
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/HDLC 編

解説・手引・文法・操作書

3000-3-D72

マニュアルの購入方法

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

HITACHI

対象製品

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

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

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

P-F1M64-32312 uCosminexus TP1/NET/HDLC 07-00

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

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

輸出時の注意

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

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

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

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

発行

2006 年 11 月 (第 1 版) 3000-3-D72

著作権

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

はじめに

このマニュアルは、OpenTP1 システムの中で、TP1/NET/HDLC を使用した AP 間通信システムの作成方法、および運用方法について説明したものです。

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

対象読者

OpenTP1 システムの通信に HDLC プロトコルを使用するシステム管理者、システム設計者、およびプログラマ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、次のマニュアルを理解されていることを前提としています。

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

マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 概要

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

第 2 章 機能

TP1/NET/HDLC を使用した場合の、コネクションの確立や論理端末の端末タイプなどの AP 間通信の仕組み、およびメッセージの送受信について説明しています。

第 3 章 メッセージ送受信インタフェース

UAP の作成方法、TP1/NET/HDLC が提供する各種メッセージ、および作成例について説明しています。

第 4 章 ユーザOWNコーディング、MCF イベントインタフェース

TP1/NET/HDLC に関連するユーザOWNコーディング、および MCF イベントのインタフェースについて説明しています。

第 5 章 システム定義

HDLC プロトコルを使用するために必要な、OpenTP1 のシステム定義の中での TP1/NET/HDLC 固有のシステム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。

はじめに

第 6 章 運用コマンド

TP1/NET/HDLC で使用する運用コマンドについて説明しています。

第 7 章 組み込み方法

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

第 8 章 障害対策

TP1/NET/HDLC 運用中に発生する障害と、TP1/NET/HDLC の対応処理、およびメッセージの処理について説明しています。

付録 A メッセージ送受信の処理の流れ

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

付録 B 障害発生時の処理の流れ

メッセージを送受信するときに発生した障害処理の流れについて説明しています。

付録 C 理由コード一覧

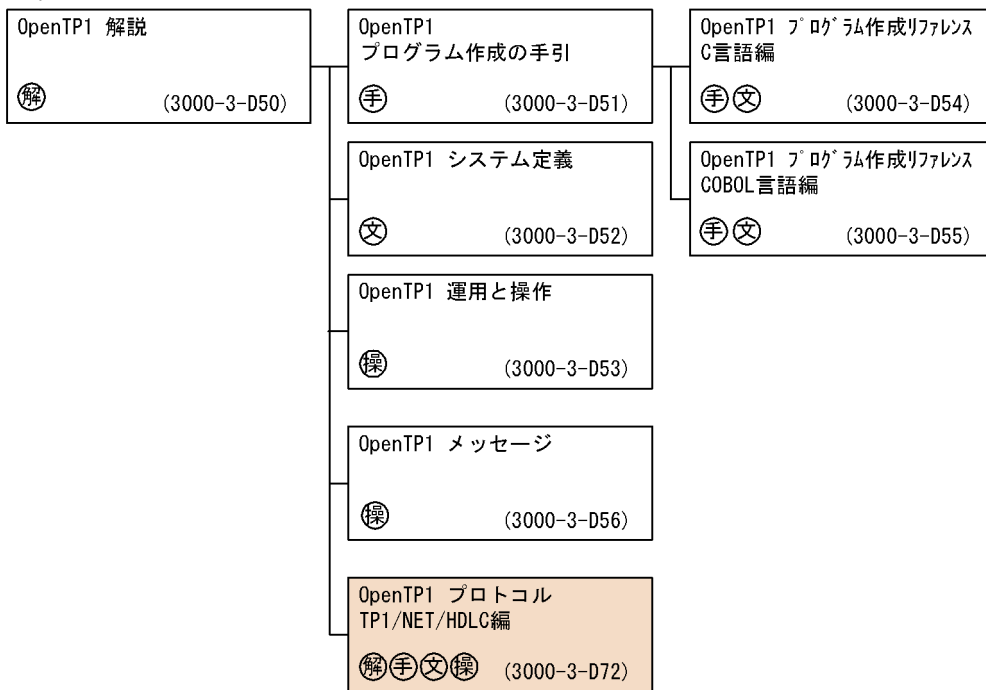
障害通知イベントが発生した場合の理由コードについて説明しています。

付録 D 用語解説

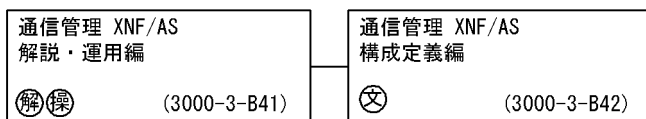
TP1/NET/HDLC で使用する用語について説明しています。

関連マニュアル

●OpenTP1 Version 7



●通信管理



<記号>


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

読書手順

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



(凡例)

 : 必ず読む項目

図中で使用する記号

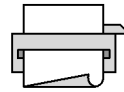
このマニュアルの図中で使用する記号を、次のように定義します。

●ワークステーション、
端末

●論理端末



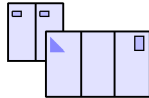
●端末プリンタ



●入出力の動作



●ホストコンピュータ



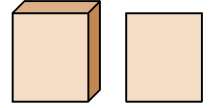
●プログラムの流れ



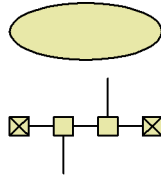
●ファイル



●プログラム



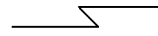
●ネットワーク



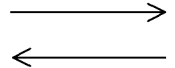
●データの流れ



●通信回線



●制御の流れ



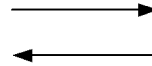
●障害



●論理回線



●その他の流れ



文法の記号

このマニュアルで使用する各種の記号を説明します。

(1) 文法記述記号

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

文法記述記号	意味
<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) <code>-i auto manual</code> -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 (A ~ F, a ~ f)
< 識別子 >	先頭がアルファベットの英数字列
< 記号名称 >	先頭が英字記号の英数字記号列
< 文字列 >	任意の文字の配列
< パス名 >	記号名称, /, および . (ピリオド) (ただし、パス名は使用する OS に依存)

このマニュアルでの表記

(1) 製品名

このマニュアルで使用する製品名称の略称を次に示します。

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

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

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

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

略語一覧

このマニュアルで使用する英略語の一覧を次に示します。

英略語	英字での表記
AP	<u>A</u> pplication <u>P</u> rogram
BID	Transaction <u>B</u> ranch <u>I</u> dentifier
CA	<u>C</u> ommunication <u>A</u> dapter
HDLC	<u>H</u> igh level <u>D</u> ata <u>L</u> ink <u>C</u> ontrol
HDLC-ABM	<u>H</u> DLC- <u>A</u> syncronous <u>B</u> alanced <u>M</u> ode
HDLC-ARM	<u>H</u> DLC- <u>A</u> syncronous <u>R</u> eoponse <u>M</u> ode
HDLC-NRM	<u>H</u> DLC- <u>N</u> ormal <u>R</u> esponse <u>M</u> ode
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork
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
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram
UAP	<u>U</u> ser <u>A</u> pplication <u>P</u> rogram

英略語	英字での表記
UOC	<u>U</u> ser <u>O</u> wn <u>C</u> oding
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（かしょ） 閉塞（へいそく）

KB（キロバイト）などの単位表記について

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

謝辞

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

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

COBOL は産業界の言語であり、特定の団体や組織の所有物ではない。

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

次に示す著作権表示付資料の著作者及び著作権者

FLOW-MATIC (Sperry Rand Corporation の商標)、

Programming for the Univac (R) I and II, Data Automation Systems,

Sperry Rand Corporation 著作権表示 1958 年、1959 年；

IBM Commercial Translator Form No.F 28-8013, IBM 著作権表示 1959 年；

FACT, DSI 27A5260-2760, Minneapolis-Honeywell, 著作権表示 1960 年

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

目次

1	概要	1
1.1	AP 間通信の概要	2
1.2	AP 間通信の形態	3
1.2.1	通信形態	3
1.2.2	適用範囲	4
1.3	ソフトウェアの構成	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	機能	9
2.1	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	14
2.1.3	論理端末とアプリケーションの型の関係	14
2.1.4	メッセージとセグメントの関係	15
2.2	AP 間通信メッセージの送受信	16
2.2.1	メッセージの送信	16
2.2.2	メッセージの受信	17
3	メッセージ送受信インタフェース	19
	メッセージ送受信インタフェースの一覧	20
	dc_mcf_receive - メッセージの受信 (C 言語)	22
	dc_mcf_resend - メッセージの再送 (C 言語)	26
	dc_mcf_send - 一方送信メッセージの送信 (C 言語)	30
	CBLDCMCF('RECEIVE ') - メッセージの受信 (COBOL 言語)	34
	CBLDCMCF('RESEND ') - メッセージの再送 (COBOL 言語)	39
	CBLDCMCF('SEND ') - 一方送信メッセージの送信 (COBOL 言語)	45
	RECEIVE - メッセージの受信 (データ操作言語)	51
	SEND - メッセージの送信 (データ操作言語)	54
	ユーザアプリケーションプログラム作成例	58

4	ユーザOWNコーディング，MCF イベントインタフェース	65
4.1	ユーザOWNコーディングインタフェース	66
4.1.1	入力メッセージの編集とアプリケーション名の決定	66
4.1.2	入力メッセージ編集 UOC インタフェース	68
4.1.3	出力メッセージの編集	73
4.1.4	出力メッセージ編集 UOC インタフェース	74
4.1.5	送信メッセージの通番編集	76
4.1.6	送信メッセージの通番編集 UOC インタフェース	78
4.1.7	UOC 作成上の注意事項	79
4.2	MCF イベントインタフェース	80
4.2.1	MCF イベントの種類	80
4.2.2	MCF イベント通知時のセグメント構成	81
4.2.3	MCF イベント情報の形式 (C 言語)	82
4.2.4	MCF イベント情報の形式 (COBOL 言語)	87
5	システム定義	97
	TP1/NET/HDLC の定義の概要	98
	TP1/NET/HDLC 固有のシステム定義の種類	100
	mcftalccn (コネクション定義の開始)	103
	mcftalcle (論理端末定義)	107
	mcftalced (コネクション定義の終了)	109
	システムサービス情報定義	110
	システムサービス共通情報定義	111
	MCF 定義オブジェクトの生成	113
	自システムの通信管理プログラムと関連づける内容	114
	定義例	115
6	運用コマンド	119
	TP1/NET/HDLC の運用コマンド	120
	mcftactcn (コネクションの確立)	121
	mcftactle (論理端末の閉塞解除)	123
	mcftdctcn (コネクションの解放)	125
	mcftdctle (論理端末の閉塞)	127

mcftlscn (コネクションの状態表示)	129
mcftlslc (論理端末の状態表示)	132

7

組み込み方法	135
7.1 TP1/NET/HDLC の組み込みの流れ	136
7.2 MCF メイン関数の作成	137
7.3 定義オブジェクトファイルの生成	140

8

障害対策	143
8.1 障害の種類と対応処理	144
8.2 コネクション障害	149
8.2.1 一方送信メッセージ送信時のコネクション障害	149
8.2.2 一方送信メッセージ受信時のコネクション障害	150
8.3 UAP 障害	152
8.3.1 セグメント受信前の UAP 障害	152
8.3.2 セグメント受信後の UAP 障害	152
8.4 UAP 閉塞	154
8.5 入力キュー障害	155
8.6 出力キュー障害	156

付録

付録 A メッセージ送受信の処理の流れ	157
付録 B 障害発生時の処理の流れ	160
付録 C 理由コード一覧	164
付録 D 用語解説	166

索引

索引	169
----	-----

図目次

図 1-1	TP1/NET/HDLC を使用したネットワーク構成例	2
図 1-2	分岐送信をする AP 間通信の例	3
図 1-3	一方受信をする AP 間通信の例	4
図 1-4	TP1/NET/HDLC を組み込んだソフトウェア構成	7
図 2-1	コネクションの確立 (発呼型)	11
図 2-2	コネクションの確立 (着呼型)	12
図 2-3	コネクションの解放 (発呼型)	13
図 2-4	コネクションの切断	14
図 2-5	メッセージの形式の変化	15
図 2-6	メッセージの送信 (単独セグメント)	16
図 2-7	メッセージの送信 (複数セグメント)	17
図 2-8	メッセージの受信	18
図 3-1	処理の流れ	58
図 4-1	アプリケーション名の決定の処理	67
図 4-2	UOC インタフェース用のパラメタとバッファの関係	72
図 4-3	MCF イベント通知時のセグメント構成	81
図 5-1	TP1/NET/HDLC のプロトコル固有定義コマンドの指定順序	102
図 5-2	TP1/NET/HDLC のシステム構成例 (XNF/AS で接続する場合)	114
図 7-1	MCF メイン関数のコーディング概要 (ANSI C, C++ の場合)	137
図 7-2	MCF メイン関数のコーディング概要 (K&R 版 C の場合)	138
図 7-3	MCF メイン関数のディレクトリへの組み込み方法	139
図 7-4	定義オブジェクトファイルの作成方法の概要	141
図 8-1	一方送信メッセージ送信前のコネクション障害	149
図 8-2	一方送信メッセージ消去前のコネクション障害	150
図 8-3	一方送信メッセージ書き込み前のコネクション障害	151
図 8-4	一方送信メッセージ書き込み後のコネクション障害	151
図 8-5	セグメント受信前の UAP 障害	152
図 8-6	セグメント受信後の UAP 障害	153
図 8-7	UAP 閉塞時の障害	154
図 8-8	メッセージ書き込み時の障害	155
図 8-9	メッセージ読み込み時の障害	156
図 A-1	一方送信メッセージの処理の流れ (メッセージ受信)	158
図 A-2	一方送信メッセージの処理の流れ (メッセージ送信)	159

図 B-1	メッセージ送信時接続障害の処理の流れ（回復可能障害の場合）	160
図 B-2	接続切断時の処理の流れ（回復可能障害，手動再確立，および論理端末自動起動の場合）	161
図 B-3	入力メッセージ編集 UOC エラーの処理の流れ	162
図 B-4	出力メッセージ編集 UOC エラーの処理の流れ	163

表目次

表 1-1	TP1/NET/HDLC で適用する伝送動作モード	4
表 1-2	TP1/NET/HDLC が適用する AP 間通信のプロトコル	7
表 2-1	論理端末，メッセージ，アプリケーションの型，UAP インタフェース，および通信形態の関係	15
表 3-1	メッセージ送受信のライブラリ関数（C 言語）	20
表 3-2	メッセージ送受信の文（COBOL 言語）	20
表 3-3	メッセージ送受信の通信文（データ操作言語）	21
表 4-1	TP1/NET/HDLC が通知する MCF イベントの種類	80
表 4-2	COBOL 言語の MCF イベント情報の内容（ERREVT1）	88
表 4-3	COBOL 言語の MCF イベント情報の内容（ERREVT2）	89
表 4-4	COBOL 言語の MCF イベント情報の内容（ERREVT3）	90
表 4-5	COBOL 言語の MCF イベント情報の内容（ERREVTA）	92
表 4-6	COBOL 言語の MCF イベント情報の内容（SCMPEVT）	93
表 4-7	COBOL 言語の MCF イベント情報の内容（CERREVT）	94
表 4-8	COBOL 言語の MCF イベント情報の内容（COPNEVT，CCLSEVT）	95
表 5-1	MCF で使用する定義ファイル	98
表 5-2	TP1/NET/HDLC 固有の定義の一覧	100
表 6-1	TP1/NET/HDLC の運用コマンド	120
表 8-1	コネクション障害発生時の処理	144
表 8-2	受信スケジュール関係の障害発生時の処理	146
表 8-3	送信スケジュール関係の障害発生時の処理	147
表 8-4	UAP 障害発生時の処理	148
表 8-5	MCF の障害発生時の処理	148
表 C-1	ERREVT2 の理由コード一覧	164
表 C-2	CERREVT の理由コード一覧	164

1

概要

TP1/NET/HDLC 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 HDLC 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/HDLC 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

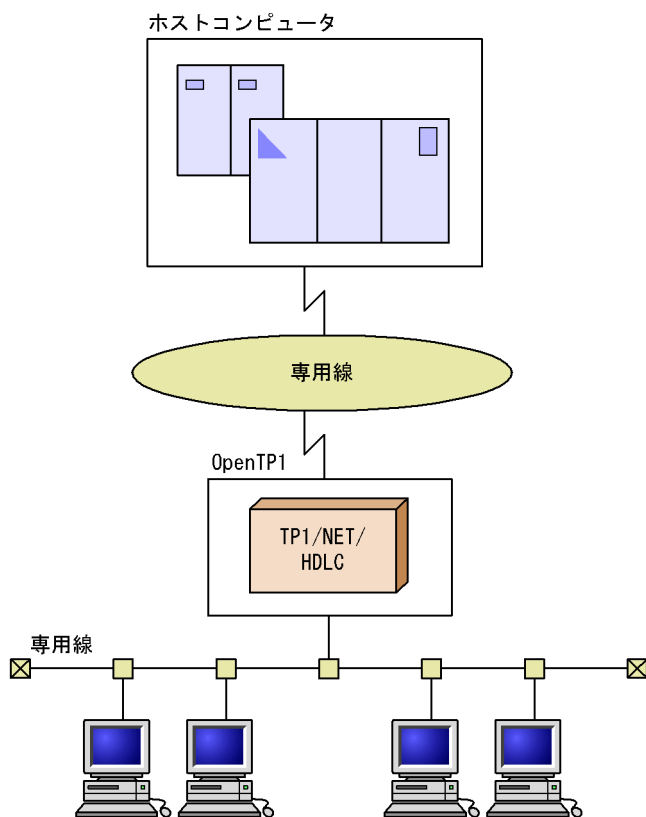
1.3 ソフトウェアの構成

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送受信のことです。TP1/NET/HDLC は、HDLC プロトコルによって AP 間通信をするプログラムです。TP1/NET/HDLC 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、その結果を送信したりできます。

TP1/NET/HDLC を使用したネットワーク構成例を次の図に示します。

図 1-1 TP1/NET/HDLC を使用したネットワーク構成例



1.2 AP 間通信の形態

AP 間通信を使用すると、自システムで発生したトランザクションを通信相手システムで処理したり、その結果を受信したりできます。また、通信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、その結果を送信したりできます。

1.2.1 通信形態

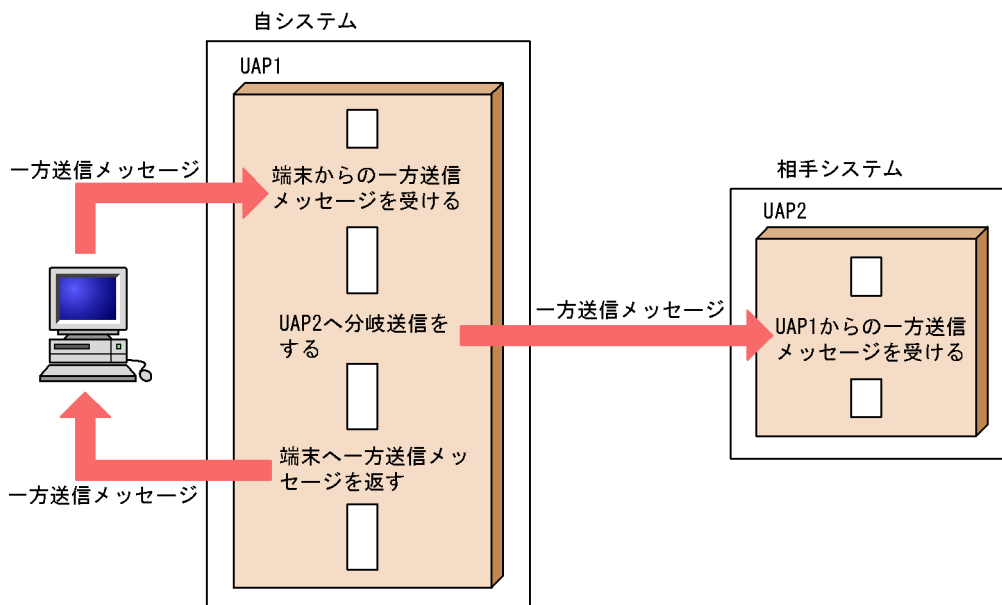
TP1/NET/HDLC を使用した AP 間通信の形態には、分岐送信形態と一方受信形態があります。

(1) 分岐送信形態

分岐送信形態とは、メッセージの送受信処理中に、別のシステムに対して分岐したメッセージを送信する通信形態です。図 1-2 の場合、相手システムへは分岐送信でメッセージを送信し、端末へは一方送信でメッセージを返しています。端末とのメッセージ送受信で使用するメッセージ、および相手システムに分岐するメッセージを、一方送信メッセージといいます。

分岐送信をする AP 間通信の例を次の図に示します。

図 1-2 分岐送信をする AP 間通信の例



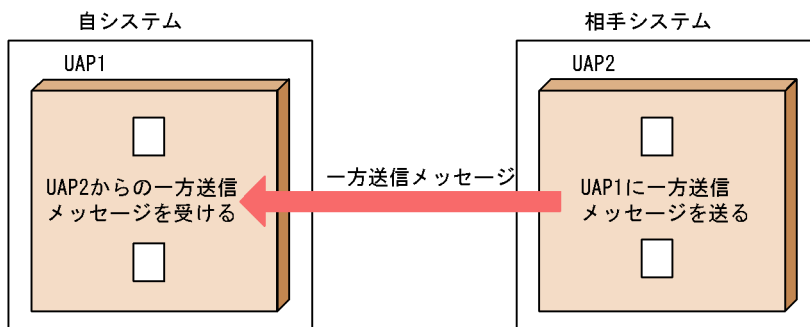
(2) 一方受信形態

一方受信形態とは、通信相手システムから一方送信メッセージを受信する通信形態です。

1. 概要

一方受信をする AP 間通信の例を次の図に示します。

図 1-3 一方受信をする AP 間通信の例



1.2.2 適用範囲

(1) 伝送動作モード

TP1/NET/HDLC で適用する伝送動作モードを次の表に示します。

どのモードを適用するかは、通信管理 (XNF/AS) のシステム定義で指定します。

表 1-1 TP1/NET/HDLC で適用する伝送動作モード

伝送動作モード	説明	適用の有無
HDLC-NRM (正規応答モード)	1次局と2次局が通常のデータ送受信を行うモードです。2次局はコマンドを受信したときに、レスポンスを1次局に送信できます。	
HDLC-ABM (非同期平衡モード)	複合局は必ずこのモードで動作する必要があります。相手複合局の許可に関係なく相互にコマンド、レスポンスを送信できます。	
HDLC-ARM (非同期応答モード)	このモードに設定された2次局は、コマンドに関係なくいつでもレスポンスを1次局に送信できます。	×

(凡例)

: 適用します。

× : 適用しません。

1次局、2次局および複合局の意味は、それぞれ次のとおりです。

- 1次局
制御局です。1次局から2次局へ送信される情報をコマンド (指令) と呼びます。
- 2次局
従属局です。2次局からの応答情報はすべてレスポンス (応答) と呼びます。
- 複合局

1 次局と 2 次局の両方の機能を持つ局です。

(2) 通信規約

TP1/NET/HDLC は、相手システムとの特殊な約束ごとの処理や、メッセージの重複、欠落などのチェックはしません。したがって、AP 間通信をするときには、相手システムの UAP と自システムの UAP との間で、これらの処理に関する規約をあらかじめ決めておく必要があります。

1.3 ソフトウェアの構成

TP1/NET/HDLC 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/HDLC を使用した MCF を実現するための前提プログラムと、TP1/NET/HDLC 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/HDLC 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 5L

- P-1M64-2131 uCosminexus TP1/Server Base 07-00 以降
- P-1M14-511 XNF/AS/BASE
- P-F1M14-5111 XNF/AS/WAN
- P-F1M14-5119 XNF/AS/HDLC

注

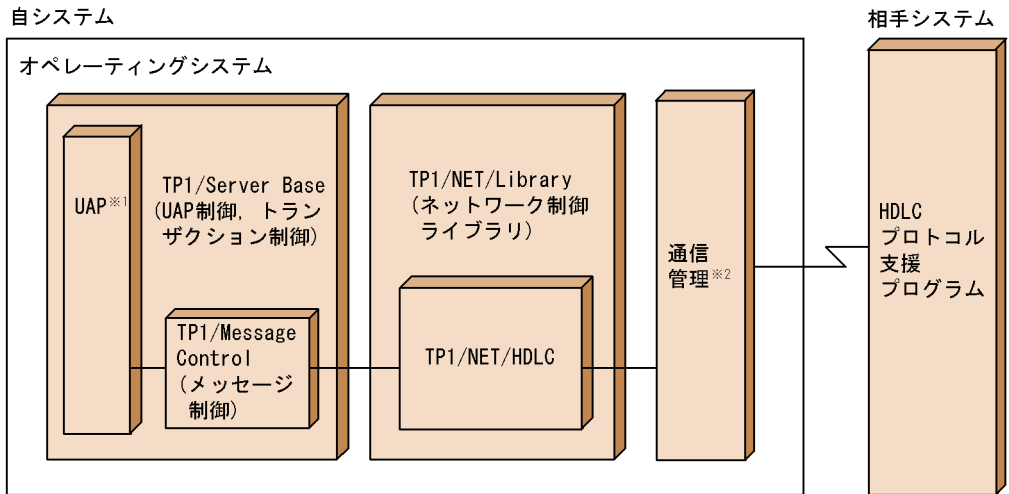
前提となる XNF のバージョンは、AIX 5L V5.1 以降に対応するバージョン以降です。

