング例は、MCF 通信構成定義 (mcftalcn -o) の ctlheader オペランドで use を指定した場合の例です。

図 4-1 UAP 作成例の処理の流れ



C 言語

C 言語 (K&R 版) を使用した UAP コーディング例を示します。

このコーディング例は、次に示すファイルで提供しています。

- PVC の場合：/BeTRAN/examples/mcf/X25/aplib/c/ap1.c
- VC の場合：/BeTRAN/examples/mcf/X25/aplib/c/ap2.c

```
#include <dcmcf.h>
#include <dcmx25c.h> 1
ex_uapl()
{
    char termnam[9] ;
    struct {
        char          mcfuse[8] ;
        dcmx25_ctl_type x25_hd ; 2
        char          data[2048] ;
    } recvddata ;
    DCLONG rdataleng ;
    DCLONG time ;
    DCLONG inbufleng ;
    struct {
        char          mcfuse[8] ;
```

4. 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作成例

```
    dcmx25_ctl_type x25_hd; 2
    char          data[2048];
} senddata ;
static char resv01[9] = "¥0" ;      /* 予備領域の初期化 */
static char resv02[9] = "¥0" ;      /* 予備領域の初期化 */
static char resv03[9] = "¥0" ;      /* 予備領域の初期化 */

char *workadd = (char *)&recvdata ;
inbufleng = sizeof(recvdata) ;
memset(&senddata, 0x00, sizeof(senddata)) ;
senddata.x25_hd.id = 0x20 ;

dc_mcf_receive(DCMCFRST,DCNOFLAGS,termnam,resv01,
               /* メッセージの受信 */
               workadd,&rdata Leng,inbufleng,&time) ;
```

(データの処理)

```
dc_mcf_send(DCMCFEMI,DCMCFOUT,"SYSTEM1",
            /* メッセージの送信(分岐送信) */
            resv01,&senddata,504,resv02,DCNOFLAGS) ;

dc_mcf_send(DCMCFEMI,DCMCFOUT,"TERMNAM1",
            /* メッセージの送信(一方送信) */
            resv01,&senddata,504,resv02,DCNOFLAGS) ;
}
```

注 1

PVC の場合のコーディングです。VC の場合は、次のようにコーディングします。
#include <dcmx2m.h>

注 2

PVC の場合のコーディングです。VC の場合は、次のようにコーディングします。
dcmx2m_ctl_type x25_hd;

COBOL 言語

COBOL 言語を使用した UAP コーディング例を示します。

このコーディング例は、/BeTRAN/examples/mcf/X25/aplib/cobol/ap.cbl のファイルで提供しています。

```
IDENTIFICATION DIVISION.
PROGRAM-ID. EXUAP1.
:
ENVIRONMENT DIVISION.
:
DATA DIVISION.
:
WORKING-STORAGE SECTION.
:
01 RCV.
02 MSG-REC          PIC X(8) VALUE 'RECEIVE '.
```

4. 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作成例

```

02 STATUS-CODE1      PIC X(5) .
02 FILLER            PIC X(3) .
02 SEG-CODE          PIC X(4) VALUE 'FRST' .
02 RTN-CODE          PIC X(4) VALUE SPACE .
02 DAY-ID            PIC 9(8) .
02 TIME-ID           PIC 9(8) .
02 SEG-LENG          PIC 9(9) COMP VALUE 2048 .
02 MCFUSE01          PIC X(4) VALUE SPACE .
02 MCFUSE02          PIC X(4) VALUE SPACE .
02 MCFUSE03          PIC X(4) VALUE SPACE .
02 MCFUSE04          PIC X(4) VALUE SPACE .
02 MCFUSE05          PIC X(8) VALUE SPACE .
02 MCFUSE06          PIC X(4) VALUE SPACE .
02 MCFUSE07          PIC X(8) VALUE SPACE .
02 MCFUSE08          PIC X(4) VALUE SPACE .
02 MCFUSE09          PIC 9(9) COMP VALUE ZERO .
02 MCFUSE10          PIC 9(9) COMP VALUE ZERO .
02 MCFUSE11          PIC X(1) VALUE SPACE .
02 MCFUSE12          PIC X(1) VALUE SPACE .
02 MCFUSE13          PIC X(14) VALUE LOW-VALUE .
01 CD1 .
02 SEG-CODE1         PIC X(4) VALUE SPACE .
02 TERM-CODE         PIC X(8) .
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 CTL-ID1           PIC X(1) VALUE X'20' .
02 CTL-CAUSE1        PIC 9(1) .
02 CTL-DIAG1         PIC 9(1) .
02 CTL-QBIT1         PIC 1(1) BIT .
02 CTL-MSGK1         PIC 1(2) BIT .
02 FILLER            PIC 1(5) BIT .
02 REC-MSGSEG1       PIC X(2044) .
01 SND1 .
02 MSG-SEND          PIC X(8) VALUE 'SEND ' .
02 STATUS-CODE2      PIC X(5) .
02 FILLER            PIC X(3) .
02 MCFUSE21          PIC X(4) VALUE SPACE .
02 MCFUSE22          PIC X(4) VALUE SPACE .
02 MCFUSE23          PIC 9(8) .
02 MCFUSE24          PIC 9(8) .
02 MCFUSE25          PIC 9(9) COMP VALUE ZERO .
02 SEND-SEG          PIC X(4) VALUE 'EMI ' .
02 SEND-SYNC         PIC X(4) VALUE SPACE .
02 SEND-NORM         PIC X(4) VALUE 'NORM' .
02 SEND-NO           PIC X(4) VALUE 'NSEQ' .
02 MCFUSE26          PIC X(8) VALUE SPACE .
02 SEND-CODE         PIC X(4) VALUE SPACE .
02 MCFUSE27          PIC X(8) VALUE SPACE .
02 MCFUSE28          PIC X(4) VALUE SPACE .
02 MCFUSE29          PIC 9(9) COMP VALUE ZERO .

```

4. 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作成例

```

02 MCFUSE30      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE31      PIC X(1) VALUE SPACE.
02 MCFUSE32      PIC X(1) VALUE SPACE.
02 MCFUSE33      PIC X(14) VALUE LOW-VALUE.
01 CD2.
02 SENDSEG-CODE2 PIC X(4) VALUE 'OUT '.
02 TERM-CODE2    PIC X(8) VALUE 'SYSTEM1'.
02 MCFUSE34      PIC X(8) VALUE SPACE.
02 MCFUSE35      PIC X(8) VALUE SPACE.
02 MCFUSE36      PIC X(28) VALUE LOW-VALUE.
01 DATA2.
02 MSGSEG-LENG2  PIC 9(9) COMP VALUE 512.
02 MCFUSE37      PIC X(8).
02 CTL-ID2       PIC X(1) VALUE X'20'.
02 FILLER        PIC 9(1).
02 FILLER        PIC 9(1).
02 CTL-QBIT2     PIC 1(1) BIT.
02 CTL-MSGK2     PIC 1(2) BIT.
02 FILLER        PIC 1(5) BIT.
02 REC-MSGSEG2   PIC X(508).
01 SND2.
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 SEND-SYNC1    PIC X(4) VALUE SPACE.
02 SEND-NORM1    PIC X(4) VALUE 'NORM'.
02 SEND-NO1      PIC X(4) VALUE 'NSEQ'.
02 MCFUSE46      PIC X(8) VALUE SPACE.
02 SEND-CODE1    PIC X(4) VALUE SPACE.
02 MCFUSE47      PIC X(8) VALUE SPACE.
02 MCFUSE48      PIC X(4) VALUE SPACE.
02 MCFUSE49      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE50      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE51      PIC X(1) VALUE SPACE.
02 MCFUSE52      PIC X(1) VALUE SPACE.
02 MCFUSE53      PIC X(14) VALUE LOW-VALUE.
01 CD3.
02 SENDSEG-CODE3 PIC X(4) VALUE 'OUT '.
02 TERM-CODE3    PIC X(8) VALUE 'TERMNAM1'.
02 MCFUSE54      PIC X(8) VALUE SPACE.
02 MCFUSE55      PIC X(8) VALUE SPACE.
02 MCFUSE56      PIC X(28) VALUE LOW-VALUE.
      :
      :

```

PROCEDURE DIVISION.

CALL 'CBLDCMCF' USING RCV CD1 DATA1.

メッセージの受信

処理1

CALL 'CBLDCMCF' USING SND1 CD2 DATA2.

**メッセージの送信(分岐)

送信)**

処理2

```
CALL 'CBLDCMCF' USING SND2 CD3 DATA2.          **メッセージの送信(一方  
送信)**
```

処理3

```
EXIT PROGRAM.
```

データ操作言語

データ操作言語を使用した UAP コーディング例を示します。

このコーディング例は、/BeTRAN/examples/mcf/X25/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. SVRA.  
  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
*  
**** ワーク変数 ****  
*  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
*  
**** メッセージ受信領域 ****  
*  
01 RECV-AREA.  
02 RE-DATALENG PIC 9(4) COMP VALUE 1028.  
02 RE-RSV1 PIC X(2).  
02 RE-ID PIC 9(2) COMP VALUE 32.  
02 FILLER PIC 9(1).  
02 FILLER PIC 9(1).  
02 RE-QBIT PIC 1(1) BIT.  
02 RE-MSGK PIC 1(2) BIT.  
02 FILLER PIC 1(5) BIT.  
02 RE-DATA PIC X(1024).  
*  
**** 一方送信メッセージ送信領域 ****  
*  
01 SEND-IO-AREA.  
02 IO-DATALENG PIC 9(4) COMP VALUE 20.  
02 IO-RSV1 PIC X(2).  
02 IO-ID PIC X(1) VALUE X'20'.  
02 IO-CAUSE PIC 9(1).  
02 IO-DIAG PIC 9(1).  
02 IO-QBIT PIC 1(1) BIT.
```

4. 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作成例

```
02 IO-MSGK          PIC 1(2) BIT.
02 FILLER           PIC 1(5) BIT.
02 IO-DATA          PIC X(16) VALUE 'SVRA SEND DATA1'.

*
**** 通信記述項 ****
*
COMMUNICATION SECTION.
*
**** メッセージの受信の通信記述項 ****
*
CD RECV-INF
  FOR INPUT
  STATUS KEY IS      RE-STATUS
  SYMBOLIC TERMINAL IS RE-TERMNAM
  MESSAGE DATE IS   RE-DATE
  MESSAGE TIME IS   RE-TIME.
*
**** メッセージ送信の通信記述項 ****
*
CD SEND-IO
  FOR OUTPUT
  STATUS KEY IS      SE-STATUS-IO
  SYMBOLIC TERMINAL IS SE-TERMNAM-IO.

PROCEDURE DIVISION.

*
**** メッセージの受信 1 ****
*
RECEIVE RECV-INF
  FIRST SEGMENT
  INTO RECV-AREA.
*
**** 一方送信メッセージの送信 2 ****
*
MOVE 'SYSTEM1' TO SE-TERMNAM-IO.
SEND SEND-IO
  FROM SEND-IO-AREA
  WITH EMI.
*
**** 一方送信メッセージの送信 3 ****
*
MOVE 'TERMNAM1' TO SE-TERMNAM-IO.
SEND SEND-IO
  FROM SEND-IO-AREA
  WITH EMI.

EXIT PROGRAM.
```

5

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 のユーザOWNコーディングのインタフェース，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。

5.1 ユーザOWNコーディングインタフェース

5.2 MCF イベントインタフェース

5.1 ユーザOWNコーディングインタフェース

メッセージ送受信の 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 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 は C 言語で作成します。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。

次に，それぞれの UOC の詳細を説明します。

5.1.1 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，受信した論理メッセージをユーザ任意の形式に変換します。また，受信した論理メッセージを基に，ユーザ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 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 通信構成定義でメッセージ編集用バッファグループ番号（`mcftalccn` の `e msgbuf`）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，入力メッセージの編集ができます。

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。

（2）アプリケーション名の決定

該当する MCF に入力メッセージ編集 UOC が登録されている場合，論理メッセージの編集と同時にアプリケーション名を決定できます。

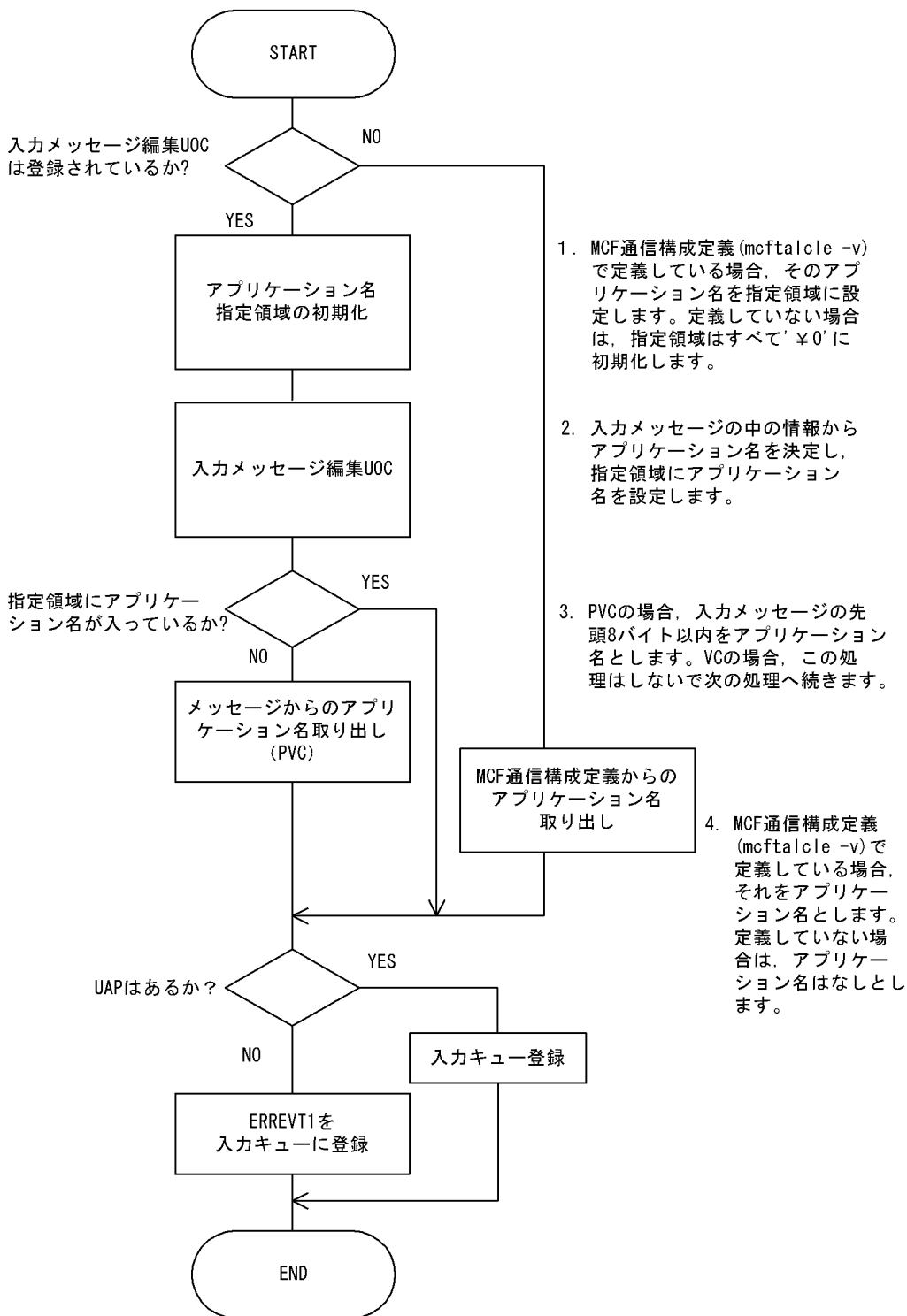
アプリケーション名は, 入力メッセージ編集 UOC が登録されていれば, UOC で決定します。入力メッセージ編集 UOC が登録されていなければ, MCF 通信構成定義 (mcftalcle -v) で指定したアプリケーション名を採用します。さらに, PVC の場合で, MCF 通信構成定義 (mcftalcle -v) でも指定されていないときは, メッセージの先頭から 8 バイト以内をアプリケーション名とします。

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のアプリケーション名の形式は, アプリケーション名格納領域の先頭から, '¥0' の手前までの 1 ~ 8 バイトの英数字です。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を不正とし, 不正アプリケーション名検出通知イベント (ERREVT1) を UAP に通知します。

メッセージの先頭からアプリケーション名を決定する場合 (PVC) のアプリケーション名の形式は, メッセージの先頭から, 空白の手前までの 1 ~ 8 バイトの英数字です。先頭から 9 バイト目までに空白がないとき, または先頭に空白がある場合は, アプリケーション名を不正とし, 不正アプリケーション名検出通知イベント (ERREVT1) を UAP に通知します。

アプリケーション名の決定の処理を次の図に示します。

図 5-1 アプリケーション名の決定の処理



注

MCF 通信構成定義 (mcftalccn -o) で "ctlheader=use" を指定した場合, 必ず入力メッセージ編集 UOC か MCF 通信構成定義 (mcftalcle -v) でアプリケーション名を決定してください。

(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。入力メッセージの編集 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 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, 「8.2 MCF メイン関数の作成」を参照してください。

5.1.2 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, 次に示す形式で呼び出します。

(1) 形式

ANSI C, C++ の場合

```
#include <dcmcf.h>
#include <dcmcfuoc.h>

DCLONG    uoc_func(dcmcf_uoc_min_n *parm)
```

K&R 版 C の場合

```
#include <dcmcf.h>
```

5. ユーザOWNコーディング, MCF イベントインタフェース

```
#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 は次に示す所定のパラメータを 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が使用
    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
    DCLONG buf_size;          ... バッファ最大長
    DCLONG seq_size;          ... バッファ使用長
    char reserve1[4];         ... 予備
    dcmcfuoc_w_type buff_id;  ... MCF内部情報1
    DCLONG buff_addr;         ... 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, 次のどちらかの値が設定されます。

DCMCF_UOC_PRO_X25

X.25 プロトコル (TP1/NET/X25 の PVC)

DCMCF_UOC_PRO_X25VC

X.25 プロトコル (TP1/NET/X25 の VC)

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。

メッセージ編集用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, ebuflist_adr には NULL が設定されます。

aplname

MCF 通信構成定義 (mcftalcle -v) で指定したアプリケーション名が設定されます。

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 の数だけ設定されま
す。

(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

5. ユーザOWNコーディング, MCF イベントインタフェース

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。

(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 ~ -19000 の範囲で指定してください。

(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には, 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

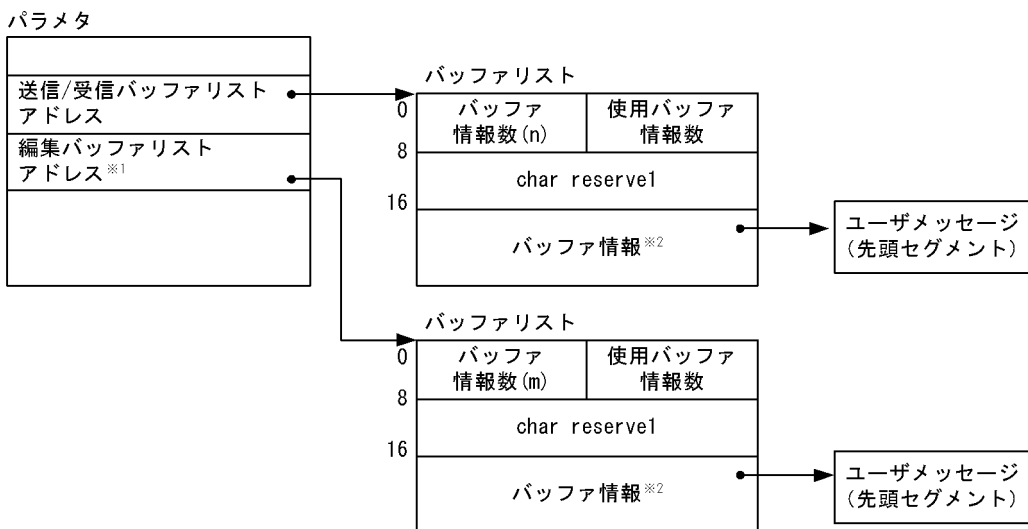
uoc_func() は次のコードでリターンしてください。

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次の図に示します。

図 5-2 UOC インタフェース用のパラメタとバッファの関係



注 1

MCF 通信構成定義 (mcfcalcn) を指定しなければ，ヌル文字となり，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 32 バイトで次の形式をしています。



5.1.3 出力メッセージの編集

出力メッセージの編集 UOC は，送信する論理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。出力メッセージ編集 UOC は，出力キューから全セグメントを読み出すと起動します。

5. 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 通信構成定義でメッセージ編集用バッファグループ番号 (mcf_talccn -e msgbuf) 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処理できます。

また，送信するメッセージのセグメントは，送信バッファ，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。

(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します。

(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します。

(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。「5.1.1(5) OpenTP1 への組み込み方法」を参照してください。

5.1.4 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 <dcmcf.h>
#include <dcmcfuoc.h>

DCLONG    uoc_func(dcmcf_uoc_mout_n *parm)
```

K&R 版 C の場合

```
#include <dcmcf.h>
#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 は次に示す所定のパラメータを parm に設定します。

(3) パラメータの内容

(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 (バッファリスト)，dcmcf_uocbufinf_n (バッファ情報) の内容

入力メッセージ編集 UOC インタフェースのバッファリストおよびバッファ情報の内容と同じです。「5.1.2(3)(b) dcmcf_uocbuff_list_n (バッファリスト) の内容」および「5.1.2(3)(c) dcmcf_uocbufinf_n (バッファ情報) の内容」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，次のどちらかの値が設定されます。

DCMCF_UOC_PRO_X25

X.25 プロトコル (TP1/NET/X25 の PVC)

DCMCF_UOC_PRO_X25VC

X.25 プロトコル (TP1/NET/X25 の VC)

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

5. ユーザOWNコーディング, MCF イベントインタフェース

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。

メッセージ編集用バッファが未定義の場合, つまり, MCF 通信構成定義の `mcf_talcen` 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。

`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, 次のどちらかの値が設定されます。ただし, `outputno_flag` が `DCMCF_UOC_OUTPUTNO_OK` のときだけ有効です。

'n'

一般一方送信メッセージ

'p'

優先一方送信メッセージ

`outputno_flag`

メッセージ出力通番の有効フラグとして, 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` (バッファリスト), `dcmcf_uocbufinf_n` (バッファ情報)

入力メッセージ編集 UOC インタフェースのバッファリストおよびバッファ情報の内容と同じです。「5.1.2(4)(b) `dcmcf_uocbuff_list_n` (バッファリスト)」および「5.1.2(4)(c) `dcmcf_uocbufinf_n` (バッファ情報)」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 ~ -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)，dcmcf_uocbufinf_n (バッファ情報)

入力メッセージ編集 UOC インタフェースのバッファリストおよびバッファ情報の内容と同じです。「5.1.2(5)(b) dcmcf_uocbuff_list_n (バッファリスト)」および「5.1.2(5)(c) dcmcf_uocbufinf_n (バッファ情報)」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	正常リターン (編集バッファでスケジューリング)
DCMCF_UOC_MSG_OK_SND	正常リターン (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，入力メッセージの編集の場合と同じです。「5.1.2(7) パラメタとバッファの関係」を参照してください。

5.1.5 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 は，受け取った通番を基に，ユーザ独自の処理をするための UOC です。例えば，通番を使用して編集した送信メッセージを再送要求します。

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。送信メッセージの通番編集 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。

(2) OpenTP1 への組み込み方法

UAP のメイン関数の中に，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。

5. ユーザOWNコーディング, MCF イベントインタフェース

(a) 形式

ANSI C, C++ の場合

```
#include <dcpcf.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

