

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/OSI-TP 編

解説・手引・文法・操作書

3000-3-D73

マニュアルの購入方法

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

対象製品

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

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

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

P-F1M64-32314 uCosminexus TP1/NET/OSI-TP 07-00

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

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

輸出時の注意

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

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

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

発行

2009 年 1 月 (第 1 版) 3000-3-D73

著作権

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

はじめに

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

このマニュアルに記載するプログラムプロダクトは次のとおりです。

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

- P-1M64-3131 uCosminexus TP1/Message Control
- P-1M64-3231 uCosminexus TP1/NET/Library
- P-F1M64-32314 uCosminexus TP1/NET/OSI-TP

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 Version7 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 OSI TP プロトコルを使用するシステム管理者およびシステム設計者を対象としています。また、オンラインや OpenTP1 システムの基礎的な知識を持っていて、次に示すマニュアルを理解し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/OSI-TP を使用したシステム間の通信 (AP 間通信) の概要について説明しています。

第 2 章 機能

TP1/NET/OSI-TP が設定する接続の確立方法、メッセージの種類と送受信の方法などについて説明しています。

第 3 章 メッセージ送受信インタフェース

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

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

はじめに

第 5 章 システム定義

OSI TP 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/OSI-TP 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、相手システムの通信定義と関連づける内容、およびシステム定義例について説明しています。

第 6 章 運用コマンド

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

第 8 章 障害対策

TP1/NET/OSI-TP の運用中に発生するおそれがある障害と、TP1/NET/OSI-TP 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 TP1/NET/OSI-TP の実装範囲

TP1/NET/OSI-TP の実装範囲について説明しています。

付録 B メッセージ送受信の処理の流れ

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

付録 C 障害発生時の処理の流れ

障害が発生した場合の処理の流れについて説明しています。

付録 D 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 E ユーザアプリケーションプログラムのコーディング例

TP1/NET/OSI-TP のユーザアプリケーションプログラムコーディング例について説明しています。

付録 F 旧製品からの移行に関する注意事項

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

付録 G バージョンアップ時の変更点

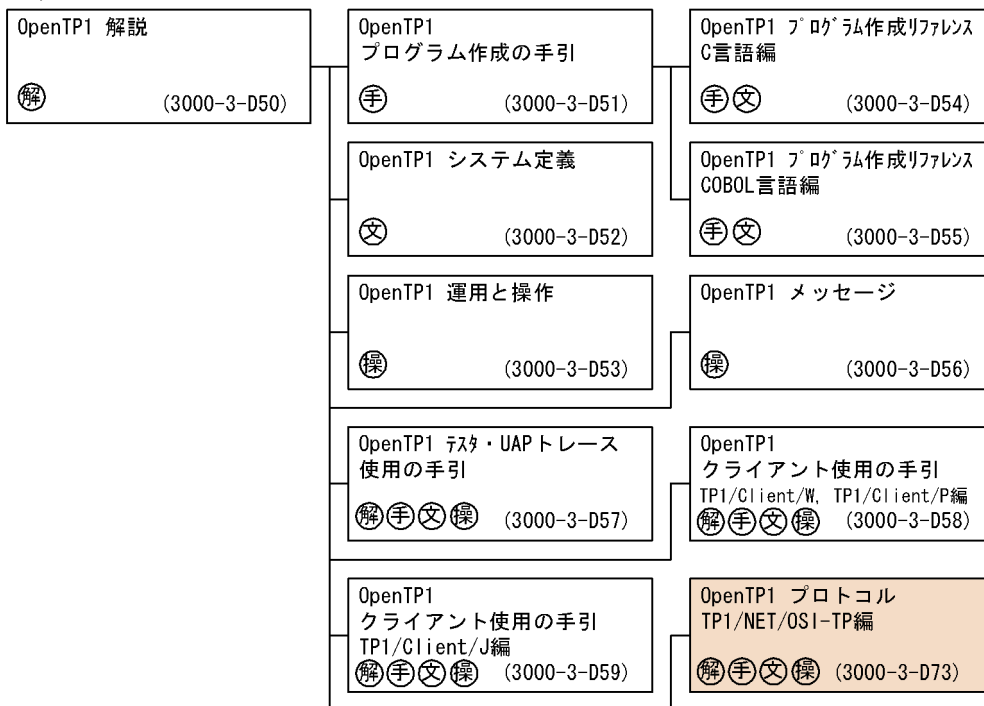
各バージョンでの関数、定義およびコマンドの変更点について説明しています。

付録 H 用語解説

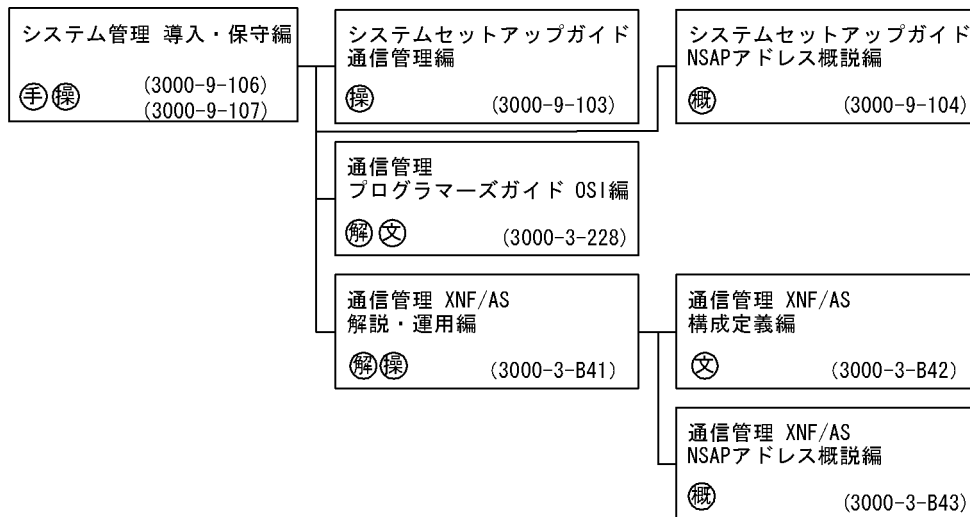
TP1/NET/OSI-TP で使用する用語について説明しています。

関連マニュアル

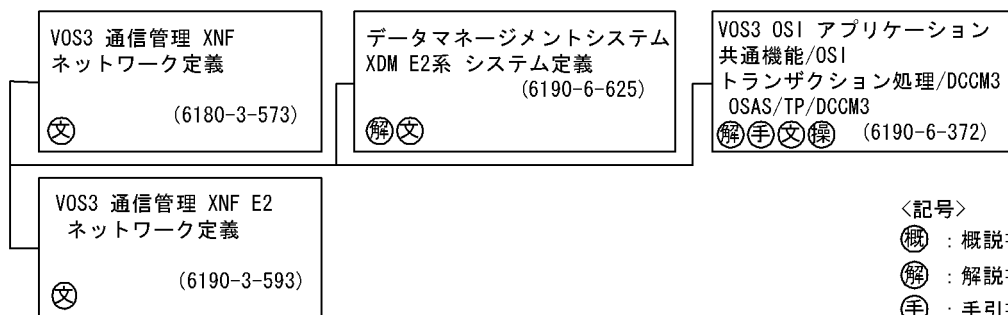
●OpenTP1 Version 7



●通信管理



●通信相手システム

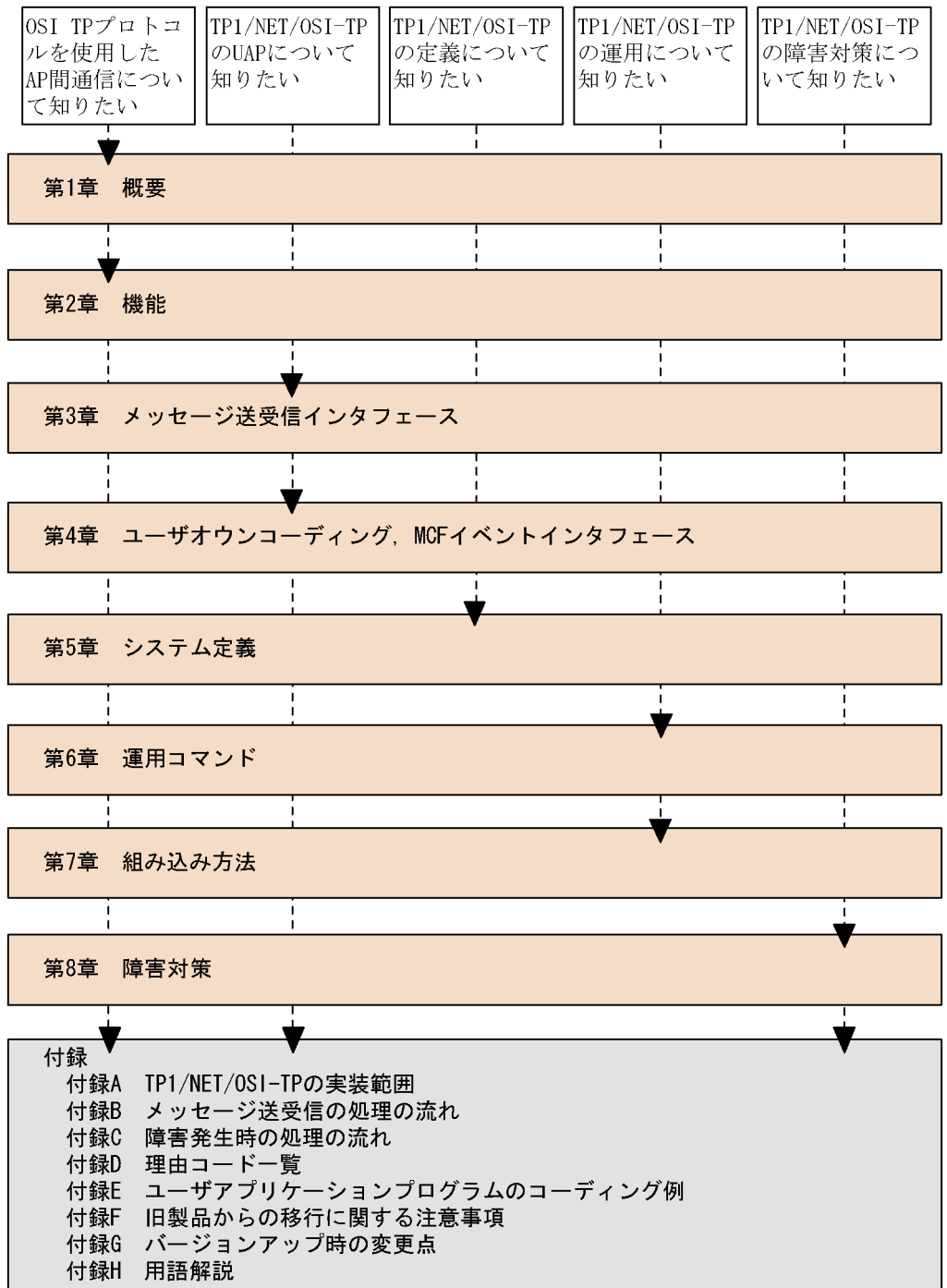


<記号>


- 概 : 概説書
- 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



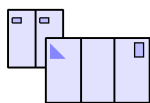
●論理回線



●入出力の動作



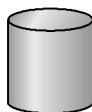
●ホストコンピュータ



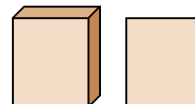
●プログラムの流れ



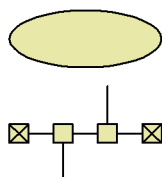
●ファイル



●プログラム



●ネットワーク



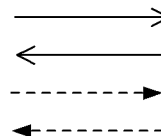
●データの流れ



●通信回線



●制御の流れ



●障害



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)

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)	この記号で仕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することを示します。 (例) { INPUT I-O } INPUT と I-O のうち、どちらかを指定できることを示します。
{{ }}	この記号で囲まれた複数の項目が一つの繰り返し項目の単位であることを示します。 (例) {{ mcftalccn mcftalclc mcftalccd }} mcftalccn, mcftalclc および mcftalccd を一組として指定することを示します。
— (下線)	この記号で示す項目は、オペランド、オプションまたはコマンド引数を省略した場合の省略時解釈値を示します。 (例) -i auto <u>manual</u> -i オプションを省略した場合、manual を省略時解釈値とすることを示します。 ただし、データ操作言語の説明の場合、この下線記号で示す予約語は、必要語なので省略できないことを示します。 下線がない予約語は、補助語なので書いても書かなくてもかまいません。
...	この記号で示す直前の一つの項目を繰り返し指定できることを示します。 ただし、項目が括弧で囲まれている場合、括弧全体が一つの項目となります。
(白三角)	空白を示します。 (例) コネクション ID1 コネクション ID2 コネクション ID1 とコネクション ID2 の間に、空白を 1 個入力することを示します。

(2) 属性表示記号

ユーザ指定値の範囲などを説明する記号です。

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

(3) 構文要素記号

ユーザ指定値の内容を説明する記号です。

はじめに

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

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/OSI-TP	TP1/NET/OSI-TP	
uCosminexus TP1/Server Base	TP1/Server Base	
XNF/AS/ACONARC	XNF/AS	XNF
XNF/AS/Base		
XNF/AS/Host Adaptor		
XNF/AS/OSI Extension		
XNF/AS/OSI Extension/Cluster		
XNF/AS/WAN		

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
ACSE	<u>A</u> ssociation <u>C</u> ontrol <u>S</u> ervice <u>E</u> lement
AP	<u>A</u> pplication <u>P</u> rogram
CCP	<u>C</u> ommunication <u>C</u> ontrol <u>P</u> rocessor
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
OSI TP	<u>O</u> pen <u>S</u> ystems <u>I</u> nterconnection <u>T</u> ransaction <u>P</u> rocessing
PSAP	<u>P</u> resentation <u>S</u> ervice <u>A</u> ccess <u>P</u> oint
RPC	<u>R</u> emote <u>P</u> rocedure <u>C</u> all
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram
UAP	<u>U</u> ser <u>A</u> pplication <u>P</u> rogram
UOC	<u>U</u> ser <u>O</u> wn <u>C</u> oding
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） 必須（ひつす） 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。

はじめに

ただし、その場合には、その刊行物のまえがきの一部として、次の謝辞を掲載しなければならない。書評などに短い文章を引用するときは、"COBOL" という名称を示せば謝辞全体を掲載する必要はない。

COBOL は産業界の言語であり、特定の団体や組織の所有物ではない。

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

次に示す著作権表示付資料の著作者及び著作権者

FLOW-MATIC (Sperry Rand Corporation の商標) ,

Programming for the Univac (R) I and II , Data Automation Systems ,
Sperry Rand Corporation 著作権表示 1958 年 , 1959 年 ;

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

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

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

目次

1	概要	1
1.1	AP 間通信の概要	2
1.2	AP 間通信の形態	4
1.2.1	通信形態	4
1.2.2	適用範囲	6
1.3	ソフトウェアの構成	10
1.3.1	前提プログラム	10
1.3.2	ソフトウェア構成の例	10
2	機能	13
2.1	AP 間通信の仕組み	14
2.1.1	コネクションの確立と解放	14
2.1.2	コネクションと論理端末の関係	20
2.1.3	コネクションと PSAP の関係	21
2.1.4	経路交代	23
2.1.5	論理端末とアプリケーションの型の関係	23
2.1.6	論理端末と UAP の関係	24
2.1.7	全二重と半二重	24
2.1.8	メッセージの分割と組み立て	25
2.1.9	アソシエーションプール管理機能	26
2.2	AP 間通信メッセージの送受信	28
2.2.1	非同期型のメッセージの受信	28
2.2.2	同期型のメッセージの送受信	29
2.2.3	制御権	32
2.2.4	ダイアログ	34
2.2.5	アプリケーション名の決定	38
2.2.6	ハンドシェイク	40
2.2.7	データ転送	42
2.2.8	エラー通知	42
2.2.9	APDU 連結	47

3	メッセージ送受信インタフェース	49
	メッセージ送受信インタフェースの一覧	51
	dc_mcf_receive - 一方送信メッセージの受信 (C 言語)	53
	dc_mcf_recvsync - 同期型メッセージの受信 (C 言語)	57
	dc_mcf_sendrecv - 同期型メッセージの送受信 (C 言語)	61
	dc_mcf_sendsync - 同期型メッセージの送信 (C 言語)	67
	CBLDCMCF('RECEIVE') - 一方送信メッセージの受信 (COBOL 言語)	72
	CBLDCMCF('RECVSYNC') - 同期型メッセージの受信 (COBOL 言語)	77
	CBLDCMCF('SENDRECV') - 同期型メッセージの送受信 (COBOL 言語)	82
	CBLDCMCF('SENDSYNC') - 同期型メッセージの送信 (COBOL 言語)	89
	DISABLE - 同期型メッセージの送信 (データ操作言語)	95
	ENABLE - 同期型メッセージの送信 (データ操作言語)	98
	RECEIVE - メッセージの受信 (データ操作言語)	101
	SEND - 同期型メッセージの送受信 (データ操作言語)	105
	サービスプリミティブ	110
	ユーザアプリケーションプログラム作成例	123
4	ユーザオウンコーディング, MCF イベントインタフェース	125
	4.1 ユーザオウンコーディングインタフェース	126
	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	126
	4.1.2 入力メッセージ編集 UOC インタフェース	127
	4.1.3 出力メッセージの編集	133
	4.1.4 出力メッセージ編集 UOC インタフェース	134
	4.1.5 UOC 作成上の注意事項	137
	4.2 MCF イベントインタフェース	139
	4.2.1 MCF イベントの種類	139
	4.2.2 MCF イベント通知時のセグメント構成	140
	4.2.3 MCF イベント情報の形式 (C 言語)	140
	4.2.4 MCF イベント情報の形式 (COBOL 言語)	145
5	システム定義	153
	TP1/NET/OSI-TP の定義の概要	154
	TP1/NET/OSI-TP 固有のシステム定義の種類	156

mcfmuap (UAP 共通定義)	160
mcftalccn (コネクション定義の開始)	161
mcftalced (コネクション定義の終了)	170
mcftalcle (論理端末定義)	171
mcftgrpcn (コネクショングループ定義の開始)	172
mcftgrped (コネクショングループ定義の終了)	173
システムサービス情報定義	174
システムサービス共通情報定義	175
MCF 定義オブジェクトの生成	177
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	178
相手システムの通信定義と関連づける内容	181
定義例	183

6 運用コマンド	189
TP1/NET/OSI-TP の運用コマンド	190
mcftactcn (コネクションの確立)	191
mcftdctcn (コネクションの解放)	194
mcftlscn (コネクションの状態表示)	197

7 組み込み方法	201
7.1 TP1/NET/OSI-TP の組み込みの流れ	202
7.2 MCF メイン関数の作成	203
7.3 定義オブジェクトファイルの生成	206

8 障害対策	209
8.1 障害の種類と対応処理	210
8.2 コネクション障害	216
8.3 ダイアログ障害	217
8.4 ユーザアプリケーションプログラム異常終了	218
8.5 ユーザアプリケーションプログラム閉塞	219

付録	221
付録 A TP1/NET/OSI-TP の実装範囲	222

付録 B	メッセージ送受信の処理の流れ	226
付録 C	障害発生時の処理の流れ	230
付録 D	理由コード一覧	232
付録 E	ユーザアプリケーションプログラムのコーディング例	234
付録 E.1	ユーザアプリケーションプログラムのコーディング例（全二重での C 言語）	234
付録 E.2	ユーザアプリケーションプログラムのコーディング例（全二重での COBOL 言語）	245
付録 E.3	ユーザアプリケーションプログラムのコーディング例（全二重でのデータ操作言語）	253
付録 E.4	ユーザアプリケーションプログラムのコーディング例（半二重での C 言語）	260
付録 E.5	ユーザアプリケーションプログラムのコーディング例（半二重での COBOL 言語）	272
付録 E.6	ユーザアプリケーションプログラムのコーディング例（半二重でのデータ操作言語）	281
付録 F	旧製品からの移行に関する注意事項	289
付録 G	バージョンアップ時の変更点	290
付録 G.1	07-00 での変更点	290
付録 H	用語解説	291

索引

293

目次		
図 1-1	TP1/NET/OSI-TP を使用したネットワーク構成の例	2
図 1-2	TP1/NET/OSI-TP の OSI7 層構造の中での機能範囲	3
図 1-3	TP1/NET/OSI-TP を使用した AP 間通信の例	5
図 1-4	メッセージの送達確認の例	7
図 1-5	メッセージの形式の例	8
図 1-6	メッセージの応答を不要とした場合	8
図 1-7	メッセージの応答を必要とした場合	9
図 1-8	TP1/NET/OSI-TP を組み込んだソフトウェア構成の例	11
図 2-1	コネクションの確立（発呼型）	15
図 2-2	コネクションの確立（着呼型）	16
図 2-3	TP1/NET/OSI-TP からのコネクションの解放	17
図 2-4	相手システムからのコネクションの解放	18
図 2-5	TP1/NET/OSI-TP からのコネクションの強制解放	19
図 2-6	相手システムの異常による解放	19
図 2-7	コネクションの切断	20
図 2-8	コネクションと PSAP の関係	22
図 2-9	論理端末（コネクション）と UAP の関係	24
図 2-10	処理層とデータの関係（単一 APDU の場合）	25
図 2-11	処理層とデータの関係（連結 APDU の場合）	26
図 2-12	アソシエーションプール管理機能の概要	27
図 2-13	非同期型のメッセージの受信	29
図 2-14	同期型のメッセージの送信，受信，および送受信	31
図 2-15	制御権委譲	32
図 2-16	制御権要求	33
図 2-17	ダイアログの開始と終了	36
図 2-18	応答を不要とした場合のメッセージ破棄の例	37
図 2-19	アプリケーション名の決定	39
図 2-20	ハンドシェイク	40
図 2-21	制御権委譲付きハンドシェイク	41
図 2-22	データ転送	42
図 2-23	UAP でのエラー発生時の処理（全二重の場合）	43
図 2-24	相手システムからのエラー通知受信の処理（全二重の場合）	43
図 2-25	UAP でのエラー発生時の処理（制御権がある場合）	44

図 2-26	相手システムからのエラー通知受信の処理（制御権がない場合）	45
図 2-27	UAP でのエラー発生時の処理（制御権がない場合）	46
図 2-28	相手システムからのエラー通知受信の処理（制御権がある場合）	46
図 2-29	APDU 連結	47
図 3-1	送受信メッセージの形式	114
図 3-2	サービスプリミティブの形式	115
図 3-3	制御情報の形式	115
図 3-4	パラメタの形式	115
図 3-5	ユーザ情報の形式	116
図 3-6	処理の流れ（全二重の場合）	123
図 3-7	処理の流れ（半二重の場合）	124
図 4-1	UOC インタフェース用のパラメタとバッファの関係	131
図 4-2	入力メッセージ編集 UOC のバッファインタフェース（APDU 連結なし）	132
図 4-3	入力メッセージ編集 UOC のバッファインタフェース（APDU 連結あり）	132
図 4-4	受信メッセージの連結	133
図 4-5	出力メッセージ編集 UOC のバッファインタフェース（APDU 連結あり）	137
図 4-6	MCF イベント通知時のセグメント構成	140
図 5-1	TP1/NET/OSI-TP のプロトコル固有定義コマンドの指定順序	159
図 5-2	通信管理プログラムによる相手システムとの接続方法および接続時の機能の違い	178
図 5-3	ネットワーク構成の例	181
図 5-4	OSAS/TP/DCCM3 の定義との関係	182
図 5-5	TP1/NET/OSI-TP のシステム構成例	184
図 7-1	MCF メイン関数のコーディング概要（ANSI C, C++ の場合）	203
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）	204
図 7-3	MCF メイン関数のディレクトリへの組み込み方法の概要	205
図 7-4	定義オブジェクトファイルの作成方法の概要	207
図 8-1	メッセージ送受信時のコネクション障害	216
図 8-2	ダイアログ障害	217
図 8-3	ダイアログ開始要求受信時の MHP 異常終了	218
図 8-4	UAP 閉塞による障害	219
図 B-1	一方送信メッセージの受信	226
図 B-2	同期型のメッセージの送信	227
図 B-3	同期型のメッセージの受信	228
図 B-4	同期型のメッセージの送受信	229
図 C-1	通信回線障害	230

表目次

表 1-1	TP1/NET/OSI-TP に適用する AP 間通信のプロトコル	11
表 2-1	TP1/NET/OSI-TP の TP プロトコルデータ単位	14
表 2-2	論理端末の端末タイプ、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	23
表 2-3	TP1/NET/OSI-TP が提供するサービスの種類	28
表 2-4	制御権が移動する契機	33
表 3-1	メッセージ送受信の関数（C 言語）	51
表 3-2	メッセージ送受信の関数に対応するプログラム（COBOL 言語）	51
表 3-3	メッセージ送受信の通信文（データ操作言語）	52
表 3-4	基本サービスプリミティブ	110
表 3-5	サービスプリミティブと UAP との対応	111
表 3-6	サービスプリミティブの連結パターンと UAP との対応	113
表 3-7	サービスプリミティブとパラメタとの対応	116
表 3-8	サービスプリミティブの突き合わせ項目（コネクション確立時）	121
表 3-9	サービスプリミティブの突き合わせ項目（ダイアログ開始時）	122
表 4-1	TP1/NET/OSI-TP が発生させる MCF イベントの種類	139
表 4-2	COBOL 言語の MCF イベント情報の内容（ERREVT1）	145
表 4-3	COBOL 言語の MCF イベント情報の内容（ERREVT2）	146
表 4-4	COBOL 言語の MCF イベント情報の内容（ERREVT3）	148
表 4-5	COBOL 言語の MCF イベント情報の内容（ERREVTA）	150
表 4-6	COBOL 言語の MCF イベント情報の内容（CERREVT）	151
表 4-7	COBOL 言語の MCF イベント情報の内容（COPNEVT, CCLSEVT）	152
表 5-1	MCF で使用する定義ファイル	154
表 5-2	TP1/NET/OSI-TP 固有の定義の種類	156
表 5-3	回線接続する場合の XNF/AS の定義との関連づけ	179
表 5-4	チャンネル接続する場合の XNF/AS の定義との関連づけ	179
表 5-5	OSI 拡張機能を使用する場合の XNF/AS の定義との関連づけ	180
表 6-1	TP1/NET/OSI-TP で使用する運用コマンド	190
表 8-1	コネクション障害と対応処理	210
表 8-2	受信スケジュール関係の障害と対応処理	211
表 8-3	送信スケジュール関係の障害と対応処理	213
表 8-4	UAP の障害と対応処理	214
表 8-5	ジャーナルの障害と対応処理	214

表 8-6	関数のリターン処理の障害と対応処理	214
表 8-7	プロシジャの障害と対応処理	215
表 A-1	TP1/NET/OSI-TP のサービスプリミティブおよび PDU	222
表 A-2	TP1/NET/OSI-TP のサービス詳細	223
表 D-1	ERREVT2 の理由コード一覧	232
表 D-2	CERREVT の理由コード一覧	232
表 E-1	TP1/NET/OSI-TP が提供するユーザアプリケーションプログラムのコーディング例	234
表 F-1	バージョン 6 以前で使用していたソースファイルの互換性	289
表 G-1	TP1/NET/OSI-TP 07-00 での動作の変更点	290

1

概要

TP1/NET/OSI-TP 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 OSI TP 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/OSI-TP 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をいいます。TP1/NET/OSI-TP は、OSI TP 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/OSI-TP 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。

OSI TP プロトコルとは、OSI 上位層（5 層、6 層、および 7 層）内の、業務対応の応用要素である TP を使用して、AP 間通信サービスを使用できるプロトコルです。

TP1/NET/OSI-TP を使用したネットワーク構成の例を図 1-1 に、TP1/NET/OSI-TP の OSI7 層の中での機能範囲を図 1-2 に示します。

図 1-1 TP1/NET/OSI-TP を使用したネットワーク構成の例

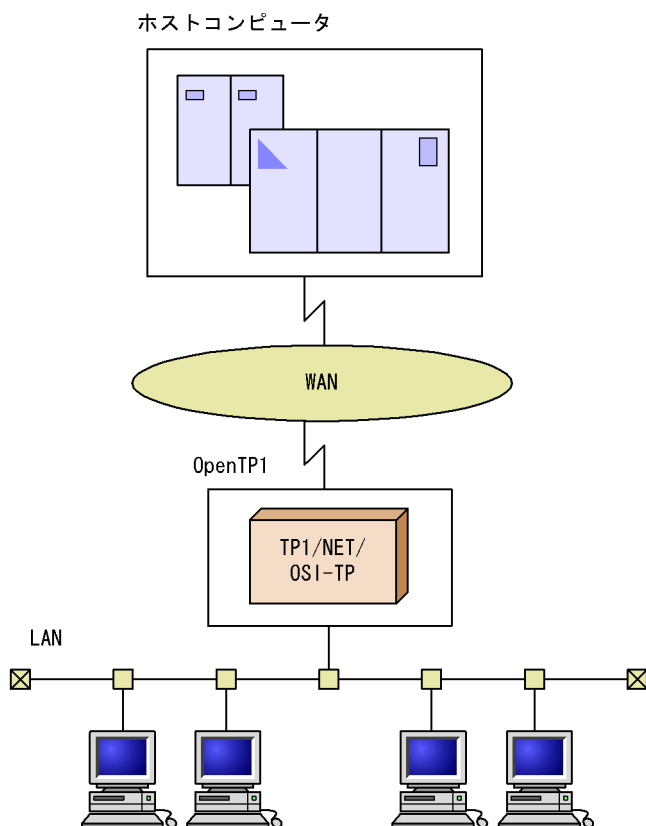
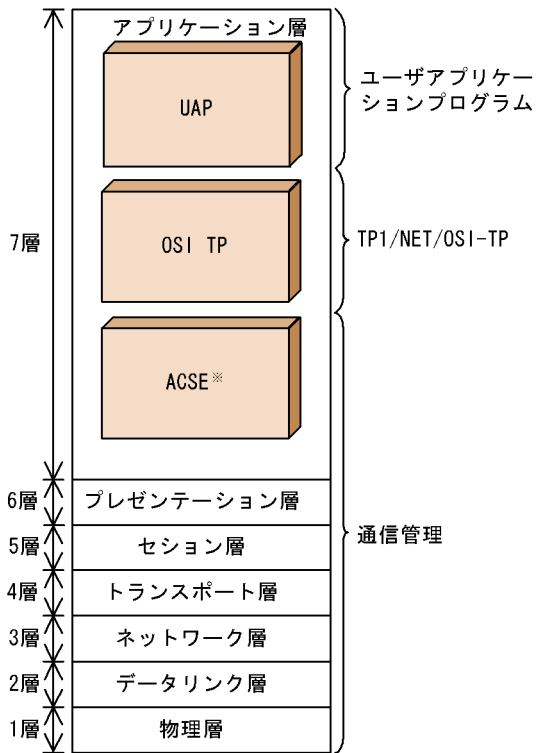


図 1-2 TP1/NET/OSI-TP の OSI7 層構造の中での機能範囲



注

ACSE : アソシエーション制御サービス要素

1.2 AP 間通信の形態

1.2.1 通信形態

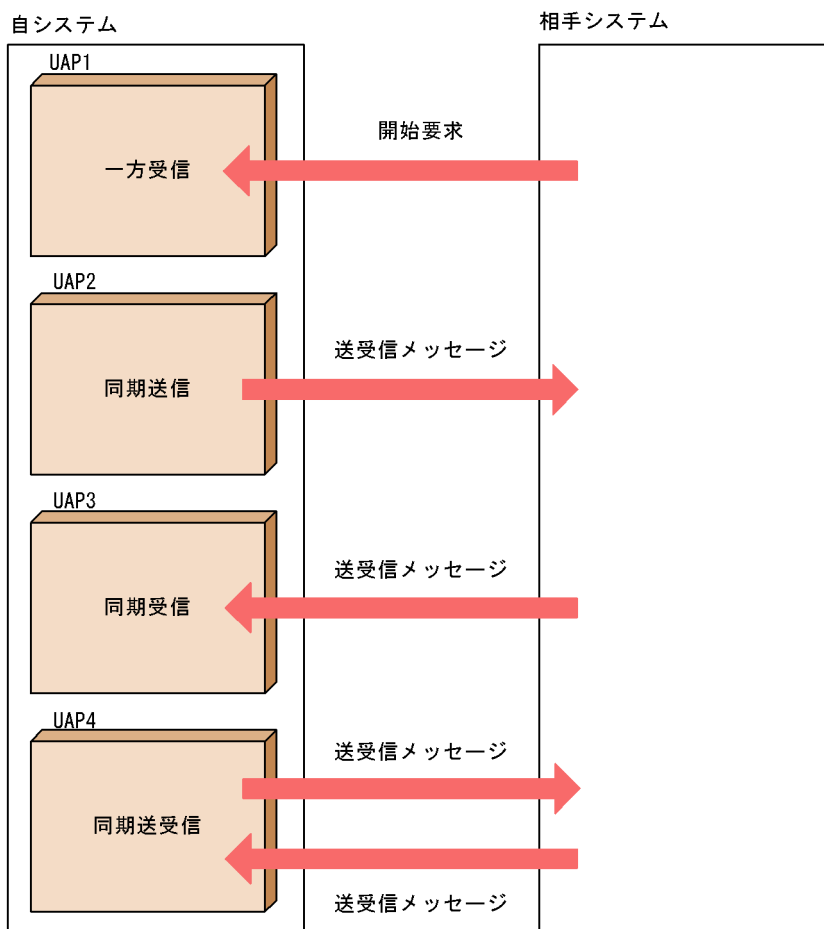
AP 間通信を使用すると、自システムで発生したトランザクションを相手システムで処理したり、その結果を受信したりできます。また、相手システムで発生したトランザクションを自システムで処理したり、その結果を送信したりできます。

TP1/NET/OSI-TP を使用した AP 間通信の形態には、次の四つがあります。

- 一方受信
- 同期送信
- 同期受信
- 同期送受信

TP1/NET/OSI-TP を使用した AP 間通信の例を次の図に示します。

図 1-3 TP1/NET/OSI-TP を使用した AP 間通信の例



(1) 一方受信

相手システムから AP 間通信の開始要求を受信する形態です。開始要求を受信すると、TP1/NET/OSI-TP はメッセージ送受信のためのアプリケーションを起動します。この形態は、AP 間通信を開始するときだけに適用されます。

(2) 同期送信

相手システムに対して、メッセージを送信する形態です。このとき、相手システムからの応答は要求しません。メッセージ送信処理が完了すると、送信を要求した UAP にリターンします。

(3) 同期受信

相手システムから、メッセージを受信する形態です。このとき、自システムから応答は返しません。メッセージを受信すると、受信を要求した UAP にリターンします。

1. 概要

(4) 同期送受信

自システムからメッセージを送信し、相手システムからの応答を受信する形態です。このとき、自システムからメッセージを送信後、相手システムからの応答を待ちます。相手システムからのメッセージを受信したときに、同期送受信を要求した UAP にリターンします。

1.2.2 適用範囲

TP1/NET/OSI-TP では、次に示す四つの機能を実装しています。

- カーネル機能
- 全二重機能
- 半二重機能
- ハンドシェイク機能

カーネル機能と全二重機能または半二重機能は OSI TP プロトコルの基本機能です。ダイアログを開始し、メッセージを送受信できるようにします。また、メッセージ送受信中のエラー通知や、ダイアログの正常終了または異常終了ができます。ハンドシェイクは、システム間でメッセージの同期合わせをするために使用する機能です。

機能の実装範囲の詳細については、「付録 A TP1/NET/OSI-TP の実装範囲」を参照してください。

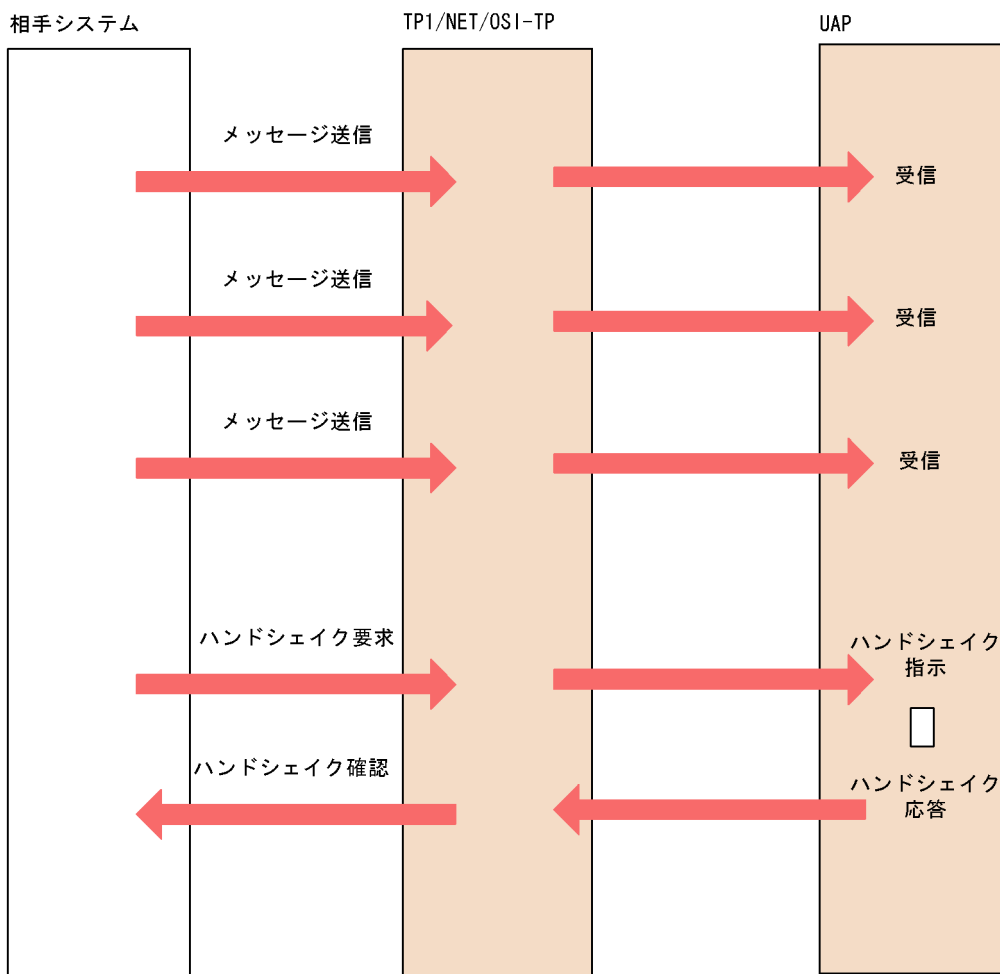
TP1/NET/OSI-TP は、相手システムとの間だけで行われる特別な処理や、メッセージの重複、欠落などのチェックはしません。したがって、AP 間通信をするときには、相手システムと自システムの UAP との間で、次に示す項目に関して事前に取り決めておく必要があります。

(1) メッセージの送達の確認

相手システムから一方的にメッセージを送信する場合、相手システムと自システムの UAP との間で連続して送信する個数や送信する回数を事前に決めておきます。また、連続して送信する場合は、送信したあとで、UAP は相手システムとの同期を取ることが必要です。これらの取り決めをすることで、資源不足による障害の発生を防ぎます。

OSI TP プロトコルのハンドシェイク機能を使用して、メッセージを相手システムが一方的に送信したあと、システム間の同期を取る例を次の図に示します。ハンドシェイクの詳細については、「2.2.6 ハンドシェイク」を参照してください。

図 1-4 メッセージの送達確認の例



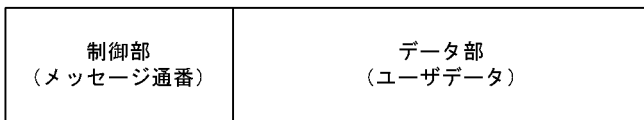
(2) メッセージの重複，および欠落のチェック

メッセージの重複，および欠落のチェックをする場合は，相手システムと自システムのUAPとの間でメッセージの形式を事前に決めておきます。例えば，メッセージの内部を制御部とデータ部分け，制御部にメッセージの通番を，データ部にはユーザの送信したいデータを格納しておきます。メッセージを受信する側が通番をチェックすることで，メッセージの重複，および欠落を検出できます。

メッセージの形式の例を次の図に示します。

1. 概要

図 1-5 メッセージの形式の例



(3) メッセージの応答の要否

OSI TP プロトコルでは、メッセージの応答が必要か不要かを選択できます。応答を不要とすると、通信するメッセージの回数を削減でき、性能が向上します。また、応答を必要とした場合は、メッセージが確実に送信されたかどうかを確認でき、信頼性が高まります。

メッセージの応答を不要とした場合の例を図 1-6 に、必要とした場合の例を図 1-7 に示します。

図 1-6 メッセージの応答を不要とした場合

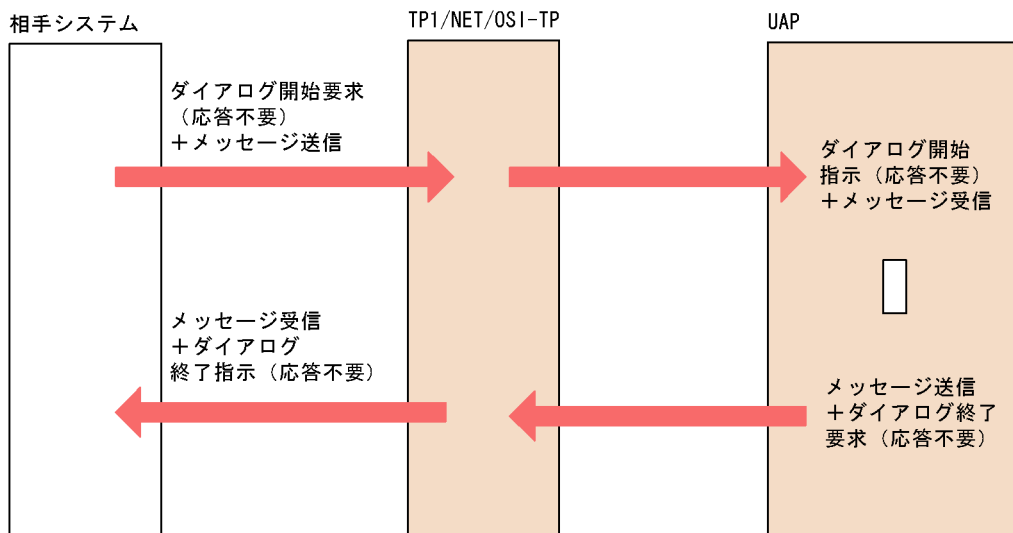
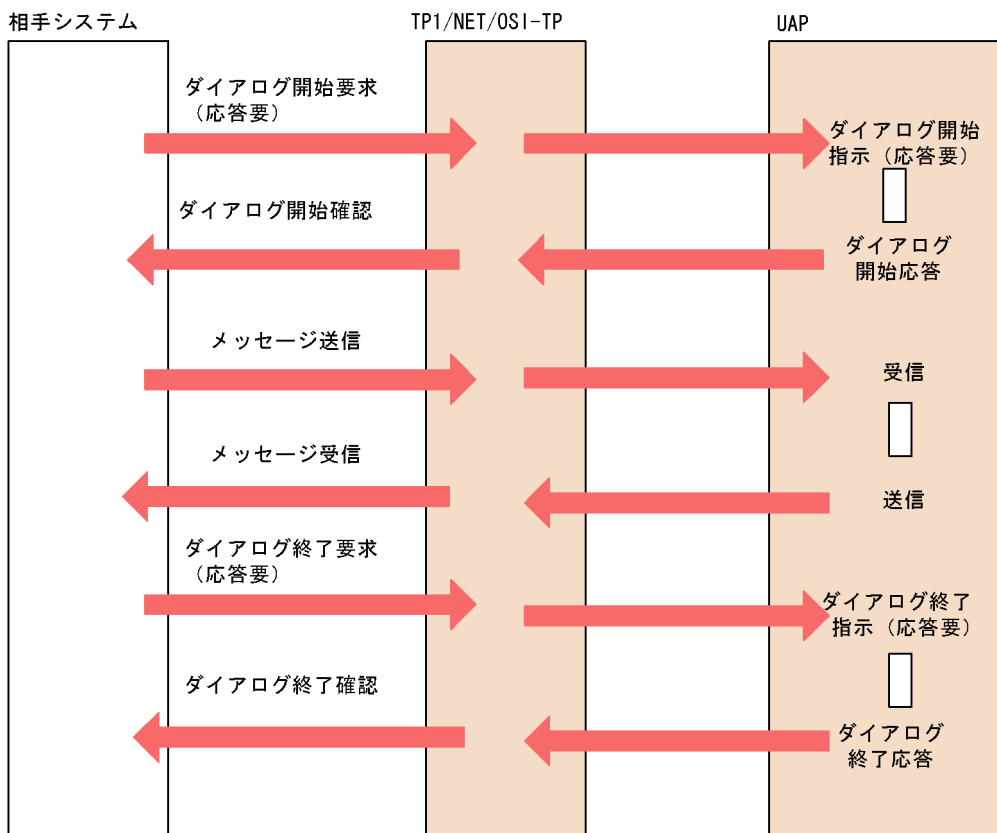


図 1-7 メッセージの応答を必要とした場合



また、ダイアログの開始要求の応答を不要とした場合、それ以後に送信したメッセージが破棄されることがあります。詳細については、「2.2.4 ダイアログ」を参照してください。

1.3 ソフトウェアの構成

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

適用 OS : AIX

- P-1M64-2131 uCosminexus TP1/Server Base 07-00 以降
- P-1M14-511 XNF/AS/BASE ¹
- P-F1M14-5111 XNF/AS/WAN ¹, P-F1M14-5112 XNF/AS/ACONARC ¹,
または P-F1M14-511D XNF/AS/OSI Extension ¹, ²

注 1

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。

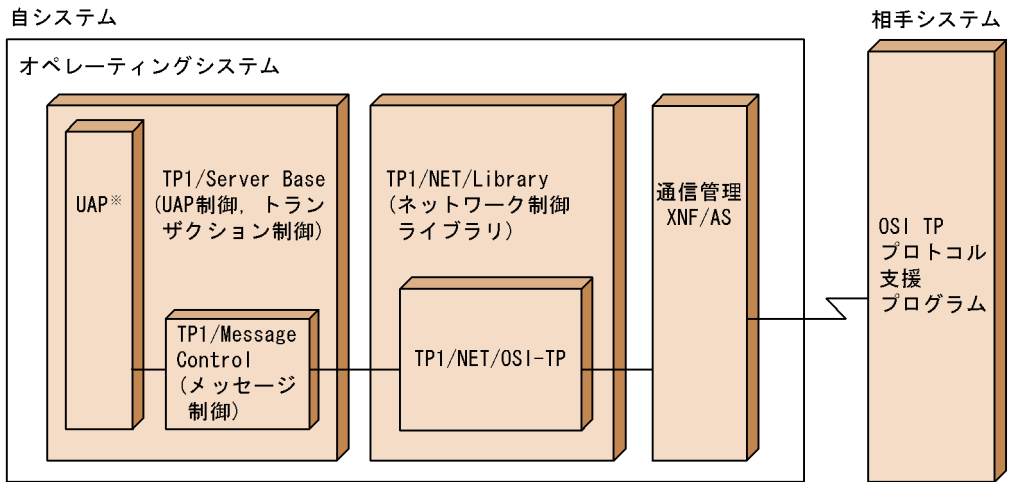
注 2

自局 IP アドレス指定機能を使用する場合は、XNF/AS/OSI Extension/Cluster が必要です。また、OSI 拡張高信頼化機能を使用する場合は、XNF/AS/Host Adapter が必要です。なお、どちらの場合も前提となるバージョンは、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/OSI-TP を組み込んだソフトウェア構成の例を次の図に示します。

図 1-8 TP1/NET/OSI-TP を組み込んだソフトウェア構成の例



注

TP1/NET/OSI-TP 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。

また、TP1/NET/OSI-TP 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次の表に示します。

表 1-1 TP1/NET/OSI-TP に適用する AP 間通信のプロトコル

プロトコル	通信相手
OSI TP プロトコル	VOS3 OSAS/TP/DCCM3

2

機能

一般に、AP 間通信をするときには、自システムと相手システムとの間であらかじめ通信上の規約（プロトコル）を決める必要があります。TP1/NET/OSI-TP は、二つのシステムの間にはコネクションという論理的通信路を設定し、メッセージを送受信します。この章では、TP1/NET/OSI-TP が設定するコネクションの確立方法、メッセージの種類と送受信の方法などについて説明します。

2.1 AP 間通信の仕組み

2.2 AP 間通信メッセージの送受信

2.1 AP 間通信の仕組み

TP1/NET/OSI-TP は、TP プロトコルデータ単位を最小の単位として、メッセージの送受信をします。

TP1/NET/OSI-TP で送受信する TP プロトコルデータ単位について、次の表に示します。

表 2-1 TP1/NET/OSI-TP の TP プロトコルデータ単位

TP プロトコルデータ単位	機能	備考
TP-BEGIN-DIALOGUE	ダイアログ開始	-
TP-END-DIALOGUE	ダイアログ終了	-
TP-U-ABORT	ダイアログの異常終了	U：ユーザでの異常終了
TP-P-ABORT		P：TP1/NET/OSI-TP での異常終了
TP-HANDSHAKE	ハンドシェイク	-
TP-DATA	データ転送	-
TP-U-ERROR	エラー通知	ダイアログは解放されません。
TP-GRANT-CONTROL	制御権委譲	-
TP-REQUEST-CONTROL	制御権要求	制御権は移動しません。
TP-HANDSHAKE-AND-GRANT-CONTROL	制御権委譲付き ハンドシェイク	-

(凡例)

- : 該当しません。

2.1.1 コネクションの確立と解放

TP1/NET/OSI-TP では、相手システムとの間に論理的通信路 (コネクション) を確立してメッセージを送受信します。コネクションは、OSI TP プロトコルのアソシエーションに対応します。

(1) コネクションの確立

コネクションの確立方法には、次の二つがあります。

- 発呼型
- 着呼型

どちらの確立方法をとるかは、MCF 通信構成定義 (mcftalccn -t) で指定します。

(a) 発呼型

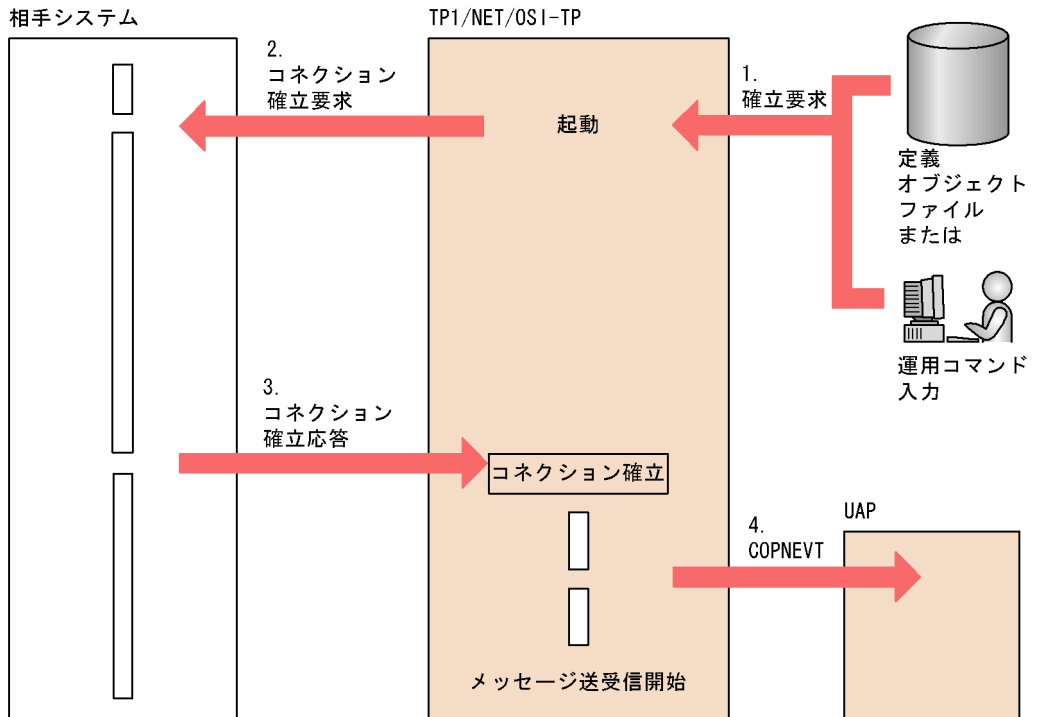
自システムからコネクションの確立を要求する方法です。

OpenTP1 システムの開始・再開時に指定した MCF 通信構成定義 (mcftalccn -i auto)

の内容、または運用コマンド (mcftactcn) の入力によって、TP1/NET/OSI-TP は相手システムに接続の確立を要求します。相手システムからの接続確立応答を受信すると、接続が確立し、メッセージの送受信ができる状態になります。

発呼型の確立方法を、次の図に示します。

図 2-1 コネクションの確立 (発呼型)



1. OpenTP1 システムが開始・再開始、または運用コマンド (mcftactcn) を入力します。
2. TP1/NET/OSI-TP は、相手システムに接続確立要求を送信します。
3. 相手システムから接続確立応答を受信すると、接続が確立します。
4. TP1/NET/OSI-TP は、接続が確立すると状態通知イベント (COPNEVT) を発生させます。

(b) 着呼型

相手システムからの接続の確立要求を受け付ける方法です。

定義オブジェクトファイルの内容によって TP1/NET/OSI-TP を起動し、相手システムからの接続確立要求を待ちます。確立要求を受けると、TP1/NET/OSI-TP は相手システムに接続確立応答を返します。応答を返した時点で接続が確立し、メッセージの送受信ができる状態になります。

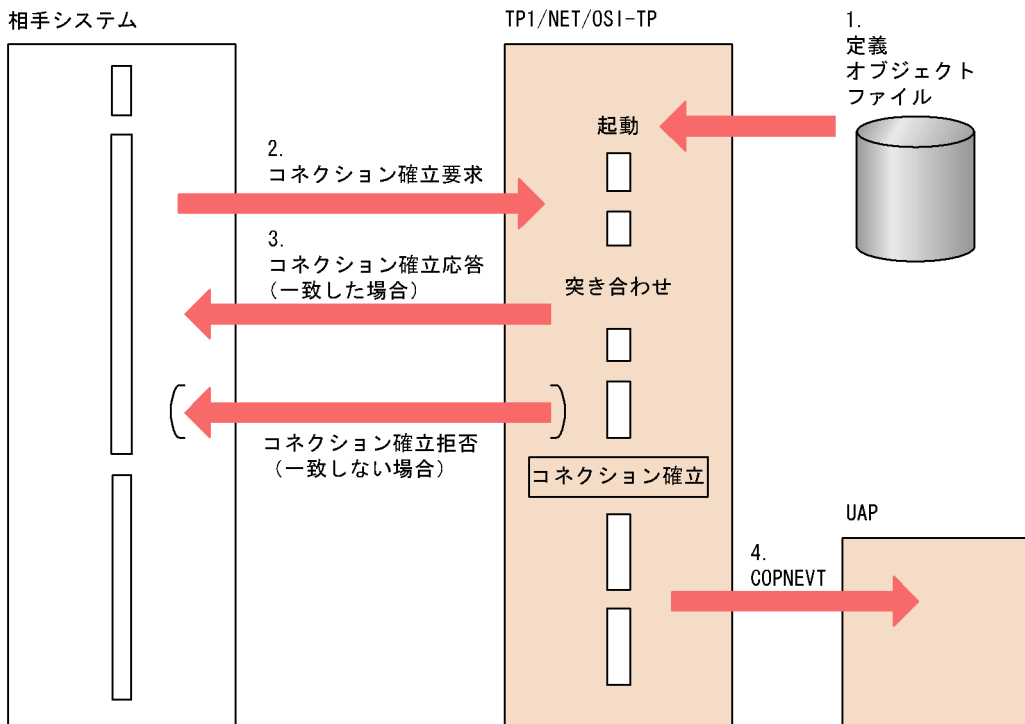
着呼型の接続の場合、TP1/NET/OSI-TP では、接続が解放されたあ

2. 機能

と、自動的に相手システムからのコネクション確立要求待ちになります。

着呼型の確立方法を、次の図に示します。

図 2-2 コネクションの確立（着呼型）



1. OpenTP1 システムの開始・再開始時、定義オブジェクトファイルの内容によって、相手システムからのコネクション確立要求を待ちます。
2. 相手システムからコネクション確立要求を受信します。
3. TP1/NET/OSI-TP は、MCF 通信構成定義情報に基づいて、相手システムからのコネクション確立要求の突き合わせ をします。
構成が一致した場合、コネクション確立応答を相手システムに送信します。一致しない場合は、相手システムに対してコネクション確立拒否の応答を送信します。
4. TP1/NET/OSI-TP は、コネクションが確立すると状態通知イベント（COPNEVT）を発生させます。