1.3.2 ソフトウェア構成の例

TP1/NET/HDLC 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。

また、TP1/NET/HDLC が AP 間通信で使用するプロトコル、および主な通信相手プログラムを表 1-2 に示します。

図 1-4 TP1/NET/HDLC を組み込んだソフトウェア構成



注 1

TP1/NET/HDLC 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。

注 2

EP8000 シリーズでは、XNF/AS を使用します。

表 1-2 TP1/NET/HDLC が適用する AP 間通信の protocols

プロトコル	主な通信相手
HDLC プロトコル	HDLC プロトコルに対応するシステム (プログラム)

2

機能

TP1/NET/HDLC は AP 間通信を提供し，メッセージの送受信をします。この章では，コネクションの確立や論理端末の端末タイプなどの AP 間通信の仕組み，およびメッセージの送受信について説明します。

2.1 AP 間通信の仕組み

2.2 AP 間通信メッセージの送受信

2.1 AP 間通信の仕組み

TP1/NET/HDLC の 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

TP1/NET/HDLC は、通信相手システムとの間で、論理的な通信路（コネクション）を確立してメッセージを送受信します。

（1）コネクションの確立

TP1/NET/HDLC からコネクションを確立します。コネクションの確立方法には、発呼型と着呼型の二つがあります。

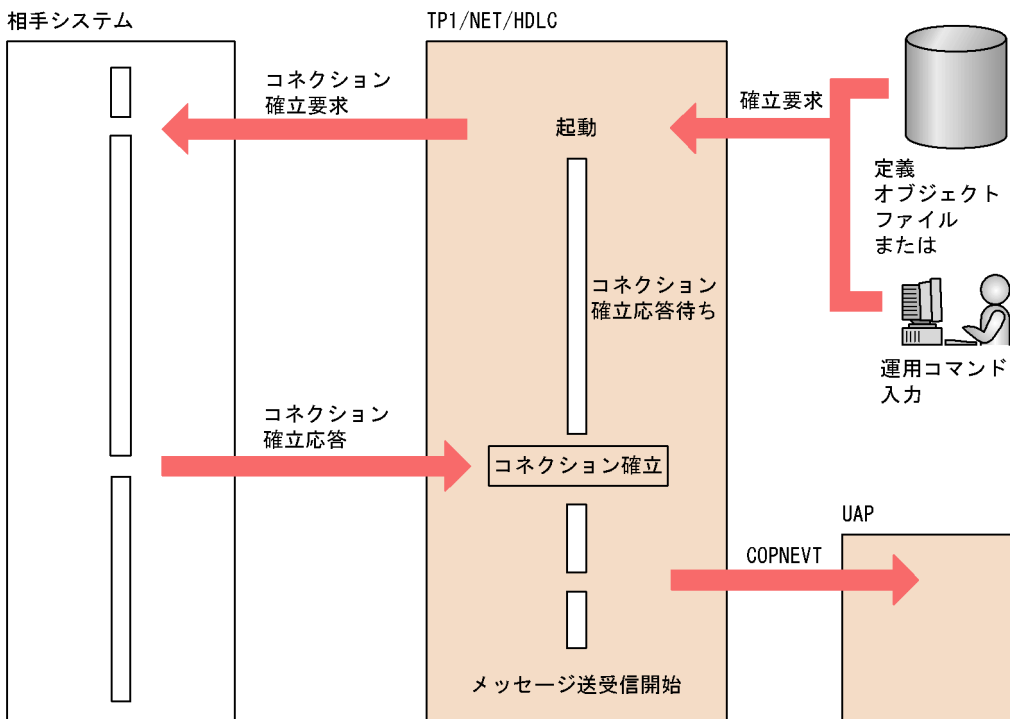
（a）発呼型（1次局側）

自システムからコネクションの確立を要求する方法です。

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入力によって、TP1/NET/HDLC は相手システムにコネクションの確立を要求します。相手システムからのコネクション確立応答を受信すると、コネクションが確立し、メッセージの送受信ができる状態になります。TP1/NET/HDLC は、状態通知イベント（COPNEVT）によって、UAP にコネクションの確立を通知します。

発呼型のコネクションの確立方法を次の図に示します。

図 2-1 コネクションの確立（発呼型）



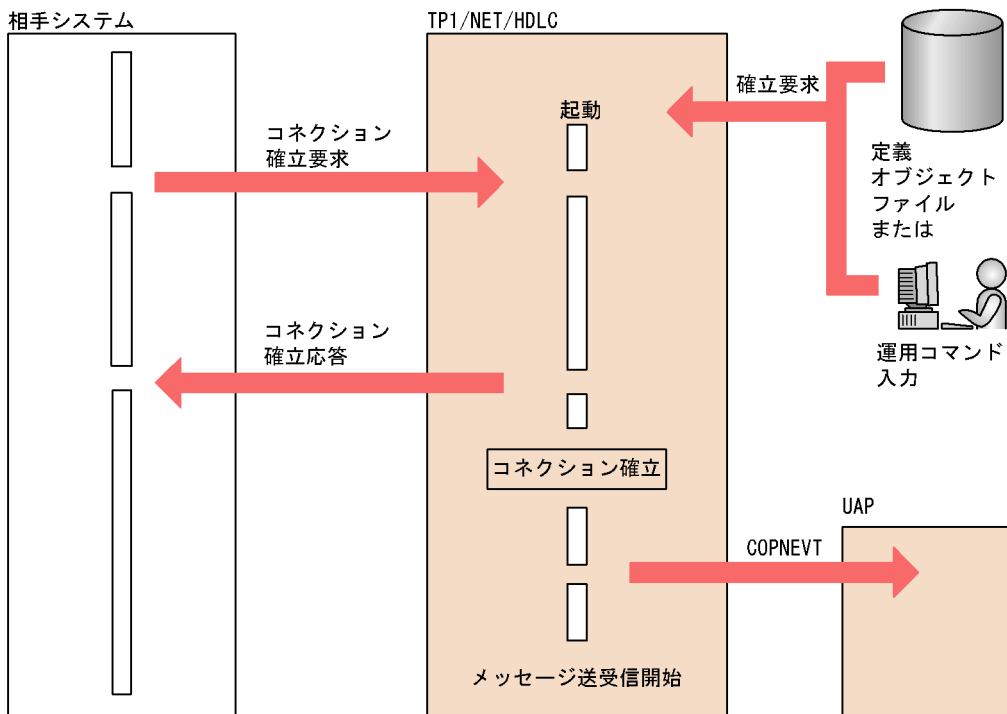
(b) 着呼型（2次局側）

相手システムからのコネクションの確立要求を受け付ける方法です。

オンライン開始時、定義オブジェクトファイルの内容、または運用コマンドによって TP1/NET/HDLC を起動し、相手システムからのコネクション確立要求を待ちます。確立要求を受けると、TP1/NET/HDLC は相手システムにコネクション確立応答を返します。応答を返した時点でコネクションが確立し、メッセージの送受信ができる状態になります。TP1/NET/HDLC は、状態通知イベント（COPNEVT）によって、UAP にコネクションの確立を通知します。

着呼型のコネクションの確立方法を次の図に示します。

図 2-2 コネクションの確立（着呼型）

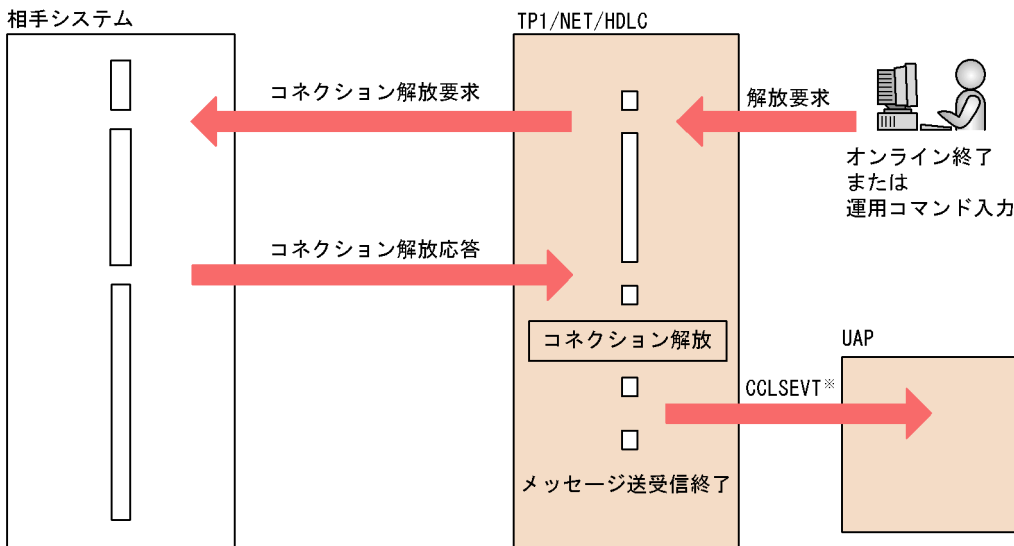


(2) コネクションの解放

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/HDLC はコネクションを解放します。解放時には、確立時の型が引き継がれます。発呼型で確立したときは自システムから解放要求をしてコネクションを解放します。着呼型で確立したときは相手システムの解放要求によってコネクションを解放します。TP1/NET/HDLC は、状態通知イベント (COLSEVT) 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。

発呼型で確立したコネクションの解放を次の図に示します。

図 2-3 コネクションの解放（発呼型）



注

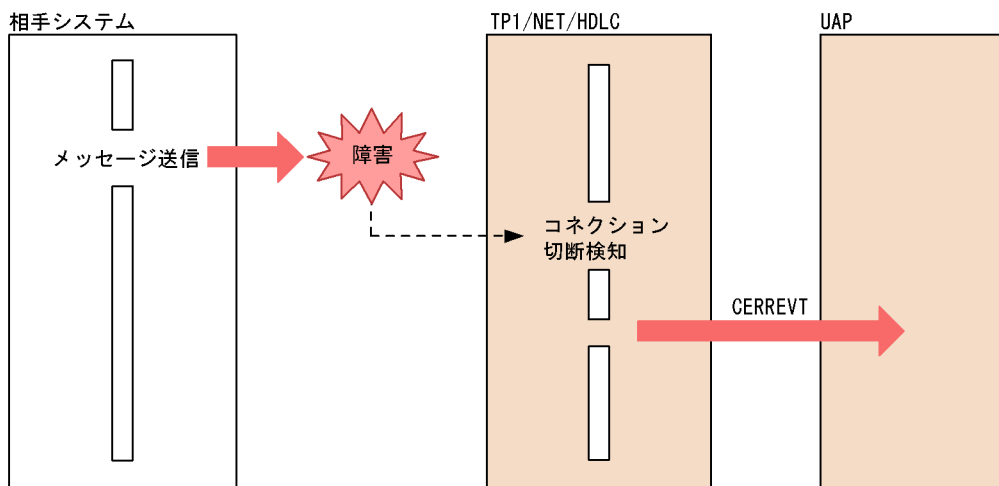
オンライン終了時は通知されません。

(3) コネクションの切断

コネクションに障害が発生すると、コネクションが切断され、メッセージが送受信できない状態になります。コネクションが切断された場合、TP1/NET/HDLC は障害通知イベント (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、「8.2 コネクション障害」を参照してください。

コネクションの切断を次の図に示します。

図 2-4 コネクションの切断



2.1.2 コネクションと論理端末の関係

TP1/NET/HDLC で扱う定義で、ユーザが意識するものとしては、コネクションと論理端末があります。コネクションは TP1/NET/HDLC が通信管理を介して、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します。論理端末は TP1/NET/HDLC と UAP との通信接点です。コネクションと論理端末の指定を対応させると、自システムと相手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。

2.1.3 論理端末とアプリケーションの型の関係

TP1/NET/HDLC で扱う論理端末の端末タイプは、any (任意型) です。この端末タイプを指定すると、TP1/NET/HDLC で使用するすべての通信形態に対応できます。

アプリケーションは、ユーザが送受信データの中で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。TP1/NET/HDLC で扱うアプリケーションの型は、非応答型 (noans) だけです。

注

アプリケーション名は、メッセージの先頭から空白の手前までの 1 から 8 バイトの英数字です。先頭から 9 バイト目に空白がないとき、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ます。入力メッセージ編集 UOC で決定する場合は、「4.1.1 入力メッセージの編集とアプリケーション名の決定」を参照してください。

論理端末、メッセージ、アプリケーションの型、UAP インタフェース、および通信形態の関係を次の表に示します。

表 2-1 論理端末，メッセージ，アプリケーションの型，UAP インタフェース，および通信形態の関係

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

2.1.4 メッセージとセグメントの関係

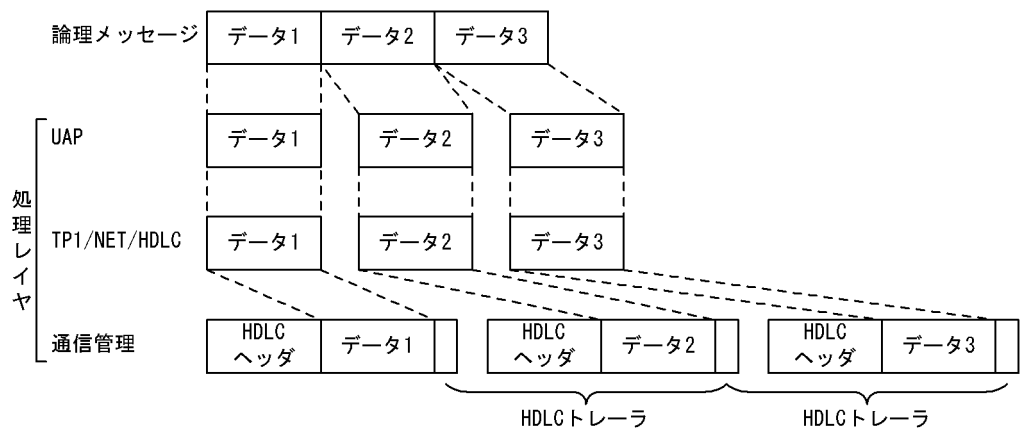
メッセージは，UAP でセグメント単位に分割してシステム間で送受信します。

メッセージの送信時，TP1/NET/HDLC は UAP から受け取ったメッセージをセグメント単位で相手システムへ送信します。送信できる 1 セグメントの長さは，1 から 32000 バイトです。ただし，相手システムによって受信できるセグメント長が異なるので確認してから送信してください。

メッセージの受信時，TP1/NET/HDLC は相手システムから受信したメッセージをセグメント単位で UAP へ渡します。

メッセージを送受信するときのメッセージの形式の変化について次の図に示します。

図 2-5 メッセージの形式の変化



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。セグメントには，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。通常はバッファ形式 1 を使用します。

2.2 AP 間通信メッセージの送受信

TP1/NET/HDLC は、上位プロトコル（ユーザプロトコル）を意識しないでメッセージを送受信できます。TP1/NET/HDLC で扱う AP 間通信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は、メッセージの送信時（分岐送信および一方送信）、メッセージの受信時（一方受信）に使用します。

2.2.1 メッセージの送信

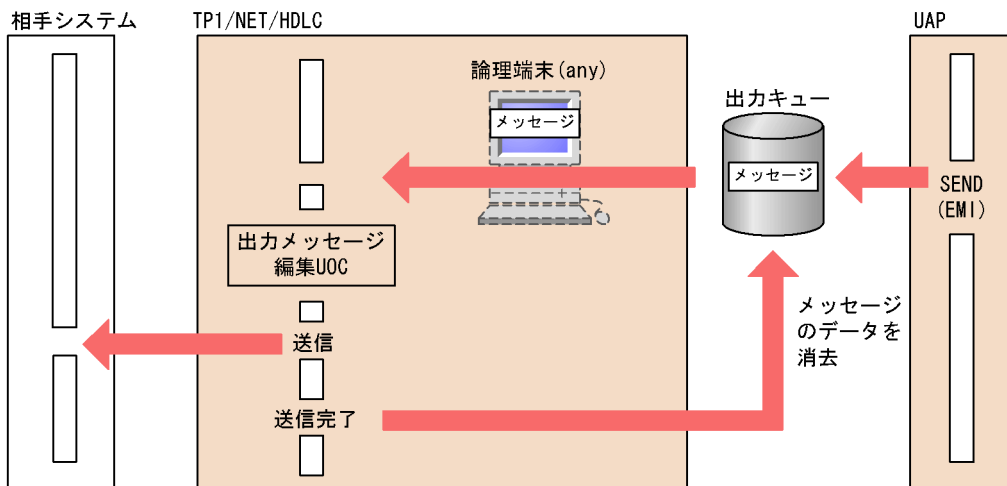
自システムの UAP から相手システムへメッセージを送る形態です。論理端末の端末タイプは any，アプリケーションの型は非応答型（noans）です。

（1）単独セグメントのメッセージ送信

UAP は、SEND 文または SEND 関数で、TP1/NET/HDLC にメッセージの送信要求をします。TP1/NET/HDLC は出力キューに書き込まれたメッセージを編集したあと、相手システムに送信します。送信が完了すると、TP1/NET/HDLC は出力キューにある送信済みメッセージのデータを消去します。

単独セグメントのメッセージの送信を次の図に示します。

図 2-6 メッセージの送信（単独セグメント）



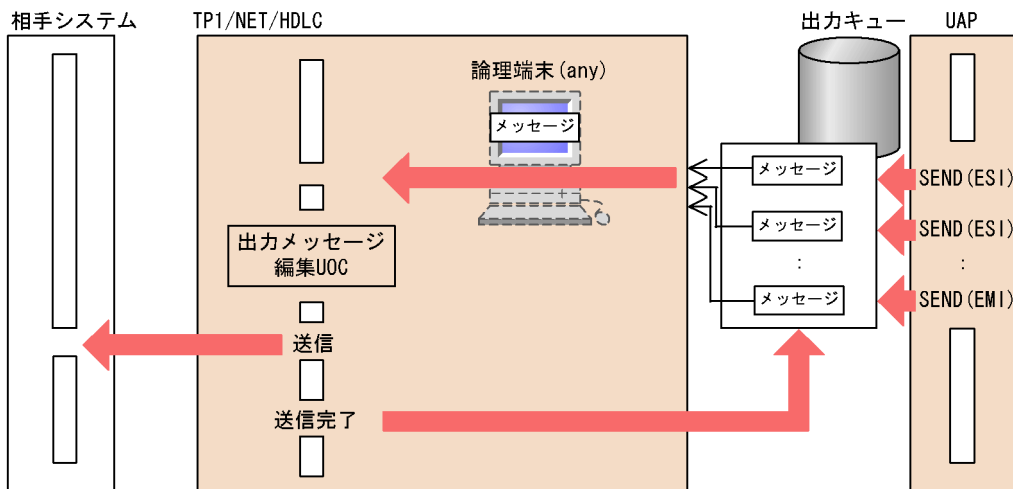
（2）複数セグメントのメッセージ送信

UAP は、SEND 文または SEND 関数（先頭または中間セグメント、および最終セグメントを指定）で、TP1/NET/HDLC にメッセージの送信要求をします。TP1/NET/HDLC は、最終セグメントのメッセージの送信が要求されると、出力キューからメッセージを取り出して編集し、送信します。該当するメッセージの送信が完了すると、TP1/NET/

HDLC は出力キューにあるデータを消去します。

複数セグメントのメッセージの送信を次の図に示します。

図 2-7 メッセージの送信（複数セグメント）



2.2.2 メッセージの受信

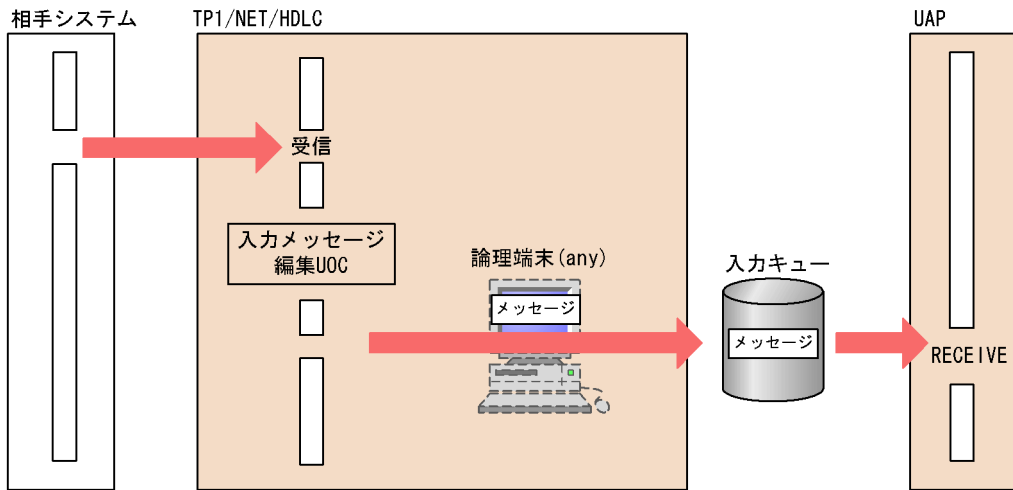
相手システムから一方的に送られたメッセージを受ける形態です。論理端末の端末タイプは any, アプリケーションの型は非応答型 (noans) です。メッセージの受信では, 単独セグメントだけを扱います。

相手システムからのメッセージを受信すると, TP1/NET/HDLC はメッセージを編集し, 入力キューに登録します。同時に, アプリケーション名に対応する UAP を起動して, 編集済みのメッセージを受信します。

メッセージの受信を次の図に示します。

2. 機能

図 2-8 メッセージの受信



3

メッセージ送受信インタフェース

TP1/NET/HDLC 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。UAP は、C 言語または COBOL 言語で作成できます。TP1/NET/HDLC は、UAP の作成、運用を支援しています。この章では、TP1/NET/HDLC に関連するメッセージ送受信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について説明します。

メッセージ送受信インタフェースの一覧

dc_mcf_receive - メッセージの受信 (C 言語)

dc_mcf_resend - メッセージの再送 (C 言語)

dc_mcf_send - 一方送信メッセージの送信 (C 言語)

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

CBLDCMCF('RESEND ') - 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 - 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/HDLC 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 言語でメッセージ送受信をする場合は、OpenTP1 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 3-1 メッセージ送受信のライブラリ関数 (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	一方送信メッセージの再送
dc_mcf_send	一方送信メッセージの送信

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

COBOL 言語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 システムの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 3-2 メッセージ送受信の文 (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 '	一方送信メッセージの再送
	'SEND '	一方送信メッセージの送信

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

データ操作言語 (COBOL 言語) 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。

TP1/NET/HDLC で固有の通信文について、次の表に示します。

表 3-3 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')

その他の通信文については、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。

dc_mcf_receive - メッセージの受信 (C 言語)

形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

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

機能

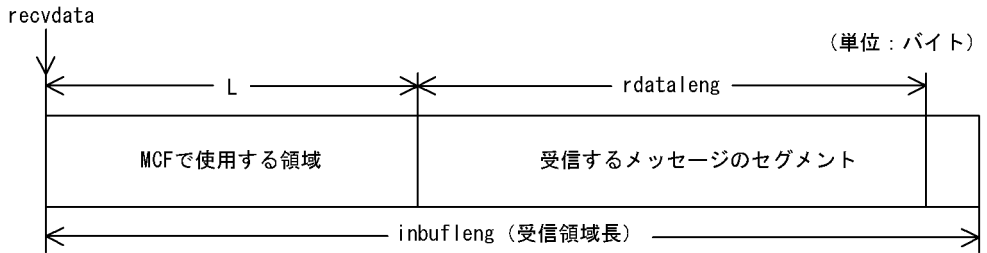
論理端末に届いたメッセージのうち、一つのセグメントを受信します。セグメントの数だけ dc_mcf_receive 関数を呼び出すと、一つの論理メッセージを受信できます。

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

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

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

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



UAP で値を設定する引数

action

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

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

DCMCFFRST

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

DCMCFSEG

中間セグメントまたは最終セグメントを受信する場合に設定します。

DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

commform

DCNOFLAGS を設定します。

termnam

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

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。

3. メッセージ送受信インタフェース

dc_mcf_receive - メッセージの受信 (C 言語)

recvdata

セグメントを受信する領域を設定します。相手システムから送信されるセグメントの最大長については、5 章の「mcftalcn (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。

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。

rdata leng

受信したセグメントの長さが返されます。

time

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

リターン値

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

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

dc_mcf_resend - メッセージの再送 (C 言語)

形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

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

機能

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

- 出力先の論理端末名称
- メッセージ通番
- メッセージ種別 (一般一方送信または優先一方送信)

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、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、メッセージキューの大きさの定義は余裕を持った値を指定してください。

UAP で値を設定する引数

action

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

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

DCMCFSEQ

再送するメッセージに出力通番を付け直す場合に設定します。

DCMCFNSEQ

再送するメッセージに出力通番を付け直さない場合に設定します。

DCMCFNORM

一般の一方送信メッセージとして再送する場合に設定します。

DCMCFPRIO

優先の一方送信メッセージとして再送する場合に設定します。

DCMCFLAST

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

commform

一方送信を示す、DCMCFOUT を設定します。

rtermnam

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

resv01

ヌル文字を設定します。

oseqid

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

DCMCFRID_NORM

一般の一方送信メッセージを対象とする場合に設定します。

DCMCFRID_PRIO

優先の一方送信メッセージを対象とする場合に設定します。

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

3. 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)