```
#include <dcpcf.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ています。

(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();
}
```

5.1.6 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、次に示す形式で呼び出します。

(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 から値が渡される引数

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

5. 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渡されます。

sendno

送信メッセージの通番が渡されます。

sendid

送信するメッセージ種別が渡されます。次のどちらかが渡されます。

DCMCF_SEND_PRIOR

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

dataleng

送信メッセージ長が渡されます。

senddata

メッセージの内容が渡されます。

(3) リターン値

リターン値	意味
DC_OK	正常に終了しました。

5.1.7 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設計してください。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 / システムコール（例：shmctl，shmget，shmop）
- メモリ操作（例：memory）
- 文字列操作（例：string）

時間取得関数

(3) UOC の異常処理

TP1/NET/X25 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。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 が実行されたりしてもよいように作成してください。

(5) ユーザ情報の操作方法

UOC でユーザ情報を参照, または設定する場合, ユーザ情報の先頭アドレスがバウンダリ調整されていない場合があります。ユーザ情報の参照および設定方法によっては, バウンダリアクセス例外が発生する場合があります。必要に応じて, メモリ操作関数 (memcpy, memset など) を使用してください。

5.2 MCF イベントインタフェース

OpenTP1 でメッセージの送受信をすると, OpenTP1 の各種システム情報が MHP に通知されます。これを MCF イベントといいます。メッセージ送受信処理でエラーや障害が発生した場合, システム内で何が起きているかが MCF イベント内容でわかります。MCF イベントに対応する MHP 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渡されて, MCF イベント処理用 MHP で回復処理をします。なお, MCF イベントの発生時は, 入力メッセージの編集 UOC は呼び出しません。

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。

5.2.1 MCF イベントの種類

TP1/NET/X25 が通知する MCF イベントを次の表に示します。

表 5-1 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 のセグメント受信関数に, 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。 アプリケーション起動時 (即時起動の場合) 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。
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	アプリケーション起動時 (タイマ起動の場合) 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。

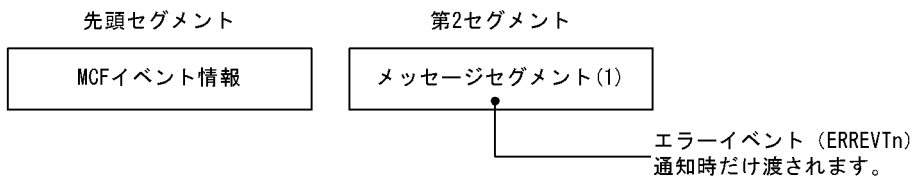
MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP の処理の例
未処理送信メッセージ廃棄通知イベント	ERREVTa	次の理由で未処理送信メッセージを破棄しました。 <ul style="list-style-type: none"> • MCF の正常終了処理時に，未処理送信メッセージの滞留時間監視の時間切れ（タイムアウト）が発生しました。 • 運用コマンド（mcftdlqle）の入力によって，出力キューが削除されました。 	未処理送信メッセージを廃棄したことを報告します。
障害通知イベント	CERREVT	通信管理プログラムのコネクション障害，または論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	メッセージの送受信ができる状態になりました。	メッセージの送受信ができることを報告します。
	CCLSEVT	メッセージの送受信ができない状態になりました。	メッセージの送受信ができないことを報告します。

5.2.2 MCF イベント通知時のセグメント構成

MCF イベントを MHP に通知する場合，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 5-3 MCF イベント通知時のセグメント構成



MCF イベントは，作成した UAP が C 言語の場合と COBOL 言語の場合で，UAP へ渡されるデータ形式が異なります。それぞれの言語の場合について，次に示します。

COBOL 言語を使用したエラーイベント（ERREVTn）の場合は，バッファ形式 1 とバッファ形式 2 と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。「5.2.4 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 ERREVTa 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。

5. ユーザOWNコーディング，MCF イベントインタフェース

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。

5.2.3 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。

PVC の場合，MCF イベントの共通ヘッダ（`dc_mcf_evtheader`），およびエラーイベント（`ERREVTn`）で使用する構造体は `<dc_mcf.h>` で定義されています。また，`CERREVT` 通知インタフェース（`dcmPCM_cerrevt`）は `<dcmPCM.h>` で，`COPNEVT`，`CCLSEVT` 通知インタフェース（`dcmx25_statevt`）は `<dcmx25.h>` で定義されています。
`<dc_mcf.h>`，`<dcmPCM.h>`，`<dcmx25.h>` の順に取り込んでください。

VC の場合，MCF イベントの共通ヘッダ（`dc_mcf_evtheader`），およびエラーイベント（`ERREVTn`）で使用する構造体は `<dc_mcf.h>` で定義されています。また，`CERREVT` 通知インタフェース（`dcmx2m_cerrevt`），および `COPNEVT`，`CCLSEVT` 通知インタフェース（`dcmx2m_statevt`）は，共に `<dcmx2m.h>` で定義されています。`<dc_mcf.h>`，`<dcmx2m.h>` の順に取り込んでください。

MCF イベントとして設定される項目を次に示します。

（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] ;             ... 論理端末名称
    char    cn_name[9] ;              ... コネクション名
    unsigned char format_kind ;       ... MCFが使用
    char    reserve01 ;               ... 予備
    DCLONG time ;                    ... メッセージ入力時刻
};
```

（b）MCF イベントとして設定される項目

`le_name`

メッセージを入力した論理端末名称が設定されます。

`ERREVT2` または `ERREVT3` で，次に示す場合は，`'*` 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 が，さら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

`ERREVT4` の場合は，メッセージを出力する論理端末名称が設定されます。

`CERREVT`，`COPNEVT`，および `CCLSEVT` の場合は無効です。

cn_name

コネクション名が設定されます。

ERREVT2 または ERREVT3 で, 次に示す場合は, '*' 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 が, さら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

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
    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

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。

(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
    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。

5. ユーザOWNコーディング, MCF イベントインタフェース

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。

(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が使用
    char reserve03[2] ; ... 予備
    char ap_name[10] ; ... 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] ; ... トランザクションブランチ
                                ID領域
};
```

(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。
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

トランザクションのブランチ ID が設定されます。

(5) ERREVT4

(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が使用
    char reserve03[2] ; ... 予備
    char ap_name[10] ; ... 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に有効
char    user_data[16] ;         ... 他プロトコルの場合に有効
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。

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) CERREVT

(a) PVC の場合の形式

PVC の CERREVT 通知インタフェース dcmpcm_cerrevt の形式を示します。

```

typedef struct {
struct dc_mcf_evtheader header ;
                                ... MCFイベント情報共通ヘッダ (40バイト)
    DCLONG err_fact ;           ... 障害要因コード (4バイト)
    DCLONG err_reason1 ;       ... 理由コード1 (4バイト)
    DCLONG err_reason2 ;       ... 理由コード2 (4バイト)
    DCLONG recover_code ;      ... 回復可否識別 (4バイト)
    DCLONG retry_code ;        ... 自動リトライ有無識別
} dcmpcm_cerrevt ;

typedef struct {
    dcmpcm_cerrevt comheader ;   ... プロトコル共通ヘッダ
    char reserve[36] ;          ... 予備
} dcmx25_cerrevt ;

```

(b) VC の場合の形式

VC の CERREVT 通知インタフェース dcmx2m_cerrevt の形式を示します。

```

typedef struct {
    struct dc_mcf_evtheader header ;
                                ... MCFイベント情報共通ヘッダ (40バイト)
    DCLONG err_fact ;           ... 障害要因コード (4バイト)
    DCLONG err_reason1 ;       ... 理由コード1 (4バイト)
    DCLONG err_reason2 ;       ... 理由コード2 (4バイト)
    DCLONG err_dctreason ;     ... 切断理由コード
    char    rname[16] ;         ... MCFが使用
    char    reserve1[26] ;      ... 予備
} dcmx2m_cerrevt ;

```

(c) MCF イベントとして設定される項目

err_fact

CERREVT の障害要因コードが次に示す値で設定されます。

5. ユーザOWNコーディング，MCF イベントインタフェース

(00000030)₁₆：コネクション障害発生

(00000031)₁₆：論理端末障害発生

err_reason1, err_reason2

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

recover_code

障害が回復できるかどうかを示す値が設定されます。

(00000000)₁₆：回復可能

(00000010)₁₆：回復不可能

retry_code

障害回復後，自動再確立をするかどうかを示す値が設定されます。

(00000000)₁₆：自動再確立をします。

(00000010)₁₆：自動再確立をしません。

err_dctreason

XNF/AS/NLI から渡されたコネクション切断理由を，加工しないで設定します。コネクション切断理由については，マニュアル「通信管理 XNF/AS 解説・運用編」を参照してください。

自システムから切断した場合は，(00000000)₁₆ を通知します。

この領域は，障害要因コードがコネクション障害 (err_fact で示すコードが (00000030)₁₆) のときだけ有効です。

(7) COPNEVT, CCLSEVT

(a) PVC の場合の形式

PVC の COPNEVT, CCLSEVT 通知インタフェース dcmx25_statevt の形式を示します。

```
typedef struct {
    struct dc_mcf_evtheader header ;    ... MCFイベント情報共通ヘッダ
    char reserve1[56] ;                ... 予備
} dcmx25_statevt ;
```

(b) VC の場合の形式

VC の COPNEVT, CCLSEVT 通知インタフェース dcmx2m_statevt の形式を示します。

```
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 rname[16] ;                   ... MCFが使用
    char reserve1[26] ;                ... 予備
} dcmx2m_statevt ;
```

5.2.4 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 5-2 ~ 表 5-7 に示します。

表 5-2 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。 <ul style="list-style-type: none"> 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	-	-
予備	56	52	8	-	-
予備	64	60	8	-	-
コネクション名	72	68	8	英数字	コネクション名です。
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進数字	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 進数字	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	-	-

5. ユーザOWNコーディング, MCF イベントインタフェース

(凡例)

- : 該当しません。または使用されません。

表 5-3 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。
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, さら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	-	-
コネクション名	72	68	8	英数字	コネクション名です。 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, さら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	-	-

5. 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 進数字	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 進数字	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 進数字	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 5-4 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。

5. 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，さら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	-	-
コネクション名	72	68	8	英数字	コネクション名です。 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，さら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	-	-
メッセージが入力された日付	96	92	8	外部 10 進数字	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 進数字	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	-	-

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
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 UAP の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 進数字	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 5-5 COBOL 言語の MCF イベント情報の内容 (ERREVT A)

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	ERREVT A 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	-	-
コネクション名	72	68	8	英数字	コネクション名です。

5. 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			
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進数字	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 進数字	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。

表 5-6 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	イベントコード「CERREVT」が設定されます。
入力元論理端末名称	8	8	英数字	無効です。
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せん。
メッセージ入力日付	32	8	外部 10 進数字	CERREVT を入力した日付です。
メッセージ入力時刻	40	8	外部 10 進数字	CERREVT を入力した時刻です。

5. 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
障害要因コード	48	4	2進数字	障害要因が設定されます。 (00000030) ₁₆ : コネクション障害 (00000031) ₁₆ : 論理端末障害
理由コード 1 ¹	52	4	2進数字	理由コード 1 が設定されます。
理由コード 2 ¹	56	4	2進数字	理由コード 2 が設定されます。
回復可否識別	60	4	2進数字	障害を回復できるかどうかを示す値が設定されます。 (00000000) ₁₆ : 回復可能状態 (00000010) ₁₆ : 回復不可能状態
自動リトライ有無識別	64	4	2進数字	障害発生後，自動再確立を するかどうかを示す値が設定 されます。 (00000000) ₁₆ : 自動再確立を します (00000010) ₁₆ : 自動再確立を しません
予備	68	36	-	-

VC の場合，60 バイト目以降は次のようになります。

切断理由コード ²	60	4	2進数字	XNF/AS/NLI から渡されたコ ネクション切断理由を，加工 しないで設定します。 自システムから切断した場 合は，(00000000) ₁₆ を通知しま す。 この項目は，障害要因コード がコネクション障害 (00000030) ₁₆ のときだけ有効 です。
予備	64	16	-	-
予備	80	28	-	-

(凡例)

- : 該当しません。または使用されません。

注 1

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。

注 2

切断理由コードについては，マニュアル「通信管理 XNF/AS 解説・運用編」を参照してくだ
さい。

5. ユーザOWNコーディング，MCF イベントインタフェース

表 5-7 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	イベントコード 「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	無効です。
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。
メッセージ入力日付	32	8	外部 10 進数字	COPNEVT, CCLSEVT を入 力した日付です。
メッセージ入力時刻	40	8	外部 10 進数字	COPNEVT, CCLSEVT を入 力した時刻です。
予備	48	56	-	-

VC の場合，48 バイト目以降は次のようになります。

予備	48	16	-	-
予備	64	16	-	-
予備	80	28	-	-

(凡例)

- : 該当しません。または，使用されません。

6

システム定義

この章では、X.25 プロトコルを使用するために必要な、OpenTP1 システム定義の中での TP1/NET/X25 固有の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 の定義の概要

TP1/NET/X25 固有のシステム定義の種類

mcftalccn (コネクション定義の開始) - PVC

mcftalccn (コネクション定義の開始) - VC

mcftalced (コネクション定義の終了) - PVC

mcftalced (コネクション定義の終了) - VC

mcftalcle (論理端末定義) - PVC

mcftalcle (論理端末定義) - VC

mcftcomn (MCF 通信構成共通定義) - PVC

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連づける内容

定義例

TP1/NET/X25 の定義の概要

TP1/NET/X25 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。

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 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

TP1/NET/X25 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 6-1 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	プロトコルごとの実行環境
	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	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、および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 6-1 の定義の中で、TP1/NET/X25 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 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、`mcfbuf` (バッファグループ定義) の `count` オペランド、`length` オペランドの指定値については、「`mcfalecn` (コネクション定義の開始)」の「注意事項」を参照してください。

TP1/NET/X25 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 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を組み込む方法については、「8. 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

X.25 プロトコルを使用して相手システムと通信するためには，TP1/NET/X25 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，通信相手の選択形式（PVC または VC）ごとに，自システムの通信管理プログラムと関連づける内容を示します。

TP1/NET/X25 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 に固有の定義の一覧を次の表に示します。

表 6-2 TP1/NET/X25 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》			
MCF 通信 構成 定義	共通 定義	mcftcomn (PVC の 場合)	-b	bufretry	受信バッファ空き待 ち有無	《yes》 no	
		(MCF 通信構成共通定 義)		retrycnt	受信バッファ空き待 ち再試行回数	符号なし整数 ((0 ~ 128)) 《1》	
		指定数 : 1		retryint	受信バッファ空き待 ち時間	符号なし整数 ((1 ~ 65535)) 《180》 (単位 : 秒)	
	PVC の プロ ト コ ル 固 有 定 義	mcftalccn (コネクショ ン定義の開始)	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	x25
				-g	sndbuf	送信用バッファグ ループ番号	符号なし整数 ((1 ~ 512))
					rvbuf	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	-	コネクション確立方 法	auto 《manual》
				-j	-	接続プロトコル	pvc
-z				vslot	仮想スロット番号	符号なし整数 ((1 ~ 1000))	
	-n	group	論理チャネルグルー プ番号	符号なし整数 ((0 ~ 15))			
number		論理チャネル番号	符号なし整数 ((1 ~ 255))				

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》		
		-x	-	コネクション自動再確立の指定	auto 《manual》	
		-a	aretrycnt	コネクション確立再試行回数	符号なし整数 ((0 ~ 65535)) 《0》	
			aretryint	コネクション確立再試行間隔	符号なし整数 ((0 ~ 2550)) 《0》(単位 : 秒)	
		-l	-	論理端末の閉塞解除の方法	auto 《manual》	
		-o	ctlheader	制御ヘッダ使用の有無	use 《nouse》	
			resetind	リセットパケットの処置方法	use 《nouse》 error	
		-u	spktsize	送信パケットサイズ	符号なし整数 ((128, 256, 512, 1024, 2048, 4096)) 《1024》	
			rpksize	受信パケットサイズ	符号なし整数 ((128, 256, 512, 1024, 2048, 4096)) 《1024》	
		-w	swndsiz e	送信ウィンドウサイズ	符号なし整数 ((1 ~ 7)) 《2》	
			rwndsiz e	受信ウィンドウサイズ	符号なし整数 ((1 ~ 7)) 《2》	
		mcftalcle (論理端末定義) 指定数 : mcftalccn と同数	-l	-	論理端末名称	1 ~ 8 文字の識別子
			-t	-	論理端末の端末タイプ	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》

6. システム定義

TP1/NET/X25 固有のシステム定義の種類

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》		
		-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	送信完了情報取得の要否	《yes》 no	
		-v	-	アプリケーション名	1 ~ 8 文字の識別子	
		mcftalced (コネクション定義の終了) 指定数: mcftalccn と同数	-	-	-	
	VC の プロ トコ ル 固 有 定 義	mcftalccn (コネクション定義の開始) 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	x25
			-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			-	コネクション確立方法	auto 《manual》	
-j			-	接続プロトコル	vc	
-z			vslot	仮想スロット番号	符号なし整数 ((1 ~ 1000))	
-a			aretry	コネクション確立再試行の有無	《yes》 no	
			aretrycnt	コネクション確立再試行回数	符号なし整数 ((0 ~ 65535)) 《0》	
			aretryint	コネクション確立再試行間隔	符号なし整数 ((0 ~ 2550)) 《0》(単位: 秒)	
-o			ctlheader	制御ヘッダ使用の有無	use 《nouse》	

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》	
			itdata	割り込みメッセージ 受信通知の要否	use 《nouse》
		-t	-	コネクションの種類	int 《rsp》
		-q	snpa	相手 SNPA アドレス	1 ~ 30 けたの 数字
			inserv	INS サービス種別	4 けたの 16 進 数字
			inspeed	INS 回線速度	4 けたの 16 進 数字
		-f	kind	MCF イベントの種別	cerr 《ccls》
	mcftalcle (論理端末定義) 指定数 : mcftalcen と 同数	-l	-	論理端末名称	1 ~ 8 文字の 識別子
		-t	-	論理端末の端末タイプ	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	送信完了情報取得の 要否	《yes》 no
		-v	-	アプリケーション名	1 ~ 8 文字の 識別子
	mcftalced (コネクション 定義の終了) 指定 数 : mcftalcen と同数	-	-	コネクション定義の 終了	-

(凡例)

- : 該当しません。

注

TP1/NET/X25 に固有の定義だけ記載してあります。このほかにも、プロトコルで共通の定義コマンド、オプション、オペランドがあります。それらについては、マニュアル「OpenTP1 システム定義」を参照してください。

6. システム定義

TP1/NET/X25 固有のシステム定義の種類

定義コマンドの指定順序

TP1/NET/X25 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す。

図 6-1 プロトコル固有定義コマンドの指定順序

$$\left(\begin{array}{l} \text{mcftalccn (コネクション定義の開始)} \\ \text{mcftalcle (論理端末定義)} \\ \text{mcftalced (コネクション定義の終了)} \end{array} \right) \begin{array}{l} \text{一組として} \\ \text{繰り返し可} \end{array}$$

注

mcftalccn と mcftalcle の指定は、1 対 1 になるようにしてください。

mcftalccn (コネクション定義の開始) - PVC

形式

```
mcftalccn -c コネクションID
          -p x25
          -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 ]
          -j pvc
          -z "vslot=仮想スロット番号"
          -n "group=論理チャネルグループ番号
             number=論理チャネル番号"
[-x auto | manual ]
[-a " {aretrycnt= { コネクション障害時再試行回数 | 0 } }
     { aretryint= { コネクション障害時再試行間隔 | 0 } } " ]
[-l auto | manual ]
[-o " {ctlheader=use | nouse }
     { resetind=use | nouse | error } " ]
[-u " {spktsize=送信パケットサイズ}
     { rpktsize=受信パケットサイズ} " ]
[-w " {swndsize=送信ウィンドウサイズ}
     { rwndsize=受信ウィンドウサイズ} " ]
```

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

OpenTP1 システム内で、一意となるコネクション ID を指定します。

-p x25

プロトコルの種別を指定します。

x25

X.25 プロトコル

-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))

メッセージ送信用バッファの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 オペランドに指定されているバッファグループ番号を指定してください。

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))

メッセージ受信用バッファのグループ番号を指定します。

6. システム定義

mcftalccn (コネクション定義の開始) - PVC

mcftbuf コマンドの -g オプションの groupno オペランドに指定されているバッファグループ番号を指定してください。

-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, 出力メッセージ編集 UOC の場合, メッセージ編集用として使用するバッファグループ番号を指定します。

このオペランドを省略した場合は, 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 オペランドで指定するバッファグループ番号を指定してください。

-i auto | manual ~ 《manual》

MCF 開始時および再開始時に, コネクションを自動的に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, 運用コマンド (mcftactcn) を入力して, コネクションを確立します。

-j pvc

接続プロトコルに, PVC を示す pvc を指定します。

-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。通信管理の構成定義の link 定義文の VASS オペランドで指定した仮想スロット番号を指定してください。

-n

(オペランド)

group= 論理チャネルグループ番号 ~ 符号なし整数 ((0 ~ 15))

論理チャネルグループ番号を指定します。相手システムと対応する値を指定してください。

number= 論理チャネル番号 ~ 符号なし整数 ((1 ~ 255))

論理チャネル番号を指定します。相手システムと対応する値を指定してください。

-x auto | manual ~ 《manual》

障害によるコネクション切断後、自動的にコネクションを再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再確立します。

manual

運用コマンド (mcftactcn) を入力して、コネクションを再確立します。

-a

(オペランド)

aretrycnt= コネクション障害時再試行回数 | 0 ~ 符号なし整数 ((0 ~ 65535)) 《0》
(単位: 回)

コネクション障害が発生した場合、MCF による障害時再試行の回数を指定します。
このオペランドを省略した場合、または 0 を指定した場合は、無限に再試行します。
また、-x オプションを省略した場合、または -x オプションで manual を指定した場合、このオペランドの指定は無効になります。

aretryint= コネクション障害時再試行間隔 | 0 ~ 符号なし整数 ((0 ~ 2550)) 《0》
(単位: 秒)

コネクション障害が発生した場合、MCF による障害時再試行の時間間隔を指定します。
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、MCF は障害時直ちに再試行します。

また、-x オプションを省略した場合、または -x オプションで manual を指定した場合、このオペランドの指定は無効になります。

-l auto | manual ~ 《manual》

コネクションの確立と同時に論理端末の閉塞を解除するかどうかを指定します。

auto

論理端末の閉塞を自動的に解除します。

manual

運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除します。

-o

(オペランド)

ctlheader=use | nouse ~ 《nouse》

UAP と TP1/NET/X25 のメッセージ送受信時に制御ヘッダを使用するかどうかを指定します。

use

6. システム定義

mcftalccn (コネクション定義の開始) - PVC

メッセージに制御ヘッダを付けます。

nouse

メッセージに制御ヘッダを付けません。

PAD メッセージの送受信をする場合、および resetind オペランドで use を指定する場合、必ず use を指定してください。

また、このオペランドで use を指定する場合、アプリケーション名を決定するために、必ず、次のどちらかの処理をしてください。

- mcftalele 定義コマンドの -v オプションの指定。
- アプリケーション名を決定する入力メッセージ編集 UOC の組み込み。

resetind=use | nouse | error ~ 《nouse》

リセットパケットを受信した場合の処置方法を指定します。

use

リセットパケットを受信したことをリセットメッセージによって UAP に通知します。このオペランドで use を指定した場合、必ず、ctlheader オペランドで use を指定してください。

nouse

リセットパケットを受信したことを UAP に通知しません。

error

コネクションを切断し、閉塞します。リセットパケットを受信したことを UAP に通知します。

この場合、送信データがパケット交換網、または回線上で破棄されていることがあります。リセットメッセージを受信した時、UAP はデータの通番合わせや再送の処理が必要です。

-u

(オペランド)

spktsize= 送信パケットサイズ | 1024 ~ 符号なし整数 ((128, 256, 512, 1024, 2048, 4096)) 《1024》

送信パケットサイズを指定します。

次の条件を満たした値を指定してください。

- 自システムの通信管理の構成定義 (link 文の max_DPDU オペランド) で指定した値以下
- 相手システムの受信パケットサイズと同じ

rpksize= 受信パケットサイズ | 1024 ~ 符号なし整数 ((128, 256, 512, 1024, 2048, 4096)) 《1024》

受信パケットサイズを指定します。

次の条件を満たした値を指定してください。

- 自システムの通信管理の構成定義 (link 文の max_DPDU オペランド) で指定し

- た値以下
- 相手システムの送信パケットサイズと同じ

-w

(オペランド)

swndsize= 送信ウィンドウサイズ | 2 ~ 符号なし整数 ((1 ~ 7)) 《2》
送信ウィンドウサイズを指定します。
パケット交換網への加入時に申請した値を指定してください。

rwndsize= 受信ウィンドウサイズ | 2 ~ 符号なし整数 ((1 ~ 7)) 《2》
受信ウィンドウサイズを指定します。
パケット交換網への加入時に申請した値を指定してください。

注意事項

-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 mcfbuf コマンドに対応しています。mcfbuf コマンドでは、1 コネクション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	mcfbuf 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	1 以上
revbuf	受信最大メッセージ長	1 以上 ¹
msgbuf ²	メッセージ編集 UOC で編集後の最大メッセージ長	1

注 1
多くの受信メッセージが連続して発生するコネクションについては、バッファグループを分けて定義してください。

注 2
UOC でメッセージの編集をしない場合は、指定しません。

mcftalccn (コネクション定義の開始) - VC

形式