注

定義の突き合わせ項目については、「3. メッセージ送受信インタフェース」の「サービスプリミティブ」を参照してください。

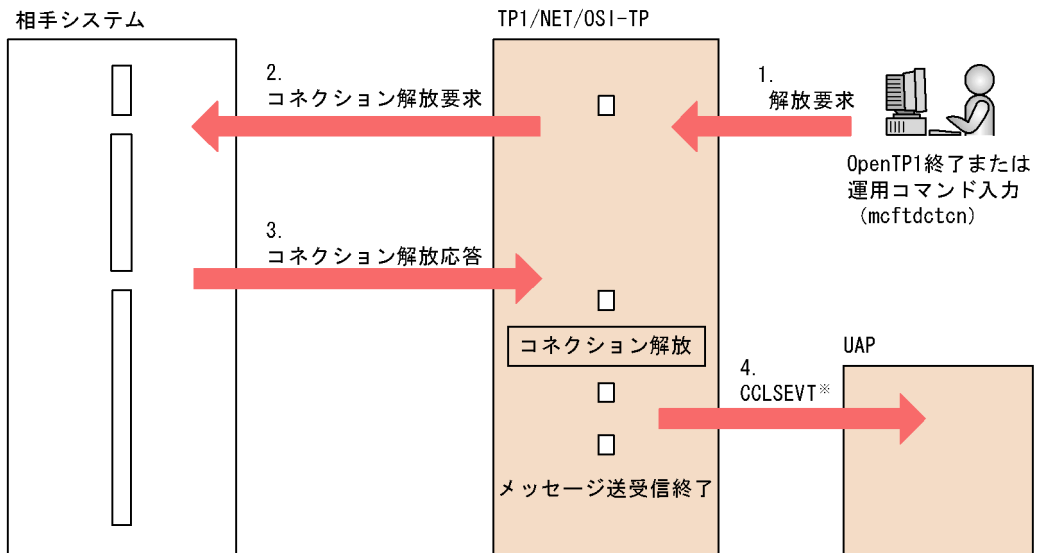
(2) コネクションの正常解放

コネクションの正常解放には、次の三つがあります。

- OpenTP1 システム終了時の解放
- 運用コマンド (mcftdctn) 入力による解放
- 相手システムからの解放要求による解放

TP1/NET/OSI-TP からの接続の解放を図 2-3 に、相手システムからの接続の解放を図 2-4 に示します。

図 2-3 TP1/NET/OSI-TP からの接続の解放

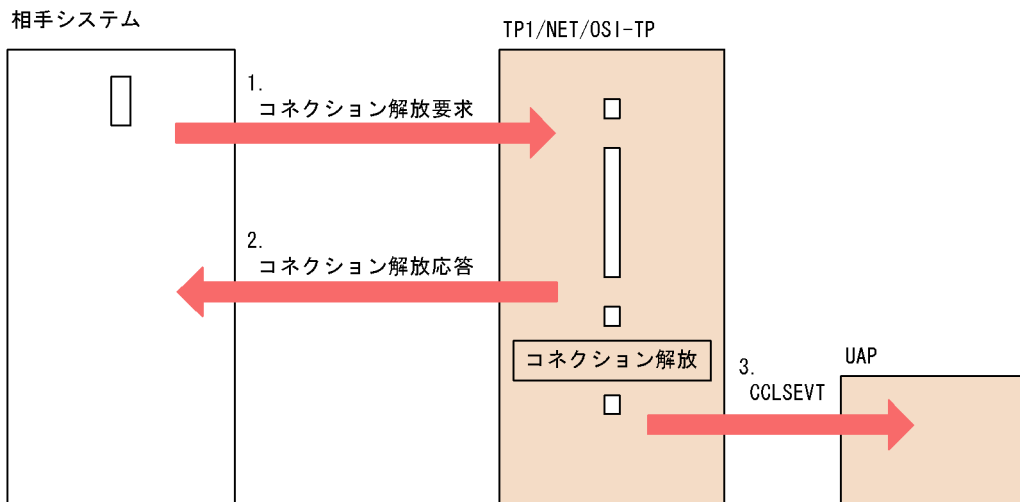


1. OpenTP1 システム終了, または運用コマンド (mcftdctn) を入力します。
2. TP1/NET/OSI-TP は, 相手システムに対してコネクション解放要求を送信します。
3. 相手システムから, コネクション解放応答を受信します。
4. TP1/NET/OSI-TP は, コネクション解放応答を受信すると, 状態通知イベント (CCLSEVT) を発生させます。

注

状態通知イベントは, OpenTP1 システム終了時は発生させません。

図 2-4 相手システムからの接続の解放



1. 相手システムからコネクション解放要求を受信します。
2. TP1/NET/OSI-TP は、相手システムにコネクション解放応答を送信します。
3. コネクション解放処理が完了すると、TP1/NET/OSI-TP は、状態通知イベント (CCLSEVT) を発生させます。

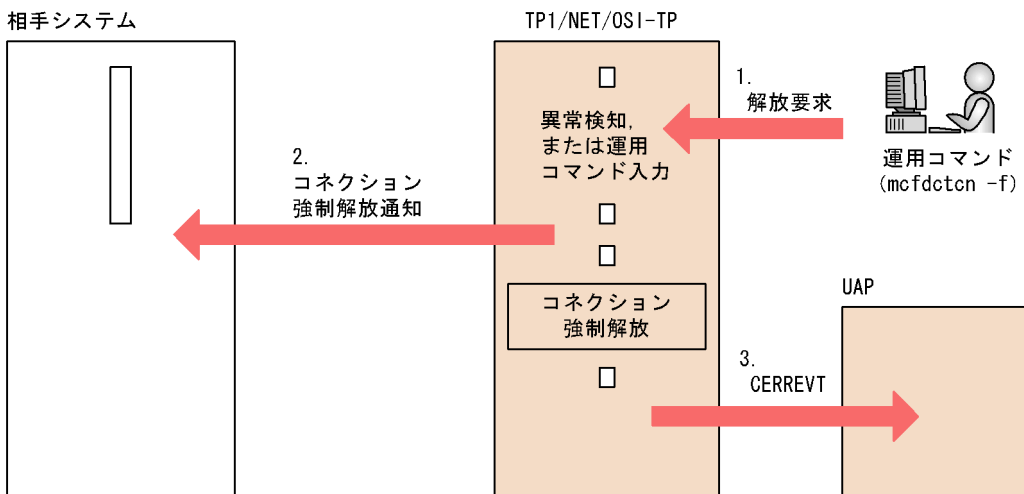
(3) コネクションの強制解放

コネクションの強制解放には、次の三つがあります。

- TP1/NET/OSI-TP で異常を検知した場合の解放
- コネクションを強制解放する運用コマンド (`mcftdeten -f`) 入力による解放
- 相手システムに何らかの異常が発生した場合の解放

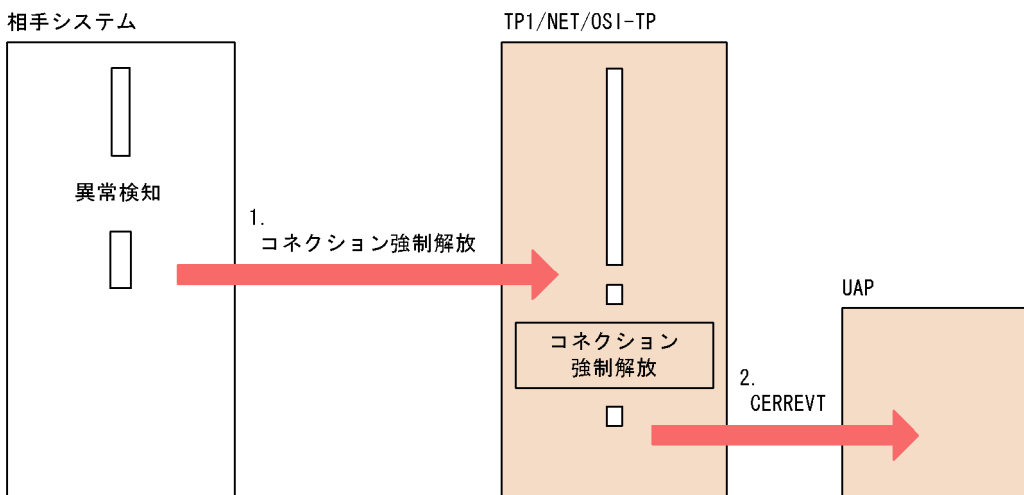
TP1/NET/OSI-TP からのコネクションの強制解放を図 2-5 に示します。また、相手システムの異常による解放を図 2-6 に示します。

図 2-5 TP1/NET/OSI-TP からの接続の強制解放



1. コネクションを強制解放する運用コマンド (mcfdtctn -f) を受け付けた場合、または TP1/NET/OSI-TP 内部で異常が発生した場合、TP1/NET/OSI-TP はコネクションを TP1/NET/OSI-TP からのコネクションの強制解放します。
2. TP1/NET/OSI-TP は、相手システムにコネクションの強制解放を通知します。
3. コネクションの強制解放の処理が完了すると、TP1/NET/OSI-TP は、状態通知イベント (CERREVT) を発生させます。

図 2-6 相手システムの異常による解放



1. 相手システムでの異常、および通信回線の障害によって、TP1/NET/OSI-TP はコネクションの強制解放を受信します。

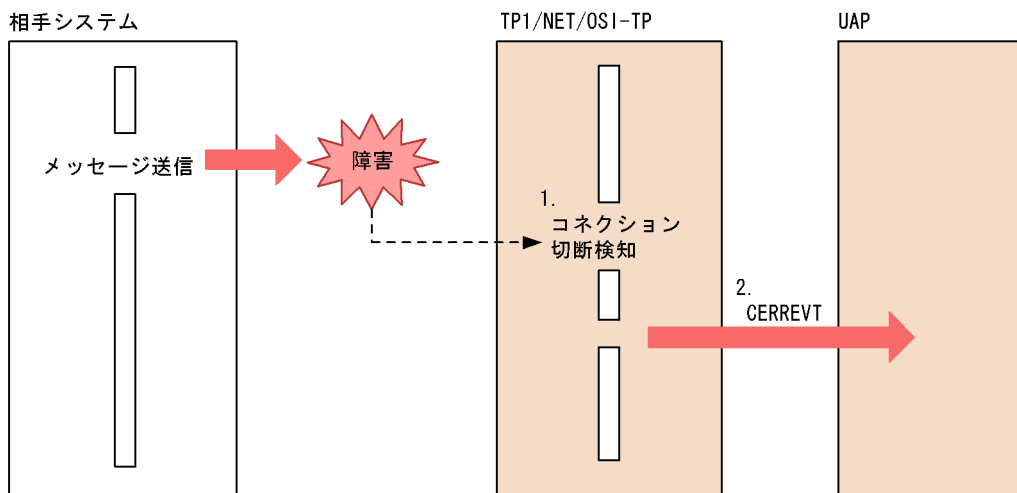
2. 機能

2. コネクションの強制解放の処理が完了すると、TP1/NET/OSI-TP は、状態通知イベント (CERREVT) を発生させます。

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/OSI-TP 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 2-7 コネクションの切断



1. メッセージ送受信時に障害が発生すると、TP1/NET/OSI-TP はコネクションの切断を検知します。
2. TP1/NET/OSI-TP は、状態通知イベント (CERREVT) を発生させます。

2.1.2 コネクションと論理端末の関係

TP1/NET/OSI-TP は、論理端末を通して、自システムの UAP とメッセージを送受信します。この論理端末は、TP1/NET/OSI-TP と UAP との通信接点に当たります。これに対してコネクションとは、TP1/NET/OSI-TP が通信管理プログラムを介して、相手システムの UAP とメッセージを送受信する際に確立するものです。

コネクションと論理端末の指定を対応させると、自システムと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、システム定義で対応させます。

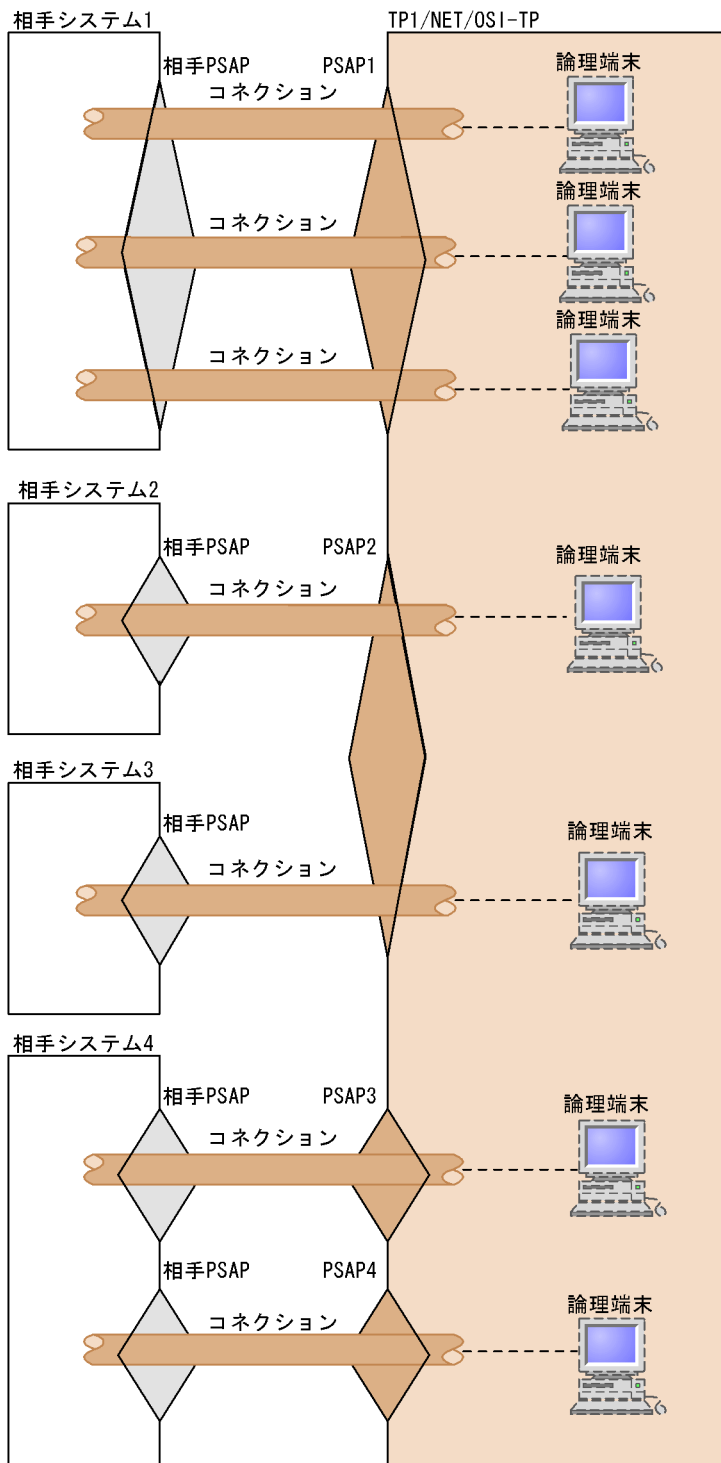
2.1.3 コネクションと PSAP の関係

PSAP とは、OSI TP プロトコルを使用してメッセージを送受信するときに、相手システムとの窓口となるアクセス点です。TP1/NET/OSI-TP では、自システムに一つ以上の PSAP を持つことができます。また、相手システムの PSAP との間に、複数のコネクションを設定できます。

コネクションと PSAP の関係を次の図に示します。

2. 機能

図 2-8 コネクションと PSAP の関係



一つの PSAP には、任意の数のコネクションを設定できます。ただし、定義で指定したコネクション数以上のコネクションは確立できません。また、一つの PSAP の中では、発呼型のコネクションと着呼型のコネクションは混在できません。

自分の PSAP と相手の PSAP が同じコネクションを複数まとめてコネクショングループとして定義できます。図 2-8 で自システムの PSAP1 と相手システム 1 の相手 PSAP との間にある三つのコネクションを使って 3 端末から発生する業務をマルチサーバ機能で処理する場合を例に挙げます。この場合は、三つのコネクションを一つのコネクショングループとして定義することで、UAP での処理のプロセス数を 3 として、3 端末から発生する業務を同じ UAP で処理できます。詳細は、「2.1.9 アソシエーションプール管理機能」を参照してください。

2.1.4 経路交代

コネクション確立要求時にネットワーク層を含む下位の層で障害が発生した場合、経路を切り替えてコネクションを再確立する機能を経路交代と呼びます。

経路交代を使用するには MCF 通信構成定義 (mcfaltccn -z) で現用経路のための現用相手 DTE アドレスと現用スロット番号 (1 対) に加えて、経路交代用の交代用相手 DTE アドレスと交代用スロット番号 (最大 7 対) を指定します。

2.1.5 論理端末とアプリケーションの型の関係

TP1/NET/OSI-TP 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ことで、TP1/NET/OSI-TP 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に指定したアプリケーション名をキーとして、一つの UAP (MHP) プロセスで実行されます。アプリケーションは、サービスの方式によって型が異なります。この型を、TP1/Message Control のアプリケーション属性の一つとして、システム定義で指定します。

TP1/NET/OSI-TP で指定するアプリケーションの型は、非応答型 (noans) です。

論理端末の端末タイプ、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 2-2 論理端末の端末タイプ、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

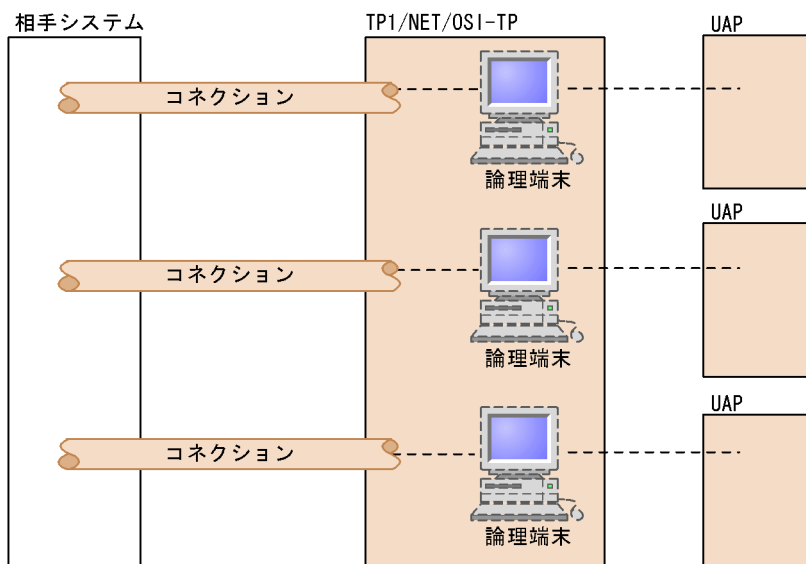
論理端末の端末タイプ	メッセージ	アプリケーションの型	UAP インタフェース	通信形態
any (任意型論理端末)	一方送信メッセージ	非応答型 (noans)	receive	一方受信
			sendsync	同期送信
			recvsync	同期受信

論理端末の端末タイプ	メッセージ	アプリケーションの型	UAP インタフェース	通信形態
			sendrecv	同期送受信

2.1.6 論理端末と UAP の関係

TP1/NET/OSI-TP で扱う論理端末と UAP プロセスは、必ず 1 対 1 で対応するようにしてください。つまり、複数の UAP プロセスを動作させる場合は、UAP プロセスと同数の論理端末とコネクションを定義する必要があります。論理端末（コネクション）と UAP の関係を次の図に示します。

図 2-9 論理端末（コネクション）と UAP の関係



ただし、アソシエーションプール管理機能を使用する場合は、論理端末と UAP を 1 対 1 で対応づける必要はありません。アソシエーションプール管理機能の詳細については、「2.1.9 アソシエーションプール管理機能」を参照してください。

2.1.7 全二重と半二重

全二重と半二重の伝送方式があります。

全二重は同時に送信動作と受信動作を互いに独立して行うことができます。

半二重は同時に送信動作または受信動作だけを行います。このとき伝送する双方のどちらか一方に制御権を与え、制御権の所有者だけがデータを送信できます。

全二重を使用するかまたは半二重を使用するかは MCF 通信構成定義 (mcfaltcn -d

control) で指定します。

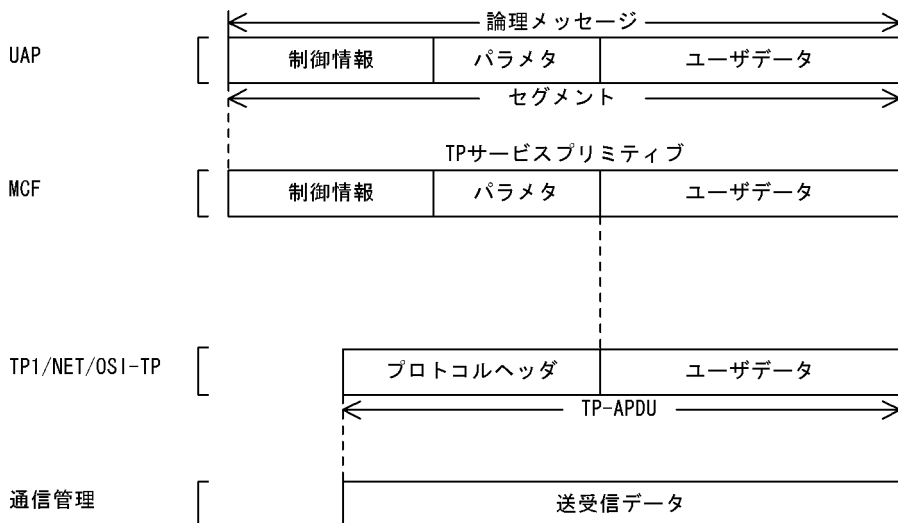
2.1.8 メッセージの分割と組み立て

業務上で意味があるメッセージの一つの単位を論理メッセージといい、メッセージ送受信の関数で扱う情報の単位をセグメントといいます。TP1/NET/OSI-TP の論理メッセージは、一つのセグメントで構成され、付けられる情報やデータの内容は、処理層によって異なります。また、ユーザが作成した固有のデータは、データの内容によって形態が異なります。

送信の場合、OpenTP1 システムの UAP で作成した論理メッセージは、MCF、TP1/NET/OSI-TP、通信管理の各処理層を介して相手システムに送信されます。相手システムからメッセージを受信する場合は逆の順番になります。

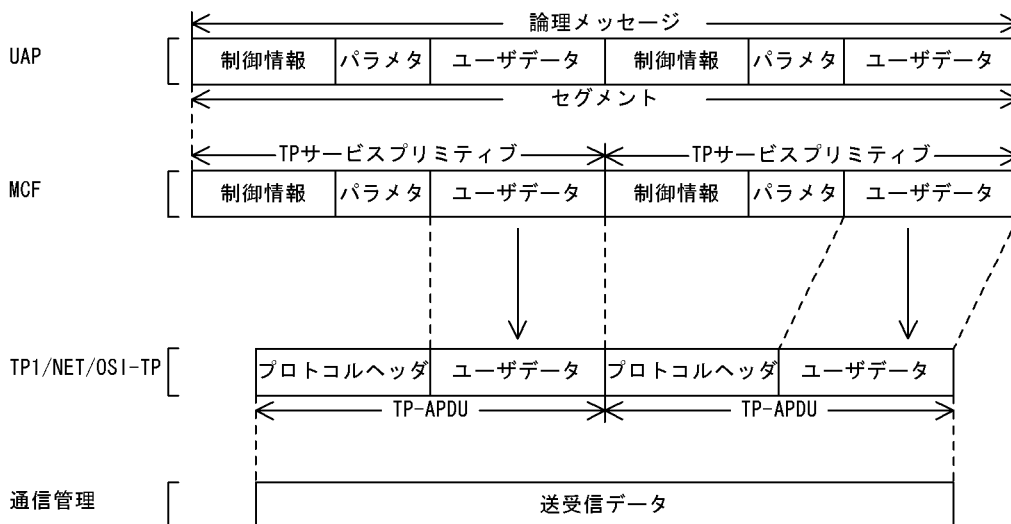
各処理層とデータの関係について、単一 APDU の場合を図 2-10 に、連結 APDU の場合を図 2-11 に示します。

図 2-10 処理層とデータの関係（単一 APDU の場合）



2. 機能

図 2-11 処理層とデータの関係（連結 APDU の場合）



注

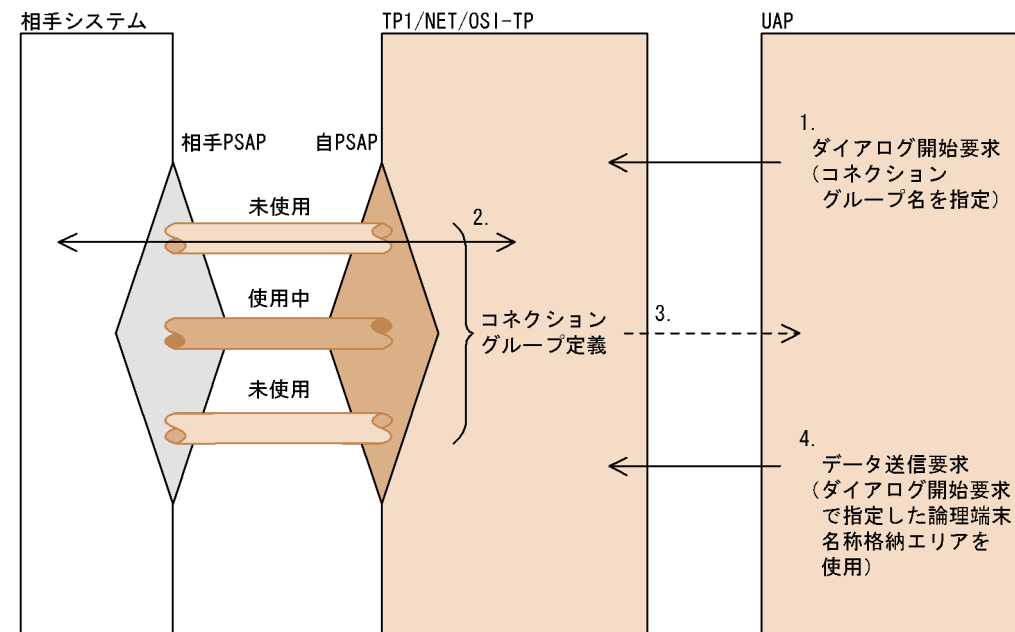
この場合は、2 個連結の例です。

メッセージ送受信の関数で処理するセグメントの先頭には、MCF で処理するヘッダ領域があります。C 言語と COBOL 言語の場合、このヘッダ領域の長さによって、バッファにはバッファ形式 1 とバッファ形式 2 があります。通常バッファ形式 1 を使用します。データ操作言語の場合、バッファ形式は一つだけです。

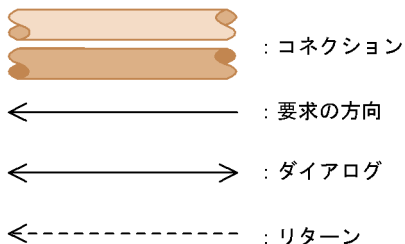
2.1.9 アソシエーションプール管理機能

UAP がダイアログ開始要求を送信する場合、通信構成定義 (mcftgrpcn -g) で指定したコネクショングループ名を指定すれば、TP1/NET/OSI-TP がコネクショングループ内の使用できるコネクションを自動的に選択し、相手システムとの間でダイアログを開始します。この場合、同じコネクショングループのコネクションでは、コネクション ID 以外の属性を同じに定義する必要があります。アソシエーションプール管理機能の概要を次の図に示します。

図 2-12 アソシエーションプール管理機能の概要



(凡例)



1. UAP はダイアログ開始要求を送信するとき、論理端末名称にコネクショングループ名を指定します。
2. TP1/NET/OSI-TP は、UAP が指定したコネクショングループ配下の未使用のコネクションを選択してダイアログを開始します。
3. ダイアログ開始要求が正常にリターンした場合、TP1/NET/OSI-TP は UAP がダイアログ開始要求で指定したコネクショングループ名を、ダイアログ開始時に選択したコネクションに対応した論理端末名称に書き換えます。ダイアログ開始要求が異常でリターンした場合は、論理端末名称への書き換えはしません。
4. ダイアログ開始要求が正常にリターンしたあと、UAP がデータ送信などを呼び出す場合、ダイアログ開始要求時に指定した論理端末名称格納エリアを使用します。

2.2 AP 間通信メッセージの送受信

この節では、TP1/NET/OSI-TP で使用するメッセージの形態、サービスの種類、および APDU 連結について説明します。

TP1/NET/OSI-TP で使用するメッセージの形態は、次の二つに分けられます。

- 非同期型的一方送信メッセージの受信
- 同期型的一方送信メッセージの送信、受信、および送受信

また、TP1/NET/OSI-TP が提供するサービスの種類を次の表に示します。

表 2-3 TP1/NET/OSI-TP が提供するサービスの種類

サービスの種類	参照先
制御権委譲 (TP-GRANT-CONTROL)	2.2.3 制御権
制御権要求 (TP-REQUEST-CONTROL)	
ダイアログ開始 (TP-BEGIN-DIALOGUE)	2.2.4 ダイアログ
ダイアログ終了 (TP-END-DIALOGUE)	
ダイアログの異常終了 (TP-U-ABORT, TP-P-ABORT)	
ハンドシェイク (TP-HANDSHAKE)	2.2.6 ハンドシェイク
制御権委譲付きハンドシェイク (TP-HANDSHAKE-AND-GRANT-CONTROL)	
データ転送 (TP-DATA)	2.2.7 データ転送
エラー通知 (TP-U-ERROR)	2.2.8 エラー通知

さらに、TP1/NET/OSI-TP では、複数のサービスを一つにまとめて送受信できる APDU 連結機能を提供しています。

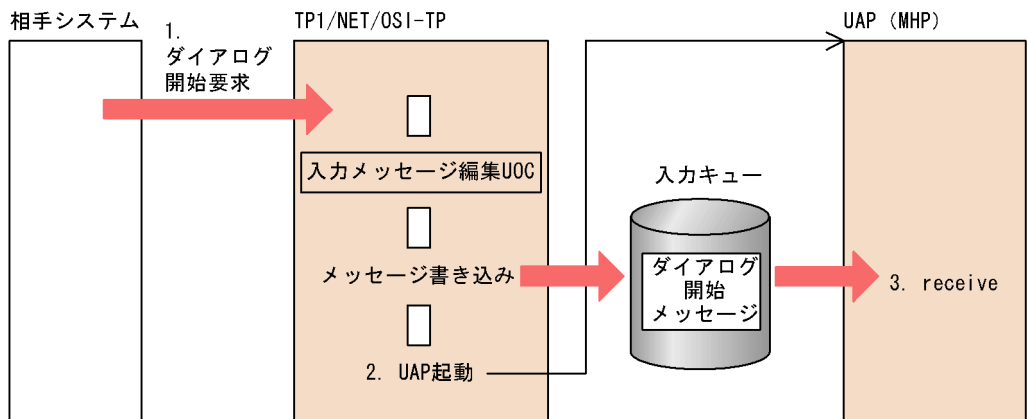
2.2.1 非同期型のメッセージの受信

非同期型とは、システム内で同期を取らない形態です。UAP が送受信要求を出したあと、各要求の処理が完了するのを待たないで、それ以降の処理を続けます。

TP1/NET/OSI-TP では、非同期型のメッセージは受信だけに使用できます。相手システムからダイアログ開始要求を受信し、該当する UAP を起動します。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です。このとき、起動できる UAP は MHP だけです。

TP1/NET/OSI-TP が相手システムからダイアログ開始要求を受信し、メッセージを UAP に渡すまでの処理を、次の図に示します。

図 2-13 非同期型のメッセージの受信



1. 相手システムから，any 型論理端末あてのダイアログ開始要求（TP-BEGIN-DIALOGUEind）を受信します。
2. TP1/NET/OSI-TP は，相手システムからのダイアログ開始要求を入力キューに書き込み，UAP（MHP）を起動します。
3. UAP（MHP）では，receive 関数を発行して，メッセージを受け取ります。

以降，このマニュアルでは，非同期のメッセージの受信を，特にメッセージの受信と呼びます。メッセージの受信には，RECEIVE 文または receive 関数を使用します。

2.2.2 同期型のメッセージの送受信

同期型とは，自システムの UAP が，送信または受信を要求し，各要求の処理の準備がすべて完了してからリターンする形態をいいます。同期型のメッセージには，送信，および受信をそれぞれ単独とするものと，送受信を連続してするものがあります。使用できるのは，sendsync 関数，recvsync 関数，および sendrecv 関数です。

(1) 同期型のメッセージの送信

自システムからメッセージを送信する形態です。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です。

UAP は，SENDSYNC 文または sendsync 関数で TP1/NET/OSI-TP にメッセージの送信要求をします。TP1/NET/OSI-TP はメッセージを送信用バッファに格納し，相手システムに送信します。メッセージの送信が完了したあと，送信を要求した UAP にリターンします。このとき，メッセージの送信の完了を受信するまで，TP1/NET/OSI-TP 内で待ち続けます。なお，無制限に待つことを避けるため，待ち時間を指定できます。

(2) 同期型のメッセージの受信

相手システムからのメッセージを受信する形態です。論理端末の端末タイプは any，ア

アプリケーションの型は非応答型です。

相手システムからのメッセージを受信すると、TP1/NET/OSI-TP はメッセージを受信用バッファに格納します。UAP は、RECVSYNC 文または `recvsync` 関数で TP1/NET/OSI-TP にメッセージの受信を要求します。相手システムからのメッセージを受信していた場合、メッセージの受信を要求した UAP に渡します。

また、相手システムからのメッセージを受信していない場合、相手システムからのメッセージを受信するまで、TP1/NET/OSI-TP 内で待ち続けます。この場合、無制限に待つことを避けるため、待ち時間を指定できます。相手システムからのメッセージを受信した時点で、メッセージの受信を要求した UAP に渡します。

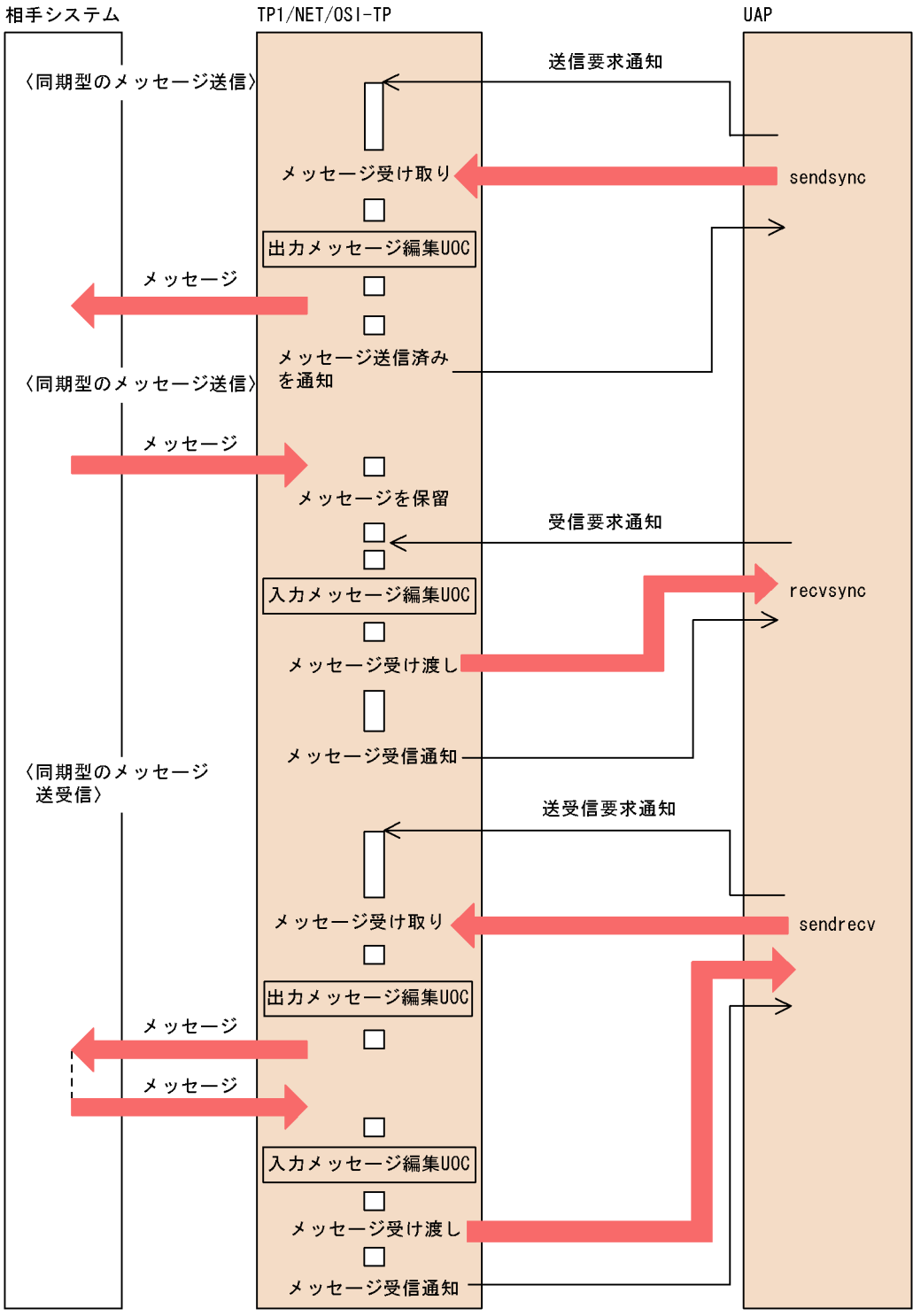
(3) 同期型のメッセージの送受信

メッセージの送信と受信とを連続して行う形態です。送信、および受信の要求を一度にできるため、処理を簡略化できます。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です。

UAP は、SENDRECV 文または `sendrecv` 関数で TP1/NET/OSI-TP にメッセージの送受信を要求します。TP1/NET/OSI-TP は、メッセージの送信処理をします。送信処理が完了したあとも、UAP にリターンしないで、引き続き受信処理をします。受信までの処理がすべて完了した時点で、UAP にリターンします。このとき、メッセージの受信を完了するまで、TP1/NET/OSI-TP 内で待ち続けます。なお、無制限に待つことを避けるため、待ち時間を指定できます。

同期型メッセージの送信、受信、および送受信を次の図に示します。

図 2-14 同期型のメッセージの送信，受信，および送受信



(4) 同期型のメッセージの送受信に関する注意事項

相手システムの処理遅延などが原因で、メッセージの送受信が完了する前に待ち時間が満了すると、同期型のメッセージ送受信で使用する関数（sendsync 関数，recvsync 関数，および sendrecv 関数）がエラーリターンします。このとき UAP には、リターン値 DCMCFRTN_73005 またはステータスコード 73005 が返却されます。

この場合は、自システムと相手システムの接続の状態をリセットする必要があります。UAP から mcftdcten -f コマンドを入力して接続を強制解放してください。

2.2.3 制御権

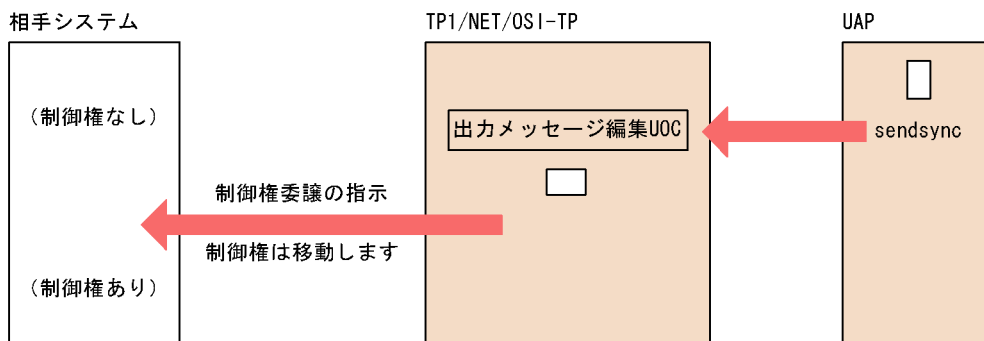
半二重機能を選択した場合、TP1/NET/OSI-TP は制御権によって、データの送信を制御します。制御権を持っている場合だけ、自システムからデータを送信要求できます。制御権は一つのダイアログの間だけに有効です。制御権の受け渡しについては、相手システムとの間で事前に取り決めておく必要があります。

(1) 制御権の委譲

制御権のあるシステムでは UAP から制御権委譲（TP-GRANT-CONTROL）の要求を受け付けると、TP1/NET/OSI-TP は相手システムに制御権委譲（TP-GRANT-CONTROL）の要求を送信します。

制御権のないシステムでは TP1/NET/OSI-TP が相手システムから制御権委譲（TP-GRANT-CONTROL）の要求を受信すると、recvsync で受信待ちになっている UAP に制御権委譲（TP-GRANT-CONTROL）の指示を通知します。このとき制御権は移動します。制御権委譲を次の図に示します。

図 2-15 制御権委譲

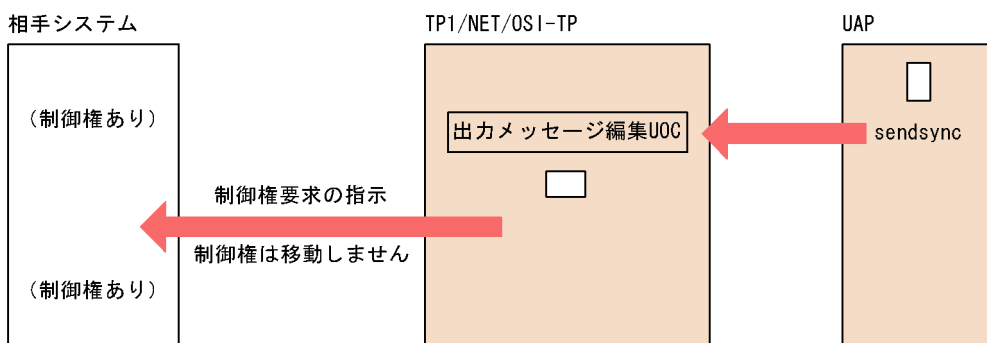


(2) 制御権の要求

制御権のないシステムでは UAP から制御権要求 (TP-REQUEST-CONTROL) の要求を受け付けると、TP1/NET/OSI-TP は相手システムに制御権要求 (TP-REQUEST-CONTROL) の要求を送信します。

制御権のあるシステムでは TP1/NET/OSI-TP が相手システムから制御権要求 (TP-REQUEST-CONTROL) の要求を受信すると、recvsync で受信待ちになっている UAP に制御権要求 (TP-REQUEST-CONTROL) の指示を通知します。しかし、そのとき制御権は移動しません。制御権を移動したい場合、制御権を委譲する必要があります。制御権要求を次の図に示します。

図 2-16 制御権要求



また、制御権を移動したくない場合、そのままデータ送信を続けることができます。制御権のない側は制御権要求の要求を出すと制御権を委譲されるとは限りません。制御権のある側からの制御権委譲を待つ必要があります。

制御権の非所有者側が制御権を委譲される契機、制御権の所有者側が制御権を委譲する契機を次の表に示します。

表 2-4 制御権が移動する契機

移動の区別	移動の契機
非所有者が委譲される	ダイアログ開始 (TP-BEGIN-DIALOGUE) 要求の送信
	エラー通知 (TP-U-ERROR) 要求の送信
	制御権委譲付きハンドシェイク要求 (TP-HANDSHAKE-AND-GRANT-CONTROL) 応答の送信
	制御権委譲 (TP-GRANT-CONTROL) 指示の受信
所有者が委譲する	ダイアログ開始 (TP-BEGIN-DIALOGUE) 指示の受信
	エラー通知 (TP-U-ERROR) 指示の受信

移動の区別	移動の契機
	制御権委譲付きハンドシェイク要求 (TP-HANDSHAKE-AND-GRANT-CONTROL) 確認の受信
	制御権委譲 (TP-GRANT-CONTROL) 要求の送信

2.2.4 ダイアログ

ダイアログとは、TP1/NET/OSI-TP が OSI TP プロトコルに準拠して提供する、メッセージ送受信のサービスです。ダイアログ開始要求 (TP-BEGIN-DIALOGUE) をすることで、メッセージの送受信ができるようになります。

(1) ダイアログの開始

ダイアログ開始には、確認型と拒否応答型があります。確認型では、ダイアログ開始要求をしたあと、相手システムから、受諾または拒否応答を必ず受信します。拒否応答型では、ダイアログの開始を受け入れない場合に、相手システムに対して拒否応答を返します。これらの型は、ダイアログ開始要求時に、確認項目 (Confirmation) で指定します。

ダイアログ開始要求をする場合には、次の三つの条件を満たす必要があります。

- コネクションが確立済みであること
- ほかのダイアログが実行されていないこと
- コンテンションの勝者であること

コンテンションとは、ダイアログ開始時に、両方のシステムから開始要求が出される状態をいいます。TP1/NET/OSI-TP では、コンテンションの勝者になるか、敗者になるかをシステム定義で指定します。勝者を指定した場合だけ、ダイアログ開始要求を出せません。

相手システムからダイアログ開始要求を受けると、TP1/NET/OSI-TP はメッセージを編集し、UAP を起動します。UAP は、編集されたメッセージを RECEIVE 文または receive 関数で受け取ったあと、応答を送信します。確認型の場合は、ダイアログ開始応答を TP1/NET/OSI-TP に送信します。拒否応答型の場合は、ダイアログ開始要求を受け入れない場合だけ、TP1/NET/OSI-TP は、拒否応答のメッセージを編集し、相手システムに返します。

UAP がコネクショングループ名を指定してダイアログ開始要求をすると、TP1/NET/OSI-TP はコネクショングループ内の使用できるコネクションを自動的に選択して、ダイアログを開始します。

(2) ダイアログの終了

ダイアログを終了するときは、ダイアログ終了要求を出します。このとき、メッセージの送受信は終了させておきます。半二重の場合、ダイアログ終了要求は制御権のある側だけが出すことができます。

ダイアログ終了には、確認型と非確認型があります。確認型では、相手システムから終了の応答を必ず受信し、非確認型では、自システムは一方的にダイアログを終了します。型の指定は、ダイアログ終了要求時に確認項目（Confirmation）で行います。

ダイアログ終了時、UAP は SENDSYNC 文または sendsync 関数でダイアログ終了要求を TP1/NET/OSI-TP に送信します。確認型の場合は、相手システムからのダイアログ終了応答を受信します。

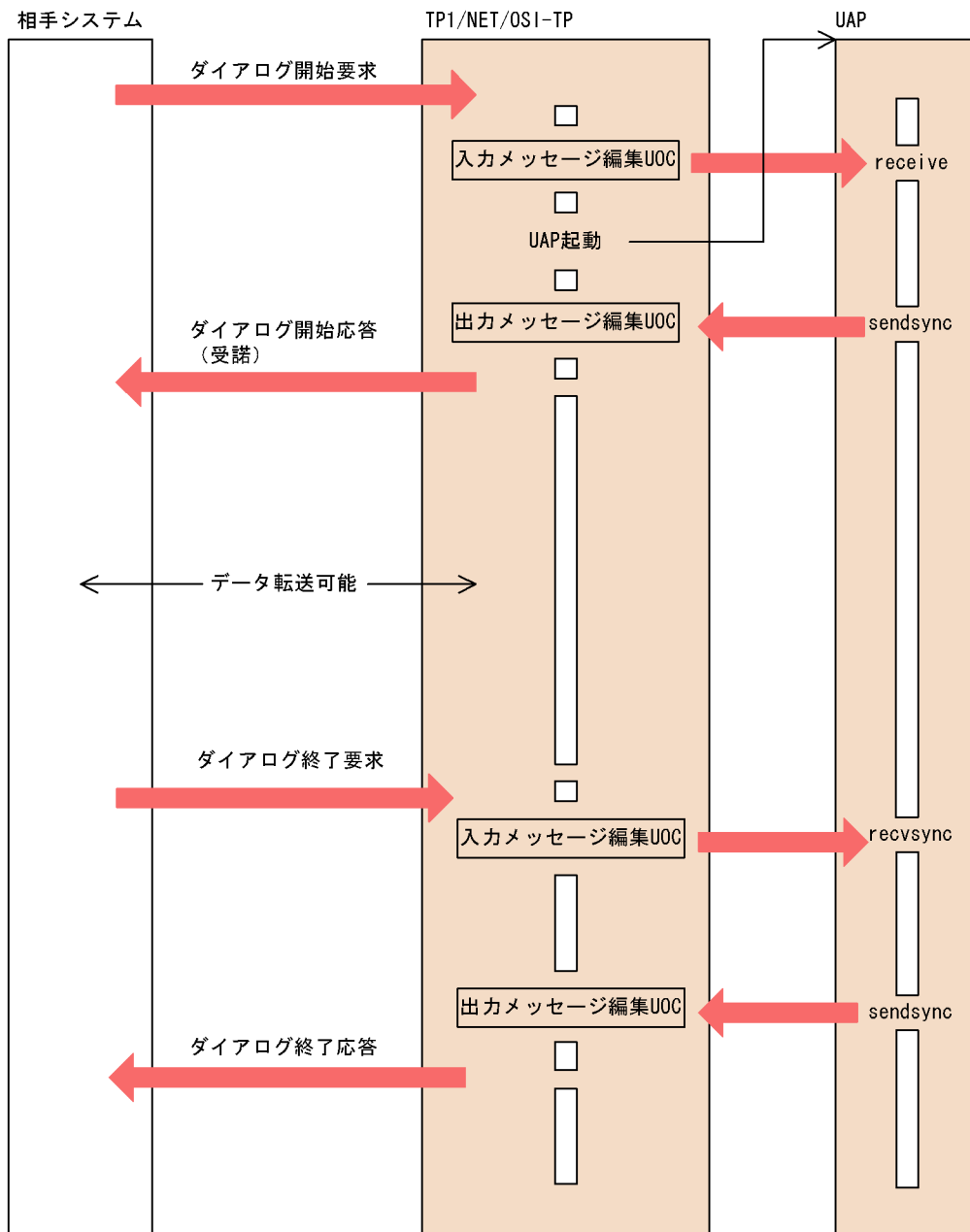
相手システムからダイアログ終了要求を受けた場合は、TP1/NET/OSI-TP は UAP にダイアログ終了要求を渡します。確認型の場合、UAP は SENDSYNC 文、または sendsync 関数でダイアログ終了応答を TP1/NET/OSI-TP に送信します。確認型でダイアログ終了を受け入れない場合は、ダイアログ終了拒否応答を送信します。

なお、ダイアログを終了しても、コネクションの確立状態は継続できます。

ダイアログによるメッセージ送受信では、プロトコル固有のインタフェースを使用します。インタフェースの詳細については、「3. メッセージ送受信インタフェース」の「サービスプリミティブ」を参照してください。

ダイアログの開始と終了を次の図に示します。

図 2-17 ダイアログの開始と終了



(3) ダイアログの異常終了

ダイアログの異常終了には、自システムおよび相手システムの UAP での異常検知による終了と、TP1/NET/OSI-TP または相手システムによる異常検知による終了とがあります。自システムおよび相手システムの UAP でのダイアログの異常終了 (TP-U-ABORT)

では、コネクションの確立状態は継続できます。TP1/NET/OSI-TP または相手システムでのダイアログ異常終了では、コネクションも強制解放されます。

(4) ダイアログと UAP

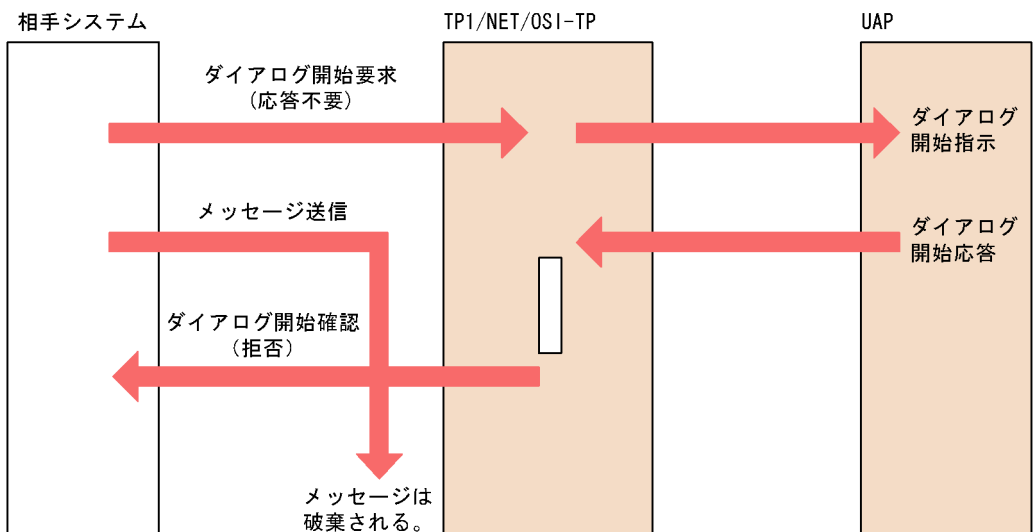
TP1/NET/OSI-TP では、一つの UAP でダイアログの開始から終了までの一連のメッセージ送受信をする必要があります。TP1/NET/OSI-TP は、UAP の終了時に未終了のダイアログを中止します。また、UAP からロールバックの資源回復要求をする場合、その要求はダイアログとは無関係です。ダイアログを継続して使用するか、中止するかの判断は UAP に依存します。

(5) メッセージの応答の要否

ダイアログの開始要求時は、メッセージの応答を必ず要求するようにしてください。応答を不要とすると、相手システムがダイアログ開始要求を拒否した場合、メッセージが破棄されることがあります。また、一方のシステムからダイアログ開始要求（応答不要）後にメッセージを送信し、ダイアログ終了要求（応答不要）を送信した場合、メッセージが相手システムに届かないことがあります。

応答を不要とした場合にメッセージを破棄する例を次の図に示します。

図 2-18 応答を不要とした場合のメッセージ破棄の例



(6) ダイアログ開始方向

TP1/NET/OSI-TP を使用して相手システムと通信する場合、ダイアログの開始要求は必ずコンテンション勝者側から行ってください。コンテンション敗者側からのダイアログ開始要求はエラーとなります。コンテンションの勝者、および敗者の指定は、相手システムとの間で事前に取り決めておいてください。

2.2.5 アプリケーション名の決定

TP1/NET/OSI-TP では、ダイアログ開始要求受信時、アプリケーション名 (MHP 名) を決定できます。アプリケーション名は、次に示す場合に決定できます。

- MCF 通信構成定義 (mcftalcle -v) でアプリケーション名が指定されている場合
- ダイアログ開始要求 (TP-BEGIN-DIALOGUEind) のパラメタである、受信側 TPSU 名称が設定されている場合
受信側 TPSU 名称の長さが 8 バイトを超える場合は、8 バイトまでを有効とします。
- 入力メッセージ編集 UOC でアプリケーション名が設定されている場合