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視されず。

otermnam

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

resv02, resv03, resv04

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_NOMSG	該当するメッセージがありません。
DCMCFRTN_BUF_SHORT	再送するメッセージのセグメントの長さが、UAP 共通定義 (mcfmuap -e) で指定した値を超えています。
DCMCFRTN_71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	MCF が終了処理中のため、メッセージの再送を受け付けられません。
DCMCFRTN_71003	メッセージキューが満杯です。
DCMCFRTN_71004	メッセージキューから取り出したメッセージを格納するバッファ (作業領域) を、メモリ上に確保できませんでした。
DCMCFRTN_71108	メッセージを再送しようとしたますが、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。
DCMCFRTN_72000	<p>< MHP の実行でリターンした場合 ></p> <ul style="list-style-type: none"> 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、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。

リターン値	意味
	< SPP の実行でリターンした場合 > トランザクションでない SPP の処理から, dc_mcf_resend 関数を呼び出しています。
DCMCFRTN_72001	rtermnam または otermnam に設定した論理端末名称が間違っています。 dc_mcf_resend 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72016	action に設定したメッセージ種別 (DCMCFNORM または DCMCFPRIO) の値が間違っています。 action に設定した値が間違っています。 oseqid に設定した値が間違っています。 opcd に設定した値が間違っています。 resv01, resv02, resv03, または resv04 に設定した値が間違っています。 引数に設定した値に間違いがあります。
DCMCFRTN_72017	action に設定した出力通番の要否 (DCMCFSEQ または DCMCFNSEQ) の値が間違っています。
DCMCFRTN_72024	commform に設定した値が間違っています。
上記以外	プログラムの破壊などによる, 予期しないエラーが発生しました。

注意事項

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

-e オプション

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

-l オプション

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

dc_mcf_send - 一方送信メッセージの送信 (C 言語)

形式

ANSI C, C++ の形式

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

K&R 版 C の形式

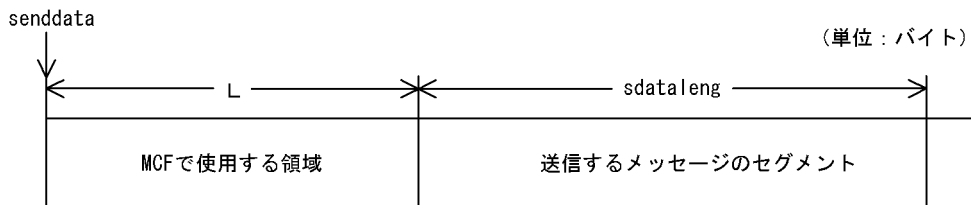
```
#include <dcmcf.h>
int dc_mcf_send(action, commform, termnam, resv01, senddata,
               sdataleng, resv02, opcd)

DCLONG      action;
DCLONG      commform;
char        *termnam;
char        *resv01;
char        *senddata;
DCLONG      sdataleng;
char        *resv02;
DCLONG      opcd;
```

機能

相手システムへ送る一方送信メッセージのうち、一つのセグメントを送信します。セグメントの数だけ dc_mcf_send 関数を呼び出すと、一つの論理メッセージを送信できます。

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



UAP で値を設定する引数

action

メッセージの最終セグメントを送信するかどうか、出力通番を付けるかどうか、使用するバッファ形式、および送信完了イベントを通知するかどうかを次の形式で設定します。


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

DCMCFESI

先頭セグメントまたは中間セグメントを送信する場合に設定します。

DCMCFEMI

最終セグメントを送信する場合に設定します。

メッセージが単一セグメントの場合も、DCMCFEMI を設定します。

メッセージの送信の終了を連絡するために、最後は必ずこの値を設定してください。

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。

DCMCFSEQ

出力通番が必要な場合に設定します。

DCMCFNSEQ

出力通番が必要ない場合に設定します。

DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

DCMCFSEVT

送信完了イベントを通知する場合に設定します。ただし、これらのイベントを処理する MHP を MCF アプリケーション定義で指定していない場合は無効です。メッセージが複数のセグメントで構成される場合、先頭セグメント送信時の設定が有効になります。

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。論理端末名称の長さは最大 8 バイトです。論理端末名称の後ろにはヌル文字を付けてください。

resv01

ヌル文字を設定します。

senddata

送信するセグメントの内容を設定した領域を設定します。一つのセグメントで 32000 バ

3. メッセージ送受信インタフェース

dc_mcf_send - 一方送信メッセージの送信 (C 言語)

イトまで送信できます。

メッセージの送信の終了を連絡する場合で、セグメントの内容がないときも、必ず設定してください。

sdataleng

送信するセグメントの長さを設定します。メッセージの送信の終了を連絡する場合で、セグメントの内容がないときは、0 を設定してください。

resv02

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	sdataleng に 32000 バイトを超える値を設定しています。
DCMCFRTN_71003	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。
	メッセージキューが満杯です。
DCMCFRTN_71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRTN_71108	メッセージを送信しようとしたが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_send 関数を呼び出しています。
	< SPP の実行でリターンした場合 > トランザクションでない SPP の処理から、dc_mcf_send 関数を呼び出しています。
DCMCFRTN_72001	termnam に設定した論理端末名称が間違っています。
	dc_mcf_send 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72005	先頭セグメントまたは中間セグメントを送信する dc_mcf_send 関数で、長さが 0 バイトのセグメントを送信しています。
DCMCFRTN_72016	action に設定したメッセージ種別 (DCMCFNORM) 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。

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

リターン値	意味
	opcd に設定した値が間違っています。
	resv01 または resv02 に設定した値が間違っています。
	引数に設定した値に間違いがあります。
DCMCFRTN_72017	action に設定した出力通番の要否 (DCMCFSEQ または DCMCFNSEQ) の値が間違っています。
DCMCFRTN_72024	commform に設定した値が間違っています。
DCMCFRTN_72026	action に設定したセグメント種別 (DCMCFESI または DCMCFEMI) の値が間違っています。
DCMCFRTN_72041	単一セグメント送信時, sdataleng に 0 以下の値を設定しています。
上記以外	プログラムの破壊などによる, 予期しないエラーが発生しました。

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

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

機能

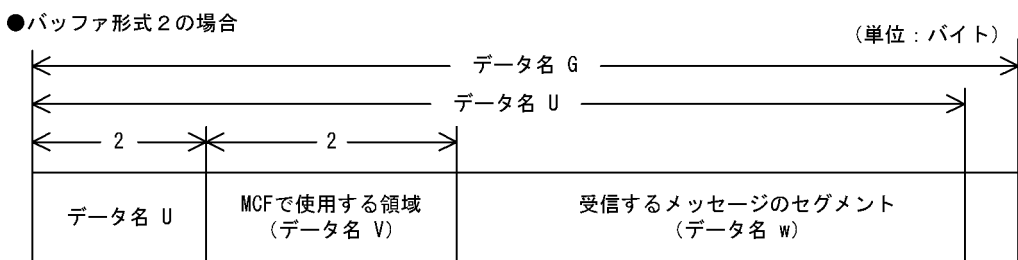
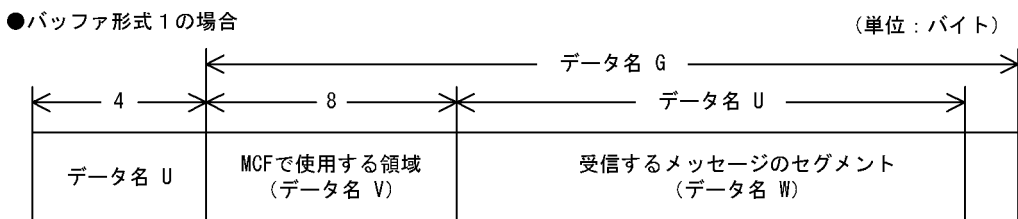
論理端末に届いたメッセージのうち、一つのセグメントを受信します。セグメントの数だけ CBLDCMCF('RECEIVE ') を呼び出すと、一つの論理メッセージを受信できません。

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/HDLC 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。



UAP で値を設定するデータ領域

データ名 A

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

データ名 C

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

VALUE 'FRST'

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

VALUE 'SEG '

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。

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

データ名 G

セグメントを受信する領域の長さを設定します。

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

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白

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は、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 Y

受信したセグメントの内容が返されます。相手システムから送信されるセグメントの最大長については、5 章の「mcftalccn (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。

3. メッセージ送受信インタフェース

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

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する CBLDCMCF('RECEIVE ') を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、データ名 C に「VALUE 'SEG'」を設定して CBLDCMCF('RECEIVE ') を呼び出してください。
71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する CBLDCMCF('RECEIVE ') を呼び出しています。直前に呼び出した CBLDCMCF('RECEIVE ') でメッセージはすべて受信しました。このステータスコードが返されたあとに、再び CBLDCMCF('RECEIVE ') を呼び出した場合は、ステータスコード 72000 が返されます。
71002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。
71108	プロセスのローカルメモリが不足しています。
72000	<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>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。

3. 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)

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

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

UAP で値を設定するデータ領域

データ名 A

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

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H, データ名 I

空白を設定します。

データ名 J

一般として再送するか優先として再送するかを設定します。

VALUE 'NORM'

一般の一方送信メッセージとして再送する場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージとして再送する場合に設定します。

空白

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。

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

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

データ名 Q, データ名 R

空白を設定します。

データ名 S

MCF で使用する領域です。

データ名 T

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

データ名 U

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

3. 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 '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となります。

データ名 W

最終出力通番を持つメッセージを再送するかどうかを設定します。

VALUE 'LAST'

最終出力通番を持つメッセージを再送する場合に設定します。

この値を設定した場合は、データ名 V に設定した値は無効となります。

空白

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。
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。

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

注意事項

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

-e オプション

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

3. 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)

-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。この場合は、どのメッセージを再送するか保証できません。

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

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

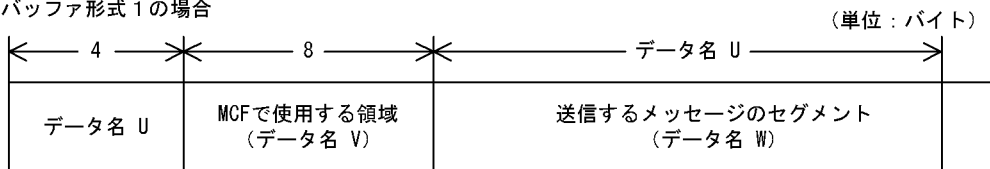
機能

相手システムへ送る一方送信メッセージのうち、一つのセグメントを送信します。セグメントの数だけ CBLDCMCF('SEND ') を呼び出すと、一つの論理メッセージを送信できます。

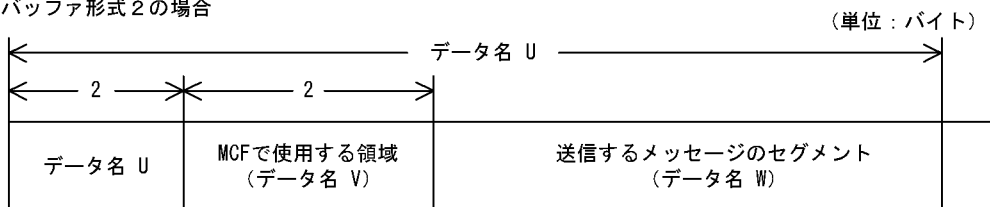
セグメントを送信する領域 (一意名 3 で示す領域) の形式を示します。

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

●バッファ形式 1 の場合



●バッファ形式 2 の場合



UAP で値を設定するデータ領域

データ名 A

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

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H

メッセージの最終セグメントを送信するかどうかを設定します。次のどちらかを設定し
 ます。

VALUE 'ESI '

先頭セグメントまたは中間セグメントを送信する場合に設定します。

VALUE 'EMI '

最終セグメントを送信する場合や、メッセージが単一セグメントの場合に、
 「VALUE 'EMI '」を設定します。

メッセージの送信の終了を連絡するために、最後は必ずこの値を設定してください。

データ名 I

空白を設定します。

データ名 J

一般の一方送信メッセージとして送信することを示す VALUE 'FORM' を設定します。また、空白を設定してもかまいません。

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' (出力通番を付けない) が設定されます。

データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白

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

データ名 N

MCF で使用する領域です。

データ名 O

一方送信を示す「VALUE 'OUT」を設定します。

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

データ名 P

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

データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 U

【バッファ形式 1 の場合】 PIC 9(9)

送信するセグメントの長さを設定します。メッセージの送信の終了を連絡する
場合で、セグメントの内容がないときは、0 を設定してください。

【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。メッセージの送信の終了を連絡する
場合で、セグメントの内容がないときは、4 を設定してください。

データ名 V

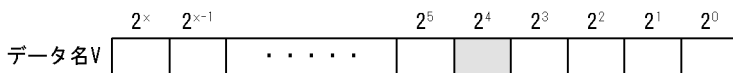
【バッファ形式 1 の場合】 PIC X(8)

【バッファ形式 2 の場合】 PIC X(2)

送信完了イベントを通知させる場合、領域の 2^4 ビットに 1 を設定します。ただし、
これらのイベントを処理する MHP を MCF アプリケーション定義で指定していない
場合は無効です。メッセージが複数のセグメントで構成される場合、先頭セグメン
ト送信時の設定が有効になります。

送信完了イベントを通知させない場合、領域の 2^4 ビットに 0 を設定します。

データ名 V とビットの設定位置の関係を次に示します。



x の値は、バッファ形式 1 の場合は 63、バッファ形式 2 の場合は 15 です。

(凡例) : 送信完了通知イベントの要否を設定するビット

データ名 W

送信するセグメントの内容を設定します。先頭セグメントまたは中間セグメントの送信
後、メッセージの送信の終了を連絡する場合にも必ず設定してください。一つのセグメン
トで 32000 バイトまで送信できます。

OpenTP1 から値が返されるデータ領域

データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	データ名 U に 32000 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する CBLDCMCF('RECEIVE') を呼び出す前に、CBLDCMCF('SEND') を呼び出しています。
	< SPP の実行でリターンした場合 > トランザクションでない SPP の処理から、CBLDCMCF('SEND') を呼び出しています。
72001	データ名 P に設定した論理端末名称が間違っています。
	CBLDCMCF('SEND') を呼び出せない論理端末を設定しています。
72005	バッファ形式 1 の先頭セグメントまたは中間セグメント送信時、データ名 U に 0 以下の値を設定しています。または、バッファ形式 2 の先頭セグメントまたは中間セグメント送信時、データ名 U に 4 以下の値を設定しています。
72016	データ名 N またはデータ名 T に設定した値が間違っています。
	データ名 J に設定した値が間違っています。
	データ名 M1 に設定した値が間違っています。
	データ名 M7 に設定した値が間違っています。
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
72020	データ名 I に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72026	データ名 H に設定した値が間違っています。

3. メッセージ送受信インタフェース

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

ステータスコード	意味
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 の単一セグメント送信時、データ名 U に 0 以下の値を設定しています。または、バッファ形式 2 の単一セグメント送信時、データ名 U に 4 以下の値を設定しています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

RECEIVE - メッセージの受信 (データ操作言語)

形式

DATA DIVISION (通信記述項) の指定

```
CD 通信記述名  
FOR INPUT  
[STATUS KEY IS データ名1]  
[SYMBOLIC TERMINAL IS データ名2]  
[MESSAGE DATE IS データ名3]  
[MESSAGE TIME IS データ名4].
```

PROCEDURE DIVISION (通信文) の指定

```
RECEIVE 通信記述名  
[FIRST] SEGMENT  
INTO 一意名1.
```

機能

次に示す CALL インタフェースの機能を実現します。

- 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

INPUT を指定します。

STATUS KEY 句

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

SYMBOLIC TERMINAL 句

入力元の論理端末名称を参照するデータ項目を指定します。

MESSAGE DATE 句

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

MESSAGE TIME 句

メッセージを受信した時刻を参照するデータ項目を指定します。HHMMSS00 (HH:時

3. メッセージ送受信インタフェース
 RECEIVE - メッセージの受信 (データ操作言語)

MM : 分 SS : 秒 00 は固定) の形式で参照できます。

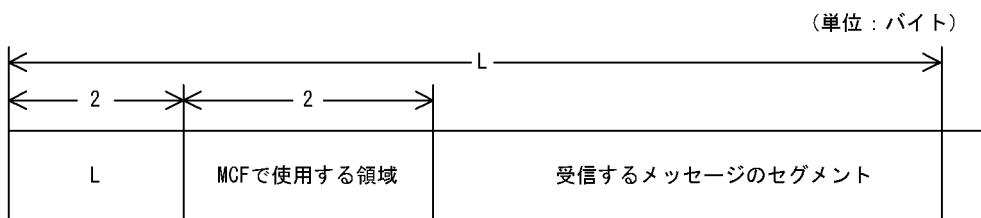
通信文に指定する項目

FIRST

先頭セグメントを受信するときに指定します。

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。相手システムから送信されるセグメントの最大長については、5章の「mcftalccn (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。一意名 1 の形式を次に示します。



ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する RECEIVE 文を、2 回以上実行しています。中間セグメントまたは最終セグメントを受信する場合は、FIRST を指定しないで RECEIVE 文を実行してください。
71001	最終セグメントを受信したあとで、その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 文でそのメッセージはすべて受信しました。 このステータスコードが返されたあとに、再び次のメッセージを受信する 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。
72000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none"> • 先頭セグメントを受信する RECEIVE 文を実行する前に、中間セグメント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 71000 が返されたあとで、RECEIVE 文を実行していません。 < SPP の実行でリターンした場合 > SPP では RECEIVE 文を実行できません。
72001	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。

3. メッセージ送受信インタフェース
RECEIVE - メッセージの受信 (データ操作言語)

ステータスコード	意味
	RECEIVE 文を実行できない論理端末を設定しています。
72013	一意名 1 の L の指定値を超えるセグメントを受信しました。一意名 1 の L の指定値を超えた部分は切り捨てられました。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72036	一意名 1 の L の指定値が不足しています。5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

SEND - メッセージの送信 (データ操作言語)

形式 1 (セグメントの内容を設定して送信する場合)

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 | 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 {ESI | EMI | 一意名2}} .
```

形式 2 (セグメントの内容を設定しないで、メッセージの送信の終了だけ連絡する場合)

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}  
  {SWITCHING MODE IS {NORMAL | データ名7}} .
```

PROCEDURE DIVISION (通信文) の指定

```
SEND 通信記述名 WITH EMI .
```

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

OUTPUT を指定します。

STATUS KEY 句

ステータスコードを受け取りたい場合に設定します。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設定します。

SYNCHRONOUS MODE 句

非同期型でメッセージを送信することを、次のどちらかで指定します。

ASYNC

非同期型のメッセージ送信

一方送信メッセージ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'...非同期型のメッセージ送信

SYNCHRONOUS MODE 句は省略してもかまいません。

SWITCHING MODE 句

一般の一方送信メッセージを送信することを、次のどちらかで指定します。

NORMAL

一般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'1'...一般の一方送信メッセージ

SWITCHING MODE 句は省略してもかまいません。

DETAIL MODE 句

一方送信メッセージの場合に、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'1'...出力通番を付けます。

'1'...出力通番を付けません。

省略した場合は、出力通番を付けません。

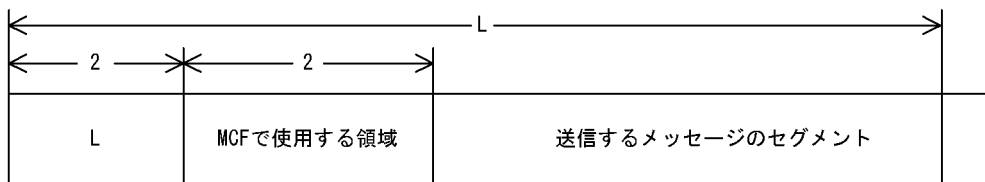
通信文に指定する項目

一意名 1

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

3. メッセージ送受信インタフェース
 SEND - メッセージの送信（データ操作言語）

（単位：バイト）



WITH 句

メッセージの最終セグメントを送信するかどうかを指定します。非同期型のメッセージ送信の場合だけ指定します。

ESI

先頭セグメントまたは中間セグメントの送信

EMI

最終セグメントまたは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

'1'...ESI（先頭セグメントまたは中間セグメントの送信）

'2'...EMI（最終セグメントまたは単一セグメントの送信）

省略した場合は、EMI（最終セグメントまたは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。
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	<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 RECEIVE 文を実行する前に、SEND 文を実行しています。 非応答型の MHP から応答メッセージを送信しています。

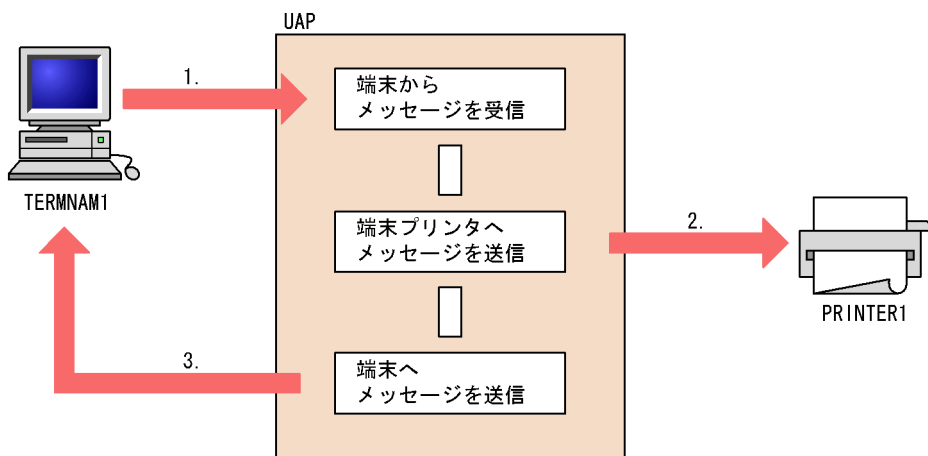
3. メッセージ送受信インタフェース
SEND - メッセージの送信 (データ操作言語)

ステータスコード	意味
	<p>< SPP の実行でリターンした場合 ></p> <ul style="list-style-type: none"> 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。 SPP では, 応答メッセージを送信する SEND 文を実行できません。
72001	<p>SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。</p> <p>SEND 文を実行できない論理端末を設定しています。</p>
72005	<p>先頭セグメントまたは中間セグメント送信時, 一意名 1 の L に 4 以下の値を設定しています。</p>
72008	<p>応答メッセージの最終セグメントを送信する 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。</p> <p>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。</p>
72011	<p>MAP NAME 句は設定できません。</p>
72013	<p>一意名 3 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 3 の L の指定値を超えた部分は切り捨てられました。</p>
72017	<p>DETAIL MODE 句に設定した値が間違っています。</p>
72018	<p>SWITCHING MODE 句に設定した値が間違っています。</p>
72020	<p>SYNCHRONOUS MODE 句に設定した値が間違っています。</p>
72024	<p>FOR 句に設定した値が間違っています。</p>
72026	<p>WITH 句に設定した値が間違っています。</p>
72036	<p>一意名 3 の L の指定値が不足しています。5 バイト以上の領域を確保してください。</p>
72037	<p>BEFORE 句に設定した値が間違っています。</p>
72041	<p>単一セグメント送信時, 一意名 1 の L に 4 以下の値を設定しています。</p>
上記以外	<p>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。</p>

ユーザアプリケーションプログラム作成例

UAP の作成例として、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 (ANSI C, C++, K&R 版 C), COBOL 言語, およびデータ操作言語で示します。

図 3-1 処理の流れ



TP1/NET/HDLC では、このコーディング例を次のファイルで提供しています。

C 言語 (ANSI C, C++) のコーディング例

`/BeTRAN/examples/mcf/HDLC/aplib/ansi/ap.c`

C 言語 (K&R 版 C) のコーディング例

`/BeTRAN/examples/mcf/HDLC/aplib/c/ap.c`

COBOL 言語のコーディング例

`/BeTRAN/examples/mcf/HDLC/aplib/cobol/ap.cbl`

データ操作言語のコーディング例

`/BeTRAN/examples/mcf/HDLC/aplib/dml/ap.c`

C 言語 (ANSI C, C++)

```
#include <string.h>
#include <stdlib.h>
#include <dcmcf.h>
void ex_uapl()
{
    char termnam[9];
    int   rtn_code;
    char  recvdata[2048];
    DCLONG rdataleng;
    DCLONG time;
```

```

DCLONG inbufleng;
char  senddata[512] ;
DCLONG sdataleng ;
static char resv01[9]="¥0" ;
static char resv02[9]="¥0" ;
static char resv03[9]="¥0" ;
char *workadd = recvdata ;

rtn_code = dc_mcf_receive(DCMCFRST,DCNOFLAGS,termnam,resv01,
                          workadd,&rdataleng,2048,&time) ;
if(rtn_code != DCMCFRTN_00000) {
    abort();
}

/*****
/*
/*   User Service Part.
/*
/*
/*****

sdataleng = strlen(senddata) + 4 ;
rtn_code = dc_mcf_send(DCMCFEMI,DCMCFOUT,"PRINTER1",
                      resv01,senddata,sdataleng,resv02,DCNOFLAGS) ;
if(rtn_code != DCMCFRTN_00000) {
    abort() ;
}

sdataleng = strlen(senddata) + 4 ;
rtn_code = dc_mcf_send(DCMCFEMI,DCMCFOUT,"TERMNAM1",
                      resv01,senddata,sdataleng,resv02,DCNOFLAGS) ;
if(rtn_code != DCMCFRTN_00000) {
    abort() ;
}
}

```

C 言語 (K&R 版 C)

```

#include <dcmcf.h>
void ex_uapl()
{
    char termnam[9] ;
    int   rtn_code ;
    char  recvdata[2048] ;
    DCLONG rdataleng ;
    DCLONG time ;
    DCLONG inbufleng;
    char  senddata[512] ;
    DCLONG sdataleng ;
    static char resv01[9]="¥0" ;
    static char resv02[9]="¥0" ;
    static char resv03[9]="¥0" ;
    char *workadd = recvdata ;

    rtn_code = dc_mcf_receive(DCMCFRST,DCNOFLAGS,termnam,resv01,
                              workadd,&rdataleng,2048,&time) ;

```

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

```
    if(rtn_code != DCMCFRTN_00000) {
        abort();
    }

/*****
/*
/*   User Service Part.
/*
/*
*****/

    sdataleng = strlen(senddata) + 4 ;
    rtn_code = dc_mcf_send(DCMCFEMI,DCMCFOUT,"PRINTER1",
                          resv01,senddata,sdataleng,resv02,DCNOFLAGS) ;
    if(rtn_code != DCMCFRTN_00000) {
        abort() ;
    }

    sdataleng = strlen(senddata) + 4 ;
    rtn_code = dc_mcf_send(DCMCFEMI,DCMCFOUT,"TERMNAM1",
                          resv01,senddata,sdataleng,resv02,DCNOFLAGS) ;
    if(rtn_code != DCMCFRTN_00000) {
        abort() ;
    }
}
```

COBOL 言語

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

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

```

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 TEXT-TYPE1 PIC X(4).
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 'ASYN'.
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.
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 TEXT-TYPE2 PIC X(2).
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).