```
mcftalccn -c コネクションID
          -p x25
          -g "sndbuf=メッセージ送信バッファグループ番号
             rcvbuf=メッセージ受信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 ]
          -j vc
          [-z " {vslot=仮想スロット番号} " ]
          [-a " {aretry=yes | no}
             {aretrycnt=コネクション確立時障害の確立再試行回数}
             {aretryint=コネクション確立時障害の確立再試行間隔} " ]
          [-o " {ctlheader=use | nouse}
             {itdata=use | nouse} " ]
          [-t int | rsp ]
          -q "snpa=相手SNPAアドレス
             {inserv=x'INSサービス種別'}
             {inspeed=x'INS回線速度'} "
          [-f " {kind=cerr | ccls} " ]
```

機能

VCの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

OpenTP1 システム内で、一意となるコネクション ID を指定します。

-p x25

プロトコルの種別を指定します。

x25

X.25 プロトコル

-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))

メッセージ送信用バッファの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 オペランドで指定するバッファグループ番号を指定してください。

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))

メッセージ受信用バッファの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 オペランドで指定するバッファグループ番号を指定してください。

い。

複数のコネクション定義を指定した場合、このオペランドの指定値は、ほかのコネクション定義と同じにしなければなりません。

-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、出力メッセージ編集 UOC の場合、メッセージ編集用として使用するバッファグループ番号を指定します。

このオペランドを省略した場合は、メッセージ編集用バッファは確保されません。

mcfstbuf コマンドの -g オプションの groupno オペランドで指定するバッファグループ番号を指定してください。

-i auto | manual ~ 《manual》

MCF 開始時および再開始時に、コネクションを自動的に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。発呼コネクションの場合に自動的に確立します。

manual

MCF 起動後、運用コマンドの mcfstactcn コマンドを入力して、コネクションを確立します。

同一相手システムに対し、複数のコネクション定義を指定した場合、このオプションの指定値は、該当するコネクション定義で同じにしなければなりません。

-j vc

接続プロトコルに、VC を示す vc を指定します。

-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。

このオペランドは、発呼コネクションの場合、必ず指定してください。

着呼コネクションの場合は指定を省略してください。指定しても無効となります。

通信管理の構成定義の 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てください。

-a

(オペランド)

6. システム定義

mcftalccn (コネクション定義の開始) - VC

aretry=yes | no ~ 《yes》

コネクション確立時に障害が発生した場合、コネクション確立の再試行をしようかを指定します。

yes

コネクション確立の再試行をします。

no

コネクション確立の再試行をしません。

aretrycnt= コネクション確立時障害の確立再試行回数 | 0 ~ 符号なし整数 ((0 ~ 65535)) 《0》

コネクション確立時に障害が発生した場合、コネクション確立の再試行を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、無限に再試行をします。

aretry 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。

なお、通信管理から再試行不可能な障害が通知された場合は、再試行を中止します。

aretryint= コネクション確立時障害の確立再試行間隔 | 0 ~ 符号なし整数 ((0 ~ 2550)) 《0》 (単位: 秒)

コネクション確立時に障害が発生した場合、コネクション確立の再試行をする時間間隔を指定します。

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、障害後直ちに再試行をします。aretry 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。

-o

(オペランド)

ctlheader=use | nouse ~ 《nouse》

UAP と TP1/NET/X25 のメッセージ送受信時に制御ヘッダを使用するかどうかを指定します。

use

メッセージに制御ヘッダを付けます。

nouse

メッセージに制御ヘッダを付けません。

PAD メッセージの送受信をする場合、および itdata オペランドで use を指定する場合、必ず use を指定してください。また、このオペランドで use を指定する場合、アプリケーション名を決定するために、必ず次のどちらかの処理をしてください。

- mcftalcle 定義コマンドの -v オプションの指定
- アプリケーション名を決定する入力メッセージ編集 UOC の組み込み

itdata=use | nouse ~ 《nouse》

割り込みメッセージ (割り込みパケット) の受信を制御ヘッダで通知するかどうか

を指定します。

use

割り込みメッセージの受信を通知します。

nouse

割り込みメッセージの受信を通知しません。割り込みメッセージのデータは破棄します。

また、このオペランドで use を指定する場合、次の指定をしたときに有効です。

```
mcfstalccn -o ctlheader=use
```

```
-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。

このオペランドは、必ず指定してください。

inserv=x'INS サービス種別' ~ 4 けたの 16 進数字

INS サービス種別を指定します。

通信管理が設定する値を指定してください。

このオペランドは、ISDN 接続の場合、必ず指定してください。また、このオペランドを指定するときは、inspeed オペランドも必ず指定してください。

x は、16 進数字で指定することを意味します。

inspeed=x'INS 回線速度' ~ 4 けたの 16 進数字

INS サービスを使用するときの回線速度を指定します。

通信管理が設定する値を指定してください。x は、16 進数字で指定することを意味します。

このオペランドは、inserv オペランドを指定したときだけ有効です。また、ISDN 接続の場合、必ず指定してください。

-f

(オペランド)

kind=cerr | ccls ~ 《ccls》

6. システム定義

mcftalccn (コネクション定義の開始) - VC

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 VC から通知する MCF イベントの種別を指定します。

cerr

CERREVT を通知します。

ccls

CCLSEVT を通知します。

注意事項

-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	1 以上
rcvbuf	受信最大メッセージ長	相手システムから連続して送信されるメッセージ数
msgbuf	メッセージ編集 UOC で編集後の最大メッセージ長	1

注

UOC でメッセージの編集をしない場合は、指定の必要はありません。

送受信バッファは、コネクション確立時にも使用します。

複数コネクションでバッファグループを共用する場合、バッファ数はコネクションごとの和を、バッファサイズはコネクション間で最大の値を指定してください。また、コネクション間で同一のサイズを指定してください。

mcftalced (コネクション定義の終了) - PVC

形式

mcftalced

機能

一つのコネクションの定義の終了を示します。

オプション

ありません。

6. システム定義

mcftalced (コネクション定義の終了) - VC

mcftalced (コネクション定義の終了) - VC

形式

mcftalced

機能

コネクションの定義の終了を示します。

オプション

ありません。

mcftalcle (論理端末定義) - PVC

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

OpenTP1 システム内で、一意となる論理端末名称を指定します。

-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、それ以後 UAP からの送信要求 (dc_mcf_send 関数または SEND 文) はエラーリターンになります。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。

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。
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、それ以後 UAP からの送信要求 (dc_mcf_send 関数または SEND 文) はエラーリターンになります。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。

6. システム定義

mcftalcle (論理端末定義) - PVC

-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 コマンドで指定するキューグループ ID (キューグループ種別は otq) のどれかを指定してください。

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てください。

-o

(オペランド)

aj=yes | no ~ 《yes》

送信完了時の情報を取得するかどうかを指定します。

yes

取得します。

no

取得しません。

-v アプリケーション名 ~ 1 ~ 8 文字の識別子

該当する論理端末で受信したメッセージの起動先アプリケーション名を指定します。

MCF アプリケーション定義 (mcfaalcap -n name) 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。

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。

mcftalcle (論理端末定義) - VC

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

OpenTP1 システム内で、一意となる論理端末名称を指定します。

-t any

この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、それ以後 UAP からの送信要求 (dc_mcf_send 関数または SEND 文) はエラーリターンとなります。
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。

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。
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、それ以後 UAP からの送信要求 (dc_mcf_send 関数または SEND 文) はエラーリターンとなります。
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となります。

6. システム定義

mcftalcle (論理端末定義) - VC

-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 quegrpid オペランドを指定してください。

quegrpid= 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 コマンドで指定するキューグループ ID (キューグループ種別は otq) のどれかを指定してください。この quegrpid 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てください。

-o

(オペランド)

aj=yes | no ~ 《yes》

送信完了時の情報を取得するかどうかを指定します。

yes

取得します。

no

取得しません。

-v アプリケーション名 ~ 1 ~ 8 文字の識別子

該当する論理端末で受信したメッセージの起動先アプリケーション名を指定します。

MCF アプリケーション定義 (mcfaalcap -n name) 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。

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。

mcftcomn (MCF 通信構成共通定義) - PVC

形式

```
mcftcomn          :  
    [-b" {bufretry=yes | no}  
      {retrycnt=受信バッファ空き待ち再試行回数 }  
      {retryint=受信バッファ空き待ち時間 }"]  
    :
```

機能

PVC を使用する場合の、受信バッファ不足時のバッファ空き待ちタイム値を定義します。

オプション

この定義コマンドには、ほかにもオプションがあります。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。

-b

(オペランド)

bufretry=yes | no ~ 《yes》

受信バッファ不足時にバッファの空き待ちをするかどうかを指定します。

yes

受信バッファ不足時に空き待ちをします。

no

受信バッファ不足時に空き待ちをしません。

retrycnt= 受信バッファ空き待ち再試行回数 | 1 ~ 符号なし整数 ((0 ~ 128)) 《1》

受信バッファ空き待ち再試行回数を指定します。

このオペランドの指定値が 0 の場合、無限回のバッファ空き待ちをします。

bufretry オペランドの指定が no の場合、このオペランドの指定は無効です。

retryint= 受信バッファ空き待ち時間 | 180 ~ 符号なし整数 ((1 ~ 65535)) 《180》

(単位：秒)

受信バッファ空き待ち時間を指定します。

bufretry オペランドの指定が no の場合、このオペランドの指定は無効です。

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づけになります。

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。

```
$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。

オペランド

```
module="TP1/NET/X25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 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 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 ~ 2048)

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。

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

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

6. システム定義

システムサービス共通情報定義

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

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 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 ~ 2048))

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。

$(\text{プロトコル制御で使用するファイル記述子数}^1) + 30^2$

注 1

コネクションの総数を 2 倍した値になります。

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、ご注意ください。

なお、1 プロセスで使用できるファイル記述子の最大数は 2048 であるため、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は、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。システム定義の変更などによって、上記のオペランドの指定値の合計が増加する場合は、OS のシステムパラメタの指定を変更してください。

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

```
mcfx25 -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 のプロトコル固有定義ファイルの構文のチェック、および定義オブジェクトファイルを作成します。

ただし、開始から再開始の間は定義オブジェクトファイルを変更できません。変更した場合、再開始時に正常に動作しないおそれがあるためご注意ください。

TP1/NET/X25 のプロトコル固有定義オブジェクトファイル以外の生成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連づける内容

TP1/NET/X25 を使用して AP 間通信をする場合、TP1/NET/X25 の定義と自システム内の通信管理の定義とを関連づける必要があります。

PVC の場合

TP1/NET/X25 の PVC を使用する場合は、自システムの通信管理との定義の関係を説明します。

TP1/NET/X25 の MCF 通信構成定義では、通信管理の構成定義の指定と一致させる値があります。MCF 通信構成定義 (mcftalccn -z) で指定する仮想スロット番号と、XNF/AS の構成定義 (link 文の VASS オペランド) の指定値を一致させてください。通信管理の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。

VC の場合

TP1/NET/X25 の VC を使用する場合は、自システムの通信管理との定義の関係を説明します。

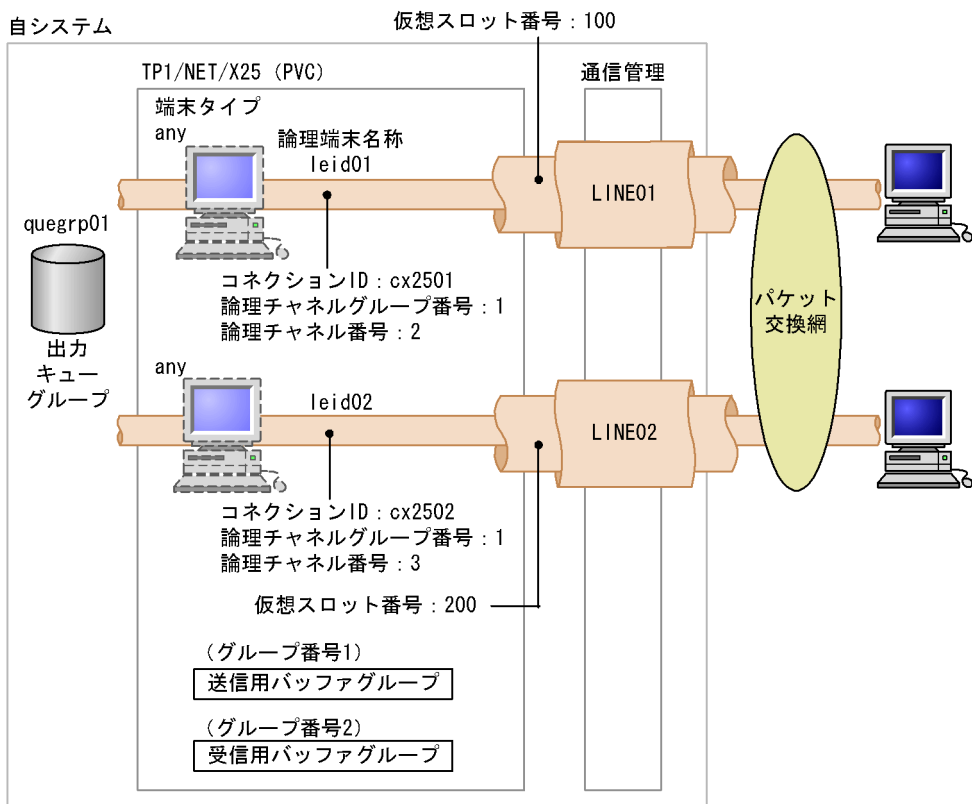
TP1/NET/X25 の MCF 通信構成定義では、通信管理の構成定義の指定と一致させる値があります。MCF 通信構成定義 (mcftalccn -z) で指定する仮想スロット番号と、XNF/AS の構成定義 (link 文の VASS オペランド) の指定値を一致させてください。通信管理の構成定義については、マニュアル「通信管理 XNF/AS 構成定義編」を参照してください。

定義例

PVC の定義例

PVC の場合の TP1/NET/X25 のシステム構成例を次の図に示します。

図 6-2 TP1/NET/X25 のシステム構成例 (PVC の場合)



ここでは、この図で示したシステム構成のコーディング例を示します。

コーディング例は、次に示すファイルで提供されています。

- MCF マネージャ定義の例
/BeTRAN/examples/mcf/X25/conf/mngr1
- 共通定義の例
/BeTRAN/examples/mcf/X25/conf/com_c1
- プロトコル固有定義の例
/BeTRAN/examples/mcf/X25/conf/com_d1

通信管理の構成定義の例 (PVC の場合)

configuration

```
version 1
max_NLI_PVC 2
max_PVC_network_connection 2
max_X25_link 2
max_X25_VASS 2
max_line 2
max_Line_adapter 1
max_link 2
;

NLI_buffer
name buffer1
number 128
size 128
;

Line_adapter
name LP01
location_code 37-08
initial_status active
;

group
name X25
type HDLC
buffer_pool buffer1
;

line
name LINE01
number 01
speed 9.6K
line_type public_PS
line_mode 80PVC
;

link
name LINK01
data_link_address 1
data_link_address2 3
connection_retry 7
connection_time 3.2
data_retry 10
data_time 4.2
busy_retry 8
busy_time 5.0
outstand 7
max_DPDU 1024
VASS 100
;

NL;

line
name LINE02
number 02
speed 9.6K
line_type public_PS
```

6. システム定義
定義例

```

        line_mode 80PVC
        ;

link
    name LINK2
    data_link_address 1
    data_link_address2 3
    connection_retry 7
    connection_time 3.2
    data_retry 10
    data_time 4.2
    busy_retry 8
    busy_time 5.0
    outstand 7
    max_DPDU 1024
    VASS 200
    ;

NL;

```

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 の MCF マネージャ定義の例 (PVC の場合)

```

#### MCFマネージャ定義 ####

mcfmenv      -m "name=mcfmngr"
mcfmcomn     -n 20 ¥
              -p 100
mcfmcname    -s "mcfsvname=_mufX25 ¥
                syssvname=mcfux25 "
mcfmuap      -d 1000 ¥
              -t "sndtim=600 ¥
                  sndrcvtim=600" ¥
              -j 32768 ¥
              -l "initseq=1 ¥
                  maxseq=100 ¥
                  minseq=1" ¥
              -e "segsz=4096" ¥
              -u "ntmetim=300"
mcfmqgid     -q "quekind=otq ¥
                quegrpid=quegrp01"
mcfmqgid     -q "quekind=itq ¥
                quegrpid=quegrp02"
mcfmexp      -g 2 ¥
              -l 20

```

TP1/NET/X25 の共通定義の例 (PVC の場合)

```

#### MCF通信構成定義 共通定義 ####

mcftenv      -s 99 ¥
              -a mcfAPLI
mcftcomn

```



```

mcfttred
mcfttrc    -t  "size = 204800      ¥
              disk = yes         ¥
              bufcnt = 100"
### (送信用)
mcftbuf    -g  "groupno = 1      ¥
              length = 512      ¥
              count  = 2  "
### (受信用)
mcftbuf    -g  "groupno = 2      ¥
              length = 512      ¥
              count  = 4  "
### (編集用)
mcftbuf    -g  "groupno = 3      ¥
              length = 512      ¥
              count  = 2  "

```

TP1/NET/X25 のプロトコル固有定義の例 (PVC の場合)

MCF通信構成定義 (PVCのプロトコル固有定義)

```

#-----<コネクション定義1>-----#
mcftalccn  -c  cx2501            ¥
           -p  x25                ¥
           -g  "sndbuf = 1        ¥
              rcvbuf = 2"        ¥
           -e  "msgbuf = 3"      ¥
           -i  auto                ¥
           -j  pvc                 ¥
           -z  "vslot = 100"     ¥
           -n  "group = 1        ¥
              number = 2"       ¥
           -x  manual              ¥
           -l  auto
#-----<論理端末定義1>-----#
mcftalcle  -l  leid01            ¥
           -t  any                 ¥
           -v  X25AP01            ¥
           -m  "mmsgcnt = 150    ¥
              dmsgcnt = 0"      ¥
           -k  "quekind = disk   ¥
              quegrp1 = quegrp01" ¥
           -o  "aj               = yes"
mcftalced
#-----<コネクション定義2>-----#
mcftalccn  -c  cx2502            ¥
           -p  x25                ¥
           -g  "sndbuf = 1        ¥
              rcvbuf = 2"        ¥
           -e  "msgbuf = 3"      ¥
           -i  auto                ¥
           -j  pvc                 ¥
           -z  "vslot = 200"     ¥
           -n  "group = 1        ¥
              number = 3"       ¥

```

6. システム定義 定義例

```

-x manual          ¥
-l auto
#-----<論理端末定義2>-----#
mcftalcle         -l leid02          ¥
                  -t any           ¥
                  -v X25AP02       ¥
                  -m "mmsgcnt = 150 ¥
                     dmsgcnt = 0"  ¥
                  -k "quekind = disk ¥
                     quegrp01"     ¥
                  -o "aj = yes"
mcftalced

```

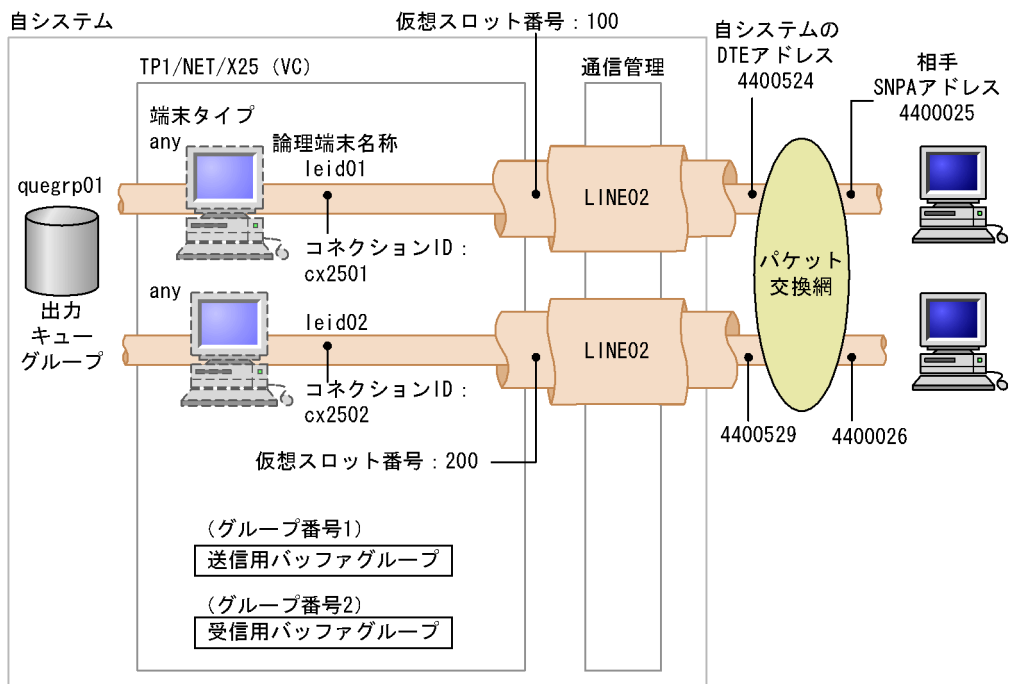
注

数字は通信管理の構成定義の仮想スロット番号に対応します。

VC の定義例

VC の場合の TP1/NET/X25 のシステム構成例を次の図に示します。

図 6-3 TP1/NET/X25 のシステム構成例 (VC の場合)



ここでは、この図で示したシステム構成のコーディング例を示します。

コーディング例は、次に示すファイルで提供されています。

- MCF マネージャ定義の例
/BeTRAN/examples/mcf/X25/conf/mngr2

- 共通定義の例
/BeTRAN/examples/mcf/X25/conf/com_c2
- プロトコル固有定義の例
/BeTRAN/examples/mcf/X25/conf/com_d2

通信管理の構成定義の例 (VC の場合)

```
configuration
  version 1
  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
```

6. システム定義
定義例

```

max_VC_LCN 20
send_window_size 2
receive_window_size 2
;

link
name link102
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 の MCF マネージャ定義の例 (VC の場合)

```

#### MCFマネージャ定義 ####

mcfmenv      -m "name=mcfmng"      ¥
mcfmcomn    -n 20                  ¥
              -p 100
mcfmcname    -s "mcfsvname=_mufX25  ¥
              syssvname=mcfux25 "
mcfmuap     -d 1000                ¥
              -t "sndtim=600        ¥
              sndrcvtim=600"       ¥
              -j 32768              ¥
              -l "initseq=1         ¥
              maxseq=100            ¥
              minseq=1              ¥
              -e "segsize=4096"     ¥
              -u "ntmetim=300"
mcfmqgid    -q "quekind=otq        ¥
              quegrp=quegrp01"
mcfmqgid    -q "quekind=itq        ¥
              quegrp=quegrp02"
mcfmexp     -g 2                    ¥
              -l 20

```

TP1/NET/X25 の共通定義の例 (VC の場合)

```

#### MCF通信構成定義 共通定義 ####

mcftenv     -s 99                  ¥

```

```

-a mcfAPLI
mcftcomn
mcfttred
mcfttrc -t "size = 204800      ¥
          disk = yes         ¥
          bufcnt = 100"
### (送信用)
mcftbuf -g "groupno = 1      ¥
          length = 512      ¥
          count  = 2  "
### (受信用)
mcftbuf -g "groupno = 2      ¥
          length = 512      ¥
          count  = 4  "
### (編集用)
mcftbuf -g "groupno = 3      ¥
          length = 512      ¥
          count  = 2  "

```

TP1/NET/X25 のプロトコル固有定義の例 (VC の場合)

MCF通信構成定義 (VCのプロトコル固有定義)

```

#-----<コネクション定義1>-----#
mcftalccn -c cx2501      ¥
          -p x25         ¥
          -g "sndbuf = 1  ¥
              rcvbuf = 2" ¥
          -e "msgbuf = 3" ¥
          -i auto        ¥
          -j vc          ¥
          -z "vslot = 100" ¥
          -q "snpa = 4400025"
#-----<論理端末定義1>-----#
mcftalcle -l leid01      ¥
          -t any         ¥
          -v X25AP01     ¥
          -m "mmsgcnt = 150 ¥
              dmsgcnt = 0"  ¥
          -k "quekind = disk ¥
              quegrp01"    ¥
          -o "aj         = yes"
mcftalced
#-----<コネクション定義2>-----#
mcftalccn -c cx2502      ¥
          -p x25         ¥
          -g "sndbuf = 1  ¥
              rcvbuf = 2" ¥
          -e "msgbuf = 3" ¥
          -i auto        ¥
          -j vc          ¥
          -z "vslot = 200" ¥
          -q "snpa = 4400026"
#-----<論理端末定義2>-----#

```

6. システム定義

定義例

```
mcftalcle  -l leid02                ¥
            -t any                  ¥
            -v X25AP02              ¥
            -m "mmsgcnt = 150      ¥
                dmsgcnt = 0"       ¥
            -k "quekind = disk     ¥
                quegrp1d = quegrp01" ¥
            -o "aj                  = yes"
mcftalced
```

注

数字は通信管理の構成定義の仮想スロット番号に対応します。

7

運用コマンド

この章では、TP1/NET/X25 で使用する運用コマンドについて説明します。

TP1/NET/X25 の運用コマンド

mcfactcn (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 の運用コマンド

TP1/NET/X25 は、オンライン中に運用コマンドを入力できます。

ここでは、TP1/NET/X25 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。

TP1/NET/X25 で使用する運用コマンドを次の表に示します。

表 7-1 TP1/NET/X25 で使用する運用コマンド

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	mefctacten	×	×	
	コネクションの解放	mefctdcten	×	×	
	コネクションの状態表示	mefctlsen	×	×	
論理端末管理	論理端末の閉塞解除	mefctactle	×	×	
	論理端末の閉塞	mefctdctle	×	×	
	論理端末の状態表示	mefctlsle	×	×	

(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。

mcftactcn (コネクションの確立)

形式

mcftactcn [-s MCF通信プロセス識別子] -c コネクションID

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 に対して、mcftactcn 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。ただし、同一コネクション ID は、重複して指定できません。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。

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。一括指定の場合も、引用符 (") で囲みます。

*: 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*"

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftactcn 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

7. 運用コマンド

mcftactcn (コネクションの確立)

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	mcftactcn 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	mcftactcn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftactcn 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
KFCA14012-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
KFCA14013-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
KFCA14014-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
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	指定したコネクショングループは登録されていません。	標準エラー出力
KFCA16832-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
KFCA16833-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
KFCA16834-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
KFCA16839-W	着呼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

mcfactle (論理端末の閉塞解除)

形式

```
mcfactle [-s MCF通信プロセス識別子] [-c コネクションID]
          -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 に対して、mcfactle 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

閉塞解除したい論理端末に対応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 は複数指定できません。また、一括指定もできません。

-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。

-c オプションを指定した場合は、指定したコネクション ID に対応する論理端末の名称を指定してください。

論理端末名称は、一度に 8 個まで指定できます。同一論理端末名称は重複して指定できません。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。

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。一括指定の場合も、引用符 (") で囲みます。

* : 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

7. 運用コマンド

mcftactle (論理端末の閉塞解除)