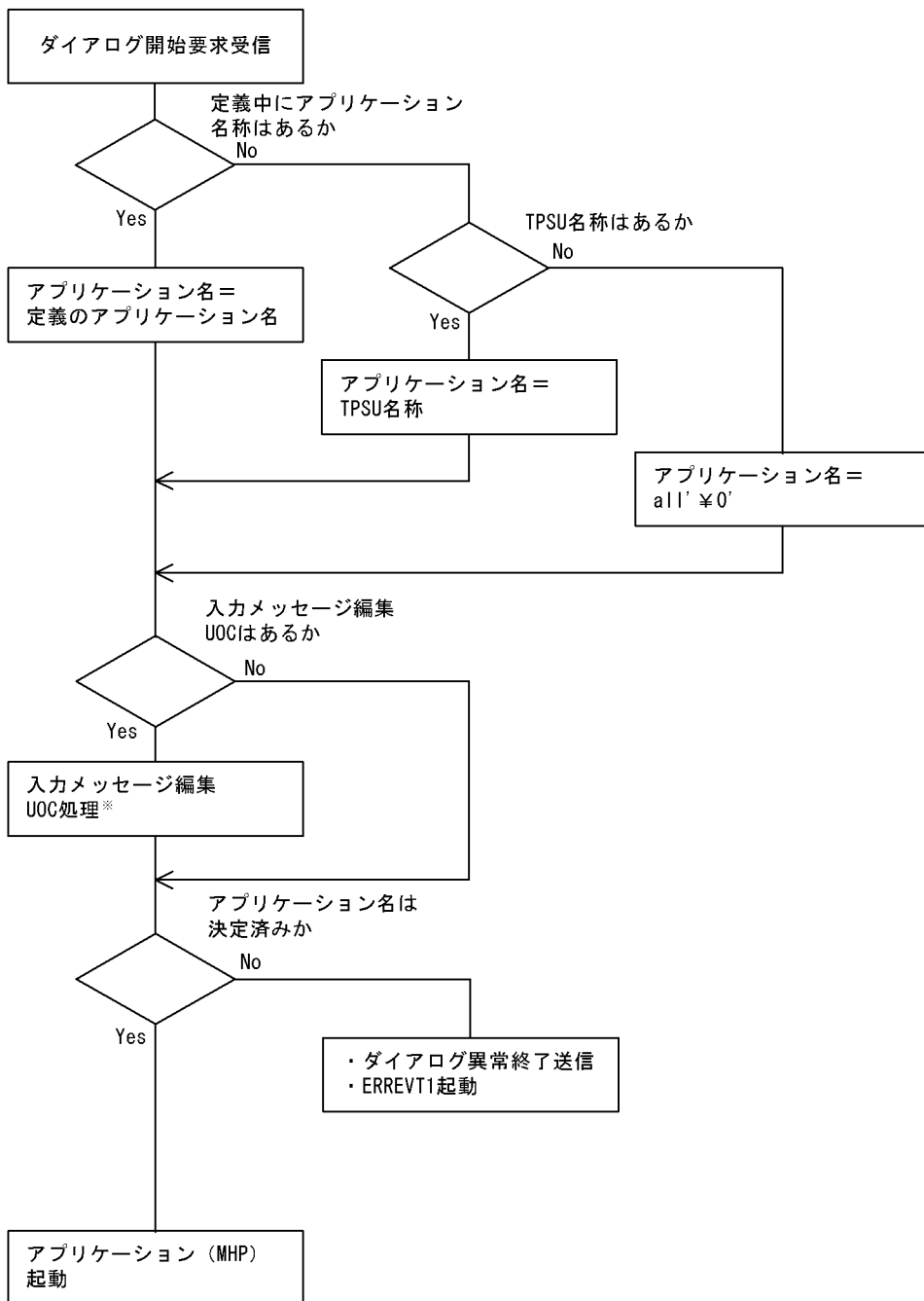
アプリケーション名の決定の優先順位を、高い順に次に示します。

1. 入力メッセージ編集 UOC での決定
2. MCF 通信構成定義 (mcftalcle -v) での決定
3. TPSU 名称での決定

アプリケーション名を決定できなかった場合、エラーイベント (ERREVT1) を起動します。

TP1/NET/OSI-TP によるアプリケーション名を決定する処理の流れを、次の図に示します。

図 2-19 アプリケーション名の決定



注

入力メッセージ編集 UOC によるアプリケーション名の決定の処理については、
「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。

2.2.6 ハンドシェイク

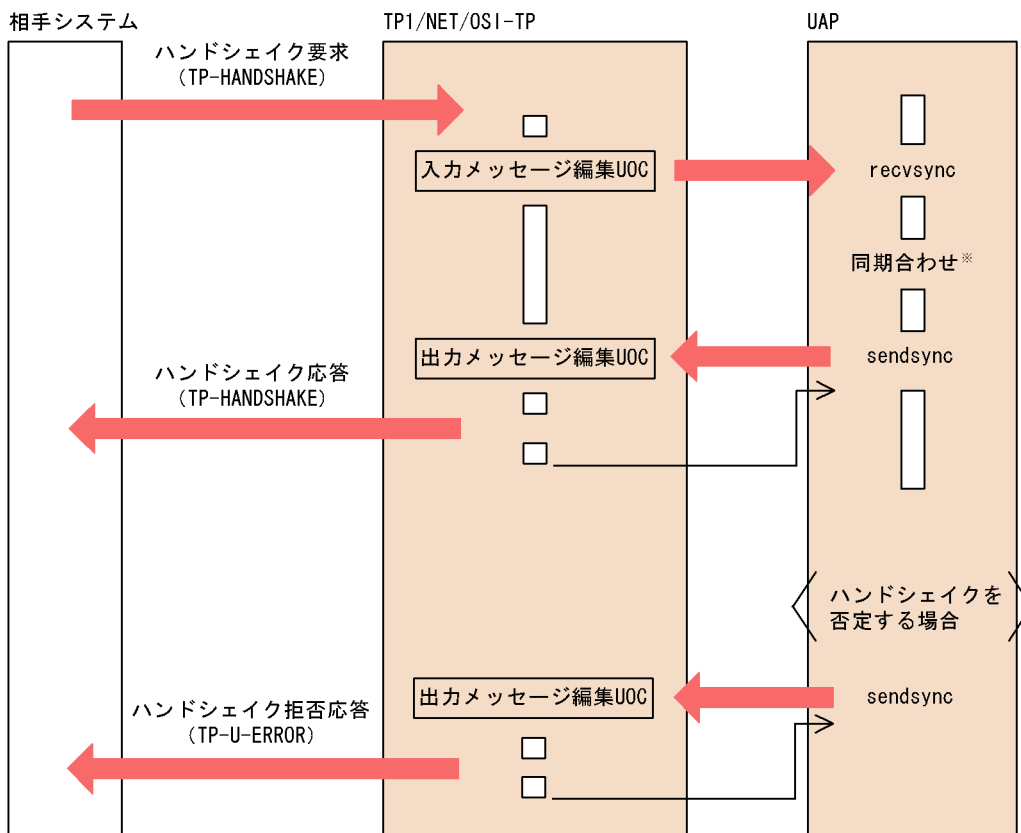
ハンドシェイクとは、システム間で同期を取ってデータを処理するための機能です。同期用の処理をユーザ側で設定する必要がないため、処理を軽減できます。

ハンドシェイク要求は、全二重の場合コンテンションでの優先権に関係なく、どちらのシステムからでも発行できます。半二重の場合制御権のある側だけが発行できます。

相手システムからハンドシェイク要求を受けると、TP1/NET/OSI-TP はメッセージを編集し、UAP に渡します。UAP は同期合わせをしたあと、相手システムにハンドシェイク応答 (TP-HANDSHAKE) を送信します。要求を拒否する場合は、ハンドシェイクの拒否応答 (TP-U-ERROR) を送信します。このとき制御権は移動します。

TP1/NET/OSI-TP がハンドシェイクの要求を受信した場合の処理を、次の図に示します。

図 2-20 ハンドシェイク



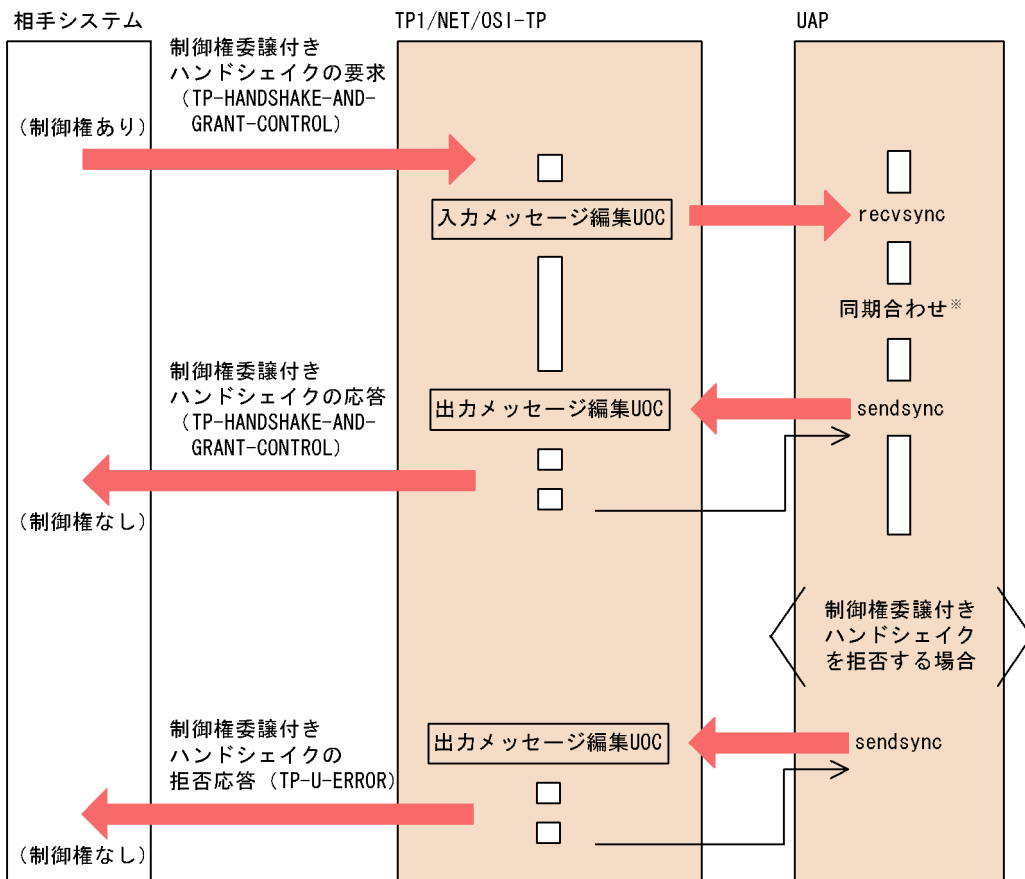
注

同期が取れなかった場合、UAP はハンドシェイクを拒否します。TP1/NET/OSI-TP は、相手システムにハンドシェイク拒否応答を返します。

なお、半二重の場合ハンドシェイクが完了しても制御権は移動しません。

半二重通信の場合、制御権委譲とハンドシェイクをまとめて一度に処理できます。相手システムから制御権委譲付きハンドシェイク (TP-HANDSHAKE-AND-GRANT-CONTROL) の要求を受けると、TP1/NET/OSI-TP はメッセージを編集し、制御権委譲付きハンドシェイク (TP-HANDSHAKE-AND-GRANT-CONTROL) の指示を UAP に渡します。UAP は受諾する場合、sendsync で制御権委譲付きハンドシェイクの応答 (TP-HANDSHAKE-AND-GRANT-CONTROL) を送信します。また、要求を拒否する場合は、制御権委譲付きハンドシェイクの拒否応答 (TP-U-ERROR) を送信します。制御権は応答の内容に関係なく移動します。制御権委譲付きハンドシェイクの処理を次の図に示します。

図 2-21 制御権委譲付きハンドシェイク



注

同期が取れなかった場合、UAP は制御権委譲付きハンドシェイクを拒否します。TP1/NET/OSI-TP は、相手システムに制御権委譲付きハンドシェイクの拒否応答を

返します。

2.2.7 データ転送

TP1/NET/OSI-TP では、自システムのユーザデータを相手システムに転送して処理できます。また、相手システムのユーザデータも受信できます。

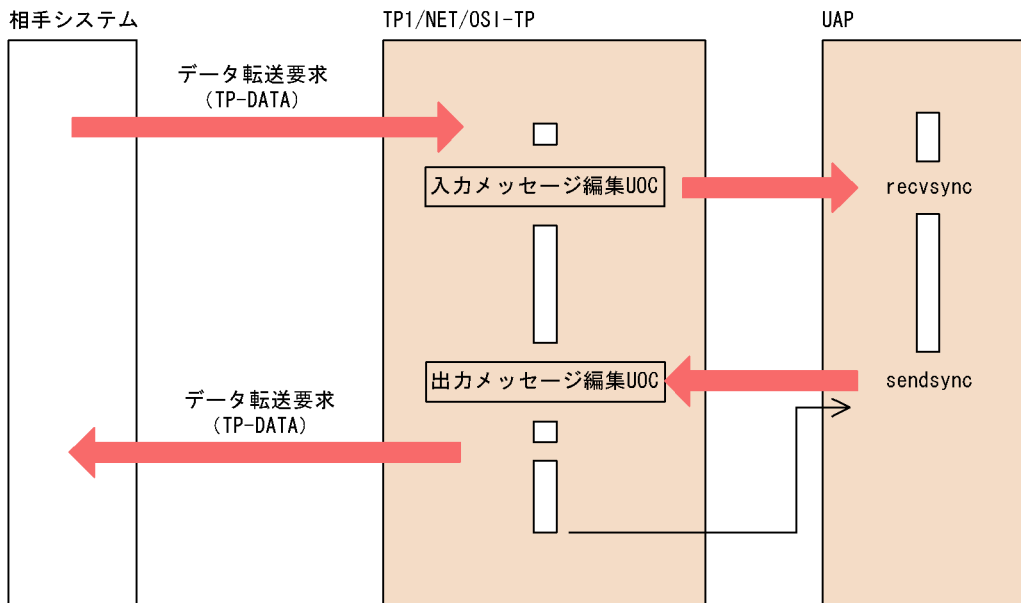
相手システムからデータ転送要求を受けると、TP1/NET/OSI-TP はメッセージを編集し、UAP に渡します。

また、自システムからデータ転送をする場合は、UAP から SENDSYNC 文または sendsync 関数による要求を受けて、相手システムにデータを送信します。

データ転送要求はダイアログ開始後、コンテンションの勝者、敗者のどちらからでも発行できます。半二重の場合、制御権のある側だけがデータ転送要求を出すことができます。

データ転送の処理を、次の図に示します。

図 2-22 データ転送



2.2.8 エラー通知

エラー通知は、自システムでエラーが発生したことを相手システムに通知する場合と、ハンドシェイク、制御権委譲付きハンドシェイク、または確認型ダイアログ終了の拒否応答として使用する場合があります。

(1) 全二重の場合

UAP から SENDSYNC 文または `sendsync` 関数で要求を受けると、TP1/NET/OSI-TP は相手システムにエラー通知を送信します。ただし、相手システムからエラー通知応答を受信した場合、TP1/NET/OSI-TP は UAP に対してエラー通知応答はしません。

相手システムからエラー通知を受信した場合、UAP は RECVSYNC 文、または `recvsync` 関数でメッセージの受信を要求します。TP1/NET/OSI-TP は相手システムにエラー通知応答を送信し、相手システムから受信したメッセージを UAP に渡します。

UAP でのエラー発生時の処理を図 2-23 に、相手システムからのエラー通知受信の処理を図 2-24 に示します。

図 2-23 UAP でのエラー発生時の処理（全二重の場合）

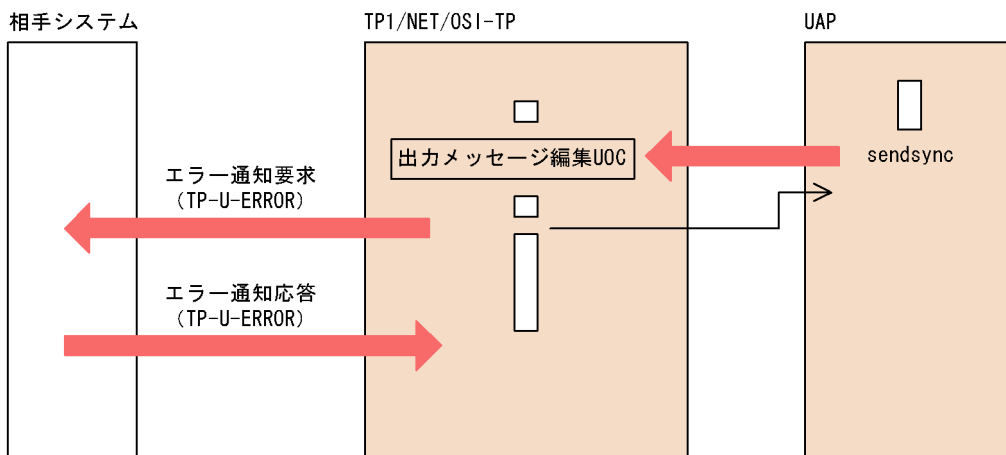
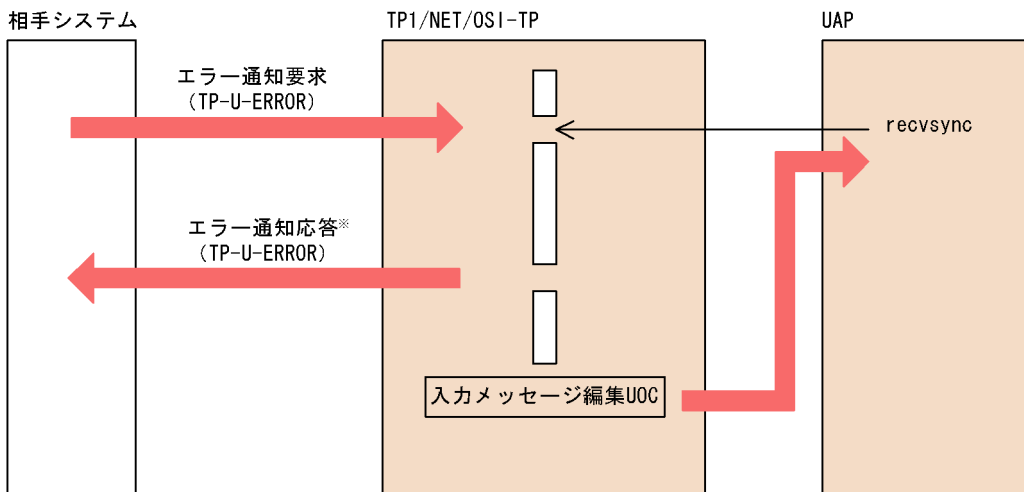


図 2-24 相手システムからのエラー通知受信の処理（全二重の場合）



2. 機能

注

次に示す場合、TP1/NET/OSI-TP はエラー通知応答を送信しません。

- ダイアログ終了要求 (Confirmation='True') の送信後、エラー通知要求を受信したとき
- ハンドシェイク要求の送信後、エラー通知要求を受信したとき

(2) 半二重の場合

(a) 制御権のある側からエラー通知の送信、ない側でエラー通知の受信

制御権のあるシステムでは UAP から SENDSYNC 文または `sendsync` 関数でエラー通知の要求を受けると、TP1/NET/OSI-TP は相手システムにエラー通知 (TP-U-ERROR) 要求を送信します。

制御権のないシステムでは、TP1/NET/OSI-TP が相手システムからエラー通知 (TP-U-ERROR) 要求を受信すると、RECVSYNC 文または `recvsync` 関数で受信待ちになっている UAP にエラー通知 (TP-U-ERROR) 指示を通知します。このとき制御権は移動しません。UAP でのエラー発生時の処理を図 2-25 に、相手システムからのエラー受信の処理を図 2-26 に示します。

図 2-25 UAP でのエラー発生時の処理 (制御権がある場合)

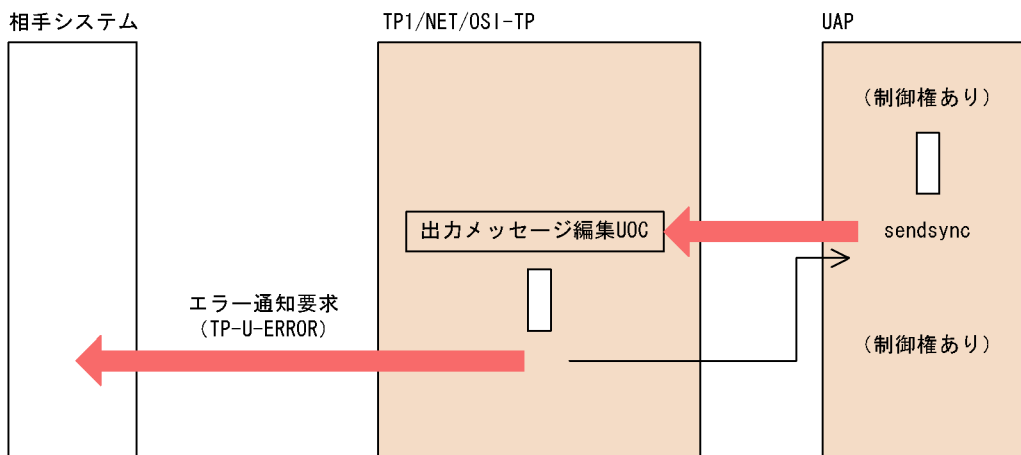
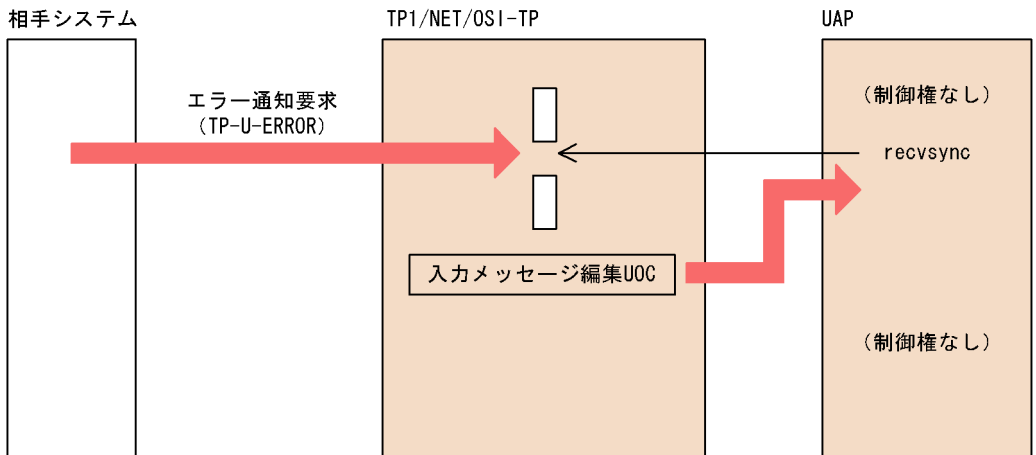


図 2-26 相手システムからのエラー通知受信の処理（制御権がない場合）



(b) 制御権のない側からエラー通知の送信，ある側でエラー通知の受信

制御権のないシステムでは UAP から SENDSYNC 文または `sendsync` 関数でエラー通知の要求を受けると，TP1/NET/OSI-TP は相手システムにエラー通知 (TP-U-ERROR) 要求を送信します。

制御権のあるシステムでは，TP1/NET/OSI-TP が相手システムからエラー通知 (TP-U-ERROR) 要求を受信すると，RECVSYNC 文または `recvsync` 関数で受信待ちになっている UAP にエラー通知 (TP-U-ERROR) 指示を通知します。エラー通知 (TP-U-ERROR) 指示を受けた UAP は，SENDSYNC 文または `sendsync` 関数で相手システムに制御権委譲 (TP-GRANT-CONTROL) 要求を送信しなければなりません。

制御権のないシステムでは相手システムから制御権委譲 (TP-GRANT-CONTROL) 要求を受信すると，RECVSYNC 文または `recvsync` 関数で受信待ちになっている UAP に制御権委譲 (TP-GRANT-CONTROL) 指示を通知します。このとき制御権は移動します。UAP でのエラー発生時の処理を図 2-27 に，相手システムからのエラー受信の処理を図 2-28 に示します。

2. 機能

図 2-27 UAP でのエラー発生時の処理（制御権がない場合）

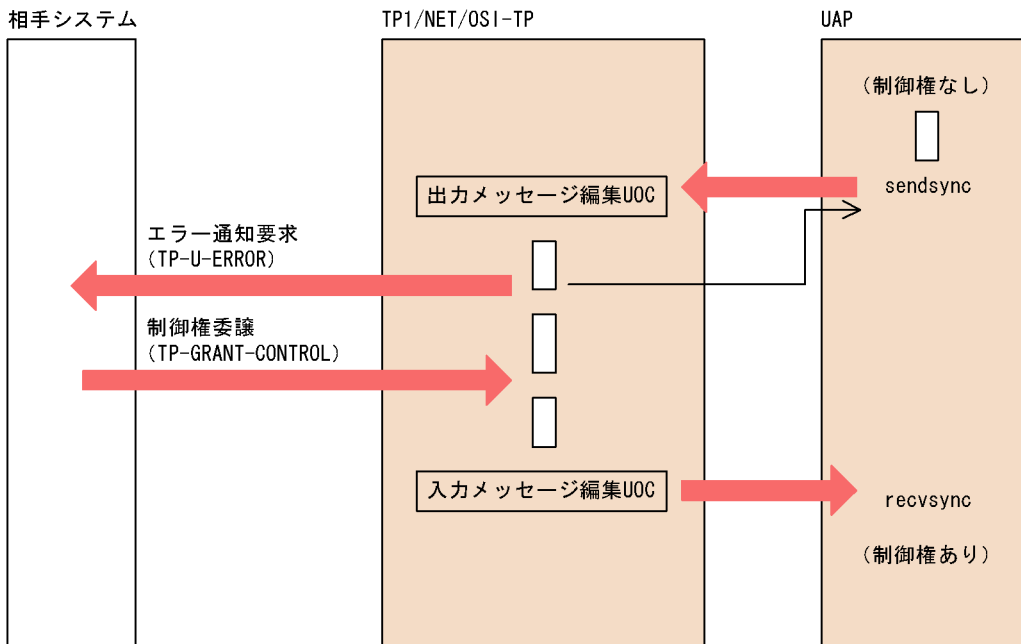
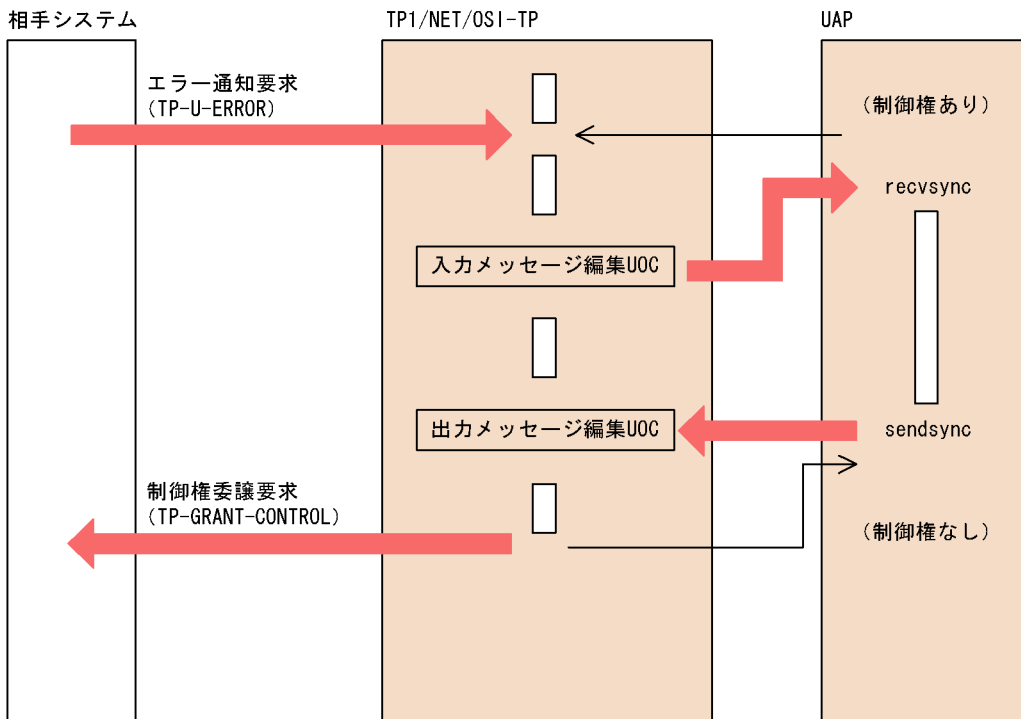


図 2-28 相手システムからのエラー通知受信の処理（制御権がある場合）



なお、エラー通知をしても、ダイアログは終了しません。ダイアログを終了させたい場合は、UAP からダイアログ終了要求を送信してください。

発生したエラーの詳細情報は、エラー通知後、データ転送で送受信する必要があります。また、エラーの詳細は、ユーザ間で事前に取り決めておきます。

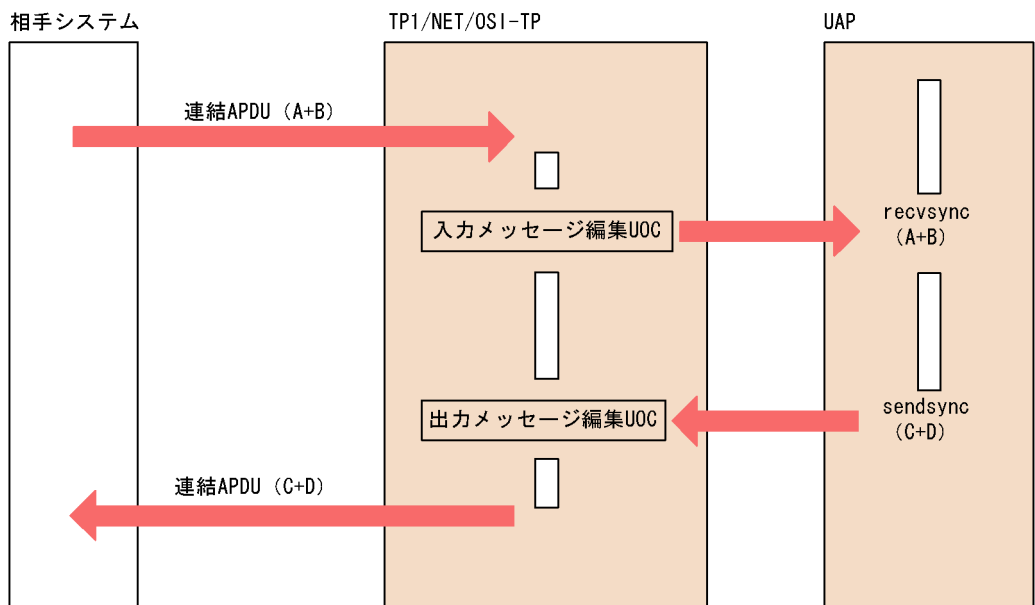
2.2.9 APDU 連結

OSI TP プロトコルには、複数の APDU をリンケージして 1 回で送受信をする、連結機能があります。TP1/NET/OSI-TP では、これを APDU 連結といい、UAP がサービスプリミティブの連結および分離を行います。連結によって、処理要求の送信回数を減らし、システムにかかる負荷を軽減できます。

サービスプリミティブまたは APDU を複数連結する場合、発行順序は許されるすべてのパターンで、また、連結の個数は送受信バッファに格納できる範囲で最大 63 個連結できます。

APDU 連結の処理を、次の図に示します。

図 2-29 APDU 連結



3

メッセージ送受信インタフェース

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

メッセージ送受信インタフェースの一覧

dc_mcf_receive - 一方送信メッセージの受信 (C 言語)

dc_mcf_recvsync - 同期型メッセージの受信 (C 言語)

dc_mcf_sendrecv - 同期型メッセージの送受信 (C 言語)

dc_mcf_sendsync - 同期型メッセージの送信 (C 言語)

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

CBLDCMCF('RECVSYNC') - 同期型メッセージの受信 (COBOL 言語)

CBLDCMCF('SENDRECV') - 同期型メッセージの送受信 (COBOL 言語)

CBLDCMCF('SENDSYNC') - 同期型メッセージの送信 (COBOL 言語)

DISABLE - 同期型メッセージの送信 (データ操作言語)

ENABLE - 同期型メッセージの送信 (データ操作言語)

RECEIVE - メッセージの受信 (データ操作言語)

SEND - 同期型メッセージの送受信 (データ操作言語)

サービスプリミティブ

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

メッセージ送受信インタフェースの一覧

TP1/NET/OSI-TP 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 システムで提供する関数を使用して UAP を作成します。

メッセージ送受信の関数を次の表に示します。

表 3-1 メッセージ送受信の関数 (C 言語)

関数名	機能
dc_mcf_receive	一方送信メッセージの受信
dc_mcf_recvsync	同期型メッセージの受信
dc_mcf_sendrecv	同期型メッセージの送受信
dc_mcf_sendsync	同期型メッセージの送信

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

COBOL 言語のメッセージ送受信

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

メッセージ送受信の関数に対応するプログラムを次の表に示します。

表 3-2 メッセージ送受信の関数に対応するプログラム (COBOL 言語)

プログラム名	データ名 A に設定する要求コード	意味
CBLDCMCF	'RECEIVE '	一方送信メッセージの受信
	'RECVSYNC'	同期型メッセージの受信
	'SENDRECV'	同期型メッセージの送受信
	'SENDSYNC'	同期型メッセージの送信

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。

3. メッセージ送受信インタフェース
 メッセージ送受信インタフェースの一覧

データ操作言語 (COBOL 言語) のメッセージ送受信

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

TP1/NET/OSI-TP 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	DISABLE	同期型メッセージの送信	CBLDCMCF('SENDSYNC')
	ENABLE	同期型メッセージの送信	CBLDCMCF('SENDSYNC')
	RECEIVE	一方送信メッセージの受信	CBLDCMCF('RECEIVE ')
		同期型メッセージの受信	CBLDCMCF('RECVSYNC')
	SEND	同期型メッセージの送信	CBLDCMCF('SENDSYNC')
		同期型メッセージの送受信	CBLDCMCF('SENDRECV')

dc_mcf_receive - 一方送信メッセージの受信 (C 言語)

形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

機能

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

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

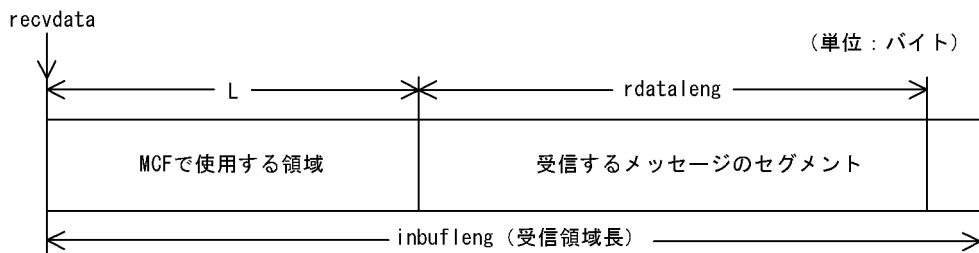
- 相手システムから送信されたメッセージ
- MCF イベント
- アプリケーション起動で渡されたメッセージ

TP1/NET/OSI-TP 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。

3. メッセージ送受信インタフェース

dc_mcf_receive - 一方送信メッセージの受信 (C 言語)



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指定します。

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

DCMCFRST

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

DCMCFSEG

中間セグメントまたは最終セグメントを受信する場合に設定します。

DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

commform

DCNOFLAGS を設定します。

termnam

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

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

resv01

ヌル文字を設定します。

recvdata

セグメントを受信する領域を設定します。

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

処理終了後、recvdata には OpenTP1 から値が返ります。

inbufleng

セグメントを受信する領域の長さを設定します。

OpenTP1 から値が返される引数

termnam

先頭セグメントを受信する場合だけ、入力元の論理端末名称が返されます。

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

recvdata

受信したセグメントの内容が返されます。

rdatale

受信したセグメントの長さが返されます。

time

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

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRT N_00000	0	正常に終了しました。
DCMCFRT N_71000	-12000	先頭セグメントを受信する dc_mcf_receive 関数を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、action に DCMCFSEG を設定して dc_mcf_receive 関数を呼び出してください。
DCMCFRT N_71001	-12001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する dc_mcf_receive 関数を呼び出しています。直前に呼び出した dc_mcf_receive 関数でメッセージはすべて受信しました。このリターン値が返されたあとに、再び dc_mcf_receive 関数を呼び出した場合は、リターン値 DCMCFRTN_72000 が返されます。
DCMCFRT N_71002	-12002	メッセージキューからの入力処理中に障害が発生しました。

3. メッセージ送受信インタフェース

dc_mcf_receive - 一方送信メッセージの受信 (C 言語)

リターン値	リターン値 (数値)	意味
		メッセージキューが閉塞されています。
DCMCFRT N_72000	-13000	<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>
DCMCFRT N_72001	-13001	termnam に設定した論理端末名称が間違っています。
DCMCFRT N_72013	-13013	inbufleng の指定値を超えるメッセージ (セグメント) を受信しました。inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRT N_72016	-13016	<p>action に設定した値が間違っています。</p> <hr/> <p>resv01 に設定した値が間違っています。</p> <hr/> <p>引数に設定した値が間違っています。</p>
DCMCFRT N_72024	-13024	commform に設定した値が間違っています。
DCMCFRT N_72025	-13025	action に設定したセグメント種別 (DCMCFRST または DCMCFSEG) の値が間違っています。
DCMCFRT N_72036	-13036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
上記以外	-	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

- : 該当しません。

dc_mcf_recvsync - 同期型メッセージの受信 (C 言語)

形式

ANSI C, C++ の形式

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

K&R 版 C の形式

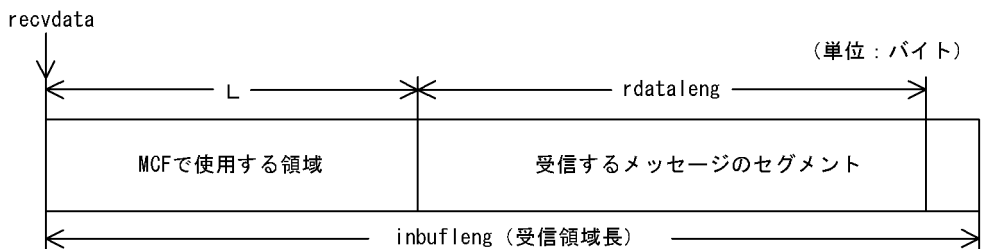
```
#include <dcmcf.h>
int dc_mcf_recvsync (action, commform, termnam, resv01,
                    recvdata, rdataleng, inbufleng, time,
                    watchtime)

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

機能

相手システムから同期型でメッセージを受信します。相手システムから送信されるメッセージは、常に一つの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

受信する論理メッセージのセグメントと、使用するバッファ形式を次の形式で指定します。

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

DCMCFFRST

単一セグメントを示す DCMCFFRST を設定します。

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

入力元の論理端末名称を設定します。論理端末名称は最大 8 バイトの長さです。論理端末の名称の最後にはヌル文字を付けてください。

resv01

ヌル文字を設定します。

recvdata

セグメントを受信する領域を設定します。

dc_mcf_recvsync 関数がリターンすると、受信したセグメントが設定されます。

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。

watchtime

dc_mcf_recvsync 関数を呼び出してから終了するまでの、監視時間を設定します。0 を設定した場合は、MCF マネージャ定義の UAP 共通定義 (mcfmuap -t) で指定した、同期受信監視時間が設定されます。

負の値を設定した場合は、時間監視はしません。

OpenTP1 から値が返される引数

recvdata

受信したメッセージのセグメントの内容が返されます。

rdataleng

受信したセグメントの長さが返されます。

time

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

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRT N_00000	0	正常に終了しました。
DCMCFRT N_71001	-12001	メッセージのセグメントを受け取ったあとで、次のメッセージ (セグメント) を受信する dc_mcf_recvsync 関数を呼び出しています。直前に呼び出した dc_mcf_recvsync 関数でセグメントは受信済みです。このリターン値を受け取ったあとに、再び dc_mcf_recvsync 関数を呼び出した場合は、リターン値 DCMCFRTN_72000 が返されます。
DCMCFRT N_71002	-12002	MCF が終了処理中のため、dc_mcf_recvsync 関数を受け付けられません。
DCMCFRT N_71108	-12108	メッセージ受信に必要な管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
DCMCFRT N_72000	-13000	dc_mcf_recvsync 関数を呼び出した位置が間違っています。UAP でセグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_recvsync 関数を呼び出しています。
DCMCFRT N_72001	-13001	termnam に設定した論理端末名称は、MCF で定義していません。 termnam に設定した論理端末名称が間違っています。 dc_mcf_recvsync 関数を呼び出せない論理端末を設定しています。
DCMCFRT N_72013	-13013	inbufleng の指定値を超えるメッセージ (セグメント) を受信しました。inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRT N_72016	-13016	action に設定できない値を設定しています。 resv01 の指す領域の値が、ヌル文字になっていません。 引数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_recvsync - 同期型メッセージの受信 (C 言語)

リターン値	リターン値 (数値)	意味
DCMCFRT N_72024	-13024	commform に DCNOFLAGS を設定していません。
DCMCFRT N_72025	-13025	action に設定した値が間違っています。
DCMCFRT N_72036	-13036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
DCMCFRT N_73001	-14001	入力元の論理端末で障害が発生しました。
DCMCFRT N_73005	-14005	dc_mcf_recvsync 関数の処理中に、watchtime に設定した監視時間を過ぎているのに、論理端末からのメッセージが受信されません。
DCMCFRT N_73010	-14010	入力、および出力メッセージ編集 UOC で障害が発生しました。
		メッセージの読み込み時に障害が発生しました。
DCMCFRT N_73015	-14015	入力元の論理端末は、ほかの UAP で仕掛り中です。
DCMCFRT N_73018	-14018	watchtime に設定した監視時間の値が間違っています。
DCMCFRT N_73020	-14020	同期型のメッセージ受信で指定した論理端末は停止中です。
DCMCFRT N_77350	-18350	サービスプリミティブの発行順が間違っています。
		サービスプリミティブの連結数が超過しています。
DCMCFRT N_77433	-18433	相手からのメッセージによって、ダイアログが未確立になりました。
DCMCFRT N_77436	-18436	相手からのメッセージによって、ダイアログを解放しています。
DCMCFRT N_77437	-18437	相手からのメッセージによって、ハンドシェイクが応答待ちになりました。
DCMCFRT N_77491	-18491	相手からのメッセージによって、エラー処理中です。
上記以外	-	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

- : 該当しません。

dc_mcf_sendrecv - 同期型メッセージの送受信 (C 言語)

形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

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

機能

同期型でメッセージを送信したあと、同期型でメッセージを受信します。送信するメッセージも受信するメッセージも、一つのセグメントで構成されます。

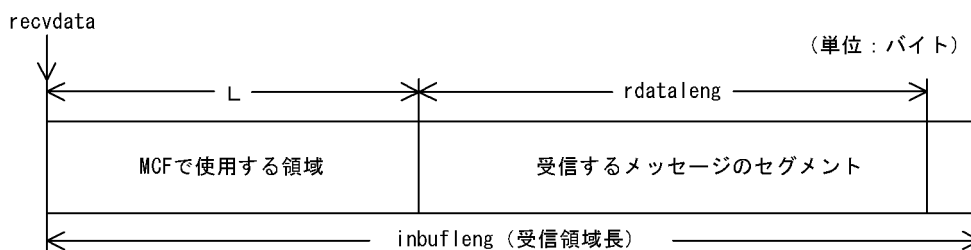
メッセージを送信すると、dc_mcf_sendrecv 関数は相手システムからの応答を待ちます。応答が届くと、そのメッセージを受信します。

セグメントを送信する領域と受信する領域の形式をそれぞれ示します。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。

3. メッセージ送受信インタフェース
 dc_mcf_sendrecv - 同期型メッセージの送受信 (C 言語)



受信領域の形式を次に示します。



UAP で値を設定する引数

action

送受信する論理メッセージのセグメントと、使用するバッファ形式を次の形式で指定します。

DCMCFEMI [| { DCMCFBUF1 | DCMCFBUF2 }]

DCMCFEMI

単一セグメントを示す DCMCFEMI を設定します。

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

同期型メッセージの送受信を示す、DCMCFIO を設定します。

termnam

メッセージを出力して応答を入力する論理端末名称を設定します。論理端末名称は最大 8 バイトの長さです。論理端末名称の最後にはヌル文字を付けてください。アソシエーション管理機能を使用する場合、ダイアログ開始要求時に論理端末名称としてコネクショングループ名を指定できます。また、termnam が示す論理端末名称領域には、

少なくとも 9 バイト (論理端末名称の最大長 8 バイト + ヌル文字 1 バイト) を UAP で確保し, termnam には文字列定数を指定しないでください。

コーディング例

(誤)

```
char *termnam="GR01";  
dc_mcf_sendrecv(..., termnam, ...);
```

(正)

```
char termnam[9];  
strcpy(termnam, "GR01");  
dc_mcf_sendrecv(..., termnam, ...);
```

注

termnam 以外の dc_mcf_sendrecv 関数の引数は省略しています。

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 バイトまで送信できます。

sdataleng

送信するセグメントの長さを設定します。

recvdata

セグメントを受信する領域を設定します。

dc_mcf_sendrecv 関数が終了すると, 受信したセグメントが設定されます。

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。

watchtime

dc_mcf_sendrecv 関数を呼び出してから終了するまでの監視時間を設定します。0 を設定した場合, MCF マネージャ定義の UAP 共通定義で指定した, 同期送受信監視時間 (mcfmuap -t sndrcvtim) が設定されます。

負の値を設定した場合は, 時間監視をしません。

3. メッセージ送受信インタフェース

dc_mcf_sendrecv - 同期型メッセージの送受信 (C 言語)

OpenTP1 から値が返される引数

recvdata

受信したセグメントの内容が返されます。

rdataleang

受信したセグメントの長さが返されます。

time

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

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRT N_00000	0	正常に終了しました。
DCMCFRT N_71002	-12002	メッセージキューへの入出力処理時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。
		sdataleang に 32000 バイトを超える値を設定しています。
DCMCFRT N_71003	-12003	メッセージキューが満杯です。
DCMCFRT N_71004	-12004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRT N_71108	-12108	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。
DCMCFRT N_72000	-13000	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_sendrecv 関数を呼び出しています。
DCMCFRT N_72001	-13001	termnam に設定した論理端末名称が間違っています。
		termnam に設定したメッセージ出力先論理端末名称は、MCF で定義していません。
		dc_mcf_sendrecv 関数を呼び出せない論理端末を設定しています。
DCMCFRT N_72013	-13013	inbufleng の指定値を超えるメッセージ (セグメント) を受信しました。 inbufleng の指定値を超えた部分は切り捨てられました。

3. メッセージ送受信インタフェース
dc_mcf_sendrecv - 同期型メッセージの送受信 (C 言語)

リターン値	リターン値 (数値)	意味
DCMCFRT N_72016	-13016	action に設定した値が間違っています。
		resv01 に設定した値が間違っています。
		引数に設定した値に間違いがあります。
DCMCFRT N_72024	-13024	commform に設定した値が間違っています。
DCMCFRT N_72026	-13026	action に設定した値が間違っています。
DCMCFRT N_72036	-13036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
DCMCFRT N_73001	-14001	出力先の論理端末で障害が発生しました。
DCMCFRT N_73005	-14005	dc_mcf_sendrecv 関数を呼び出してから、watchtime に設定した監視時間を過ぎてても、端末からの応答がありません。
DCMCFRT N_73010	-14010	入力または出力メッセージ編集 UOC で障害が発生しました。
		メッセージの読み込み時に障害が発生しました。
		メッセージの編集エラーが発生しました。
DCMCFRT N_73015	-14015	出力先の論理端末は、ほかの UAP で仕掛り中です。
DCMCFRT N_73018	-14018	watchtime に設定した値が間違っています。
DCMCFRT N_73020	-14020	メッセージ出力先の論理端末は、停止しています。
DCMCFRT N_77001	-18001	指定したコネクショングループ内に使用できるコネクションがありません。
DCMCFRT N_77301	-18301	送信セグメント長と、「制御情報 + パラメタ + ユーザデータ」のサイズが一致していません。
DCMCFRT N_77302	-18302	制御情報のサイズの設定が間違っています。
DCMCFRT N_77303	-18303	転送手順の設定が間違っています。
DCMCFRT N_77304	-18304	サービスプリミティブコードの設定が間違っています。
DCMCFRT N_77310	-18310	パラメタ情報サイズの設定が間違っています。
DCMCFRT N_77311	-18311	Initiating-TPSU-Title に設定した値が間違っています。
DCMCFRT N_77312	-18312	Recipient-AP-Title 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_sendrecv - 同期型メッセージの送受信 (C 言語)

リターン値	リターン値 (数値)	意味
DCMCFRT N_77313	-18313	Recipient-TPSU-Title に設定した値が間違っています。
DCMCFRT N_77314	-18314	Functional-Units に設定した値が間違っています。
DCMCFRT N_77315	-18315	Quality-of-Service に設定した値が間違っています。
DCMCFRT N_77316	-18316	Application-Context-Name に設定した値が間違っています。
DCMCFRT N_77317	-18317	Confirmation に設定した値が間違っています。
DCMCFRT N_77318	-18318	Result に設定した値が間違っています。
DCMCFRT N_77321	-18321	Confirmation-Urgency に設定した値が間違っています。
DCMCFRT N_77330	-18330	ユーザデータのサイズが間違っています。
DCMCFRT N_77331	-18331	APDU の連結状態が間違っています。
DCMCFRT N_77332	-18332	ユーザデータのサイズが設定されていません。
DCMCFRT N_77333	-18333	ユーザデータが設定できません。
DCMCFRT N_77350	-18350	サービスプリミティブの発行順が間違っています。
		サービスプリミティブの連結数が超過しています。
DCMCFRT N_77351	-18351	送信バッファが確保できません。
DCMCFRT N_77433	-18433	相手からのメッセージによって、ダイアログが未確立になりました。
DCMCFRT N_77436	-18436	相手からのメッセージによって、ダイアログを解放中です。
DCMCFRT N_77437	-18437	相手からのメッセージによって、ハンドシェイクが応答待ちになりました。
DCMCFRT N_77491	-18491	相手からのメッセージによって、エラー処理中です。
上記以外	-	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

- : 該当しません。

dc_mcf_sendsync - 同期型メッセージの送信 (C 言語)

形式

ANSI C, C++ の形式

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

K&R 版 C の形式

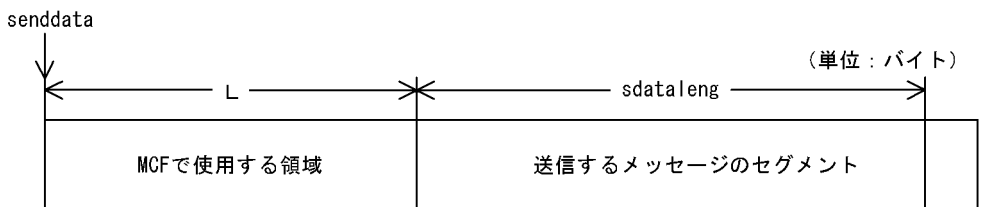
```
#include <dcmcf.h>
int dc_mcf_sendsync (action, commform, termnam, resv01,
                    senddata, sdataleNG, resv02, opcd,
                    watchtime)
```

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

機能

相手システムへ同期型でメッセージを送信します。メッセージは、一つのセグメントで構成されます。

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



UAP で値を設定する引数

action

単一セグメントの送信, および使用するバッファ形式を次の形式で指定します。

```
DCMCFEMI [ | { DCMCFBUF1 | DCMCFBUF2 } ]
```

DCMCFEMI

単一セグメントを示す DCMCFEMI を設定します。

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

出力先の論理端末名称を設定します。論理端末名称は最大 8 バイトの長さです。論理端末名称の最後にはヌル文字を付けてください。アソシエーションプール管理機能を使用する場合, ダイアログ開始要求時に論理端末名称としてコネクショングループ名を指定できます。また, termnam が示す論理端末名称領域には, 少なくとも 9 バイト (論理端末名称の最大長 8 バイト + ヌル文字 1 バイト) を UAP で確保し, termnam には文字列定数を指定しないでください。

コーディング例

(誤)

```
char *termnam="GR01";  
dc_mcf_sendsync(..., termnam, ...);
```

(正)