```

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

```

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.
*
CALL 'CBLDCMCF' USING SND1 CD2 DATA2.
*
CALL 'CBLDCMCF' USING SND2 CD3 DATA2.

EXIT PROGRAM.

```

データ操作言語

```

*
*****
*           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-DATA             PIC X(1024).
*
*****
*           一方送信, および分岐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-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.
*
*****
*           メッセージの送信の通信記述項           *
*****
*
CD SEND-IO
  FOR OUTPUT
  STATUS KEY IS          SE-STATUS-IO
  SYMBOLIC TERMINAL IS  SE-TERMNAM-IO.
*
*
PROCEDURE DIVISION.
*
*****
*           メッセージの受信の通信記述項           *
*****
*
  RECEIVE RECV-INF
    FIRST SEGMENT
    INTO RECV-AREA.
*
*****
*           分岐メッセージの送信の通信記述項           *
*****

```

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

```
*
MOVE 'PRINTER1' 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の通信記述項           *
*****
*
MOVE 'TERMNAM1' TO SE-TERMNAM-IO.
SEND SEND-IO
FROM SEND-IO-AREA
WITH EMI.
EXIT PROGRAM.
```

4

ユーザOWNコーディング， MCF イベントインタフェース

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

4.1 ユーザOWNコーディングインタフェース

4.2 MCF イベントインタフェース

4.1 ユーザOWNコーディングインタフェース

メッセージ送受信の 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）

TP1/NET/HDLC 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
- 送信メッセージ通番編集 UOC

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

4.1.1 入力メッセージの編集とアプリケーション名の決定

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

UOC は，UAP を起動するメッセージの最終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には，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（mcftalccn-e）で，メッセージ編集用バッファグループ番号を定義します。

（1）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

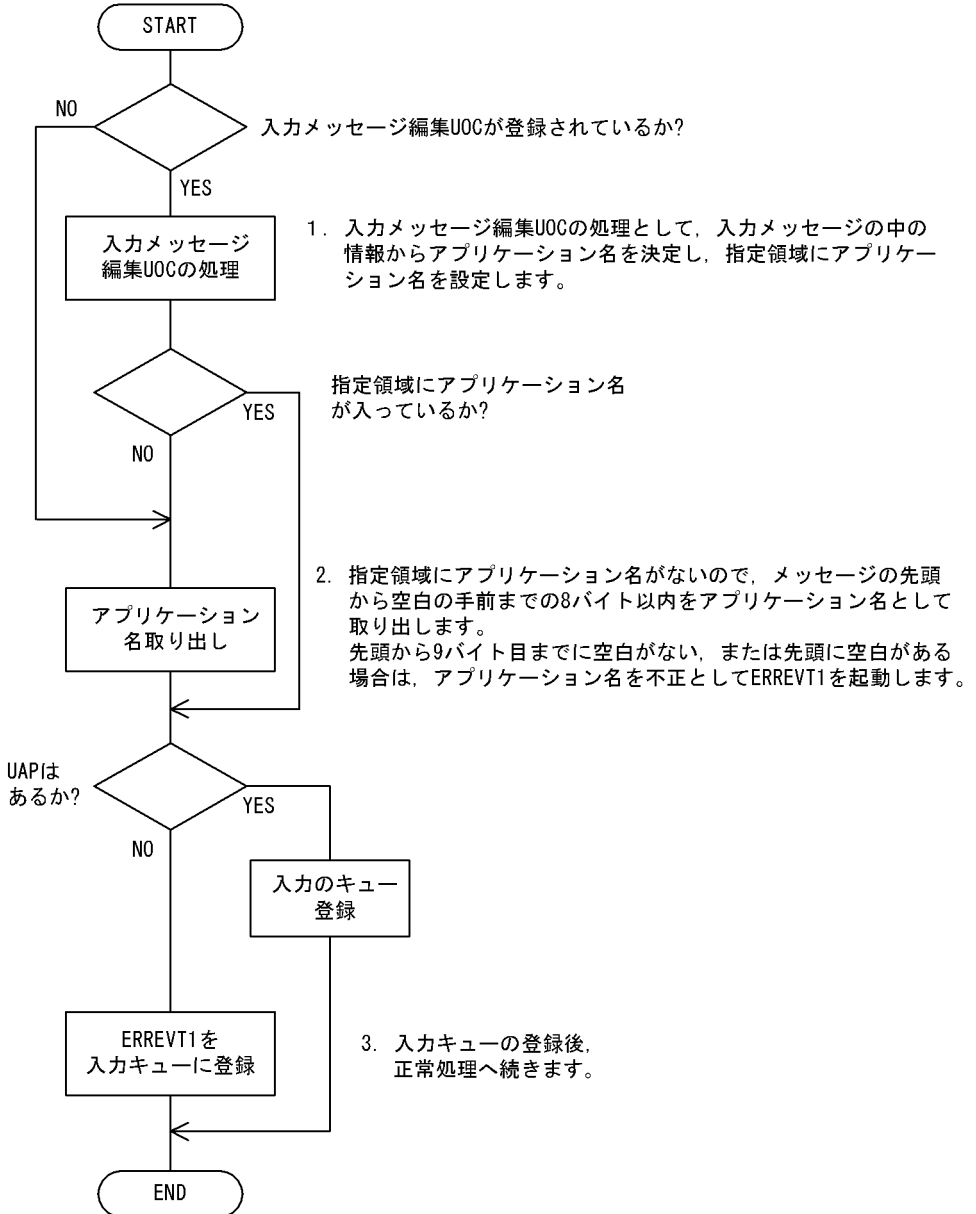
該当する MCF に入力メッセージ編集 UOC が登録されている場合，論理メッセージの受信と同時にアプリケーション名を決定できます。

UOC でアプリケーション名を決定する場合，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' がいないときは, アプリケーション名を不正とし, ERREVT1 を起動します。

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 4-1 アプリケーション名の決定の処理



4. ユーザOWNコーディング，MCF イベントインタフェース

(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) を起動します。

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 (_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 で設定したパラメタ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) を起動します。

(5) OpenTP1 への組み込み方法

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

4.1.2 入力メッセージ編集 UOC インタフェース

入力メッセージ編集 UOC は，次に示す形式で呼び出します。

(1) 形式

ANSI C, C++ の形式

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

DCLONG uoc_func(dcmcf_uoc_min_n *parm)
```

K&R 版 C の形式

```
#include <dcmcf.h>
#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) パラメータの内容

(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 - 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称
    char reserve2[7];         ...予備
    char *pro_indv_ifa;        ...MCF使用領域
    DCLONG rtn_detail;        ...詳細リターンコード
    char reserve3[8];         ...予備
} dcmcf_uoc_min_n;
```

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

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

(c) dcmcf_uocbufinf_n (バッファ情報) の内容

```
typedef struct {
    char *buf_adr;            ...バッファアドレス
    DCULONG buf_size;         ...バッファ最大長
    DCULONG seg_size;         ...バッファ使用長
    char reserve1[4];         ...予備
    dcmcfuoc_w_type buff_id;  ...MCF内部情報1
    DCMLONG buff_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, 次の値が設定されます。

4. ユーザOWNコーディング, MCF イベントインタフェース

DCMCF_UOC_PRO_HDL

HDLC プロトコル

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。

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

NULL 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

バッファ情報の数が設定されます。

buf_array

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

(c) dcmcf_uocbufinf_n (バッファ情報)

buf_adr

バッファのアドレスが設定されます。

buf_size

バッファの最大長が設定されます。

seg_size

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

buff_id, buff_addr

MCF で使用するパラメタです。

(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 -19000 ~ -19999 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。

(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。

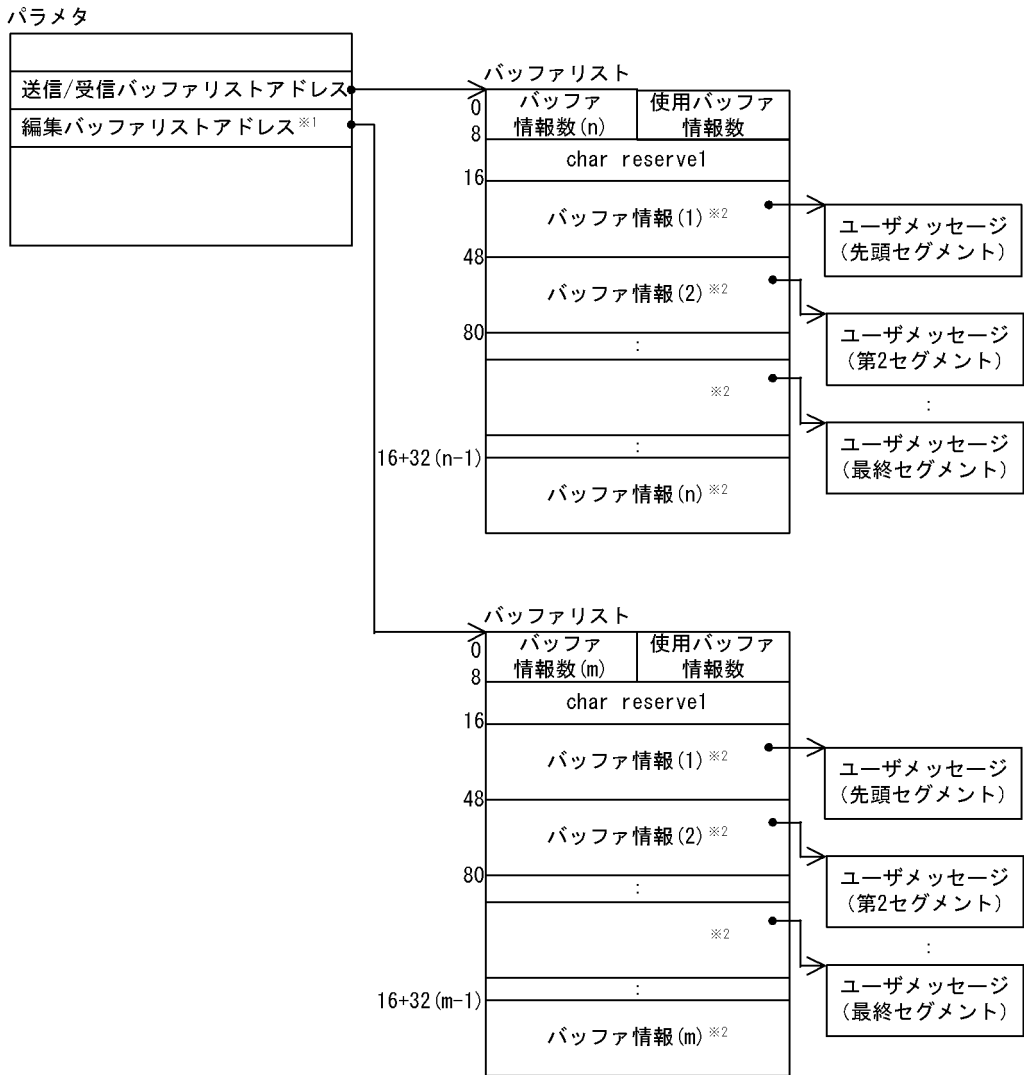
リターン値	意味
DCMCF_UOC_MSG_OK	正常リターン (編集バッファでスケジューリング)
DCMCF_UOC_MSG_OK_RCV	正常リターン (受信バッファでスケジューリング)
DCMCF_UOC_MSG_NG	メッセージ編集エラー

(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。

4. ユーザOWNコーディング, MCF イベントインタフェース

図 4-2 UOC インタフェース用のパラメタとバッファの関係



注 1
mcfaltccn コマンドの -e オプションを指定しない場合, NULL となり, バッファリストとバッファは確保されません。

注 2
バッファ情報は次の形式をしています。

0	バッファ情報
4	バッファアドレス
8	バッファ最大長
12	バッファ使用長
16	予備
28	MCF内部情報
32	予備

4.1.3 出力メッセージの編集

出力メッセージの編集 UOC は、一方送信メッセージの編集をする UOC です。出力メッセージの編集 UOC は、UAP が発行した送信メッセージを相手システムに実際に送信する前に処理するように位置させます。出力キューから全セグメントを読み出すと起動します。

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

(1) 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、出力メッセージの編集処理ができます。

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

(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. ユーザOWNコーディング, MCF イベントインタフェース

(4) OpenTP1 への組み込み方法

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

4.1.4 出力メッセージ編集 UOC インタフェース

出力メッセージ編集 UOC は、次に示す形式で呼び出します。

(1) 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

```
#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;        ...MCF使用領域
    char reserve2[1];         ...予備
    char *pro_indv_ifa;       ...MCF使用領域
    DCLONG rtn_detail;        ...詳細リターンコード
    char reserve3[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

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

(4) MCF が値を設定する項目

(a) `dcmcf_uoc_mout_n`

`pro_kind`

プロトコル種別として, 次の値が設定されます。

`DCMCF_UOC_PRO_HDL`

HDLC プロトコル

`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。

`buflist_adr`

送信用バッファリストのアドレスが設定されます。

`ebuflist_adr`

編集用バッファリストのアドレスが設定されます。

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。

`output_no`

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

`msg_type`

メッセージ種別として, 次の値が設定されます。

'n'

一般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`

MCF で使用するパラメタです。

`pro_indv_ifa`

MCF で使用するパラメタです。

4. ユーザOWNコーディング，MCF イベントインタフェース

(b) `dcmcf_uocbuff_list_n` (バッファリスト), `dcmcf_uocbufinf_n` (バッファ情報)

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

(5) ユーザが値を設定する項目

(a) `dcmcf_uoc_mout_n`

`rtn_detail`

詳細リターンコードを設定します。

このコードは，UOC が `DCMCF_UOC_MSG_NG` をリターンしたときに，MCF に渡されます。MCF は，詳細リターンコードをメッセージログファイルに出力します。詳細リターンコードは -19000 ~ -19999 の範囲で指定してください。

(b) `dcmcf_uocbuff_list_n` (バッファリスト), `dcmcf_uocbufinf_n` (バッファ情報)

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

(6) リターン値

`uoc_func()` は次のコードでリターンしてください。

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

(7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係は，入力メッセージ編集の場合と同じです。「4.1.2(7) パラメタとバッファの関係」を参照してください。

4.1.5 送信メッセージの通番編集

(1) 通番編集

送信メッセージの通番編集 UOC では，受け取った通番を基に，ユーザ独自の処理ができます。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。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，送信メッセージの通番編集 UOC の関数アドレスを登録して

おきます。UAP のメイン関数に登録する `dc_mcf_register` 関数の形式を次に示します。

(a) 形式

ANSI C, C++ の形式

```
#include<dcmcf.h>

int dc_mcf_register(DCLONG flags, DCLONG(*uoc_addr)(DCLONG flags,
char *termname, DCLONG sendno,
DCLONG sendid, DCLONG dataleng,
char *senddata))
```

K&R 版 C の形式

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

(b) ユーザが値を設定する引数

flags

DCMCF_SEND_UOC を指定します。

uoc_addr

flags に対応する UOC のアドレスを指定します。

(c) リターン値

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

(d) メイン関数への登録例

```
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();
}
```

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

4.1.6 送信メッセージの通番編集 UOC インタフェース

送信メッセージの通番編集 UOC は、次に示す形式で、send_uoc 関数として作成します。
なお、UOC の関数名はユーザの任意です。

(1) 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

(2) MCF が値を設定する項目

flags

送信メッセージの通番編集 UOC がいつ呼ばれたかが設定されます。

DCMCF_SEND_DML

メッセージを送信する関数または命令文が呼び出されたときを意味します。

termname

送信先の論理端末名称が設定されます。

sendno

送信メッセージの通番が設定されます。

sendid

送信するメッセージ種別が設定されます。

DCMCF_SEND_NORM

一般の一方送信メッセージ

dataleng

送信メッセージ長が設定されます。

senddata

セグメントの内容が設定されます。

(3) リターン値

リターン値	意味
DC_OK	正常に終了しました。

4.1.7 UOC 作成上の注意事項

UOC 作成上の注意事項を次に示します。

(1) UOC の構造

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

(2) UOC で使用できる関数

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

メモリ操作をする関数

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

時間取得関数

(3) UOC の異常処理

TP1/NET/HDLC 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() を呼び出すと、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 が呼ばれたりしてもよいように作成してください。

(5) UOC の詳細リターンコード

UOC で障害を検出した場合、UOC の種類によっては詳細リターンコード (rtn_detail) が MCF に返されます。MCF は UOC エラーリターン時に、この詳細コードをメッセージログファイルに出力します。なお、詳細リターンコードの値の範囲は、-19000 から -19999 までです。

4.2 MCF イベントインタフェース

TP1/NET/HDLC でメッセージ送受信をすると，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のが MCF イベントの内容でわかります。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編集 UOC は呼び出しません。詳細についてはマニュアル「OpenTP1 プログラム作成の手引」を参照してください。

4.2.1 MCF イベントの種類

TP1/NET/HDLC が通知する MCF イベントの種類を次の表に示します。

表 4-1 TP1/NET/HDLC が通知する MCF イベントの種類

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報告します。
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破棄しました。 <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，または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。 アプリケーションの即時起動時に障害が発生しました。 MHP のアプリケーション，サービスグループがセキュア状態です。 	メッセージを廃棄したことを報告します。
UAP 異常終了通知イベント	ERREVT3	MHP のセグメント受信関数に，セグメントを渡したあとに MHP の異常終了が発生しました。	UAP 異常終了時の対処障害メッセージを送信します。
タイマ起動メッセージ廃棄通知イベント	ERREVT4	アプリケーションのタイマ起動時に障害が発生しました。	メッセージを廃棄したことを報告します。

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	ERREVTn	次の理由で未処理送信メッセージを破棄しました。 <ul style="list-style-type: none"> MCF の正常終了処理時に，未処理送信メッセージの滞留時間監視の時間切れ（タイムアウト）が発生しました。 運用コマンド（mcftdlqle）の入力によって，出力キューが削除されました。 	未処理送信メッセージを廃棄したことを報告します。
送信完了通知イベント	SCMPEVT	メッセージが正常に送信されました。	メッセージが送信され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	コネクションが確立しました。または，論理端末が閉塞解除しました。	コネクションが確立したことまたは論理端末が閉塞解除したことを報告します。
	CCLSEVT	コネクションが正常に解放されました。	コネクションが解放されたことを報告します。

注

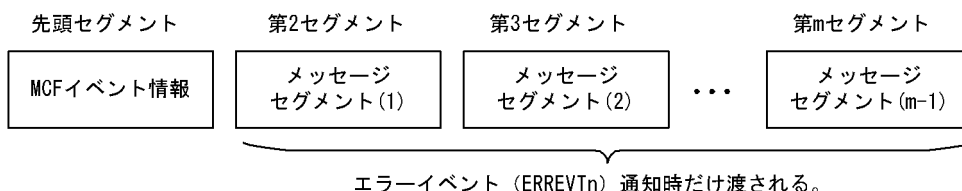
MCF アプリケーション定義（mcfaalcap.g）の recvmsg オペランドに r を指定した場合，または dc_mcf_rollback の action に DCMCFRTRY または DCMCFRRTN を指定した場合は除きます。

4.2.2 MCF イベント通知時のセグメント構成

MCF イベントを MHP に通知する場合，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 セグメント以降に処理できなかったメッセージセグメントを最終セグメントまで設定します。

MCF イベント通知時のセグメント構成を次の図に示します。

図 4-3 MCF イベント通知時のセグメント構成



4. ユーザOWNコーディング，MCF イベントインタフェース

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

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

このため，それ以降の項目の位置にずれがあります。「4.2.4 MCF イベント情報の形式 (COBOL 言語)」のエラーイベントの表では ERREVT1，ERREVT2，ERREVT3，および ERREVT4 のバッファ形式別に位置 (バイト) を分けて説明しています。

なお，ERREVT4 については，マニュアル「OpenTP1 プログラム作成リファレンス」の該当する言語編を参照してください。

4.2.3 MCF イベント情報の形式 (C 言語)

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

エラーイベント (ERREVTn) で使用する構造体は，<dcmcf.h> で定義してあります。なお，dcmpcm_cerrevt は，<dcmpcm.h> で定義されています。<dcmhds.h> の前に <dcmpcm.h> を取り込んでおいてください。また，dc_mcf_evtheader は，<dc_mcf.h> で定義されています。<dc_mcf.h>，<dcmpcm.h>，<dcmhds.h> の順に取り込んでください。

(1) MCF イベントの共通ヘッダ

(a) 形式

```
struct dc_mcf_evtheader {
    char mcfevt_name[9];           ...MCFイベントコード
    char le_name[16];             ...入力元論理端末名称
                                   (ERREVT1, ERREVT2, ERREVT3の場合)
                                   出力先論理端末名称
                                   (ERREVT4の場合)
    char cn_name[9];              ...コネクション名
    unsigned char format_kind;    ...MCF使用領域
    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

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

ただし，ERREVT2 または ERREVT3 で，次に示す場合は，'*' 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
- 上記のエラー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

ERREVT_A では, メッセージを出力する論理端末名称が設定されます。
 SCMPEVT では, メッセージ送信が完了した論理端末名称が設定されます。
 CERREVT, COPNEVT, または CCLSEVT では無効です。

cn_name

コネクション名が設定されます。

ただし, 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
- 上記のエラー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イベント共通ヘッダ
    char reserve01[12] ;      ...予備
    char reserve02[10] ;      ...予備
    char reserve03[2] ;       ...予備
    char ap_name[10] ;        ...アプリケーション名
                                (メッセージに対応する
                                アプリケーション名)
    char reserve04[2] ;       ...予備
};
```

(b) MCF イベントとして設定される項目

ap_name

次に示すどちらかが設定されます。

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

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

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イベント共通ヘッダ
```

4. ユーザOWNコーディング, MCF イベントインタフェース

```
char reserve01[12] ;      ...予備
char reserve02[10] ;      ...予備
char reserve03[2] ;       ...予備
char ap_name[10] ;        ...アプリケーション名
                           (メッセージに対応する
                           アプリケーション名)
short reason_code ;      ...理由コード
};
```

(b) MCF イベントとして設定される項目

ap_name

エラーになった UAP のアプリケーション名が設定されます。

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

MHP 以外から送信された場合は, ヌル文字が設定されます。

reason_code

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

(4) ERREVT3

(a) 形式

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

(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) ERREVT

(a) 形式

```
struct dc_mcf_evta_type {
    struct dc_mcf_evtheader  evtheader ;
                                ...MCFイベント共通ヘッダ
    char reserve01[12] ;        ...予備
    char map_name[10] ;        ...MCF使用領域
    char reserve03[2] ;        ...予備
    char ap_name[10] ;         ...アプリケーション名
                                (正常終了したメッセージの
                                アプリケーション名)
    char reserve04[2] ;        ...予備
    char reserve05[32] ;       ...予備
    char reserve06[32] ;       ...予備
    char reserve07[36] ;       ...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) SCMPEVT

(a) 形式

```
struct dc_mcf_scmpevt_type {
    struct dc_mcf_evtheader  evtheader ;
                                ...MCFイベント共通ヘッダ
    DCLONG output_no ;        ...出力通番
    char map_name[9] ;        ...MCF使用領域
    char msg_type ;           ...メッセージ種別
    char reason_code ;        ...起動理由
    char reserve01[9] ;       ...予備
    char pro_indv_inf[16] ;    ...MCF使用領域
    char reserve02[16];       ...予備
};
```

(b) MCF イベントとして設定される項目

output_no

出力通番が設定されます。通番がない場合は，-1 が設定されます。

msg_type

メッセージ種別が，次に示す値で設定されます。

4. ユーザOWNコーディング, MCF イベントインタフェース

n

一般メッセージ

p

優先メッセージ

reason_code

起動理由が, 次に示す値で設定されます。

メッセージが正常に送信されました。

(7) CERREVT

(a) 形式