```
-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
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	mcftactle 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	mcftactle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftactle 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	指定した接続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	指定した接続 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	mcftactle 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	指定した接続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
KFCA14016-W	論理端末自動起動指定のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA14017-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
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16831-W	接続が未確立のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA16833-W	接続確立処理中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA16834-W	接続解放処理中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA16836-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

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  
          [-f]
```

機能

コネクションを解放します。

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 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。ただし、同一コネクション ID は、重複して指定できません。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。

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。一括指定の場合も、引用符 (") で囲みます。

*: 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*"
```

```
-f
```

該当するコネクションを強制的に解放します。

7. 運用コマンド

meftdctcn (コネクションの解放)

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	meftdctcn 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	meftdctcn 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	meftdctcn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	meftdctcn 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
KFCA10501-I	ヘルプメッセージ	標準出力
KFCA14014-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
KFCA14015-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
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	指定したコネクショングループは登録されていません。	標準エラー出力
KFCA16831-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
KFCA16833-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
KFCA16834-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] [-c コネクションID]
          -l 論理端末名称
```

機能

論理端末を強制的に閉塞します。

PVC の場合、メッセージ送信中にこのコマンドが入力された場合、メッセージを出力キューから取り出す前の状態に戻します。その後、コネクションを強制的に解放します。

VC の場合、メッセージ送信中にこのコマンドが入力された場合、論理端末が使用中であるため、コマンドは拒否されます。その後、コネクションは解放されません。

なお、論理端末が閉塞状態でも、UAP から送信要求できます。

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 に対して、mcftdctl 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

閉塞したい論理端末に対応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 は複数指定できません。また、一括指定もできません。

-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。

-c オプションを指定した場合、指定したコネクション ID に対応する論理端末名称を指定してください

論理端末名称は、一度に 8 個まで指定できます。同一論理端末名称は、重複して指定できません。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。

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。一括指定の場合も、引用符 (") で囲みます。

7. 運用コマンド

mcftdctl (論理端末の閉塞)

* : 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
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	mcftdctl 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	mcftdctl コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftdctl 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	指定した接続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	指定された接続 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
KFCA10504-I	ヘルプメッセージ	標準出力
KFCA14016-W	論理端末自動起動指定のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA14018-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
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16835-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
KFCA16838-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

注意事項

メッセージの受信仕掛り中にこのコマンドを実行した場合、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。

メッセージの送信仕掛り中にこのコマンドを実行した場合、mcfldctle コマンドはエラーリターンします。このとき、一方送信メッセージの送信仕掛り中であれば論理端末は閉塞されません。それ以外のときは、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 [-d]
```

機能

コネクションの状態を標準出力に表示します。

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 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。ただし、同一コネクション ID は、重複して指定できません。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。

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。一括指定の場合も、引用符 (") で囲みます。

*: 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*"
```

```
-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

PVC の場合

```
mmm cccccccc ppp sssss ddd uuuuu  
  llllllllll ttt
```

注

-d オプションを指定しないで mcfllscn コマンドを実行した場合は、「mmm cccccccc ppp sssss ddd uuuuu」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
X25 : X.25 プロトコル
- 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
- ddd : 詳細ステータス (保守情報)
- uuuuu : 接続プロトコル種別
(PVC) : PVC 接続
- llllllll : 論理端末名称
- ttt : 論理端末の端末タイプ
ANY : any 型

VC の場合

```
mmm cccccccc ppp sssss uuuu  
  llllllllll ttt
```

注

-d オプションを指定しないで mcfllscn コマンドを実行した場合は、「mmm cccccccc ppp sssss uuuu」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
X25 : X.25 プロトコル
- sssss : コネクション状態
ACT : 確立
ACT/B : 確立処理中
DCT : 解放

7. 運用コマンド

mcfctlscn (コネクションの状態表示)

DCT/B : 解放処理中

- uuuu : 接続プロトコル
(VC) : VC 接続
- llllll : 論理端末名称
- ttt : 論理端末の端末タイプ
ANY : any 型

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcfctlscn 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	mcfctlscn 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
KFCA10362-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	mcfctlscn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcfctlscn 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	指定したコネクショングループ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] [-c コネクションID]
          -l 論理端末名称 [-q] [-t]
```

機能

論理端末の状態を標準出力に表示します。

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 に対して、mcftlsle 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

状態を表示したい論理端末に対応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 は複数指定できません。また、一括指定もできません。

-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。

-c オプションを指定した場合、指定したコネクション ID に対応する論理端末名称を指定してください

論理端末名称は、一度に 8 個まで指定できます。同一論理端末名称は、重複して指定できません。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。

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。一括指定の場合も、引用符 (") で囲みます。

*: 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

7. 運用コマンド

mcftlsle (論理端末の状態表示)

-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]
      IO xxxxxxxxxxxx yyyyyyyyyy zzzzzzzzzz
      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
DCT : 閉塞
- 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
- IO : 非同期型問合せ応答メッセージ
- 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	mcftlsle 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	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	表示情報を表示します。	標準出力
KFCA10365-I	表示情報を表示します。	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10371-I	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

8

組み込み方法

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

8.1 TP1/NET/X25 の組み込みの流れ

8.2 MCF メイン関数の作成

8.3 定義オブジェクトファイルの生成

8.1 TP1/NET/X25 の組み込みの流れ

TP1/NET/X25 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 を起動するためには、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しておく必要があります。詳細は、「8.2 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 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。

また、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オペレーティングシステムの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、「6. システム定義」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作成

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、「8.3 定義オブジェクトファイルの生成」を参照してください。

8.2 MCF メイン関数の作成

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

TP1/NET/X25 を起動するためには、ユーザが MCF メイン関数をコーディングし、翻訳（コンパイル）および結合（リンケージ）を行って TP1/NET/X25 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、PVC の場合は mcfplx25 コマンドを、VC の場合は mcfplx25vc 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 の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 8-1 および図 8-2 に示します。また、ディレクトリへの組み込み方法を図 8-3 に示します。

なお、このコーディング例を次のファイルで提供しています。

- 図 8-1（ANSI C、C++ の場合）
PVC の場合：/BeTRAN/examples/mcf/X25/cmlib/ansi/com1.c
VC の場合：/BeTRAN/examples/mcf/X25/cmlib/ansi/com2.c
- 図 8-2（K&R 版 C の場合）
PVC の場合：/BeTRAN/examples/mcf/X25/cmlib/c/com1.c
VC の場合：/BeTRAN/examples/mcf/X25/cmlib/c/com2.c

図 8-1 MCF メイン関数のコーディング概要（ANSI C、C++ の場合）

```

#include <dcmx25m.h>*                               /*TP1/NET/X25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*/ 3.
extern dcmcf_uoc_t dcmcf_uoctbl; /*UOCテーブルextern宣言   */
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 */
    dc_mcf_svstart(); /*スタート関数呼び出し   */ 5.
    return 0;
}

```

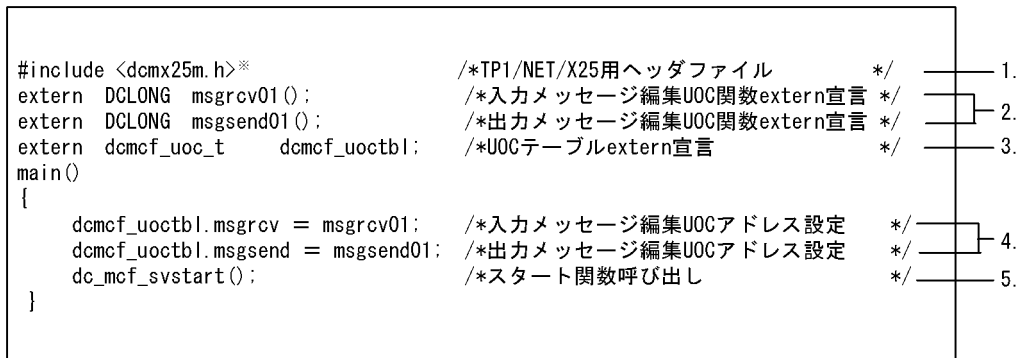
8. 組み込み方法

注

PVC の場合のコーディングです。VC の場合は、次のようにコーディングしてください。

```
#include <dcmx2p.h>
```

図 8-2 MCF メイン関数のコーディング概要 (K & R 版 C の場合)



注

PVC の場合のコーディングです。VC の場合は、次のようにコーディングしてください。

```
#include <dcmx2p.h>
```

1. TP1/NET/X25 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。これらの UOC を使用する場合だけ、コーディングしてください。
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。2. の UOC を使用する場合だけ、コーディングしてください。

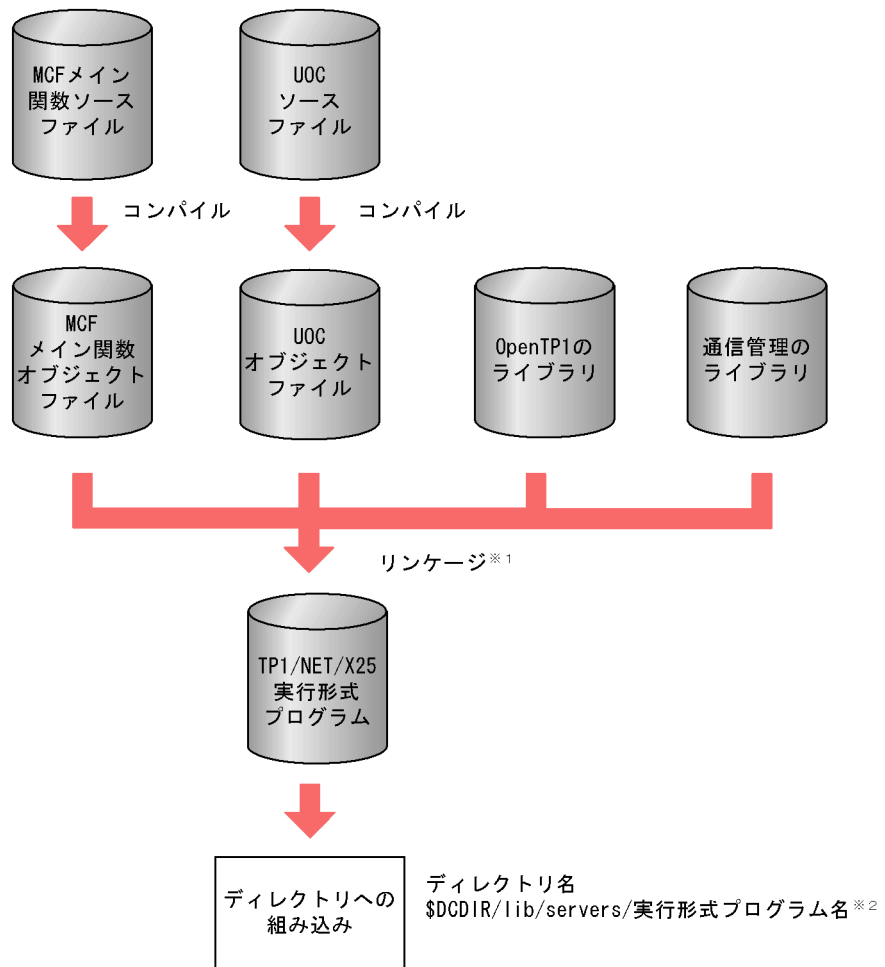
4. 各 UOC 関数のアドレスを、次に示すシステム提供変数に設定します。

```
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*/
```

これらの UOC を使用する場合だけ、コーディングしてください。

5. スタート関数を呼び出します。必ずコーディングしてください。

図 8-3 MCF メイン関数のディレクトリへの組み込み方法



注 1
PVC の場合は mcfplx25 コマンド，VC の場合は mcfplx25vc コマンドでリンケージします。

注 2
TP1/NET/X25 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。

8.3 定義オブジェクトファイルの生成

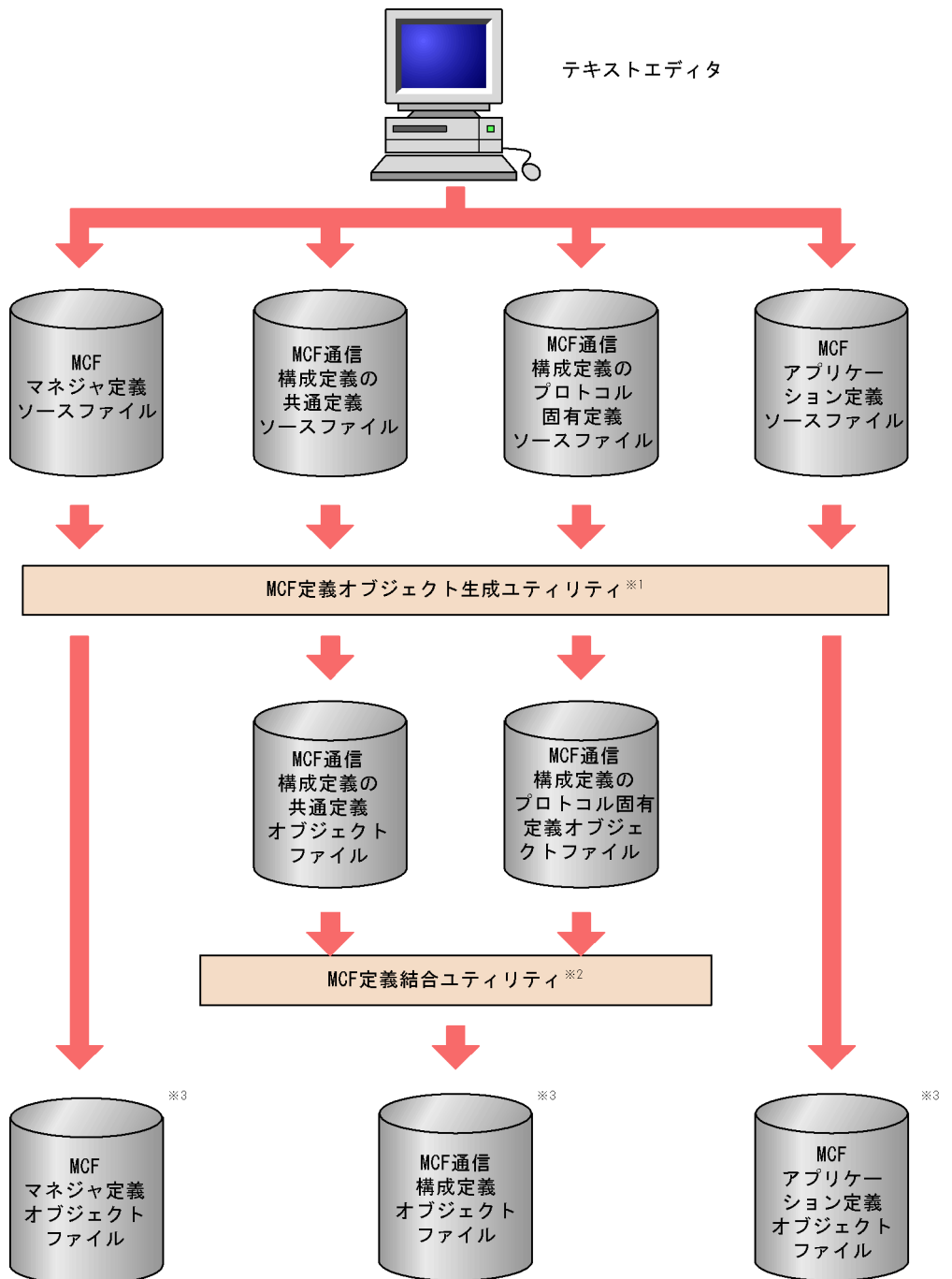
定義オブジェクトファイルを次の手順で生成します。

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おそれがあるためご注意ください。

1. オペレーティングシステムのテキストエディタを使用して、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 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 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ます。

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 8-4 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。

8. 組み込み方法

```
mcfXXXX -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
```

mcfXXXX は、ソースファイルごとに異なります。

- mcfmngnr : 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
- mcfx25 : MCF 通信構成定義のプロトコル (TP1/NET/X25) 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 mcfx25 コマンドについては「6. システム定義」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。

```
mcflink -i 共通定義オブジェクトファイル名  
        TP1/NET/X25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。

9

PVC の障害対策

この章では、PVC を使用した場合の障害時の処理とユーザの対策，および障害発生時の処理の流れについて説明します。

9.1 障害の種類と対応処理 (PVC)

9.2 コネクション障害 (PVC)

9.3 スケジュール関係の障害 (PVC)

9.4 UAP 障害 (PVC)

9.1 障害の種類と対応処理 (PVC)

運用中に障害が発生すると、TP1/NET/X25 はシステムを回復します。このとき、システム定義の指定によって、MCF イベント処理用 MHP も起動できます。

PVC の場合の TP1/NET/X25 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 9-1 コネクション障害と対応処理 (PVC)

障害内容	TP1/NET/X25 の処理 (PVC)	ユーザの処理
コネクション確立時障害 (回復不可能)	(1) コネクション障害を通知するメッセージログ (KFCA14200-W) を出力します。 (2) コネクション閉塞を通知するメッセージログ (KFCA14002-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、mcftactcn コマンドを入力します。
コネクション確立時障害 (回復可能)	(1) コネクション障害を通知するメッセージログ (KFCA14200-W) を出力します。 (2) 回復動作を通知するメッセージログ (KFCA14201-I) を出力します。 (3) 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。
コネクション切断障害 (回復不可能)	(1) コネクション障害を通知するメッセージログ (KFCA14200-W) を出力します。 (2) コネクション解放を通知するメッセージログ (KFCA14001-I) を出力します。 (3) コネクション閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 (4) CERREVT (コネクション切断・回復不可能) を起動します。	障害の要因を取り除いたあと、mcftactcn コマンドを入力します。
コネクション切断障害 (回復可能)	(1) コネクション障害を通知するメッセージログ (KFCA14200-W) を出力します。 (2) コネクション解放を通知するメッセージログ (KFCA14001-I) を出力します。 (3) CERREVT (コネクション切断・回復可能) を起動します。 自動起動の場合 (4) 回復動作を通知するメッセージログ (KFCA14201-I) を出力します。 (5) リトライタイマの指定によって、一定時間経過後、再確立処理をします。	手動起動の場合、(3) のあとで mcftactcn コマンドを入力します。

障害内容	TP1/NET/X25 の処理 (PVC)	ユーザの処理
受信バッファオーバーフロー	<ol style="list-style-type: none"> (1) 受信メッセージ破棄, 受信バッファオーバーフローを通知するメッセージログ (KFCA14203-W) を出力します。 (2) コネクション解放を通知するメッセージログ (KFCA14001-I) を出力します。 (3) コネクション閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 (4) CERREVT (受信バッファオーバーフロー) を起動します。 	システム定義を修正します。
プロトコルアドレス不正	<ol style="list-style-type: none"> (1) プロトコルアドレス不正を通知するメッセージログ (KFCA14204-W) を出力します。 (2) コネクション閉塞を通知するメッセージログ (KFCA14002-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。 	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"> (1) ローカルメモリ不足を通知するメッセージログ (KFCA10618-E) を出力します。 (2) グループバッファ取得失敗を通知するメッセージログ (KFCA14005-E) を出力し, 一定の時間内にほかで使用中のバッファの解放を待ちます。 	ありません。
	時間内に解放された場合 <ol style="list-style-type: none"> (3) 処理を続行します。 	
	時間内に解放されなかった場合 <ol style="list-style-type: none"> (3) 受信メッセージ破棄・受信バッファ不足を通知するメッセージログ (KFCA14203-W) を出力します。 (4) コネクション解放を通知するメッセージログ (KFCA14001-I) を出力します。 (5) コネクション閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 (6) 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 (KFCA14200-W) を出力します。 (2) コネクション解放を通知するメッセージログ (KFCA14001-I) を出力します。 (3) コネクション閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 (4) CERREVT (コネクション切断) を起動します。 (5) メッセージは出力キューに残し, 再開始時に再送されます。 	障害の要因を取り除いたあと, mcftacten コマンドを入力します。
リセット受信	<ol style="list-style-type: none"> (1) リセット受信を通知するメッセージログ (KFCA14205-W) を出力します。 (2) 受信中のメッセージを破棄し, 送信中のメッセージをリセットします。 	メッセージの通番を合わせて再送処理をします。

9. PVC の障害対策

障害内容	TP1/NET/X25 の処理 (PVC)	ユーザの処理
	リセット通知あり指定の場合 (3) UAP を起動します。	
	リセット通知なし指定の場合 (3) 何もしません。	
	コネクション閉塞指定の場合 (3) コネクション解放を通知するメッセージログ (KFCA14001-I) を出力します。 (4) コネクション閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 (5) CERREVT (リセット受信) を起動し、リトライタイマの指定によって、一定時間経過後、再確立処理をします。	

(2) 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 9-2 受信スケジュール関係の障害と対応処理 (PVC)

障害内容	TP1/NET/X25 の処理 (PVC)	ユーザの処理
ユーザアプリケーション名称未定義	(1) ERREVT1 を起動します。	システム定義を修正します。
MCF イベントの名称未定義 • ERREVT x • C x x x EVT	(1) 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	(1) ERREVT2 を起動します。 (2) 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。	ありません。
入力キュー書き込み障害	(1) ERREVT2 を起動します。 (2) 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。	ありません。
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	(1) 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。 (2) 論理端末を閉塞します。 (3) 論理端末閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 (4) CERREVT (UOC 障害) を起動します。	ありません。

障害内容	TP1/NET/X25 の処理 (PVC)	ユーザの処理
	<p>メッセージ送信中の場合</p> <p>(5) 送信中のメッセージを取り出し前の状態に戻します。</p> <p>(6) コネクション解放を通知するメッセージログ (KFCA14001-I) を出力します。</p> <p>(7) コネクション閉塞を通知するメッセージログ (KFCA14002-E) を出力します。</p> <p>(5) CERREVT (コネクション強制解放) を起動します。</p>	
アプリケーション名称形式不正	<p>(1) 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。</p> <p>(2) ERREVT1 を起動します。</p>	ありません。

(3) 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 9-3 送信スケジュール関係の障害と対応処理 (PVC)

障害内容	TP1/NET/X25 の処理 (PVC)	ユーザの処理
出力キュー読み込み障害	<p>(1) 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。</p> <p>(2) CERREVT (出力キュー読み込み障害) を起動します。</p> <p>(3) 該当するメッセージを消去します。</p> <p>(4) 論理端末を閉塞します。</p> <p>(5) 論理端末閉塞を通知するメッセージログ (KFCA14004-I) を出力します。</p>	<ul style="list-style-type: none"> コネクションを解放します。 相手システムへ出力キュー障害を通知します (send)。
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	<p>(1) 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。</p> <p>(2) CERREVT (UOC 障害) を起動します。</p> <p>(3) 該当するメッセージを消去します。</p> <p>(4) 論理端末を閉塞します。</p> <p>(5) 論理端末閉塞を通知するメッセージログ (KFCA14004-I) を出力します。</p>	<ul style="list-style-type: none"> コネクションを解放します。 相手システムへ出力キュー障害を通知します (send)。
メッセージ消し込み障害	<p>(1) 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。</p> <p>(2) 処理を続行します。</p>	ありません。

9. PVC の障害対策

障害内容	TP1/NET/X25 の処理 (PVC)	ユーザの処理
送信バッファオーバーフロー	<ol style="list-style-type: none"> (1) 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。 (2) 論理端末閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 (3) 論理端末を閉塞します。 (4) 該当するメッセージを消去します。 (5) CERREVT (送信バッファオーバーフロー) を起動します。 	<ul style="list-style-type: none"> • コネクションを解放します。 • システム定義, または UAP を修正します。
送信バッファ不足	<ol style="list-style-type: none"> (1) ローカルメモリ不足を通知するメッセージログ (KFCA10618-E) を出力します。 (2) 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。 (3) 該当するメッセージを取り出し前の状態に戻します。 (4) 論理端末閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 (5) 論理端末を閉塞します。 (6) CERREVT (送信バッファ不足) を起動します。 	<ul style="list-style-type: none"> • 一定時間経過後, mftactle コマンドを入力して論理端末の閉塞を解除し, メッセージ再送信します。 • 送信バッファの面数を見直します。

(4) UAP 障害

表 9-4 UAP 障害と対応処理 (PVC)

障害内容	TP1/NET/X25 の処理 (PVC)	ユーザの処理
メッセージ受信前の UAP 異常終了	(1) ERREVT2 を起動します。	相手システムへ UAP 異常終了を通知します (send)。
メッセージ受信後の UAP 異常終了	(1) ERREVT3 を起動します。	相手システムへ UAP 異常終了を通知します (send)。

(5) MCF 障害

表 9-5 MCF 障害と対応処理 (PVC)

障害内容	TP1/NET/X25 の処理 (PVC)	ユーザの処理
内部論理矛盾 他プログラムプログラク トとのインタフェース エラー	<ol style="list-style-type: none"> (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) コネクション閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 (3) コネクション障害の場合, CERREVT を起動します。 	ありません。

9.2 コネクション障害 (PVC)

コネクションが切断された場合、TP1/NET/X25 はコネクションを再確立します。通信管理が管理している通信機器に障害が発生した場合も、同様にコネクション障害として処理します。TP1/NET/X25 によって自動回復できない障害の場合は、障害の原因を取り除いたあと、運用コマンド (mcfactcn) を入力してください。