```
char termnam[9];  
strcpy(termnam, "GR01");  
dc_mcf_sendsync(..., termnam, ...);
```

注

termnam 以外の dc_mcf_sendsync 関数の引数は省略しています。

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 バイトまで送信できます。送信セグメントの長さが 0 の場合で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。

resv02

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

watchtime

dc_mcf_sendsync 関数を呼び出してから終了するまでの監視時間を設定します。0 を設定した場合は、MCF マネージャ定義の UAP 共通定義 (mcfmuap -t) で指定した同期送信監視時間が設定されます。

負の値を設定した場合は、時間監視をしません。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRT N_00000	0	正常に終了しました。
DCMCFRT N_71002	-12002	sdataleng 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。
DCMCFRT N_71003	-12003	メッセージキューが満杯です。
DCMCFRT N_71004	-12004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRT N_71108	-12108	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。
DCMCFRT N_72000	-13000	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_sendsync 関数を呼び出しています。
DCMCFRT N_72001	-13001	termnam に設定した論理端末名称が間違っています。 termnam に設定した出力先の論理端末名称は、MCF で定義していません。

3. メッセージ送受信インタフェース

dc_mcf_sendsync - 同期型メッセージの送信 (C 言語)

リターン値	リターン値 (数値)	意味
		dc_mcf_sendsync 関数を呼び出せない論理端末を設定しています。
DCMCFRT N_72016	-13016	action に設定できない値を設定しています。 opcd に設定した値が間違っています。 resv01, 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。
DCMCFRT N_72024	-13024	commform に設定した値が間違っています。
DCMCFRT N_72026	-13026	action に設定したセグメント種別 (DCMCFEMI) の値が間違っています。
DCMCFRT N_72041		sdataleng に 0, またはマイナス値を設定しています。
DCMCFRT N_73001	-14001	出力先の論理端末で障害が発生しました。
DCMCFRT N_73005	-14005	dc_mcf_sendsync 関数を呼び出してから, watchtime に設定した時間を経過しましたが, 端末からの応答がありません。
DCMCFRT N_73010	-14010	出力メッセージ編集 UOC で障害が発生しました。 メッセージの読み込み時に障害が発生しました。
DCMCFRT N_73015	-14015	出力先の論理端末は, ほかの UAP で仕掛り中です。
DCMCFRT N_73018	-14018	watchtime に設定した監視時間が間違っています。
DCMCFRT N_73020	-14020	出力先の論理端末は停止しています。
DCMCFRT N_77001	-18001	指定したコネクショングループ内に使用できるコネクションがありません。
DCMCFRT N_77301	-18301	送信セグメントサイズと, 「制御情報 + パラメタ + ユーザデータ」のサイズとが一致していません。
DCMCFRT N_77302	-18302	制御情報のサイズの設定が間違っています。
DCMCFRT N_77303	-18303	転送手順の設定が間違っています。
DCMCFRT N_77304	-18304	サービスプリミティブコードの設定が間違っています。
DCMCFRT N_77310	-18310	パラメタ情報サイズの設定が間違っています。
DCMCFRT N_77311	-18311	Initiating-TPSU-Title に設定した値が間違っています。
DCMCFRT N_77312	-18312	Recipient-AP-Title 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_sendsync - 同期型メッセージの送信 (C 言語)

リターン値	リターン値 (数値)	意味
DCMCFRT N_77313	-18313	Recipient-TPSU-Title に設定した値が間違っています。
DCMCFRT N_77314	-18314	Functional-Units に設定した値が間違っています。
DCMCFRT N_77315	-18315	Quality-of-Service に設定した値が間違っています。
DCMCFRT N_77316	-18316	Application-Context-Name に設定した値が間違っています。
DCMCFRT N_77317	-18317	Confirmation に設定した値が間違っています。
DCMCFRT N_77318	-18318	Result に設定した値が間違っています。
DCMCFRT N_77321	-18321	Confirmation-Urgency に設定した値が間違っています。
DCMCFRT N_77330	-18330	ユーザデータのサイズが間違っています。
DCMCFRT N_77331	-18331	APDU の連結状態が間違っています。
DCMCFRT N_77332	-18332	ユーザデータのサイズが設定されていません。
DCMCFRT N_77333	-18333	ユーザデータが設定できません。
DCMCFRT N_77350	-18350	サービスプリミティブの発行順が間違っています。 サービスプリミティブの連結数が超過しています。
DCMCFRT N_77433	-18433	相手からのメッセージによって、ダイアログが未確立になりました。
DCMCFRT N_77436	-18436	相手からのメッセージによって、ダイアログを解放中です。
DCMCFRT N_77437	-18437	相手からのメッセージによって、ハンドシェイクが応答待ちになりました。
DCMCFRT N_77491	-18491	相手からのメッセージによって、エラー処理中です。
上記以外	-	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

- : 該当しません。

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

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

機能

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

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

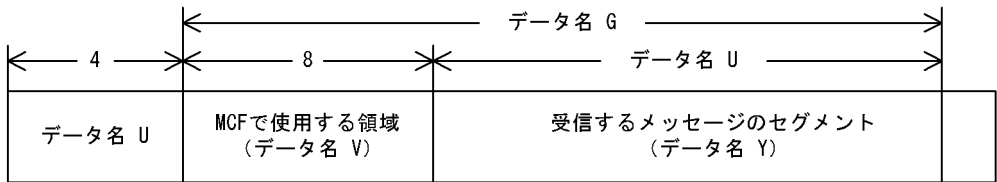
- 相手システムから送信されたメッセージ
- MCF イベント
- アプリケーション起動で渡されたメッセージ

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

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

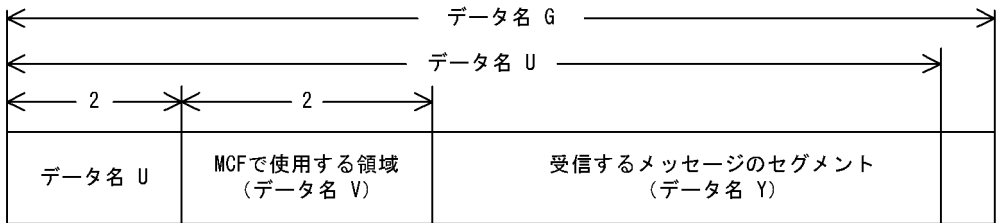
●バッファ形式 1 の場合

(単位: バイト)



●バッファ形式 2 の場合

(単位: バイト)



UAP で値を設定するデータ領域

データ名 A

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

データ名 C

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

VALUE 'FRST'

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

VALUE 'SEG '

中間セグメントまたは最終セグメントを受信する場合に設定します。

データ名 D

空白を設定します。

3. メッセージ送受信インターフェース

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

データ名 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指定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白

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

データ名 N

MCF で使用する領域です。

データ名 O

空白を設定します。

データ名 P

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

先頭セグメントの受信処理終了後, データ名 P には OpenTP1 から値が返ります。

データ名 Q, 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は 32763 バイトです。

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。

3. メッセージ送受信インタフェース

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

ステータスコード	意味
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	データ名 G の指定値を超えるメッセージ (セグメント) を受信しました。データ名 G の指定値を超えた部分は切り捨てられました。 32763 バイトを超えるメッセージ (セグメント) を受信しました。32763 バイトを超えた部分は切り捨てられました。
72016	データ名 D に設定した値が間違っています。 データ名 N またはデータ名 T に設定した値が間違っています。 データ名 M7 に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72025	データ名 C に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
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('RECVSYNC') - 同期型メッセージの受信 (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 'RECVSYNC'.  
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.  
02 データ名M6 PIC X(1) VALUE SPACE.  
02 データ名M7 PIC X(1).  
02 データ名N PIC X(14) VALUE LOW-VALUE.  
01 一意名2.  
02 データ名O PIC X(4) VALUE SPACE.  
02 データ名P PIC X(8).  
02 データ名Q PIC X(8) VALUE SPACE.  
02 データ名R PIC X(8) VALUE SPACE.  
02 データ名T PIC X(28) VALUE LOW-VALUE.  
01 一意名3.  
02 データ名U PIC 9(x) COMP.  
02 データ名V PIC X(x).  
02 データ名Y PIC X(n).
```

機能

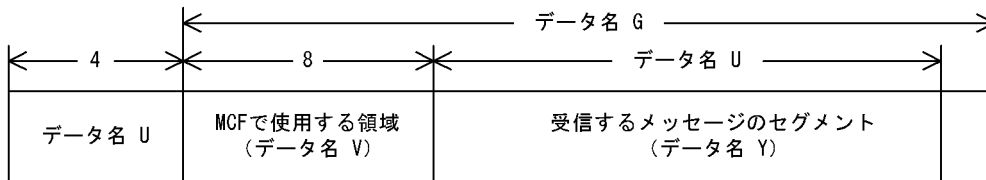
相手システムから同期型でメッセージを受信します。相手システムから送信されるメッセージは、常に一つのセグメントで構成されます。

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。

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

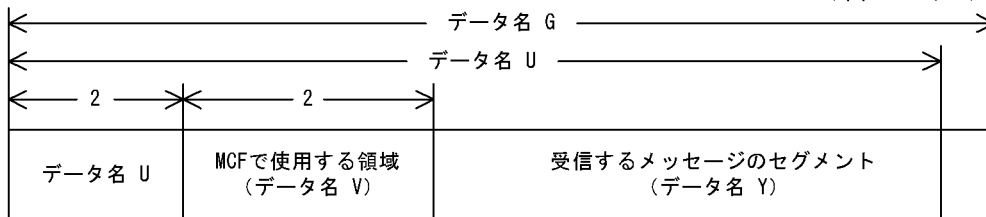
●バッファ形式1の場合

(単位: バイト)



●バッファ形式2の場合

(単位: バイト)



UAP で値を設定するデータ領域

データ名 A

同期型メッセージの受信を示す要求コード「VALUE 'RECVSYNC」を設定します。

データ名 C

「VALUE 'FRST」を設定します。

データ名 D

空白を設定します。

データ名 G

セグメントを受信する領域の長さを設定します。

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

空白を設定します。

データ名 M4

0を設定します。

データ名 M5

CBLDCMCF('RECVSYNC') を呼び出してから終了するまでの, 監視時間を設定します。
 0を設定した場合は, MCF マネージャ定義の UAP 共通定義 (mcfmuap -t) で指定した同期受信監視時間が設定されます。

負の値を設定した場合は、時間監視をしません。

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指定します。

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

入力元の論理端末名称を設定します。論理端末名称は最大 8 バイトの長さです。8 バイトに満たない名称を設定する場合は、後ろを空白で埋めてください。

データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 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。

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

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 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は 32763 バイトです。

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。
71001	メッセージのセグメントを受け取ったあとで、次のメッセージ (セグメント) を受信する CBLDCMCF('RECVSYNC') を呼び出しています。 直前に呼び出した CBLDCMCF('RECVSYNC') でセグメントは受信済みです。 このステータスコード値を受け取ったあとに、再び CBLDCMCF('RECVSYNC') を呼び出した場合は、ステータスコード 72000 が返ります。
71002	MCF が終了処理中のため、CBLDCMCF('RECVSYNC') を受け付けられません。
71108	メッセージ受信に必要な管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
72000	CBLDCMCF('RECVSYNC') を呼び出した位置が間違っています。MHP でセグメントを受信する CBLDCMCF('RECEIVE') を呼び出す前に、CBLDCMCF('RECVSYNC') を呼び出しています。 ステータスコード 71001 を受け取ったあとに、再び CBLDCMCF('RECVSYNC') を呼び出しています。
72001	設定した論理端末名称は、MCF で定義していません。 設定した論理端末名称が間違っています。 CBLDCMCF('RECVSYNC') を呼び出せない論理端末を設定しています。
72013	データ名 G の指定値を超えるメッセージ (セグメント) を受信しました。 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。

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

ステータスコード	意味
	<バッファ形式 2 の場合> 32763 バイトを超えるメッセージ (セグメント) を受信しました。 32763 バイトを超えた部分は切り捨てられました。
72016	データ名 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 Q に設定した値が間違っています。 データ名 M7 に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72025	データ名 C に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
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設定してください。
73001	入力元の論理端末で、障害が発生しました。
73005	同期受信処理中、設定した監視時間を過ぎているのに、論理端末からのメッセージが受信されません。
73010	入力、および出力メッセージ編集 UOC で障害が発生しました。
73015	入力元の論理端末は、ほかの UAP で仕掛り中です。
73018	データ名 M5 に設定した監視時間の値が間違っています。
73020	CBLDCMCF('RECVSYNC') で指定した論理端末は停止しています。
77350	サービスプリミティブの発行順が間違っています。 サービスプリミティブの連結数が超過しています。
77433	相手からのメッセージによって、ダイアログが未確立になりました。
77436	相手からのメッセージによって、ダイアログを解放しています。
77437	相手からのメッセージによって、ハンドシェイクが応答待ちになりました。
77491	相手からのメッセージによって、エラー処理中です。
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('SENDRECV') - 同期型メッセージの送受信 (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 一意名4
```

DATA DIVISION の指定

```
01 一意名1.
02 データ名A    PIC X(8)  VALUE 'SENDRECV'.
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.
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)  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.
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 'IO '.
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).
01 一意名4.
02 データ名X    PIC 9(x)  COMP.
02 データ名Y1   PIC X(x)  VALUE SPACE.
02 データ名Y2   PIC X(1).
02 データ名Z    PIC X(n).
```

機能

同期型でメッセージを送信したあと、同期型でメッセージを受信します。送信するメッ

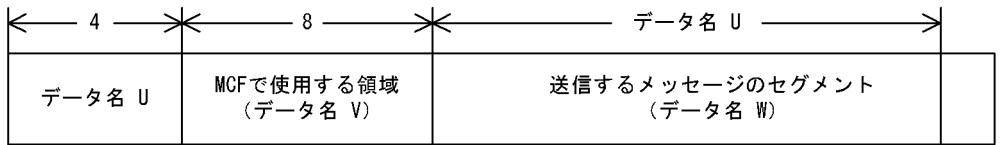
セージも受信するメッセージも、一つのセグメントで構成されます。

メッセージを受信すると、CBLDCMCF('SENDREC') は相手システムからの応答を待ちます。応答が届くと、そのメッセージを受信します。

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

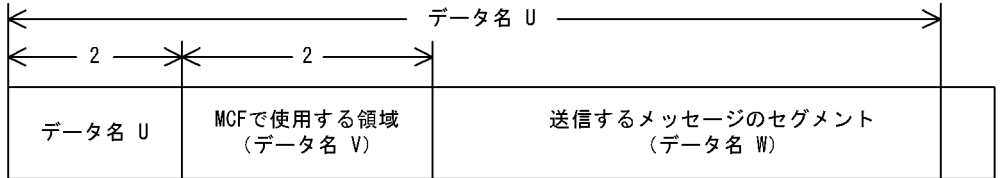
●バッファ形式 1 の場合

(単位: バイト)



●バッファ形式 2 の場合

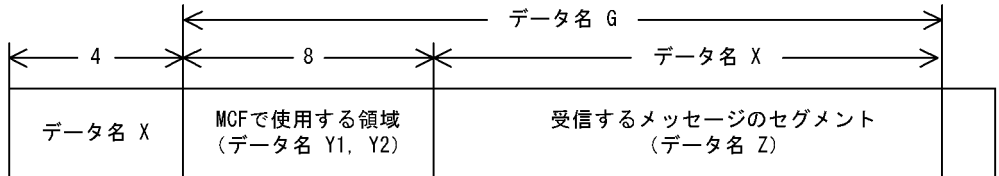
(単位: バイト)



セグメントを受信する領域 (一意名 4 で示す領域) の形式を次に示します。

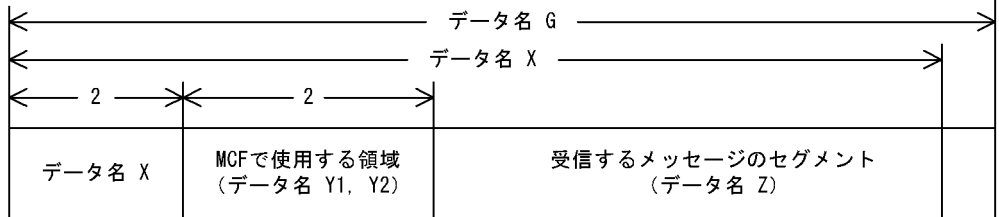
●バッファ形式 1 の場合

(単位: バイト)



●バッファ形式 2 の場合

(単位: バイト)



UAP で値を設定するデータ領域

データ名 A

同期型メッセージの送受信を示す要求コード「VALUE 'SENDREC'」を設定します。

3. メッセージ送受信インタフェース

CBLDCMCF(SENDRECV) - 同期型メッセージの送受信 (COBOL 言語)

データ名 C, データ名 D

空白を設定します。

データ名 G

セグメントを受信する領域の長さを設定します。

データ名 H

単一セグメントを示す「VALUE 'EMI '」を設定します。

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

0 を設定します。

データ名 M5

CBLDCMCF(SENDRECV) を呼び出してから終了するまでの最大時間を設定します。0 を設定した場合, MCF マネージャ定義の UAP 共通定義で指定した, 同期送受信監視時間 (mcfmuap -t sndrevtim) が設定されます。

負の値を設定した場合は, 時間を監視しません。

データ名 M6

空白を指定します。

データ名 M7

使用するバッファ形式を指定します。

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

同期型メッセージの送受信を示す「VALUE 'O 」を設定します。

データ名 P

メッセージを出力して応答を入力する論理端末名称を設定します。論理端末名称は最大 8 バイトの長さです。8 バイトに満たない名称を設定する場合は、後ろを空白で埋めてください。

データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 U

【バッファ形式 1 の場合】PIC 9(9)

送信するメッセージのセグメントの長さを設定します。

【バッファ形式 2 の場合】PIC 9(4)

送信するメッセージのセグメントの長さ + 4 を設定します。

データ名 V

【バッファ形式 1 の場合】PIC X(8)

【バッファ形式 2 の場合】PIC X(2)

MCF で使用する領域です。

データ名 W

送信メッセージのセグメントを設定します。

データ名 Y1

【バッファ形式 1 の場合】PIC X(7)

【バッファ形式 2 の場合】PIC X(1)

空白を設定します。

データ名 Y2

MCF で使用する領域です。

3. メッセージ送受信インタフェース

CBLDCMCF('SENDRECV') - 同期型メッセージの送受信 (COBOL 言語)

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 X

【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 Z

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は 32763 バイトです。

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入出力処理時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、メッセージの送受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	先頭セグメントを受信する CBLDCMCF('RECEIVE ') を呼び出す前に、CBLDCMCF('SENDRECV') を呼び出しています。

3. メッセージ送受信インタフェース
CBLDCMCF('SENDRECV') - 同期型メッセージの送受信 (COBOL 言語)

ステータスコード	意味
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('SENDRECV') を呼び出せない論理端末を設定しています。
72013	データ名 G の指定値を超えるセグメントを受信しました。データ名 G の指定値を超えた部分は切り捨てられました。 <バッファ形式 2 の場合> 32763 バイトを超えるメッセージ (セグメント) を受信しました。 32763 バイトを超えた部分は切り捨てられました。
72016	データ名 M4 に設定した値が間違っています。 データ名 M7 に設定した値が間違っています。 データ名 N またはデータ名 T に設定した値が間違っています。
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。
72026	データ名 H に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
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設定してください。
72041	バッファ形式 1 の場合はデータ名 U に 0 以下の値を設定しています。バッファ形式 2 の場合はデータ名 U に 4 以下の値を設定しています。
73001	出力先の論理端末で障害が発生しました。
73005	データ名 M5 に設定した時間が経過しましたが、論理端末からの応答がありません。
73010	入力または出力メッセージ編集 UOC で障害が発生しました。 メッセージの読み込み時に障害が発生しました。 メッセージの編集エラーが発生しました。
73015	出力先の論理端末は、ほかの UAP で仕掛り中です。
73018	データ名 M5 に設定した値が間違っています。
73020	出力先の論理端末は停止しています。
77001	指定したコネクショングループ内に使用できるコネクションがありません。
77301	送信セグメント長と、「制御情報 + パラメタ + ユーザデータ」のサイズが一致していません。
77302	制御情報のサイズの設定が間違っています。
77303	転送手順の設定が間違っています。
77304	サービスプリミティブコードの設定が間違っています。
77310	パラメタ情報サイズの設定が間違っています。
77311	Initiating-TPSU-Title に設定した値が間違っています。
77312	Recipient-AP-Title 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

CBLDRCMF('SENDRECV') - 同期型メッセージの送受信 (COBOL 言語)

ステータスコード	意味
77313	Recipient-TPSU-Title に設定した値が間違っています。
77314	Functional-Units に設定した値が間違っています。
77315	Quality-of-Service に設定した値が間違っています。
77316	Application-Context-Name に設定した値が間違っています。
77317	Confirmation に設定した値が間違っています。
77318	Result に設定した値が間違っています。
77321	Confirmation-Urgency に設定した値が間違っています。
77330	ユーザデータのサイズが間違っています。
77331	APDU の連結状態が間違っています。
77332	ユーザデータのサイズが設定されていません。
77333	ユーザデータが設定できません。
77350	サービスプリミティブの発行順が間違っています。
	サービスプリミティブの連結数が超過しています。
77433	相手からのメッセージによって、ダイアログが未確立になりました。
77436	相手からのメッセージによって、ダイアログを解放しています。
77437	相手からのメッセージによって、ハンドシェイクが応答待ちになりました。
77491	相手からのメッセージによって、エラー処理中です。
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('SENDSYNC') - 同期型メッセージの送信 (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 'SENDSYNC'.  
  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)  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.  
  02 データ名M6     PIC X(1)  VALUE SPACE.  
  02 データ名M7     PIC X(1).  
  02 データ名N      PIC X(14) VALUE LOW-VALUE.  
01 一意名2.  
  02 データ名O      PIC X(4)  VALUE SPACE.  
  02 データ名P      PIC X(8).  
  02 データ名Q      PIC X(8)  VALUE SPACE.  
  02 データ名R      PIC X(8)  VALUE SPACE.  
  02 データ名T      PIC X(28) VALUE LOW-VALUE.  
01 一意名3.  
  02 データ名U      PIC 9(x)  COMP.  
  02 データ名V      PIC X(x).  
  02 データ名W      PIC X(n).
```

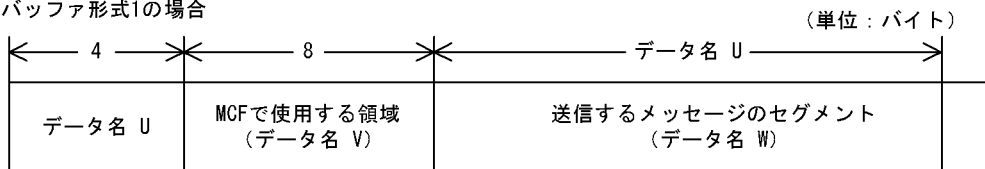
機能

相手システムへ同期型でメッセージを送信します。メッセージは、一つのセグメントで構成されます。

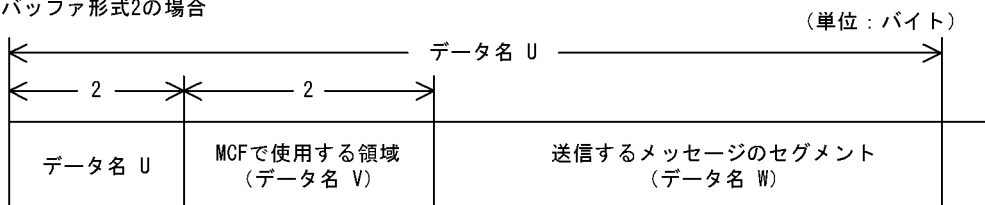
セグメントを送信する領域 (一意名3 で示す領域) の形式を次に示します。

3. メッセージ送受信インタフェース
 CBLDCMCF('SENDSYNC') - 同期型メッセージの送信 (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

同期型メッセージの送信を示す要求コード「VALUE 'SENDSYNC'」を設定します。

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H

単一セグメントの送信を示す「VALUE 'EMI '」を設定します。

データ名 I, 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

0 を設定します。

データ名 M5

CBLDCMCF('SENDSYNC') を呼び出しから終了するまでの監視時間を設定します。0 を設定した場合は, MCF マネージャ定義の UAP 共通定義 (mcfmuap -t) で指定した同期送信監視時間が設定されます。

負の値を設定した場合は、時間監視をしません。

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指定します。

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

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

データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 U

【バッファ形式 1 の場合】PIC 9(9)

送信するセグメントの長さを設定します。

【バッファ形式 2 の場合】PIC9(4)

送信するメッセージのセグメントの長さ + 4 を設定します。

データ名 V

【バッファ形式 1 の場合】PICX(8)

3. メッセージ送受信インタフェース
 CBLDCMCF('SENDSYNC') - 同期型メッセージの送信 (COBOL 言語)

【バッファ形式 2 の場合】PICX(2)

MCF で使用する領域です。

データ名 W

送信するメッセージのセグメントを設定します。

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	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	先頭セグメントを受信する CBLDCMCF('RECEIVE ') を呼び出す前に、CBLDCMCF('SENDSYNC') 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。 データ名 P に設定したメッセージ出力先の論理端末名称は、MCF で定義していません。 CBLDCMCF('SENDSYNC') を呼び出せない論理端末を設定しています。
72016	データ名 M1 に設定した値が間違っています。 データ名 M7 に設定した値が間違っています。 データ名 N またはデータ名 T に設定した値が間違っています。
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。
72026	データ名 H に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72041	バッファ形式 1 の場合はデータ名 U に 0、またはマイナスの値を設定していません。バッファ形式 2 の場合はデータ名 U に 4 以下、またはマイナスの値を設定していません。
73001	出力先の論理端末で障害が発生しました。

3. メッセージ送受信インタフェース
CBLDCMCF('SENDSYNC') - 同期型メッセージの送信 (COBOL 言語)

ステータスコード	意味
73002	同期送信時、メッセージ送信先の論理端末と TP1/NET/OSI-TP との間のコネクションが切断されています。または、CBLDCMCF('SENDSYNC') が CBLDCMCF('RECVSYNC') を呼び出してから現在の送信要求を呼び出すまでの間に、メッセージ送信先の論理端末と TP1/NET/OSI-TP の間のコネクションが切断されました。
73005	データ名 M5 に設定した時間が経過しましたが、論理端末からの応答がありません。
73010	出力メッセージ編集 UOC で障害が発生しました。
	メッセージの読み込み時に障害が発生しました。
73015	出力先の論理端末は、ほかの UAP で仕掛けり中です。
73018	データ名 M5 に設定した値が間違っています。
73020	出力先の論理端末は停止しています。
77001	指定したコネクショングループ内に使用できるコネクションがありません。
77301	送信セグメント長と、「制御情報 + パラメタ + ユーザデータ」のサイズが一致していません。
77302	制御情報のサイズの設定が間違っています。
77303	転送手順の設定が間違っています。
77304	サービスプリミティブコードの設定が間違っています。
77310	パラメタ情報サイズの設定が間違っています。
77311	Initiating-TPSU-Title に設定した値が間違っています。
77312	Recipient-AP-Title に設定した値が間違っています。
77313	Recipient-TPSU-Title に設定した値が間違っています。
77314	Functional-Units に設定した値が間違っています。
77315	Quality-of-Service に設定した値が間違っています。
77316	Application-Context-Name に設定した値が間違っています。
77317	Confirmation に設定した値が間違っています。
77318	Result に設定した値が間違っています。
77321	Confirmation-Urgency に設定した値が間違っています。
77330	ユーザデータのサイズの設定が間違っています。
77331	APDU の連結状態の設定が間違っています。
77332	ユーザデータのサイズが設定されていません。
77333	ユーザデータが設定できません。
77350	サービスプリミティブの発行順が間違っています。
	サービスプリミティブの連結数が超過しています。
77433	相手からのメッセージによって、ダイアログが未確立になりました。
77436	相手からのメッセージによって、ダイアログを解放しています。

3. メッセージ送受信インタフェース

CBLDCMCF('SENDSYNC') - 同期型メッセージの送信 (COBOL 言語)

ステータスコード	意味
77437	相手からのメッセージによって、ハンドシェイクが応答待ちになりました。
77491	相手からのメッセージによって、エラー処理中です。
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。

DISABLE - 同期型メッセージの送信（データ操作言語）

形式

DATA DIVISION（通信記述項）の指定

```
CD 通信記述名  
FOR I-O  
  [STATUS KEY IS データ名1]  
SYMBOLIC TERMINAL IS データ名2.
```

PROCEDURE DIVISION（通信文）の指定

```
DISABLE 通信記述名 WITH 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。

- 同期型メッセージの送信 CBLDCMCF('SENDSYNC')

通信記述項に設定する項目

FOR 句

次の値を指定します。

I-O

同期型メッセージの送信

STATUS KEY 句

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

SYMBOLIC TERMINAL 句

論理端末名称を設定したデータ項目を指定します。

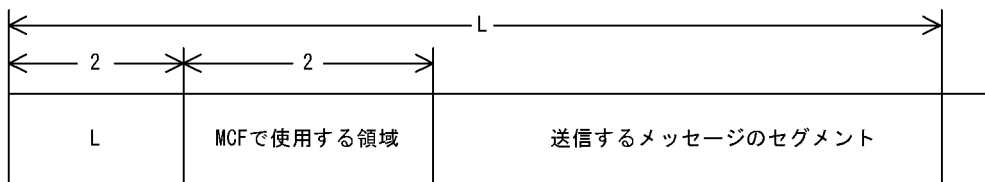
通信文に指定する項目

WITH 句

同期型メッセージのセグメント送信領域を示すデータ項目を指定します。送信するセグメントの形式（一意名1）を次に示します。

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

（単位：バイト）



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	メッセージセグメント長に 32000 バイトを超える値を指定しています。 MCF が終了処理中のため、DISABLE 文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。
71108	メッセージ送信に必要な管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
72000	DISABLE 文を実行した位置が間違っています。 MHP の先頭セグメントを受信する RECEIVE 文を実行する前に、DISABLE 文を実行しています。
72001	論理端末名称が間違っています。 設定した論理端末名称は、MCF で定義していません。
72014	ダイアログ開始を示す ENABLE 文が実行されていません。
72024	FOR 句に指定した値が間違っています。
72041	メッセージの送信に関する実行の組み合わせが間違っています。 送信セグメント長に、4 以下の値を指定しています。
73001	出力先の論理端末で障害が発生しました。
73005	監視時間を過ぎているのに、同期送信要求が完了しません。
73010	入力、および出力メッセージ編集 UOC で障害が発生しました。
73015	出力先論理端末は、すでにほかの UAP でダイアログを行っています。
73020	出力先論理端末が停止しています。
77001	指定したコネクショングループ内に使用できるコネクションがありません。
77301	送信メッセージ長と、「制御情報 + パラメタ + ユーザ情報」のサイズの合計とが一致していません。
77302	制御情報サイズの設定が間違っています。
77303	転送手順の設定が間違っています。
77304	サービスプリミティブの種別コードの設定が間違っています。
77310	パラメタサイズの設定が間違っています。

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

ステータスコード	意味
77317	Confirmation に設定した値が間違っています。
77330	ユーザデータのサイズの設定が間違っています。
77331	APDU の連結状態の設定が間違っています。
77332	ユーザデータがありません。
77333	ユーザデータが設定できません。
77350	サービスプリミティブの発行順が間違っています。
	サービスプリミティブの連結数が超過しています。
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。

ENABLE - 同期型メッセージの送信 (データ操作言語)

形式

DATA DIVISION (通信記述項) の指定

```
CD 通信記述名  
FOR I-O  
  {STATUS KEY IS データ名1}  
SYMBOLIC TERMINAL IS データ名2  
SYNCHRONOUS MODE IS {SYNC|データ名3}.
```

PROCEDURE DIVISION (通信文) の指定

```
ENABLE 通信記述名 WITH 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。

- 同期型メッセージの送信 CBLDCMCF('SENDSYNC')

通信記述項に設定する項目

FOR 句

次の値を指定します。

I-O

同期型メッセージの送信

STATUS KEY 句

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

SYMBOLIC TERMINAL 句

論理端末名称を設定したデータ項目を指定します。

SYNCHRONOUS MODE 句

次のどちらかの値を指定します。

SYNC

メッセージの連結をしません。

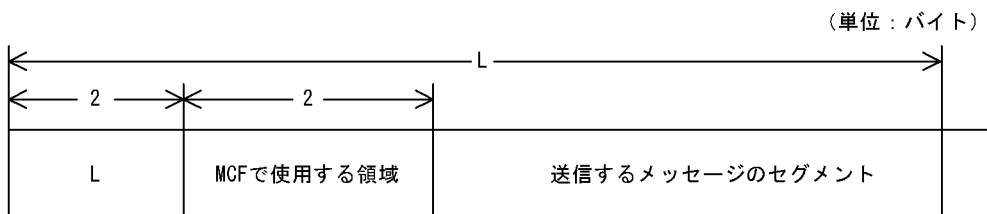
データ名 3

次の値を設定したデータ項目
'1'...メッセージの連結をしません。

通信文に指定する項目

WITH 句

同期型メッセージのセグメント送信領域を示すデータ項目を指定します。送信するセグメントの形式（一意名 1）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。
71002	メッセージセグメント長に 32000 バイトを超える値を指定しています。 MCF が終了処理中のため、ENABLE 文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。
71108	メッセージ送信に必要な管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
72000	ENABLE 文を実行した位置が間違っています。 MHP の先頭セグメントを受信する RECEIVE 文を実行する前に、ENABLE 文を実行しています。
72001	論理端末名称が間違っています。 設定した論理端末名称は、MCF で定義していません。
72020	SYNCHRONOUS MODE 句に指定した値が間違っています。
72024	FOR 句に指定した値が間違っています。
72041	メッセージの送信に関するの実行の組み合わせが間違っています。 送信セグメント長に、4 以下の値を指定しています。
73001	出力先の論理端末で障害が発生しました。
73005	監視時間を過ぎているのに、同期送信要求が完了しません。
73010	入力、および出力メッセージ編集 UOC で障害が発生しました。
73015	出力先論理端末は、すでにほかの UAP でダイアログを行っています。

3. メッセージ送受信インタフェース

ENABLE - 同期型メッセージの送信 (データ操作言語)

ステータスコード	意味
73016	TP1/NET/OSI-TP がメッセージを送受信できない端末に対して、サービスプリミティブの送信をしました。
73020	出力先論理端末が停止しています。
77001	指定したコネクショングループ内に使用できるコネクションがありません。
77301	送信メッセージ長と、「制御情報 + パラメタ + ユーザ情報」のサイズの合計とが一致していません。
77302	制御情報サイズが間違っています。
77303	転送手順が間違っています。
77304	サービスプリミティブの種別コードが間違っています。
77310	パラメタサイズが間違っています。
77311	Initiating-TPSU-Title に設定した値が間違っています。
77312	Recipient-AP-Title に設定した値が間違っています。
77313	Recipient-TPSU-Title に設定した値が間違っています。
77314	Functional-Units に設定した値が間違っています。
77315	Quality-of-Service に設定した値が間違っています。
77316	Application-Context-Name に設定した値が間違っています。
77317	Confirmation に設定した値が間違っています。
77318	Result に設定した値が間違っています。
77330	ユーザデータのサイズが間違っています。
77331	APDU の連結状態が間違っています。
77332	ユーザデータがありません。
77333	ユーザデータが設定できません。
77350	サービスプリミティブの発行順が間違っています。
	サービスプリミティブの連結が超過しています。
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|I-O}
   [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]
   [SYNCHRONOUS MODE IS {SYNC|データ名6}]
   [WAITING TIME IS データ名11].
```

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')
- 同期型メッセージの受信 CBLDCMCF('RECVSYNC')

通信記述項に設定する項目

FOR 句

次のどちらかの値を指定します。

INPUT

一方送信メッセージの受信

I-O

同期型メッセージの受信

STATUS KEY 句

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

SYMBOLIC TERMINAL 句

入力元の論理端末名称を参照するデータ項目を指定します。

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）

MESSAGE DATE 句

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

MESSAGE TIME 句

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

SYNCHRONOUS MODE 句

同期型メッセージの受信の場合に指定します。次のどちらかの値を指定します。

SYNC

同期型メッセージの受信

データ名 6

次の値を設定したデータ項目

'1'...同期型メッセージの受信

WAITING TIME 句

同期型メッセージを受信する場合の、監視時間の値を設定したデータ項目を指定します。

データ名 11

監視時間を HHMMSS00（HH：時 MM：分 SS：秒 00は固定）の形式で指定します。指定できる値の範囲は10秒～10分で、10秒単位です。

省略した場合は、2分を設定します。

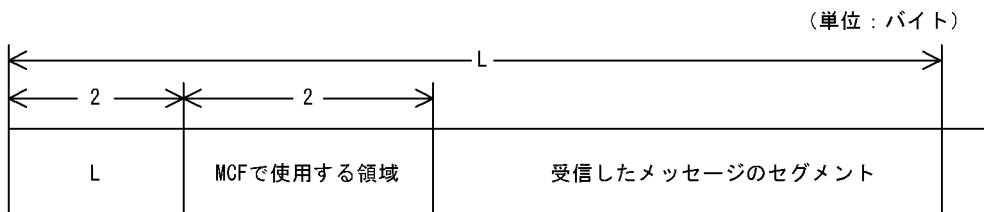
通信文に指定する項目

FIRST

先頭セグメントを受信する場合に指定します。

一意名 1

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



ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する RECEIVE 文を、2 回以上実行しています。中間セグメントまたは最終セグメントを受信する場合は、FIRST を設定しないで RECEIVE 文を実行してください。
71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する RECEIVE 文を実行しています。直前に実行した RECEIVE 文でメッセージはすべて受信しました。 このステータスコードが返されたあとに、再び RECEIVE 文を実行した場合は、ステータスコード 72000 が返されます。
71002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。 メッセージキューが割り当てられていません。 MCF が終了処理中のため、メッセージの受信を受け付けられません。
71108	メッセージ受信に必要な管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > <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 71001 が返されたあとで、RECEIVE 文を実行しています。 < SPP の実行でリターンした場合 > SPP では RECEIVE 文を実行できません。
72001	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。 RECEIVE 文を実行できない論理端末を設定しています。
72013	一意名 1 の L の指定値を超えるメッセージ (セグメント) を受信しました。一意名 1 の L の指定値を超える部分は切り捨てられました。 32763 バイトを超えるメッセージ (セグメント) を受信しました。32763 バイトを超えた部分は切り捨てられました。
72016	WAITING 句に設定した値が間違っています。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72036	一意名 1 の L が不足しています。5 バイト以上の領域を確保してください。
73001	入力元の論理端末で障害が発生しました。
73005	RECEIVE 文が、設定した監視時間を過ぎても終了しません。
73010	入力、または出力メッセージの編集 UOC でエラーが発生しました。 メッセージの読み込み中に障害が発生しました。

3. メッセージ送受信インタフェース

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
73018	RECEIVE 文を実行した場合に、データ名 11 に設定した監視時間が間違っています。
73020	RECEIVE 文で指定した論理端末は停止中です。
77350	サービスプリミティブの発行順が間違っています。 サービスプリミティブの連結数が超過しています。
77433	相手からのメッセージによって、ダイアログが未確立になりました。
77436	相手からのメッセージによって、ダイアログを解放中です。
77437	相手からのメッセージによって、ハンドシェイクが応答待ちになりました。
77491	相手からのメッセージによって、エラー処理中です。
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。

SEND - 同期型メッセージの送受信（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 I-O
   {STATUS KEY IS データ名1}
   {SYMBOLIC TERMINAL IS データ名2}
   {SYNCHRONOUS MODE IS {SYNC|データ名6}}
   {WAITING TIME IS データ名11}.
```

PROCEDURE DIVISION（通信文）の指定

```
SEND 通信記述名 FROM 一意名1
     [BEFORE RECEIVING MESSAGE INTO 一意名3].
```

機能

次に示す CALL インタフェースの機能を実現します。

- 同期型メッセージの送信 CBLDCMCF('SENDSYNC')
- 同期型メッセージの送受信 CBLDCMCF('SENDRECV')

通信記述項に設定する項目

FOR 句

次の値を指定します。

I-O

同期型メッセージの送信，または同期型メッセージの送受信

STATUS KEY 句

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

SYMBOLIC TERMINAL 句

論理端末名称を設定したデータ項目を指定します。

SYNCHRONOUS MODE 句

次のどちらかの値を指定します。

SYNC

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

同期型メッセージの送信

データ名 6

次の値を設定したデータ項目
 '1'...同期型メッセージの送信

WAITING TIME 句

同期型メッセージ送信または、同期型メッセージ送受信の場合の、限界経過時間の値を指定します。

データ名 11

限界経過時間の値を、HHMMSS00 (HH:時 MM:分 SS:秒 00は固定)の形式で指定します。

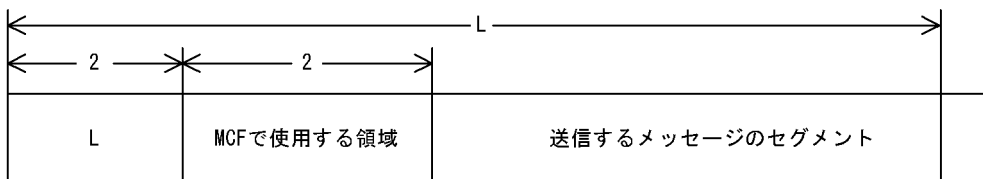
この句を省略、またはデータ名 11 に '00000000' を設定した場合は、MCF マネジャ定義の UAP 共通定義 (mcfmuap -t) で指定した同期受信監視時間を設定します。

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。

(単位: バイト)



BEFORE 句

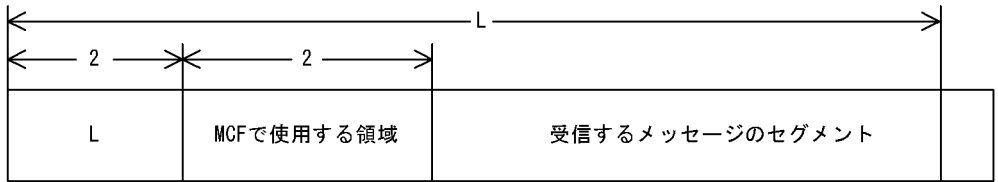
同期型メッセージ送受信の場合に設定します。メッセージ送信後にその端末から受信したメッセージを受け取る場合に指定します。

一意名 3

受信メッセージを格納するデータ項目

同期型メッセージを送受信する場合の、受信するセグメントの領域 (一意名 3) の形式を次に示します。

(単位: バイト)



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	FOR 句に I-O を設定していますが、SYNCHRONOUS MODE 句で SYNC を設定していません。 < MHP の実行でリターンした場合 > 先頭セグメントを受信する RECEIVE 文を実行する前に、SEND 文を実行しています。 < SPP の実行でリターンした場合 > トランザクションでない SPP の処理から、SEND 文を実行しています。
72001	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。 SEND 文を実行できない論理端末を設定しています。
72013	一意名 1 の L の指定値を超えるセグメントを受信しました。 一意名 1 の L の指定値を超えた部分は切り捨てられました。 32763 バイトを超えるセグメントを受信しました。 32763 バイトを超えた部分は切り捨てられました。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72036	一意名 3 のセグメントを受信する領域の長さに指定した受信領域長が不足しています。5 バイト以上の領域を確保してください。
72037	BEFORE 句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

SEND - 同期型メッセージの送受信（データ操作言語）

ステータスコード	意味
72041	一意名 1 の L に 4 以下, またはマイナスの値を設定しています。
72045	NEXT TRANSACTION 句は設定できません。
72047	NEXT TRANSACTION 句は設定できません。
73001	出力先の論理端末で障害が発生しました。
73005	WAITING TIME 句に設定した時間が経過しましたが, 論理端末からの応答がありません。
73010	入力または出力メッセージ編集 UOC で障害が発生しました。 メッセージの読み込み時に障害が発生しました。
73015	出力先の論理端末は, ほかの UAP で仕掛けり中です。
73018	WAITING TIME 句に設定した値が間違っています。
73020	出力先の論理端末は停止しています。
77001	指定したコネクショングループ内に使用できるコネクションがありません。
77301	送信メッセージ長と「制御情報 + パラメタ + ユーザデータ」のサイズが一致していません。
77302	制御情報のサイズの設定が間違っています。
77303	転送手順の設定が間違っています。
77304	サービスプリミティブコードの設定が間違っています。
77310	パラメタ情報サイズの設定が間違っています。
77311	Initiating-TPSU-Title に設定した値が間違っています。
77312	Recipient-AP-Title に設定した値が間違っています。
77313	Recipient-TPSU-Title に設定した値が間違っています。
77314	Functional-Units に設定した値が間違っています。
77315	Quality-of-Service に設定した値が間違っています。
77316	Application-Context-Name に設定した値が間違っています。
77317	Confirmation に設定した値が間違っています。
77318	Result に設定した値が間違っています。
77321	Confirmation-Urgency に設定した値が間違っています。
77330	ユーザデータのサイズが間違っています。
77331	APDU の連結状態が間違っています。
77332	ユーザデータのサイズが設定されていません。
77333	ユーザデータが設定できません。
77350	サービスプリミティブの発行順が間違っています。 サービスプリミティブの連結数が超過しています。
77433	相手からのメッセージによって, ダイアログが未確立になりました。
77436	相手からのメッセージによって, ダイアログを解放しています。

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

ステータスコード	意味
77437	相手からのメッセージによって、ハンドシェイクが応答待ちになりました。
77491	相手からのメッセージによって、エラー処理中です。
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。

サービスプリミティブ

TP1/NET/OSI-TP が提供するサービスプリミティブのインタフェースについて説明します。

サービスプリミティブとは、OSI が規定する OSI 参照モデルで、それぞれの階層が利用するサービスです。TP1/NET/OSI-TP では、サービスプリミティブを利用して、メッセージの送受信をします。

サービスプリミティブは、UAP のコーディング時に設定します。このとき、まず UAP の引数を設定してから、使用するサービスプリミティブの領域と具体的な指定値を設定します。また、サービスプリミティブは複数連結して送信できます。

TP1/NET/OSI-TP がメッセージ送受信で提供する基本サービスプリミティブを表 3-4 に、サービスプリミティブと UAP との対応を表 3-5 に、サービスプリミティブの連結パターンと UAP との対応を表 3-6 に示します。

表 3-4 基本サービスプリミティブ

サービスプリミティブ	機能	全二重	半二重			
			制御権あり		制御権なし	
			送信	受信	送信	受信
TP-BEGIN-DIALOGUEreq/ind	ダイアログ開始要求 / 指示	-	-	-	-	-
TP-BEGIN-DIALOGUErsp/cnf	ダイアログ開始応答 / 確認					
TP-END-DIALOGUEreq/ind	ダイアログ終了要求 / 指示			×	×	
TP-END-DIALOGUErsp/cnf	ダイアログ終了応答 / 確認		×			×
TP-U-ERRORreq/ind	エラー発生のお知らせ要求 / 指示 <ul style="list-style-type: none"> • TP-HANDSHAKEreq 確認型 • TP-END-DIALOGUEreq • TP-HANDSHAKE-AND-GRANT-CONTROLreq に対する拒否応答 					
TP-U-ABORTreq/ind	UAP でのダイアログ異常終了要求 / 指示					
TP-P-ABORTind	TP1/NET/OSI-TP でのダイアログ異常終了指示					
TP-HANDSHAKEreq/ind	ハンドシェイク要求 / 指示			×	×	
TP-HANDSHAKErsp/cnf	ハンドシェイク応答 / 確認		×			×

サービスプリミティブ	機能	全二重	半二重			
			制御権あり		制御権なし	
			送信	受信	送信	受信
TP-DATAreq/ind	データ転送要求 / 指示			×	×	
TP-GRANT-CONTROLreq/ind	制御権委譲要求 / 指示	×		×	×	
TP-REQUEST-CONTROLreq/ind	制御権要求要求 / 指示	×	×			×
TP-HANDSHAKE-AND-GRANT-CONTROLreq/ind	制御権委譲付きハンドシェイク要求 / 指示	×		×	×	
TP-HANDSHAKE-AND-GRANT-CONTROLrsp/cnf	制御権委譲付きハンドシェイク応答 / 確認	×	×			×

(凡例)

- : 使用できます。
- × : 使用できません。
- : 該当しません。

表 3-5 サービスプリミティブと UAP との対応

サービス	プリミティブ	UAP			サービスプリミティブ種別コード
		C 言語	COBOL 言語 ¹	データ操作言語	
TP-BEGIN-DIALOGUE	req	• sendsync • sendrecv	• SENDSYNC • SENDRECV	ENABLE	BDR
	ind	receive	RECEIVE	RECEIVE	BDI
	rsp (受諾)	• sendsync • sendrecv	• SENDSYNC • SENDRECV	ENABLE	BDPA
	rsp (拒否)	sendsync	SENDSYNC	DISABLE	BDPN
	cnf	• recvsync • sendrecv	• RECVSYN • SENDRECV	RECEIVE	BDC
TP-END-DIALOGUE	req	• sendsync • sendrecv ₂	• SENDSYNC • SENDRECV ₂	DISABLE	EDR
	ind	• recvsync • sendrecv	• RECVSYN • SENDRECV	RECEIVE	EDI
	rsp	• sendsync	• SENDSYNC	DISABLE	EDP

3. メッセージ送受信インタフェース
サービスプリミティブ

サービス	プリミティブ	UAP			サービス プリミ ティブ種 別コード
		C 言語	COBOL 言語 ¹	データ操作 言語	
	cnf	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	EDC
TP-U-ERROR	req	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	UER
	ind	recvsync	RECVSYNC	RECEIVE	UEI
TP-U-ABORT	req	sendsync	SENDSYNC	DISABLE	UAR
	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	UAI
TP-P-ABORT	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	PAI
TP-HANDSHAK E	req	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	HSR
	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	HSI
	rsp	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	HSP
	cnf	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	HSC
TP-DATA	req	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	TDR
	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	TDI
TP-GRANT-CON TROL	req	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	GCR
	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	GCI
TP-REQUEST-C ONTROL	req	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	RCR
	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	RCI
TP-HANDSHAK E-AND-GRANT-C ONTROL	req	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	HGR
	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	HGI

サービス	プリミティブ	UAP			サービス プリミ ティブ種 別コード
		C 言語	COBOL 言語 ¹	データ操作 言語	
	rsp	<ul style="list-style-type: none"> • sendsync • sendrecv 	<ul style="list-style-type: none"> • SENDSYNC • SENDRECV 	SEND	HGP
	cnf	<ul style="list-style-type: none"> • recvsync • sendrecv 	<ul style="list-style-type: none"> • RECVSYNC • SENDRECV 	RECEIVE	HGC

注 1

COBOL 言語のインタフェースは、プログラム名 CBLDCMCF で使用するデータ名を示しません。

注 2

サービスプリミティブのパラメタ Confirmation が 'True' の場合だけ使用できます。

表 3-6 サービスプリミティブの連結パターンと UAP との対応

サービスプリミティブ連結パターンの分類 (サービスプリミティブ連結パターン例)	UAP		
	C 言語	COBOL 言語 ¹	データ操作 言語
先頭が BDR または BDPA (BDR+TDR,BDPA+TDR など)	<ul style="list-style-type: none"> • sendsync • sendrecv 	<ul style="list-style-type: none"> • SENDSYNC • SENDRECV 	ENABLE
先頭が BDI (BDI+TDI,BDI+TDI+EDI など)	receive	RECEIVE	RECEIVE
先頭が BDR または BDPA で最後が EDR (BDPA+TDR+EDR,BDR+TDR+EDR ³ など)	<ul style="list-style-type: none"> • sendsync • sendrecv ₂ 	<ul style="list-style-type: none"> • SENDSYNC • SENDRECV ₂ 	DISABLE
最後が EDR (TDR+EDR,BDPA+EDR,HSP+EDR など)	<ul style="list-style-type: none"> • sendsync • sendrecv ₂ 	<ul style="list-style-type: none"> • SENDSYNC • SENDRECV ₂ 	DISABLE
その他の送信 (TDR+HSR,TDR+TDR など)	<ul style="list-style-type: none"> • sendsync • sendrecv 	<ul style="list-style-type: none"> • SENDSYNC • SENDRECV 	SEND
その他の受信 (TDI+EDI,BDC+TDI,BDC+EDI,BDC+TDI+EDI,TDI+TDI など)	<ul style="list-style-type: none"> • recvsync • sendrecv 	<ul style="list-style-type: none"> • RECVSYNC • SENDRECV 	RECEIVE

注 1

COBOL 言語のインタフェースは、プログラム名 CBLDCMCF で使用するデータ名を示しません。

注 2

3. メッセージ送受信インタフェース サービスプリミティブ

サービスプリミティブのパラメタ Confirmation が 'True' の場合だけ使用できます。

注 3

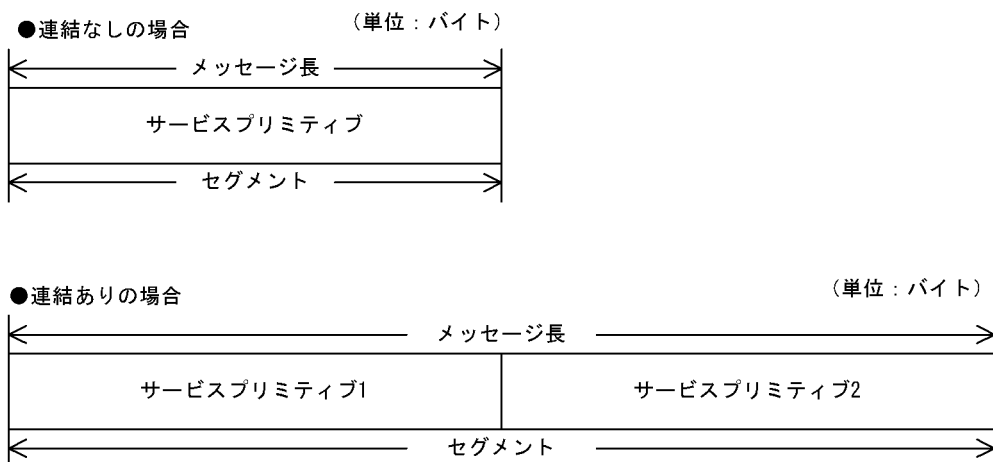
この組み合わせでは BDR は拒否応答型だけ，EDR は確認型だけ使用できます。

メッセージの形式

送受信メッセージ

TP1/NET/OSI-TP で送受信するメッセージの形式を次の図に示します。送受信メッセージは，UAP で複数の APDU を連結し，一括して処理できます。

図 3-1 送受信メッセージの形式



セグメント

セグメントは，UAP が送受信するデータ単位です。セグメントは一つ以上のサービスプリミティブから構成されます。

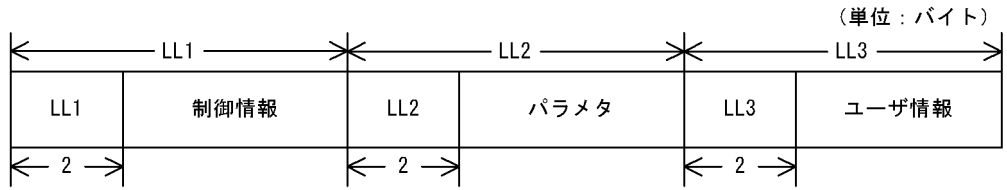
サービスプリミティブ

サービスプリミティブは，制御情報，パラメタ，およびユーザ情報から構成されます。また，サービスプリミティブは複数連結できます。連結をしない場合，セグメントとサービスプリミティブは 1 対 1 に対応し，連結をする場合は 1 対 n に対応します。

パラメタおよびユーザ情報がない場合，または指定しない場合は，LL2 または LL3 に $(0002)_{16}$ を設定します。

サービスプリミティブの形式を次の図に示します。

図 3-2 サービスプリミティブの形式

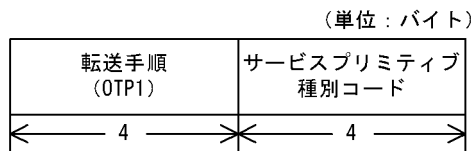


制御情報

制御情報は、転送手順とサービスプリミティブ種別コードから構成されます。サービスプリミティブ種別コードについては、表 3-5 を参照してください。

制御情報の形式を次の図に示します。

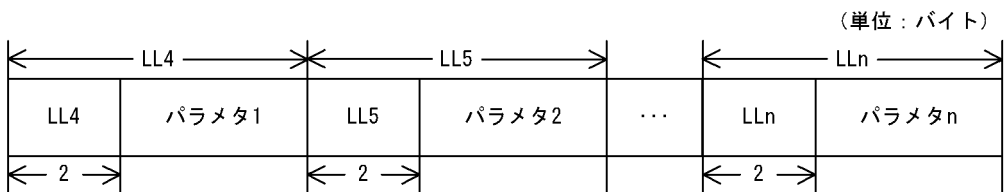
図 3-3 制御情報の形式



パラメタ

パラメタの形式を次の図に示します。該当するパラメタを指定しない場合は、そのパラメタに対応する LL に $(0002)_{16}$ を設定します。

図 3-4 パラメタの形式



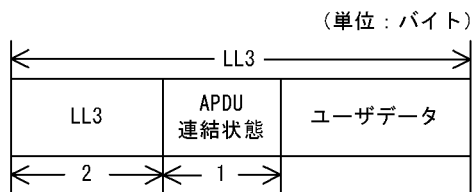
ユーザ情報

ユーザ情報は、APDU 連結状態とユーザデータから構成されます。

ユーザ情報の形式を、次の図に示します。

3. メッセージ送受信インタフェース サービスプリミティブ

図 3-5 ユーザ情報の形式



APDU 連結状態は、次に示す値を 16 進数字で設定します。

- (A0)：単一 APDU (構造化されている)
- (80)：単一 APDU (構造化されていない)
- (A1)：連結 APDU (構造化されている)
- (81)：連結 APDU (構造化されていない)
- (A2)：ビット列データ (構造化されている)
- (82)：ビット列データ (構造化されていない)

ユーザデータは、任意のデータを転送構文形式で設定します。

パラメタとの関係

サービスプリミティブは、機能ごとに、設定するパラメタが異なります。

省略した場合、省略したパラメタに対応する領域の長さは $(0002)_{16}$ となります。

サービスプリミティブとパラメタとの対応を次の表に示します。

表 3-7 サービスプリミティブとパラメタとの対応

サービス	プリミティブ	パラメタ 1	パラメタ 2	パラメタ 3	パラメタ 4	パラメタ 5	パラメタ 6	パラメタ 7	ユーザ情報の設定
TP-BEGIN-DIALOGUE	req	(1) 1	(2) 1	(3) 1	(4) 2	(5) 1	(6) 2	(7) 2	
	ind	(1) 3	(4) 2	(7) 2	-	-	-	-	
	rsp	(8) 2	-	-	-	-	-	-	
	cnf	(4) 3	(8) 2	(9) 3	(10) 2	-	-	-	
TP-END-DIALOGUE	req	(7) 2	-	-	-	-	-	-	×
	ind	(7) 2	-	-	-	-	-	-	×
	rsp	-	-	-	-	-	-	-	×
	cnf	-	-	-	-	-	-	-	×