```
typedef struct {
    struct dc_mcf_evtheader header ;
                                ...MCF イベント共通ヘッダ
    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
    DCLONG retry_code ;        ... 自動再確立有無識別
} dcmPCM_cerrevt ;
typedef struct {
    dcmPCM_cerrevt comheader    ... プロトコル共通ヘッダ
    DCLONG err_dctreason ;       ... 切断理由コード
    char reserve1[32] ;          ... 予備
} dcmhds_cerrevt ;
```

(b) 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。

(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, 次に示す値で設定されます。

(00000000)₁₆

回復可能

(00000010)₁₆

回復不可能

retry_code

障害発生後，自動再確立するかどうか，次に示す値で設定されます。

(00000000)₁₆

自動再確立します。

(00000010)₁₆

自動再確立をしません。

err_dctreason

XNF/AS/HDLC から通知された接続の切断理由コードが設定されます。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。

この領域は，障害要因コードが接続障害（err_fact で示すコードが (00000030)₁₆）のときだけ有効です。

(8) COPNEVT , CCLSEVT

(a) 形式

```

typedef struct {
    struct dc_mcf_evtheader  header ;
                                ...MCFイベント共通ヘッダ
    char reserve1[56] ;        ...予備
} dcmhds_statevt ;

```

(b) MCF イベントとして設定される項目

ありません。

4.2.4 MCF イベント情報の形式 (COBOL 言語)

COBOL 言語の場合はセグメントの並びとして渡されます。COBOL 言語の UAP の場合，MCF イベントの内容を表 4-2 ~ 表 4-8 に示します。このとき，データ形式 1 とデータ形式 2 で先頭の有無とこれによる位置 (バイト) が異なるため，データ形式別に位置 (バイト) を分けて説明します。

4. ユーザOWNコーディング，MCF イベントインタフェース

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

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

(凡例)

- : 該当しません。または、使用されません。

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

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

4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
メッセージが入力された時刻	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

理由コードの詳細については，「付録 C 理由コード一覧」を参照してください。

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベントコード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT3 を示す '3 ' が設定されます。
入力元論理端末名称	12	8	8	英数字	メッセージを入力した論理端末名称です。次に示す場合は， '*' が設定されます。 <ul style="list-style-type: none"> • SPP がアプリケーション起動をした MHP で，障害が発生した場合 • 上記の障害発生時に MCF イベントとして起動された MHP が，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。

4. ユーザOWNコーディング, MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
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	-	-
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。

4. ユーザOWNコーディング，MCF イベントインタフェース

表 4-5 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です。 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	-	-
予備	72	68	8	-	-
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	-	-

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
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 4-6 COBOL 言語の MCF イベント情報の内容 (SCMPEVT)

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
長さ (形式 1 のときだけ)	0	-	2	-	MCF が使用します。
予備 (形式 1 のときだけ)	2	-	2	-	-
イベントコード	4	0	8	英数字	'SCMPEVT' が設定されます。
出力先論理端末名称	12	8	8	英数字	メッセージの送信が完了した論理端末名称です。
予備	20	16	8	-	-
コネクション名	28	24	8	英数字	コネクション名です。
メッセージが入力された日付	36	32	8	外部 10 進数字	送信完了イベントが通知された日付です。YYYYMMDD の形式です。 YYYY : 西暦の年 MM : 月 DD : 日
メッセージが入力された時刻	44	40	8	外部 10 進数字	送信完了イベントが通知された時刻です。HHMMSS00 の形式です。 HH : 時 MM : 分 SS : 秒 00 は固定です。
出力通番	52	48	4	2 進数字	出力通番です。出力通番がない場合は、(FFFFFFF) ₁₆ が設定されます。
出力マップ名	56	52	8	英数字	MCF が使用します。
メッセージ種別	64	60	1	英数字	メッセージ種別が設定されます。 n : 一般メッセージ p : 優先一方送信メッセージ
理由コード	65	61	1	英数字	理由コードが設定されます。 : 送信が正常に完了しました。
予備	66	62	9	-	-

4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
プロトコル個別 情報	75	71	16	英数字	MCF が使用します。
予備	91	87	19	-	-

(凡例)

- : 該当しません。または，使用されません。

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

項目	位置 (バ イト)	長さ (バ イト)	属性	内容
イベントコード	0	8	英数字	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	障害の発生した論理端末名称が設定さ れます。コネクション障害時は無効で 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	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。
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	発生した障害が，自動回復できるかが 設定されます。 (00000000) ₁₆ : 回復可能障害 (00000010) ₁₆ : 回復不可能障害
自動再確立有無識別 ²	64	4	2 進数字	障害発生後，コネクションを自動再確 立しているかが設定されます。 (00000000) ₁₆ : 再確立実行中 (00000010) ₁₆ : 再確立未実行

項目	位置 (バイト)	長さ (バイト)	属性	内容
切断理由コード	68	4	2進数字	XNF/AS/HDLC から通知されたコネクションの切断理由コードが設定されます。 障害レベルがコネクション障害 (00000030) ₁₆ のときだけ有効です。 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。
予備	72	32	-	-

(凡例)

- : 該当しません。または，使用されません。

注

相手システムから UERR を受信した場合で，UERR の利用者情報が設定されていたとき，利用者情報を CERREVT の第 2 セグメントに設定します。

注 1

理由コード 1，および理由コード 2 は，「付録 C 理由コード一覧」を参照してください。

注 2

コネクションの自動再起動指定によって，TP1/NET/HDLC が再確立している場合は，再確立実行中が設定されます。コネクションが手動起動指定の場合は，再確立未実行が設定されます。

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

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	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	論理端末の閉塞解除を通知する COPNEVT の場合は，閉塞解除した論理端末名称が設定されます。 コネクション確立を通知する COPNEVT および CCLSEVT では無効です。
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。

4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)	長さ (バイト)	属性	内容
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	-	-

(凡例)

- : 該当しません。または，使用されません。

5

システム定義

この章では、OpenTP1 のシステム定義の中での TP1/NET/HDLC 固有のシステム定義について説明します。また、自システム内にある通信管理プログラムと関連づける内容、およびシステム定義例について説明します。

TP1/NET/HDLC の定義の概要

TP1/NET/HDLC 固有のシステム定義の種類

mcftalccn (コネクション定義の開始)

mcftalcle (論理端末定義)

mcftalced (コネクション定義の終了)

システムサービス情報定義

システムサービス共通情報定義

MCF 定義オブジェクトの生成

自システムの通信管理プログラムと関連づける内容

定義例

TP1/NET/HDLC の定義の概要

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

OpenTP1 のネットワークコミュニケーション定義の中での定義

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

使用する定義ファイル

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

表 5-1 MCF で使用する定義ファイル

定義の種類	定義のソースファイル	定義の内容
MCF マネージャ定義	MCF マネージャ定義ソースファイル	MCF 全体の実行環境
MCF 通信構成定義	共通定義ソースファイル	プロトコルごとの実行環境
	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指定して作成します。それらの中には、プロトコルで共通のものと、プロトコルに固有のものがあります。表 5-1 の定義の中で、TP1/NET/HDLC に固有の定義があるものを次に示します。

- MCF マネージャ定義
- MCF 通信構成定義

この章では、TP1/NET/HDLC 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。プロトコル共通の定義については、マニュアル「OpenTP1 システム定義」を参照してください。ただし、`mcftbuf` (バッファグループ定義) の `length`、`count` オペランドの指定値については、「`mcftalecn` (コネクション定義の開始)」の注意事項に記載してあります。

TP1/NET/HDLC 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/HDLC 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/HDLC を組み込む方法については、「7. 組み込み方法」を参照してください。

通信定義の内容の関連づけ

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

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

5. システム定義

TP1/NET/HDLC 固有のシステム定義の種類

TP1/NET/HDLC 固有のシステム定義の種類

MCF 通信構成定義のうち、TP1/NET/HDLC に固有の定義の一覧を次の表に示します。

表 5-2 TP1/NET/HDLC 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		プロトコル共通のコマンドだけで指定できます。共通のコマンドについては、マニュアル「OpenTP1 システム定義」を参照してください。			
	プロトコル固有定義	mcftalcen (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 hdlcs	
			-g	sndbuf	メッセージ送信用バッファ ループ番号	符号なし整数 ((1 ~ 512))
				rcvbuf	メッセージ受信用バッファ ループ番号	符号なし整数 ((1 ~ 512))
			-e	msgbuf	メッセージ編集用バッファ ループ番号	符号なし整数 ((1 ~ 512))
				count	メッセージ編集用バッファ数	符号なし整数 ((1 ~ 131070))
			-i	-	システム開始時、および再開時にコネクションを自動的に確立するかどうか	auto 《manual》
			-m	mode	使用する通信管理	xnfas
			-z	vslot	仮想スロット番号	符号なし整数 ((1 ~ 1000))
	-x	-	システム再開時にコネクションを自動的に確立するかどうか	auto 《manual》		

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》		
		-a	aretrycnt	コネクションの確立再試行回数	符号なし整数 ((1 ~ 65535)) 《0》	
			aretryint	コネクションの確立再試行間隔	符号なし整数 ((1 ~ 2550)) 《0》 (単位: 秒)	
		-l	-	論理端末の閉塞解除方法	auto 《manual》	
		-r	errinf	CEEREVT に切断理由コードを設定するか	yes 《no》	
	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	-	この論理端末のタイプ	any
			-m	mmsgcnt	メモリキュー内の出力メッセージの最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
				dmsgcnt	ディスクキュー内の出力メッセージの最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
			-k	quekind	出力キューの媒体の種類	《memory》 disk
				quegrpid	キューグループ ID	1 ~ 8 文字の識別子
-o			aj	送信完了時の情報を取得するかどうかの指定	《yes》 no	
mcftalced (コネクション定義の終了) 指定数: mcftalccn と同数		-	-	コネクション定義の終了	-	

(凡例)

- : 該当する内容がないことを表します。

注

TP1/NET/HDLC に固有の定義だけ記載してあります。このほかに、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。

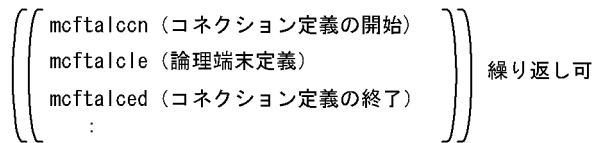
5. システム定義

TP1/NET/HDLC 固有のシステム定義の種類

定義の指定順序

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



TP1/NET/HDLC のプロトコル固有定義の指定には次の条件があります。

- mcftalccn コマンドと mcftalcle コマンドとの関係は 1 対 1 です。
- mcftalced コマンドは省略できません。
- mcftalcle コマンドと mcftalced コマンドの関係は 1 対 1 です。

mcftalccn (コネクション定義の開始)

形式

```
mcftalccn  -c コネクションID
            -p hdlcs
            -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]
               [count=メッセージ編集用バッファ数] " ]
            [-i auto | manual ]
            -m "mode=xnfas"
            -z "vslot=仮想スロット番号"
            [-x auto | manual ]
            [-a " [aretrycnt=コネクション障害時再試行回数]
               [aretryint=コネクション障害時再試行間隔] " ]
            [-l auto | manual ]
            [-r "errinf=yes | no" ]
```

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

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

-p hdlcs

プロトコルの種別を指定します。

hdlcs

HDLC 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。

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。

5. システム定義

mcftalccn (コネクション定義の開始)

-e

(オペランド)

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

入力および出力メッセージ編集 UOC 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。

count= メッセージ編集用バッファ数 ~ 符号なし整数 ((1 ~ 131070))

入力および出力編集メッセージ UOC の場合、メッセージ編集用として使用するバッファの数を指定します。

msgbuf オペランドで指定するメッセージ編集用バッファグループ番号に対応する mcftbuf コマンドの -g オプションの count、および extend オペランドで指定するバッファ数の中から、メッセージ編集用に使用するバッファ数を指定してください。また、このオペランドの指定は mcftbuf コマンドの -g オプションの count、および extend オペランドで指定されたバッファ数の合計値を超える指定はできません。-e オプションの msgbuf オペランドを省略した場合は、このオペランドの指定は無効です。

-i auto | manual ~ 《manual》

OpenTP1 システム開始時および再開時にコネクションを自動的に確立するかどうかを指定します。

auto

OpenTP1 システム開始時および再開時にコネクションを自動的に確立します。

manual

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

-m

(オペランド)

mode=xnfas

使用する通信管理を指定します。

xnfas

XNF/AS を使用します。

このオペランドは必ず指定してください。

-z

(オペランド)

vslot= 仮想スロット番号 ~ 符号なし整数 ((1 ~ 1000))

仮想スロット番号を指定します。

通信管理定義の link 文の VASS オペランドに指定した仮想スロット番号を指定してください。

-x auto | manual ~ 《manual》

コネクション障害時に、切断されたコネクションを自動的に再確立するかどうかを指定します。

auto

障害回復後、コネクションを自動的に再確立します。

manual

MCF 起動後、運用コマンド mcfstactcn コマンド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 ~ 符号なし整数 ((0 ~ 65535)) 《0》

(単位 : 回)

コネクション確立後に障害が発生した場合、MCF が行う確立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、MCF は通信管理から再試行不要の通知を受けるまで、確立再試行を繰り返します。運用コマンド mcfstalccn コマンドの -x オプションで manual を指定した場合は、このオペランドの指定は無効です。

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

位 : 秒)

コネクション確立後に障害が発生した場合、MCF が行う確立再試行の間隔を指定します。

このオペランドを省略した場合、または 0 を指定した場合は、MCF はすぐに確立再試行を行います。運用コマンド mcfstalccn コマンドの -x オプションで manual を指定した場合は、このオペランドの指定は無効です。

-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

コネクションの確立と同時に、自動的に論理端末の閉塞を解除します。

manual

運用コマンド mcfstactle コマンドを入力して論理端末の閉塞を解除します。運用コマ

5. システム定義

mcftalccn (コネクション定義の開始)

ンドを入力しない場合、論理端末は閉塞状態のままになります。

-r

(オペランド)

errinf=yes | no ~ 《no》

XNF/AS/HDLC から通知されたコネクションの切断理由コードを CERREVT に設定するかどうかを指定します。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。

yes

CERREVT に切断理由コードを設定します。

no

CERREVT に切断理由コードを設定しません。

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に対応するバッファグループ定義の 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	length オペランド	count オペランド
sndbuf	送信最大データ長 ¹	最大セグメント分割数以上
revbuf	受信最大データ長 ¹	(最大セグメント分割数 + 1) 以上
msgbuf ²	入力および出力メッセージ編集 UOC で編集したあとの最大データ長	最大セグメント分割数以上

注 1

length オペランドの最大値です。XNF/AS 接続時は 8160 バイトです。

注 2

入力および出力メッセージ編集 UOC を使用しない場合は不要です。

mcftalcle (論理端末定義)

形式

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

機能

論理端末に関する環境を定義します。

オプション

-l 論理端末名称 ~ 1 ~ 8 文字の識別子

論理端末名称を指定します。

この論理端末名称は、ほかの mcftalcle コマンドの -l オプションで指定する論理端末名称とは異なる指定をしてください。

-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となります。ただし、実際に待ち合わせをできる出力メッセージ数は動的共用メモリの容量に依存します。

dmsgcnt= ディスク出力メッセージ最大格納数 ~ 符号なし整数 ((0 ~ 65535)) 《0》
ディスクキューで待ち合わせをする出力メッセージの最大格納数を指定します。
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。

5. システム定義
mcftalcle (論理端末定義)

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。ただし、実際に待ち合わせをできる出力メッセージ数はメッセージキューファイルの容量に依存します。

-k

(オペランド)

quekind=memory | disk ~ 《memory》

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

memory

メモリキューだけに割り当てます。

disk

ディスクキューおよびメモリキューに割り当てます。

disk を指定した場合、必ず 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ます。

-o

(オペランド)

aj=yes | no ~ 《yes》

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

yes

メッセージ送信完了ジャーナルを取得します。

no

メッセージ送信完了ジャーナルを取得しません。

mcftalced (コネクション定義の終了)

形式

mcftalced

機能

一つのコネクション定義の終了を示します。

オプション

ありません。

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/HDLC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/HDLC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 通信プロセスのためのものとアプリケーション起動プロセスのためのものがあります。

MCF 実行形式プログラムは、MCF 通信プロセス同士、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/HDLC 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/HDLC 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。

set 形式のオペランド

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

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

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

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

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

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

5. システム定義

システムサービス共通情報定義

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 ~ 2016))

各 MCF 通信プロセスでアクセスするファイル数の中の最大値を指定します。

ファイル記述子の最大数を求める計算式を次に示します。

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

注 1

TP1/NET/HDLC の場合、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 内の各 MCF 通信プロセスごとに計算し、その結果の中で最大値が 500 より大きい場合は、その値を指定します。500 以下の場合は、500 を指定します。

指定値を超えてファイルのアクセスが発生した場合、その超過分はソケット用ファイル記述子使用数として扱われます。この場合、max_socket_descriptors オペランドの指定値から max_open_fds オペランドの指定値を減算した超過分が、実際のソケット用ファイル記述子の最大数になりますので、注意してください。

なお、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、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。

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

```
mcfhdlcs -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/HDLC のプロトコル固有定義ファイルの構文をチェックし、定義オブジェクトファイルを作成します。

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

TP1/NET/HDLC 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。

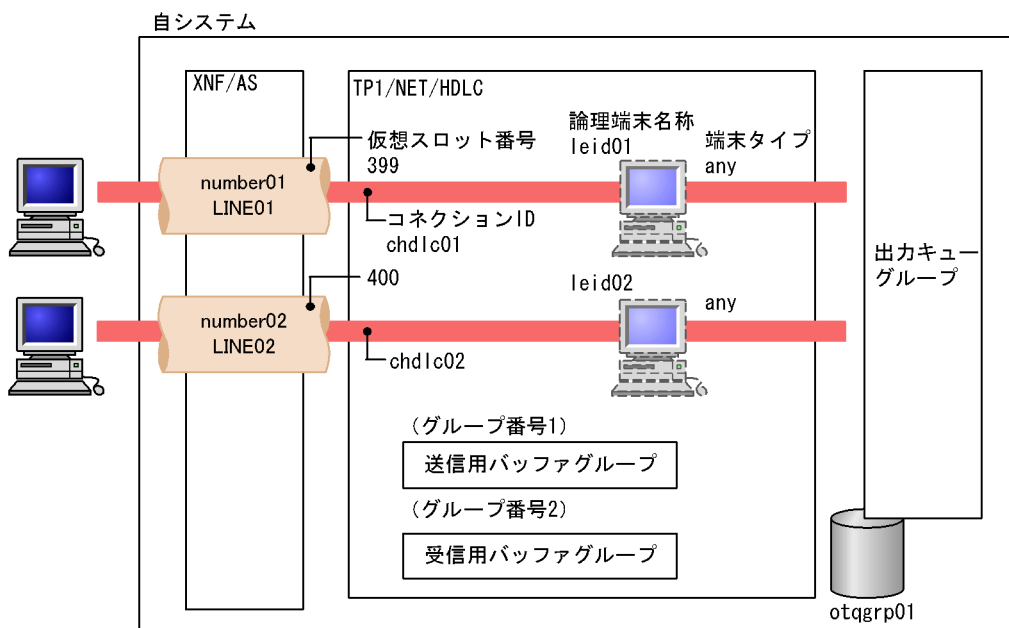
自システムの通信管理プログラムと関連づける内容

TP1/NET/HDLC を使用して AP 間通信をする場合、TP1/NET/HDLC のシステム定義と、自システム内の通信管理 (XNF/AS) の定義とを一致させる必要があります。

MCF 通信構成定義の `mcftalcen -z` で指定するスロット番号 (0 ~ 1000) と、XNF/AS のリンク定義文 (link 文の VASS オペランド) の指定値とを合わせてください。

XNF/AS で接続する場合の TP1/NET/HDLC のシステム構成例を次の図に示します。

図 5-2 TP1/NET/HDLC のシステム構成例 (XNF/AS で接続する場合)



定義例

XNF/AS の通信構成定義例，および TP1/NET/HDLC のシステム定義例（MCF マネージャ定義の例，MCF 通信構成定義共通の例，および MCF 通信構成定義固有の例）を示します。また，TP1/NET/HDLC のシステム定義例は次のファイルで提供しています。

- MCF マネージャ定義の例：/BeTRAN/examples/mcf/HDLC/conf/mngr
- MCF 通信構成定義共通の例：/BeTRAN/examples/mcf/HDLC/conf/com_c1
- MCF 通信構成定義固有の例：/BeTRAN/examples/mcf/HDLC/conf/com_c2

XNF/AS の定義の詳細については，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。

XNF/AS 通信構成定義例

```
configuration
  version 1
  max_line_adapter 10
  max_line 10
  max_link 10
  max_HDLCpass_link 10
  ;

HDLC_buffer
  name HDLCbuf
  number 1000
  size 8160
  ;

Line_adapter
  location_code 10-68 1
  name LADP01
  adapter_type HDLC
  initial_status active
  auto_start yes
  ;

group
  name ABM01
  type HDLC
  buf_pool HDLCbuf
  ;

line
  name LINE01
  number 01
  line_type leased_line
  line_mode ABM_DTE
  speed 9.6K
  ;

link
  name LINK01
```

5. システム定義 定義例

```
VASS 399 2
max_DPDU 8160
data_link_address 1
data_link_address2 2
auto_start yes
;

line
name LINE02
number 02
line_type leased_line
line_mode ABM_DCE
speed 9.6K
;

link
name LINK02
VASS 400 2
max_DPDU 8160
data_link_address 3
data_link_address2 4
auto_start yes
;
```

注 1
回線アダプタの実装位置によって異なります。

注 2
それぞれスロット 399, スロット 400 を使用します。

MCF マネジャ定義の例

mcfmenv	-m	"name=mcfmngvr"	
mcfmcomm	-n	20	¥
	-p	300	¥
	-j	4096	¥
mcfmcname	-s	"mcfsvname=_muHDLCL syssvname=mcfu01"	¥
mcfmuap	-d	65535	¥
	-t	"sndtim=3600 sndrcvtim=3600"	¥
	-j	32768	¥
	-l	"initseq=1 maxseq=9999 minseq=1"	¥
	-e	"segsz=8160"	¥
mcfmqgid	-q	"quekind=otq quegrp=otqgrp01"	¥
mcfmqgid	-q	"quekind=itq quegrp=itqgrp01"	¥
mcfmexp	-g	100	¥
	-l	30	¥

```
-i dg ¥
-o dg ¥
```

MCF 通信構成定義共通の例

```
mcftenv    -s 01 ¥
           -a apliHDLc
mcfcomn    -j 32768
mcfttred   -m 10
mcfttim    -t "btim=5 ¥
             mtim=180 ¥
             rmtim=180"
mcfttrc    -t "size=20480 ¥
             disk=yes ¥
             bufcnt=50 ¥
             trccnt=3" ¥
           -m del
mcftsts    -a 10 ¥
           -l 10
mcftbuf    -g "groupno=1 ¥
             length=8160 ¥
             count=256 ¥
             extend=256"
mcftbuf    -g "groupno=2 ¥
             length=8160 ¥
             count=256 ¥
             extend=256"
mcftbuf    -g "groupno=3 ¥
             length=8160 ¥
             count=256 ¥
             extend=256"
```

MCF 通信構成定義固有の例

```
mcftalccn  -c chdlc01 ¥
           -p hdlcs ¥
           -m "mode=xnfas" ¥
           -z "vslot=399 " ¥
           -g "sndbuf=1 ¥
             rcvbuf=2" ¥
mcftalcle  -l leid01 ¥
           -t any ¥
           -m "mmsgcnt=20 ¥
             dmsgcnt=20"
mcftalced
```

5. システム定義 定義例

```
mcftalccn  -c  chdlc02
            -p  hdlcs
            -m  "mode=xfas"
            -z  "vslot=400 "
            -g  "sndbuf=1
                rcvbuf=2"
mcftalcle  -l  leid02
            -t  any
            -m  "mmsgcnt=20
                dmsgcnt=20"

mcftalced
```

注

ネットワーク定義の仮想スロット番号に対応します。

6

運用コマンド

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

TP1/NET/HDLC の運用コマンド

mcftactcn (コネクションの確立)

mcftactle (論理端末の閉塞解除)

mcftdctcn (コネクションの解放)

mcftdctle (論理端末の閉塞)

mcftlscn (コネクションの状態表示)

mcftlsle (論理端末の状態表示)

TP1/NET/HDLC の運用コマンド

ここでは、TP1/NET/HDLC 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/HDLC で使用する運用コマンドの一覧を、次の表に示します。

表 6-1 TP1/NET/HDLC の運用コマンド

機能		コマンド 名称	定義中組 み込み	オフ ライ ン 中 に 実 行	オン ライ ン 中 に 実 行	UAP から 実行
コネクション管理	コネクションの確立	mftactcn	×	×		
	コネクションの解放	mftdctcn	×	×		
	コネクションの状態表示	mftlscn	×	×		
論理端末管理	論理端末の閉塞解除	mftactle	×	×		
	論理端末の閉塞	mftdctle	×	×		
	論理端末の状態表示	mftlsle	×	×		

(凡例)

- : 組み込み、または実行ができます。
- ×: 組み込み、または実行ができません。

mcfactcn (コネクションの確立)