(1) 一方送信メッセージ送信時のコネクション障害

相手システムへ一方送信メッセージ送信中に障害が発生した場合、メッセージはロールバックされ、コネクション再確立時に再送信されます。

また、相手システムへ一方送信メッセージ送信後に障害が発生した場合、メッセージは二重に送信されます。このとき相手システムの UAP は、メッセージに通番を付けることで、二重送信をチェックします。

一方送信メッセージ送信中のコネクション障害の例を図 9-1 に、一方送信メッセージ送信後のコネクション障害の例を図 9-2 に示します。

図 9-1 一方送信メッセージ送信中のコネクション障害 (PVC)

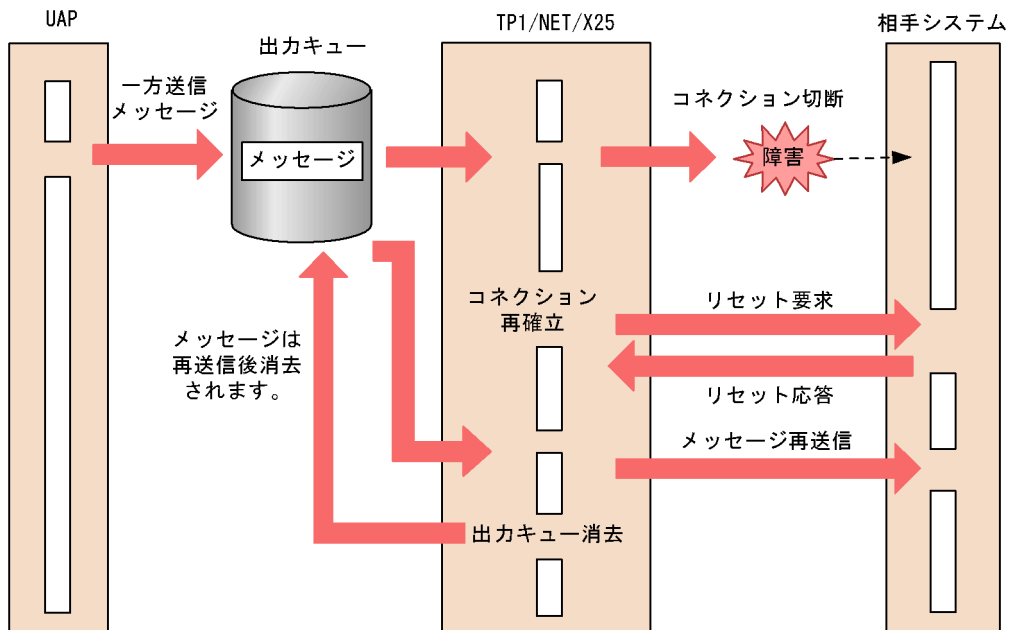
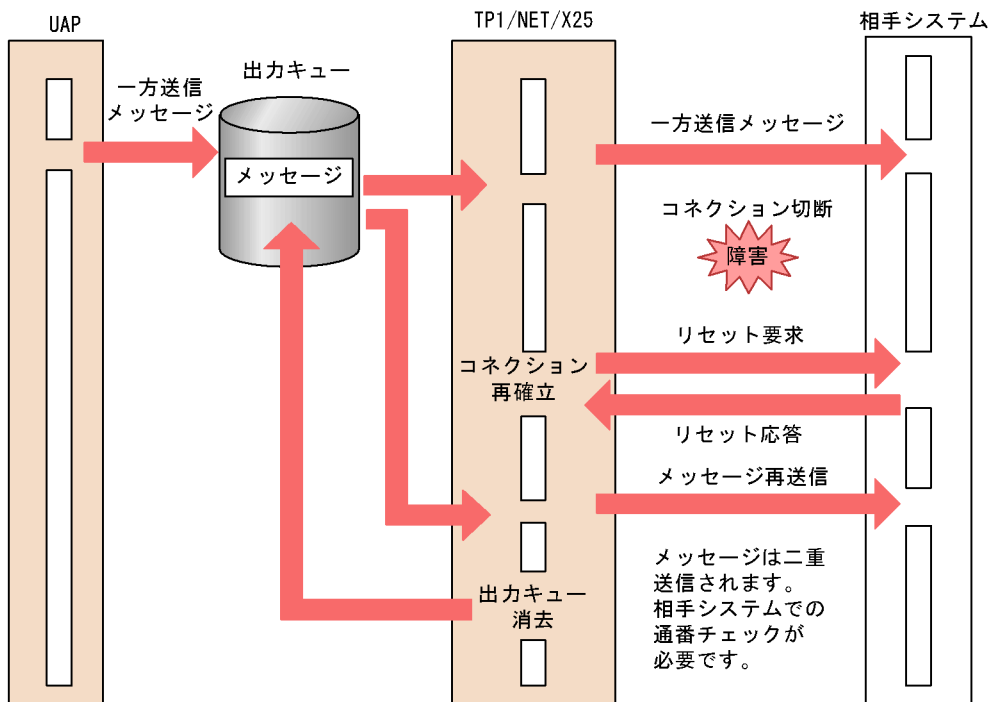


図 9-2 一方送信メッセージ送信後のコネクション障害 (PVC)



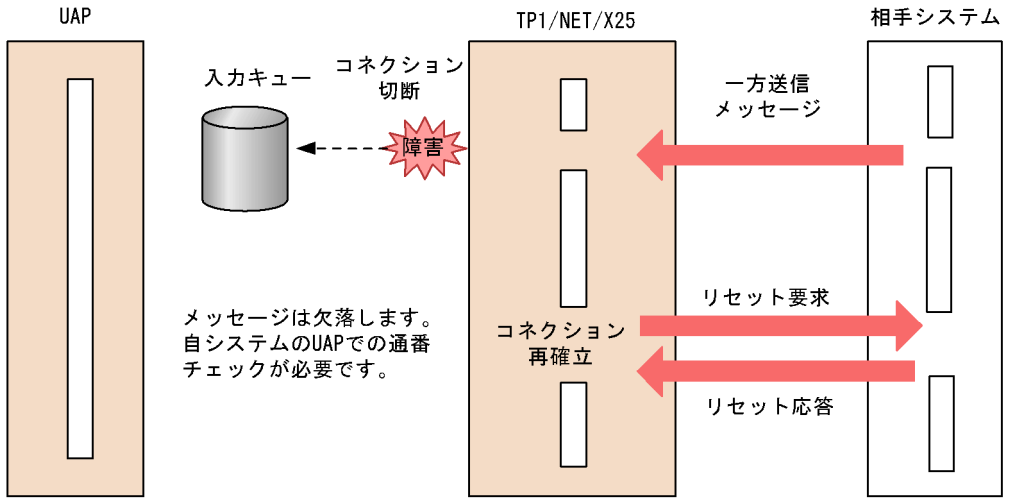
(2) 一方送信メッセージ受信時のコネクション障害

(a) 入力キュー書き込み前

メッセージは欠落します。このとき自システムの UAP は、メッセージに通番を付けることで、欠落しているかどうかをチェックします。

一方送信メッセージ書き込み前のコネクション障害の例を次の図に示します。

図 9-3 一方送信メッセージ書き込み前のコネクション障害 (PVC)

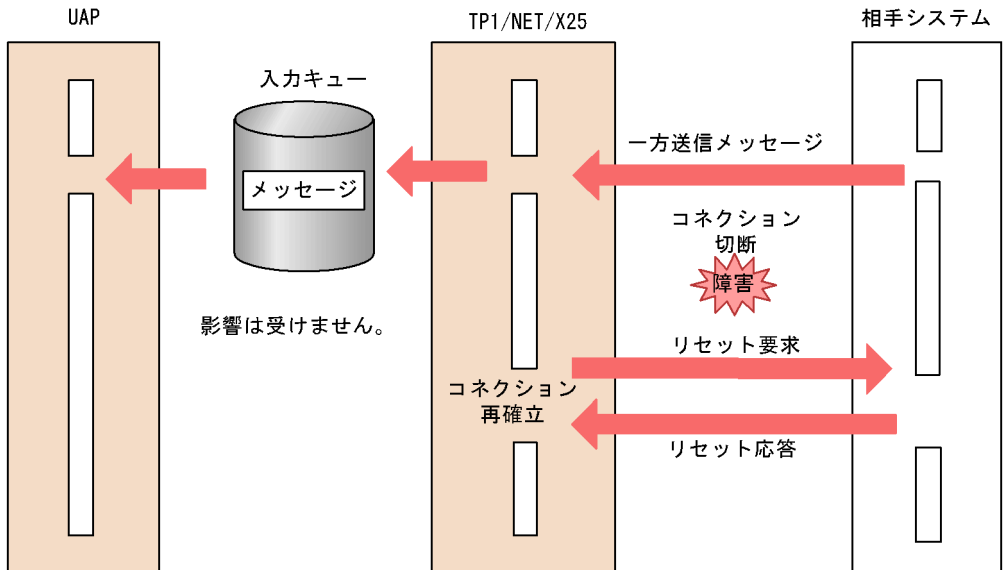


(b) 入力キュー書き込み後

メッセージは影響を受けません。

一方送信メッセージ書き込み後のコネクション障害の例を次の図に示します。

図 9-4 一方送信メッセージ書き込み後のコネクション障害 (PVC)



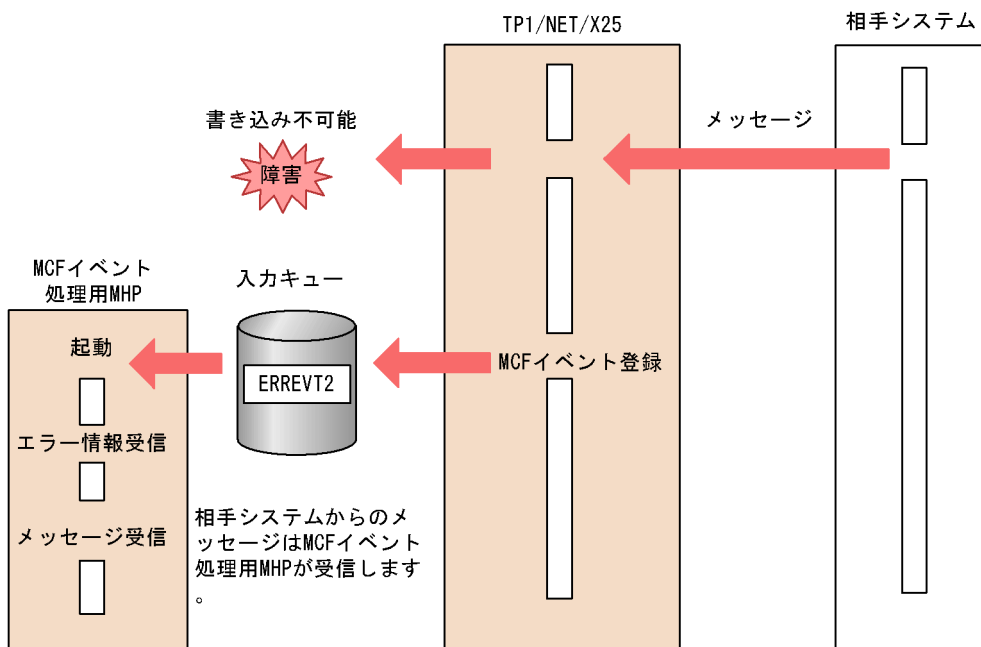
9.3 スケジュール関係の障害（PVC）

（1）入力キュー障害

メッセージを入力キューに書き込むときに障害が発生した場合、TP1/NET/X25 は入力キューに MCF イベント ERREVT2 を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

入力キュー障害の例を次の図に示します。

図 9-5 入力キュー障害（PVC）

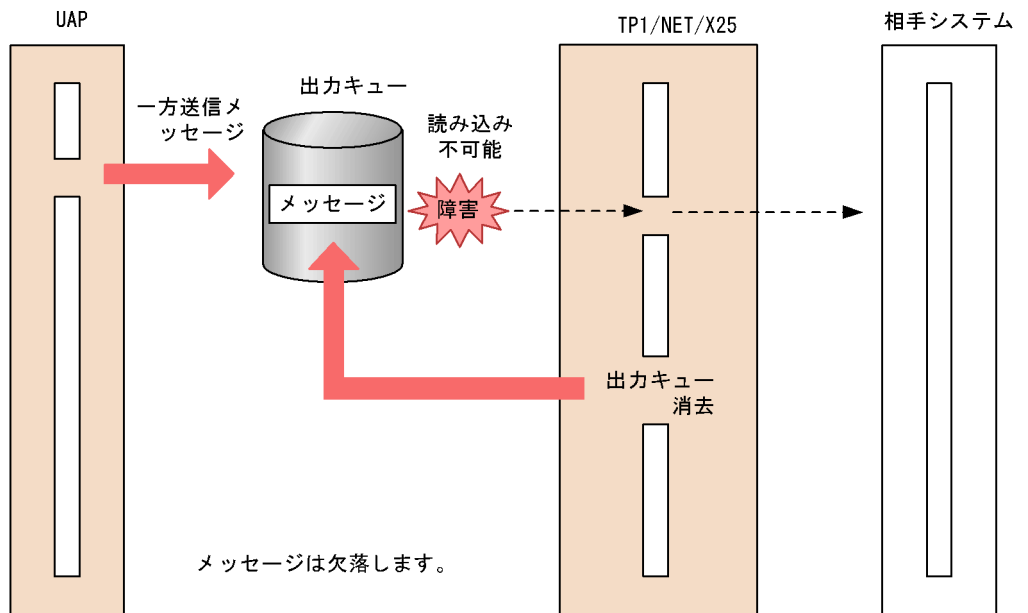


（2）出力キュー障害

メッセージを出力キューから読み込むときに障害が発生した場合、メッセージは欠落します。

出力キュー障害の例を次の図に示します。

図 9-6 出力キュー障害 (PVC)



9.4 UAP 障害 (PVC)

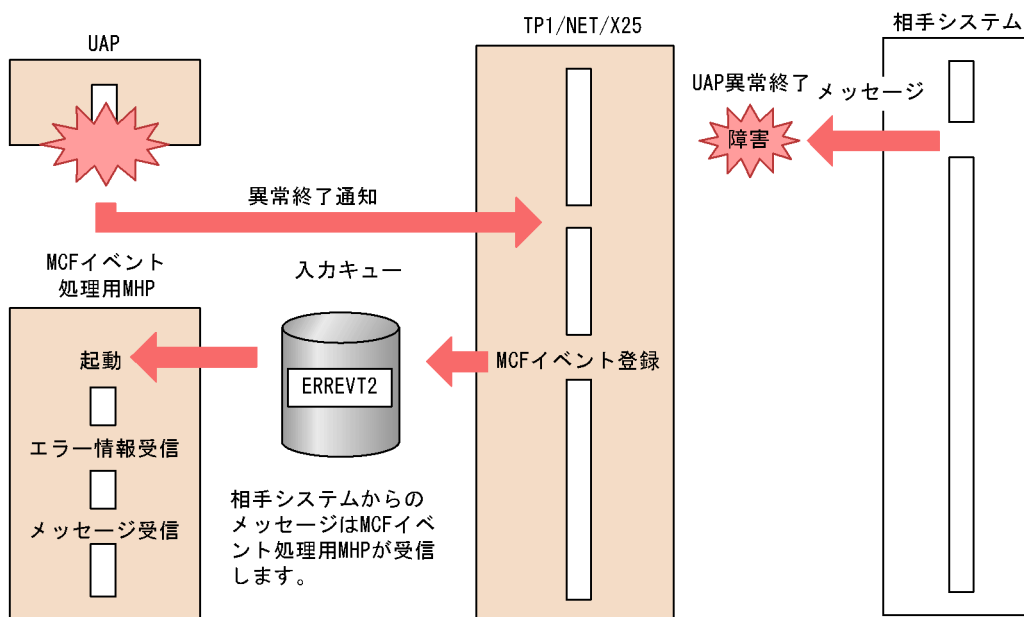
UAP が異常終了した場合、TP1/NET/X25 は入力キューに MCF イベントを登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

(1) メッセージ受信前の UAP 異常終了

TP1/NET/X25 は、入力キューに MCF イベント ERREVT2 を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

メッセージ受信前の UAP 異常終了の例を次の図に示します。

図 9-7 メッセージ受信前の UAP 異常終了 (PVC)

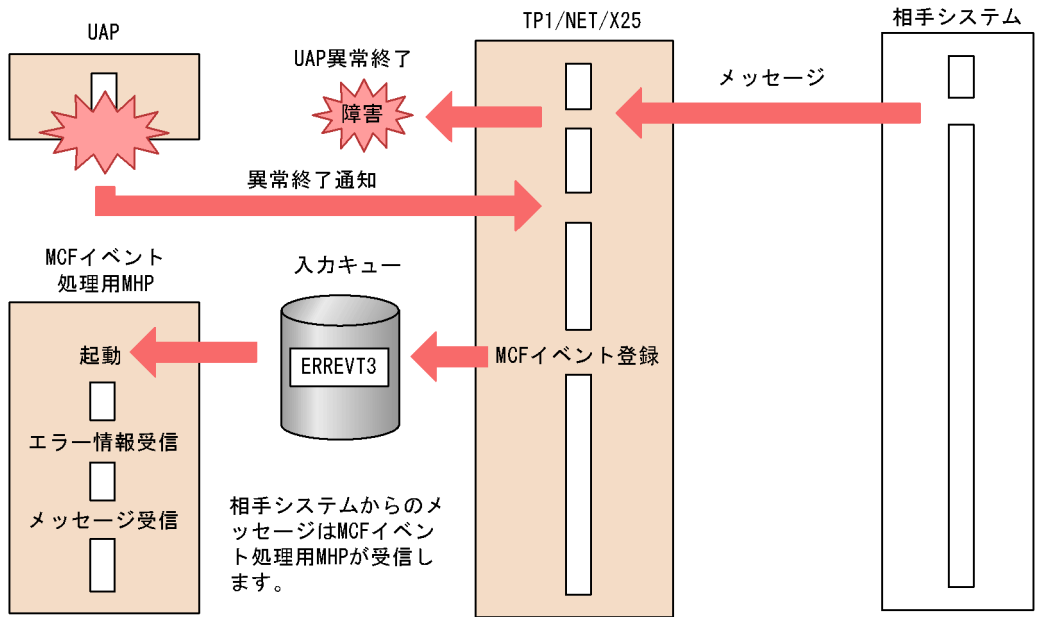


(2) メッセージ受信後の UAP 異常終了

TP1/NET/X25 は、入力キューに MCF イベント ERREVT3 を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

メッセージ受信後の UAP 異常終了の例を次の図に示します。

図 9-8 メッセージ受信後の UAP 異常終了 (PVC)



10 VC の障害対策

この章では、VC を使用した場合の障害時の処理とユーザの対策、および障害発生時の処理の流れについて説明します。

10.1 障害の種類と対応処理 (VC)

10.2 コネクション障害 (VC)

10.3 スケジュール関係の障害 (VC)

10.4 UAP 障害 (VC)

10.1 障害の種類と対応処理 (VC)

運用中に障害が発生すると、TP1/NET/X25 はシステムを回復します。このとき、システム定義の指定によって、MCF イベント処理用 MHP も起動できます。

VC の場合の TP1/NET/X25 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 10-1 コネクション障害と対応処理 (VC)

障害内容	TP1/NET/X25 の処理 (VC)	ユーザの処理
コネクション確立時障害 (回復不可能)	(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 (コネクション障害) を起動します。	障害の要因を取り除いたあと、mcfactcn コマンドを入力します。
着呼側コネクション確立時障害	(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。	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。
コネクション確立時障害 (回復可能)	(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 (3) 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。 (4) 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。 (5) CERREVT (コネクション障害) を起動します。	障害の要因を取り除いたあと、mcfactcn コマンドを入力します。
コネクション切断	(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) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (3) CERREVT (コネクション切断・回復不可能)、または CCLSEVT を起動します。	障害の要因を取り除いたあと、mcfactcn コマンドを入力します。
メッセージ送信時障害 (回復不可能)	(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) CERREVT (送信障害) を起動します。 (3) 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。 (4) CERREVT (コネクション障害) を起動します。 (5) メッセージは出力キューに残し、再開時に再送されます。	障害の要因を取り除いたあと、mcfactcn コマンドを入力します。

障害内容	TP1/NET/X25 の処理 (VC)	ユーザの処理
メッセージ送信時リセットパケット受信	(1) 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。 (2) 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。 (3) CERREVT (送信障害) を起動します。 (4) 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。 (5) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (6) CERREVT (コネクション障害) を起動します。 (7) 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	(1) 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。 (2) 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。 (3) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (4) CERREVT (受信バッファオーバーフロー) を起動します。 (5) 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。
プロトコルアドレス不正	(1) 下位層障害を通知するメッセージログ (KFCA16913-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 (コネクション障害) を起動します。	システム定義を修正します。
受信バッファ不足	(1) 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。 (2) 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。 (3) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (4) 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	(1) リセット受信を通知するメッセージログ (KFCA16916-E) を出力します。 (2) 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。 (3) コネクションを解放します。 (4) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (5) CERREVT (コネクション障害) を起動します。 (6) 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) 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 10-2 受信スケジュール関係の障害と対応処理 (VC)

障害内容	TP1/NET/X25 の処理 (VC)	ユーザの処理
ユーザアプリケーション名称未定義	<ol style="list-style-type: none"> (1) ERREVT1 を起動します。 (2) 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。 (3) 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。 (4) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (5) CERREVT (コネクション障害) を起動します。 	システム定義を修正します。
MCF イベントの名称未定義 • ERREVT x • C x x x EVT	<ol style="list-style-type: none"> (1) 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"> (1) 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。 (2) ERREVT2 を起動します。 (3) 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。 (4) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (5) 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。 (6) CERREVT (コネクション障害) 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, 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。
入力キュー書き込み障害	<ol style="list-style-type: none"> (1) ERREVT2 を起動します。 (2) 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。 (3) 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。 (4) 受信メッセージを破棄します。 (5) 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。 (6) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (7) CERREVT (コネクション障害) 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, 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。

障害内容	TP1/NET/X25 の処理 (VC)	ユーザの処理
入力メッセージ編集 UOC エラー	(1) 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。 (2) CERREVT (UOC 障害) を起動します。 (3) 受信メッセージを破棄します。 (4) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (5) 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。 (6) CERREVT (コネクション障害) を起動します。	入力メッセージ編集 UOC 処理を見直します。
アプリケーション名形式不正	(1) 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。 (2) ERREVT1 を起動します。 (3) 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。 (4) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (5) CERREVT (コネクション障害) を起動します。	システム定義を修正します。

(3) 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 10-3 送信スケジュール関係の障害と対応処理 (VC)

障害内容	TP1/NET/X25 の処理 (VC)	ユーザの処理
出力キュー読み込み障害	(1) 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。 (2) CERREVT (出力キュー読み込み障害) を起動します。 (3) 論理端末を閉塞します。 (4) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (5) 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。 (6) CERREVT (コネクション障害) 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, mcfactcn コマンドを入力します。 • 相手からの確立要求を待ちます。

10. VC の障害対策

障害内容	TP1/NET/X25 の処理 (VC)	ユーザの処理
出力メッセージ編集 UOC エラー	<ol style="list-style-type: none"> (1) 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。 (2) 該当するメッセージを消去します。 (3) 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。 (4) 論理端末を閉塞します。 (5) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (6) CERREVT (UOC 障害) を起動します。 (7) 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。 (8) CERREVT (コネクション障害) を起動します。 	出力メッセージ編集 UOC 処理を見直します。
メッセージ消し込み障害	<ol style="list-style-type: none"> (1) 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。 (2) 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"> (1) ローカルメモリ不足を通知するメッセージログ (KFCA10618-E) を出力します。 (2) 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。 (3) 該当するメッセージを取り出し前の状態に戻します。 (4) 送信メッセージ中断を通知するメッセージログ (KFCA10608-W) を出力します。 (5) 論理端末を閉塞します。 (6) 論理端末閉塞を通知するメッセージログ (KFCA16851-I) を出力します。 (7) CERREVT (送信バッファ不足) を起動します。 (8) 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。 (9) CERREVT (コネクション障害) を起動します。 	送信バッファ数を見直します。

(4) UAP 障害

表 10-4 UAP 障害と対応処理 (VC)

障害内容	TP1/NET/X25 の処理 (VC)	ユーザの処理
メッセージ受信前の UAP 異常終了	(1) ERREVT2 を起動します。	UAP を見直します。
メッセージ受信後の UAP 異常終了	(1) ERREVT3 を起動します。	UAP を見直します。

(5) MCF 障害

表 10-5 MCF 障害と対応処理 (VC)

障害内容	TP1/NET/X25 の処理 (VC)	ユーザの処理
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)<10189-E>(障害情報取得) (2) 内部論理矛盾を通知するメッセージログ (KFCA16897-E, KFCA16899-E, または KFCA13211-E) を出力します。	保守情報 (\$DCDIR/ spool ディレクトリ以 下) を退避します。
	KFCA16897-E の場合 (3) 処理を続行します。	
	KFCA16899-E または KFCA13211-E の場合 (3) MCF を異常終了します。	