3. メッセージ送受信インタフェース
サービスプリミティブ

サービス	プリミティブ	パラメータ1	パラメータ2	パラメータ3	パラメータ4	パラメータ5	パラメータ6	パラメータ7	ユーザ情報の設定
TP-U-ERROR	req	-	-	-	-	-	-	-	×
	ind	-	-	-	-	-	-	-	×
TP-U-ABORT	req	-	-	-	-	-	-	-	
	ind	(10) ₂	-	-	-	-	-	-	
TP-P-ABORT	ind	(9) ₂	(10) ₂	-	-	-	-	-	×
TP-HANDSHAKE	req	(11) ₃	-	-	-	-	-	-	×
	ind	-	-	-	-	-	-	-	×
	rsp	-	-	-	-	-	-	-	×
	cnf	-	-	-	-	-	-	-	×
TP-DATA	req	-	-	-	-	-	-	-	
	ind	-	-	-	-	-	-	-	
TP-GRANT-CONTROL	req	-	-	-	-	-	-	-	×
	ind	-	-	-	-	-	-	-	×
TP-REQUEST-CONTROL	req	-	-	-	-	-	-	-	×
	ind	-	-	-	-	-	-	-	×
TP-HANDSHAKE-AND-GRANT-CONTROL	req	(11) ₂	-	-	-	-	-	-	×
	ind	-	-	-	-	-	-	-	×
	rsp	-	-	-	-	-	-	-	×
	cnf	-	-	-	-	-	-	-	×

(凡例)

- (1) : Initiating-TPSU-Title
- (2) : Recipient-AP-Title
- (3) : Recipient-TPSU-Title
- (4) : Functional-Units
- (5) : Quality-of-Service
- (6) : Application-Context-Name
- (7) : Confirmation
- (8) : Result

3. メッセージ送受信インタフェース サービスプリミティブ

- (9) : Diagnostic
- (10) : Rollback
- (11) : Confirmation-Urgency
 - : 設定は必須です。
 - : 設定できます。
 - x : 設定できません。
 - : 該当しません。

注 1
ユーザが選択して設定できます。設定しない場合は、領域だけを確保してください。

注 2
必ず設定します。

注 3
条件が一致する場合だけ設定できます。

パラメタの詳細

Initiating-TPSU-Title および Recipient-TPSU-Title

型コード	TPSU-Title
------	------------

型コード

TPSU 名称の型を設定します。

61 : T61 String (T61 文字列)

PR : Printable String (印字可能文字列)

IN : Integer (整数)

TPSU-Title

TPSU 名称を設定します。名称の長さは最大 64 バイトです。

Initiating-TPSU-Title : 起動側 TPSU 名称

Recipient-TPSU-Title : 受信側 TPSU 名称

Recipient-AP-Title

Recipient-AP-Title

Recipient-AP-Title

受信側 AP 名称を設定します。TP1/NET/OSI-TP では、領域だけを設定します。パラメタの LL には $(0002)_{16}$ を設定してください。

Functional-Units

FU	予備領域
----	------

FU

使用する機能単位を 16 進数字で設定します。

(40)₁₆ : 全二重機能を使用します。

(48)₁₆ : 全二重機能およびハンドシェイク機能を使用します。

(80)₁₆ : 半二重機能を使用します。

(88)₁₆ : 半二重機能およびハンドシェイク機能を使用します。

予備領域

TP1/NET/OSI-TP が使用する領域です。1 バイトの長さの領域を設定してください。

Quality-of-Service

Quality-of-Service

Quality-of-Service

サービス品質を設定します。TP1/NET/OSI-TP では、領域だけを設定します。パラメタの LL には (0002)₁₆ を設定してください。

Application-Context-Name

Application-Context-Name

Application-Context-Name

アプリケーションコンテキスト名を設定します。TP1/NET/OSI-TP では、領域だけを設定します。パラメタの LL には (0002)₁₆ を設定してください。

Confirmation

*

Confirmation

ダイアログ開始および終了の要求に対して、応答が必要かどうかを確認するための値を設定します。

- ダイアログ開始 (TP-BEGIN-DIALOGUE) の場合
AL : すべての要求に対して、応答が必要です。

3. メッセージ送受信インタフェース サービスプリミティブ

NE：要求を拒否する場合だけ，応答が必要です。

- ダイアログ終了（TP-END-DIALOGUE）の場合

TR：応答が必要です。

FA：応答は必要ありません。

Result

*

Result

サービスの要求に対して，受け入れたかどうかの結果を示す値を設定します。

AC：要求を受け入れました。

RP：要求を拒否しました（TP1/NET/OSI-TP）。

RU：要求を拒否しました（UAP）。

Diagnostic

*

Diagnostic

診断を示す値を 16 進数字で設定します。

- ダイアログ開始（TP-BEGIN-DIALOGUE）の場合
 - (0001)₁₆：受信側 TPSU 名称が定義されていません。
 - (0002)₁₆：TPSU が利用できません（永続的）。
 - (0003)₁₆：TPSU が利用できません（一時的）。
 - (0004)₁₆：受信側 TPSU 名称の指定が必要です。
 - (0005)₁₆：機能単位が提供されていません。
 - (0006)₁₆：機能単位の組み合わせが提供されていません。
 - (0008)₁₆：その他の理由
- ダイアログ異常終了（TP-P-ABORT）の場合
 - (0001)₁₆：永続的障害が発生しました。
 - (0002)₁₆：トランザクションの開始を拒否します。
 - (0003)₁₆：一時的障害が発生しました。
 - (0004)₁₆：プロトコルエラーです。
 - (0005)₁₆：ダイアログ終了同士の衝突が発生しました。

Rollback

*

Rollback

ロールバックするかどうかを示す値を設定します。

FA：ロールバックしません。

Confirmation-Urgency

*

Confirmation-Urgency

即時確認をするかどうかを示す値を設定します。

ただし、半二重でのハンドシェイク (TP-HANDSHAKE) の場合、設定できません。

UR：即時確認をします。

NR：即時確認をしません。

突き合わせ項目

メッセージを送受信する場合、TP1/NET/OSI-TP は、相手システムから受信したサービスプリミティブを、自システムで設定したパラメタと突き合わせます。突き合わせたパラメタが一致しない場合は、該当するサービスプリミティブを処理しないで、相手システムに異常を通知します。

サービスプリミティブの突き合わせ項目を、表 3-8 および表 3-9 に示します。

表 3-8 サービスプリミティブの突き合わせ項目 (コネクション確立時)

突き合わせ項目	チェック内容
protocol-version	version1 が設定されていること。
contention-winner-assignment	MCF 通信構成定義 (mcftalccn -y) で指定した値と対応していること。

注

MCF 通信構成定義との対応については、「5. システム定義」の「相手システムの通信定義と関連づける内容」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
サービスプリミティブ

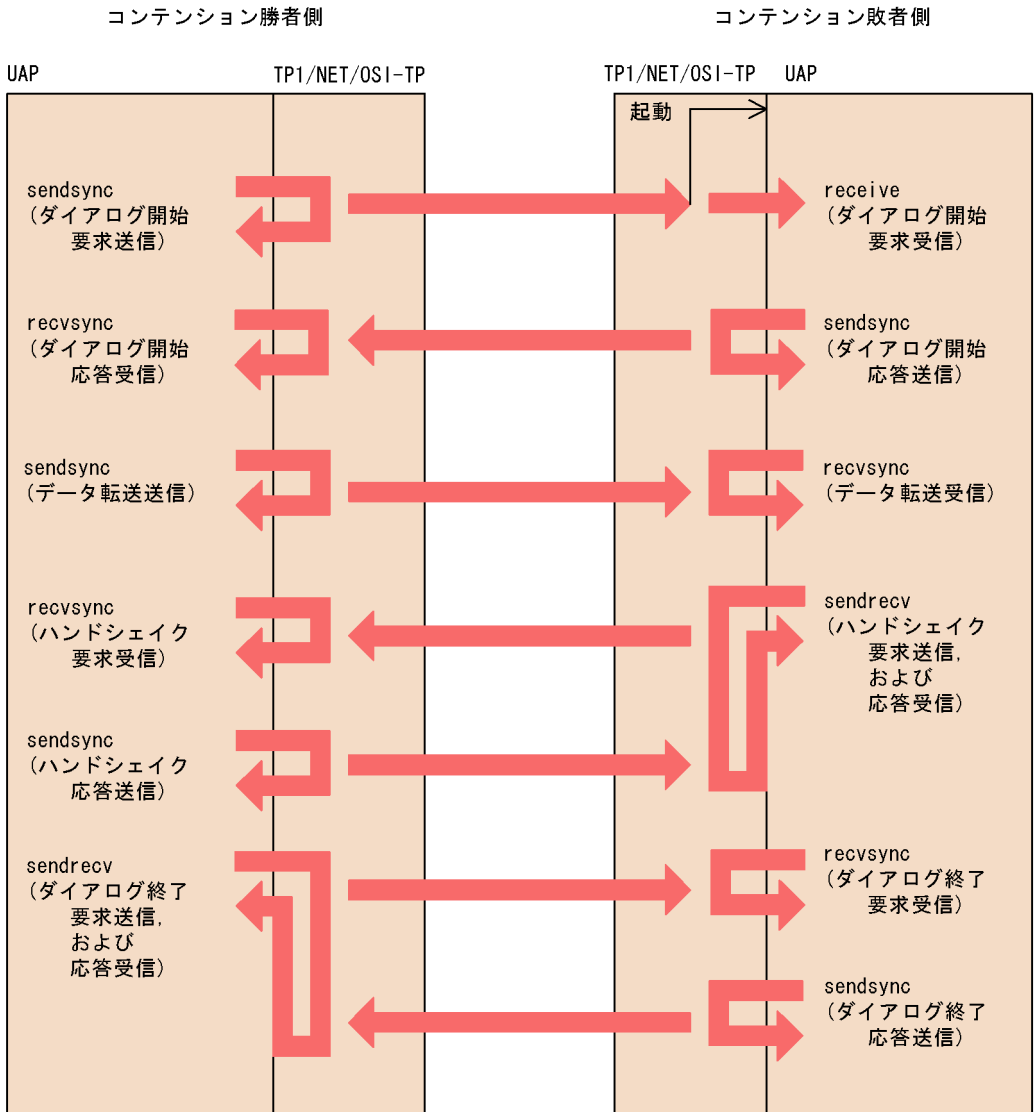
表 3-9 サービスプリミティブの突き合わせ項目（ダイアログ開始時）

突き合わせ項目		チェック内容
Functional Units	Shared Control または Polarized Control	どちらかが設定されていること。
	Handshake	設定されていること、または設定されていないこと。

ユーザアプリケーションプログラム作成例

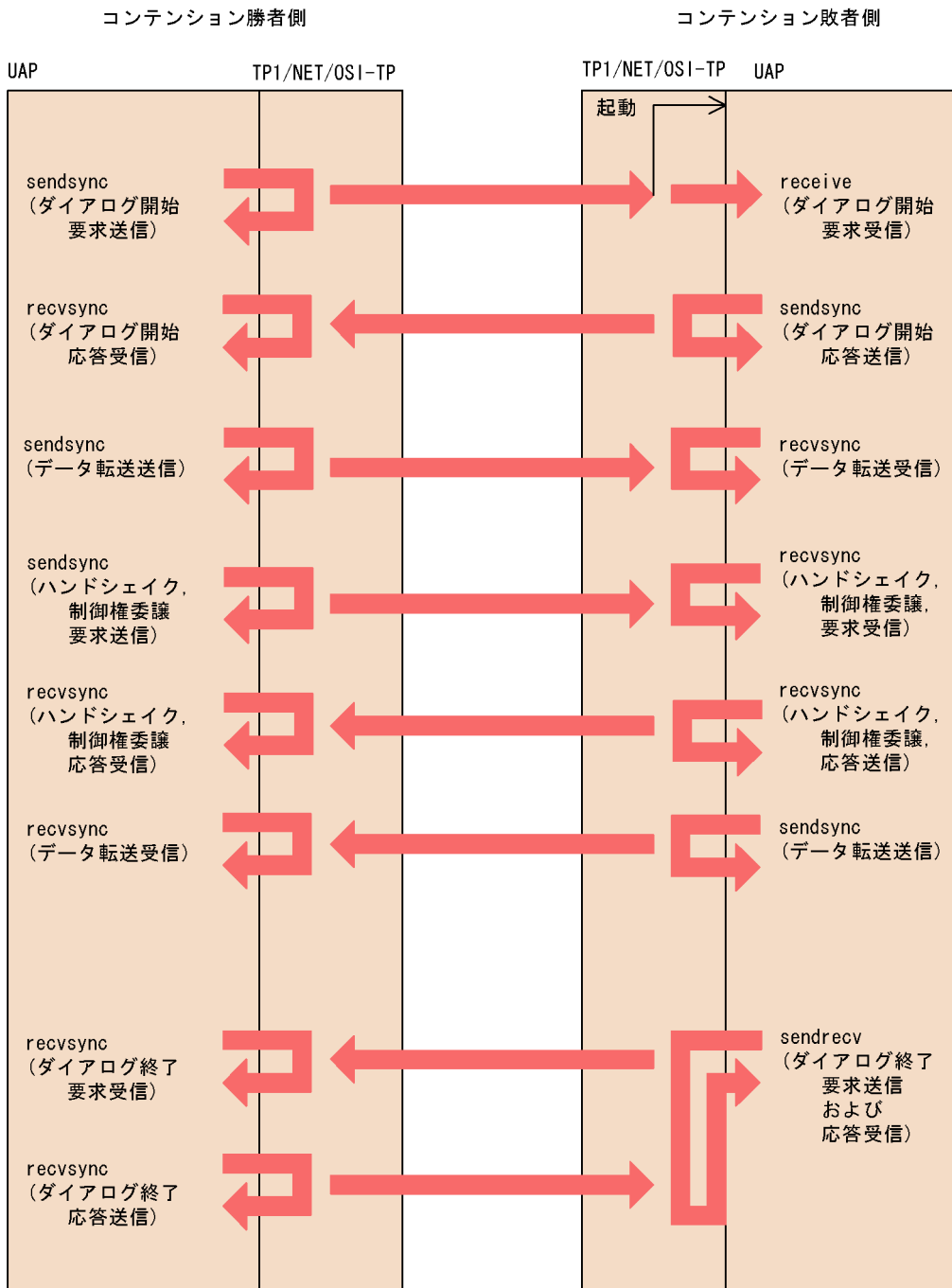
メッセージ送受信の処理の流れを全二重と半二重に分けて図 3-6 および図 3-7 に示します。C 言語、および COBOL 言語のコーディング例については、「付録 E ユーザアプリケーションプログラムのコーディング例」を参照してください。

図 3-6 処理の流れ (全二重の場合)



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

図 3-7 処理の流れ (半二重の場合)



4

ユーザOWNコーディング， MCF イベントインタフェース

この章では，TP1/NET/OSI-TP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）といいます。

TP1/NET/OSI-TP で使用する UOC を次に示します。

- 入力メッセージの編集 UOC
- 出力メッセージの編集 UOC

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

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の編集をする UOC です。

論理メッセージをユーザ任意の形式に変換したり，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したりできます。

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

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

（1）入力メッセージの編集

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

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

（2）アプリケーション名の決定

該当する MCF 通信プロセス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'がないときは，アプリケーション名を不正とし，エラーイベント（ERREVT1）を起動します。

アプリケーション名の決定の処理については，「2.2.5 アプリケーション名の決定」を参照してください。

（3）UOC エラーリターン処理

UOC から DCMCF_UOC_MSG_NG でリターンした場合，コネクションを解放し，障害通知イベント（CERREVT）を通知します。

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

（4）UOC パラメタ不正の場合の処理

UOC で設定したパラメタに不正があった場合，コネクションを解放し，障害通知イベント（CERREVT）を通知します。

（5）OpenTP1 への組み込み方法

スタート関数（dc_mcf_svstart）を発行する MCF メイン関数に，作成した UOC の関数アドレスを設定します。入力メッセージの編集 UOC の関数アドレスは任意に決められます。UOC 関数をコンパイルして生成した UOC オブジェクトファイルを，UOC 関数を登録した MCF メイン関数と結合して，TP1/NET/OSI-TP 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++ の場合

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

K&R 版 C の場合

4. ユーザOWNコーディング, MCF イベントインタフェース

```
#include <dcmcf.h>
#include <dcmotm.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];
    DCLONG rcv_prim;                ...受信サービスプリミティブ
    dcmcf_uocbuff_list_n *buflist_adr; ...受信バッファリストアドレス
    dcmcf_uocbuff_list_n *ebuflist_adr; ...編集バッファリストアドレス
    char aplname[9];                ...アプリケーション名
    char reserve2[7];
    char *pro_indv_ifa;              ...MCF使用領域
    DCLONG rtn_detail;              ...詳細リターンコード
    char reserve3[16];
} 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];
    dcmcf_uocbufinf_n buf_array[DCMCF_UOC_BUFF_MAX]; ...バッファ情報
} dcmcf_uocbuff_list_n;
```

(c) dcmcf_uocbufinf_n (バッファ情報) の内容

```
typedef struct {
    char *buf_adr;                   ...バッファアドレス
    DCLONG buf_size;                 ...バッファ最大長
    DCLONG seg_size;                 ...バッファ使用長
    char reserve1[4];
    dcmcfuoc_w_type buff_id;         ...MCF内部情報1
    DCLONG buff_addr;                ...MCF内部情報2
    char reserve2[4];
```

```
} dcmcf_uocbufinf_n;
```

(4) MCF が値を設定する項目

(a) dcmcf_uoc_min_n

pro_kind

プロトコル種別として，次の値が設定されます。

DCMCF_UOC_PRO_TP

OSI TP プロトコル

le_name

メッセージを入力した論理端末の名称が設定されます。

rcv_prim

受信サービスプリミティブとして，次のどちらかの値が設定されます。

DCMCF_UOC_RCV_BRD

ダイアログ開始要求 (TP-BEGIN-DIALOGUEind) 受信

DCMCF_UOC_RCV_REP_RE

上記以外の関数受信

buflist_adr

受信用バッファリストのアドレスが設定されます。

ebuflist_adr

編集用バッファリストのアドレスが設定されます。

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

aplname

MCF 通信構成定義の mcftalcle コマンドの -v オプションで指定したアプリケーション名，またはダイアログ開始要求時に受信側 TPSU 名称で設定したアプリケーション名が設定されます。

pro_indv_ifa

MCF で使用するパラメタです。

(b) dcmcf_uocbuff_list_n (バッファリスト)

buf_num

バッファ情報の数として 1 が設定されます。

buf_array

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

4. ユーザOWNコーディング，MCF イベントインタフェース

(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，-19999 ~ -19000 の範囲で設定してください。

(b) dcmcf_uocbuff_list_n (バッファリスト)

used_buf_num

使用したバッファ情報の数を設定します。

使用したバッファ情報数には，1 を設定してください。

(c) dcmcf_uocbufinf_n (バッファ情報)

seg_size

バッファの使用長を設定します。

(6) リターン値

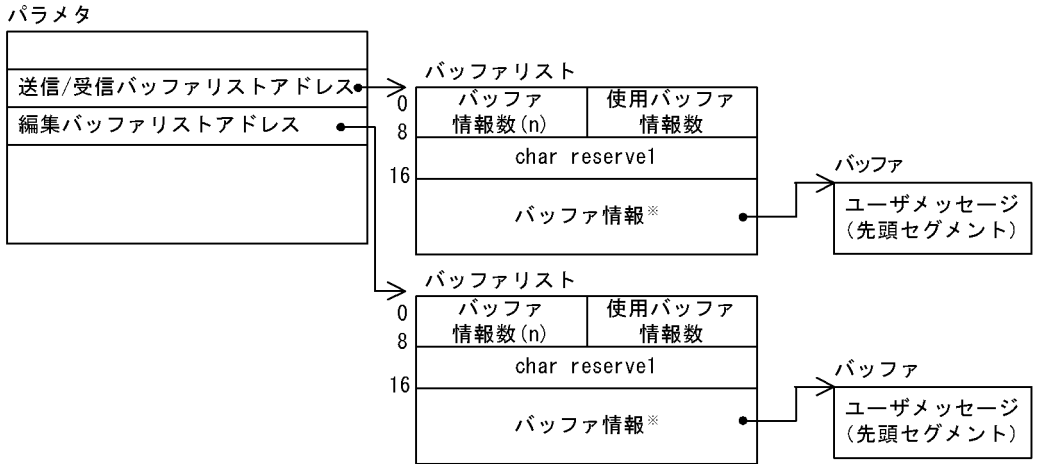
uoc_func() は次のコードでリターンしてください。

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

(7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係を次の図に示します。

図 4-1 UOC インタフェース用のパラメタとバッファの関係



注※ バッファ情報は32バイトで次の形式をしています。

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

(8) バッファインタフェース

入力メッセージ編集 UOC を使用する場合，APDU 連結があるかどうかによって，バッファのインタフェースが異なります。

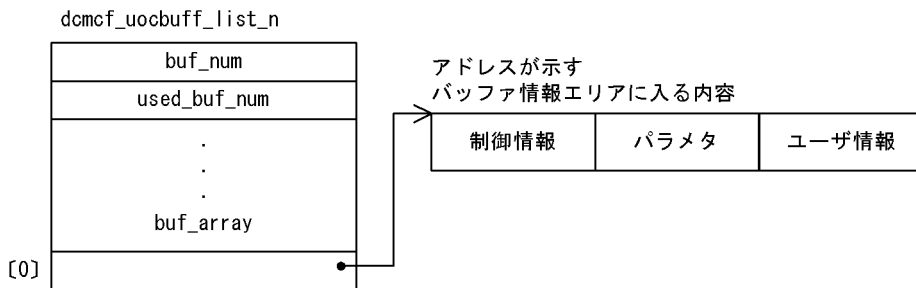
(a) APDU 連結がない場合

APDU 連結がない場合，セグメント数は 1 になります。ユーザセグメントは，バッファ情報エリアの 1 番目が使用されます。

APDU 連結がない場合のバッファインタフェース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 入力メッセージ編集 UOC のバッファインタフェース（APDU 連結なし）

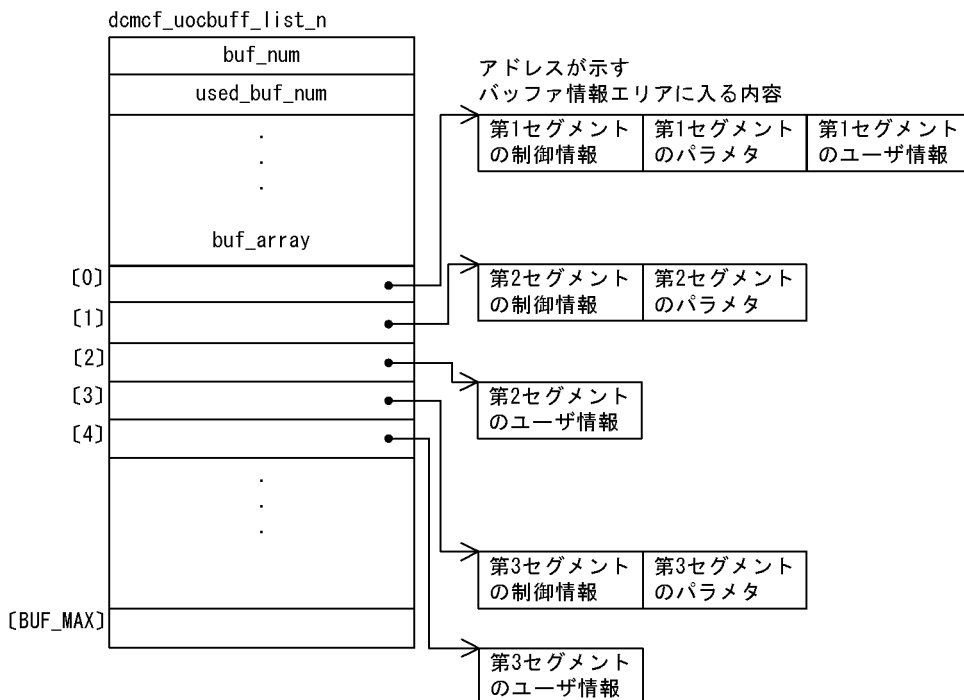


(b) APDU 連結がある場合

APDU 連結がある場合，複数セグメントのメッセージが入力されます。第 1 セグメントは，「(a) APDU 連結がない場合」と同様にバッファ情報エリアの 1 番目に入ります。第 2 セグメント以下は，各セグメントの制御情報およびパラメタと，ユーザ情報とがそれぞれ異なるバッファ情報エリアに入ります。

APDU 連結がある場合のバッファインタフェースを次の図に示します。

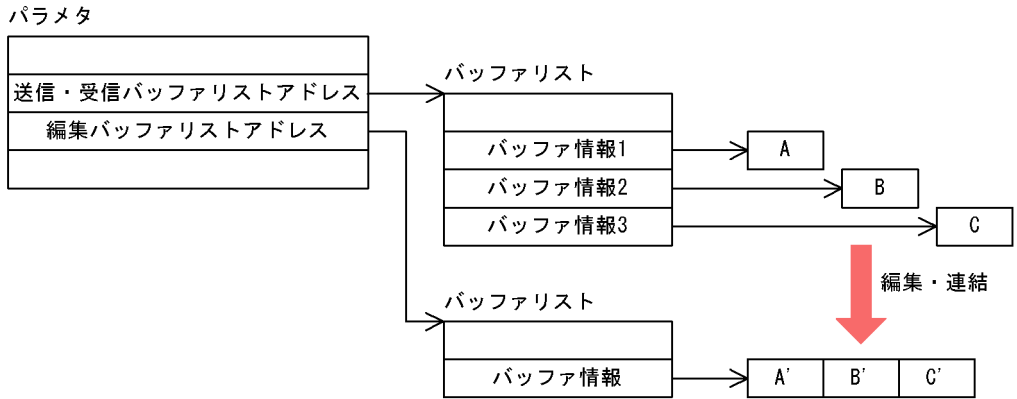
図 4-3 入力メッセージ編集 UOC のバッファインタフェース（APDU 連結あり）



TP1/NET/OSI-TP では，受信バッファ数に関係なく，使用できる編集バッファ数は一つだけです。編集バッファでスケジューリングをする場合，受信メッセージの編集後，連

結させたメッセージを編集バッファに設定してください。受信メッセージの連結の仕組みを次の図に示します。

図 4-4 受信メッセージの連結



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 通信構成定義でメッセージ編集用バッファグループ番号 (mcftalccn -e msgbuf) を定義します。

(1) 出力メッセージの編集

送信するメッセージが格納されている送信バッファ，および MCF 通信構成定義で指定した編集バッファを引き渡します。UOC では，これらのバッファを使用して，出力メッセージの編集処理ができます。

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

(2) UOC エラーリターン処理

UOC から DCMCF_UOC_MSG_NG でリターンした場合，コネクションを解放し，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，コネクションを解放し，障害通知イベント

4. ユーザOWNコーディング, MCF イベントインタフェース

(CERREVT) を通知します。該当するメッセージは破棄します。

(4) OpenTP1 への組み込み方法

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

4.1.4 出力メッセージ編集 UOC インタフェース

出力メッセージ編集 UOC は、次に示す形式で呼び出します。

(1) 形式

ANSI C , C++ の場合

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

K&R 版 C の場合

```
#include <dcmcf.h>
#include <dcmotm.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];
    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内部情報
    char   reserve2[1];
```

```

char    *pro_indv_ifa;           ...プロトコル個別インタフェース
                                   領域アドレス
DCLONG  rtn_detail;             ...詳細リターンコード
char    reserve3[20];
} dcmcf_uoc_mout_n;

```

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

入力メッセージ編集 UOC インタフェースのバッファリストおよびバッファ情報の内容と同じです。「4.1.2(3)(b) dcmcf_uocbuff_list_n (バッファリスト) の内容」および「4.1.2(3)(c) dcmcf_uocbufinf_n (バッファ情報) の内容」を参照してください。

(4) MCF が値を設定する項目

(a) dcmcf_uoc_mout_n

pro_kind

プロトコル種別として, 次の値が設定されます。

DCMCF_UOC_PRO_TP

OSI TP プロトコル

le_name

メッセージを入力した論理端末の名称が設定されます。

buflist_adr

送信用バッファリストのアドレスが設定されます。

ebuflist_adr

編集用バッファリストのアドレスが設定されます。

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。ただし, outputno_flag が DCMCF_UOC_OUTPUTNO_OK のときだけ有効です。

's'

同期送信メッセージ

outputno_flag

メッセージ出力通番有効フラグとして, 次のどちらかの値が設定されます。

DCMCF_UOC_OUTPUTNO_OK

メッセージ出力通番を有効にします。

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_OUTPUTNO_NG

メッセージ出力通番を無効にします。

resend_flag

再送フラグとして, 次の値が設定されます。

'n'

再送を要求しません。

pro_indv_ifa

MCF で使用するパラメタです。

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

入力メッセージ編集 UOC インタフェースのバッファリストおよびバッファ情報の内容と同じです。「4.1.2(4)(b) dcmcf_uocbuff_list_n (バッファリスト)」および「4.1.2(4)(c) dcmcf_uocbufinf_n (バッファ情報)」を参照してください。

(5) ユーザが値を設定する項目

(a) dcmcf_uoc_mout_n

rtn_detail

詳細リターンコードを設定します。

このコードは, UOC が DCMCF_UOC_MSG_NG をリターンしたときに, MCF に渡されます。

MCF は, 詳細リターンコードをメッセージログファイルに出力します。

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

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

入力メッセージ編集 UOC インタフェースのバッファリストおよびバッファ情報の内容と同じです。「4.1.2(5)(b) dcmcf_uocbuff_list_n (バッファリスト)」および「4.1.2(5)(c) dcmcf_uocbufinf_n (バッファ情報)」を参照してください。

(6) リターン値

uoc_func() は次のコードでリターンしてください。

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

(7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係は, 入力メッセージ編集 UOC 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。

(8) バッファインタフェース

出力メッセージ編集 UOC を使用する場合，APDU 連結があるかどうかによって，バッファのインタフェースが異なります。

(a) APDU 連結がない場合

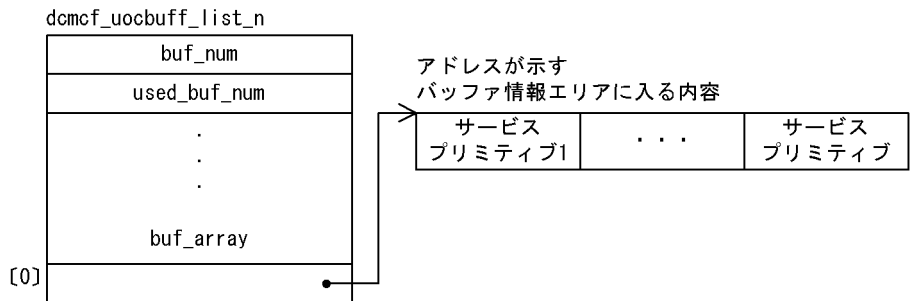
APDU 連結がない場合，入力メッセージ編集 UOC と同様に，セグメント数は 1 になります。ユーザセグメントは，バッファ情報エリアの 1 番目が使用されます。「4.1.2(8) バッファインタフェース」を参照してください。

(b) APDU 連結がある場合

APDU 連結がある場合，セグメント数は 1 になります。ユーザセグメントは，バッファ情報エリアの 1 番目が使用されます。UAP がメッセージ送信要求時に，セグメント領域に設定した内容が設定されます。

APDU 連結がある場合のバッファインタフェースを次の図に示します。

図 4-5 出力メッセージ編集 UOC のバッファインタフェース (APDU 連結あり)



4.1.5 UOC 作成上の注意事項

UOC 作成上の注意事項を次に示します。

(1) UOC の構造

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

(2) UOC で使用できる関数

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

メモリ操作をする関数

4. ユーザOWNコーディング，MCF イベントインタフェース

- データ領域管理（例：malloc，free）
- 共有メモリ管理関数 / システムコール（例：shmctl，shmget，shmop）
- メモリ操作（例：memory）
- 文字列操作（例：string）

時間取得関数

(3) UOC の異常処理

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

(4) UOC の実行タイミング

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

(5) ユーザ情報の操作方法

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

4.2 MCF イベントインタフェース

OpenTP1 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かが 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の編集 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/OSI-TP が発生させる MCF イベントの種類を次の表に示します。

表 4-1 TP1/NET/OSI-TP が発生させる 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 での処理の例
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，受信メッセージを廃棄しました。 <ul style="list-style-type: none"> 入力キューに障害が発生しました。 MHP のサービス，サービスグループ，アプリケーションが閉塞しました。 MHP の先頭セグメント受信文に，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 の先頭セグメントの受信文に，セグメントを渡したあとに，MHP の異常終了が発生しました。	UAP 異常終了時の対処障害メッセージを送信します。
タイマ起動メッセージ廃棄通知イベント	ERREVT4	アプリケーションのタイマ起動時に障害が発生しました。	メッセージを廃棄したことを報告します。
未処理送信メッセージ廃棄通知イベント	ERREVT5	MCF の正常終了処理時の，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）でメッセージを廃棄しました。	未処理送信メッセージを廃棄したことを報告します。

4. ユーザOWNコーディング，MCF イベントインタフェース

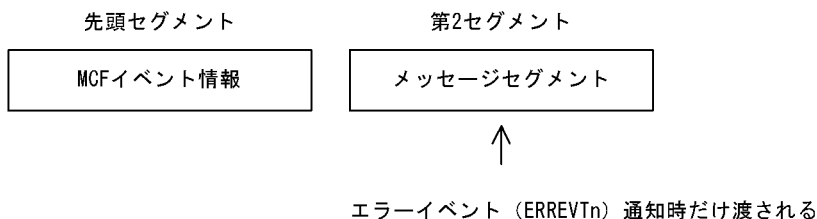
MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP での処理の例
障害通知イベント	CERREVT	通信管理プログラムのコネクション障害，または論理端末障害が発生しました。	コネクション，または論理端末に障害が発生したことを報告します。
状態通知イベント	COPNEVT	コネクションが確立しました。	コネクションが確立したことを報告します。
	CCLSEVT	コネクションが正常に解放されました。	コネクションが解放されたことを報告します。

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 セグメントに処理できなかったメッセージセグメントを設定します。

MCF イベント通知時のセグメント構成を次の図に示します。

図 4-6 MCF イベント通知時のセグメント構成



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。

COBOL 言語を使用した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

ト情報の先頭部分 (ヘッダ) の形式は, 各イベントに共通です。

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, dc_mcf_evtheader は, <dcmcf.h> で定義されています。<demotm.h> の前に <dcmcf.h> を取り込んでおいてください。

(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, CERREVT,
                                   COPNEVTおよびCCLSEVT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[1] ;           ... 予備
    DCLONG time ;                 ... メッセージ入力時刻
};
```

(b) MCF イベントとして設定される項目

le_name

メッセージを入力した論理端末名称が設定されます。

ERREVT2 および ERREVT3 で, 次を示す場合には, '*' が設定されます。

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

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

cn_name

コネクション名が設定されます。

ERREVT2 および ERREVT3 で, 次を示す場合には, '*' が設定されます。

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

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

4. ユーザOWNコーディング, MCF イベントインタフェース

```
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。

- 形式不正のアプリケーション名
- 定義されていないアプリケーション名

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

(3) ERREVT2

(a) 形式

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

(b) MCF イベントとして設定される項目

ap_name

エラーになった UAP のアプリケーション名が設定されます。

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

MHP 以外から送信された場合はヌル文字が設定されます。

reason_code

ERREVT2 の理由コードが設定されます。理由コードの詳細については、「付録 D
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

トランザクションランチ ID が設定されます。

(5) ERREVT4

(a) 形式

```

struct dc_mcf_evta_type {
    struct dc_mcf_evtheader  evtheader ;
                                ... MCFイベント共通ヘッダ
    char reserve01[12] ;      ... 予備
    char map_name[10] ;      ... MCF使用領域
    char reserve03[2] ;      ... 予備
    char ap_name[10] ;       ... アプリケーション名
                                (正常終了したメッセージ
                                のアプリケーション名)
    char reserve04[2] ;      ... 予備
    char reserve05[32] ;     ... 予備
    char reserve06[32] ;     ... 予備
    DCLONG user_leng ;       ... 他プロトコルの場合の使用領域
};

```

4. ユーザOWNコーディング, MCF イベントインタフェース

```
char    user_data[16] ;           ... 他プロトコルの場合の使用領域
char    reserve07[16] ;         ... 予備
} ;
```

(b) MCF イベントとして設定される項目

ap_name

正常終了したメッセージのアプリケーション名が設定されます。

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

MHP 以外から送信された場合はヌル文字が設定されます。

(6) CERREVT

(a) 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 err_rcv_action ;    ... 回復動作情報
    char    group_name[16] ;    ... コネクショングループ名
    char    reserve1[26] ;      ... 予備
} dcmotm_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

err_reason1, err_reason2

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

err_rcv_action

CERREVT 起動時に, 回復動作情報として次の値が設定されます。

DCMOTM_RSV_MANUAL

コマンド入力による手動回復

DCMOTM_RSV_AUTO

システムによる自動回復

(7) COPNEVT, CCLSEVT

(a) 形式

```
typedef struct {
```

```

struct dc_mcf_evtheader header ;
... MCFイベント共通ヘッダ
DCLONG cls_rcv_action;
... 回復動作情報
... (CCLSEVTの場合だけ有効)
char reserve1[12];
... 予備
char group_name[16];
... コネクショングループ名
char reserve2[26];
... 予備
} dcmotm_statevt ;

```

(b) MCF イベントとして設定される項目

cls_rcv_action

CCLSEVT 起動時に，回復動作情報として次の値が設定されます。

DCMOTM_RSV_MANUAL

コマンド入力による手動回復

DCMOTM_RSV_AUTO

システムによる自動回復

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 イベント情報の内容を表 4-2 ~ 表 4-7 に示します。

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベントコード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT1 を示す '1' が設定されます。
入力元論理端末名称	12	8	8	英数字	メッセージを入力した論理端末名称です。
予備	20	16	20	-	-
アプリケーション名	40	36	8	英数字	次に示すどちらかが設定されます。 <ul style="list-style-type: none"> 形式不正となったアプリケーション名 定義されていないアプリケーション名
予備	48	44	8	-	-

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			
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	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	104	100	8	外部 10 進	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
予備	112	108	16	-	-

(凡例)

- : 該当しません。または，使用されません。

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

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

4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
入力元論理端末名称	12	8	8	英数字	メッセージを入力した論理端末名称です。 次に示す場合には，'*' が設定されます。 <ul style="list-style-type: none"> SPP がアプリケーション起動をした MHP で障害が発生した場合 上記の障害発生時に MCF イベントとして起動された MHP によって，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	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	104	100	8	外部 10 進	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。

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			
理由コード	112	108	4	外部 10 進	理由コードが設定されます。
予備	116	112	12	-	-

(凡例)

- : 該当しません。または、使用されません。

注

理由コードの内容については、「付録 D 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。
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	-	-

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			
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	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	104	100	8	外部 10 進	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
予備	112	108	16	-	-
サービス名	128	124	31	英数字	異常が発生した UAP のアプリケーション名に対応するサービス名です。
予備	159	155	1	-	-
サービスグループ名	160	156	31	英数字	異常が発生した 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	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	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	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	104	100	8	外部 10 進	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
予備	112	108	16	-	-
予備	128	124	31	-	-
予備	159	155	1	-	-
予備	160	156	31	-	-
予備	191	187	1	-	-

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 (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
予備	16	8	-	-
入力元コネクション名	24	8	英数字	コネクション名です。
メッセージ入力日付	32	8	外部 10進	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	CERREVT を入力した時刻です。
障害要因コード	48	4	2進	障害要因コードが設定されます。 (00000030) ₁₆ : コネクション障害
理由コード 1	52	4	2進	理由コード 1 が設定されます。
理由コード 2	56	4	2進	理由コード 2 が設定されます。
回復動作情報	60	4	2進	障害時，システムを回復する方法を示す値が設定されます。 (ffffff) ₁₆ : コマンド入力による手動回復 (00000000) ₁₆ : システムによる自動回復
コネクショングループ名	64	16	英数字	入力元論理端末およびコネクションが属するコネクショングループ名が設定されます。 入力元論理端末およびコネクションがコネクショングループに属さない場合，スペースが設定されます。
予備	80	28	英数字	-

4. ユーザOWNコーディング，MCF イベントインタフェース

(凡例)

- : 該当しません。または，使用されません。

注

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

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

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	イベントコード「COPNEVT」，または「CCLSEVT」が設定されます。
入力元論理端末名称	8	8	英数字	入力元論理端末名称が設定されます。
予備	16	8	-	-
入力元コネクション名	24	8	英数字	コネクション名が設定されます。
メッセージ入力日付	32	8	外部 10 進	COPNEVT, CCLSEVT を入力した日付です。
メッセージ入力時刻	40	8	外部 10 進	COPNEVT, CCLSEVT を入力した時刻です。
回復動作情報	48	4	2 進	障害時，システムを回復する方法を示す値が設定されます。 (ffffff) ₁₆ : コマンド入力による手動回復 (00000000) ₁₆ : システムによる自動回復
予備	52	12	-	-
コネクショングループ名	64	16	英数字	入力元論理端末およびコネクションが属するコネクショングループ名が設定されます。 入力元論理端末およびコネクションがコネクショングループに属さない場合，スペースが設定されます。
予備	80	28	英数字	-

(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、OSI TP 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/OSI-TP 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、相手システムの通信定義と関連づける内容、およびシステム定義例について説明します。

TP1/NET/OSI-TP の定義の概要

TP1/NET/OSI-TP 固有のシステム定義の種類

mcfmuap (UAP 共通定義)

mcftalccn (コネクション定義の開始)

mcftalced (コネクション定義の終了)

mcftalcle (論理端末定義)

mcftgrpcn (コネクショングループ定義の開始)

mcftgrped (コネクショングループ定義の終了)

システムサービス情報定義

システムサービス共通情報定義

MCF 定義オブジェクトの生成

自システムの通信管理プログラム (XNF/AS) と関連づける内容

相手システムの通信定義と関連づける内容

定義例

TP1/NET/OSI-TP の定義の概要

TP1/NET/OSI-TP 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や、相手システムの通信定義と、システム定義内容を関連づける必要があります。

OpenTP1 のネットワークコミュニケーション定義の中での定義

OpenTP1 のネットワークコミュニケーション定義のうち、TP1/NET/OSI-TP 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/OSI-TP 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 マネジャ定義	MCF マネ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/OSI-TP に固有の定義があるのは、「MCF 通信構成定義」だけです。

この章では、TP1/NET/OSI-TP に固有の定義コマンド、オプション、およびオペランドについて説明します。

プロトコルで共通の定義については、マニュアル「OpenTP1 システム定義」を参照してください。ただし、`mcfbuf` (バッファグループ定義) の `length` オペランド、および `count` オペランドの指定値については、「`mcfalecn` (コネクション定義の開始)」の「注意事項」を参照してください。

TP1/NET/OSI-TP 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/OSI-TP 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/OSI-TP 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

OSI TP プロトコルを使用して相手システムと通信するためには、TP1/NET/OSI-TP 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や、相手システムの通信定義と関連づける必要があります。

この章では、TP1/NET/OSI-TP と通信管理 XNF/AS とで関連づける内容を示します。また、相手システム (OSAS/TP/DCCM3) のネットワーク定義と関連づける内容を示します。

5. システム定義

TP1/NET/OSI-TP 固有のシステム定義の種類

TP1/NET/OSI-TP 固有のシステム定義の種類

TP1/NET/OSI-TP に固有の定義の種類を、次の表に示します。

表 5-2 TP1/NET/OSI-TP 固有の定義の種類

定義名	コマンド	オプション・オペランド		定義内容	指定値 ((値範囲)) 《省略時解釈値》	
MCF マネージャ定義	mcfmuap	-t	sndtim	同期送信監視時間 (SENDSYNC)	符号なし整数 (単位 : 秒) ((0 ~ 65535)) 《 0 》	
			sndrcvtim	同期送受信監視時間 (SENDRECV)	符号なし整数 (単位 : 秒) ((0 ~ 65535)) 《 0 》	
			recvtim	同期受信監視時間 (RECVSYNC)	符号なし整数 (単位 : 秒) ((0 ~ 65535)) 《 0 》	
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	mcfmgrpcn (コネクショングループ定義の開始) 指定数 : 0 ~ 256	-g	-	コネクショングループ名	1 ~ 8 文字の識別子
		mcfaltccn (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	tp
			-n	-	自システムの PSAP アドレス	1 ~ 142 けたの 16 進数字
			-q	-	通信相手システムの PSAP アドレス	1 ~ 186 けたの 16 進数字
			-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	符号なし整数 ((1 ~ 512))

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》	
		-e	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))
			count	メッセージ編集用バッファ数	符号なし整数 ((1 ~ 131070))
		-m	mode	使用する通信管理	xnfas
		-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》
		-u	-	通信相手システムの種別	ht
		-w	nomltim	プロトコル監視時間	符号なし整数 (単位: 秒) ((0 または 10 ~ 65535)) 《60》
		-b	bretry	コネクション確立後に障害が発生した場合にコネクション確立再試行するかどうか	《yes》 no
			bretrycnt	コネクション確立後に障害が発生した場合のコネクション確立再試行の回数	符号なし整数 (単位: 回) ((0 ~ 65535)) 《0》
			bretryint	コネクション確立後に障害が発生した場合のコネクション確立再試行の時間間隔	符号なし整数 (単位: 秒) ((0 ~ 2550)) 《60》
		-t	-	起動型および受信型の種別	int rsp
		-z	dtea	現用相手 DTE アドレス	1 ~ 32 けたの 16 進数字
			slot	現用スロット番号	符号なし整数 ((0 ~ 65535))
			altn_dtea	交代用相手 DTE アドレス	1 ~ 32 けたの 16 進数字 (n:1 ~ 7 を昇順に指定)
			altn_slot	交代用スロット番号	符号なし整数 (n: 1 ~ 7 を昇順に指定) ((0 ~ 65535))

5. システム定義

TP1/NET/OSI-TP 固有のシステム定義の種類

定義名	コマンド	オプション・オペランド	定義内容	指定値 ((値範囲)) 《省略時解釈値》
		-l -	TL クラス	0 《2》 4
		-y -	コンテンション勝者および敗者の種別	int rsp
		-o -	応用コンテキスト名	1 ~ 64 けたの 16 進数字
		-j -	ユーザ ASE 抽象構文名	1 ~ 64 けたの 16 進数字
		-d control	全二重および半二重の種別	《shared》 polarized
		handshake	ハンドシェイクを使用するかどうか	《yes》 no
		-v cnflevt	コネクション起動側でのコネクション確立の失敗時に、状態通知イベント (CERREVT) を通知するかどうか	yes 《no》
	mcftalcle (論理端末定義) 指定数 : 1 ~ 512	-l -	論理端末名称	1 ~ 8 文字の識別子
		-t -	論理端末タイプ	any
		-v -	アプリケーション名	1 ~ 8 文字の識別子
	mcftalced (コネクション定義の終了) 指定数 : mcftalccn と同数	- -	コネクション定義の終了	-
	mcftgrped (コネクショングループ定義の終了) 指定数 : mcftgrpen と同数	- -	コネクショングループ定義の終了	-

(凡例)

- : 該当しません。

注

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

定義の指定順序

TP1/NET/OSI-TP 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。MCF 通信構成定義コマンドを指定するときは、必ずこの順序に従ってください。

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

(1) コネクションループを指定しない場合

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)		
⋮	繰り返し指定可能		
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)		

(2) コネクションループを指定する場合

}	mcftgrpcn	(コネクショングループ定義の開始)	}	※
	mcftalccn	(コネクション定義の開始)		
	mcftalced	(コネクション定義の終了)		
	⋮	繰り返し指定可能		
	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)			
	mcftgrped	(コネクショングループ定義の終了)		

(3) 混在する場合

前述の(1)と(2)が混在する場合、前述の定義(1)、(2)または(2)、(1)を交互に繰り返し指定します。

注

mcftalccn、mcftalcle、mcftalced は 1 : 1 : 1 になるように指定します。

mcfmuap (UAP 共通定義)

形式

```
mcfmuap      :  
      [-t " {sndtim=同期送信監視時間}  
          {sndrcvtim=同期送受信監視時間}  
          {recvtim=同期受信監視時間} " ]  
      :
```

機能

UAP に共通する環境を定義します。

オプション

この定義コマンドには、ほかにもオプションがあります。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

-t

(オペランド)

sndtim= 同期送信監視時間 ~ <符号なし整数> ((0 ~ 65535)) 《0》(単位: 秒)
同期型のメッセージ送信の仕掛け開始 (sendsync (EMI) 発行) から仕掛け終了 (sendsync 終了) までの限界監視時間を指定します。このオペランドでは、相手システムから処理完了が報告されるまでの時間を監視します。0 を指定した場合、送信時間の監視はしません。

sndrcvtim= 同期送受信監視時間 ~ <符号なし整数> ((0 ~ 65535)) 《0》(単位: 秒)
同期型のメッセージ送受信の仕掛け開始 (sendrecv (EMI) 発行) から仕掛け終了 (sendrecv 終了) までの限界監視時間を指定します。このオペランドでは、相手システムからの応答時間を監視します。0 を指定した場合、送受信時間の監視はしません。

recvtim= 同期受信監視時間 ~ <符号なし整数> ((0 ~ 65535)) 《0》(単位: 秒)
同期型のメッセージ受信の仕掛け開始 (recvsync (EMI) 発行) から仕掛け終了 (recvsync 終了) までの限界監視時間を指定します。このオペランドでは、相手システムから処理完了が報告されるまでの時間を監視します。0 を指定した場合、受信時間の監視はしません。

mcftalccn (コネクション定義の開始)

形式

```
mcftalccn  -c コネクションID
           -p tp
           -n x'自システムのPSAPアドレス'
           -q x'通信相手システムのPSAPアドレス'
           -g "sndbuf = メッセージ送信用バッファグループ番号" 1
              rcvbuf = メッセージ受信用バッファグループ番号" 1
           [-e " [ {msgbuf = メッセージ編集用バッファグループ番号} 1, 2
              count = メッセージ編集用バッファ数 ] " ] 1, 2
           -m "mode = xnfas" 1
           [-i auto | manual] 1
           -u ht
           [-w " {nomltim = プロトコル監視時間} " ] 1
           [-b " {bretry = yes | no} 1
              {bretrycnt = コネクション確立再試行回数} 1
              {bretryint = コネクション確立再試行間隔} " ] 1
           -t int | rsp 1, 2, 3
           -z " {dtea = x'現用相手DTEアドレス'} 1
              slot = 現用スロット番号 1
              {altn_dtea = x'交代用相手DTEアドレス'} 1
              {altn_slot = 交代用スロット番号} " 1
           [-l 0 | 2 | 4] 1, 2
           [-y int | rsp] 1, 2, 3
           -o x'応用コンテキスト名' 1
           -j x'ユーザASE抽象構文名' 1
           [-d " {control = shared | polarized} 1
              {handshake = yes | no} " ] 1
           [-v " {cnflevt = yes | no} " ]
```

注 1

-n オプション (自システムの PSAP アドレス) が同じとき、指定を同じにする必要があります。

注 2

コネクションを複数指定する場合、-n オプション (自システムの PSAP アドレス) と、-q オプション (通信相手システムの PSAP アドレス) の対が同じとき、指定を同じにする必要があります。

注 3

同一のコネクショングループ定義内のコネクションの場合、指定値を同じにする必要があります。

5. システム定義

mcftalccn (コネクション定義の開始)

機能

コネクションに関する環境を定義します。

オプション

-c コネクション ID ~ < 1 ~ 8 文字の識別子 >

OpenTP1 システム内で、一意となるコネクション ID を指定します。

-p tp

プロトコルの種別を指定します。

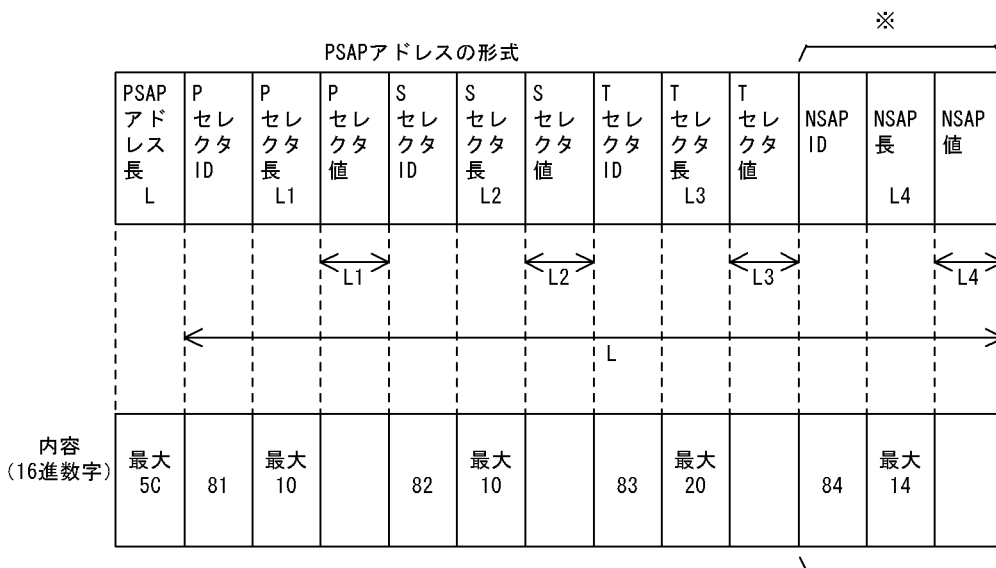
tp

OSI TP プロトコル

-n x' 自システムの PSAP アドレス ' ~ < 1 ~ 142 けたの 16 進数字 >

自システムの PSAP アドレスを指定します。x は、16 進数形式で指定することを意味します。自システムの PSAP アドレスは、他の通信構成定義ファイルで指定する値と重複しないよう指定してください。

PSAP アドレスの形式を次に示します。



注

NSAP アドレスは、通信相手システムの PSAP アドレスの場合だけ指定します。

NSAP アドレスの形式については、次のマニュアルを参照してください。

- マニュアル「システムセットアップガイド NSAP アドレス概説編」
- マニュアル「システムセットアップガイド 通信管理編」

- マニュアル「通信管理 XNF/AS NSAP アドレス概説編」
- マニュアル「通信管理 XNF/AS 構成定義編」

-q x' 通信相手システムの PSAP アドレス' ~ < 1 ~ 186 けたの 16 進数字 >

通信相手システムの PSAP アドレスを指定します。x は、16 進数形式で指定することを意味します。

PSAP アドレスの形式については、-n オプションを参照してください。

-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グループ番号を指定します。

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

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

メッセージ受信用バッファグループ番号を指定します。

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

-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 を使用する場合に、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、および出力メッセージ編集 UOC 使用時に、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。

メッセージ編集用バッファグループ番号 (msgbuf オペランドで指定) に対応する mcftbuf コマンドで指定するバッファ数 (-g オプションの count オペランドおよび extend オペランド) 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。2 以上の値を指定しても、入力、および出力メッセージ編集 UOC で使用できる編集バッファ数は一つだけです。

また、この count オペランドで指定するメッセージ編集用バッファ数は、mcftbuf コマンドで指定するバッファ数 (-g オプションの count オペランドおよび extend オペランド) の合計値を超える指定はできません。

5. システム定義

mcftalccn (コネクション定義の開始)

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。

-m

(オペランド)

mode=xfas

使用する通信管理を指定します。

xfas

XNF/AS を使用します。

-i auto | manual ~ 《manual》

コネクションの確立方法を指定します。

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) を入力してコネクションを確立します。

-u ht

通信相手システムの種別を指定します。

ht

通信相手システムがホストコンピュータであることを示します。

-w

(オペランド)

nomltim= プロトコル監視時間 ~ <符号なし整数> ((0, 10 ~ 65535)) 《60》(単位 : 秒)

コネクション確立時の監視時間を指定します。

プロトコル監視時間に 0 を指定した場合、時間監視はしません。

-b

(オペランド)

bretry=yes | no ~ 《yes》

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

yes

コネクションの確立再試行をします。

no

コネクションの確立再試行をしません。

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

コネクション確立時に障害が発生した場合、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、無限に確立再試行を繰り返します。

bretry オペランドで no を指定した場合、bretrycnt オペランドの指定は無効になります。

通信管理から再試行不可能な障害が通知された場合、または相手システムから確立要求に対する拒否応答を受けた場合は、再試行を中止します。

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

コネクション確立時に障害が発生した場合、MCFが行う確立再試行の間隔を指定します。0を指定した場合、障害が発生するたびにコネクションの確立再試行をします。

bretry オペランドで no を指定した場合、bretryint オペランドの指定は無効になります。

-t int | rsp

コネクションの起動、または受信の種別を指定します。

int

自システムが起動側です。

rsp

自システムが受信側です。

自システムで、一つの PSAP アドレスに複数のコネクションを確立する場合、各コネクションで定義する mcftalccn コマンドの -n オプション (自システムの PSAP アドレス) の指定値を一致させます。また、その場合は、-t オプションの指定値もほかの mcftalccn コマンドと一致させてください。