形式

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

機能

コネクションを確立します。

オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID ~ 1 ~ 8 文字の識別子

確立するコネクションのコネクション ID を指定します。

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

同一コネクション ID は、重複して指定できません。また、コネクション ID は、* を使って一括指定ができます。一括指定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。一括指定するときは、引用符 (") で囲んで指定します。

* : すべてのコネクションを確立します。

先行文字列 * : 先行文字列で始まるすべてのコネクションを確立します。

複数指定の例 cnn1 , cnn2 , cnn3 を指定する場合

```
-c "cnn1    cnn2    cnn3"
```

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

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

出力メッセージ

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

6. 運用コマンド

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
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	コネクション確立処理済みのため、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

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ます。

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

同一コネクション ID は、重複して指定できません。また、コネクション ID は、* を使って一括指定ができます。一括指定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。一括指定するときは、引用符 (") で囲んで指定します。

* : すべての論理端末の閉塞を解除します。

先行文字列 * : 先行文字列で始まるすべての論理端末の閉塞を解除します。

複数指定の例 len1 , len2 , len3 を指定する場合

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

一括指定の例 len で始まるすべての論理端末を指定する場合

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

6. 運用コマンド

mcftactle (論理端末の閉塞解除)

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
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
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定されたコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
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

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。複数指定するときは、引用符 (") で囲んで、コネクション ID とコネクション ID との間を空白で区切ります。

同一コネクション ID は、重複して指定できません。また、コネクション ID は、* を使って一括指定ができます。一括指定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。一括指定するときは、引用符 (") で囲んで指定します。

* : すべてのコネクションを解放します。

先行文字列 * : 先行文字列で始まるすべてのコネクションを解放します。

複数指定の例 cnn1, cnn2, cnn3 を指定する場合

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

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

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

```
-f
```

該当するコネクションを強制的に解放します。

このオプションを指定すると、該当するコネクションが仕掛り中の場合、仕掛り中の処理を終了しないで無条件に解放します。

このオプションの指定を省略すると、該当するコネクションが仕掛り中の場合、このコ

6. 運用コマンド

mcftdcten (コネクションの解放)

マンドは異常終了します。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftdcten 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
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	mcftdcten 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	mcftdcten コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftdcten 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

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

論理端末を閉塞します。

コマンド入力時に送信中であったメッセージの扱いについては、「8.1 障害の種類と対応処理」を参照してください。なお、論理端末が閉塞されても、UAP からの送信要求はできます。

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ます。

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

同一コネクション ID は、重複して指定できません。また、コネクション ID は、* を使って一括指定ができます。一括指定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。一括指定するときは、引用符 (") で囲んで指定します。

* : すべての論理端末を閉塞します。

先行文字列 * : 先行文字列で始まるすべての論理端末を閉塞します。

複数指定の例 len1, len2, len3 を指定する場合

6. 運用コマンド

mcftdctl (論理端末の閉塞)

```
-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
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
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定されたコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
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

注意事項

受信仕掛り中に運用コマンド (mcftdctl) を入力した場合、受信仕掛り中のメッセージを破棄します。以降の受信メッセージは入力キューに登録しません。

送信仕掛り中に運用コマンド (mcftdctl) を入力した場合は、送信処理が中断されず。UAP からの送信メッセージは、出力キュー上に格納されます。

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出力します。

オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID ~ 1 ~ 8 文字の識別子

状態を表示するコネクションのコネクション ID を指定します。

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

同一コネクション ID は、重複して指定できません。また、コネクション ID は、* を使って一括指定ができます。一括指定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。一括指定するときは、引用符 (") で囲んで指定します。

* : すべてのコネクションの状態を表示します。

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

複数指定の例 cnn1 , cnn2 , cnn3 を指定する場合

```
-c "cnn1    cnn2    cnn3"
```

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

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

-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

6. 運用コマンド

mcftlscn (コネクションの状態表示)

```
mmm cccccc ppp sssss dddd bbbbbbbb  
lllllllll ttt
```

注

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

- mmm : MCF 識別子
- cccccc : コネクション ID
- ppp : プロトコル種別
hds...HDLC プロトコル
- sssss : コネクションの状態
ACT...確立
ACT/B...確立処理中
DCT...解放
DCT/B...解放処理中
- dddd : 詳細ステータス (保守情報)
- bbbbbbbb : コネクション確立時からのビジー発生回数統計情報
99999999 回を超えた場合, ***** を表示します。
- lllllll : 論理端末名称
- ttt : 論理端末タイプ

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftlscn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftlscn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10360-I	状態表示を開始します。	標準出力
KFCA10361-I	標準情報を表示します。	標準出力
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	mcftlscn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftlscn コマンドが異常終了しました。	標準エラー出力

出力メッセージ ID	内容	出力先
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

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出力します。

オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 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ます。

論理端末名称は、一度に 8 個まで指定できます。複数指定するときは、引用符 (") で囲んで、論理端末名称と論理端末名称との間を空白で区切ります。同一コネクション ID は、重複して指定できません。また、コネクション ID は、* を使って一括指定ができません。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。一括指定するときは、引用符 (") で囲んで指定します。

*: すべての論理端末の状態を表示します。

先行文字列 *: 先行文字列で始まるすべての論理端末の状態を表示します。

複数指定の例 len1, len2, len3 を指定する場合

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

一括指定の例 len で始まるすべての論理端末を指定する場合

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

-q

指定した論理端末に対応する出力キューの保留状態を表示します。

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

-t

指定した論理端末がセキュア状態かどうかを表示します。

出力形式

```
mmm llllllllll sss {ggg} {tttt}
  SYNC xxxxxxxxxxxx yyyyyyyyyy zzzzzzzzzz
    IO      :           :           :
  PRIO     :           :           :
  NORM     :           :           :
  iii ooo
```

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

6. 運用コマンド

mcftlsle (論理端末の状態表示)

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	標準出力
KFCA10365-I	表示情報	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10371-I	mcftlsle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftlsle コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定したコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10395-E	指定したコネクションには未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10505-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

7

組み込み方法

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

7.1 TP1/NET/HDLC の組み込みの流れ

7.2 MCF メイン関数の作成

7.3 定義オブジェクトファイルの生成

7.1 TP1/NET/HDLC の組み込みの流れ

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

(2) MCF サービス名の登録

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

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

詳細は、マニュアル「OpenTP1 システム定義」を参照してください。

(3) システムサービス情報定義ファイルの作成

システムサービス情報定義ファイルをテキストエディタで作成します。作成するファイルのパス名は、「\$DCDIR/lib/sysconf/ システムサービス情報定義ファイル名」です。ファイルの定義形式については、5章の「システムサービス情報定義」を参照してください。

(4) 定義オブジェクトファイルの生成

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

7.2 MCF メイン関数の作成

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

TP1/NET/HDLC を起動するためには、ユーザが MCF メイン関数を作成し、コンパイル、およびリンケージを行って TP1/NET/HDLC 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、mcfplhds 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 7-1 と図 7-2 に示します。また、ディレクトリへの組み込み方法を図 7-3 に示します。

なお、これらのコーディング例は、次のファイルで提供しています。

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

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

```

#include <dcmhds.h>                /*TP1/NET/HDLC用ヘッダファイル */ 1.
extern DCLONG msgrcv01(dcmcf_uoc_min_n *); /*入力メッセージ編集UOC関数extern宣言 */ 2.
extern DCLONG msgsend01(dcmcf_uoc_mout_n *); /*出力メッセージ編集UOC関数extern宣言 */ 2.

extern dcmcf_uoc_t dcmcf_uoctbl;    /*UOCテーブルextern宣言 */ 3.

int main()
{
    dcmcf_uoctbl.msgrcv = (dcmcf_uocfunc)msgrcv01;
                                /*入力メッセージ編集UOCアドレス設定 */ 4.
    dcmcf_uoctbl.msgsend = (dcmcf_uocfunc)msgsend01;
                                /*出力メッセージ編集UOCアドレス設定 */ 4.

    dc_mcf_svstart();            /*スタート関数の呼び出し */ 5.
    return 0;
}

```

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

```

#include <dcmhdsm.h>                /*TP1/NET/HDLC用ヘッダファイル */ 1.
extern DCLONG msgrcv01();          /*入力メッセージ編集UOC関数extern宣言 */ 2.
extern DCLONG msgsend01();        /*出力メッセージ編集UOC関数extern宣言 */

extern dcmcf_uoc_t dcmcf_uoctbl; /*UOCテーブルextern宣言 */ 3.

main()
{
    dcmcf_uoctbl.msgrcv = msgrcv01; /*入力メッセージ編集UOCアドレス設定 */ 4.
    dcmcf_uoctbl.msgsend = msgsend01; /*出力メッセージ編集UOCアドレス設定 */

    dc_mcf_svstart();              /*スタート関数の呼び出し */ 5.
}

```

1. TP1/NET/HDLC で提供するヘッダファイルを取り込みます。
2. 使用する UOC 関数を extern 宣言します。UOC のリターン値は DCLONG 型にしてください。
UOC をまったく使用しない場合、このコーディングは必要ありません。
3. UOC テーブルを extern 宣言します。UOC を使用する場合、必ずこのとおりにコーディングしてください。
UOC をまったく使用しない場合、このコーディングは必要ありません。
4. 各 UOC 関数のアドレスを、次に示すシステム提供変数に設定します。使用する UOC だけコーディングしてください。

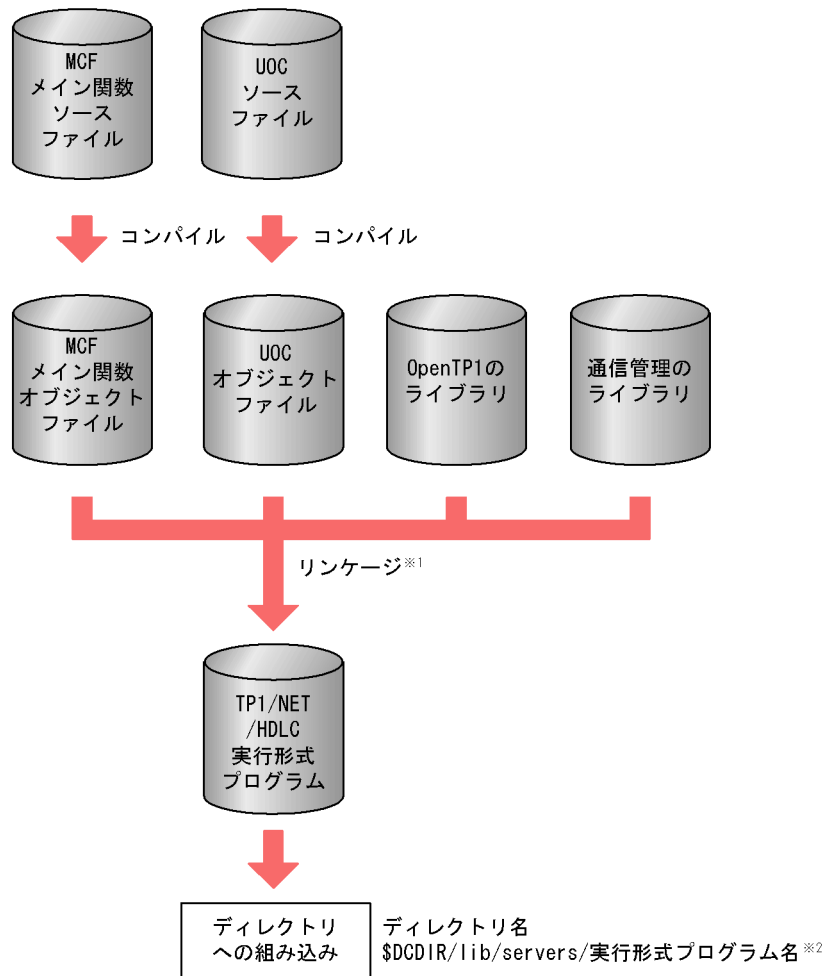
```

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

```

- UOC をまったく使用しない場合、このコーディングは必要ありません。
5. スタート関数を呼び出します。MCF メイン関数には必ずコーディングしてください。

図 7-3 MCF メイン関数のディレクトリへの組み込み方法



注 1
mcfplhds コマンドでリンケージします。

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

7.3 定義オブジェクトファイルの生成

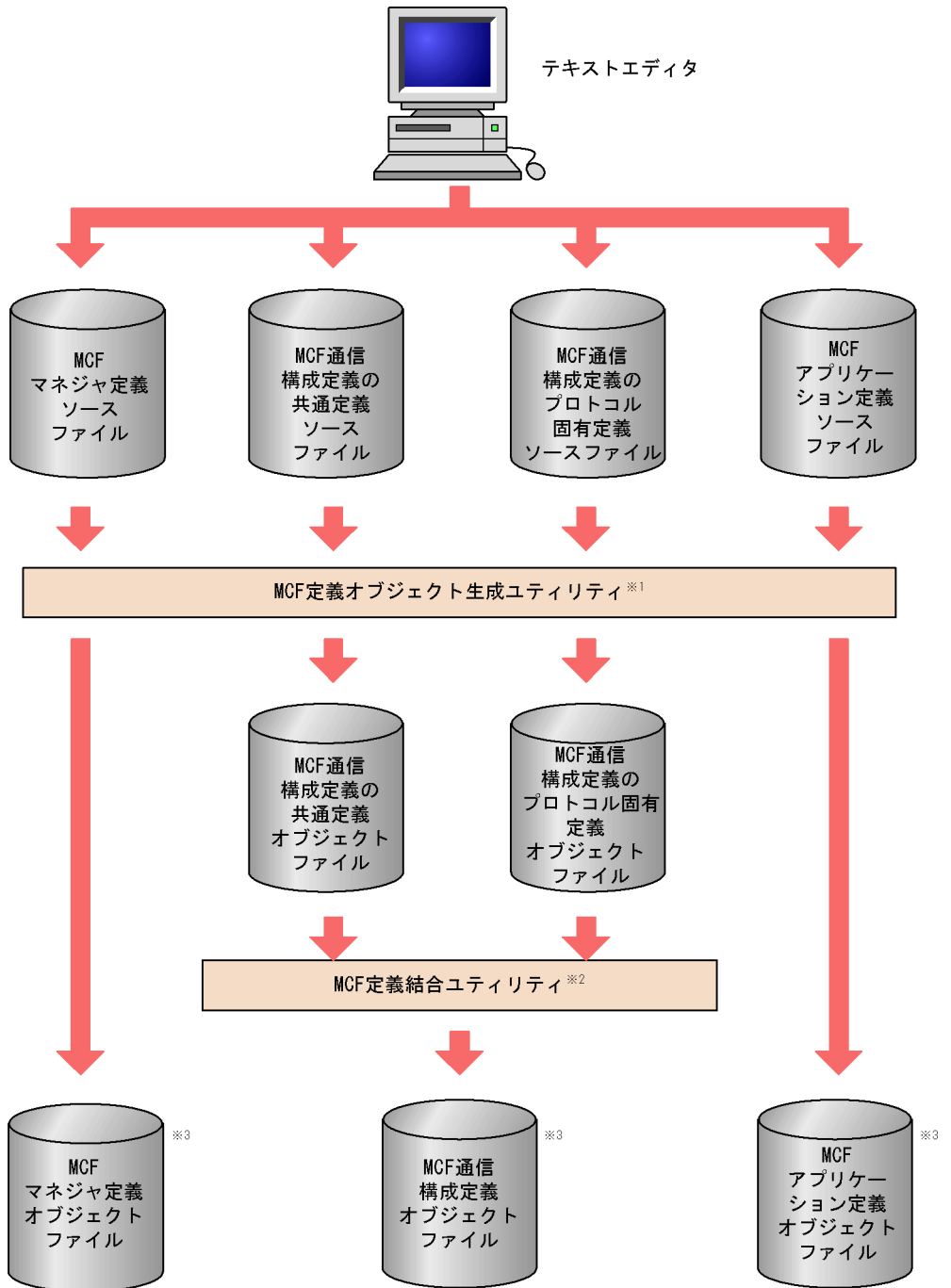
定義オブジェクトファイルを次の手順で生成します。

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場合がありますのでご注意ください。

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

定義オブジェクトファイルの作成方法の概要を次の図に示します。

図 7-4 定義オブジェクトファイルの作成方法の概要



注 1
次に示すコマンドで生成します。

7. 組み込み方法

```
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。

- mcfmngrr : MCF マネージャ定義ソースファイル
- mcfcomn : MCF 通信構成定義の共通定義ソースファイル
- mcfhdlcs : MCF 通信構成定義のプロトコル (TP1/NET/HDLC) 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 mcfhdlcs コマンドについては 5 章の「MCF 定義オブジェクトの生成」を、その他のコマンドについてはマニュアル「OpenTP1 システム定義」を参照してください。

注 2

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

```
mcflink -i 共通定義オブジェクトファイル名  
        TP1/NET/HDLC定義オブジェクトファイル名  
        -o 出力オブジェクトファイル名
```

注 3

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

8

障害対策

この章では、TP1/NET/HDLC 運用中に発生する障害と、TP1/NET/HDLC の対応処理、およびメッセージの処理について説明します。

8.1 障害の種類と対応処理

8.2 コネクション障害

8.3 UAP 障害

8.4 UAP 閉塞

8.5 入力キュー障害

8.6 出力キュー障害

8.1 障害の種類と対応処理

TP1/NET/HDLC の障害発生時の処理について、次に示す障害の種類ごとに説明します。

- コネクション障害
- 受信スケジュール関係（入力キュー，入力メッセージ編集 UOC）の障害
- 送信スケジュール関係（出力キュー，出力メッセージ編集 UOC）の障害
- UAP 障害
- MCF の障害

（１）コネクション障害

コネクション障害時の TP1/NET/HDLC の障害処理については、「8.2 コネクション障害」を参照してください。

表 8-1 コネクション障害発生時の処理

障害の内容	TP1/NET/HDLC の処理	ユーザの処理
コネクション確立時障害（回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14103-W）を出力します。 2. コネクションの閉塞を通知するメッセージログ（KFCA14002-E）を出力します。 3. CERREVT（コネクションの確立不可能）を起動します。 	障害の要因を取り除いたあと、運用コマンド（mcftactcn）を入力して、再びコネクションを確立します。
コネクション確立時障害（回復可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14103-W）を出力します。 2. 回復動作を通知するメッセージログ（KFCA14100-I）を出力します。 3. 一定時間（リトライタイムの指定時間）経過後にコネクションの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）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14103-W）を出力します。 2. コネクションの解放を通知するメッセージログ（KFCA14001-I）を出力します。 3. コネクションの閉塞を通知するメッセージログ（KFCA14002-E）を出力します。 4. CERREVT（コネクションの切断，コネクション回復不可能）を起動します。 	障害の要因を取り除いたあと、運用コマンド（mcftactcn）を入力して、再びコネクションを確立します。

障害の内容	TP1/NET/HDLC の処理	ユーザの処理
コネクション切断障害（回復可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14103-W）を出力します。 2. コネクションの解放を通知するメッセージログ（KFCA14001-I）を出力します。 3. CERREVT（コネクションの切断，コネクション回復可能）を起動します。 4. コネクション再開始自動確立の場合，コネクションの回復起動を通知するメッセージログ（KFCA14000-I）を出力します。 5. コネクション再開始自動確立の場合，一定時間（リトライタイマの指定時間）経過後にコネクションを再確立します。 	リトライ回数を超過した場合，またはコネクション再開始手動確立の場合は，運用コマンド（mefctactcn）を入力して，再びコネクションを確立します。
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，および受信メッセージのオーバーフローを通知するメッセージログ（KFCA14104-W）を出力します。 2. コネクションの解放を通知するメッセージログ（KFCA14001-I）を出力します。 3. コネクションの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てください。
メッセージ送信時障害（回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14103-W）を出力します。 2. コネクションの解放を通知するメッセージログ（KFCA14001-I）を出力します。 3. コネクションの閉塞を通知するメッセージログ（KFCA14002-E）を出力します。 4. CERREVT（コネクションの切断）を起動します。 5. メッセージは出力キューに残留し，リラン時に再送信されます。 	障害の要因を取り除いたあと，運用コマンド（mefctactcn）を入力して，再びコネクションを確立します。
通信管理送信バッファオーバーフロー	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14103-W）を出力します。 2. コネクションの解放を通知するメッセージログ（KFCA14001-I）を出力します。 3. コネクションの閉塞を通知するメッセージログ（KFCA14002-E）を出力します。 4. CERREVT（コネクションの切断）を起動します。 5. メッセージは出力キューに残留し，リラン時に再送信されます。 	システム定義，またはUAPを修正してください。

（2）受信スケジュール関係（入力キュー，入力メッセージ編集UOC）の障害

入力キュー障害時の障害処理については、「8.5 入力キュー障害」を参照してください。

8. 障害対策

表 8-2 受信スケジュール関係の障害発生時の処理

障害の内容	TP1/NET/HDLC の処理	ユーザの処理
UAP 名未定義	ERREVT1 を起動します。	システム定義を修正してください。
MCF イベントの名称未定義 <ul style="list-style-type: none"> • ERREVTx • CxxxEVT 	メッセージ, および MCF イベントを破棄します。	ありません。
<ul style="list-style-type: none"> • MHP サービス, サービスグループ閉塞 • 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。 	ありません。
入力メッセージ編集 UOC エラー	<ol style="list-style-type: none"> 1. UOC エラーを通知するメッセージログ (KFCA10611-E) を出力します。 2. 論理端末を閉塞します。 3. 論理端末の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。
入力メッセージ編集 UOC パラメタ不正	<ol style="list-style-type: none"> 1. 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. 論理端末の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。
UAP 名形式不正	<ol style="list-style-type: none"> 1. UAP 名の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。 	ありません。

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

出力キュー障害時の障害処理については、「8.6 出力キュー障害」を参照してください。

表 8-3 送信スケジュール関係の障害発生時の処理

障害の内容	TP1/NET/HDLC の処理	ユーザの処理
出力キュー読み込み障害	<ol style="list-style-type: none"> 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 CERREVT (出力キュー読み込み障害) を起動します。 該当するメッセージを消去します。 論理端末を閉塞します。 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 	コネクションを解放してください。また、相手システムに出力キュー障害を通知 (send) してください。
出力メッセージ編集 UOC エラー	<ol style="list-style-type: none"> UOC エラーを通知するメッセージログ (KFCA10611-E) を出力します。 CERREVT (UOC 障害) を起動します。 該当するメッセージを消去します。 論理端末を閉塞します。 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 	コネクションを解放してください。また、相手システムに出力キュー障害を通知 (send) してください。
出力メッセージ編集 UOC パラメタ不正	<ol style="list-style-type: none"> UOC パラメタ不正を通知するメッセージログ (KFCA10620-E) を出力します。 CERREVT (UOC 障害) を起動します。 該当するメッセージを消去します。 論理端末を閉塞します。 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 	コネクションを解放してください。また、相手システムに出力キュー障害を通知 (send) してください。
メッセージ消し込み障害	<ol style="list-style-type: none"> メッセージ送信完了障害を通知するメッセージログ (KFCA10617-E) を出力します。 処理を続行します。 	ありません。
送信バッファオーバーフロー	<ol style="list-style-type: none"> 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 論理端末を閉塞します。 該当するメッセージを消去します。 CERREVT (受信バッファオーバーフロー) を起動します。 	コネクションを解放してください。また、システム定義、または UAP を修正してください。

8. 障害対策

障害の内容	TP1/NET/HDLC の処理	ユーザの処理
送信バッファ不足	<ol style="list-style-type: none"> 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。 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。 該当するメッセージを取り出し前の状態に戻します。 コネクションの障害を通知するメッセージログ (KFCA14004-I) を出力します。 論理端末を閉塞します。 CERREVT (送信バッファ不足) を起動します。 	一定時間経過後、運用コマンド (mcftactle) を入力して論理端末の閉塞を解除し、メッセージを再送信してください。また、送信バッファの面数を見直してください。
送信処理仕掛り中の mcftactle コマンド入力	<ol style="list-style-type: none"> 論理端末を閉塞します。 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 該当するメッセージを取り出し前の状態に戻します。 コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 CERREVT (コネクション切断、仕掛り中の mcftactle コマンド入力) を起動します。 	ありません。

(4) UAP 障害

表 8-4 UAP 障害発生時の処理

障害の内容	TP1/NET/HDLC の処理	ユーザの処理
セグメント受信前の UAP 異常終了	ERREVT2 を起動します。	相手システムに UAP の異常終了を通知 (send) してください。
セグメント受信後の UAP 異常終了	ERREVT3 を起動します。	相手システムに UAP の異常終了を通知 (send) してください。

(5) MCF の障害

表 8-5 MCF の障害発生時の処理

障害の内容	TP1/NET/HDLC の処理	ユーザの処理
<ul style="list-style-type: none"> 内部論理矛盾 他 PP とのインタフェースエラー 	<ol style="list-style-type: none"> 障害情報の取得を通知するメッセージログ (KFCA10189-E) を出力します。 コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 CERREVT (コネクションレベル) を起動します。 デバッグモードの場合、MCF を異常終了します。 	ありません。