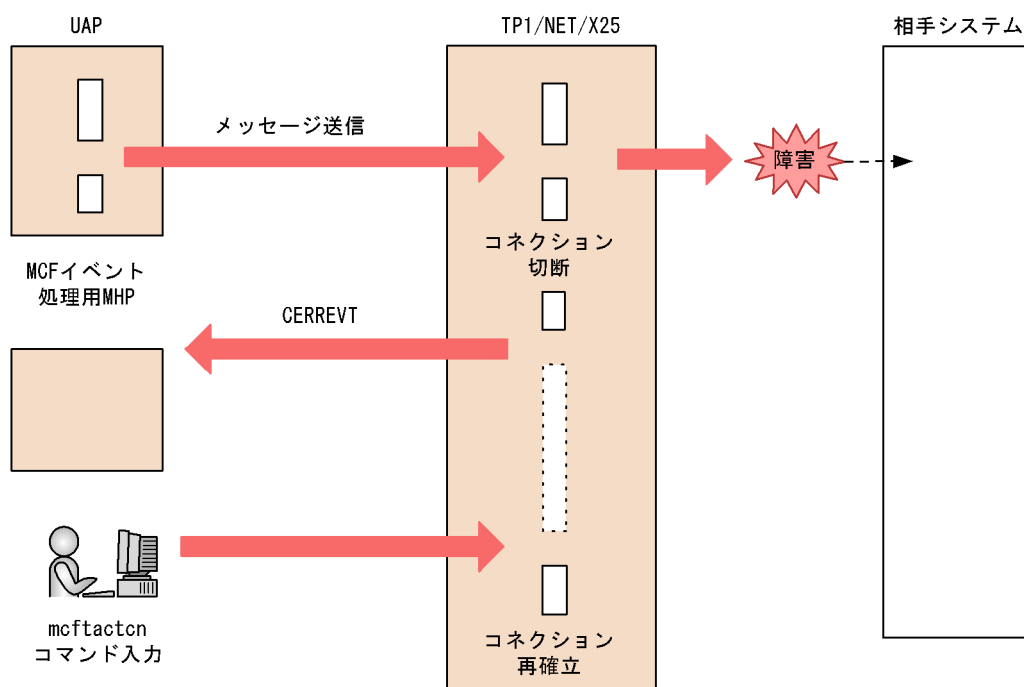
10.2 コネクション障害（VC）

TP1/NET/X25 は、通信管理が管理している通信機器に障害が発生した場合、コネクション障害として処理をします。TP1/NET/X25 によって自動回復できない場合、障害の原因を取り除いたあと、運用コマンド（`mcftactcn`）を入力し、コネクションの再確立をしてください。

また、メッセージ送受信中にコネクション障害が発生した場合、コネクションは切断されます。コネクションが切断されると、送受信中のメッセージは消去される場合があります。したがって、メッセージは通番などで管理するようにしてください。

コネクション障害の例を次の図に示します。

図 10-1 コネクション障害（VC）



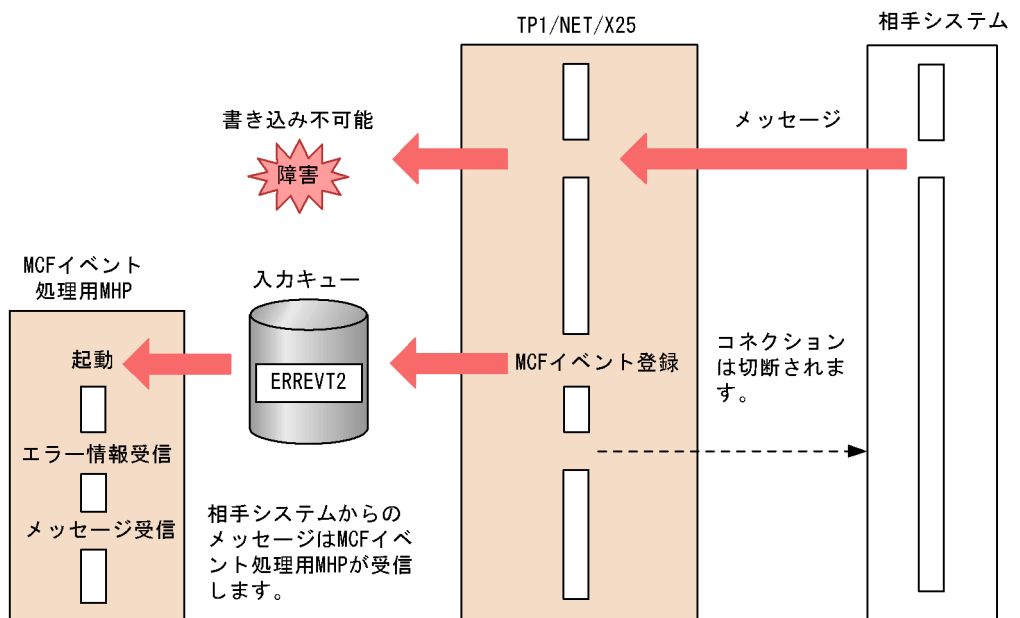
10.3 スケジュール関係の障害（VC）

（1）入力キュー障害

メッセージを入力キューに書き込むときに障害が発生した場合、TP1/NET/X25 は入力キューに MCF イベント ERREVT2 を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

入力キュー障害の例を次の図に示します。

図 10-2 入力キュー障害（VC）

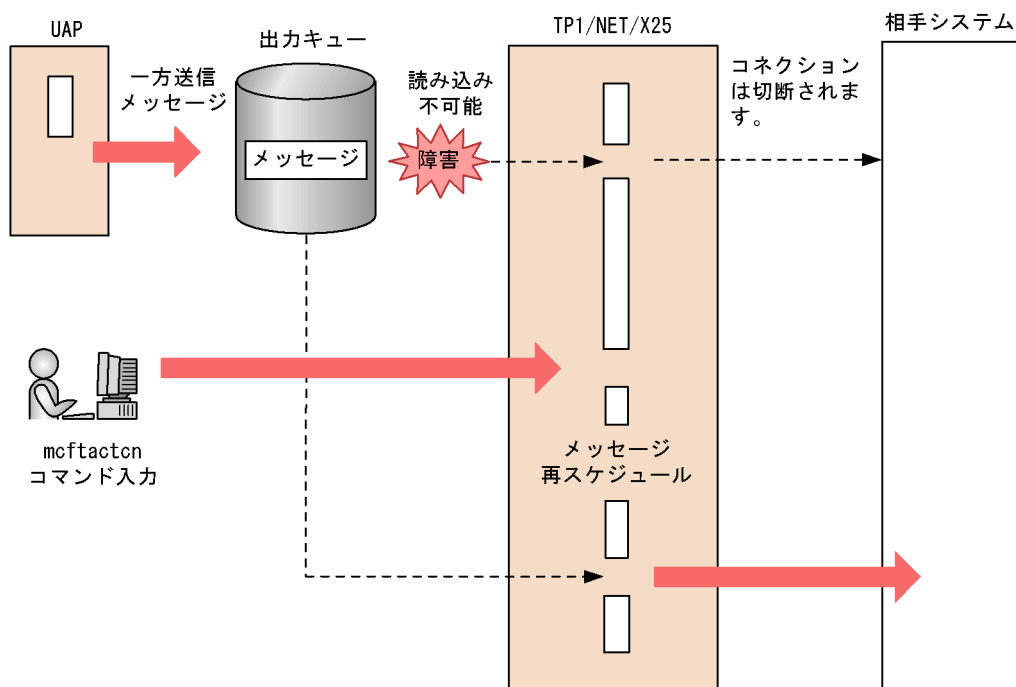


（2）出力キュー障害

メッセージを出力キューから読み込むときに障害が発生した場合、コネクションは切断し、メッセージは再スケジュールされます。

出力キュー障害の例を次の図に示します。

図 10-3 出力キュー障害 (VC)



10.4 UAP 障害 (VC)

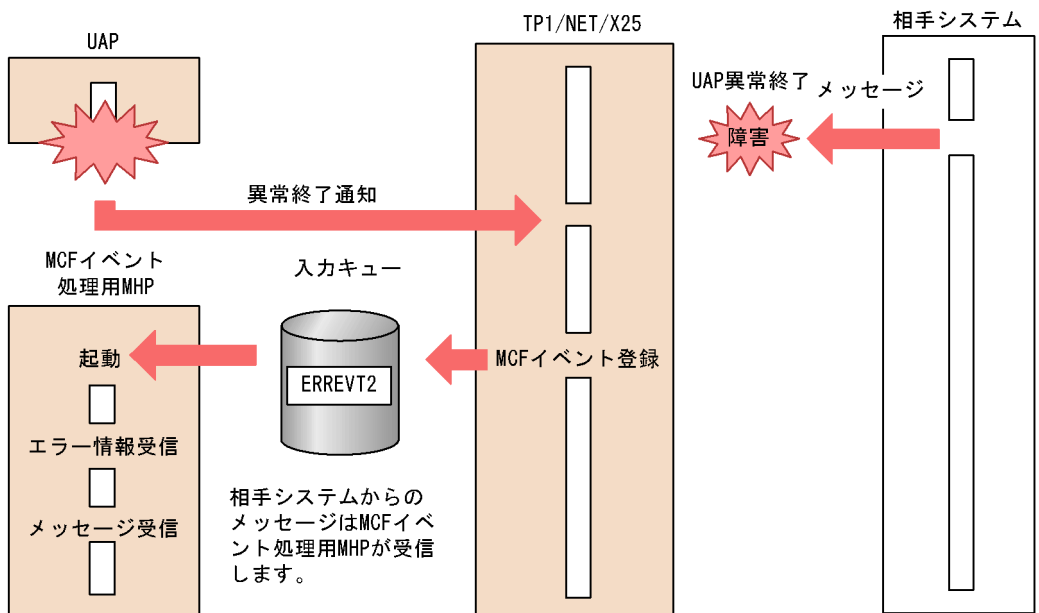
UAP が異常終了した場合、TP1/NET/X25 は入力キューに MCF イベントを登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

(1) メッセージ受信前の UAP 異常終了

TP1/NET/X25 は、入力キューに MCF イベント ERREVT2 を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

メッセージ受信前の UAP 異常終了の例を次の図に示します。

図 10-4 メッセージ受信前の UAP 異常終了 (VC)

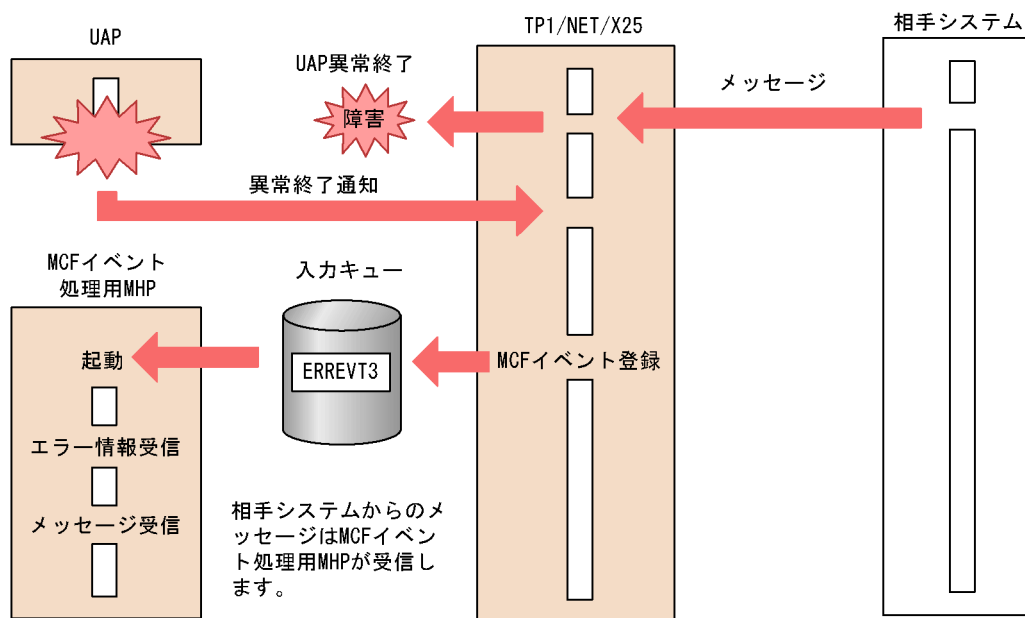


(2) メッセージ受信後の UAP 異常終了

TP1/NET/X25 は、入力キューに MCF イベント ERREVT3 を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

メッセージ受信後の UAP 異常終了の例を次の図に示します。

図 10-5 メッセージ受信後のUAP 異常終了 (VC)



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 旧製品からの移行に関する注意事項

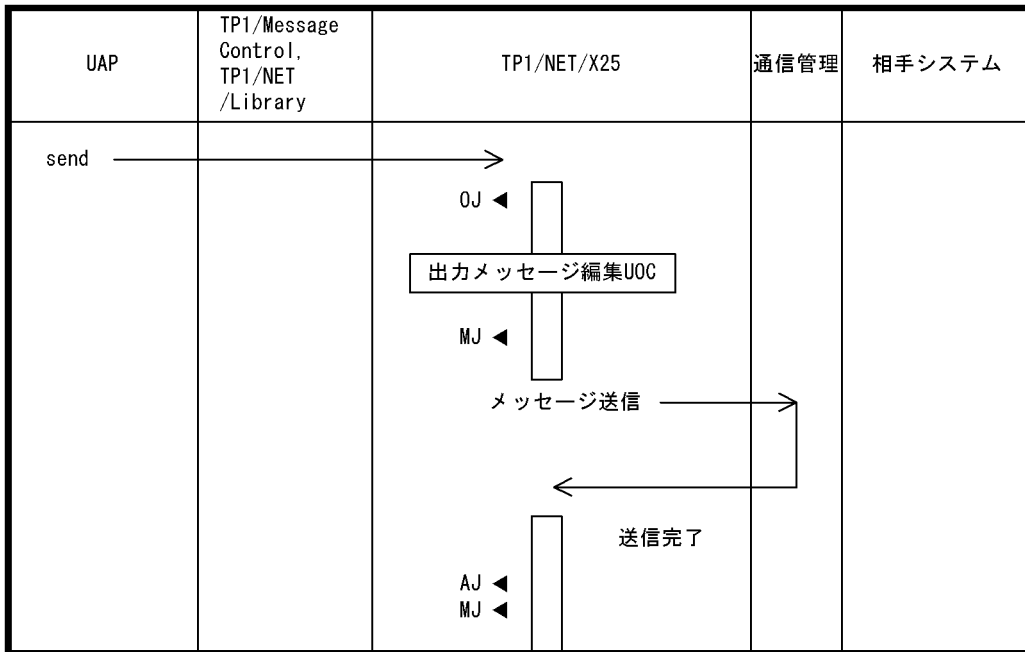
付録 E バージョンアップ時の変更点

付録 F 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、およびジャーナルの取得タイミングは、PVCとVCで共通です。メッセージの送信、受信の流れをパケット分割なしの場合とパケット分割ありの場合に分けて、図 A-1 ~ 図 A-4 に示します。

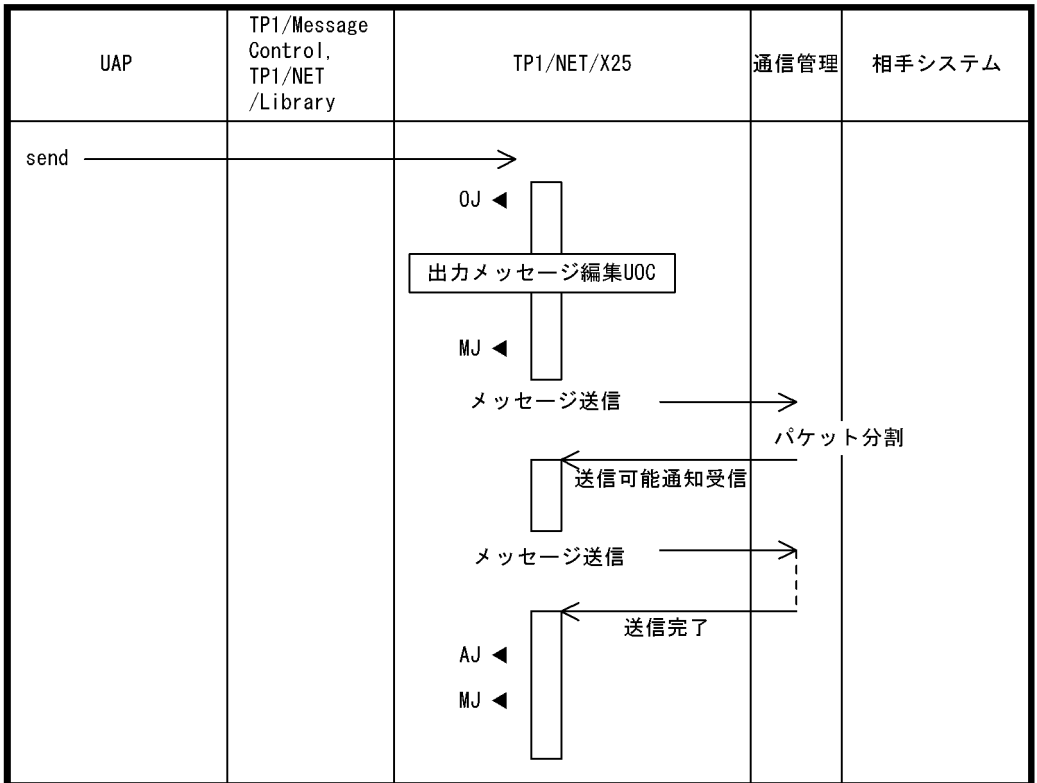
図 A-1 メッセージ送信の処理の流れ (パケット分割なし)



(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-2 メッセージ送信の処理の流れ (パケット分割あり)



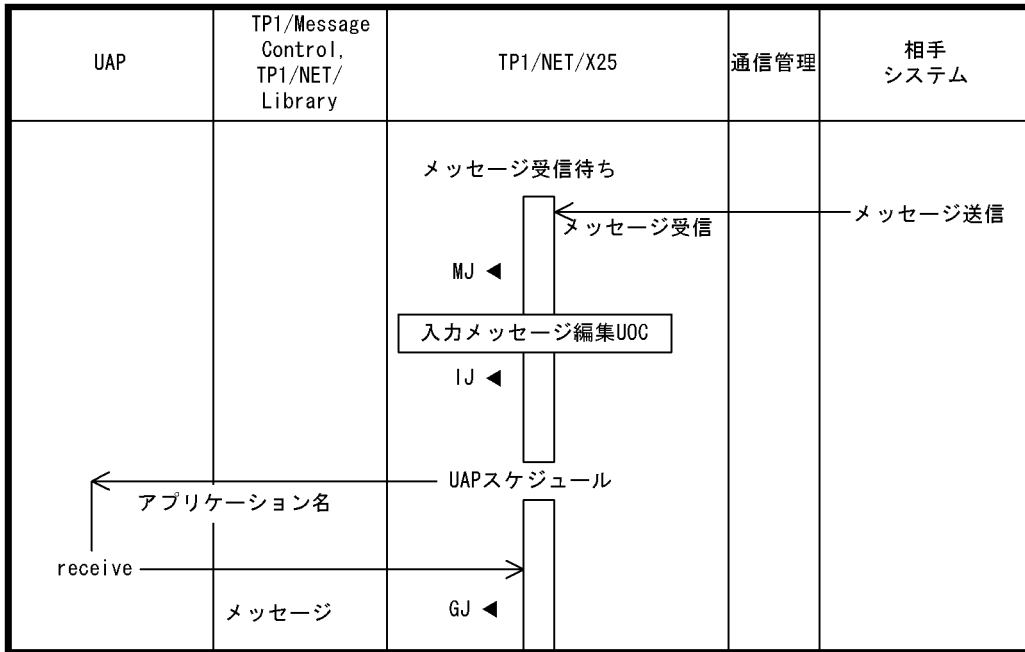
(凡例)

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

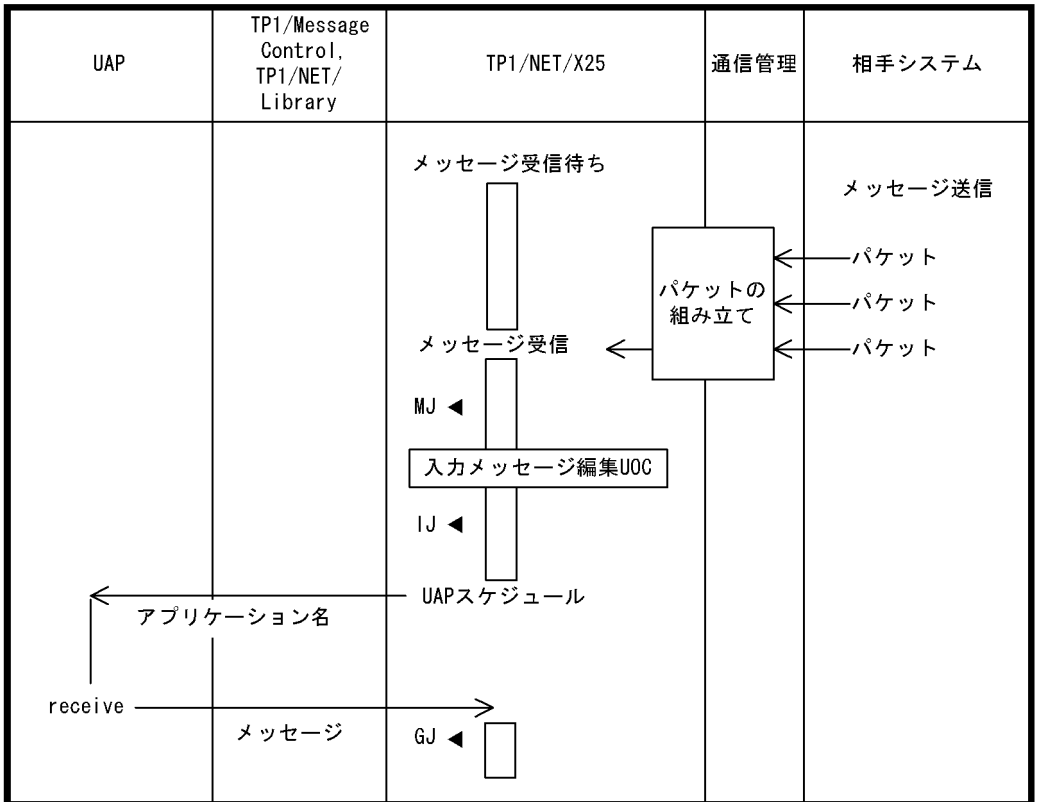
図 A-3 メッセージ受信の処理の流れ (パケット分割なし)



(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-4 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

付録 B 障害発生時の処理の流れ

TP1/NET/X25 の障害発生時の処理の流れを PVC と VC に分けて説明します。

付録 B.1 PVC の障害

PVC の障害発生時の処理の流れを図 B-1 ~ 図 B-4 に示します。

図 B-1 コネクション確立時の障害 (PVC)

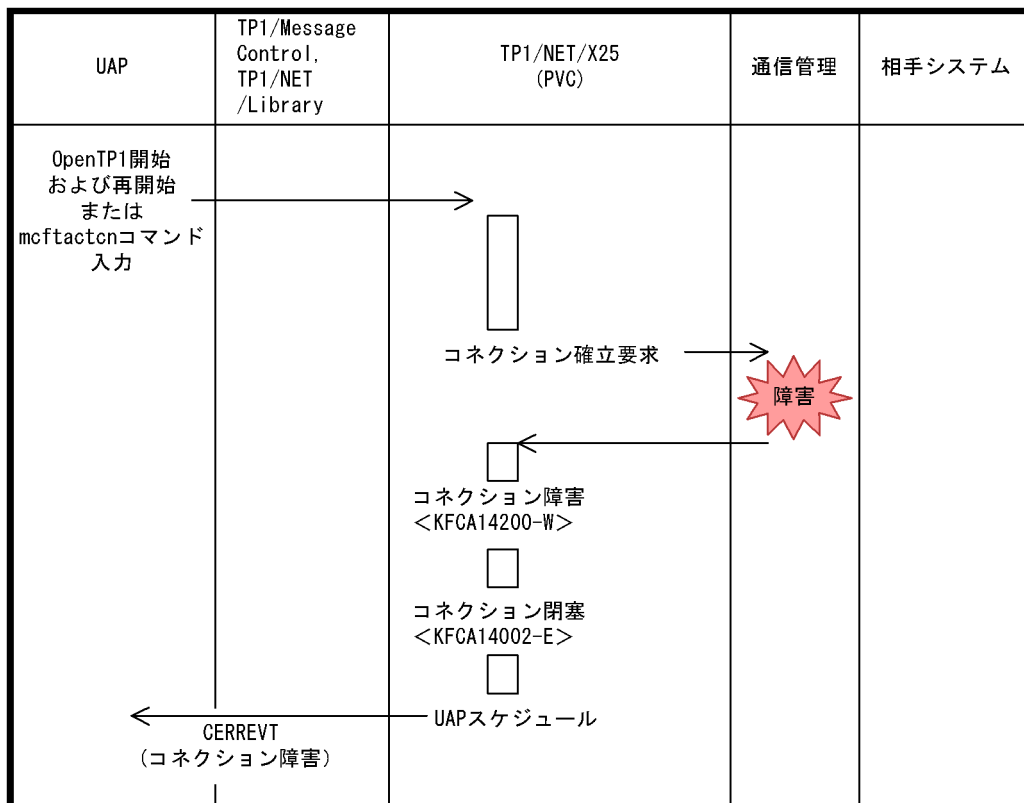


図 B-2 コネクション切断 (回復可能障害・手動再確立・論理端末自動起動)(PVC)