-z

(オペランド)

dtea='現用相手 DTE アドレス' ~ <1 ~ 32 けたの 16 進数字>

現用の相手システムの DTE アドレスを指定します。x は、16 進数形式で指定することを意味します。

slot= 現用スロット番号 ~ <符号なし整数> ((0 ~ 65535))

現用のスロット番号を指定します。

alt_n_dtea='交代用相手 DTE アドレス'(n: 1 ~ 7) ~ 1 ~ 32 けたの 16 進数字

5. システム定義

mcftalccn (コネクション定義の開始)

経路交代用の通信相手システムの DTE アドレスを指定します。x は、16 進数形式で指定することを意味します。

n は交代の順序であり、1 から昇順に指定する必要があります。

altn_slot= 交代用スロット番号 (n: 1 ~ 7) ~ <符号なし整数> ((0 ~ 65535))

経路交代用のスロット番号を指定します。

n は交代の順序であり、1 から昇順に指定する必要があります。

また、altn_dtea の n と altn_slot の n は対として同じ値で指定してください。

-l0|2|4 《2》

使用する TL クラスを指定します。

0

TL クラス 0 を使用します。

2

TL クラス 2 を使用します。

4

TL クラス 4 を使用します。

RFC 1006 接続 (XNF/AS/OSI Extension を使用) の場合は TL クラス 0, WAN 接続の場合は TL クラス 2, LAN 接続の場合は TL クラス 4 を指定してください。ただし、LAN 接続の場合でも TL クラス 2 を使用する場合がありますので、ネットワークの構成に合わせて指定する必要があります。

ただし、TL クラス 4 は使用できません。

-y int | rsp

コンテンツの勝者または敗者の種別を指定します。このオプションを省略した場合、mcftalccn コマンドの -t オプションで指定した値が仮定されます。

int

起動側のシステムが勝者となります。

rsp

受信側のシステムが勝者となります。

-o x' 応用コンテキスト名 ' ~ < 1 ~ 64 けたの 16 進数字 >

応用コンテキスト名を指定します。

-j x' ユーザ ASE 抽象構文名 ' ~ < 1 ~ 64 けたの 16 進数字 >

ユーザ ASE 抽象構文名を指定します。

-d

(オペランド)

control = shared | polarized ~ 《shared》

全二重機能と半二重機能のどちらを使用するかを指定します。

shared

全二重機能を使用します。

polarized

半二重機能を使用します。

handshake=yes | no ~ 《yes》

ハンドシェイクを使用するかどうかを指定します。

yes

ハンドシェイクを使用します。

no

ハンドシェイクを使用しません。

-v

(オペランド)

cnflevt=yes | no ~ 《no》

コネクション起動側でコネクション確立に失敗したときに、状態通知イベント (CERREVT) を通知するかどうかを指定します。

-t オプションに *rsp* (コネクション受信側) を指定した場合、このオプションの指定は無効です。この場合、コネクション確立時に障害が発生しても、CERREVT は通知しません。

yes

コネクション起動側でのコネクション確立の失敗時に、CERREVT を通知しません。

no

コネクション起動側でのコネクション確立の失敗時に、CERREVT を通知しません。

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は、バッファグループ定義の *mcftbuf* コマンドに対応しています。mcftbuf コマンドでは、1 コネクション単位に次の表に示す資源が必要です。バッファグループ定義については、マニュアル「OpenTP1 システム定義」を参照してください。

5. システム定義

mcftalccn (コネクション定義の開始)

バッファ種別	length オペランド ¹	count オペランド
sndbuf	<ul style="list-style-type: none"> ダイアログ開始要求時 (TP-BEGIN-DIALOGUEreq) 起動側 TPSU 名称長 ² + 受信側 TPSU 名称長 + ユーザデータ長 以上 ダイアログ開始要求以外 ユーザデータ長 以上 	2 以上
rcvbuf	sndbuf と同じ	<ul style="list-style-type: none"> 起動側 相手システムから連続して送られてくるメッセージ数 + 1 受信側 相手システムから連続して送られてくるメッセージ数 + 2 + 自システムでの異なる PSAP 数
msgbuf	最大セグメント長以上	1

注 1

APDU 連結をする場合、連結するサービスごとに上記の長さを求めます。合計した値を、連結したサービス全体の長さとして指定してください。

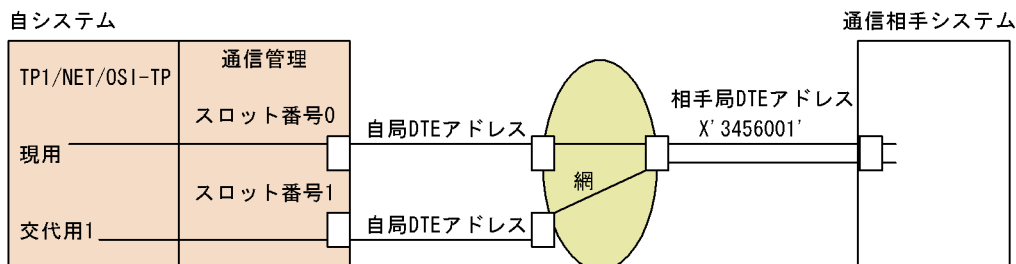
注 2

TPSU 名称長の最大値は 64 バイトです。

なお、mcftbuf コマンドの length オペランドでは、ユーザレベルでの最大セグメント長を指定します。相手システムによっては、相手システムのバッファ資源として、最大セグメント長にプロトコルヘッダ長を加算した長さで定義する場合があるので注意してください。

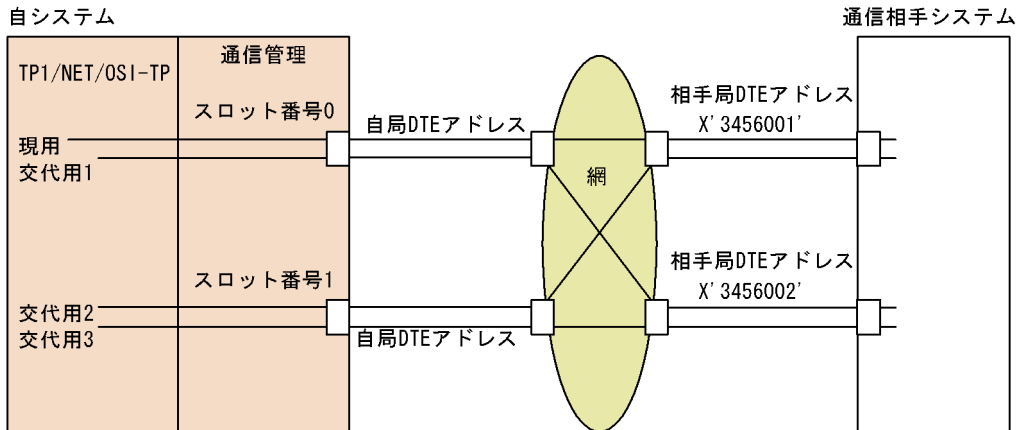
-z オプションで経路交代の定義をする場合、次のように指定します。

```
-z "dtea=x'3456001'
    slot=0
    alt1_dtea=x'3456001'
    alt1_slot=1"
```

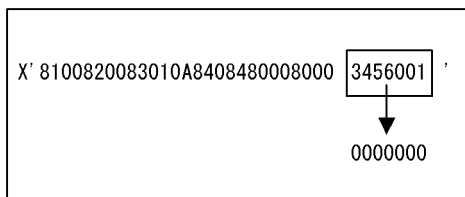


```
-z "dtea=x'3456001'
```

```
slot=0
alt1_dtea=x'3456002'
alt1_slot=0
alt2_dtea=x'3456001'
alt2_slot=1
alt3_dtea=x'3456002'
alt3_slot=1"
```



複数の相手局 DTE アドレスを使用する場合、MCF 構成定義で指定する通信相手システムの PSAP アドレスで、DTE アドレスの部分にはヌル文字を設定します。



ネットワーク層以下の層で障害が発生しても、経路交代機能を使用すると、1 回のコネクション確立要求に対して現用と交代用の合計経路数だけ再試行します。

また、コネクション確立再試行回数を指定しておくで、次に示す回数だけ再試行します。

(現用と交代用の合計経路数) × (コネクション確立再試行回数)

5. システム定義

mcftalced (コネクション定義の終了)

mcftalced (コネクション定義の終了)

形式

mcftalced

機能

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。

mcftalcle (論理端末定義)

形式

```
mcftalcle  -l  論理端末名称  
           -t  any  
           [-v アプリケーション名]
```

機能

論理端末に関する環境を定義します。

オプション

-l 論理端末名称 ~ < 1 ~ 8 文字の識別子 >

OpenTP1 システム内で、一意となる論理端末名称を指定します。

-t any

この論理端末の端末タイプを指定します。

any

任意型論理端末

-v アプリケーション名 ~ < 1 ~ 8 文字の識別子 >

入力メッセージを受信した場合に起動するアプリケーション名称 (MHP) を指定します。MCF アプリケーション属性定義 (mcfalcap -n オプションの name オペランド) で定義した名称を指定してください。MCF アプリケーション属性定義の詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

このオプションを省略した場合は、ダイアログ開始要求 (TP-BEGIN-DIALOGUE) の受信側 TPSU 名称、または入力メッセージ編集 UOC で指定された値をアプリケーション名とします。

5. システム定義

mcftgrpcn (コネクショングループ定義の開始)

mcftgrpcn (コネクショングループ定義の開始)

形式

mcftgrpcn -g コネクショングループ名

機能

コネクショングループ定義の開始を示します。

オプション

-g コネクショングループ名 ~ < 1 ~ 8 文字の識別子 >

コネクショングループ名を指定します。論理端末名称またはコネクション ID と同じにならないように指定してください。

mcftgrped (コネクショングループ定義の終了)

形式

mcftgrped

機能

コネクショングループ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/OSI-TP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/OSI-TP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/OSI-TP 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/OSI-TP 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 ~ 2048))
```

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

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

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

注 1

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

注 2

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

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 ~ 2048))

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) + 30²

注 1

TP1/NET/OSI-TP 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 MCF 通信プロセスごとに計算し、その結果の中で最大値が 500 より大きい場合は、その値を指定します。500 以下の場合は、500 を指定します。指定値を超えてファイルのアクセスが発生した場合、その超過分は、ソケット用ファイル記述子使用数として扱われます。この場合、max_socket_descriptors オペランドの指定値から max_open_fds オペランドの指定値を減算した超過分が、実際のソケット用ファイル記述子の最大数になりますので、ご注意ください。

なお、1 プロセスで使用できるファイル記述子の最大数は 2048 であるため、このオペランドには、次の条件を満たす値を指定してください。

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

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

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

注意事項

max_socket_descriptors オペランドの指定値と max_open_fds オペランドの指定値の合計は、OS のシステムパラメタで指定する「1 プロセスでオープンできるファイル数」を超えないようにする必要があります。システム定義の変更などによって、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

```
mcfostp -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/OSI-TP 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おそれがあるためご注意ください。

TP1/NET/OSI-TP 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>

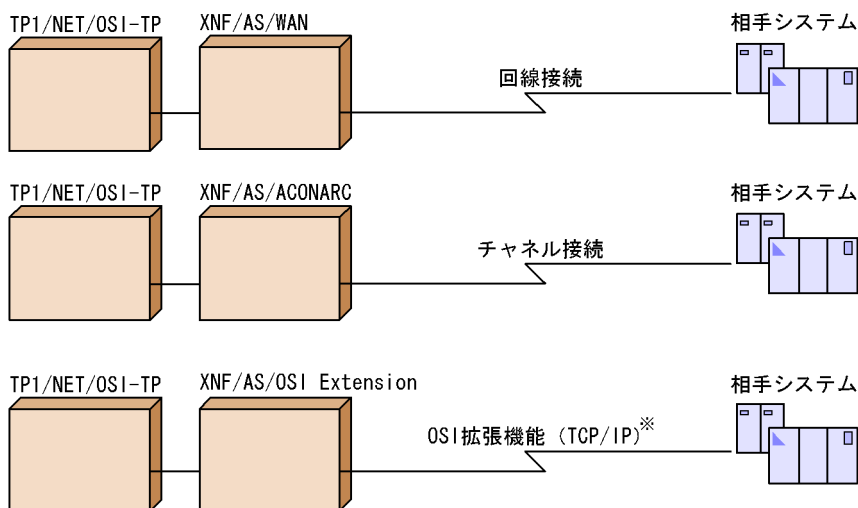
定義オブジェクトを格納するファイル名を指定します。

自システムの通信管理プログラム (XNF/AS) と関連づける内容

相手システムとの接続方法や接続時に使用したい機能によって、使用する通信管理プログラムは次の図に示すとおり異なります。

通信管理プログラムの詳細については、マニュアル「通信管理 XNF/AS 解説・運用編」およびマニュアル「通信管理 XNF/AS 構成定義編」を参照してください。

図 5-2 通信管理プログラムによる相手システムとの接続方法および接続時の機能の違い



注※

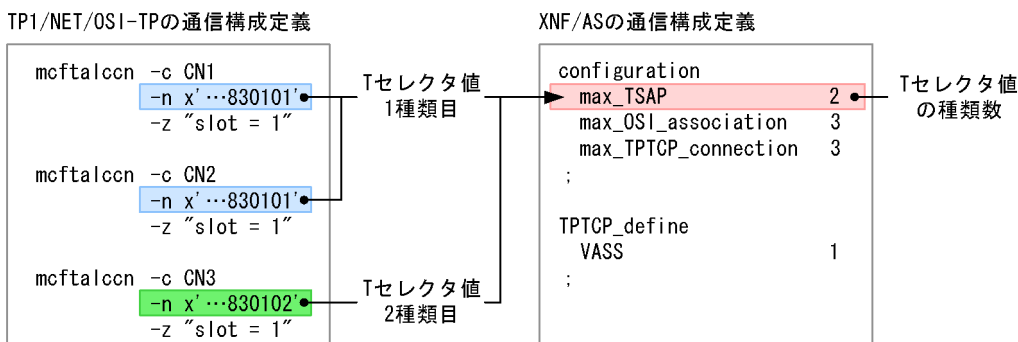
自局IPアドレス指定機能を使用する場合は、XNF/AS/OSI Extension/Clusterが必要です。また、OSI拡張高信頼化機能を使用する場合は、XNF/AS/Host Adaptorが必要です。

図中のどの通信管理プログラムを使用する場合でも、TP1/NET/OSI-TPのMCF通信構成定義 (mcftalccn) で指定する定義の内容と、XNF/ASのシステム定義時に指定する定義の内容とを関連づける必要があります。関連づける必要のある項目を次に示します。

- 最大コネクション数
- Tセレクト値 (種類数)
- スロット番号

注

例えば、次の図に示すように三つのmcftalccnコマンドを定義した場合は、Tセレクト値の種類数である2をXNF/ASの通信構成定義と関連づけます。



関連づける項目は、相手システムとの接続方式および接続時の機能によって異なります。使用する接続方式および接続時の機能別に、次に示します。

回線接続する場合

回線接続する場合は、次の表に示すとおりに定義を関連づけてください。

表 5-3 回線接続する場合の XNF/AS の定義との関連づけ

項番	TP1/NET/OSI-TP での定義内容		XNF/AS での定義内容	
	コマンド	内容	定義文	オペランド
1	mcftalccn	定義数 (コネクション数)	configuration	max_OSI_association
2				max_TC_class02
3	mcftalccn	-n オプションで指定する T セレクタ値の種類数	configuration	max_TSAP
4	mcftalccn	-z オプション (スロット番号)	link	VASS

チャネル接続する場合

チャネル接続する場合は、次の表に示すとおりに定義を関連づけてください。

表 5-4 チャネル接続する場合の XNF/AS の定義との関連づけ

項番	TP1/NET/OSI-TP での定義内容		XNF/AS での定義内容	
	コマンド	内容	定義文	オペランド
1	mcftalccn	定義数 (コネクション数)	configuration	max_OSI_association
2				max_CHANNEL_TC
3	mcftalccn	-n オプションで指定する T セレクタ値の種類数	configuration	max_TSAP

5. システム定義

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

項番	TP1/NET/OSI-TP での定義内容		XNF/AS での定義内容	
	コマンド	内容	定義文	オペランド
4	mcftalcen	-z オプション (スロット番号)	CHANNEL_VC	VASS

OSI 拡張機能を使用する場合

OSI 拡張機能を使用する場合は、次の表に示すとおりに定義を関連づけてください。

表 5-5 OSI 拡張機能を使用する場合の XNF/AS の定義との関連づけ

項番	TP1/NET/OSI-TP での定義内容		XNF/AS での定義内容	
	コマンド	内容	定義文	オペランド
1	mcftalcen	定義数 (コネクション数)	configuration	max_OSI_association
2				max_TPTCP_connection
3	mcftalcen	-n オプションで指定する T セレクタ値の種類数	configuration	max_TSAP
4	mcftalcen	-z オプション (スロット番号)	TPTCP_define	VASS

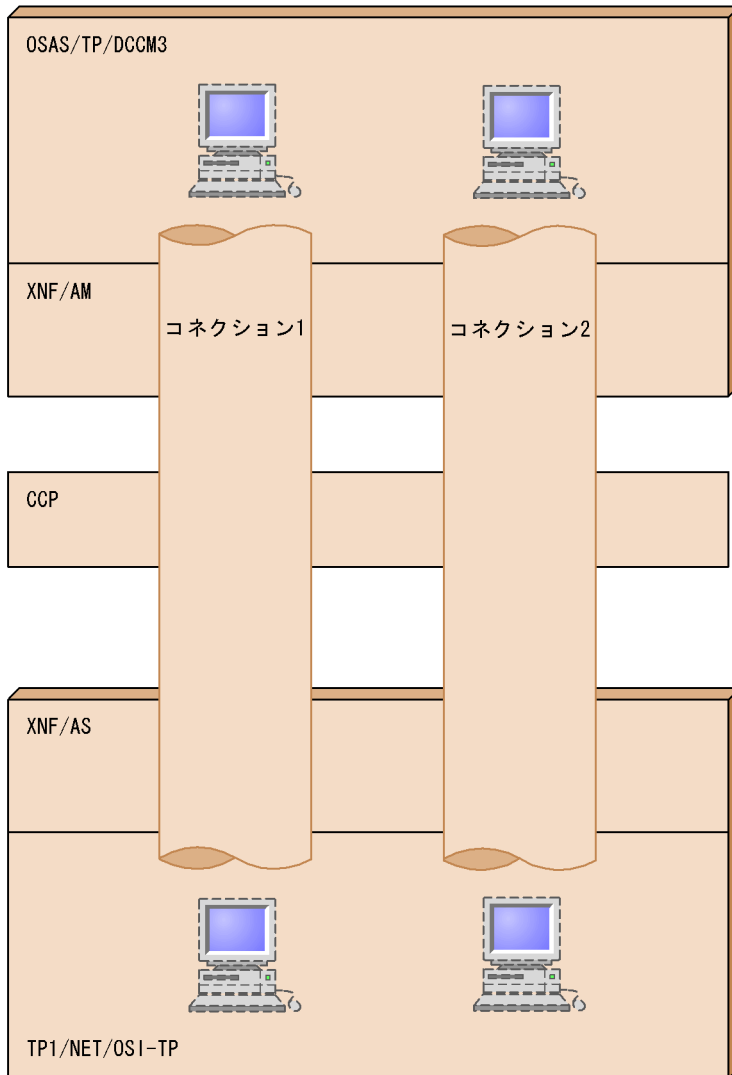
注

自局 IP アドレス指定機能を使用する場合、該当する定義文は、TPTCP_slot になります。また、OSI 拡張高信頼化機能を使用する場合、該当する定義文は、TPTCP_VC になります。

相手システムの通信定義と関連づける内容

TP1/NET/OSI-TP を使用する場合のネットワーク構成の例を次の図に示します。

図 5-3 ネットワーク構成の例



(凡例)
CCP : 通信制御処理装置

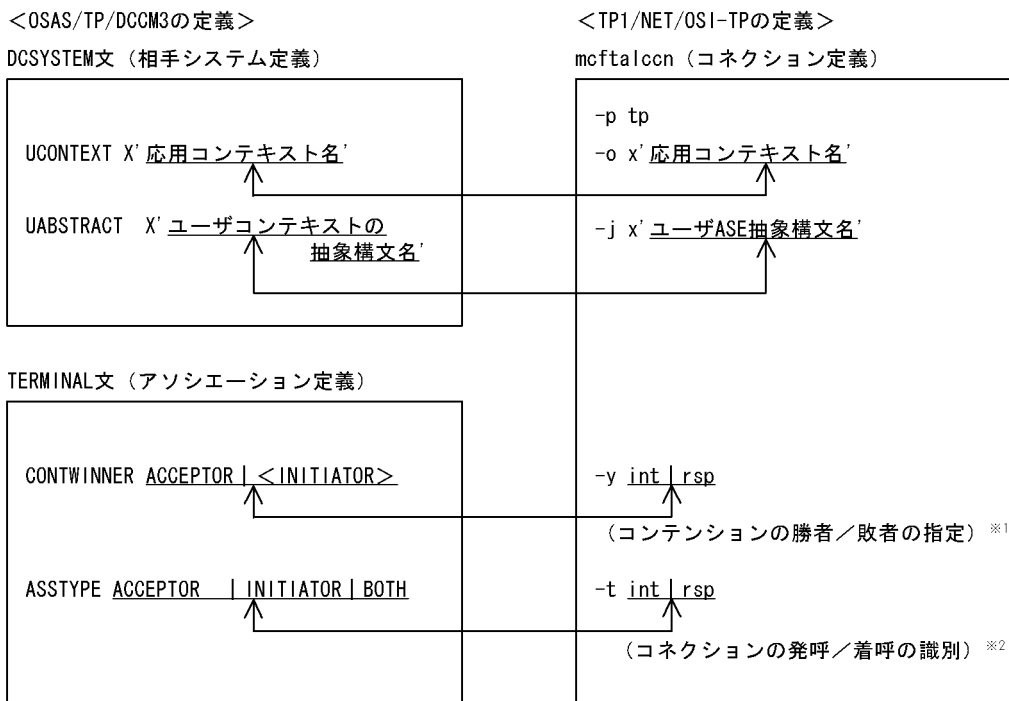
この場合、TP1/NET/OSI-TP は、OSAS/TP/DCCM3 と定義を関連づける必要があります。

5. システム定義

相手システムの通信定義と関連づける内容

通信相手システムが OSAS/TP/DCCM3 の場合の TP1/NET/OSI-TP のシステム定義と OSAS/TP/DCCM3 の定義との関係を次の図に示します。OSAS/TP/DCCM3 の定義については、マニュアル「VOS3 OSAS/TP/DCCM3」を参照してください。

図 5-4 OSAS/TP/DCCM3 の定義との関係



注 1

相手システムの CONTWINNER の値によって、-y オプションの指定が異なります。

- ACCEPTOR の場合、rsp を指定します。
- INITIATOR の場合、int を指定します。

注 2

相手システムの ASSTYPE の値によって、-t オプションの指定が異なります。

- ACCEPTOR の場合、int を指定します。
- INITIATOR の場合、rsp を指定します。
- BOTH の場合、rsp または int を指定します。

定義例

ここでは、TP1/NET/OSI-TP を使用した全二重の場合のシステム定義の例を示します。

TP1/NET/OSI-TP のシステム構成例を図 5-5 に、この構成に沿った定義例をそのあとに示します。

なお、TP1/NET/OSI-TP では、全二重の場合のシステム定義のコーディング例を次のファイルで提供しています。

全二重の場合のシステム定義の例

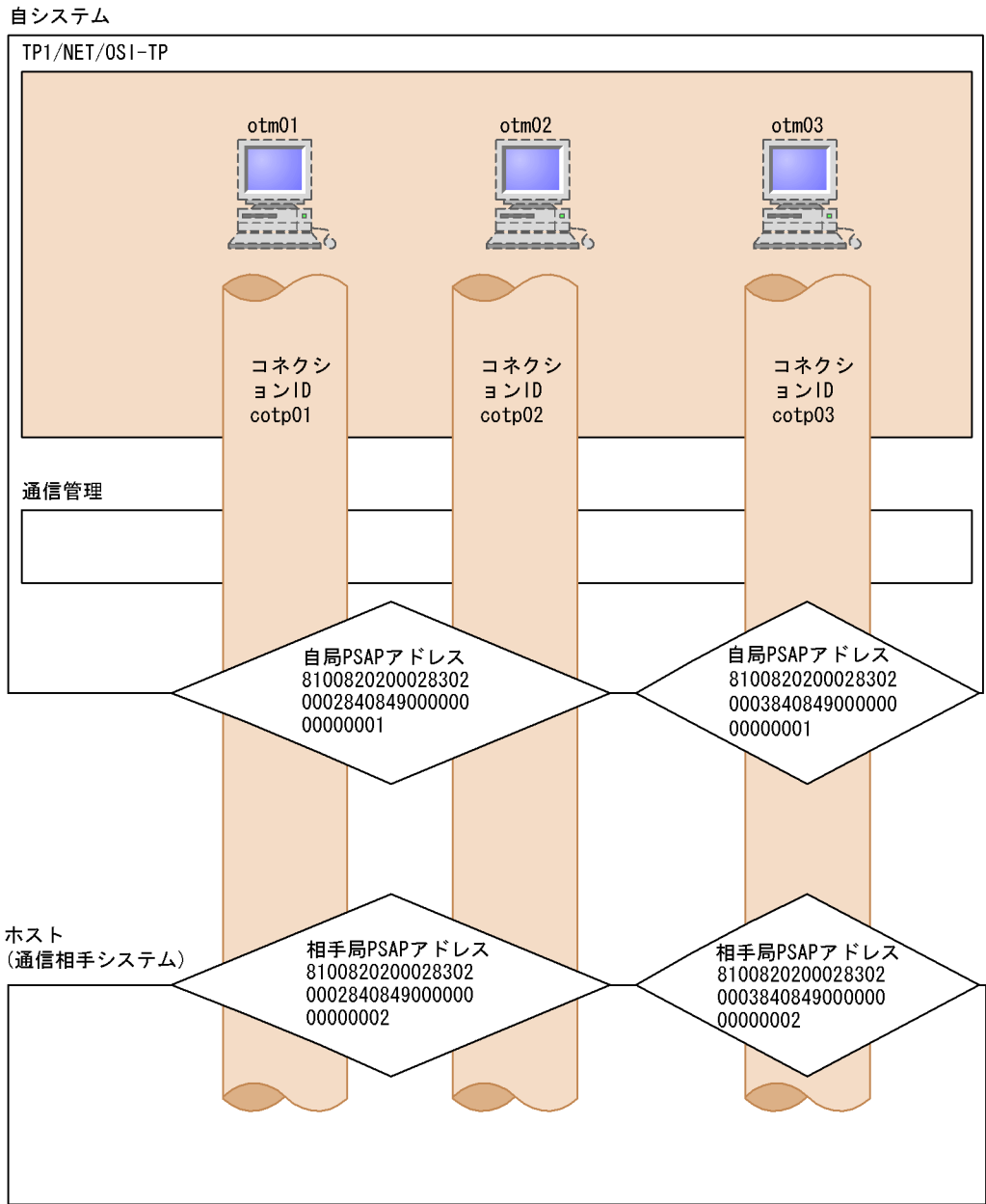
- /BeTRAN/examples/mcf/OSITP/conf/mngr1
- /BeTRAN/examples/mcf/OSITP/conf/com_c1
- /BeTRAN/examples/mcf/OSITP/conf/com_d1

また、半二重の場合のシステム定義では、全二重の場合に指定する `mcfalcen` の定義に次の指定を追加します。

```
-d "control=polarized"
```

5. システム定義
定義例

図 5-5 TP1/NET/OSI-TP のシステム構成例



TP1/NET/OSI-TP の MCF マネージャ環境定義例

```
# ----- ファイル1 MCFマネージャ環境定義 ----- #
mcfmenv    -m "name"          =TPmngt"
#
mcfmcomm   -p 100
#
```

```

mcfmcname -s "mcfsvname =_muTPrsp                ¥
              syssvname =mcfutp"
mcfmcname -s "mcfsvname =_muTPint                ¥
              syssvname =mcfutp"
#
mcfmuap    -t "sndtim=0                            ¥
              sndrcvtim=0                          ¥
              recvtim=0"
#
mcfmsts    -v 60
# ----- ファイル1 MCFマネージャ環境定義 終わり ----- #

```

TP1/NET/OSI-TP の MCF 通信構成定義の例 (共通定義)

```

# ----- ファイル2 MCF通信構成定義 共通定義 ----- #
mcfteenv   -s 01                                    ¥
            -a APCNF01
mcfcomn    -j 32768
mcftrred   -m 10
mcftrtim    -t "btim = 1                            ¥
              mtim = 180                            ¥
              rmtim = 0"
mcftrtrc   -t "size = 20480                          ¥
              disk = yes                            ¥
              bufcnt = 100                          ¥
              trccnt = 3"                            ¥
            -m del
mcftrbuf   -g "groupno=1                            ¥
              length =2048                          ¥
              count =35                             ¥
              extend =1"
mcftrbuf   -g "groupno=2                            ¥
              length =2048                          ¥
              count =35                             ¥
              extend =1"
mcftrbuf   -g "groupno=4                            ¥
              length =32768                          ¥
              count =10"                            ¥
# ----- ファイル2 MCF通信構成定義 共通定義 終わり ----- #

```

TP1/NET/OSI-TP の MCF 通信構成定義の例 (固有定義)

```

# ----- ファイル3 MCF通信構成定義 固有定義 ----- #
# ----- #
###コネクション定義開始###
mcfalccn   -c cotp01                                ¥
            -p tp                                    ¥
            -n x'0a81008202000283020002'          ¥
            -q x'148100820200028302000284084900000000000002' ¥
            -g "sndbuf =1                            ¥
              rcvbuf =2"                            ¥
            -e "msgbuf=4                            ¥
              count=1"                              ¥
            -m "mode=xnfas"                          ¥
            -i auto                                  ¥
            -u ht                                    ¥

```

5. システム定義
定義例

```

-w "nomltim=00" ¥
-b "bretry =yes ¥
   bretrycnt=2 ¥
   bretryint=10" ¥
-t int ¥
-z "slot=3" ¥
-y int ¥
-o x'28ce2a060201' ¥
-j x'28ce2a060101' ¥
# ----- #
###論理端末定義###
mcftalcle -l otm01 ¥
          -t any
# ----- #
###コネクション定義終了###
mcftalced
# ----- #
###コネクション定義開始###
mcftalccn -c cotp02 ¥
          -p tp ¥
          -n x'0a81008202000283020002' ¥
          -q x'1481008202000283020002840849000000000000002' ¥
          -g "sndbuf =1 ¥
             rcvbuf =2" ¥
          -e "msgbuf=4 ¥
             count=1" ¥
          -m "mode=xnfas" ¥
          -i auto ¥
          -u ht ¥
          -w "nomltim=00" ¥
          -b "bretry =yes ¥
             bretrycnt=2 ¥
             bretryint=10" ¥
          -t int ¥
          -z "slot=3" ¥
          -y int ¥
          -o x'28ce2a060201' ¥
          -j x'28ce2a060101' ¥
# ----- #
###論理端末定義###
mcftalcle -l otm02 ¥
          -t any
# ----- #
###コネクション定義終了###
mcftalced
# ----- #
###コネクション定義開始###
mcftalccn -c cotp03 ¥
          -p tp ¥
          -n x'0a81008202000283020003' ¥
          -q x'1481008202000283020003840849000000000000002' ¥
          -g "sndbuf =1 ¥
             rcvbuf =2" ¥
          -e "msgbuf=4 ¥
             count=1" ¥

```



```

-m "mode=xnfas" ¥
-i auto ¥
-u ht ¥
-w "nomltim=00" ¥
-b "bretry =yes ¥
    bretrycnt=2 ¥
    bretryint=10" ¥
-t int ¥
-z "slot=3" ¥
-y int ¥
-o x'28ce2a060201' ¥
-j x'28ce2a060101' ¥
# ----- #
###論理端末定義###
mcftalcle -l otm03 ¥
          -t any
# ----- #
###コネクション定義終了###
mcftalced
# ----- ファイル3 MCF通信構成定義 固有定義 終わり ----- #

```


6

運用コマンド

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

TP1/NET/OSI-TP の運用コマンド

mcftactcn (コネクションの確立)

mcftdctcn (コネクションの解放)

mcftlscn (コネクションの状態表示)

TP1/NET/OSI-TP の運用コマンド

ここでは、TP1/NET/OSI-TP で使用する運用コマンドについて説明します。

ここでは、TP1/NET/OSI-TP 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/OSI-TP で使用する運用コマンドを、次の表に示します。

表 6-1 TP1/NET/OSI-TP で使用する運用コマンド

機能	コマンド名称	定義 中組 み込 み	オフ ライ ン 中 に 実 行	オン ライ ン 中 に 実 行	UAP から 実行
コネクション管理	コネクションの確立	mefctacten	×	×	
	コネクションの解放	meftdcten	×	×	
	コネクション、および 論理端末の状態表示	meftlsen	×	×	

(凡例)

：組み込み、または実行ができます。

×：組み込み、または実行ができません。

mcfactcn (コネクションの確立)

形式

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

機能

コネクションを確立します。

オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数を指定できません。

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、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名称を指定します。

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

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

* : すべてのコネクションを確立します。

先行文字列 * : 先行文字列で始まるすべてのコネクションを確立します。

複数指定の例 cnn1, cnn2, cnn3 を指定する場合

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

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

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

-g コネクショングループ名 ~ < 1 ~ 8 文字の識別子 >

確立するコネクショングループのコネクショングループ名を指定します。

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

6. 運用コマンド

mcftactcn (コネクションの確立)

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

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

* : すべてのコネクショングループ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グループのコネクションを確立します。

複数指定の例 gnn1, gnn2, gnn3 を指定する場合

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

一括指定の例 gnn で始まるすべてのコネクショングループを指定する場合

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

出力メッセージ

メッセージ ID	内容	出力先
KFCA10350-I	mcftactcn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftactcn コマンド入力元への応答を失敗しました。	メッセージログ ファイル
KFCA10371-I	mcftactcn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftactcn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10500-I	ヘルプメッセージ	標準出力
KFCA13930-E	MCF 運用コマンド処理中に障害が発生しました。	標準エラー出力
KFCA13932-E	コネクションが確立済みのため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13933-E	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

6. 運用コマンド
mcfactcn (コネクションの確立)

メッセージ ID	内容	出力先
KFCA13934-E	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
KFCA13942-E	着呼モードのコネクションのため、運用コマンドは受け付けられません。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	mcfactcn コマンドで指定したコネクショングループ名は登録されていません。	標準エラー出力

mcftdctcn (コネクションの解放)

形式

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

機能

コネクションを解放します。

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を指定できません。

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

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

* : すべてのコネクションを解放します。

先行文字列 * : 先行文字列で始まるすべてのコネクションを解放します。

複数指定の例 cnn1, cnn2, cnn3 を指定する場合

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

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

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

-g コネクショングループ名 ~ < 1 ~ 8 文字の識別子 >

解放するコネクショングループのコネクショングループ名を指定します。

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

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

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

* : すべてのコネクショングループ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グループのコネクションを解放します。

複数指定の例 gnn1, gnn2, gnn3 を指定する場合

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

一括指定の例 gnn で始まるすべてのコネクショングループを指定する場合

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

```
-f
```

該当するコネクションを強制的に解放します。

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強制的に解放します。

出力メッセージ

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

6. 運用コマンド

mcftdctn (コネクションの解放)

メッセージ ID	内容	出力先
KFCA10501-I	ヘルプメッセージ	標準出力
KFCA13930-E	MCF 運用コマンド処理中に障害が発生しました。	標準エラー出力
KFCA13931-E	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
KFCA13933-E	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
KFCA13934-E	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
KFCA13941-E	コネクション使用中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	mcftdctn コマンドで指定したコネクショングループ名は登録されていません。	標準エラー出力

mcftlscn (コネクションの状態表示)

形式

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

機能

コネクションの状態を標準出力で表示します。

オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は、複数を指定できません。

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

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

* : すべてのコネクションの状態を表示します。

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

複数指定の例 cnn1 , cnn2 , cnn3 を指定する場合

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

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

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

-g コネクショングループ名 ~ < 1 ~ 8 文字の識別子 >

状態を表示するコネクショングループのコネクショングループ名を指定します。

6. 運用コマンド

mcftlscn (コネクションの状態表示)

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

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

*:すべてのコネクショングループにある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グループにあるコネクションの状態を表示します。

複数指定の例 gnn1, gnn2, gnn3 を指定する場合

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

一括指定の例 gnn で始まるすべてのコネクショングループを指定する場合

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

```
-d
```

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

このオプションの指定を省略すると、コネクションの状態だけを表示します。

出力形式

```
mmm cccccccc ppp sssss dddd gggggggg  
lllllllll ttt aaaa bbbb
```

注

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

- mmm : MCF 識別子
- cccccccc : コネクション ID
- ppp : プロトコル種別
TP...OSI TP 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)
- gggggggg : コネクショングループ名 (コネクショングループ配下のコネクションの場合だけ表示)
- llllllll : 論理端末名称

- ttt : 論理端末タイプ
- aaaa : 詳細ステータスおよびイベント (保守情報)
- bbbb : 詳細ステータスおよびイベント (保守情報)

出力メッセージ

メッセージ ID	内容	出力先
KFCA10350-I	mcfllscn 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	mcfllscn 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	mcfllscn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcfllscn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10502-I	ヘルプメッセージ	標準出力
KFCA13930-E	TP1/NET/OSI-TP で論理エラーが発生しました。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	mcfllscn コマンドで指定したコネクショングループ名は登録されていません。	標準エラー出力

7

組み込み方法

この章では、TP1/NET/OSI-TP 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/OSI-TP 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/OSI-TP の組み込みの流れ

TP1/NET/OSI-TP 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/OSI-TP 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/OSI-TP 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. システム定義」の「システムサービス情報定義」を参照してください。

(4) 定義オブジェクトファイルの生成

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/OSI-TP は、OpenTP1 プロセスサービスによって起動されます。

TP1/NET/OSI-TP を起動するためには、ユーザが MCF メイン関数をコーディングし、コンパイル、およびリンケージを行って TP1/NET/OSI-TP 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、mcfpltp 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、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を次のファイルで提供しています。

- /BeTRAN/examples/mcf/OSITP/cmlib/ansi/com.c
- /BeTRAN/examples/mcf/OSITP/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 <dcmotp.h>                /*TP1/NET/OSI-TP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 <dcmotp.h>                /*TP1/NET/OSI-TP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/OSI-TP 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。

```

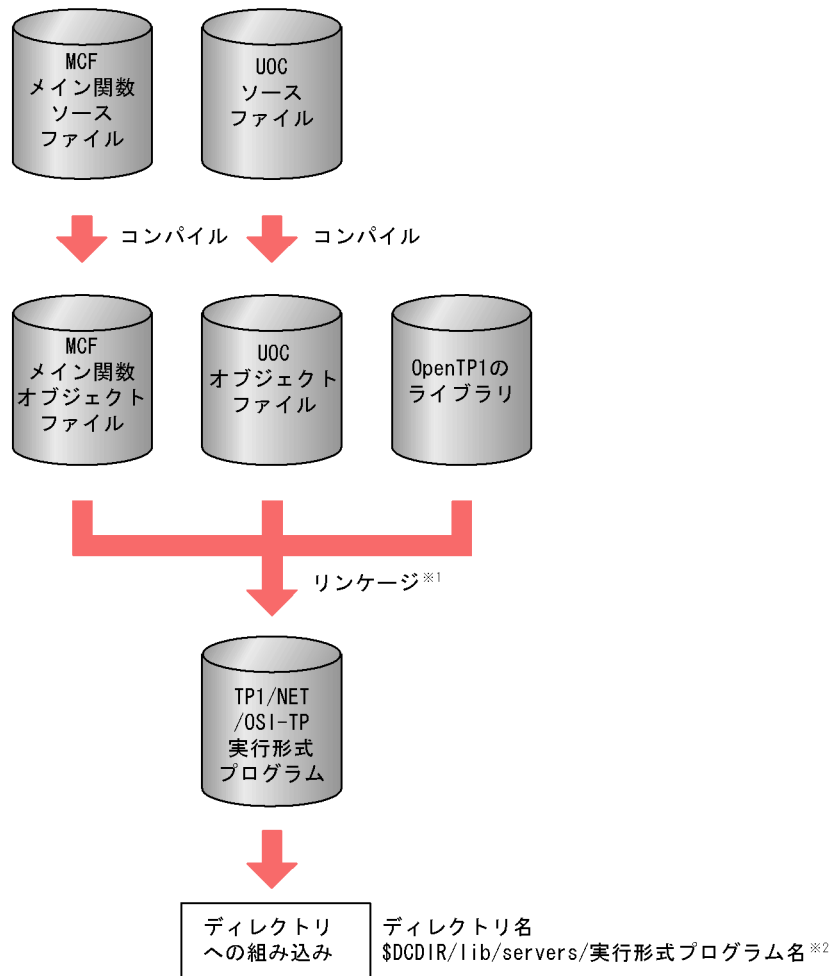
dcmcf_uoctbl.msgrcv /*入力メッセージ編集UOCアドレス*/
dcmcf_uoctbl.msgsend /*出力メッセージ編集UOCアドレス*/