8.2 コネクション障害

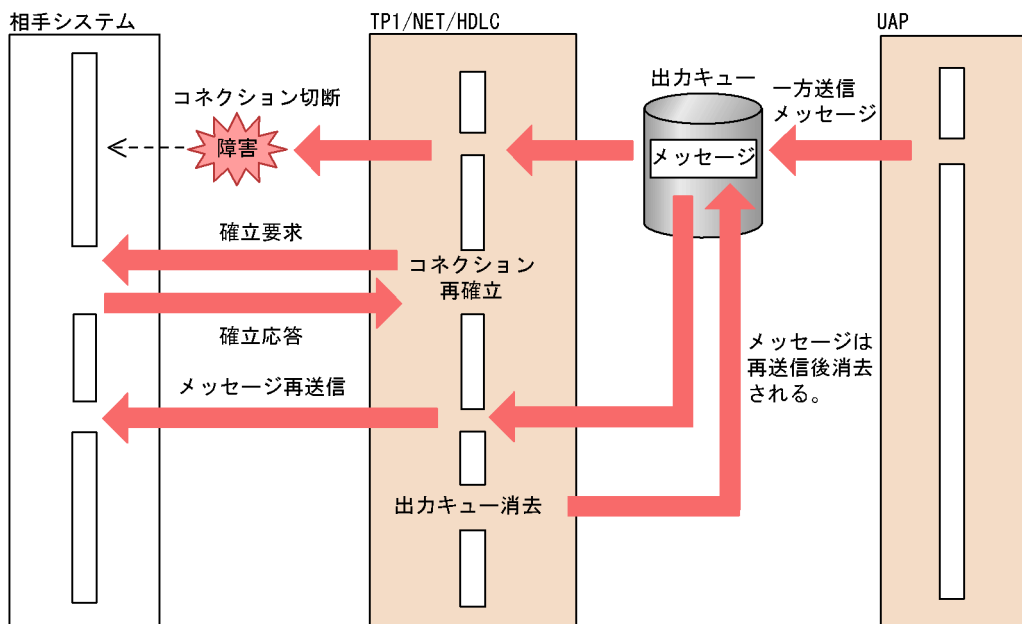
コネクションが切断された場合、TP1/NET/HDLC はコネクションを再確立します。通信管理が管理している通信機器に障害が発生した場合も、コネクション障害として処理します。TP1/NET/HDLC が自動回復できない障害の場合、障害の要因を取り除いたあと、運用コマンド (mcftacten) を入力してコネクションの確立を再試行してください。

コネクション障害が発生した場合の TP1/NET/HDLC の処理は、障害の発生したタイミングによって異なります。障害の発生するタイミングごとの TP1/NET/HDLC の処理を説明します。

8.2.1 一方送信メッセージ送信時のコネクション障害

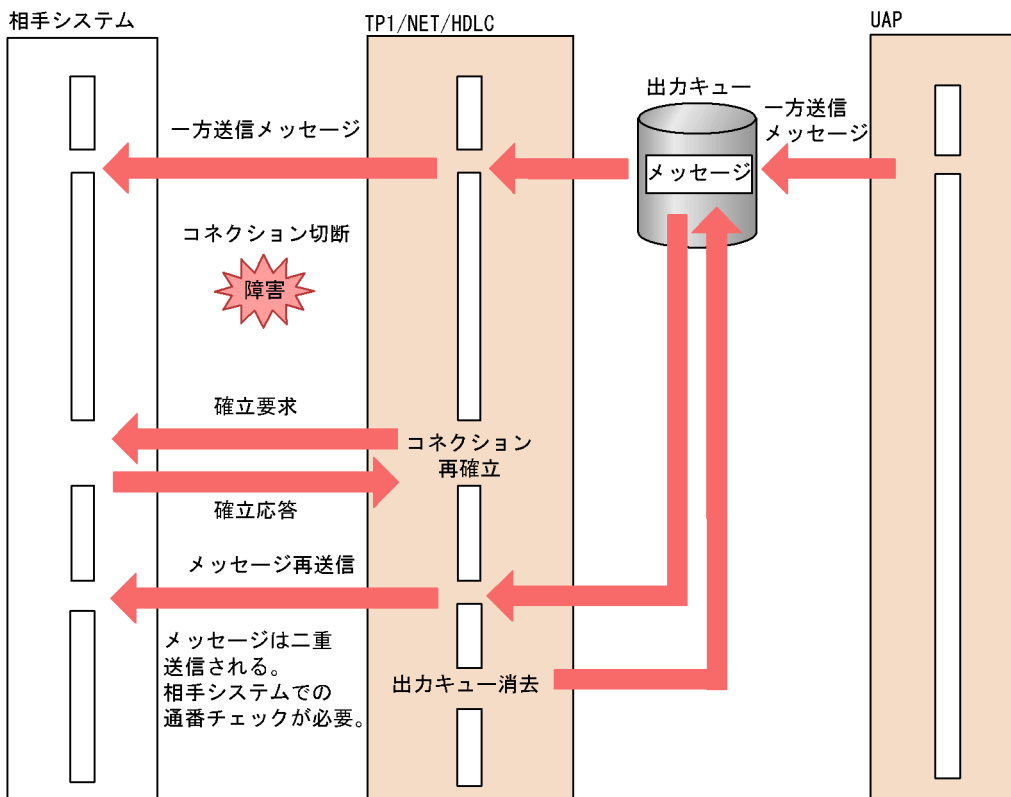
一方送信メッセージを送信する前にコネクション障害が発生した場合、メッセージはロールバックされ、コネクションの再確立時に再送信されます。一方送信メッセージを送信する前にコネクション障害が発生した場合の処理の流れを次の図に示します。

図 8-1 一方送信メッセージ送信前のコネクション障害



一方送信メッセージを出力キューから消去する前にコネクション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て二重送信を管理します。一方送信メッセージを消去する前にコネクション障害が発生した場合の処理の流れを次の図に示します。

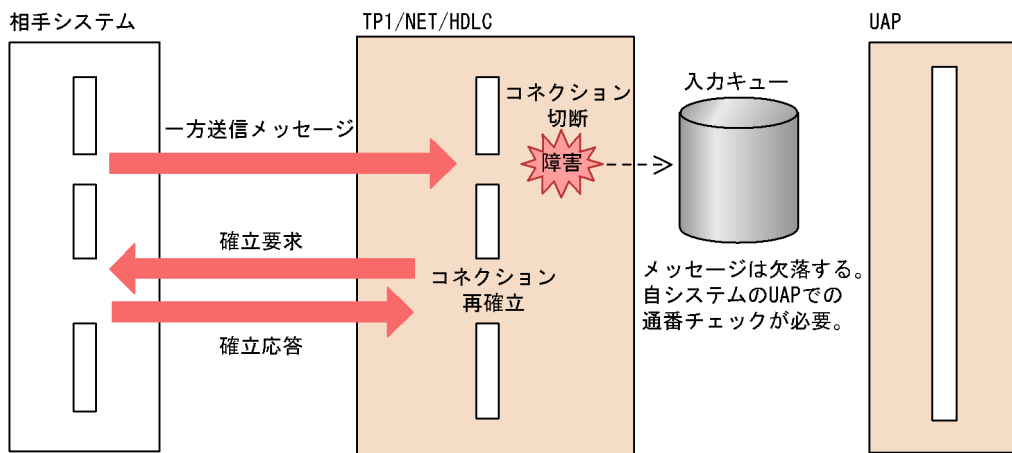
図 8-2 一方送信メッセージ消去前のコネクション障害



8.2.2 一方送信メッセージ受信時のコネクション障害

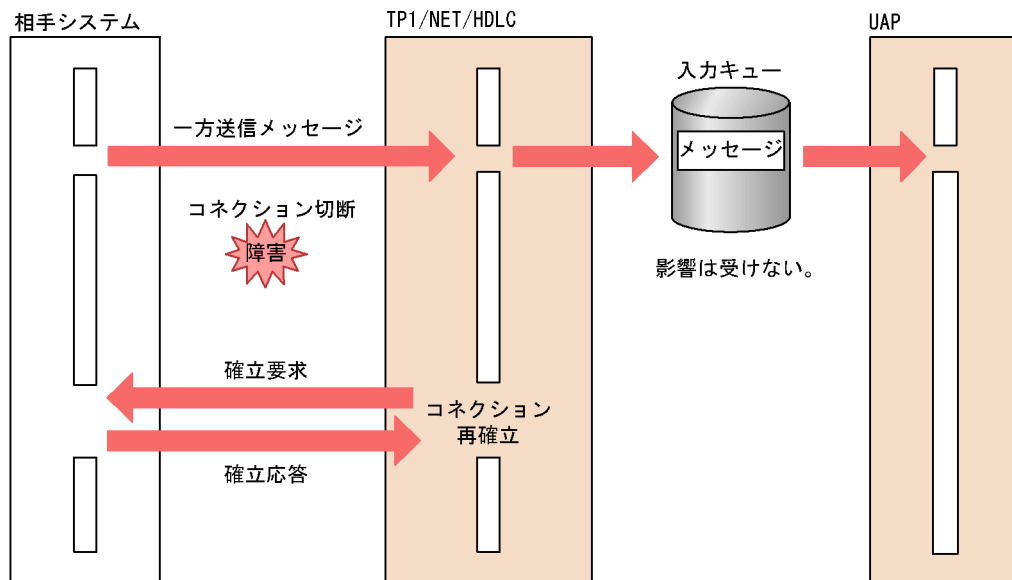
一方送信メッセージを入力キューに書き込む前にコネクション障害が発生した場合、メッセージは欠落します。このとき、自システムのUAPは、メッセージに通番を付けてメッセージが欠落しているかどうかを管理します。一方送信メッセージを書き込む前にコネクション障害が発生した場合の処理の流れを次の図に示します。

図 8-3 一方送信メッセージ書き込み前のコネクション障害



一方送信メッセージを入力キューに書き込んだあとにコネクション障害が発生した場合、メッセージに影響はありません。一方送信メッセージを書き込んだあとにコネクション障害が発生した場合の処理の流れを次の図に示します。

図 8-4 一方送信メッセージ書き込み後のコネクション障害



8.3 UAP 障害

UAP 障害が発生すると UAP が異常終了します。

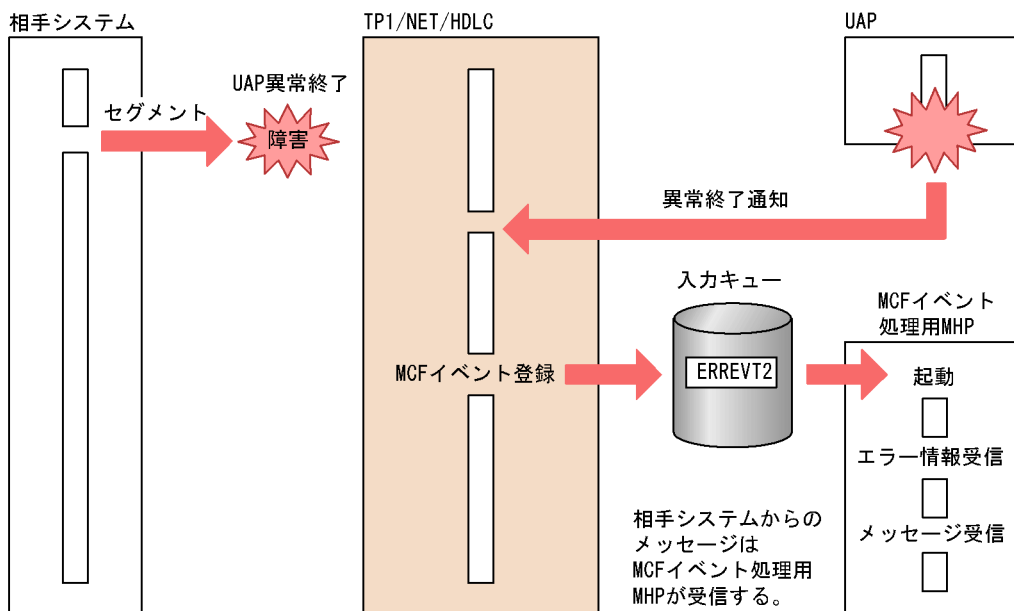
UAP が異常終了した場合、TP1/NET/HDLC は入力キューに MCF イベントを登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

UAP 障害が発生した場合の TP1/NET/HDLC の処理は、障害の発生したタイミングによって異なります。障害の発生するタイミングごとの TP1/NET/HDLC の処理を説明します。

8.3.1 セグメント受信前の UAP 障害

セグメント受信前に UAP 障害が発生した場合、TP1/NET/HDLC 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。セグメント受信前に UAP 障害が発生した場合の処理の流れを次の図に示します。

図 8-5 セグメント受信前の UAP 障害

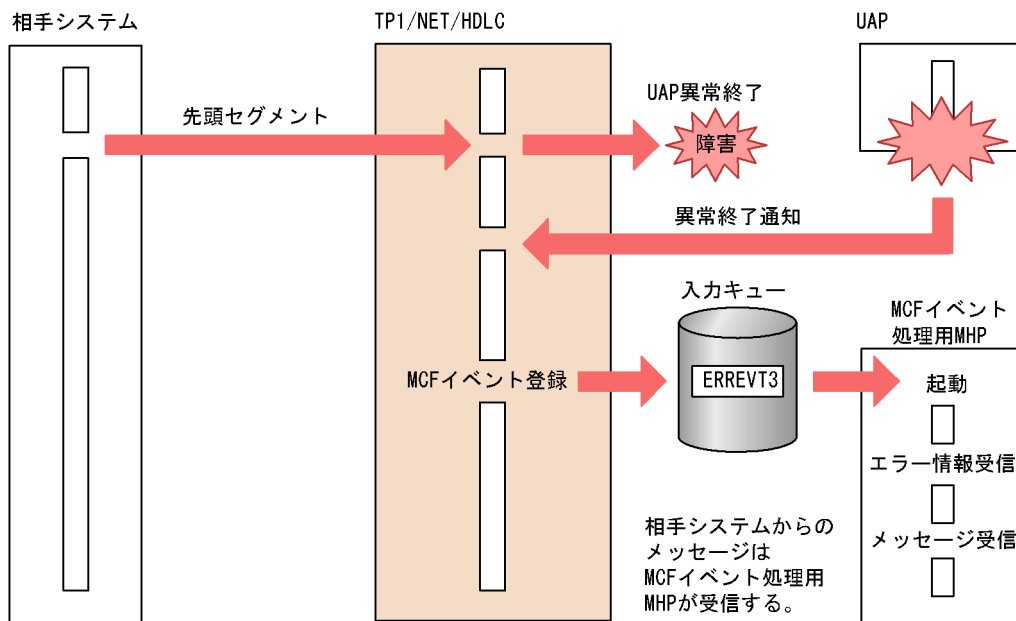


8.3.2 セグメント受信後の UAP 障害

セグメント受信後に UAP 障害が発生した場合、TP1/NET/HDLC 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。セグメント受信後に UAP 障害が発生した場合の処理の流れを次の図に示します。

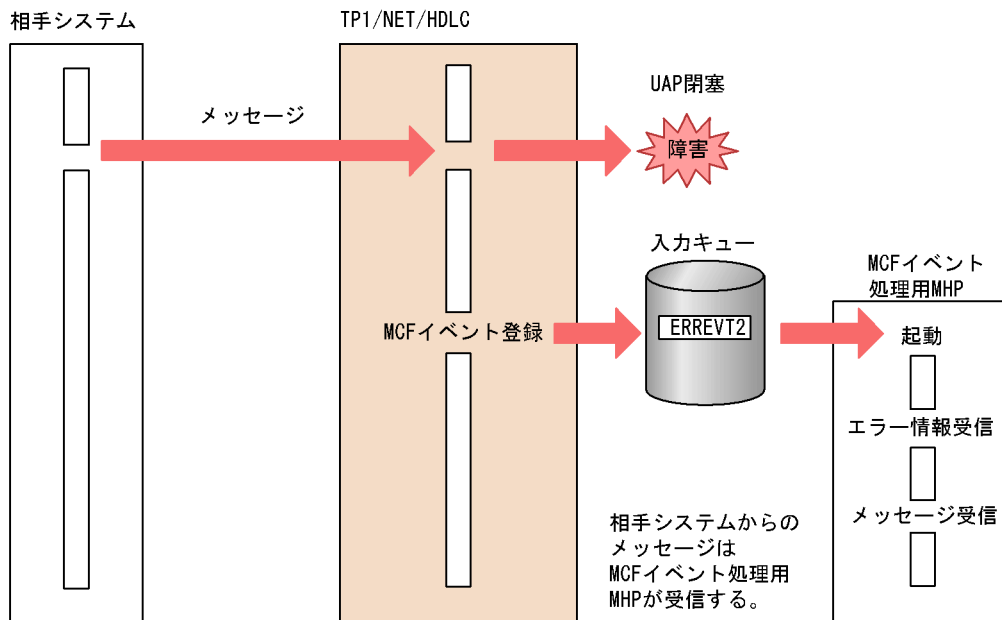
図 8-6 セグメント受信後の UAP 障害



8.4 UAP 閉塞

UAP が閉塞して使用できない場合、TP1/NET/HDLC 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。UAP が閉塞した場合の処理の流れを次の図に示します。

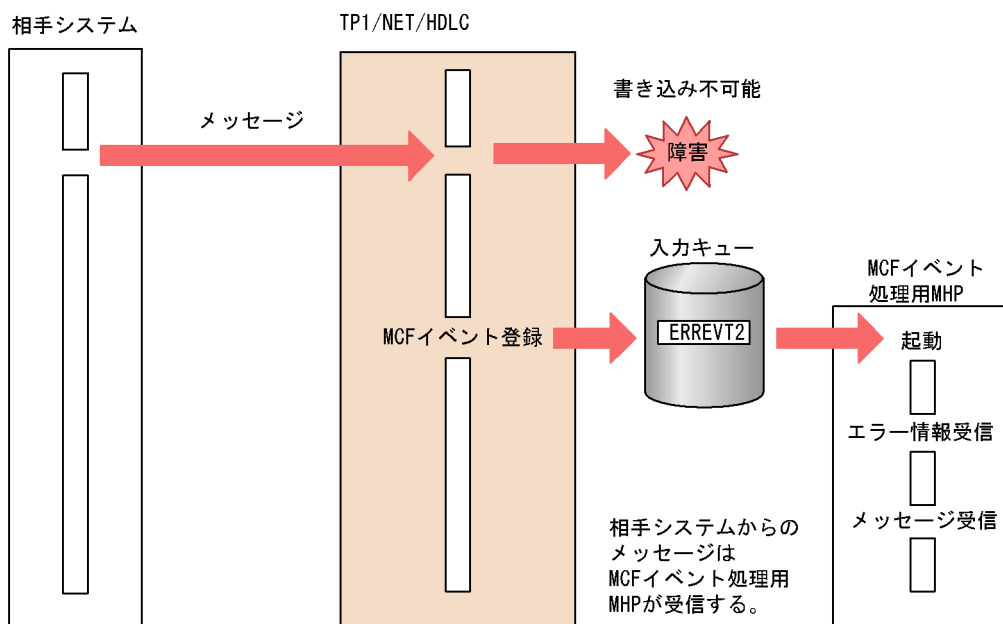
図 8-7 UAP 閉塞時の障害



8.5 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/HDLC 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。メッセージ書き込み時に障害が発生した場合の処理の流れを次の図に示します。

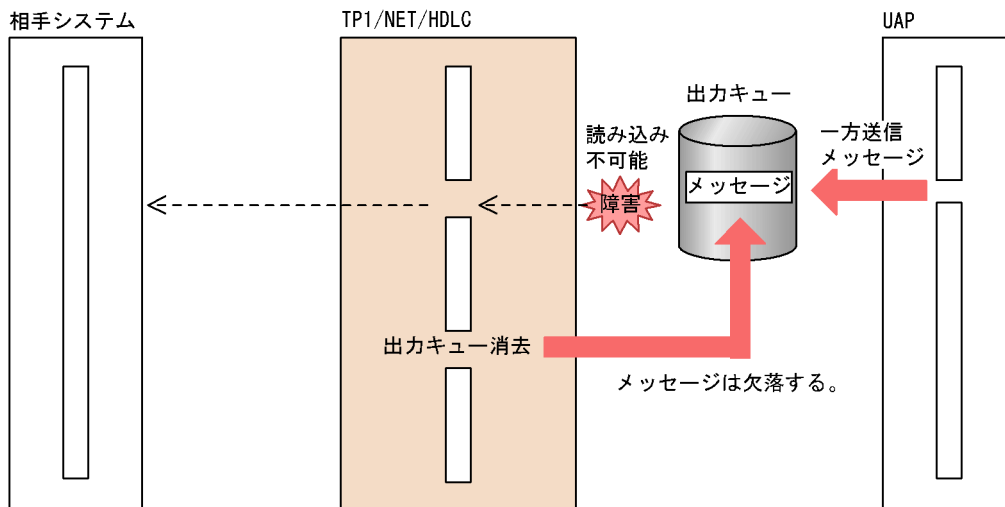
図 8-8 メッセージ書き込み時の障害



8.6 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。メッセージ読み込み時に障害が発生した場合の処理の流れを次の図に示します。

図 8-9 メッセージ読み込み時の障害



付録

付録 A メッセージ送受信の処理の流れ

付録 B 障害発生時の処理の流れ

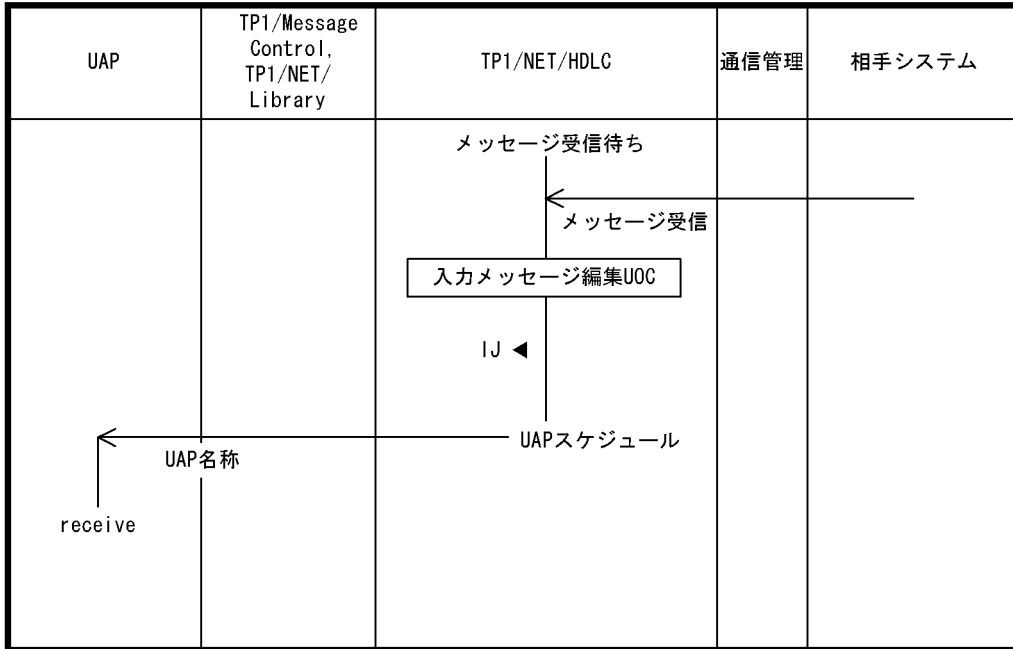
付録 C 理由コード一覧

付録 D 用語解説

付録 A メッセージ送受信の処理の流れ

メッセージを送受信するときのデータの流れ，ジャーナルの取得タイミングなどを，メッセージの種類ごとに示します。

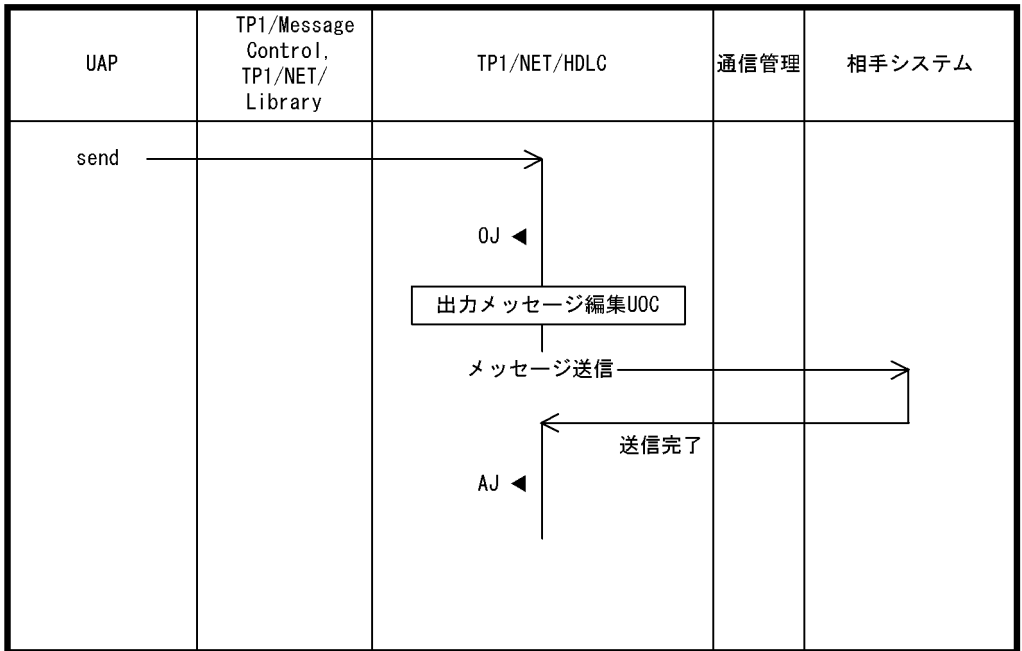
図 A-1 一方送信メッセージの処理の流れ（メッセージ受信）



(凡例)