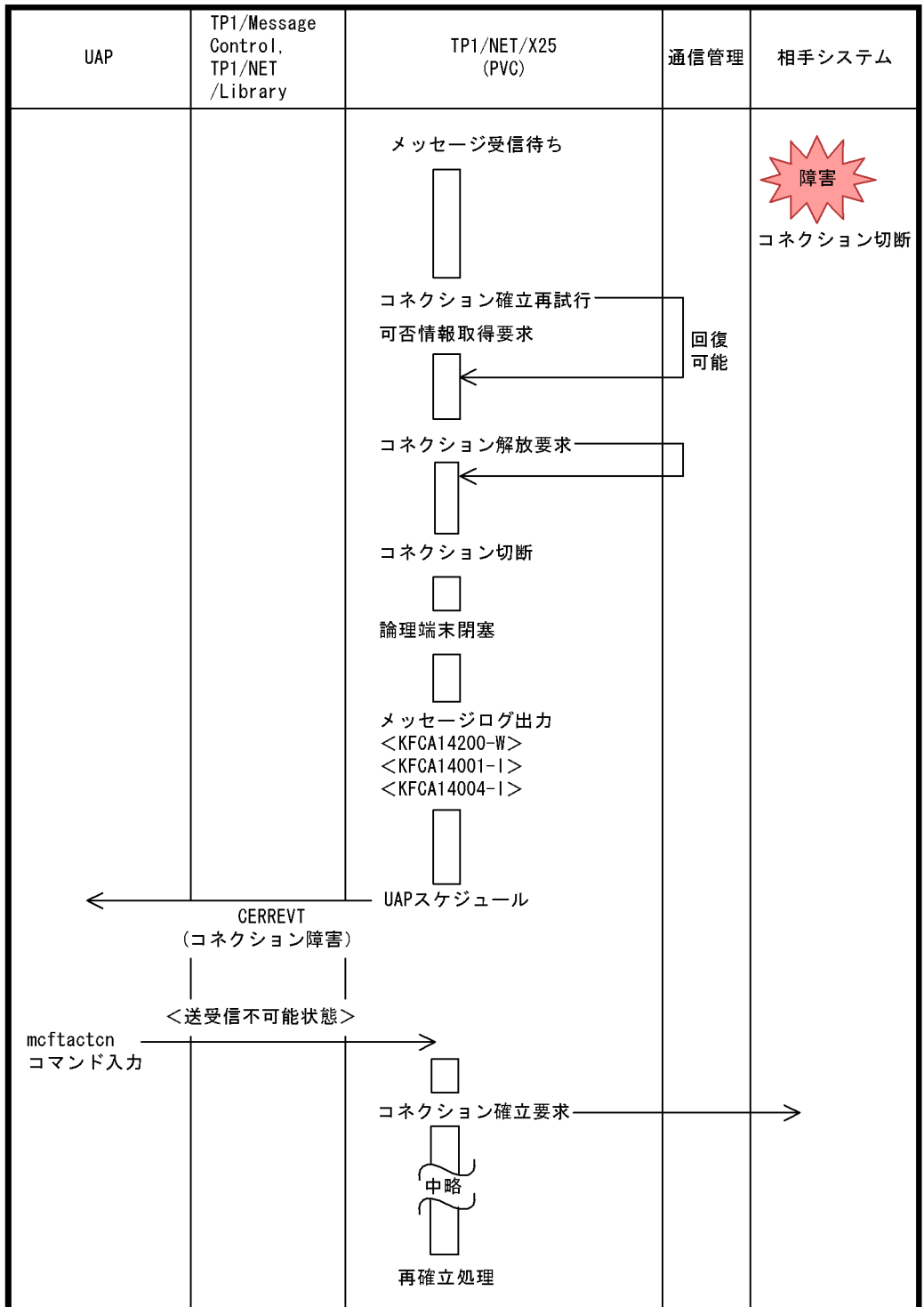


図 B-3 入力メッセージ編集 UOC 障害 (PVC)

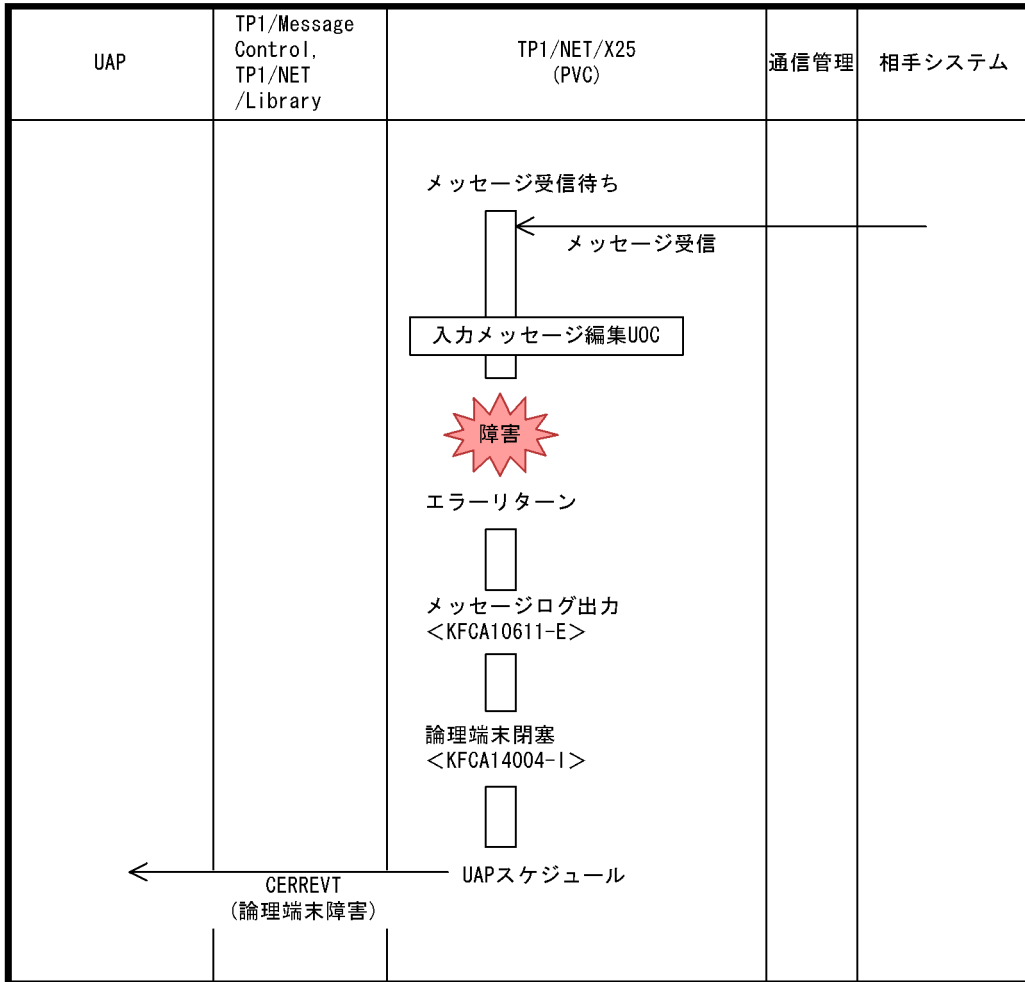
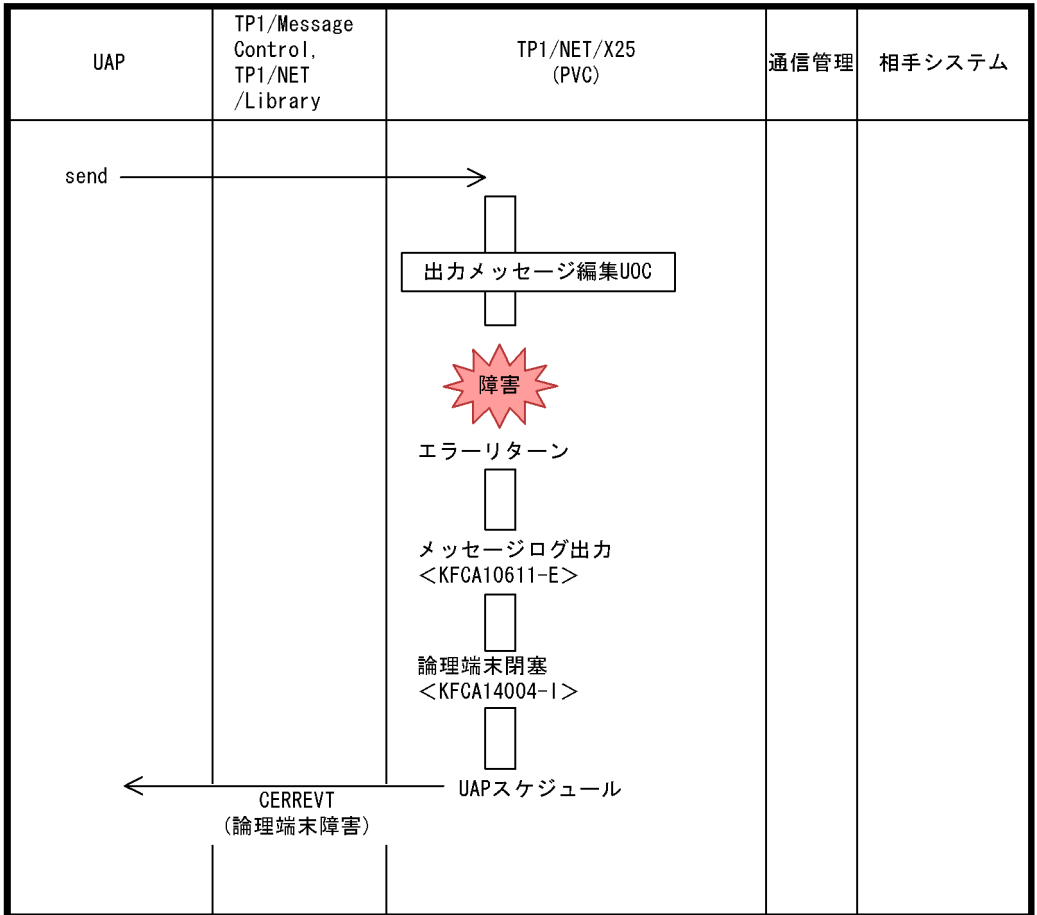


図 B-4 出力メッセージ編集 UOC 障害 (PVC)



付録 B.2 VC の障害

VC の障害発生時の処理の流れを図 B-5 ~ 図 B-8 に示します。

図 B-5 コネクション確立時の障害 (VC)

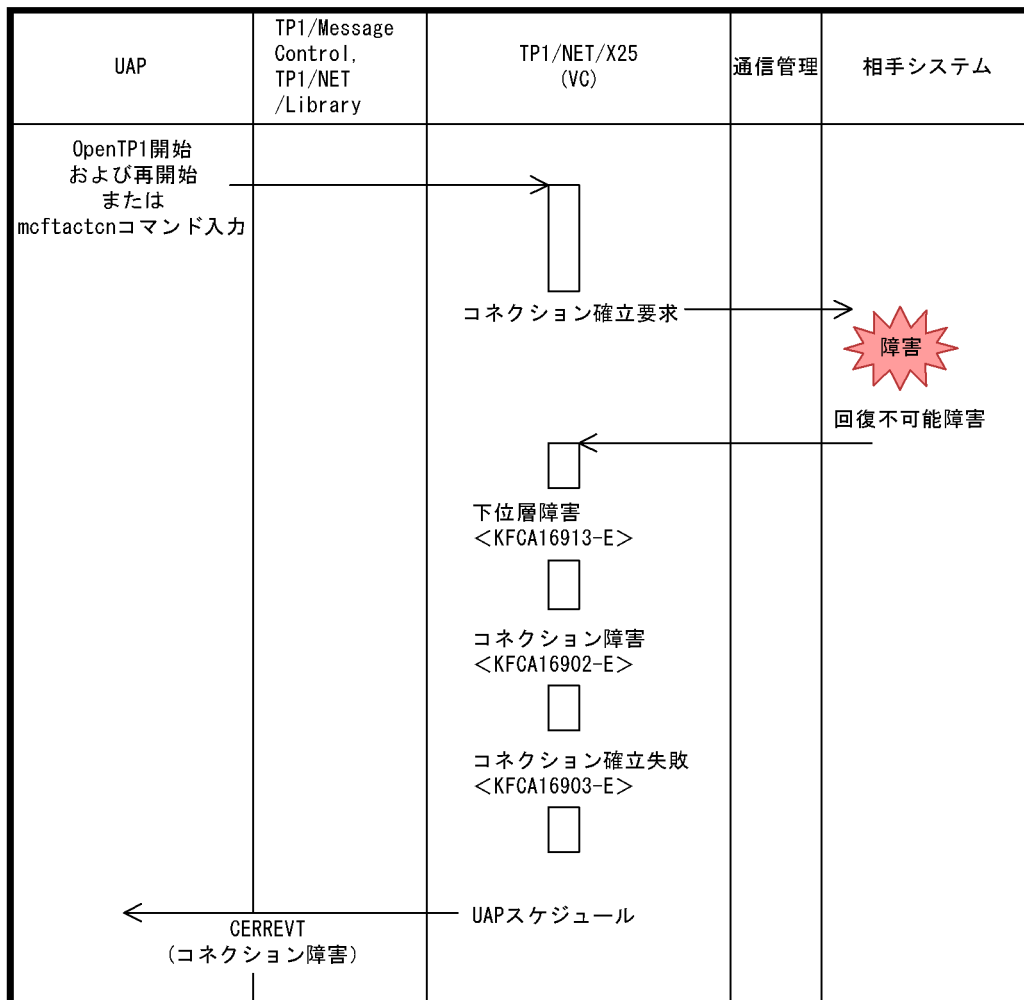
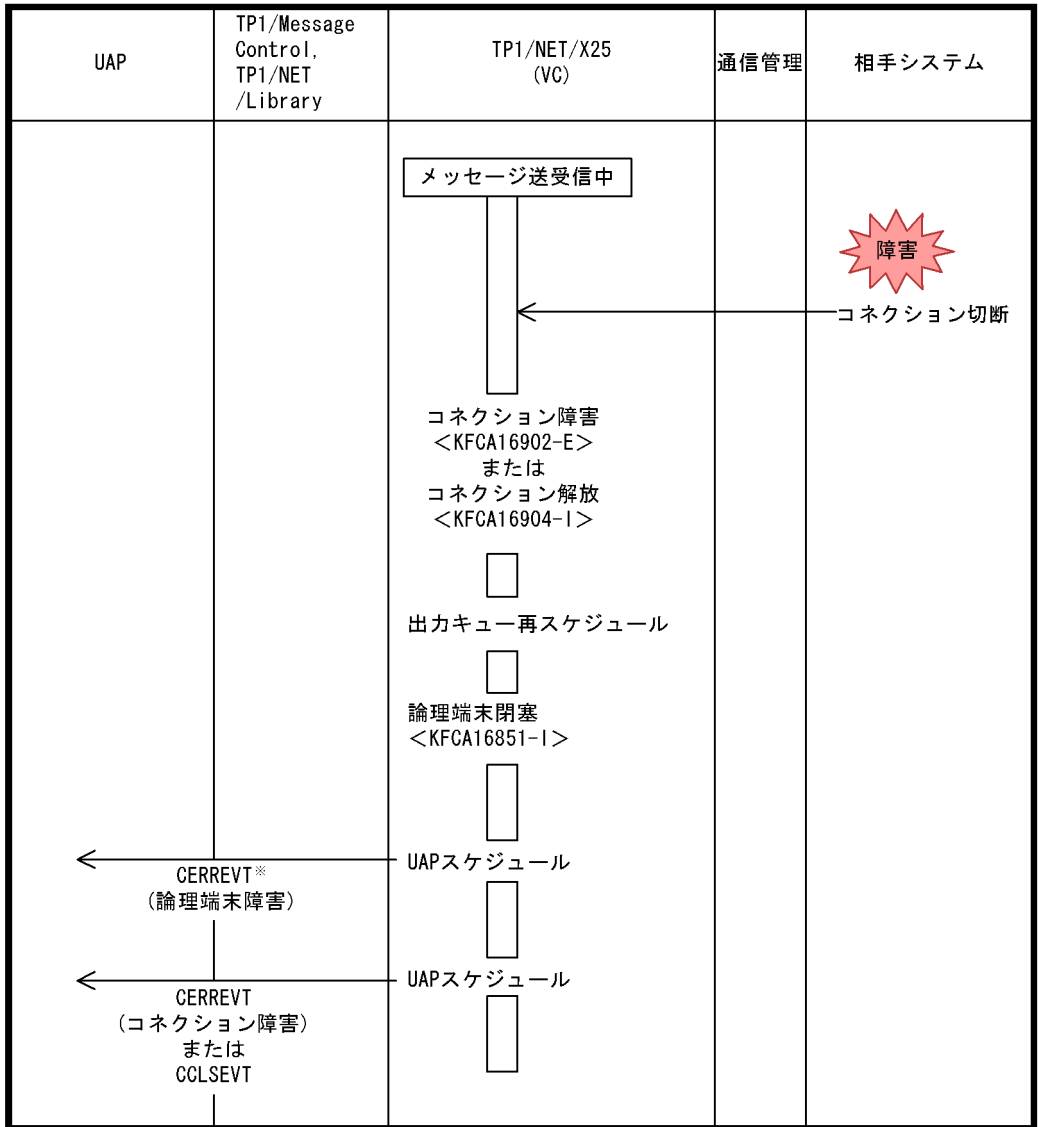


図 B-6 コネクション切断 (VC)



注

送信仕掛り中のメッセージがあるときだけ通知します。

図 B-7 入力メッセージ編集 UOC 障害 (VC)

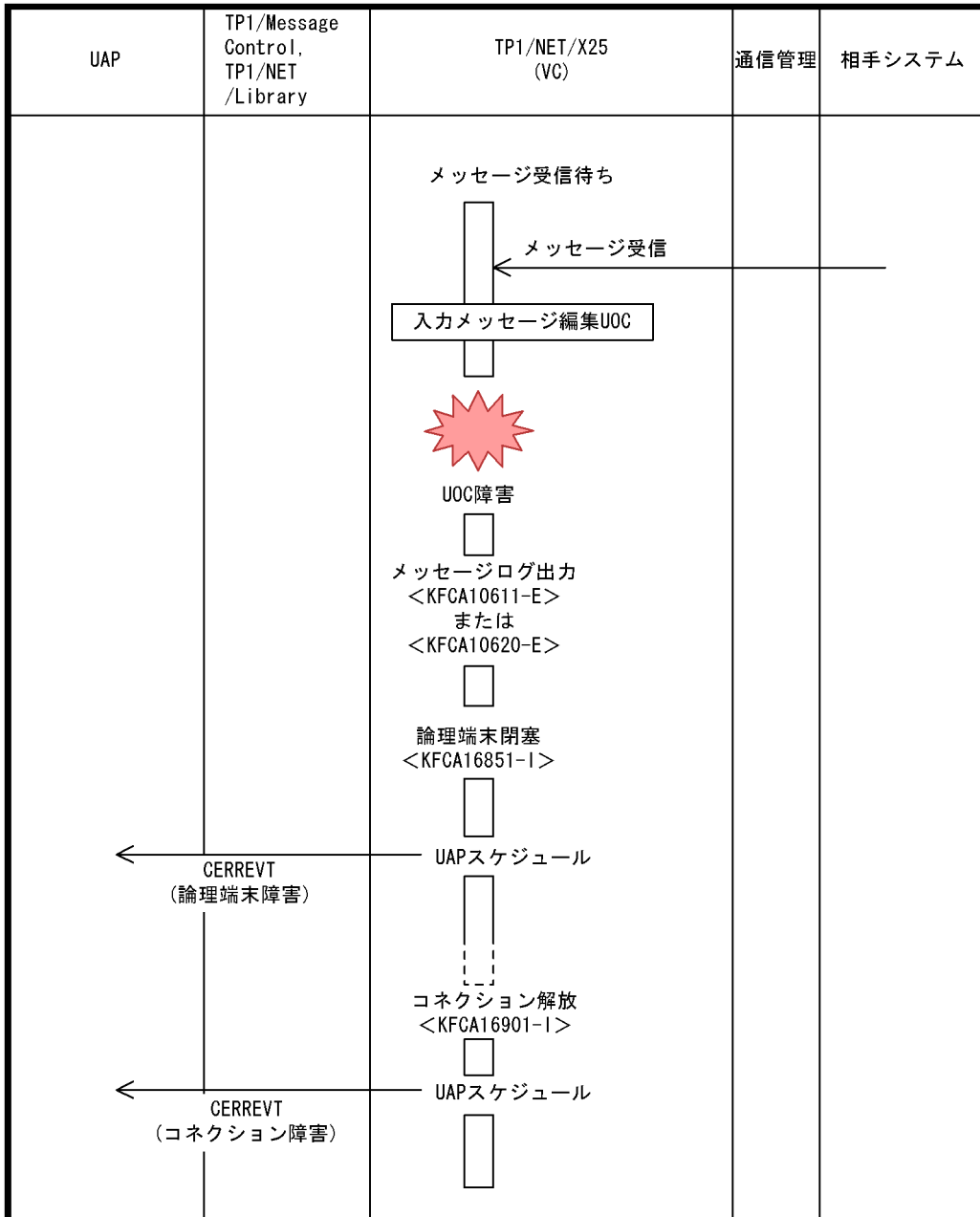
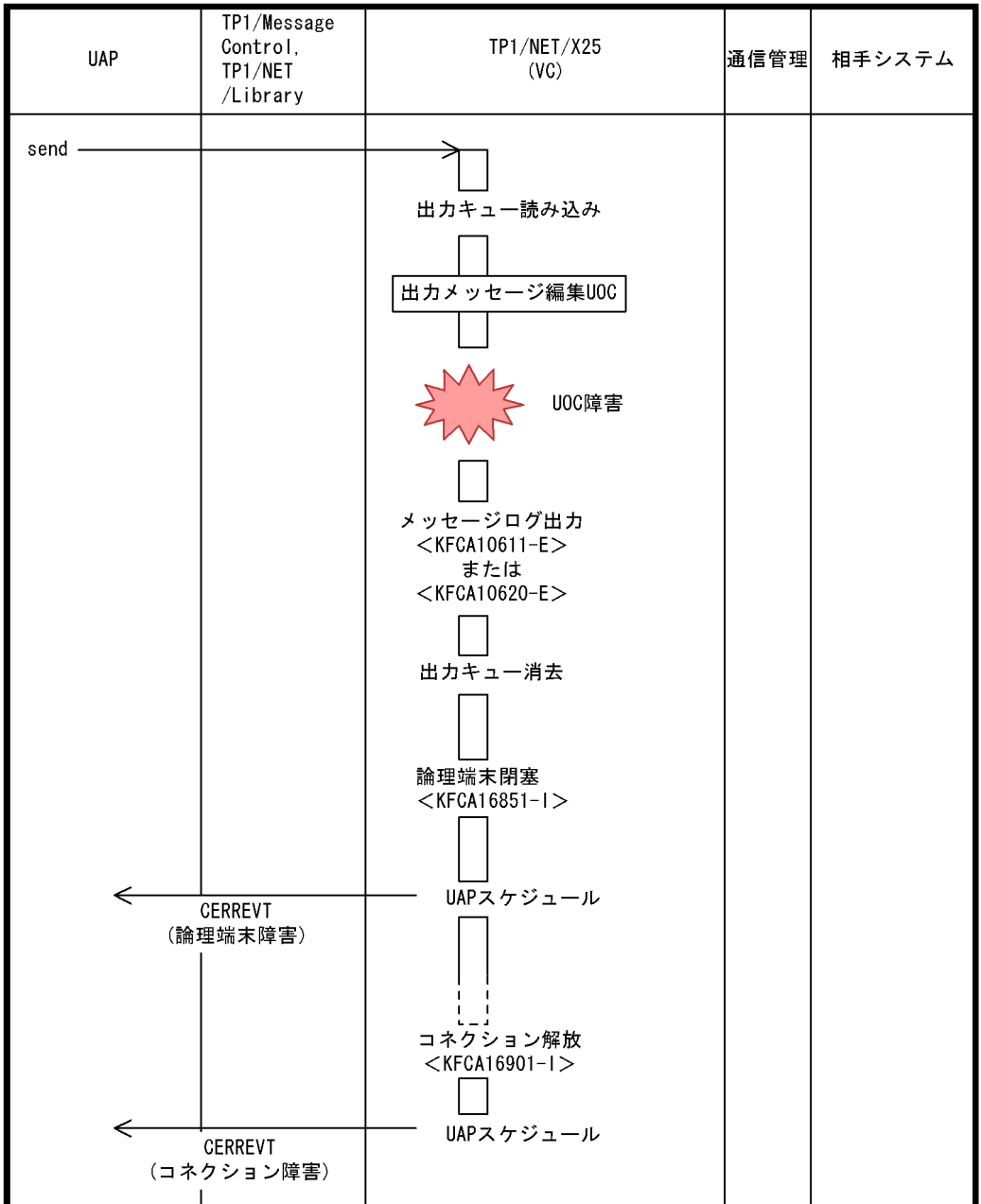


図 B-8 出力メッセージ編集 UOC 障害 (VC)