```

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の概要



注 1
mcfpltp コマンドでリンケージします。

注 2
TP1/NET/OSI-TP 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

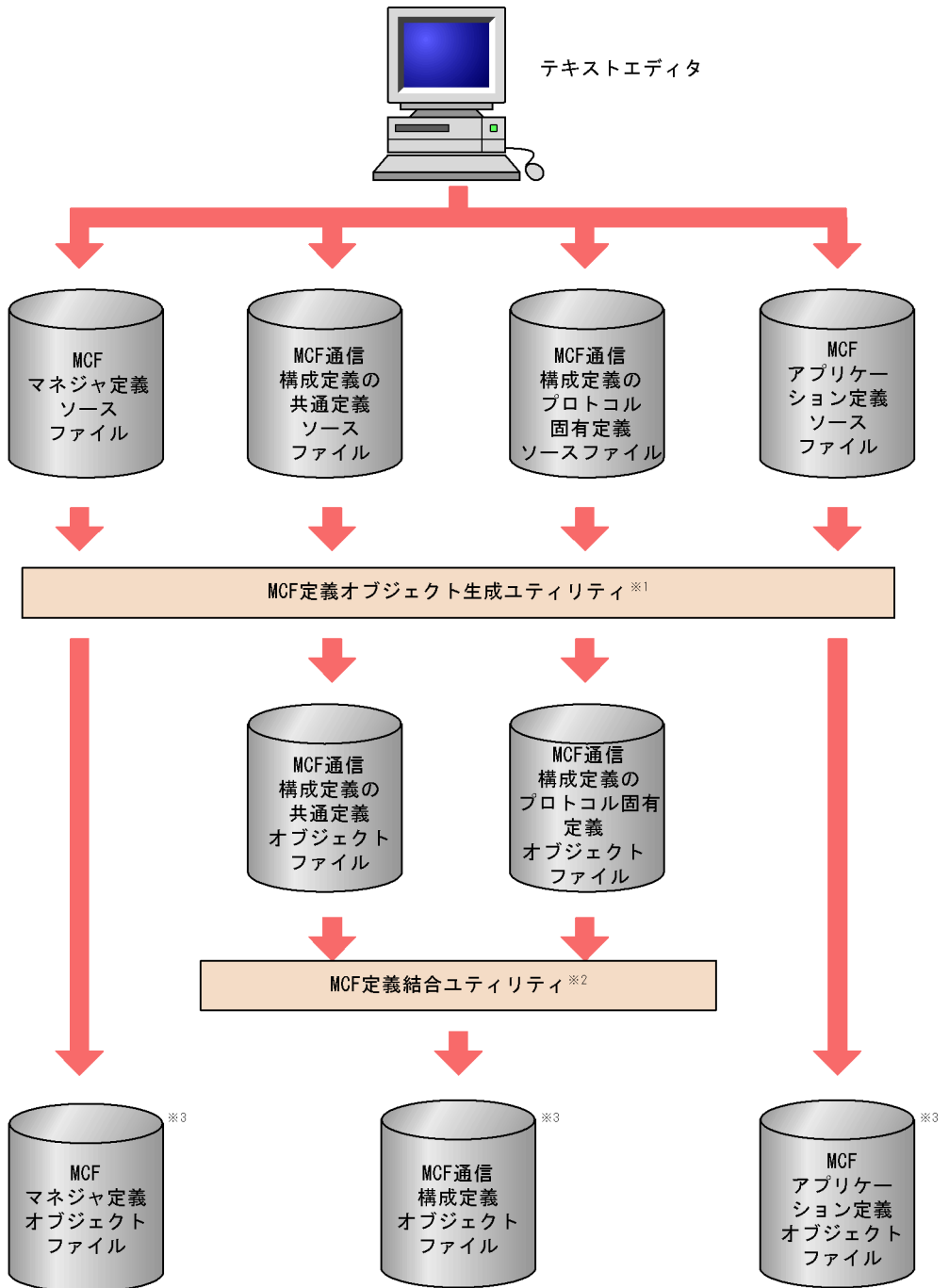
定義オブジェクトファイルを次の手順で生成します。

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おそれがあるためご注意ください。

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

定義オブジェクトファイルの作成方法の概要を次の図に示します。

図 7-4 定義オブジェクトファイルの作成方法の概要



注 1

7. 組み込み方法

次に示すコマンドで生成します。

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

mcfXXXX は、ソースファイルごとに異なります。

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

注 2

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

```
mcflink -i 共通定義オブジェクトファイル名
         TP1/NET/OSI-TP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/OSI-TP の運用中に発生するおそれがある障害と、TP1/NET/OSI-TP 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 ダイアログ障害

8.4 ユーザアプリケーションプログラム異常終了

8.5 ユーザアプリケーションプログラム閉塞

8.1 障害の種類と対応処理

TP1/NET/OSI-TP 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障害（入力キュー，入力メッセージ編集 UOC）
- 送信スケジュール関係障害（出力キュー，出力メッセージ編集 UOC）
- UAP 障害
- ジャーナル障害（IJ，MJ）
- 関数のリターン処理障害
- プロシジャ障害

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

TP1/NET/OSI-TP 運用中の障害と対応処理を次に示します。

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

（1）コネクション障害

表 8-1 コネクション障害と対応処理

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
下位層障害 リトライオーバ (発呼時)	<ol style="list-style-type: none"> 1. TP-P-ABORTind (recvsync 中) を通知します。 2. CERREVT を起動し、下位層障害を通知する理由コード 1「ABORT」、理由コード 2「XNF」を出力します。 3. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 	<p>障害の要因を取り除いたあと、次に示す処理をします。</p> <ul style="list-style-type: none"> • 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
TP プロトコ ル障害	<ol style="list-style-type: none"> 1. TP-P-ABORTind (recvsync 中) を通知します。 2. TP-ABORT-RI provider を送信します。 3. CERREVT を起動し、TP プロトコル障害を通知する理由コード 1「ABORT」、理由コード 2「OTP」を出力します。 4. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 	<p>障害の要因を取り除いたあと、次に示す処理をします。</p> <ul style="list-style-type: none"> • 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。

(2) 受信スケジュール関係障害（入力キュー、入力メッセージ編集 UOC）

表 8-2 受信スケジュール関係の障害と対応処理

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
TP-BEGIN-DIALOGUEind 受信時の UOC エラー リターン	<ol style="list-style-type: none"> 1. TP-ABORT-RI provider を送信します。 2. CERREVT を起動し、理由コード 1「UOC」、理由コード 2「詳細リターンコード」を出力します。 3. メッセージ入力障害を通知するメッセージログ (KFCA10604-E)、および UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 4. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 	<ul style="list-style-type: none"> • 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
TP-BEGIN-DIALOGUEind 以外の UOC エラーリターン	<ol style="list-style-type: none"> 1. TP-P-ABORTind (recvsync 中) を通知します。 2. TP-ABORT-RI provider を送信します。 3. CERREVT を起動し、理由コード 1「UOC」、理由コード 2「詳細リターンコード」を出力します。 4. 同期メッセージ応答障害を通知するメッセージログ (KFCA10606-E)、および UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 5. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 	<ul style="list-style-type: none"> • 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
UOC パラメ タ指定誤り	<ol style="list-style-type: none"> 1. TP-P-ABORTind (recvsync 中) を通知します。 2. TP-ABORT-RI provider を送信します。 3. CERREVT を起動し、理由コード 1「UOC」、理由コード 2「詳細リターンコード」を出力します。 4. 同期メッセージ応答障害を通知するメッセージログ (KFCA10606-E)、および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。 5. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 	<p>UOC を修正後、次に示す処理をします。</p> <ul style="list-style-type: none"> • 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
アプリケー ション名未定 義または形式 不正	<ol style="list-style-type: none"> 1. TP-ABORT-RI provider を送信します。 2. ERREVT1 を起動し、理由コード 1「MCF」理由コード 2「APL」を出力します。 3. アプリケーション名取得失敗を通知するメッセージログ (KFCA10610-E) を出力します。 4. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 	<p>システム定義を修正後、次に示す処理をします。</p> <ul style="list-style-type: none"> • 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
MCF イベント名未定義 • ERREVTn • CXXXEVT	処理を続行します。	ありません。

8. 障害対策

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
入力キュー書き込み障害 スケジュール障害 UAP 閉塞 MHP サービス、サービスグループ閉塞	<ol style="list-style-type: none"> 1. TP-ABORT-RI provider を送信します。 2. CERREVT を起動し、理由コード 1「MCF」、理由コード 2「ITQ」を出力します。 3. メッセージ入力障害を通知するメッセージログ (KFCA10604-E) を出力します。 4. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 5. ERREVT2 を起動します。 	障害の要因を取り除いたあと、次に示す処理をします。 <ul style="list-style-type: none"> • 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
受信バッファオーバーフロー	<ol style="list-style-type: none"> 1. TP-ABORT-RI provider を送信します。 2. CERREVT を起動し、理由コード 1「MCF」、理由コード 2「NOBUF」を出力します。 3. UAP にエラーリターンします。 4. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 	システム定義を修正後、次に示す処理をします。 <ul style="list-style-type: none"> • 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
編集バッファオーバーフロー	ありません。	UOC で、必要に応じてエラーリターンします。
バッファ数不足 (処理 1 か処理 2 のどちらかを実行します)	処理 1 <ol style="list-style-type: none"> 1. TP-ABORT-RI provider を送信します。 2. CERREVT を起動し、理由コード 1「MCF」、理由コード 2「NOBUF」を出力します。 3. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 	システム定義を修正後、次に示す処理をします。 <ul style="list-style-type: none"> • 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
	処理 2 <ol style="list-style-type: none"> 1. 資源不足を通知するメッセージログ (KFCA13212-E) を出力します。 2. MCF プロセスを異常終了します。 	

(3) 送信スケジュール関係障害（出力キュー，出力メッセージ編集 UOC）

表 8-3 送信スケジュール関係の障害と対応処理

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
UOC エラーリターン	<ol style="list-style-type: none"> 1. TP-ABORT-RI provider を送信します。 2. CERREVT を起動し，理由コード 1「MCF」，理由コード 2「OTGET」を出力します。 3. メッセージ出力障害を通知するメッセージログ (KFCA10605-E)，および UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 4. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 5. UAP にエラーリターンします。 	<ul style="list-style-type: none"> • 発呼時 mcfactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
UOC パラメータ指定誤り	<ol style="list-style-type: none"> 1. TP-ABORT-RI provider を送信します。 2. CERREVT を起動し，理由コード 1「MCF」，理由コード 2「OTGET」を出力します。 3. メッセージ出力障害を通知するメッセージログ (KFCA10605-E)，およびパラメータ不正を通知するメッセージログ (KFCA10620-E) を出力します。 4. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 5. UAP にエラーリターンします。 	<p>UOC を修正後，次に示す処理をします。</p> <ul style="list-style-type: none"> • 発呼時 mcfactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
メッセージ消し込み障害	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"> 1. TP-ABORT-RI provider を送信します。 2. CERREVT を起動し，理由コード 1「ABORT」，理由コード 2「NOBUF」を出力します。 3. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 	<p>システム定義を修正後，次に示す処理をします。</p> <ul style="list-style-type: none"> • 発呼時 mcfactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
編集バッファオーバーフロー	ありません。	UOC で，必要に応じてエラーリターンします。
バッファ数不足 (処理 1 か処理 2 のどちらかを実行します)	<p>処理 1</p> <ol style="list-style-type: none"> 1. TP-ABORT-RI provider を送信します。 2. CERREVT を起動し，理由コード 1「MCF」，理由コード 2「NOBUF」を出力します。 3. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 	<p>システム定義を修正後，次に示す処理をします。</p> <ul style="list-style-type: none"> • 発呼時 mcfactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。

8. 障害対策

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
	処理 2 1. 資源不足を通知するメッセージログ (KFCA13212-E) を出力します。 2. MCF プロセスを異常終了します。	

(4) UAP 障害

表 8-4 UAP の障害と対応処理

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
ダイアログ中の UAP 異常終了 ダイアログ完了前の UAP 終了 (MHP, SPP)	1. TP-ABORT-RI provider を送信します。 2. CERREVT を起動し、理由コード 1「MCF」、理由コード 2「UAPAB」を出力します。 3. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 4. ERREVT3 を起動します。	障害の要因を取り除いたあと、次に示す処理をします。 <ul style="list-style-type: none"> • 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。
パラメタエラー プリミティブシーケンスエラー	エラーリターンします。	UAP を修正してください。

(5) ジャーナル障害 (IJ, MJ)

表 8-5 ジャーナルの障害と対応処理

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
ジャーナル取得障害	処理を続行します。	ありません。

(6) 関数のリターン処理障害

表 8-6 関数のリターン処理の障害と対応処理

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
リモートプロシジャコール障害	1. TP-ABORT-RI provider を送信します。 2. CERREVT を起動し、理由コード 1「MCF」、理由コード 2「SYCER」を出力します。 3. 同期メッセージ応答障害を通知するメッセージログ (KFCA10606-E) を出力します。 4. コネクション障害を通知するメッセージログ (KFCA13902-E) を出力します。 5. UAP (XXXXsync0) にエラーリターンします。	• 発呼時 mcftactcn コマンドを入力してコネクションを確立します。 • 着呼時 コネクションの確立を待ちます。

(7) プロシジャ障害

表 8-7 プロシジャの障害と対応処理

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
内部論理矛盾	1. 処理続行を通知するメッセージログ (KFCA13997-E), コネクション強制解放を通知するメッセージログ (KFCA13998-E), または MCF 異常終了を通知するメッセージログ (KFCA13999-E) を出力します。 2. 必要に応じて MCF を異常終了させます。	保守情報 (\$DCDIR/spool ディレクトリ以下) を退避してください。
プロセス異常終了	1. OS によるコア出力をします。 2. 通信管理によるコネクション解放をします。	

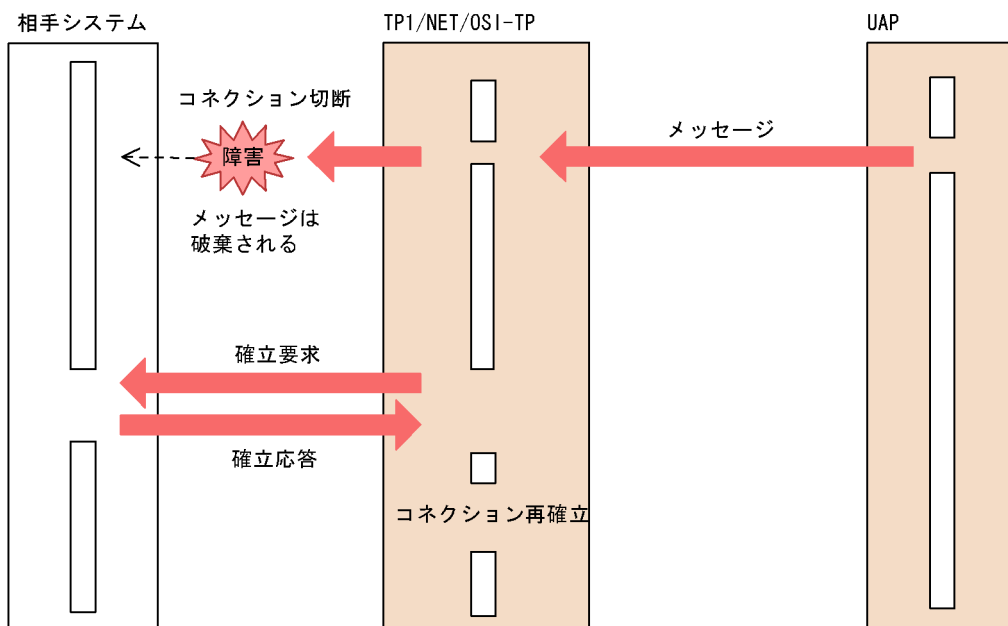
8.2 コネクション障害

コネクションが切断された場合、発呼型のシステムでは、運用コマンド（mcftacten）を入力してコネクションを再確立します。着呼型のシステムでは、自動的に相手システムからのコネクション確立を待ちます。自システムで管理している通信機器に障害が発生した場合も、同様にコネクション障害として処理します。

メッセージ送受信時にコネクション障害が発生した場合、メッセージは破棄されます。

メッセージ送受信時のコネクション障害の例を次の図に示します。

図 8-1 メッセージ送受信時のコネクション障害

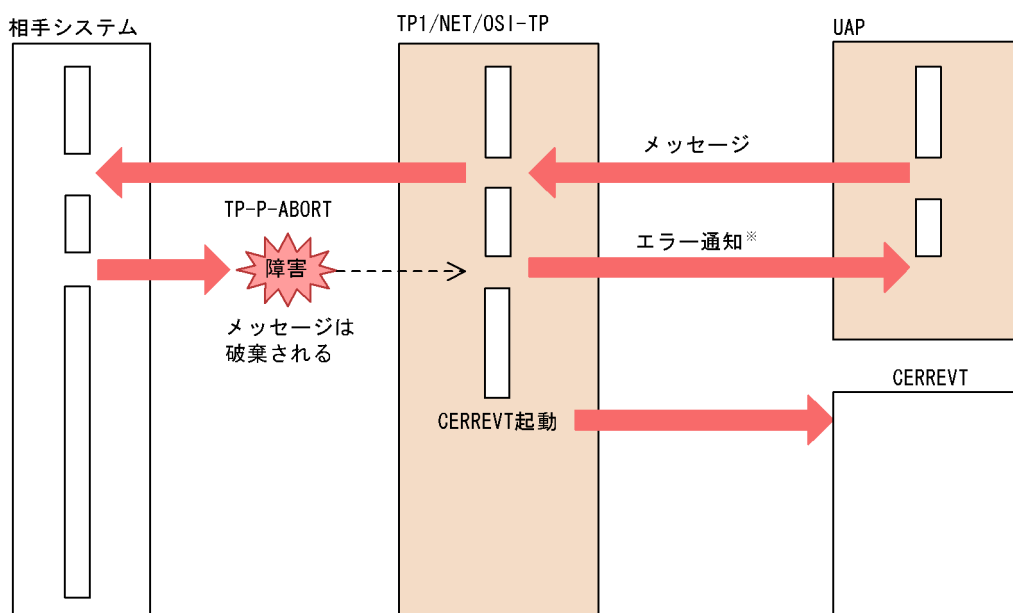


8.3 ダイアログ障害

ダイアログ中に障害が発生した場合，TP1/NET/OSI-TP は UAP にエラー通知をしたあと，CERREVT を起動します。メッセージは，コネクション障害時と同様に破棄されます。

ダイアログ障害の例を，次の図に示します。

図 8-2 ダイアログ障害



注※

エラー通知 (TP-P-ABORTind) は障害が発生した時点で，UAPがrecvsync，または，sendrecvを呼び出している場合だけ通知されます。

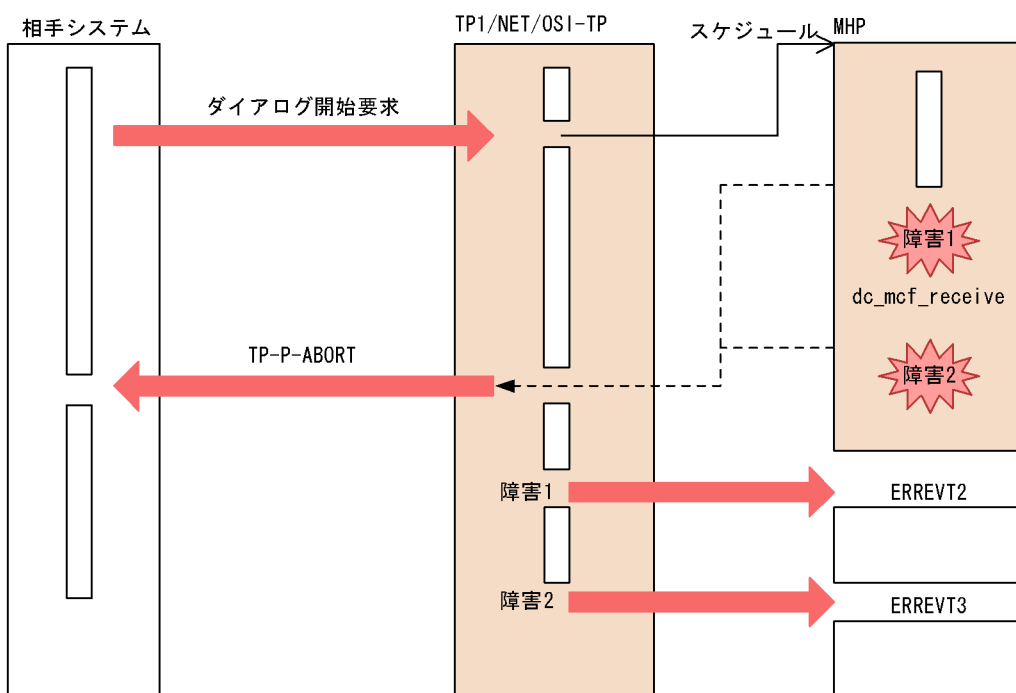
8.4 ユーザアプリケーションプログラム異常終了

UAP が異常終了した場合、TP1/NET/OSI-TP は相手システムに異常終了を通知し、MCF イベントに対応する MHP を起動します。このとき、メッセージは破棄されます。

(1) ダイアログ開始要求受信時の MHP 異常終了

TP1/NET/OSI-TP がダイアログ開始要求受信時、MHP をスケジュールしたあとに MHP が異常終了した場合、ERREVT2 または ERREVT3 が起動されます。ダイアログ開始要求受信時の MHP 異常終了の例を次の図に示します。

図 8-3 ダイアログ開始要求受信時の MHP 異常終了



(2) ダイアログ中の UAP (MHP, SPP) 終了

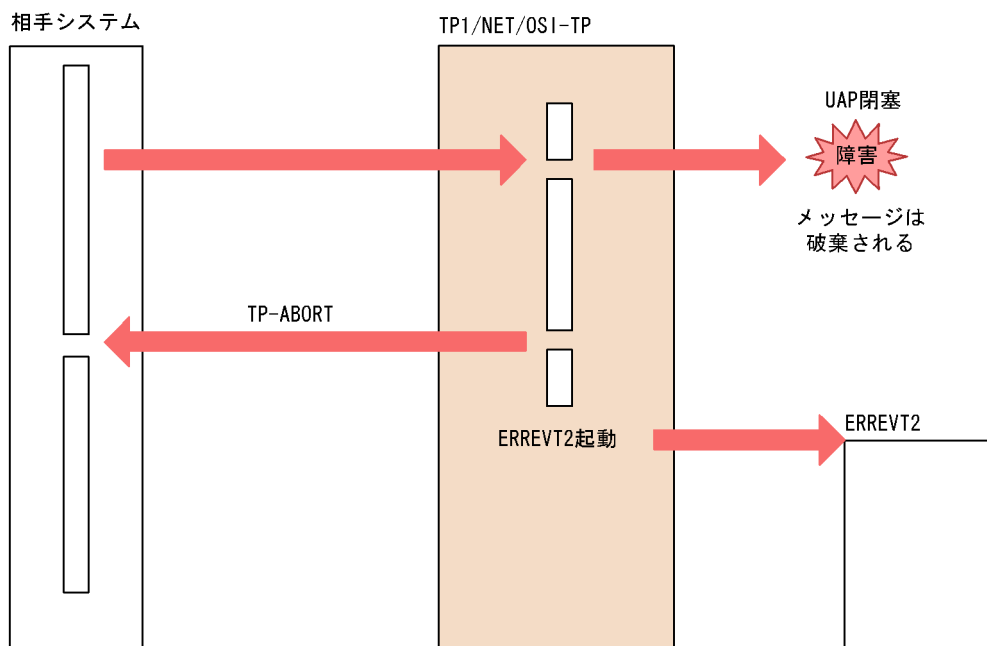
ダイアログ中に UAP (MHP, SPP) が終了 (正常・異常) した場合、TP1/NET/OSI-TP は接続を切断 (TP-P-ABORT) し、CERREVT を起動します。

8.5 ユーザアプリケーションプログラム閉塞

UAP が閉塞していて使用できない場合，TP1/NET/OSI-TP は相手システムに対して UAP 閉塞による異常終了を通知し，ERREVT2 を起動します。このとき，メッセージは破棄されます。

UAP 閉塞による障害の例を次の図に示します。

図 8-4 UAP 閉塞による障害



付録

付録 A TP1/NET/OSI-TP の実装範囲

付録 B メッセージ送受信の処理の流れ

付録 C 障害発生時の処理の流れ

付録 D 理由コード一覧

付録 E ユーザアプリケーションプログラムのコーディング例

付録 F 旧製品からの移行に関する注意事項

付録 G バージョンアップ時の変更点

付録 H 用語解説

付録 A TP1/NET/OSI-TP の実装範囲

TP1/NET/OSI-TP の実装範囲を次に示します。

(1) 実装範囲

OSI/TP プロトコル IS 版 (ISO/IEC 10026-1, 2, 3)

- カーネル機能
- 全二重機能
- 半二重機能
- ハンドシェイク機能

(2) サービスプリミティブ

表 A-1 TP1/NET/OSI-TP のサービスプリミティブおよび PDU

サービス名	プリミティブコード				TP-APDU 名
	req	ind	rsp	cnf	
TP-BEGIN-DIALOGUE					<ul style="list-style-type: none"> • TP-BEGIN-DIALOGUE-RI • TP-BEGIN-DIALOGUE-RC
TP-END-DIALOGUE					<ul style="list-style-type: none"> • TP-END-DIALOGUE-RI • TP-END-DIALOGUE-RC
TP-U-ERROR			-	-	<ul style="list-style-type: none"> • TP-U-ERROR-RI • TP-U-ERROR-RC
TP-U-ABORT			-	-	TP-U-ABORT-RI
TP-P-ABORT	-		-	-	TP-P-ABORT-RI
TP-HANDSHAKE					<ul style="list-style-type: none"> • TP-HANDSHAKE-RI • TP-HANDSHAKE-RC
TP-DATA			-	-	(P-DATA に対応)
TP-GRANT-CONTROL			-	-	TP-GRANT-CONTROL-RI
TP-REQUEST-CONTROL			-	-	TP-REQUEST-CONTROL-RI
TP-HANDSHAKE-AND-GRANT-CONTROL					<ul style="list-style-type: none"> • TP-HANDSHAKE-AND-GRANT-CONTROL-RI • TP-HANDSHAKE-AND-GRANT-CONTROL-RC

(凡例)

- : 該当するサービスで指定できます。
- : 該当するサービスで指定できません。

(3) サービス詳細

表 A-2 TP1/NET/OSI-TP のサービス詳細

サービス名 (省略時)	パラメタ	プリミティブの設定値および要否				設定内容
		req	ind	rsp	cnf	
TP-BEGIN-DIALOGUE (B-D)	Initiating-A P-Title	-	-	-	-	B-Dreq 送信元の AEI に関する情報
	Initiating-A PI-Identifier	-	-	-	-	
	Initiating-A E-Qualifier	-	-	-	-	
	Initiating-A EI-Identifier	-	-	-	-	
	Initiating-T PSU-Title		(=)	-	-	B-Dreq 送信元の TPSUI 識別情報
	Recipient-A P-Title		-	-	-	リモート TPSUI が存在 する AEI に関する情報
	Recipient-A PI-Identifier	-	-	-	-	
	Recipient-A E-Qualifier	-	-	-	-	
	Recipient-A EI-Identifier	-	-	-	-	
	Recipient-T PSU-Title		-	-	-	リモート TPSUI 識別情 報
	Functional- Units		(=)	-	1	"Dialogue" "Shared Control" (全二 重) "Polarized Control" (半 二重) "Handshake"
	Quality-of-S ervice		-	-	-	サービス品質に関する 情報
	Application- Context-na me		-	-	-	アプリケーションコン テキスト名に関する情 報
Confirmatio n		(=)	-	-	"always" (応答要) "negative" (拒否時だけ 応答要)	
Result	-	-	2		"accepted" "rejected(provider)" "rejected(user)"	

サービス名 (省略時)	パラメタ	プリミティブの設定値および要否				設定内容
		req	ind	rsp	cnf	
	Diagnostic	-	-	-	3	"recipient-unknown" 4 "recipient-tpsu-title-un known" "tpsu-not-available(per manent)" "tpsu-not-available(tran sient)" "recipient-tpsu-title-re quired" "functional-unit-not-su pported" "functional-unit-combi nation-not-supported" "no-reason-given"
	Rollback	-	-	-		"false"
	User-Data		(=)			ユーザ情報
TP-END-DIALOG UE (E-D)	Confirmatio n		(=)	-	-	"true" (応答要) "false" (応答不要)
TP-U-ERROR (U-ERR)	-	-	-	-	-	-
TP-U-ABORT (U-ABT)	Rollback	-		-	-	"false"
	User-Data		(=)	-	-	ユーザ情報
TP-P-ABORT (P-ABT)	Diagnostic	-		-	-	"permanent-failure" "begin-transaction-reje ct" "transient-failure" "protocol-error" "end-dialogue-collision"
	Rollback	-		-	-	"false"
TP-HANDSHAKE (TP-H)	Confirmatio n-Urgency	5	-	-	-	"urgent" (即時応答要) "normal" (遅延応答可)
TP-GRANT-CON TROL(G-C)	-	-	-	-	-	-
TP-REQUEST-CO NTROL(R-C)	-	-	-	-	-	-
TP-HANDSHAKE -AND-GRANT-CO NTROL(H-G)	Confirmatio n-Urgency		-	-	-	"urgent" (即時応答要) "normal" (遅延応答可)

(凡例)

: 必ず指定します。

- : ユーザで任意に指定できます。
- : 領域だけを設定します。
- : 条件が一致した場合指定できます。
- (=): req で指定した値と同じ値が設定されます。
- : 該当しません。

注 1

Result が "rejected(provider)" で、かつ Diagnostic が "functional-unit-not-supported" の場合に指定します。

注 2

"rejected(provider)" は指定できません。

注 3

Result が "rejected(provider)" の場合に指定します。

注 4

"recipient-unknown" は通知されません。

注 5

Functional-Units が "Shared Control" の場合に指定します。

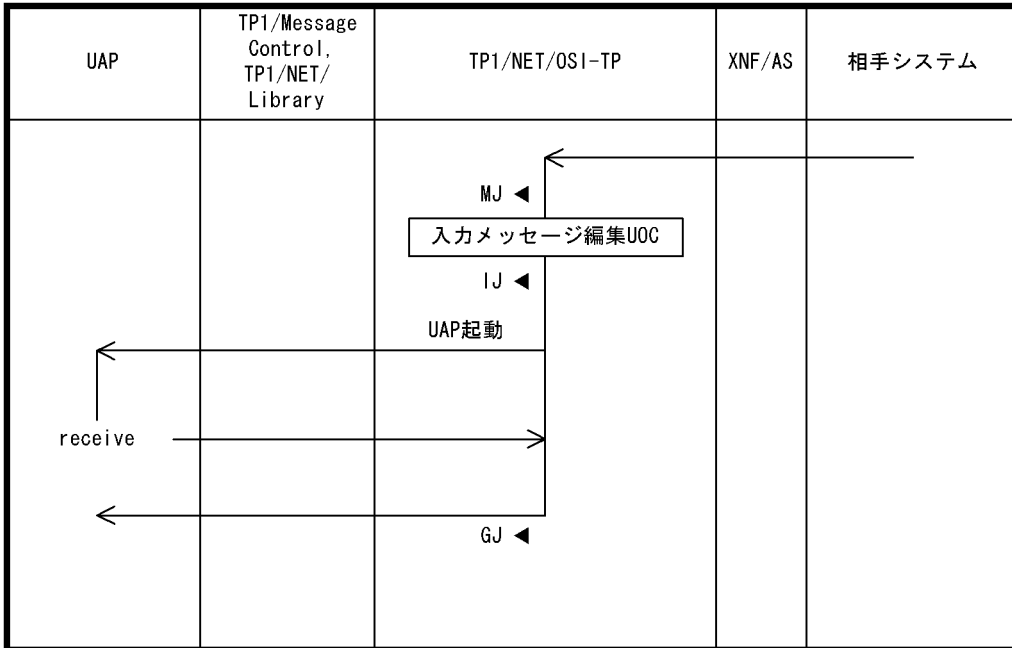
(パラメタの説明)

- Initiating-AP-Title : 起動側 AP 名称
- Initiating-API-Identifier : 起動側 API 識別子
- Initiating-AE-Qualifier : 起動側 AE 修飾子
- Initiating-AEI-Identifier : 起動側 AEI 識別子
- Initiating-TPSU-Title : 起動側 TPSU 名称
- Recipient-AP-Title : 受信側 AP 名称
- Recipient-API-Identifier : 受信側 API 識別子
- Recipient-AE-Qualifier : 受信側 AE 修飾子
- Recipient-AEI-Identifier : 受信側 AEI 識別子
- Recipient-TPSU-Title : 受信側 TPSU 名称
- Functional-Units : 機能単位
- Quality-of-Service : サービス品質
- Application-Context-name : アプリケーションコンテキスト名
- Confirmation : 確認
- Result : 結果
- Diagnostic : 診断
- Rollback : ロールバック
- User-Data : ユーザデータ
- Confirmation-Urgency : 確認緊急性

付録 B 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を示します。

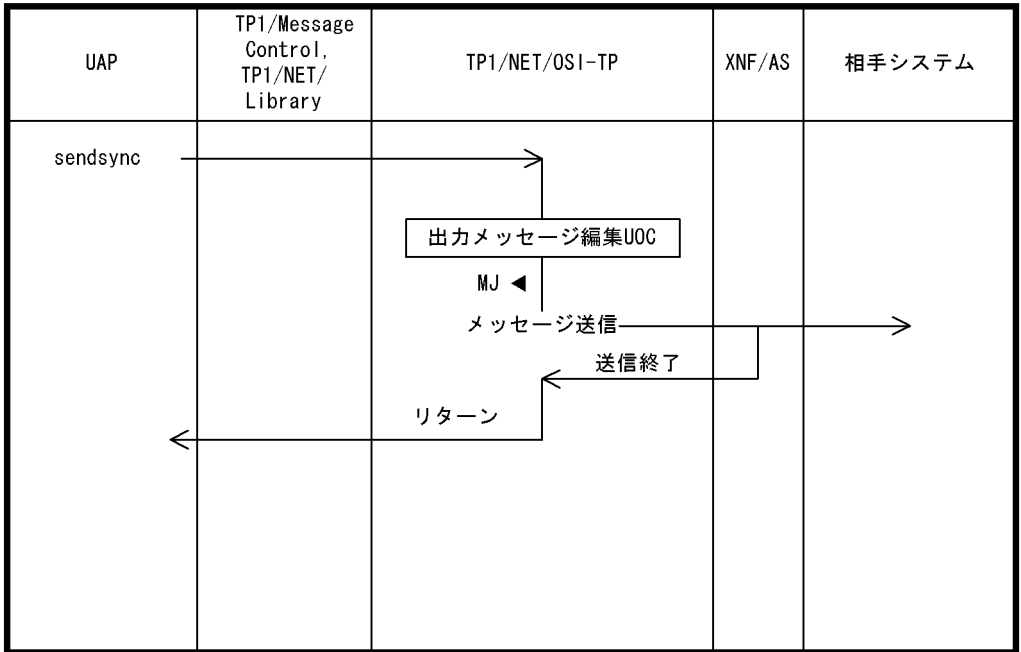
図 B-1 一方送信メッセージの受信



(凡例)

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

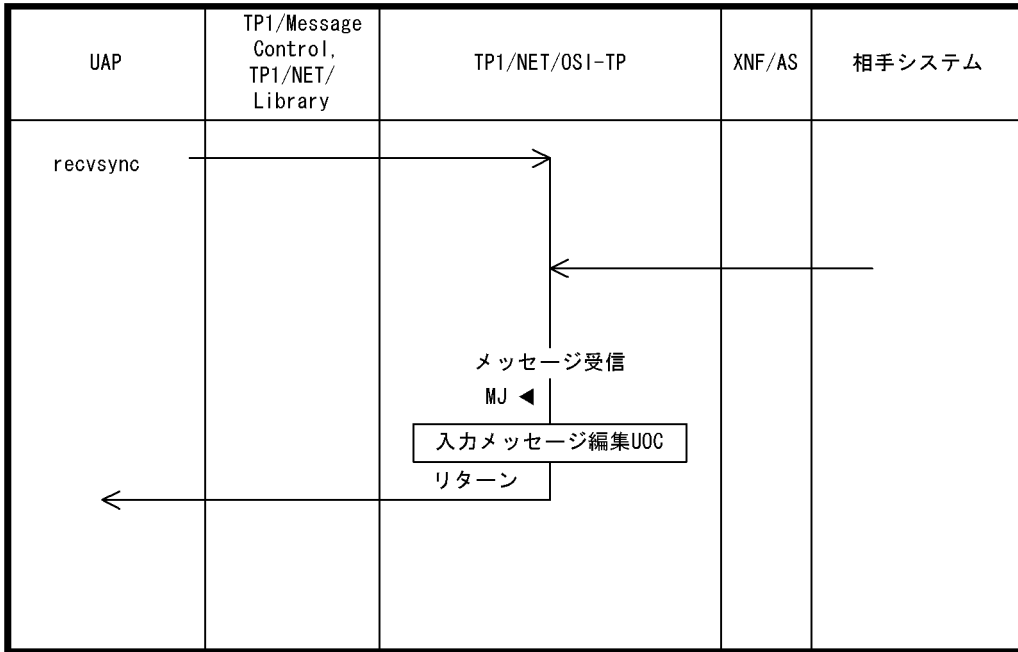
図 B-2 同期型のメッセージの送信



(凡例)

MJ ◀ : メッセージジャーナル取得

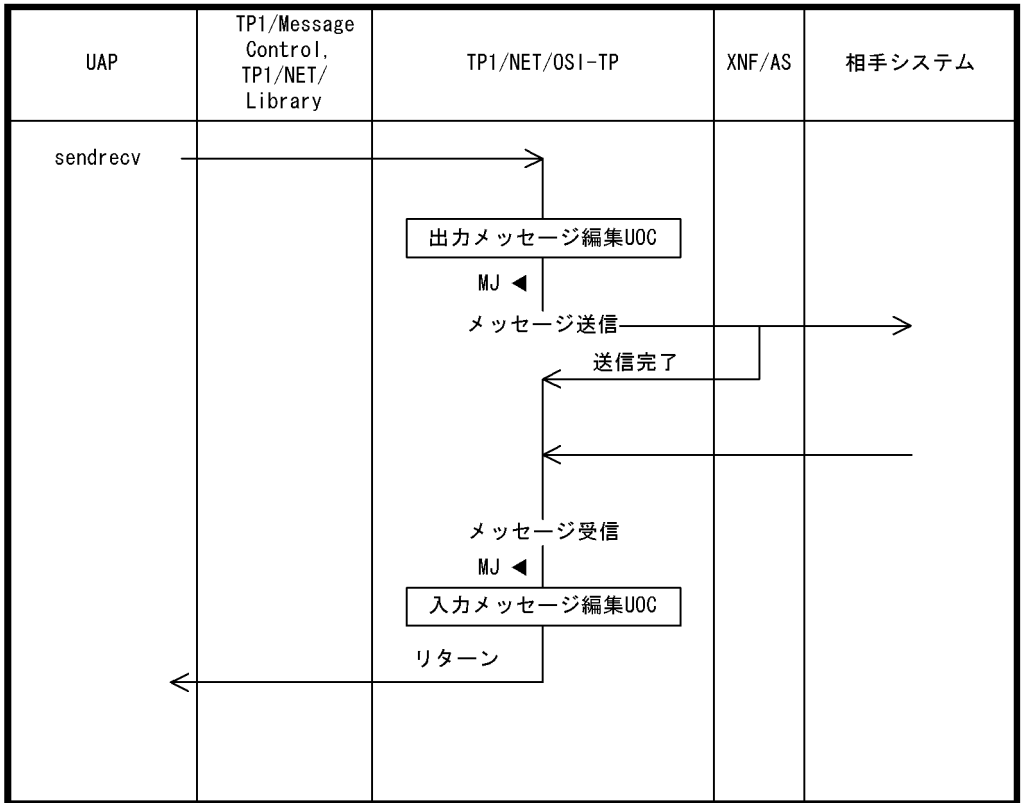
図 B-3 同期型のメッセージの受信



(凡例)

MJ ◀: メッセージジャーナル取得

図 B-4 同期型のメッセージの送受信



(凡例)

MJ ◀: メッセージジャーナル取得

付録 C 障害発生時の処理の流れ

障害発生時の処理の流れを示します。

図 C-1 通信回線障害

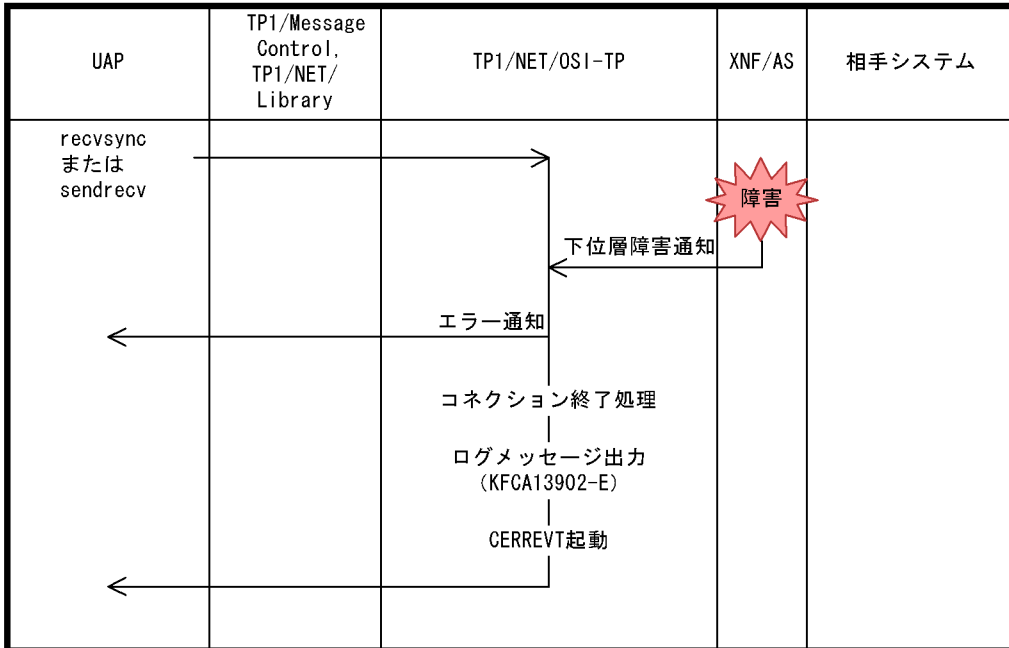
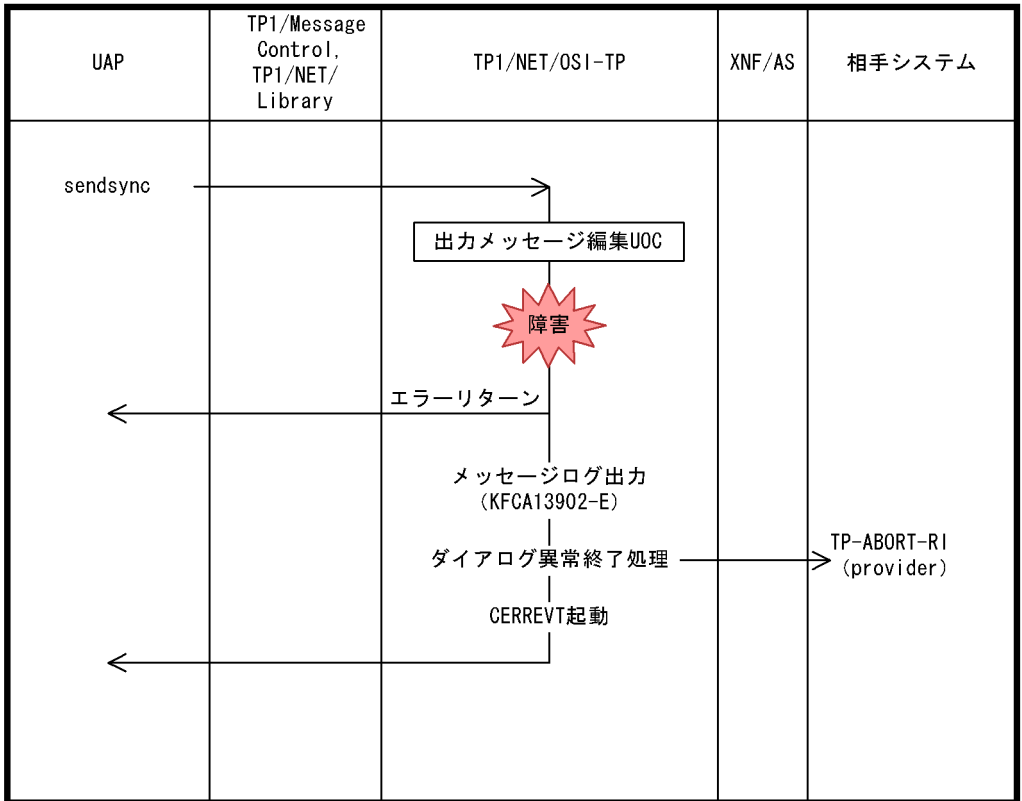


図 C-2 TP1/NET/OSI-TP での障害



付録 D 理由コード一覧

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

表 D-1 ERREVT2 の理由コード一覧

C 言語の理由コード (16 進数字)	COBOL 言語の理由コード (外部 10 進)	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	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 D-2 CERREVT の理由コード一覧

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件	障害レベル
DCMOTM_RSN1_MCF (MCF 障害) (00000001)	DCMOTM_RSN2_ITQ (00000000)	メッセージ入力障害	コネクション
	DCMOTM_RSN2_APL (00000001)	アプリケーション名 取得失敗	コネクション

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件	障害レベル
	DCMOTM_RSN2_OTGET (00000002)	メッセージ出力障害	コネクション
	DCMOTM_RSN2_OTCMP (00000003)	メッセージ送信処理 障害	コネクション
	DCMOTM_RSN2_UAPAB (00000005)	UAP 異常による強制 解放	コネクション
	DCMOTM_RSN2_SYCER (00000006)	UAP への同期リター ン失敗	コネクション
	DCMOTM_RSN2_ENDER (00000007)	終了処理中のメッ セージ拒否	コネクション
	DCMOTM_RSN2_DCTCN_F (20000000)	meftdctcn -f による強 制解放	コネクション
	DCMOTM_RSN2_ENDTO (20000001)	監視時間超過による 強制解放	コネクション
	DCMOTM_RSN2_NOBUF (20000002)	バッファ取得失敗に よる強制解放	コネクション
	DCMOTM_RSN2_ERROR (ffffff)	MCF 内部矛盾による 強制解放	コネクション
	その他	上記以外の障害 (理由コード 2 は保守 情報)	コネクション
DCMOTM_RSN1_UOC(U OC 検出障害) (00000003)	UOC からの詳細リターンコー ド	ユーザ (UOC) 検出 障害	コネクション
DCMOTM_RSN1_ACTER (コネクション障害) (00000005)	不定	コネクションの確立 失敗	コネクション
DCMOTM_RSN1_ABORT (プロトコル障害) (00000006)	DCMOTM_RSN2_XNF (00000010)	下位層障害	コネクション
	DCMOTM_RSN2_OTP (00000011)	TP1/NET/OSI-TP 検 出プロトコル障害	コネクション
その他	不定	上記以外の障害 (理由コード 1, 2 は 保守情報)	コネクション

付録 E ユーザアプリケーションプログラムのコーディング例

ここでは、ユーザアプリケーションプログラムのコーディング例を示します。

なお、TP1/NET/OSI-TP では、全二重の場合のコーディング例と半二重の場合のコーディング例を表 E-1 に示すファイルで提供しています。

表 E-1 TP1/NET/OSI-TP が提供するユーザアプリケーションプログラムのコーディング例

分類	プログラミング言語	提供するファイルの名称
全二重の場合	C 言語 (K&R 版)	<ul style="list-style-type: none"> • コンテンション勝者側 (SPP) /BeTRAN/examples/mcf/OSITP/aplib/c/sv1.c • コンテンション敗者側 (MHP) /BeTRAN/examples/mcf/OSITP/aplib/c/ap1.c
	COBOL 言語	<ul style="list-style-type: none"> • コンテンション勝者側 (SPP) /BeTRAN/examples/mcf/OSITP/aplib/cobol/sv1.cbl • コンテンション敗者側 (MHP) /BeTRAN/examples/mcf/OSITP/aplib/cobol/ap1.cbl
	データ操作言語	<ul style="list-style-type: none"> • コンテンション勝者側 (SPP) /BeTRAN/examples/mcf/OSITP/aplib/dml/sv1.cbl • コンテンション敗者側 (MHP) /BeTRAN/examples/mcf/OSITP/aplib/dml/ap1.cbl
半二重の場合	C 言語 (K&R 版)	<ul style="list-style-type: none"> • コンテンション勝者側 (SPP) /BeTRAN/examples/mcf/OSITP/aplib/c/sv2.c • コンテンション敗者側 (MHP) /BeTRAN/examples/mcf/OSITP/aplib/c/ap2.c
	COBOL 言語	<ul style="list-style-type: none"> • コンテンション勝者側 (SPP) /BeTRAN/examples/mcf/OSITP/aplib/cobol/sv2.cbl • コンテンション敗者側 (MHP) /BeTRAN/examples/mcf/OSITP/aplib/cobol/ap2.cbl
	データ操作言語	<ul style="list-style-type: none"> • コンテンション勝者側 (SPP) /BeTRAN/examples/mcf/OSITP/aplib/dml/sv2.cbl • コンテンション敗者側 (MHP) /BeTRAN/examples/mcf/OSITP/aplib/dml/ap2.cbl

付録 E.1 ユーザアプリケーションプログラムのコーディング例 (全二重での C 言語)

(1) コンテンション勝者側 (SPP)

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

#define      WAITTIME      60
```



```

void SEND01(in, in_len, out, out_len)

    char          *in      ;
    unsigned      DCLONG   in_len  ;
    char          *out     ;
    unsigned      DCLONG   out_len  ;
{
    /*  引数定義      */          /*引数を使用している関数*/
    /* sendsync, recvsync, sendrecv, receive */
    DCLONG  action      ;/* 0      0      0      0 */
    DCLONG  commform    ;/* 0      0      0      0 */
    char    termnam[16] ;/* 0      0      0      0 */
    char    resv01[16] ;/* 0      0      0      0 */
    char    senddata[256];/* 0      -      0      - */
    DCLONG  sdataleng   ;/* 0      -      0      - */
    char    recvdata[256];/* -      0      0      0 */
    DCLONG  rdataleng   ;/* -      0      0      0 */
    DCLONG  time        ;/* -      0      0      0 */
    DCLONG  wachtime    ;/* 0      0      0      - */
    char    resv02[16] ;/* 0      -      -      - */
    DCLONG  opcd        ;/* 0      -      -      - */
    DCLONG  inbufleng   ;/* -      0      0      0 */

    /*  変数定義 & 初期化  */
    int     rtn = 0x00  ;

    /*  引数初期化      */
    action      =  NULL;
    commform    =  NULL;
    memset(termnam, 0, 16) ;
    memset(resv01, 0, 16) ;
    memset(senddata, 0, 256);
    sdataleng   =  NULL;
    memset(recvdata, 0, 256);
    rdataleng   =  NULL;
    time        =  NULL;
    wachtime    =  NULL;
    memset(resv02, 0, 16) ;
    opcd        =  NULL;
    inbufleng   =  NULL;

    /*
    /*      ダイアログ開始要求送信 (TP-BEGIN-DIALOGUE)
    /*
    /*      sendsync引数設定 (初期化)
    action      =  DCMCFEMI ;
    commform    =  DCNOFLAGS ;
    strcpy(termnam, "otm01") ;
    resv01[0]   =  NULL ;
    /* senddataイニシャルイズは省略 */
    sdataleng   =  54 ;
    resv02[0]   =  NULL ;
    opcd        =  DCNOFLAGS ;
    wachtime    =  WAITTIME ;

    /*  senddata 設定  */

```

```

/*  先頭8バイト目までは、MCFが使用する  */
senddata[8] = 0x00 ; /* レンクス 01 */
senddata[9] = 0x0a ; /* サービスプリミティブの長さ */

strcpy(senddata + 10, "OTP1") ; /* 送信手順(OTP1) */
strcpy(senddata + 14, "BDR ") ;
/* サービスプリミティブコード */
/* TP_BEGIN_DIALOGUE req */

senddata[18] = 0x00 ; /* レンクス 02 */
senddata[19] = 0x2a ; /* パラメタフィールドのトータル */

senddata[20] = 0x00 ; /* レンクス 03 */
senddata[21] = 0x0d ; /* 起動側TPSUタイトルの長さ */

strcpy(senddata + 22, "PR") ; /* 型コード */
strcpy(senddata + 24, "SEND TPSU");
/* 起動側TPSUタイトル(パラメタ1) */

senddata[33] = 0x00 ; /* レンクス 04 */
senddata[34] = 0x02 ; /* 受信側AP(パラメタ 2省略) */

senddata[35] = 0x00 ; /* レンクス 05 */
senddata[36] = 0x0d ; /* 受信側TPSUタイトルの長さ */
strcpy(senddata + 37, "PR") ; /* 型コード */
strcpy(senddata + 39, "RECV TPSU");
/* 受信側TPSUタイトル(パラメタ3) */

senddata[48] = 0x00 ; /* レンクス 06 */
senddata[49] = 0x04 ; /* 予備の長さ */

senddata[50] = 0x48 ;
/*全二重&ハンドシェイク機能単位(パラメタ4) */
senddata[51] = 0xff ; /* 予備 */

senddata[52] = 0x00 ; /* レンクス 07 */
senddata[53] = 0x02 ; /* サービス品質(パラメタ5省略) */

senddata[54] = 0x00 ; /* レンクス 08 */
senddata[55] = 0x02 ;
/* アプリケーションコンテキスト(パラメタ 6) */

senddata[56] = 0x00 ; /* レンクス 09 */
senddata[57] = 0x04 ; /* 確認項目の長さ */

strcpy(senddata + 58, "AL") ; /* 確認項目(パラメタ 7) */
/* all(同期型) */

senddata[60] = 0x00 ; /* レンクス 10 */
senddata[61] = 0x02 ; /* ユ - ザ情報(省略) */

/*      メッセージ送信開始(sendsync)      */

rtn = dc_mcf_sendsync(action, commform, termnam, resv01,
                    senddata, sdataleng, resv02, opcd,
                    wachtime);

if ( rtn != DCMCFRTN_00000 )
{

```

```

        goto ERROR;
    }
    /*      メッセージ送信終了 (sendsync)      */

/*
/*      ダイアログ開始応答受信 (TP-BEGIN-DIALOGUE)      */
/*
/*
/*      recvsync引数設定 (初期化)      */
action          = DCMCFRST ;
commform        = DCNOFLAGS ;
resv01[0]       = NULL ;
/*recvdataイニシャライズは省略*/
rdataleng       = NULL ;
inbufleng       = 256 ;
time            = NULL ;
wachttime       = WAITTIME ;
/*      メッセージ受信開始 (recvsync)      */

rtn = dc_mcf_recvsync(action, commform, termnam, resv01,
                      recvdata, &rdataleng, inbufleng,
                      &time, wachttime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      メッセージ受信終了 (recvsync)      */

/*
/*      データ送信 (TP-DATA)
/*
/*
/*      sendsync引数設定 (初期化)      */
action          = DCMCFEMI ;
commform        = DCNOFLAGS ;
resv01[0]       = NULL ;
sdataleng       = 23 ;
resv02[0]       = NULL ;
opcd            = DCNOFLAGS ;
wachttime       = WAITTIME ;

/*      senddata clear      */
memset (senddata, 0, 256) ;

/*      senddata 設定      */
/*      先頭8バイト目までは、MCFが使用する      */
senddata[8] = 0x00 ; /* レンクス 01 */
senddata[9] = 0x0a ; /* サービスプリミティブの長さ */

strcpy (senddata + 10, "OTP1"); /* 送信手順 (OTP1) */
strcpy (senddata + 14, "TDR ");
/*      サービスプリミティブコード */
/*      TP_DT req */

senddata[18] = 0x00 ; /* レンクス 02 */
senddata[19] = 0x02 ; /* パラメタ (省略) */

```

```

senddata[20] = 0x00    ; /* レンゲス 03 */
senddata[21] = 0x0b    ; /* ユーザ情報の長さ */

senddata[22] = 0x80    ; /* 連結状態 */

strcpy(senddata + 23, "USERDATA"); /* ユーザデータ */

/*      データ送信(sendsync)開始      */

rtn = dc_mcf_sendsync(action, commform, termnam, resv01,
                      senddata, sdataleng, resv02, opcd,
                      wachtime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      データ送信(sendsync)終了      */

/*      ハンドシェイク受信(TP-HANDSHAKE)      */
/*      */
/*      */
/*      recvsync引数設定 (初期化)      */
action          = DCMCFRST    ;
commform        = DCNOFLAGS   ;
resv01[0]       = NULL        ;
rdataleng       = NULL        ;
inbufleng       = 256         ;
time            = NULL        ;
wachtime        = WAITTIME    ;

/*      recvdata clear      */
memset(recvdata, 0, 256)      ;

/*      メッセージ受信開始(recvsync)      */

rtn = dc_mcf_recvsync(action, commform, termnam, resv01,
                      recvdata, &rdataleng, inbufleng,
                      &time, wachtime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      メッセージ受信終了(recvsync)      */

/*      ハンドシェイク送信(TP-HANDSHAKE)      */
/*      */
/*      */
/*      sendsync引数設定 (初期化)      */
action          = DCMCFEMI    ;
commform        = DCNOFLAGS   ;
resv01[0]       = NULL        ;
sdataleng       = 14         ;
resv02[0]       = NULL        ;
opcd            = DCNOFLAGS   ;

```

```

wachttime          =   WAITTIME      ;

/*  senddata clear  */
memset(senddata, 0, 256)          ;

/*  senddata 設定  */
/*  先頭8バイト目までは、MCFが使用する  */
senddata[8] = 0x00      ; /* レンクス 01 */
senddata[9] = 0x0a     ; /* サービスプリミティブの長さ */

strcpy(senddata + 10, "OTP1"); /* 送信手順(OTP1) */
strcpy(senddata + 14, "HSP "); /* サービスプリミティブコード */

senddata[18] = 0x00     ; /* レンクス 02 */
senddata[19] = 0x02     ; /* パラメタ(省略) */

senddata[20] = 0x00     ; /* レンクス 3 */
senddata[21] = 0x02     ; /* ユーザ情報(省略) */

/*  データ送信開始(sendsync)  */

rtn = dc_mcf_sendsync(action, commform, termnam, resv01,
                      senddata, sdataleng, resv02, opcd,
                      wachttime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*  データ送信終了(sendsync)  */

/*  ダイアログ終了要求送受信(TP-END-DIALOGUE)  */
/*  */

/*  sendrecv引数設定  */
action          =   DCMCFEMI      ;
commform        =   DCMCFIO      ;
resv01[0]       =   NULL         ;
sdataleng       =   18          ;
rdataleng       =   NULL         ;
inbufleng       =   256         ;
time            =   NULL         ;
wachttime       =   WAITTIME     ;

/*  senddata and recvdata clear  */
memset(senddata, 0, 256)          ;
memset(recvdata, 0, 256)         ;

/*  senddata 設定  */
/*  先頭8バイト目までは、MCFが使用する  */
senddata[8] = 0x00      ; /* レンクス 01 */
senddata[9] = 0x0a     ; /* サービスプリミティブの長さ */

strcpy(senddata + 10, "OTP1"); /* 送信手順(OTP1) */
strcpy(senddata + 14, "EDR "); /* サービスプリミティブコード */

```

```

                                /* TP_END_DIALOGUE req */

senddata[18] = 0x00    ; /* レングス 02 */
senddata[19] = 0x06    ; /* パラメタ全体の長さ */

senddata[20] = 0x00    ; /* レングス 03 */
senddata[21] = 0x04    ; /* 確認項目の長さ */

strcpy(senddata + 22, "TR")    ; /* 確認項目(パラメタ1) */

senddata[24] = 0x00    ; /* レングス 04 */
senddata[25] = 0x02    ; /* ユーザ情報(省略) */

/*      メッセージ送受信開始(sendrecv)      */

rtn = dc_mcf_sendrecv(action, commform, termnam, resv01,
                      senddata, sdataleng, recvdata,
                      &rdataleng, inbufleng, &time,
                      wachtime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      メッセージ送受信終了(sendrecv)      */

ERROR : ;
}

```

(2) コンテンション敗者側 (MHP)

```

#include <dcmcf.h>
#include <string.h>

#define      WAITTIME      60

void      RECV01(in, in_len, out, out_len)

        char          *in          ;
unsigned      DCLONG      in_len    ;
        char          *out         ;
unsigned      DCLONG      out_len   ;
{
    /*      引数定義      */
    /* 引数を使用している関数*/
    /* sendsync, recvsync, sendrecv, receive */
    DCLONG      action      ;/* 0      0      0      0 */
    DCLONG      commform    ;/* 0      0      0      0 */
    char        termnam[16] ;/* 0      0      0      0 */
    char        resv01[16]  ;/* 0      0      0      0 */
    char        senddata[256] ;/* 0      -      0      - */
    DCLONG      sdataleng   ;/* 0      -      0      - */
    char        recvdata[256] ;/* -      0      0      0 */
    DCLONG      rdataleng   ;/* -      0      0      0 */
    DCLONG      time        ;/* -      0      0      0 */
    DCLONG      wachtime    ;/* 0      0      0      - */
    char        resv02[16]  ;/* 0      -      -      - */
    DCLONG      opcd        ;/* 0      -      -      - */
}

```

```

DCLONG inbufleng      ;/* -          0          0          0 */

/* 変数定義 & 初期化 */
int      rtn = NULL ;

/* 引数初期化 */
action   = NULL;
commform = NULL;
memset( termnam, 0, 16) ;
memset( resv01, 0, 16) ;
memset( senddata, 0, 256);
sdataleng = NULL;
memset( recvdata, 0, 256);
rdataleng = NULL;
time      = NULL;
wachtime  = NULL;
memset( resv02, 0, 16) ;
opcd     = NULL;
inbufleng = NULL;

/*
/*      ダイアログ開始要求受信 (TP-BEGIN-DIALOGUE)      */
/*
/*
/*      receive引数設定 (初期化)      */
action   = DCMCFRST;
commform = DCNOFLAGS;
termnam[0] = NULL ;
/*recvdataイニシャライズは省略*/
resv01[0] = NULL ;
rdataleng = NULL ;
inbufleng = 256 ;
time      = NULL ;

/*      ダイアログ開始要求受信開始 (receive)      */

rtn = dc_mcf_receive(action, commform, termnam, resv01,
                    recvdata, &rdataleng, inbufleng,
                    &time);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      ダイアログ開始要求受信終了 (receive)      */

/*
/*      ダイアログ開始応答送信 (TP-BEGIN-DIALOGUE)      */
/*
/*
/*      sendsync引数設定 (初期化)      */
action   = DCMCFEMI ;
commform = DCNOFLAGS ;
resv01[0] = NULL ;
sdataleng = 18 ;
resv02[0] = NULL ;
opcd     = DCNOFLAGS ;
wachtime = WAITTIME ;

```



```

}
/*      メッセージ受信終了 (recvsync)      */

/*      *      */
/*      ハンドシェイク (TP-HANDSHAKE)      */
/*      *      */

/*      sendrecv引数設定 (初期化)      */
action          =   DCMCFEMI      ;
commform        =   DCMCFIO      ;
resv01[0]       =   NULL         ;
sdataleng       =   18           ;
rdataleng       =   NULL         ;
inbufleng       =   256         ;
time            =   NULL         ;
wachttime       =   WAITTIME    ;

/*      senddata clear      */
memset(senddata, 0, 256);

/*      recvdata clear      */
memset(recvdata, 0, 256);

/*      senddata 設定      */
/*      先頭8バイト目までは、MCFが使用する      */
senddata[8] = 0x00 ; /* レングス 01 */
senddata[9] = 0x0a ; /* サービスプリミティブの長さ */

strcpy(senddata + 10, "OTP1") ; /* 送信手順(OTP1) */
strcpy(senddata + 14, "HSR ") ; /* サービスプリミティブ */
/* TP_HANDSHAKE */

senddata[18] = 0x00 ; /* レングス 02 */
senddata[19] = 0x06 ; /* パラメタ全体の長さ */

senddata[20] = 0x00 ; /* レングス 03 */
senddata[21] = 0x04 ; /* 緊急確認項目の長さ */

strcpy(senddata + 22, "NR") ;
/* 緊急確認項目 (パラメタ 1) */

senddata[24] = 0x00 ; /* レングス 04 */
senddata[25] = 0x02 ; /* ユーザ情報(省略) */

/*      メッセージ送受信開始 (sendrecv)      */

rtn = dc_mcf_sendrecv(action, commform, termnam, resv01,
                      senddata, sdataleng, recvdata,
                      &rdataleng, inbufleng, &time,
                      wachttime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      メッセージ送受信終了 (sendrecv)      */

/*      *      */

```

```

/*      ダイアログ終了要求受信 (TP-END-DIALOGUE)          */
/*                                                              */

/*      recvsync引数設定 (初期化)          */
action      =   DCMCFRST      ;
commform    =   DCNOFLAGS    ;
resv01[0]   =   NULL         ;
rdataleng   =   NULL         ;
inbufleng   =   256         ;
time        =   NULL         ;
wachttime   =   WAITTIME    ;

/*      recvdata clear          */
memset(recvdata, 0, 256)      ;

/*      メッセージ受信開始 (recvsync)          */

rtn = dc_mcf_recvsync(action, commform, termnam, resv01,
                      recvdata, &rdataleng, inbufleng,
                      &time, wachttime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      メッセージ受信終了 (recvsync)          */

/*                                                              */
/*      ダイアログ終了応答送信 (TP-END-DIALOGUE)          */
/*                                                              */

/*      sendsync引数設定 (初期化)          */
action      =   DCMCFEMI     ;
commform    =   DCNOFLAGS    ;
resv01[0]   =   NULL         ;
sdataleng   =   14          ;
resv02[0]   =   NULL         ;
opcd        =   DCNOFLAGS    ;
wachttime   =   WAITTIME    ;

/*      senddata clear          */
memset(senddata, 0, 256)      ;

/*      senddata 設定          */
/*      先頭8バイト目までは、MCFが使用する          */
senddata[8] = 0x00 ;          /* レングス 01 */
senddata[9] = 0x0a ;          /* サービスプリミティブの長さ */

strcpy(senddata + 10, "OTP1") ; /* 送信手順(OTP1) */
strcpy(senddata + 14, "EDP ") ; /* サービスプリミティブコード */

senddata[18] = 0x00 ;          /* レングス 02 */
senddata[19] = 0x02 ;          /* パラメタ(省略) */
senddata[20] = 0x00 ;          /* レングス 03 */
senddata[21] = 0x02 ;          /* ユーザ情報(省略) */

/*      メッセージ送信開始 (sendsync)          */

```

```

rtn = dc_mcf_sendsync(action, commform, termnam, resv01,
                      senddata, sdataleng, resv02, opcd,
                      wachtime);

/*      メッセージ送信終了(sendsync)      */

ERROR : ;
}

```

付録 E.2 ユーザアプリケーションプログラムのコーディング例 (全二重での COBOL 言語)

(1) コンテンション勝者側 (SPP)

```

IDENTIFICATION  DIVISION.
PROGRAM-ID.     SEND01.

ENVIRONMENT     DIVISION.

DATA            DIVISION.
WORKING-STORAGE SECTION.

01  SND-S1.
    02  MSG-SNDSYNC  PIC X(8)  VALUE  'SENDSYNC'.
    02  DATA1      PIC X(5) .
    02  FILLER      PIC X(3) .
    02  DATA2      PIC X(4)  VALUE  SPACE.
    02  DATA3      PIC X(4)  VALUE  SPACE.
    02  DATA4      PIC 9(8) .
    02  DATA5      PIC 9(8) .
    02  DATA6      PIC 9(9)  COMP  VALUE  ZERO.
    02  SND-S-SEG   PIC X(4)  VALUE  'EMI '.
    02  DATA7      PIC X(4)  VALUE  SPACE.
    02  DATA8      PIC X(4)  VALUE  SPACE.
    02  DATA9      PIC X(4)  VALUE  SPACE.
    02  DATA10     PIC X(8)  VALUE  SPACE.
    02  DATA11     PIC X(4)  VALUE  SPACE.
    02  DATA12     PIC X(8)  VALUE  SPACE.
    02  DATA13     PIC X(4)  VALUE  SPACE.
    02  DATA14     PIC 9(9)  COMP  VALUE  ZERO.
    02  DATA15     PIC 9(9)  COMP  VALUE  ZERO.
    02  DATA16     PIC X(1)  VALUE  SPACE.
    02  DATA17     PIC X(1)  VALUE  SPACE.
    02  DATA18     PIC X(14) VALUE  LOW-VALUE.

01  SND-S2.
    02  SEG-CODE    PIC X(4)  VALUE  SPACE.
    02  TRM-NAME    PIC X(8)  VALUE  'otm01 '.
    02  DATA19     PIC X(8)  VALUE  SPACE.
    02  DATA20     PIC X(8)  VALUE  SPACE.
    02  DATA21     PIC X(28) VALUE  LOW-VALUE.

01  SND-S3.
    02  DATA22     PIC 9(9)  COMP  VALUE  54.
    02  DATA23     PIC X(8) .
    02  SEND-DATA.