IJ ◀: メッセージ入力ジャーナル取得

図 A-2 一方送信メッセージの処理の流れ（メッセージ送信）



(凡例)

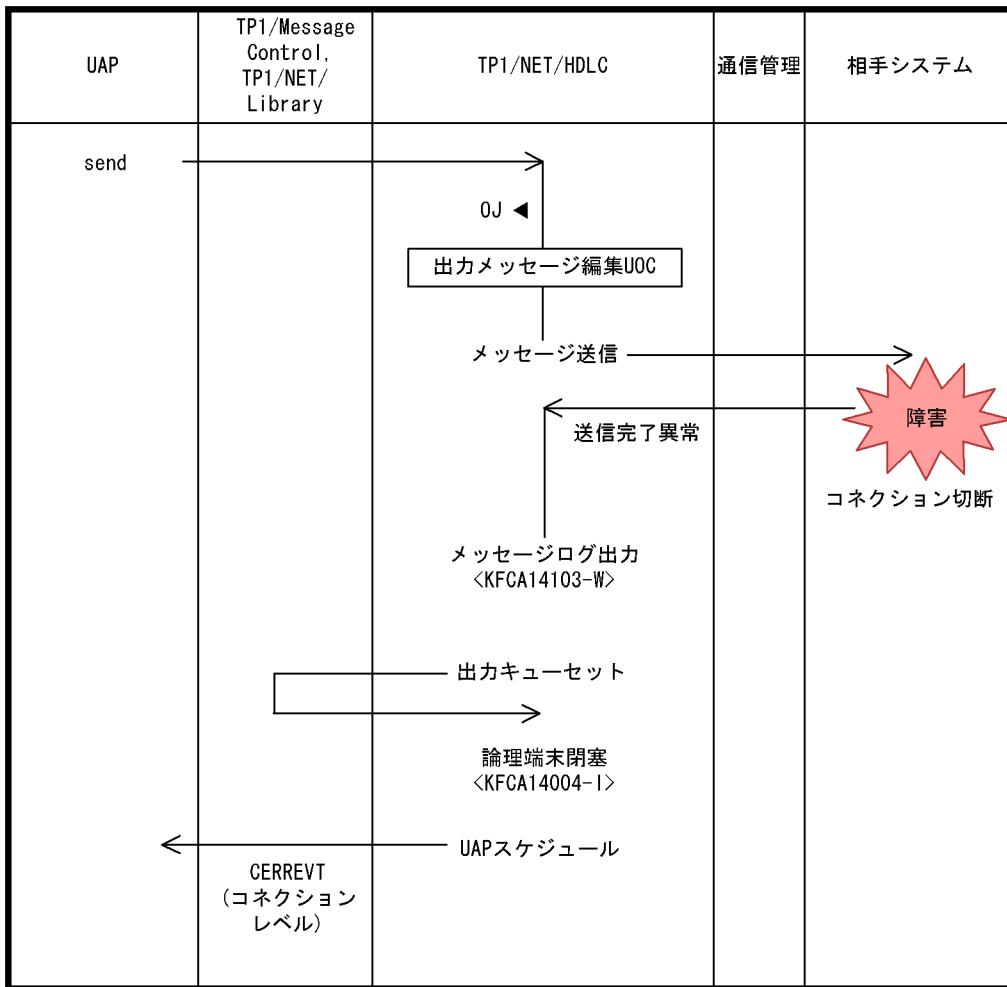
0J ◀: メッセージ出力ジャーナル取得

AJ ◀: メッセージ送信完了ジャーナル取得

付録 B 障害発生時の処理の流れ

メッセージを送受信するときに発生した障害処理の流れについて示します。

図 B-1 メッセージ送信時コネクション障害の処理の流れ（回復可能障害の場合）



(凡例)

0J ◀: メッセージ出力ジャーナル取得

図 B-2 コネクション切断時の処理の流れ（回復可能障害，手動再確立，および論理端末自動起動の場合）

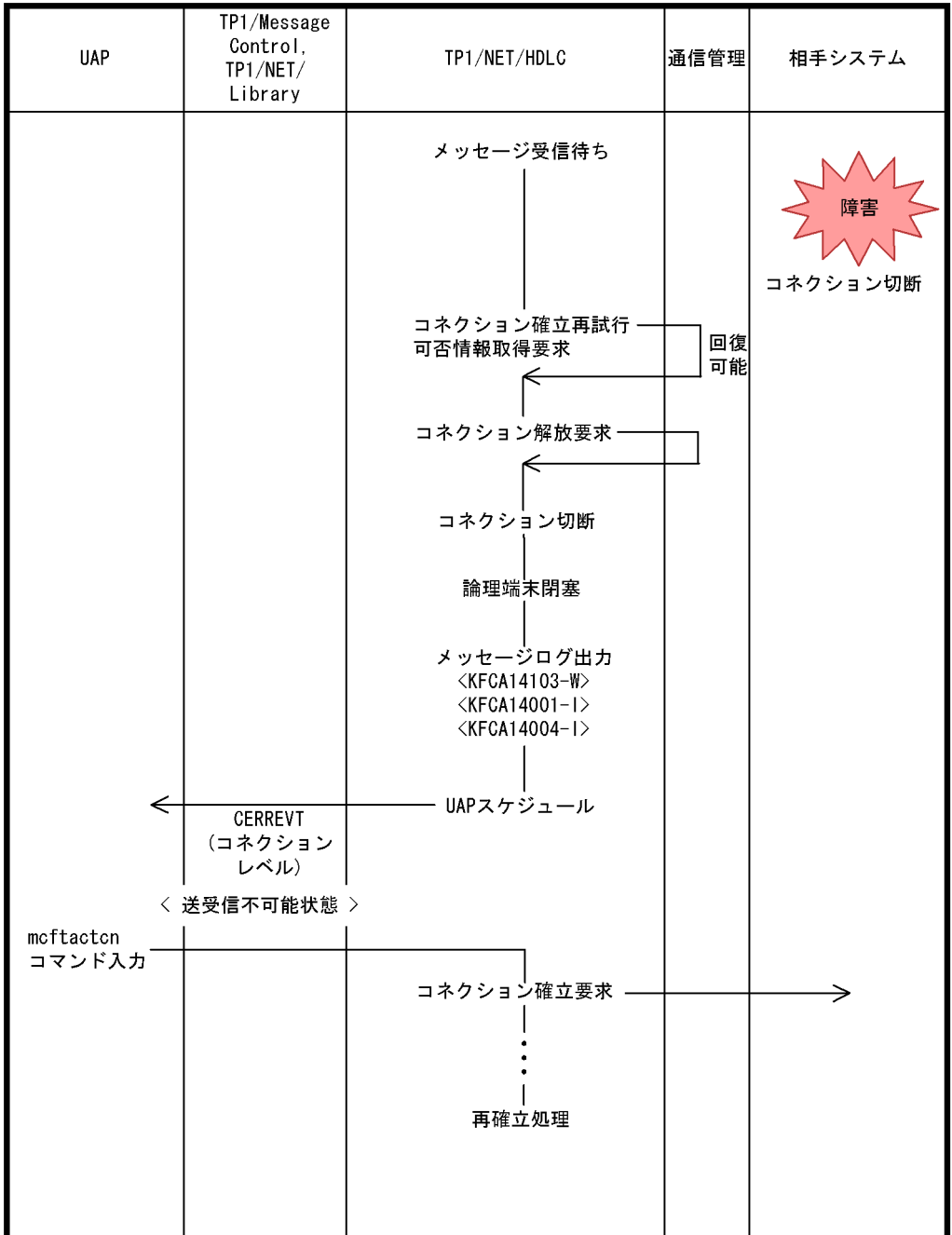


図 B-3 入力メッセージ編集 UOC エラーの処理の流れ

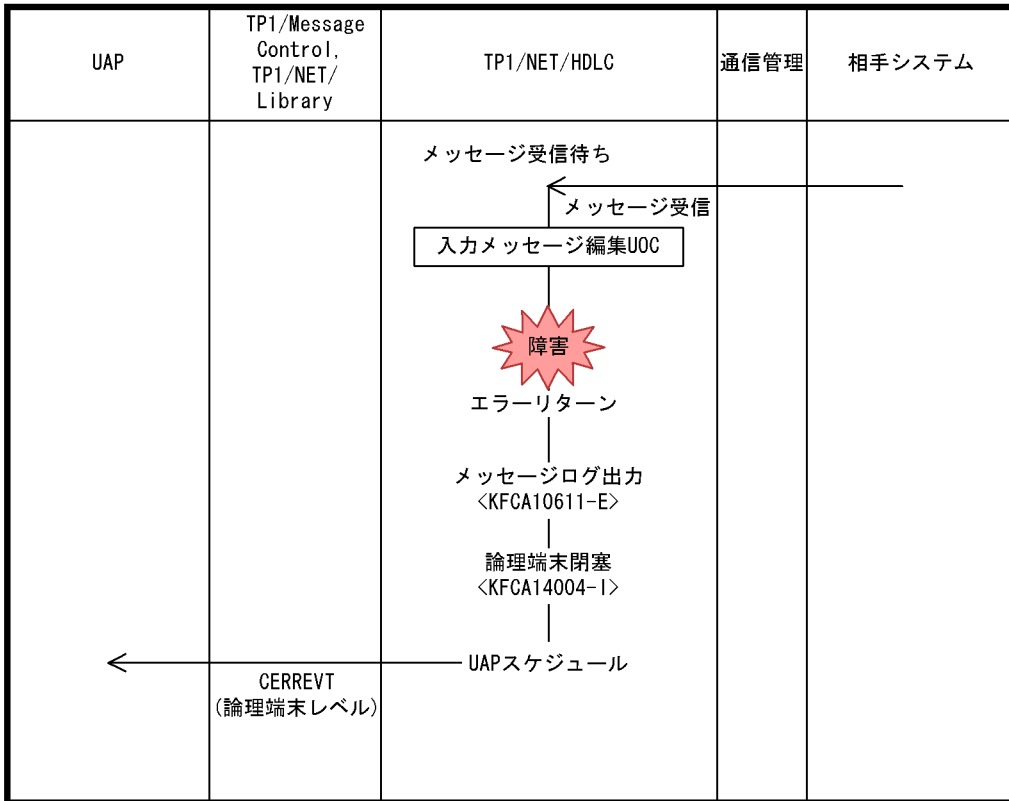
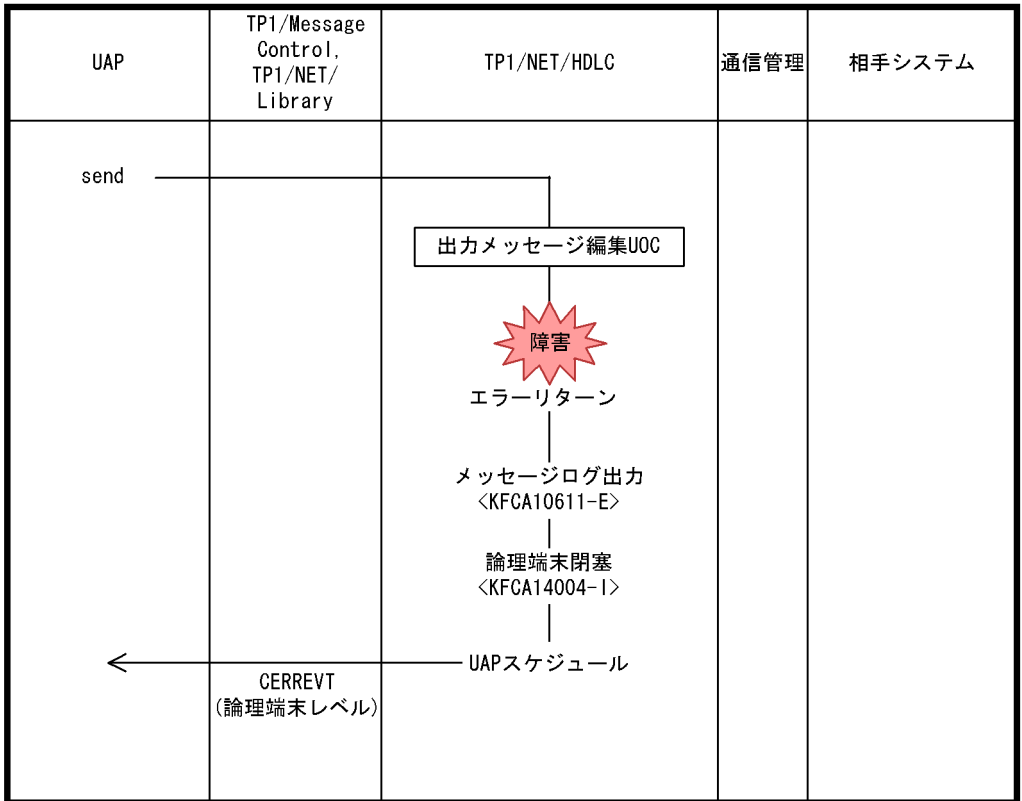


図 B-4 出力メッセージ編集 UOC エラーの処理の流れ



付録 C 理由コード一覧

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

表 C-1 ERREVT2 の理由コード一覧

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

表 C-2 CERREVT の理由コード一覧

理由コード 1 (16 進数)	理由コード 2 (16 進数)	発生条件	発生箇所
DCMHDS_RSN1_MCF (0x00000001)(MCF 障害)	DCMHDS_RSN2_CNNC (0x00000000)	コネクション確立不可能	コネクション
	DCMHDS_RSN2_CNER (0x00000001)	コネクション切断	コネクション
	DCMHDS_RSN2_RBFO F (0x00000002)	受信バッファオーバーフロー	コネクション

理由コード 1 (16 進数)	理由コード 2 (16 進数)	発生条件	発生箇所
	DCMHDS_RSN2_ERRO R (0x00000003)	受信バッファ面数の定義 または MCF 内部論理矛 盾	コネクショ ン
	DCMHDS_RSN2_XSBO F (0x00000011)	通信管理の送信バッファ オーバーフロー	コネクショ ン
	DCMHDS_RSN2_SBF0 F (0x00000004)	送信バッファオーバーパ フロー	論理端末
	DCMHDS_RSN2_OTGE T (0x00000005)	出力キュー読み込み障害	論理端末
	DCMHDS_RSN2_SBFST (0x00000006)	送信バッファ不足	論理端末
	DCMHDS_RSN2_DCTL E (0x00000012)	送信処理仕掛け中の mcftdctl コマンド入力	論理端末
DCMHDS_RSN1_UOC (0x00000003)(UOC 障害)	UOC からの詳細リターン コード	UOC エラーリターン	論理端末
	DCMHDS_RSN2_BCNT (0x00000001)	使用バッファ数不正	論理端末
	DCMHDS_RSN2_SEG (0x00000002)	有効セグメント不正	論理端末
	DCMHDS_RSN2_BADR (0x00000003)	編集バッファアドレス不 正	論理端末
	DCMHDS_RSN2_TIDP (0x00000004)	タイマセット中のタイマ セット指示	論理端末
	DCMHDS_RSN2_TIVL (0x00000005)	タイマセット値不正	論理端末
	DCMHDS_RSN2_TICD (0x00000006)	タイマセット/リセット 指示種別不正	論理端末

付録 D 用語解説

(英字)

AJ (メッセージ送信完了ジャーナル)

TP1/NET/HDLC 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)

メッセージ受信時に取得される情報です。メッセージ長、論理端末名称などで構成されます。

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

IJ (メッセージ入力ジャーナル)

入力キューに入力された情報です。入力メッセージ通番、論理端末名称、メッセージ種別、および入力メッセージの情報です。

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)

メッセージに関する情報です。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す。

OJ の取得タイミングは UAP から send を受け付けた時です。

(力行)

コネクション

TP1/NET/HDLC が AP 間通信をするときに、自システムと相手システムの間に確立する論理的通信路です。

(サ行)

出力キュー

処理の結果として出力メッセージを管理する待ち行列です。

(ナ行)

入力キュー

処理要求として入力メッセージを管理する待ち行列です。

(ラ行)

論理端末

TP1/NET/HDLC と UAP との通信接点です。TP1/NET/HDLC 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 (メッセージ送信完了ジャーナル) 166
AP 間通信 2
AP 間通信の概要 2
AP 間通信の形態 3
AP 間通信の仕組み 10
AP 間通信メッセージの送受信 16

C

CBLDCMCF('RECEIVE') 34
CBLDCMCF('RESEND') 39
CBLDCMCF('SEND') 45
CCLSEVT 87, 95
CERREVT 86, 94
CERREVT の理由コード一覧 164
COBOL 言語のメッセージ送受信 20
COPNEVT 87, 95
C 言語のメッセージ送受信 20

D

dc_mcf_receive 22
dc_mcf_resend 26
dc_mcf_send 30

E

ERREVT1 83, 88
ERREVT2 83, 89
ERREVT2 の理由コード一覧 164
ERREVT3 84, 90
ERREVT4 85, 92

G

GJ (メッセージ受信ジャーナル) 166

I

IJ (メッセージ入力ジャーナル) 166

M

max_open_fds 112
max_socket_descriptors 111
MCF 6
mcfactcn 121
mcfactcle 123
mcfalccn 103
mcfalced 109
mcfalcle 107
mcfbuf 103, 104, 106
mcfdetcn 125
mcfdetcle 127
mcfllscn 129
mcfllsle 132
MCF アプリケーション定義 98
MCF イベント 80
MCF イベントインタフェース 80
MCF イベント処理用 MHP 82
MCF イベント通知時のセグメント構成 81
MCF イベントの共通ヘッダ 82
MCF イベントの種類 80
MCF サービス名の登録 136
MCF 通信構成定義 98
MCF 通信プロセスでアクセスするファイルの最大数 112
MCF 定義オブジェクトの生成 113
MCF で使用する定義ファイル 98
MCF の障害 148
MCF マネージャ定義 98
MCF メイン関数の作成 136, 137
MJ (メッセージジャーナル) 166

O

OJ (メッセージ出力ジャーナル) 166

R

RECEIVE 51

S

SCMPEVT 85, 93
SEND 54

T

TP1/Message Control 6
TP1/NET/HDLC が適用する AP 間通信の
プロトコル 7
TP1/NET/HDLC で適用する伝送動作モード
4
TP1/NET/HDLC の運用コマンド 120
TP1/NET/HDLC の組み込みの流れ 136
TP1/NET/HDLC の定義の概要 98
TP1/NET/HDLC を組み込んだソフトウェア
構成 7
TP1/NET/HDLC を使用したネットワーク構
成例 2
TP1/NET/Library 6

U

UAP 異常終了通知イベント 80
UAP 障害 148, 152
UAP 閉塞 154
UOC 作成上の注意事項 79
UOC で使用できる関数 79
UOC の異常処理 79
UOC の構造 79
UOC の実行タイミング 79
UOC の詳細リターンコード 79

あ

アプリケーションの型 14
アプリケーション名 14
アプリケーション名の決定 66
アプリケーション名の決定の処理 67

い

一方受信形態 3
一方送信メッセージの処理の流れ(メッセ
ージ受信) 158

一方送信メッセージの処理の流れ(メッセ
ージ送信) 159
一方送信メッセージの送信(COBOL 言語)
45
一方送信メッセージの送信(C 言語) 30

う

運用コマンド 119

か

概要 1
仮想スロット番号 105

き

機能 9
キューグループ ID 108

く

組み込み方法 135

こ

コネクション 10, 14, 166
コネクション ID 103
コネクション障害 144, 149
コネクション障害時再試行回数 105
コネクション障害時再試行間隔 105
コネクション障害の処理の流れ 160
コネクション切断時の処理の流れ 161
コネクション定義の開始 103
コネクション定義の終了 109
コネクションと論理端末の関係 14
コネクションの解放 12, 125
コネクションの確立 10, 121
コネクションの確立と解放 10
コネクションの状態表示 129
コネクションの切断 13

し

システムサービス共通情報定義 111
システムサービス情報定義 110

システムサービス情報定義ファイルの作成
136

システム定義 97

受信スケジュール関係（入力キュー，入力
メッセージ編集 UOC）の障害 145

出力キュー 166

出力キュー障害 156

出力メッセージの編集 73

出力メッセージの編集 UOC 73

出力メッセージの割り当て先 108

出力メッセージ編集 UOC インタフェース
74

出力メッセージ編集 UOC エラーの処理の流
れ 163

障害対策 143

障害通知イベント 81

障害の種類と対応処理 144

障害発生時の処理の流れ 160

状態通知イベント 81

せ

切断理由コード 106

そ

送信完了通知イベント 81

送信スケジュール関係（出力キュー，出力
メッセージ編集 UOC）の障害 146

送信メッセージの通番編集 76

送信メッセージの通番編集 UOC インタ
フェース 78

ソケット用ファイル記述子の最大数 111

ソフトウェア構成の例 6

ソフトウェアの構成 6

た

タイマ起動メッセージ廃棄通知イベント 80
端末タイプ 107

つ

通信管理プログラムと関連づける内容 114

通信形態 3

て

定義オブジェクトファイルの生成 136, 140

定義の指定順序 102

定義例 115

ディスク出力メッセージ最大格納数 107

データ操作言語（COBOL 言語）のメッセー
ジ送受信 20

に

入力キュー 167

入力キュー障害 155

入力メッセージの編集 66

入力メッセージの編集とアプリケーション名
の決定 66

入力メッセージ編集 UOC 66

入力メッセージ編集 UOC インタフェース
68

入力メッセージ編集 UOC エラーの処理の流
れ 162

ふ

不正アプリケーション名検出通知イベント
80

分岐送信形態 3

み

未処理送信メッセージ廃棄通知イベント 81

め

メッセージ受信用バッファグループ番号 103

メッセージ制御機能 6

メッセージ送受信インタフェース 19

メッセージ送受信インタフェースの一覧 20

メッセージ送受信の処理の流れ 158

メッセージ送受信の通信文（データ操作言
語）21

メッセージ送受信の文（COBOL 言語）20

メッセージ送受信のライブラリ関数（C 言
語）20

メッセージ送信用バッファグループ番号 103

メッセージとセグメントの関係 15

メッセージの再送 (COBOL 言語) 39
メッセージの再送 (C 言語) 26
メッセージの受信 17
メッセージの受信 (COBOL 言語) 34
メッセージの受信 (C 言語) 22
メッセージの受信 (データ操作言語) 51
メッセージの送信 16
メッセージの送信 (データ操作言語) 54
メッセージ廃棄通知イベント 80
メッセージ編集用バッファグループ番号 104
メッセージ編集用バッファ数 104
メモリ出力メッセージ最大格納数 107

ゆ

ユーザアプリケーションプログラム作成例
58
ユーザOWNコーディング 66
ユーザOWNコーディングインタフェース
66

り

理由コード一覧 164

ろ

論理端末 14, 167
論理端末, メッセージ, アプリケーションの
型, UAP インタフェース, および通信形態
の関係 15
論理端末定義 107
論理端末とアプリケーションの型の関係 14
論理端末の状態表示 132
論理端末の端末タイプ 14
論理端末の閉塞 127
論理端末の閉塞解除 123
論理端末名称 107

ソフトウェアマニュアルのサービス ご案内

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

1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しております。

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

ホームページのメニューは次のとおりです。

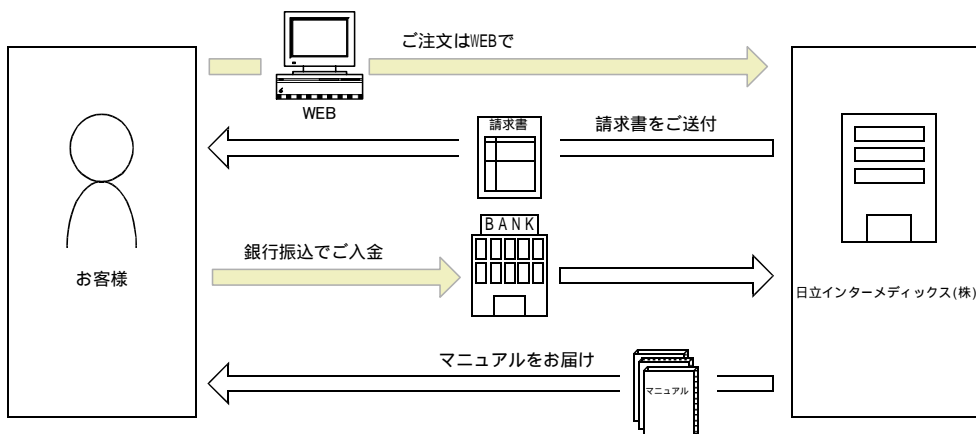
マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル情報	複数マニュアルを格納したCD-ROMマニュアルを提供しています。どの製品に対応したCD-ROMマニュアルがあるか、を参照できます。
マニュアルのご購入	日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からお申し込みできます。 (詳細は「3. マニュアルのご注文」を参照してください。)
Web提供マニュアル一覧	インターネットで参照できるマニュアルの一覧を提供しています。 (詳細は「2. インターネットからのマニュアル参照」を参照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

2. インターネットからのマニュアル参照(ソフトウェアサポートサービス)

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

3. マニュアルのご注文

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



下記 URL にアクセスして必要事項を入力してください。

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

ご注文いただいたマニュアルについて、請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。なお、送料は弊社で負担します。

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