付録 C 理由コード一覧

付録 C.1 ERREVT2 の理由コード

ERREVT2 の理由コードと、コードが示す ERREVT2 の通知理由を次の表に示します。

表 C-1 ERREVT2 の理由コード

C 言語の理由コード (16 進数字)	COBOL 言語の理由コード (外部 10 進数字)	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。

付録 C.2 PVC の理由コード

PVC の場合の障害通知イベント (CERREVT) 通知時の理由コードを示します。コネクション障害の理由コードを表 C-2 に、論理端末障害の理由コードを表 C-3 に示します。

表 C-2 コネクション障害の理由コード (PVC)

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
DCMX25_RSN1_MCF (00000001)MCF 障害	DCMX25_RSN2_CNNC (00000000)	コネクション確立不可能
	DCMX25_RSN2_CNER (00000001)	コネクション切断
	DCMX25_RSN2_ERROR (00000003)	MCF 内部論理矛盾
	DCMX25_RSN2_RBFOF (00000002)	受信バッファオーバーフロー
	DCMX25_RSN2_RBFST (00000010)	受信バッファ不足
	DCMX25_RSN2_RESET (00000011)	リセット受信
	DCMX25_RSN2_LEBLK (20000000)	コネクション強制解放 (送信中の論理端末閉塞)

表 C-3 論理端末障害の理由コード (PVC)

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
DCMPCM_RSN1_MCF (00000001)MCF 障害	DCMX25_RSN2_SBFOF (00000004)	送信バッファオーバーフロー
	DCMX25_RSN2_OTGET (00000005)	出力キュー読み込み障害
	DCMX25_RSN2_SBFST (00000006)	送信バッファ不足
	DCMX25_RSN2_CHDER (00000012)	制御ヘッダ不正
DCMX25_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
	DCMX25_RSN2_BCNT (00000001)	使用バッファ数不正
	DCMX25_RSN2_SEG (00000002)	有効セグメント不正
	DCMX25_RSN2_BADR (00000003)	編集バッファアドレス不正

付録 C.3 VC の理由コード

VC の場合の障害通知イベント (CERREVT) 通知時の理由コードを示します。コネクション障害の理由コードを表 C-4 に、論理端末障害の理由コードを表 C-5 に示します。

表 C-4 コネクション障害の理由コード (VC)

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
DCMX2M_RSN1_MCF (00000001)MCF 障害	DCMX2M_RSN2_CNNC (00000000)	コネクション確立不可能
	DCMX2M_RSN2_CNER (00000001)	コネクション切断
	DCMX2M_RSN2_RBFOF (00000002)	受信バッファオーバーフロー
	DCMX2M_RSN2_RBFST (00000010)	受信バッファ不足
	DCMX2M_RSN2_RESET (00000011)	リセットパケット受信
	DCMX2M_RSN2_BADAD (00000015)	プロトコルアドレス不正
	DCMX2M_RSN2_EVTNG (00000016)	MCF イベント処理用 MHP 起動失敗

表 C-5 論理端末障害の理由コード (VC)

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
DCMX2M_RSN1_MCF (00000001)MCF 障害	DCMX2M_RSN2_SBFOF (00000004)	送信バッファオーバーフロー
	DCMX2M_RSN2_OTGET (00000005)	出力キュー読み込み障害
	DCMX2M_RSN2_SBFST (00000006)	送信バッファ不足
	DCMX2M_RSN2_SNDERR (00000007)	メッセージ送信失敗
DCMX2M_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
	DCMX2M_RSN2_BCNT (00000001)	使用バッファ数不正
	DCMX2M_RSN2_SEG (00000002)	有効セグメント不正
	DCMX2M_RSN2_BADR (00000003)	編集バッファリストアドレス不正
	DCMX2M_RSN2_SNDERR (00000007)	メッセージ送信失敗

付録 D 旧製品からの移行に関する注意事項

(1) ソースファイルの互換性

バージョン 6 以前からバージョン 7 へ移行する場合の各種ソースファイルの互換性について説明します。

バージョン 6 以前からバージョン 7 へ移行する場合、バージョン 6 以前で使用していたソースファイルをそのまま使用できないことがあります。ソースファイルの互換性は、次の表に示すとおりです。

表 D-1 バージョン 6 以前で使用していたソースファイルの互換性

ソースファイルを作成した言語	互換性
C 言語 (32 ビット)	UAP および UOC のソースファイルを変更しないで使用できます。
COBOL 言語	UAP のソースファイルおよび MCF 通信構成定義 (プロトコル固有の定義) の定義ソースファイルを変更しないで使用できます。

(2) 適用 OS による差異

バージョン 6 以前の HI-UX/WE2 版の TP1/NET/X25 から、バージョン 7 の AIX 版の TP1/NET/X25 に移行する場合、次に示す内容に差異があるため、注意してください。

- UAP のインタフェース
- UAP のオブジェクトファイルを作成するコマンド

(a) UAP のインタフェース

UAP のインタフェースで、mcftalccn コマンドの -o オプションの ctlheader オペランドに use を指定して、送受信データに制御ヘッダを付加する場合、制御ヘッダ内のエントリへの設定および参照方法に差異があります。

PVC を使用する場合

制御ヘッダの形式を定義している <demx25c.h> が異なります。

次の図に示すとおり、HI-UX/WE2 版ではビットフィールド構造体ですが、AIX 版では char 型で表します。

■ dcmx25c.h の定義内容 (HI-UX/WE2)

```
typedef struct {
    unsigned char id ;
    unsigned char cause ;
    unsigned char diag ;
    struct dcmx25_ctlbit_tag {
        unsigned char q_bit :1;
        unsigned char msgk_bit :2;
        unsigned char rsv :5;
    } dcmx25_ctlbit ;
} dcmx25_ctl_type ;
```

■ dcmx25c.h の定義内容 (AIX)

```
typedef struct {
    unsigned char id ;
    unsigned char cause ;
    unsigned char diag ;
    unsigned char dcmx25_ctlbit ;
} dcmx25_ctl_type ;
```



各ビットへのアクセスは、ビット単位からバイト単位に変更となります。したがって、制御ヘッダ内への設定および参照は、ビット演算で設定および参照するようにします。

データパケットの Q ビット設定例 (HI-UX/WE2)

```
dcmx25_ctlbit.q_bit = 1 ;
```

データパケットの Q ビット設定例 (AIX)

```
dcmx25_ctlbit |= 0x80 ;
```

VC を使用する場合

制御ヘッダの形式を定義している <dcmx2m.h> が異なります。

次の図に示すとおり、HI-UX/WE2 版ではビットフィールド構造体ですが、AIX 版では char 型で表します。

■ dcmx2m.h の定義内容 (HI-UX/WE2)

```
typedef struct {
    unsigned char id ;
    unsigned char rsv1 ;
    unsigned char rsv2 ;
    struct dcmx2m_h_ctlbit_tag {
        unsigned char q_bit :1;
        unsigned char k_msg :2;
        unsigned char rsv :5;
    } dcmx2m_ctlbit ;
} dcmx2m_ctl_type ;
```

■ dcmx2m.h の定義内容 (AIX)

```
typedef struct {
    unsigned char id ;
    unsigned char rsv1 ;
    unsigned char rsv2 ;
    unsigned char dcmx2m_h_ctlbit ;
} dcmx2m_ctl_type ;
```



各ビットへのアクセスは、ビット単位からバイト単位に変更となります。したがって、制御ヘッダ内への設定および参照は、ビット演算で設定および参照するようにします。

データパケットのQビット設定例 (HI-UX/WE2)

```
dcmx2m_ctlbit.q_bit = 1 ;
```

データパケットのQビット設定例 (AIX)

```
dcmx2m_ctlbit |= 0x80 ;
```

(b) UAP のオブジェクトファイルを作成するコマンド

TP1/NET/X25 上の MCF メイン関数, UOC, および UAP で, <dcmx25c.h> または, <dcmx2m.h> を使用する場合, UAP 作成時のコンパイルオプションに, -DDCMCFAIX を追加する必要があります。

UAP のオブジェクトファイルを作成するときのコマンド入力例を次に示します。

コマンド入力例

```
cc -c -qchars=signed -DDCMCFAIX UAPのソースファイル名 -I$DCDIR/  
include
```

付録 E バージョンアップ時の変更点

各バージョンでの変更点を次に示す分類ごとに示します。

- 関数，定義およびコマンドの追加と削除
- 動作の変更
- 関数，定義およびコマンドのデフォルト値の変更

付録 E.1 07-00 での変更点

TP1/NET/X25 07-00 での関数，定義およびコマンドの追加と削除はありません。

TP1/NET/X25 07-00 での動作の変更点を次に示します。

表 E-1 TP1/NET/X25 07-00 での動作の変更点

分類	内容
関数	C 言語の関数について，32 ビットアーキテクチャでのインタフェースと 64 ビットアーキテクチャでのインタフェースを統一した。

TP1/NET/X25 07-00 での関数，定義およびコマンドのデフォルト値の変更はありません。

付録 F 用語解説

(英字)

AJ (メッセージ送信完了ジャーナル)

TP1/NET/X25 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。

GJ (メッセージ受信ジャーナル)

TP1/NET/X25 で取得する履歴情報の一つです。メッセージ入力通番、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 UAP に receive 関数で引き渡す直前です。

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。入力メッセージ通番、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 で取得する情報の一つです。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。

PVC (Permanent Virtual Circuit)

X.25 プロトコルの通信相手の選択方式の一つです。通信相手を固定してメッセージの送受信をする方式です。

VC (Virtual Call)

X.25 プロトコルの通信相手の選択方式の一つです。通信相手を固定しないで、メッセージの送受信を開始するときに、相手システムのダイヤル番号を指定する方式です。

(力行)

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 の通信管理側の通信接点であり、MCF と通信管理はコネクション単位にメッセージを送受信します。

(ラ行)

論理端末

MCF の UAP 側の通信接点であり，MCF と UAP は論理端末単位にメッセージを送受信します。

索引

A

AJ 227
any〔PVC〕12
any〔VC〕23
AP 間通信の概要 2
AP 間通信の形態 4

C

CBLDCMCF('RECEIVE') 48
CBLDCMCF('RESEND') 56
CBLDCMCF('SEND') 62
CCLSEVT 106, 114
CERREVT 105, 112
COBOL 言語のメッセージ送受信 30
COPNEVT 106, 114
C 言語のメッセージ送受信 30

D

dc_mcf_receive 32
dc_mcf_resend 38
dc_mcf_send 43

E

ERREVT1 103, 107
ERREVT2 103, 108
ERREVT2 の理由コード 220
ERREVT3 104, 109
ERREVT4 104, 111

G

GJ 227

I

IJ 227
INS 回線速度〔VC〕131
INS サービス種別〔VC〕131

M

max_open_fds 142
max_socket_descriptors 141
mcfactcn 157
mcfactcle 159
mcfalccn〔PVC〕123
mcfalccn〔VC〕128
mcfalced〔PVC〕133
mcfalced〔VC〕134
mcfalcle〔PVC〕135
mcfalcle〔VC〕137
mftcomn〔PVC〕139
mftdctcn 161
mftdctcle 163
mftlscn 166
mftlslc 169
MCF アプリケーション定義 116
MCF イベント 100
MCF イベント一覧 100
MCF イベントインタフェース 100
MCF イベント情報の形式 (COBOL 言語) 107
MCF イベント情報の形式 (C 言語) 102
MCF イベント処理用 MHP 100
MCF イベント通知時のセグメント構成 101
MCF イベントの共通ヘッダ 102
MCF イベントの種類 100
MCF サービス名の登録 174
MCF 障害と対応処理 (PVC) 186
MCF 障害と対応処理 (VC) 201
MCF 通信構成共通定義〔PVC〕139
MCF 通信構成定義 116
MCF 通信プロセスでアクセスするファイルの最大数 142
MCF 定義オブジェクトの生成 144
MCF で使用する定義ファイル 116
MCF マネージャ定義 116
MCF マネージャ定義の例 (PVC の場合) 148
MCF マネージャ定義の例 (VC の場合) 152

MCF メイン関数のコーディング概要 (ANSI C, C++ の場合) 175
 MCF メイン関数のコーディング概要 (K & R 版 C の場合) 176
 MCF メイン関数の作成 175
 MCF メイン関数のディレクトリへの組み込み方法 177
 MJ 227
 module 140

N

noans [PVC] 12
 noans [VC] 23

O

OJ 227

P

PVC 227
 PVC の AP 間通信の仕組み 10
 PVC の AP 間通信メッセージの送受信 15
 PVC の機能 9
 PVC の障害 212
 PVC の障害対策 181
 PVC のメッセージ受信の流れ 16
 PVC のメッセージ送信の流れ 15
 PVC の理由コード 220

R

RECEIVE 69

S

SEND 73

T

TP1/NET/X25 固有の定義 118
 TP1/NET/X25 の組み込みの流れ 174
 TP1/NET/X25 の定義 116
 TP1/NET/X25 を組み込んだソフトウェア構成の例 6

TP1/NET/X25 を使用したネットワーク構成の例 2

U

UAP 障害 (PVC) 192
 UAP 障害 (VC) 205
 UAP 障害と対応処理 (PVC) 186
 UAP 障害と対応処理 (VC) 200
 UOC 84
 UOC インタフェース用のパラメタとバッファの関係 91
 UOC 作成上の注意事項 98
 UOC で使用できる関数 98
 UOC の異常処理 99
 UOC の構造 98
 UOC の実行タイミング 99

V

VC 227
 VC から通知する MCF イベントの種別 132
 VC の AP 間通信の仕組み 18
 VC の AP 間通信メッセージの送受信 26
 VC の機能 17
 VC のコネクション確立の流れ (着呼) 20
 VC のコネクション確立の流れ (発呼) 19
 VC の障害 215
 VC の障害対策 195
 VC のメッセージ受信の流れ 27
 VC のメッセージ送信の流れ 26
 VC の理由コード 221

あ

相手 SNPA アドレス [VC] 131
 相手システムからのコネクションの解放 [VC] 21
 アプリケーション名 [PVC] 136
 アプリケーション名 [VC] 138
 アプリケーション名の決定 84

い

一方受信形態 4

一方受信形態の AP 間通信の例 4
 一方送信メッセージ 4
 一方送信メッセージ書き込み後のコネクション障害 (PVC) 189
 一方送信メッセージ書き込み前のコネクション障害 (PVC) 189
 一方送信メッセージ送信後のコネクション障害 (PVC) 188
 一方送信メッセージ送信中のコネクション障害 (PVC) 187
 一方送信メッセージの受信 (COBOL 言語) 48
 一方送信メッセージの受信 (C 言語) 32
 一方送信メッセージの受信 [PVC] 15
 一方送信メッセージの受信 [VC] 26
 一方送信メッセージの送信 (COBOL 言語) 62
 一方送信メッセージの送信 (C 言語) 43
 一方送信メッセージの送信 (データ操作言語) 73
 一方送信メッセージの送信 [PVC] 15
 一方送信メッセージの送信 [VC] 26

う

運用コマンド 155

か

概要 1
 仮想スロット番号 [PVC] 124
 仮想スロット番号 [VC] 129

き

キューグループ ID [PVC] 136
 キューグループ ID [VC] 138
 旧製品からの移行に関する注意事項 223
 共通定義の例 (PVC の場合) 148
 共通定義の例 (VC の場合) 152

く

組み込み方法 173

こ

コネクション 227
 コネクション [PVC] 10
 コネクション [VC] 18
 コネクション ID [PVC] 123
 コネクション ID [VC] 128
 コネクション確立時障害の確立再試行回数 [VC] 130
 コネクション確立時障害の確立再試行間隔 [VC] 130
 コネクション確立の再試行 [VC] 20, 130
 コネクション障害 (PVC) 187
 コネクション障害 (VC) 202
 コネクション障害時再試行回数 [PVC] 125
 コネクション障害時再試行間隔 [PVC] 125
 コネクション障害と対応処理 (PVC) 182
 コネクション障害と対応処理 (VC) 196
 コネクション障害の理由コード (PVC) 221
 コネクション障害の理由コード (VC) 222
 コネクション切断後の再確立 [PVC] 11
 コネクション定義の開始 [PVC] 123
 コネクション定義の開始 [VC] 128
 コネクション定義の終了 [PVC] 133
 コネクション定義の終了 [VC] 134
 コネクションと論理端末の関係 [PVC] 12
 コネクションと論理端末の関係 [VC] 22
 コネクションの解放 161
 コネクションの解放 [PVC] 11
 コネクションの解放 [VC] 20
 コネクションの確立 157
 コネクションの確立 [PVC] 10
 コネクションの確立 [VC] 18
 コネクションの確立と解放 [PVC] 10
 コネクションの確立と解放 [VC] 18
 コネクションの種類 [VC] 131
 コネクションの状態表示 166
 コネクションの切断 [VC] 22
 コネクションを自動的に確立 [PVC] 124
 コネクションを自動的に確立 [VC] 129

し

自システムからのコネクションの解放〔VC〕
 21
 システム構成例 (PVC の場合) 146
 システム構成例 (VC の場合) 150
 システムサービス共通情報定義 141
 システムサービス情報定義 140
 システムサービス情報定義ファイルの作成
 174
 システム定義 115
 自動解放〔VC〕20
 自動確立〔VC〕18
 受信ウィンドウサイズ〔PVC〕127
 受信スケジュール関係の障害と対応処理
 (PVC) 184
 受信スケジュール関係の障害と対応処理
 (VC) 198
 受信パケットサイズ〔PVC〕126
 受信バッファ空き待ち再試行回数〔PVC〕
 139
 受信バッファ空き待ち時間〔PVC〕139
 出力キュー障害 (PVC) 191
 出力キュー障害 (VC) 204
 出力メッセージの編集 91
 出力メッセージの割り当て先〔PVC〕136
 出力メッセージの割り当て先〔VC〕138
 出力メッセージ編集 UOC インタフェース
 92
 手動解放〔VC〕21
 手動確立〔VC〕18
 障害の種類と対応処理 (PVC) 182
 障害の種類と対応処理 (VC) 196
 障害発生時の処理の流れ 212

す

スケジュール関係の障害 (PVC) 190
 スケジュール関係の障害 (VC) 203

せ

制御ヘッダを使用〔PVC〕125
 制御ヘッダを使用〔VC〕130
 接続プロトコル〔PVC〕124

接続プロトコル〔VC〕129
 前提プログラム 6

そ

送信ウィンドウサイズ〔PVC〕127
 送信完了時の情報を取得〔PVC〕136
 送信完了時の情報を取得〔VC〕138
 送信スケジュール関係の障害と対応処理
 (PVC) 185
 送信スケジュール関係の障害と対応処理
 (VC) 199
 送信パケットサイズ〔PVC〕126
 送信メッセージの通番編集 95
 送信メッセージの通番編集 UOC インタ
 フェース 97
 ソケット用ファイル記述子の最大数 141
 ソフトウェアの構成 6

ち

着呼コネクション〔VC〕18
 着呼コネクションの確立〔VC〕19

つ

通信管理の構成定義の例 (PVC の場合) 146
 通信管理の構成定義の例 (VC の場合) 151
 通信管理プログラムと関連づける内容 145
 通信形態 4

て

定義オブジェクトファイルの生成 178
 定義コマンドの指定順序 122
 定義の種類 118
 定義例 146
 ディスク出力メッセージ最大格納数〔PVC〕
 135
 ディスク出力メッセージ最大格納数〔VC〕
 137
 データ操作言語 (COBOL 言語) のメッセー
 ジ送受信 30
 適用範囲 5
 伝送動作モード 5

に

入力キュー障害 (PVC) 190
 入力キュー障害 (VC) 203
 入力メッセージの編集 84
 入力メッセージ編集 UOC インタフェース
 87
 任意型 [PVC] 12
 任意型 [VC] 23

は

発呼コネクション [VC] 18
 発呼コネクションの確立 [VC] 18
 バッファ形式 [PVC] 14
 バッファ形式 [VC] 24
 バッファの空き待ち [PVC] 139

ひ

非応答型 [PVC] 12
 非応答型 [VC] 23

ふ

プロトコル固有定義の例 (PVC の場合) 149
 プロトコル固有定義の例 (VC の場合) 153
 プロトコルの種別 [PVC] 123
 プロトコルの種別 [VC] 128
 分岐送信形態 4
 分岐送信形態の AP 間通信の例 4

め

メッセージジャーナル 227
 メッセージ受信後の UAP 異常終了 (PVC)
 193
 メッセージ受信後の UAP 異常終了 (VC)
 206
 メッセージ受信ジャーナル 227
 メッセージ受信前の UAP 異常終了 (PVC)
 192
 メッセージ受信前の UAP 異常終了 (VC)
 205
 メッセージ受信用バッファグループ番号
 [PVC] 123

メッセージ受信用バッファグループ番号
 [VC] 128
 メッセージ出力ジャーナル 227
 メッセージ送受信インタフェース 29
 メッセージ送受信インタフェースの一覧 30
 メッセージ送受信の関数 (C 言語) 30
 メッセージ送受信の関数に対応するプログラ
 ム (COBOL 言語) 30
 メッセージ送受信の流れ 208
 メッセージ送受信の通信文 (データ操作言
 語) 31
 メッセージ送信完了ジャーナル 227
 メッセージ送信用バッファグループ番号
 [PVC] 123
 メッセージ送信用バッファグループ番号
 [VC] 128
 メッセージとセグメントの関係 [PVC] 14
 メッセージとセグメントの関係 [VC] 24
 メッセージ入力ジャーナル 227
 メッセージの再送 (COBOL 言語) 56
 メッセージの再送 (C 言語) 38
 メッセージの受信 (データ操作言語) 69
 メッセージ編集用バッファグループ番号
 [PVC] 124
 メッセージ編集用バッファグループ番号
 [VC] 129
 メモリ出力メッセージ最大格納数 [PVC]
 135
 メモリ出力メッセージ最大格納数 [VC] 137

ゆ

ユーザアプリケーションプログラム作成例
 77
 ユーザOWNコーディング 84
 ユーザOWNコーディングインタフェース
 84
 ユーザ情報の操作方法 99

よ

用語解説 227

り

リセットパケットを受信した場合の処置方法
〔PVC〕 126
理由コード一覧 220

ろ

論理端末 228
論理端末〔PVC〕 12
論理端末〔VC〕 22
論理端末障害の理由コード（PVC） 221
論理端末障害の理由コード（VC） 222
論理端末定義〔PVC〕 135
論理端末定義〔VC〕 137
論理端末とアプリケーションの型との関係
〔PVC〕 12
論理端末とアプリケーションの型との関係
〔VC〕 23
論理端末の状態表示 169
論理端末の端末タイプ〔PVC〕 135
論理端末の端末タイプ〔VC〕 137
論理端末の閉塞 163
論理端末の閉塞〔PVC〕 13
論理端末の閉塞〔VC〕 23
論理端末の閉塞解除 159
論理端末の閉塞解除〔PVC〕 13
論理端末の閉塞解除〔VC〕 24
論理端末の閉塞と閉塞解除〔PVC〕 13
論理端末の閉塞と閉塞解除〔VC〕 23
論理端末の閉塞を解除〔PVC〕 125
論理端末名称〔PVC〕 135
論理端末名称〔VC〕 137
論理チャネルグループ番号〔PVC〕 124
論理チャネル番号〔PVC〕 124
論理メッセージ〔PVC〕 14
論理メッセージ〔VC〕 24

わ

割り込みメッセージの受信を通知〔VC〕 131

ソフトウェアマニュアルのサービス ご案内

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います。

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	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
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. インターネットでのマニュアル公開

2種類のマニュアル公開サービスを実施しています。

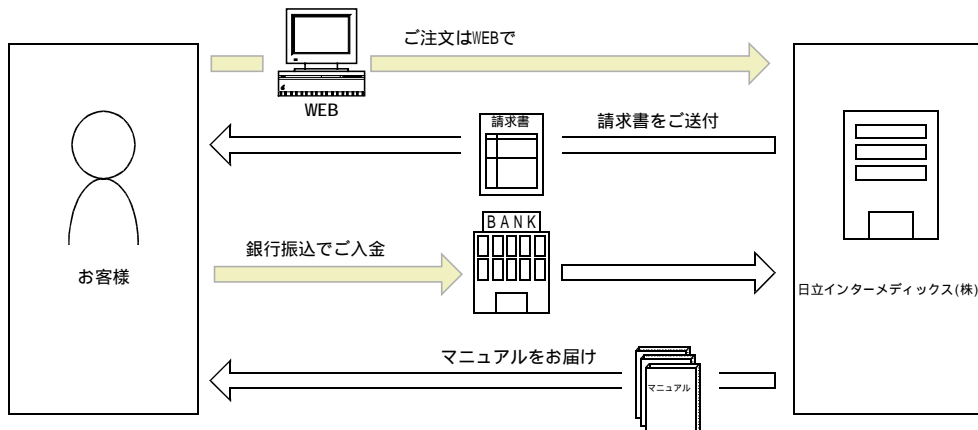
(1) マニュアル情報ホームページ「オンラインマニュアル」での公開

製品をよりご理解いただくためのご参考として、一部製品のマニュアルを公開しています。

(2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開

ソフトウェアサポートサービスご契約のお客様向けにマニュアルを公開しています。公開しているマニュアルの一覧、本サービスの対象となる契約の種別などはマニュアル情報ホームページの「サポートサービス」をご参照ください。

3. マニュアルのご注文



マニュアル情報ホームページの「マニュアルのご購入」にアクセスし、お申し込み方法をご確認のうえWEBからご注文ください。ご注文先は日立インターメディアックス(株)となります。

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。

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。