```

付録 E ユーザアプリケーションプログラムのコーディング例

```

03      S-DATA1      PIC X(1)  VALUE  X'00' .
03      S-DATA2      PIC X(1)  VALUE  X'0a' .
03      S-DATA3      PIC X(4)  VALUE  'OTP1' .
03      S-DATA4      PIC X(4)  VALUE  'BDR' .
03      S-DATA5      PIC X(1)  VALUE  X'00' .
03      S-DATA6      PIC X(1)  VALUE  X'2a' .
03      S-DATA7      PIC X(1)  VALUE  X'00' .
03      S-DATA8      PIC X(1)  VALUE  X'0d' .
03      S-DATA8-1    PIC X(2)  VALUE  'PR' .
03      S-DATA9      PIC X(9)  VALUE  'SEND_TPSU' .
03      S-DATA10     PIC X(1)  VALUE  X'00' .
03      S-DATA11     PIC X(1)  VALUE  X'02' .
03      S-DATA12     PIC X(1)  VALUE  X'00' .
03      S-DATA13     PIC X(1)  VALUE  X'0d' .
03      S-DATA13-1   PIC X(2)  VALUE  'PR' .
03      S-DATA14     PIC X(9)  VALUE  'RECV_TPSU' .
03      S-DATA15     PIC X(1)  VALUE  X'00' .
03      S-DATA16     PIC X(1)  VALUE  X'04' .
03      S-DATA17     PIC X(1)  VALUE  X'48' .
03      S-DATA18     PIC X(1)  VALUE  X'ff' .
03      S-DATA19     PIC X(1)  VALUE  X'00' .
03      S-DATA20     PIC X(1)  VALUE  X'02' .
03      S-DATA21     PIC X(1)  VALUE  X'00' .
03      S-DATA22     PIC X(1)  VALUE  X'02' .
03      S-DATA23     PIC X(1)  VALUE  X'00' .
03      S-DATA24     PIC X(1)  VALUE  X'04' .
03      S-DATA25     PIC X(2)  VALUE  'AL' .
03      S-DATA26     PIC X(1)  VALUE  X'00' .
03      S-DATA27     PIC X(1)  VALUE  X'02' .
01      SND-S4 .
02      MSG-SNDSYNC  PIC X(8)  VALUE  'SENDSYNC' .
02      DATA25      PIC X(5) .
02      FILLER       PIC X(3) .
02      DATA26      PIC X(4)  VALUE  SPACE .
02      DATA27      PIC X(4)  VALUE  SPACE .
02      DATA28      PIC 9(8) .
02      DATA29      PIC 9(8) .
02      DATA30      PIC 9(9)  COMP  VALUE  ZERO .
02      SND-S-SEG    PIC X(4)  VALUE  'EMI' .
02      DATA31      PIC X(4)  VALUE  SPACE .
02      DATA32      PIC X(4)  VALUE  SPACE .
02      DATA33      PIC X(4)  VALUE  SPACE .
02      DATA34      PIC X(8)  VALUE  SPACE .
02      DATA35      PIC X(4)  VALUE  SPACE .
02      DATA36      PIC X(8)  VALUE  SPACE .
02      DATA37      PIC X(4)  VALUE  SPACE .
02      DATA38      PIC 9(9)  COMP  VALUE  ZERO .
02      DATA39      PIC 9(9)  COMP  VALUE  ZERO .
02      DATA40      PIC X(1)  VALUE  SPACE .
02      DATA41      PIC X(1)  VALUE  SPACE .
02      DATA42      PIC X(14) VALUE  LOW-VALUE .
01      SND-S5 .
02      SEG-CODE     PIC X(4)  VALUE  SPACE .
02      TRM-NAME     PIC X(8)  VALUE  'otm01' .
02      DATA43      PIC X(8)  VALUE  SPACE .
02      DATA44      PIC X(8)  VALUE  SPACE .
02      DATA45      PIC X(28) VALUE  LOW-VALUE .

```

```

01  SND-S6.
02  DATA46          PIC 9(9)      COMP VALUE 23.
02  DATA48          PIC X(8) .
02  SEND-DATA2.
03  S-DATA281       PIC X(1)      VALUE  X'00' .
03  S-DATA282       PIC X(1)      VALUE  X'0a' .
03  S-DATA283       PIC X(4)      VALUE  'OTP1' .
03  S-DATA284       PIC X(4)      VALUE  'TDR ' .
03  S-DATA285       PIC X(1)      VALUE  X'00' .
03  S-DATA286       PIC X(1)      VALUE  X'02' .
03  S-DATA287       PIC X(1)      VALUE  X'00' .
03  S-DATA288       PIC X(1)      VALUE  X'0b' .
03  S-DATA289       PIC X(1)      VALUE  X'80' .
03  S-DATA290       PIC X(8)      VALUE  'USERDATA' .

01  SND-S7.
02  MSG-SNDSYNC     PIC X(8)      VALUE  'SENDSYNC' .
02  DATA49         PIC X(5) .
02  FILLER          PIC X(3) .
02  DATA50         PIC X(4)      VALUE  SPACE.
02  DATA51         PIC X(4)      VALUE  SPACE.
02  DATA52         PIC 9(8) .
02  DATA53         PIC 9(8) .
02  DATA54         PIC 9(9)      COMP  VALUE  ZERO.
02  SND-S-SEG       PIC X(4)      VALUE  'EMI ' .
02  DATA55         PIC X(4)      VALUE  SPACE.
02  DATA56         PIC X(4)      VALUE  SPACE.
02  DATA57         PIC X(4)      VALUE  SPACE.
02  DATA58         PIC X(8)      VALUE  SPACE.
02  DATA59         PIC X(4)      VALUE  SPACE.
02  DATA60         PIC X(8)      VALUE  SPACE.
02  DATA61         PIC X(4)      VALUE  SPACE.
02  DATA62         PIC 9(9)      COMP  VALUE  ZERO.
02  DATA63         PIC 9(9)      COMP  VALUE  ZERO.
02  DATA64         PIC X(1)      VALUE  SPACE.
02  DATA65         PIC X(1)      VALUE  SPACE.
02  DATA66         PIC X(14)     VALUE  LOW-VALUE.

01  SND-S8.
02  SEG-CODE        PIC X(4)      VALUE  SPACE.
02  TRM-NAME        PIC X(8)      VALUE  'otm01 ' .
02  DATA67         PIC X(8)      VALUE  SPACE.
02  DATA68         PIC X(8)      VALUE  SPACE.
02  DATA69         PIC X(28)     VALUE  LOW-VALUE.

01  SND-S9.
02  DATA70         PIC 9(9)      COMP  VALUE  14.
02  DATA71         PIC X(8) .
02  SEND-DATA3.
03  S-DATA29        PIC X(1)      VALUE  X'00' .
03  S-DATA30        PIC X(1)      VALUE  X'0a' .
03  S-DATA31        PIC X(4)      VALUE  'OTP1' .
03  S-DATA32        PIC X(4)      VALUE  'HSP ' .
03  S-DATA33        PIC X(1)      VALUE  X'00' .
03  S-DATA34        PIC X(1)      VALUE  X'02' .
03  S-DATA35        PIC X(1)      VALUE  X'00' .
03  S-DATA36        PIC X(1)      VALUE  X'02' .

01  RCV-S1.
02  MSG-RCVSYNC    PIC X(8)      VALUE  'RCVSYNC' .

```

付録 E ユーザアプリケーションプログラムのコーディング例

```

02     DATA73     PIC X(5) .
02     FILLER      PIC X(3) .
02     RCV-S-SEG   PIC X(4)   VALUE 'FRST' .
02     DATA74     PIC X(4)   VALUE  SPACE .
02     DATA75     PIC 9(8) .
02     DATA76     PIC 9(8) .
02     RCV-S-LEG   PIC 9(9)   COMP   VALUE   256 .
02     DATA77     PIC X(4)   VALUE  SPACE .
02     DATA78     PIC X(4)   VALUE  SPACE .
02     DATA79     PIC X(4)   VALUE  SPACE .
02     DATA80     PIC X(4)   VALUE  SPACE .
02     DATA81     PIC X(8)   VALUE  SPACE .
02     DATA82     PIC X(4)   VALUE  SPACE .
02     DATA83     PIC X(8)   VALUE  SPACE .
02     DATA84     PIC X(4)   VALUE  SPACE .
02     DATA85     PIC 9(9)   COMP   VALUE   ZERO .
02     DATA86     PIC 9(9)   COMP   VALUE   ZERO .
02     DATA87     PIC X(1)   VALUE  SPACE .
02     DATA88     PIC X(1)   VALUE  SPACE .
02     DATA89     PIC X(14)  VALUE  LOW-VALUE .
01    RCV-S2 .
02     DATA90     PIC X(4)   VALUE  SPACE .
02     TRM-NAME    PIC X(8)   VALUE  'otm01' .
02     DATA92     PIC X(8)   VALUE  SPACE .
02     DATA93     PIC X(8)   VALUE  SPACE .
02     DATA94     PIC X(28)  VALUE  LOW-VALUE .
01    RCV-S3 .
02     DATA95     PIC 9(9)   COMP .
02     DATA96     PIC X(8) .
02     DATA97     PIC X(1024) .
01    SND-RCV1 .
02     MSG-SNDRCV  PIC X(8)   VALUE  'SENDRECV' .
02     DATA101    PIC X(5) .
02     FILLER      PIC X(3) .
02     DATA102    PIC X(4)   VALUE  SPACE .
02     DATA103    PIC X(4)   VALUE  SPACE .
02     DATA104    PIC 9(8) .
02     DATA105    PIC 9(8) .
02     SND-RCV-LEG PIC 9(9)   COMP   VALUE   256 .
02     SND-RCV-SEG PIC X(4)   VALUE  'EMI' .
02     DATA106    PIC X(4)   VALUE  SPACE .
02     DATA107    PIC X(4)   VALUE  SPACE .
02     DATA108    PIC X(4)   VALUE  SPACE .
02     DATA109    PIC X(8)   VALUE  SPACE .
02     DATA110    PIC X(4)   VALUE  SPACE .
02     DATA111    PIC X(8)   VALUE  SPACE .
02     DATA112    PIC X(4)   VALUE  SPACE .
02     PRO-CD      PIC 9(9)   COMP   VALUE   ZERO .
02     DATA113    PIC 9(9)   COMP   VALUE   ZERO .
02     DATA113-1  PIC X(1)   VALUE  SPACE .
02     DATA113-2  PIC X(1)   VALUE  SPACE .
02     DATA114    PIC X(14)  VALUE  LOW-VALUE .
01    SND-RCV2 .
02     MSG         PIC X(4)   VALUE  'IO' .
02     TRM-NAME    PIC X(8)   VALUE  'otm01' .
02     DATA115    PIC X(8)   VALUE  SPACE .
02     DATA116    PIC X(8)   VALUE  SPACE .

```

```

01      02      DATA117      PIC X(28)  VALUE  LOW-VALUE.
      SND-RCV3.
      02      MSG-SEG-LEG  PIC 9(9)    COMP    VALUE  18.
      02      DATA119      PIC X(8) .
      02      SNDRCV-DATA.
          03      SR-DATA1      PIC X(1)  VALUE  X'00' .
          03      SR-DATA2      PIC X(1)  VALUE  X'0a' .
          03      SR-DATA3      PIC X(4)  VALUE  'OTP1' .
          03      SR-DATA4      PIC X(4)  VALUE  'EDR ' .
          03      SR-DATA5      PIC X(1)  VALUE  X'00' .
          03      SR-DATA6      PIC X(1)  VALUE  X'06' .
          03      SR-DATA7      PIC X(1)  VALUE  X'00' .
          03      SR-DATA8      PIC X(1)  VALUE  X'04' .
          03      SR-DATA9      PIC X(2)  VALUE  'TR' .
          03      SR-DATA10     PIC X(1)  VALUE  X'00' .
          03      SR-DATA11     PIC X(1)  VALUE  X'02' .
01      SND-RCV4.
      02      DATA120      PIC 9(9)    COMP.
      02      DATA121-1     PIC X(7)    VALUE  SPACE.
      02      DATA121-2     PIC X(1) .
      02      DATA123      PIC X(248) .

PROCEDURE      DIVISION.
CALL  'CBLDCMCF'  USING  SND-S1  SND-S2  SND-S3.

CALL  'CBLDCMCF'  USING  RCV-S1  RCV-S2  RCV-S3.

CALL  'CBLDCMCF'  USING  SND-S4  SND-S5  SND-S6.

CALL  'CBLDCMCF'  USING  RCV-S1  RCV-S2  RCV-S3.

CALL  'CBLDCMCF'  USING  SND-S7  SND-S8  SND-S9.

CALL  'CBLDCMCF'  USING  SND-RCV1  SND-RCV2  SND-RCV3  SND-RCV4.

EXIT  PROGRAM.

```

(2) コンテンション敗者側 (MHP)

```

IDENTIFICATION  DIVISION.
PROGRAM-ID.     RECV01.

ENVIRONMENT     DIVISION.

DATA            DIVISION.
WORKING-STORAGE SECTION.

01      RCV1.
      02      MSG-RCV      PIC X(8)  VALUE  'RECEIVE ' .
      02      DATA1      PIC X(5) .
      02      FILLER      PIC X(3) .
      02      SEG-CODE     PIC X(4)  VALUE  'FRST' .
      02      DATA2      PIC X(4)  VALUE  SPACE.
      02      DATA3      PIC 9(8) .
      02      DATA4      PIC 9(8) .
      02      SEG-LENG     PIC 9(9)  COMP    VALUE  256.
      02      DATA5      PIC X(4)  VALUE  SPACE.

```

付録E ユーザアプリケーションプログラムのコーディング例

```

02     DATA6           PIC X(4)    VALUE  SPACE.
02     DATA7           PIC X(4)    VALUE  SPACE.
02     DATA8           PIC X(4)    VALUE  SPACE.
02     DATA9           PIC X(8)    VALUE  SPACE.
02     DATA10          PIC X(4)    VALUE  SPACE.
02     DATA11          PIC X(8)    VALUE  SPACE.
02     DATA12          PIC X(4)    VALUE  SPACE.
02     DATA13          PIC 9(9)    COMP  VALUE  ZERO.
02     DATA13-1        PIC 9(9)    COMP  VALUE  ZERO.
02     DATA13-2        PIC X(1)    VALUE  SPACE.
02     DATA13-3        PIC X(1)    VALUE  SPACE.
02     DATA14          PIC X(14)   VALUE  LOW-VALUE.
01     RCV2.
02     DATA15          PIC X(4)    VALUE  SPACE.
02     RCV-TRM-NAM      PIC X(8) .
02     DATA17          PIC X(8)    VALUE  SPACE.
02     DATA18          PIC X(8)    VALUE  SPACE.
02     DATA20          PIC X(28)   VALUE  LOW-VALUE.
01     RCV3.
02     DATA21          PIC 9(9)    COMP.
02     DATA22          PIC X(8) .
02     DATA23          PIC X(1024) .
01     SND1.
02     MSG-SNDSYNC      PIC X(8)    VALUE  'SENDSYNC' .
02     DATA24          PIC X(5) .
02     FILLER           PIC X(3) .
02     DATA25          PIC X(4)    VALUE  SPACE.
02     DATA26          PIC X(4)    VALUE  SPACE.
02     DATA27          PIC 9(8) .
02     DATA28          PIC 9(8) .
02     DATA29          PIC 9(9)    COMP  VALUE  ZERO.
02     SND-S-SEG        PIC X(4)    VALUE  'EMI' .
02     DATA30          PIC X(4)    VALUE  SPACE.
02     DATA31          PIC X(4)    VALUE  SPACE.
02     DATA32          PIC X(4)    VALUE  SPACE.
02     DATA33          PIC X(8)    VALUE  SPACE.
02     DATA34          PIC X(4)    VALUE  SPACE.
02     DATA35          PIC X(8)    VALUE  SPACE.
02     DATA36          PIC X(4)    VALUE  SPACE.
02     DATA37          PIC 9(9)    COMP  VALUE  ZERO.
02     DATA38          PIC 9(9)    COMP  VALUE  ZERO.
02     DATA39          PIC X(1)    VALUE  SPACE.
02     DATA40          PIC X(1)    VALUE  SPACE.
02     DATA41          PIC X(14)   VALUE  LOW-VALUE.
01     SND2.
02     SEG-CODE         PIC X(4)    VALUE  SPACE.
02     S2-TRM-NAM       PIC X(8) .
02     DATA42          PIC X(8)    VALUE  SPACE.
02     DATA43          PIC X(8)    VALUE  SPACE.
02     DATA44          PIC X(28)   VALUE  LOW-VALUE.
01     SND3.
02     DATA45          PIC 9(9)    COMP  VALUE  18.
02     DATA46          PIC X(8) .
02     SEND-DATA.
03     S-DATA1         PIC X(1)    VALUE  X'00' .
03     S-DATA2         PIC X(1)    VALUE  X'0a' .
03     S-DATA3         PIC X(4)    VALUE  'OTP1' .

```



```

03 S-DATA4 PIC X(4) VALUE 'BDPA' .
03 S-DATA5 PIC X(1) VALUE X'00' .
03 S-DATA6 PIC X(1) VALUE X'06' .
03 S-DATA7 PIC X(1) VALUE X'00' .
03 S-DATA8 PIC X(1) VALUE X'04' .
03 S-DATA9 PIC X(2) VALUE 'AC' .
03 S-DATA10 PIC X(1) VALUE X'00' .
03 S-DATA11 PIC X(1) VALUE X'02' .
01 RCV-S1 .
02 MSG-RCVSYNC PIC X(8) VALUE 'RCVSYNC' .
02 DATA48 PIC X(5) .
02 FILLER PIC X(3) .
02 RCV-S-SEG PIC X(4) VALUE 'FRST' .
02 DATA49 PIC X(4) VALUE SPACE .
02 DATA50 PIC 9(8) .
02 DATA51 PIC 9(8) .
02 RCV-S-LEG PIC 9(9) COMP VALUE 256 .
02 DATA52 PIC X(4) VALUE SPACE .
02 DATA53 PIC X(4) VALUE SPACE .
02 DATA54 PIC X(4) VALUE SPACE .
02 DATA55 PIC X(4) VALUE SPACE .
02 DATA56 PIC X(8) VALUE SPACE .
02 DATA57 PIC X(4) VALUE SPACE .
02 DATA58 PIC X(8) VALUE SPACE .
02 DATA59 PIC X(4) VALUE SPACE .
02 DATA60 PIC 9(9) COMP VALUE ZERO .
02 DATA61 PIC 9(9) COMP VALUE ZERO .
02 DATA62 PIC X(1) VALUE SPACE .
02 DATA63 PIC X(1) VALUE SPACE .
02 DATA64 PIC X(14) VALUE LOW-VALUE .
01 RCV-S2 .
02 DATA65 PIC X(4) VALUE SPACE .
02 R2-TRM-NAM PIC X(8) .
02 DATA67 PIC X(8) VALUE SPACE .
02 DATA68 PIC X(8) VALUE SPACE .
02 DATA69 PIC X(28) VALUE LOW-VALUE .
01 RCV-S3 .
02 DATA70 PIC 9(9) COMP .
02 DATA71 PIC X(8) .
02 DATA72 PIC X(1024) .
01 SND-RCV1 .
02 MSG-SNDRCV PIC X(8) VALUE 'SENDRCV' .
02 DATA76 PIC X(5) .
02 FILLER PIC X(3) .
02 DATA77 PIC X(4) VALUE SPACE .
02 DATA78 PIC X(4) VALUE SPACE .
02 DATA79 PIC 9(8) .
02 DATA80 PIC 9(8) .
02 SND-RCV-LEG PIC 9(9) COMP VALUE 256 .
02 SND-RCV-SEG PIC X(4) VALUE 'EMI' .
02 DATA81 PIC X(4) VALUE SPACE .
02 DATA82 PIC X(4) VALUE SPACE .
02 DATA83 PIC X(4) VALUE SPACE .
02 DATA84 PIC X(8) VALUE SPACE .
02 DATA85 PIC X(4) VALUE SPACE .
02 DATA86 PIC X(8) VALUE SPACE .
02 DATA87 PIC X(4) VALUE SPACE .

```

付録 E ユーザアプリケーションプログラムのコーディング例

```

02     PRO-CD           PIC 9(9)    COMP    VALUE    ZERO.
02     DATA88          PIC 9(9)    COMP    VALUE    ZERO.
02     DATA88-1       PIC X(1)    VALUE    SPACE.
02     DATA88-2       PIC X(1)    VALUE    SPACE.
02     DATA89          PIC X(14)   VALUE    LOW-VALUE.
01     SND-RCV2.
02     MSG              PIC X(4)    VALUE    ' IO  '.
02     SR-TRM-NAM      PIC X(8) .
02     DATA90          PIC X(8)    VALUE    SPACE.
02     DATA91          PIC X(8)    VALUE    SPACE.
02     DATA92          PIC X(28)   VALUE    LOW-VALUE.
01     SND-RCV3.
02     MSG-SEG-LEG     PIC 9(9)    COMP    VALUE    18.
02     DATA94          PIC X(8) .
02     SNDRCV-DATA.
03     SR-DATA1        PIC X(1)    VALUE    X'00'.
03     SR-DATA2        PIC X(1)    VALUE    X'0a'.
03     SR-DATA3        PIC X(4)    VALUE    'OTP1'.
03     SR-DATA4        PIC X(4)    VALUE    'HSR '.
03     SR-DATA5        PIC X(1)    VALUE    X'00'.
03     SR-DATA6        PIC X(1)    VALUE    X'06'.
03     SR-DATA7        PIC X(1)    VALUE    X'00'.
03     SR-DATA8        PIC X(1)    VALUE    X'04'.
03     SR-DATA9        PIC X(2)    VALUE    'NR'.
03     SR-DATA10       PIC X(1)    VALUE    X'00'.
03     SR-DATA11      PIC X(1)    VALUE    X'02'.
01     SND-RCV4.
02     DATA95          PIC 9(9)    COMP.
02     DATA96-1       PIC X(7)    VALUE    SPACE.
02     DATA96-2       PIC X(1) .
02     DATA98          PIC X(248) .
01     SND4.
02     MSG-SNDSYNC     PIC X(8)    VALUE    'SENDSYNC'.
02     DATA100        PIC X(5) .
02     FILLER          PIC X(3) .
02     DATA101        PIC X(4)    VALUE    SPACE.
02     DATA102        PIC X(4)    VALUE    SPACE.
02     DATA103        PIC 9(8) .
02     DATA104        PIC 9(8) .
02     DATA105        PIC 9(9)    COMP    VALUE    ZERO.
02     SND-S-SEG       PIC X(4)    VALUE    'EMI '.
02     DATA106        PIC X(4)    VALUE    SPACE.
02     DATA107        PIC X(4)    VALUE    SPACE.
02     DATA108        PIC X(4)    VALUE    SPACE.
02     DATA109        PIC X(8)    VALUE    SPACE.
02     DATA110        PIC X(4)    VALUE    SPACE.
02     DATA111        PIC X(8)    VALUE    SPACE.
02     DATA112        PIC X(4)    VALUE    SPACE.
02     DATA113        PIC 9(9)    COMP    VALUE    ZERO.
02     DATA114        PIC 9(9)    COMP    VALUE    ZERO.
02     DATA115        PIC X(1)    VALUE    SPACE.
02     DATA116        PIC X(1)    VALUE    SPACE.
02     DATA117        PIC X(14)   VALUE    LOW-VALUE.
01     SND5.
02     SEG-CODE        PIC X(4)    VALUE    SPACE.
02     S5-TRM-NAM     PIC X(8) .
02     DATA120        PIC X(8)    VALUE    SPACE.

```

```

02      DATA121      PIC X(8)    VALUE  SPACE.
02      DATA122      PIC X(28)   VALUE  LOW-VALUE.
01  SND6.
02      DATA123      PIC 9(9)    COMP    VALUE  14.
02      DATA124      PIC X(8) .
02      SEND-DATA.
03      S-DATA11      PIC X(1)    VALUE  X'00'.
03      S-DATA12      PIC X(1)    VALUE  X'0a'.
03      S-DATA13      PIC X(4)    VALUE  'OTP1'.
03      S-DATA14      PIC X(4)    VALUE  'EDP '.
03      S-DATA15      PIC X(1)    VALUE  X'00'.
03      S-DATA16      PIC X(1)    VALUE  X'02'.
03      S-DATA17      PIC X(1)    VALUE  X'00'.
03      S-DATA18      PIC X(1)    VALUE  X'02'.

PROCEDURE          DIVISION.
CALL  'CBLDCMCF'    USING  RCV1      RCV2      RCV3.
MOVE  RCV-TRM-NAM  TO      S2-TRM-NAM.
CALL  'CBLDCMCF'    USING  SND1      SND2      SND3.
MOVE  RCV-TRM-NAM  TO      R2-TRM-NAM.
CALL  'CBLDCMCF'    USING  RCV-S1    RCV-S2    RCV-S3.
MOVE  RCV-TRM-NAM  TO      SR-TRM-NAM.
CALL  'CBLDCMCF'    USING  SND-RCV1  SND-RCV2  SND-RCV3  SND-RCV4.
MOVE  RCV-TRM-NAM  TO      R2-TRM-NAM.
CALL  'CBLDCMCF'    USING  RCV-S1    RCV-S2    RCV-S3.
MOVE  RCV-TRM-NAM  TO      S5-TRM-NAM.
CALL  'CBLDCMCF'    USING  SND4      SND5      SND6.

EXIT  PROGRAM.

```

付録 E.3 ユーザアプリケーションプログラムのコーディング例 (全二重でのデータ操作言語)

(1) コンテンション勝者側

```

*
*****
*   SPPサーブプログラム                               *
*****
*
IDENTIFICATION DIVISION.

PROGRAM-ID. SEND01.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*   ワーク変数   *
DATA DIVISION.
WORKING-STORAGE SECTION.
*   ダイアログ開始要求送信領域   *
01  SEND-AREA1.
02  SE-DATALENG1      PIC 9(4)    COMP  VALUE  58.
02  SE-RSV1           PIC X(2) .
02  SE-DATA1.
03  S1-DATA1         PIC X(1)    VALUE  X'00'.

```

付録 E ユーザアプリケーションプログラムのコーディング例

```

03 S1-DATA2 PIC X(1) VALUE X'0a'.
03 S1-DATA3 PIC X(4) VALUE 'OTP1'.
03 S1-DATA4 PIC X(4) VALUE 'BDR '.
03 S1-DATA5 PIC X(1) VALUE X'00'.
03 S1-DATA6 PIC X(1) VALUE X'2a'.
03 S1-DATA7 PIC X(1) VALUE X'00'.
03 S1-DATA8 PIC X(1) VALUE X'0d'.
03 S1-DATA8-1 PIC X(2) VALUE 'PR'.
03 S1-DATA9 PIC X(9) VALUE 'SEND_TPSU'.
03 S1-DATA10 PIC X(1) VALUE X'00'.
03 S1-DATA11 PIC X(1) VALUE X'02'.
03 S1-DATA12 PIC X(1) VALUE X'00'.
03 S1-DATA13 PIC X(1) VALUE X'0d'.
03 S1-DATA13-1 PIC X(2) VALUE 'PR'.
03 S1-DATA14 PIC X(9) VALUE 'RECV_TPSU'.
03 S1-DATA15 PIC X(1) VALUE X'00'.
03 S1-DATA16 PIC X(1) VALUE X'04'.
03 S1-DATA17 PIC X(1) VALUE X'48'.
03 S1-DATA18 PIC X(1) VALUE X'ff'.
03 S1-DATA19 PIC X(1) VALUE X'00'.
03 S1-DATA20 PIC X(1) VALUE X'02'.
03 S1-DATA21 PIC X(1) VALUE X'00'.
03 S1-DATA22 PIC X(1) VALUE X'02'.
03 S1-DATA23 PIC X(1) VALUE X'00'.
03 S1-DATA24 PIC X(1) VALUE X'04'.
03 S1-DATA25 PIC X(2) VALUE 'AL'.
03 S1-DATA26 PIC X(1) VALUE X'00'.
03 S1-DATA27 PIC X(1) VALUE X'02'.
* ダイアログ開始応答受信領域 *
01 RECV-AREA1.
02 RE-DATALENG1 PIC 9(4) COMP VALUE 1028.
02 RE-RSV1 PIC X(2).
02 RE-DATA1 PIC X(1024).
* データ転送要求送信領域 *
01 SEND-AREA2.
02 SE-DATALENG2 PIC 9(4) COMP VALUE 27.
02 SE-RSV2 PIC X(2).
02 SE-DATA2.
03 S2-DATA1 PIC X(1) VALUE X'00'.
03 S2-DATA2 PIC X(1) VALUE X'0a'.
03 S2-DATA3 PIC X(4) VALUE 'OTP1'.
03 S2-DATA4 PIC X(4) VALUE 'TDR '.
03 S2-DATA5 PIC X(1) VALUE X'00'.
03 S2-DATA6 PIC X(1) VALUE X'02'.
03 S2-DATA7 PIC X(1) VALUE X'00'.
03 S2-DATA8 PIC X(1) VALUE X'0b'.
03 S2-DATA9 PIC X(1) VALUE X'80'.
03 S2-DATA10 PIC X(8) VALUE 'USERDATA'.
* ハンドシェイク指示受信領域 *
01 RECV-AREA2.
02 RE-DATALENG2 PIC 9(4) COMP VALUE 1028.
02 RE-RSV2 PIC X(2).
02 RE-DATA2 PIC X(1024).
* ハンドシェイク応答送信領域 *
01 SEND-AREA3.
02 SE-DATALENG3 PIC 9(4) COMP VALUE 18.
02 SE-RSV3 PIC X(2).

```

```

02 SE-DATA3 .
03 S3-DATA1          PIC X(1)  VALUE X'00' .
03 S3-DATA2          PIC X(1)  VALUE X'0a' .
03 S3-DATA3          PIC X(4)  VALUE 'OTP1' .
03 S3-DATA4          PIC X(4)  VALUE 'HSP ' .
03 S3-DATA5          PIC X(1)  VALUE X'00' .
03 S3-DATA6          PIC X(1)  VALUE X'02' .
03 S3-DATA7          PIC X(1)  VALUE X'00' .
03 S3-DATA8          PIC X(1)  VALUE X'02' .
*   ダイアログ終了要求送信領域   *
01 SEND-AREA4 .
02 SE-DATALENG4      PIC 9(4)  COMP VALUE 22 .
02 SE-RSV4           PIC X(2) .
02 SE-DATA4 .
03 S4-DATA1          PIC X(1)  VALUE X'00' .
03 S4-DATA2          PIC X(1)  VALUE X'0a' .
03 S4-DATA3          PIC X(4)  VALUE 'OTP1' .
03 S4-DATA4          PIC X(4)  VALUE 'EDR ' .
03 S4-DATA5          PIC X(1)  VALUE X'00' .
03 S4-DATA6          PIC X(1)  VALUE X'06' .
03 S4-DATA7          PIC X(1)  VALUE X'00' .
03 S4-DATA8          PIC X(1)  VALUE X'04' .
03 S4-DATA9          PIC X(2)  VALUE 'TR' .
03 S4-DATA10         PIC X(1)  VALUE X'00' .
03 S4-DATA11         PIC X(1)  VALUE X'02' .

*   ダイアログ終了確認受信領域   *
01 RECV-AREA3 .
02 RE-DATALENG3      PIC 9(4)  COMP VALUE 1028 .
02 RE-RSV3           PIC X(2) .
02 RE-DATA3          PIC X(1024) .
*   通信記述項   *
COMMUNICATION SECTION .
*   ダイアログ開始要求の送信   *
CD SEND-OT1
  FOR I-O
    STATUS KEY IS          SE-STATUS1
    SYMBOLIC TERMINAL IS  SE-TERMNAM1
    SYNCHRONOUS MODE IS  SYNC .
*   ダイアログ開始指示の受信   *
CD RECV-IN1
  FOR I-O
    STATUS KEY IS          RE-STATUS1
    SYMBOLIC TERMINAL IS  RE-TERMNAM1
    MESSAGE DATE IS       RE-DATE1
    MESSAGE TIME IS       RE-TIME1
    SYNCHRONOUS MODE IS  SYNC
    WAITING TIME IS RE-WAIT1 .
*   データ転送要求の送信   *
CD SEND-OT2
  FOR I-O
    STATUS KEY IS          SE-STATUS2
    SYMBOLIC TERMINAL IS  SE-TERMNAM2
    SYNCHRONOUS MODE IS  SYNC .
*   ハンドシェイク指示の受信   *
CD RECV-IN2
  FOR I-O

```

```

STATUS KEY IS RE-STATUS2
SYMBOLIC TERMINAL IS RE-TERMNAM2
MESSAGE DATE IS RE-DATE2
MESSAGE TIME IS RE-TIME2
SYNCHRONOUS MODE IS SYNC
WAITING TIME IS RE-WAIT2.
* ハンドシェイク応答の送信 *
CD SEND-OT3
FOR I-O
STATUS KEY IS SE-STATUS3
SYMBOLIC TERMINAL IS SE-TERMNAM3
SYNCHRONOUS MODE IS SYNC.
* ダイアログ終了要求の送信 *
CD SEND-OT4
FOR I-O
STATUS KEY IS SE-STATUS4
SYMBOLIC TERMINAL IS SE-TERMNAM4.
* ダイアログ終了確認の受信 *
CD RECV-IN3
FOR I-O
STATUS KEY IS RE-STATUS3
SYMBOLIC TERMINAL IS RE-TERMNAM3
MESSAGE DATE IS RE-DATE3
MESSAGE TIME IS RE-TIME3
SYNCHRONOUS MODE IS SYNC
WAITING TIME IS RE-WAIT3.
* 通信文 *
PROCEDURE DIVISION.
* ダイアログ開始要求の送信 *
MOVE 'otm01' TO SE-TERMNAM1.
SEND SEND-OT1
FROM SEND-AREA1.
* ダイアログ開始確認の受信 *
MOVE 'otm01' TO RE-TERMNAM1.
MOVE '00010000' TO RE-WAIT1.
RECEIVE RECV-IN1
SEGMENT
INTO RECV-AREA1.
* データ転送要求の送信 *
MOVE 'otm01' TO SE-TERMNAM2.
SEND SEND-OT2
FROM SEND-AREA2.
* ハンドシェイク指示の受信 *
MOVE 'otm01' TO RE-TERMNAM2.
MOVE '00010000' TO RE-WAIT2.
RECEIVE RECV-IN2
SEGMENT
INTO RECV-AREA2.
* ハンドシェイク応答の送信 *
MOVE 'otm01' TO SE-TERMNAM3.
SEND SEND-OT3
FROM SEND-AREA3.
* ダイアログ終了要求の送信 *
MOVE 'otm01' TO SE-TERMNAM4.
DISABLE SEND-OT4
WITH SEND-AREA4.
* ダイアログ終了確認の受信 *

```

```

MOVE 'otm01' TO RE-TERMNAM3.
MOVE '00010000' TO RE-WAIT3.
RECEIVE RECV-IN3
      SEGMENT
      INTO RECV-AREA3.
*   終了処理   *
      EXIT PROGRAM.

```

(2) コンテンション敗者側

```

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

      PROGRAM-ID. RECV01.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*   ワーク変数   *
DATA DIVISION.
WORKING-STORAGE SECTION.
*   ダイアログ開始指示受信領域   *
01   RECV-AREA1.
      02 RE-DATALENG1      PIC 9(4)  COMP VALUE 1028.
      02 RE-RSV1          PIC X(2) .
      02 RE-DATA1         PIC X(1024) .
*   ダイアログ開始応答送信領域   *
01   SEND-AREA1.
      02 SE-DATALENG1     PIC 9(4)  COMP VALUE 22.
      02 SE-RSV1         PIC X(2) .
      02 SE-DATA1.
          03 S1-DATA1     PIC X(1)  VALUE X'00'.
          03 S1-DATA2     PIC X(1)  VALUE X'0a'.
          03 S1-DATA3     PIC X(4)  VALUE 'OTP1'.
          03 S1-DATA4     PIC X(4)  VALUE 'BDPA'.
          03 S1-DATA5     PIC X(1)  VALUE X'00'.
          03 S1-DATA6     PIC X(1)  VALUE X'06'.
          03 S1-DATA7     PIC X(1)  VALUE X'00'.
          03 S1-DATA8     PIC X(1)  VALUE X'04'.
          03 S1-DATA9     PIC X(2)  VALUE 'AC'.
          03 S1-DATA10    PIC X(1)  VALUE X'00'.
          03 S1-DATA11    PIC X(1)  VALUE X'02'.
*   データ転送指示受信領域   *
01   RECV-AREA2.
      02 RE-DATALENG2     PIC 9(4)  COMP VALUE 1028.
      02 RE-RSV2         PIC X(2) .
      02 RE-DATA2        PIC X(1024) .
*   ハンドシェイク要求送信領域   *
01   SEND-AREA2.
      02 SE-DATALENG2     PIC 9(4)  COMP VALUE 22.
      02 SE-RSV2         PIC X(2) .
      02 SE-DATA2.
          03 S2-DATA1     PIC X(1)  VALUE X'00'.
          03 S2-DATA2     PIC X(1)  VALUE X'0a'.

```

付録 E ユーザアプリケーションプログラムのコーディング例

```

03 S2-DATA3 PIC X(4) VALUE 'OTP1'.
03 S2-DATA4 PIC X(4) VALUE 'HSR'.
03 S2-DATA5 PIC X(1) VALUE X'00'.
03 S2-DATA6 PIC X(1) VALUE X'06'.
03 S2-DATA7 PIC X(1) VALUE X'00'.
03 S2-DATA8 PIC X(1) VALUE X'04'.
03 S2-DATA9 PIC X(2) VALUE 'NR'.
03 S2-DATA10 PIC X(1) VALUE X'00'.
03 S2-DATA11 PIC X(1) VALUE X'02'.
* ハンドシェイク確認受信領域 *
01 RECV-AREA3.
02 RE-DATALENG3 PIC 9(4) COMP VALUE 1028.
02 RE-RSV3 PIC X(2).
02 RE-DATA3 PIC X(1024).
* ダイアログ終了指示受信領域 *
01 RECV-AREA4.
02 RE-DATALENG4 PIC 9(4) COMP VALUE 1028.
02 RE-RSV4 PIC X(2).
02 RE-DATA4 PIC X(1024).
* ダイアログ終了応答送信領域 *
01 SEND-AREA3.
02 SE-DATALENG3 PIC 9(4) COMP VALUE 18.
02 SE-RSV3 PIC X(2).
02 SE-DATA3.
03 S3-DATA1 PIC X(1) VALUE X'00'.
03 S3-DATA2 PIC X(1) VALUE X'0a'.
03 S3-DATA3 PIC X(4) VALUE 'OTP1'.
03 S3-DATA4 PIC X(4) VALUE 'EDP'.
03 S3-DATA5 PIC X(1) VALUE X'00'.
03 S3-DATA6 PIC X(1) VALUE X'02'.
03 S3-DATA7 PIC X(1) VALUE X'00'.
03 S3-DATA8 PIC X(1) VALUE X'02'.
* 通信記述項 *
COMMUNICATION SECTION.
* ダイアログ開始要求の受信 *
CD RECV-IN1
FOR INPUT
STATUS KEY IS RE-STATUS1
SYMBOLIC TERMINAL IS RE-TERMNAM1
MESSAGE DATE IS RE-DATE1
MESSAGE TIME IS RE-TIME1.
* ダイアログ開始応答の送信 *
CD SEND-OT1
FOR I-O
STATUS KEY IS SE-STATUS1
SYMBOLIC TERMINAL IS SE-TERMNAM1
SYNCHRONOUS MODE IS SYNC.
* データ転送指示の受信 *
CD RECV-IN2
FOR I-O
STATUS KEY IS RE-STATUS2
SYMBOLIC TERMINAL IS RE-TERMNAM2
MESSAGE DATE IS RE-DATE2
MESSAGE TIME IS RE-TIME2
SYNCHRONOUS MODE IS SYNC
WAITING TIME IS RE-WAIT2.
* ハンドシェイク要求の送信 *

```



```

CD SEND-OT2
FOR I-O
STATUS KEY IS SE-STATUS2
SYMBOLIC TERMINAL IS SE-TERMNAM2
SYNCHRONOUS MODE IS SYNC.
* ハンドシェイク確認の受信 *
CD RECV-IN3
FOR I-O
STATUS KEY IS RE-STATUS3
SYMBOLIC TERMINAL IS RE-TERMNAM3
MESSAGE DATE IS RE-DATE3
MESSAGE TIME IS RE-TIME3
SYNCHRONOUS MODE IS SYNC
WAITING TIME IS RE-WAIT3.
* ダイアログ終了指示の受信 *
CD RECV-IN4
FOR I-O
STATUS KEY IS RE-STATUS4
SYMBOLIC TERMINAL IS RE-TERMNAM4
MESSAGE DATE IS RE-DATE4
MESSAGE TIME IS RE-TIME4
SYNCHRONOUS MODE IS SYNC
WAITING TIME IS RE-WAIT4.
* ダイアログ終了応答の送信 *
CD SEND-OT3
FOR I-O
STATUS KEY IS SE-STATUS3
SYMBOLIC TERMINAL IS SE-TERMNAM3.
* 通信文 *
PROCEDURE DIVISION.
* ダイアログ開始指示の受信 *
RECEIVE RECV-IN1
FIRST SEGMENT
INTO RECV-AREA1.
* ダイアログ開始応答の送信 *
MOVE RE-TERMNAM1 TO SE-TERMNAM1.
SEND SEND-OT1
FROM SEND-AREA1.
* データ転送指示の受信 *
MOVE RE-TERMNAM1 TO RE-TERMNAM2.
MOVE '00010000' TO RE-WAIT2.
RECEIVE RECV-IN2
SEGMENT
INTO RECV-AREA2.
* ハンドシェイク要求の送信 *
MOVE RE-TERMNAM1 TO SE-TERMNAM2.
SEND SEND-OT2
FROM SEND-AREA2.
* ハンドシェイク確認の受信 *
MOVE RE-TERMNAM1 TO RE-TERMNAM3.
MOVE '00010000' TO RE-WAIT3.
RECEIVE RECV-IN3
SEGMENT
INTO RECV-AREA3.
* ダイアログ終了指示の受信 *
MOVE RE-TERMNAM1 TO RE-TERMNAM4.
MOVE '00010000' TO RE-WAIT4.

```

```

        RECEIVE RECV-IN4
        SEGMENT
        INTO RECV-AREA4.
*   ダイアログ終了応答の送信   *
        MOVE RE-TERMNAM1 TO SE-TERMNAM3.
        DISABLE SEND-OT3
        WITH SEND-AREA3.
*   終了処理   *
        EXIT PROGRAM.

```

付録 E.4 ユーザアプリケーションプログラムのコーディング例 (半二重での C 言語)

(1) コンテンション勝者側 (SPP)

```

#include <dcmcf.h>
#include <string.h>

#define        WAITTIME        60

void        SEND01(in, in_len, out, out_len)

        char                *in        ;
        unsigned            DCLONG    in_len    ;
        char                *out       ;
        unsigned            DCLONG    out_len   ;
{
    /*   引数定義   */
        /* 引数を使用している関数 */
        /* sendsync, recvsync, sendrecv, receive */
        DCLONG    action            ;/* 0            0            0            0 */
        DCLONG    commform          ;/* 0            0            0            0 */
        char      termnam[16]       ;/* 0            0            0            0 */
        char      resv01[16]        ;/* 0            0            0            0 */
        char      senddata[256]     ;/* 0            -            0            - */
        DCLONG    sdataleng         ;/* 0            -            0            - */
        char      recvdata[256]     ;/* -            0            0            0 */
        DCLONG    rdataleng         ;/* -            0            0            0 */
        DCLONG    time              ;/* -            0            0            0 */
        DCLONG    wachtime          ;/* 0            0            0            - */
        char      resv02[16]        ;/* 0            -            -            - */
        DCLONG    opcd              ;/* 0            -            -            - */
        DCLONG    inbufleng         ;/* -            0            0            0 */

    /*   変数定義 & 初期化   */
        int        rtn = 0x00    ;

    /*   引数初期化   */
        action      =    NULL;
        commform    =    NULL;
        memset(termnam, 0, 16)   ;
        memset(resv01, 0, 16)    ;
        memset(senddata, 0, 256);
        sdataleng   =    NULL;
        memset(recvdata, 0, 256);
        rdataleng   =    NULL;

```

```

time           = NULL;
wachttime     = NULL;
memset(resv02, 0, 16) ;
opcd          = NULL;
inbufleng     = NULL;

/*
/*   ダイアログ開始要求送信 (TP-BEGIN-DIALOGUE)   */
/*
/*
/*   sendsync引数設定 (初期化)   */
action        = DCMCFEMI ;
commform      = DCNOFLAGS ;
strcpy(terminam, "otm01") ;
resv01[0]     = NULL ;
/* senddataイニシャライズは省略 */
sdataleng     = 54 ;
resv02[0]     = NULL ;
opcd          = DCNOFLAGS ;
wachttime     = WAITTIME ;

/* senddata 設定 */
/* 先頭8バイト目までは、MCFが使用する */
senddata[8] = 0x00 ; /* レングス 01 */
senddata[9] = 0x0a ; /* サービスプリミティブの長さ */

strcpy(senddata + 10, "OTP1") ; /* 送信手順(OTP1) */
strcpy(senddata + 14, "BDR ") ;
/* サービスプリミティブコード */
/* TP-BEGIN-DIALOGUE req */

senddata[18] = 0x00 ; /* レングス 02 */
senddata[19] = 0x2a ; /* パラメタ全体の長さ */

senddata[20] = 0x00 ; /* レングス 03 */
senddata[21] = 0x0d ; /* 発呼側TPSUタイトルの長さ */

strcpy(senddata + 22, "PR") ; /* 型コード */
strcpy(senddata + 24, "SEND TPSU");
/* 発呼側TPSUタイトル(パラメタ1) */

senddata[33] = 0x00 ; /* レングス 04 */
senddata[34] = 0x02 ; /* 着呼側AP(パラメタ 2省略) */

senddata[35] = 0x00 ; /* レングス 05 */
senddata[36] = 0x0d ; /* 着呼側TPSUタイトルの長さ */

strcpy(senddata + 37, "PR") ; /* 型コード */
strcpy(senddata + 39, "RECV TPSU") ;
/* 着呼側TPSUタイトル(パラメタ3) */

senddata[48] = 0x00 ; /* レングス 06 */
senddata[49] = 0x04 ; /* 予備の長さ */

senddata[50] = 0x88 ;
/* 半二重&ハンドシェイク機能単位(パラメタ 4) */
senddata[51] = 0xff ; /* 予備 */

senddata[52] = 0x00 ; /* レングス 07 */

```



```

/*   sendsync引数設定 (初期化)   */
action          =   DCMCFEMI      ;
commform       =   DCNOFLAGS     ;
resv01[0]      =   NULL          ;
sdataleng      =   23            ;
resv02[0]      =   NULL          ;
opcd           =   DCNOFLAGS     ;
wachtme        =   WAITTIME     ;

/*   senddata clear   */
memset(senddata, 0, 256)        ;

/*   senddata 設定   */
/*   先頭8バイト目までは、MCFが使用する   */
senddata[8] = 0x00 ; /* レングス 01 */
senddata[9] = 0x0a ; /* サービスプリミティブの長さ */

strcpy(senddata + 10, "OTP1") ; /* 送信手順(OTP1) */
strcpy(senddata + 14, "TDR ") ;
/* サービスプリミティブコード */
/* TP-DATA req */

senddata[18] = 0x00 ; /* レングス 02 */
senddata[19] = 0x02 ; /* パラメタ(省略) */

senddata[20] = 0x00 ; /* レングス 03 */
senddata[21] = 0x0b ; /* ユーザ情報の長さ */

senddata[22] = 0x80 ; /* 連結状態 */

strcpy(senddata + 23, "USERDATA") ; /* ユーザデータ */

/*   データ送信(sendsync)開始   */

rtn = dc_mcf_sendsync(action, commform, termnam, resv01,
                      senddata, sdataleng, resv02, opcd,
                      wachtme);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*   データ送信(sendsync)終了   */

/*
/*   ハンドシェイク&制御権譲渡要求送信   */
/*   (TP-HANDSHAKE-AND-GRANT-CONTROL)   */
/*
/*
/*   sendsync引数設定 (初期化)   */
action          =   DCMCFEMI      ;
commform       =   DCNOFLAGS     ;
resv01[0]      =   NULL          ;
sdataleng      =   18            ;
resv02[0]      =   NULL          ;
opcd           =   DCNOFLAGS     ;
wachtme        =   WAITTIME     ;

```

```

memset(senddata, 0, 256);

/* sendsync 設定 */
/* 先頭8バイト目までは、MCFが使用する */
senddata[8] = 0x00 ; /* レングス 01 */
senddata[9] = 0x0a ; /* サービスプリミティブの長さ */

strcpy(senddata + 10, "OTP1") ; /* 送信手順(OTP1) */
strcpy(senddata + 14, "HGR ") ;
/* サービスプリミティブコード */
/* TP-HANDSHAKE-AND- */
/* GRANT-CONTROL req */

senddata[18] = 0x00 ; /* レングス 02 */
senddata[19] = 0x06 ; /* パラメタ全体の長さ */

senddata[20] = 0x00 ; /* レングス 03 */
senddata[21] = 0x04 ; /* 即時確認の長さ */

strcpy(senddata+22, "UR") ; /* 即時確認(パラメタ 1) */

senddata[24] = 0x00 ; /* レングス 04 */
senddata[25] = 0x02 ; /* ユーザ情報(省略) */

/* メッセージ送信開始(sendsync) */

rtn = dc_mcf_sendsync(action, commform, termnam, resv01,
                    senddata, sdataleng, resv02, opcd,
                    wachtime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/* メッセージ送信終了(sendsync) */

/*
/*          ハンドシェイク&制御権譲渡応答受信          */
/*          (TP-HANDSHAKE-AND-GRANT-CONTROL)          */
/*
/*
/*          recvsync引数設定(初期化)          */
action          = DCMCFRST          ;
commform        = DCNOFLAGS        ;
resv01[0]       = NULL              ;
/*recvdataイニシャライズは省略*/
rdataleng       = NULL              ;
inbufleng       = 256               ;
time            = NULL              ;
wachtime        = WAITTIME         ;

memset(senddata, 0, 256);

/* メッセージ受信開始(recvsync) */

rtn = dc_mcf_recvsync(action, commform, termnam, resv01,
                    recvdata, &rdataleng,
                    inbufleng, &time, wachtime);

```

```

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      メッセージ受信終了 (recvsync)      */

/*                                          */
/*      TP_DATA受信 (TP-DATA)              */
/*                                          */

/*      recvsync引数設定 (初期化)      */
/*                                          */
action      = DCMCFRST      ;
commform    = DCNOFLAGS    ;
resv01[0]   = NULL         ;
rdataleng   = NULL         ;
inbufleng   = 256          ;
time        = NULL         ;
wachttime   = WAITTIME     ;

/*      recvdata clear      */
memset (recvdata, 0, 256) ;

/*      データ受信開始 (recvsync)      */

rtn = dc_mcf_recvsync (action, commform, termnam, resv01,
                      recvdata, & rdataleng,
                      inbufleng, & time, wachttime );

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      データ受信終了 (recvsync)      */

/*                                          */
/*      ダイアログ終了要求受信 (TP-END-DIALOGUE)      */
/*                                          */

/*      recvsync引数設定 (初期化)      */
/*                                          */
action      = DCMCFRST      ;
commform    = DCNOFLAGS    ;
resv01[0]   = NULL         ;
rdataleng   = NULL         ;
inbufleng   = 256          ;
time        = NULL         ;
wachttime   = WAITTIME     ;

/*      recvdata clear      */
memset (recvdata, 0, 256) ;

/*      メッセージ受信開始 (recvsync)      */

rtn = dc_mcf_recvsync (action, commform, termnam, resv01,
                      recvdata, & rdataleng,
                      inbufleng, & time, wachttime);

if ( rtn != DCMCFRTN_00000 )
{

```

```

        goto ERROR;
    }
    /*      メッセージ受信終了(recvsync)      */

/*      ダイアログ終了応答送信(TP-END-DIALOGUE)      */
/*      */
/*      sendsync引数設定 (初期化)      */
action          =   DCMCFEMI          ;
commform        =   DCNOFLAGS         ;
resv01[0]       =   NULL              ;
sdataleng       =   14                ;
resv02[0]       =   NULL              ;
opcd            =   DCNOFLAGS         ;
wachtme         =   WAITTIME         ;

/*      senddata clear      */
memset(senddata, 0, 256)      ;

/*      senddata 設定      */
/*      先頭8バイト目までは、MCFが使用する      */
senddata[8] = 0x00 ; /* レンクス 01 */
senddata[9] = 0x0a ; /* サービスプリミティブの長さ */

strcpy(senddata + 10, "OTP1") ; /* 送信手順(OTP1) */
strcpy(senddata + 14, "EDP ") ;
/* サービスプリミティブコード */
/* TP-END-DIALOGUE rsp */

senddata[18] = 0x00 ; /* レンクス 02 */
senddata[19] = 0x02 ; /* パラメタ(省略) */

senddata[20] = 0x00 ; /* レンクス 03 */
senddata[21] = 0x02 ; /* ユーザ情報(省略) */

/*      メッセージ送信開始(sendsync)      */

rtn = dc_mcf_sendsync(action, commform, termnam, resv01,
                      senddata, sdataleng, resv02, opcd,
                      wachtme);

/*      メッセージ送信終了(sendsync)      */

ERROR : ;
}

```

(2) コンテンション敗者側 (MHP)

```

#include <dcmcf.h>
#include <string.h>

#define      WAITTIME      60

void      RECV01(in, in_len, out, out_len)

            char          *in          ;
            unsigned      DCLONG      in_len  ;

```



```

        char          *out      ;
unsigned   DCLONG    out_len  ;
{
    /*  引数定義      */          /*引数を使用している関数*/
        /* sendsync, recvsync, sendrecv, receive */
    DCLONG  action      ;/*  0      0      0      0 */
    DCLONG  commform    ;/*  0      0      0      0 */
    char    termnam[16] ;/*  0      0      0      0 */
    char    resv01[16]  ;/*  0      0      0      0 */
    char    senddata[256];/*  0      -      0      - */
    DCLONG  sdataleng   ;/*  0      -      0      - */
    char    recvdata[256];/* -      0      0      0 */
    DCLONG  rdataleng   ;/* -      0      0      0 */
    DCLONG  time        ;/* -      0      0      0 */
    DCLONG  wachtime    ;/*  0      0      0      - */
    char    resv02[16]  ;/*  0      -      -      - */
    DCLONG  opcd        ;/*  0      -      -      - */
    DCLONG  inbufleng   ;/* -      0      0      0 */

    /*  変数定義 & 初期化  */
    int     rtn = NULL  ;

    /*  引数初期化      */
    action      =  NULL;
    commform    =  NULL;
    memset( termnam, 0, 16) ;
    memset( resv01, 0, 16) ;
    memset( recvdata, 0, 256);
    sdataleng   =  NULL;
    memset( recvdata, 0, 256);
    rdataleng   =  NULL;
    time        =  NULL;
    wachtime    =  NULL;
    memset( resv02, 0, 16) ;
    opcd        =  NULL;
    inbufleng   =  NULL;

    /*
    /*      ダイアログ開始要求受信 (TP-BEGIN-DIALOGUE)
    /*
    /*      receive引数設定 (初期化)
    action      =  DCMCFRST ;
    commform    =  DCNOFLAGS ;
    termnam[0]  =  NULL ;
    /*recvdataイニシャルイズは省略*/
    resv01[0]   =  NULL ;
    rdataleng   =  NULL ;
    inbufleng   =  256 ;
    time        =  NULL ;

    /*      ダイアログ開始要求受信開始 (receive)
    rtn = dc_mcf_receive(action, commform, termnam, resv01,
        recvdata, & rdataleng, inbufleng,
        & time);

    if ( rtn != DCMCFRTN_00000 )

```

```

{
    goto ERROR;
}
/*      ダイアログ開始要求受信終了 (receive)      */

/*      ダイアログ開始応答送信 (TP-BEGIN-DIALOGUE)      */
/*      */
/*      */
/*      sendsync 引数設定 (初期化)      */
action          = DCMCFEMI      ;
commform        = DCNOFLAGS     ;
resv01[0]       = NULL          ;
sdatalelng      = 18            ;
resv02[0]       = NULL          ;
opcd            = DCNOFLAGS     ;
wachtme         = WAITTIME     ;

/*      senddata 設定      */
/*      先頭8バイト目までは、MCFが使用する      */
senddata[8] = 0x00 ; /* レングス 01 */
senddata[9] = 0x0a ; /* サービスプリミティブの長さ */

strcpy(senddata + 10, "OTP1") ; /* 送信手順(OTP1) */
strcpy(senddata + 14, "BDPA") ;
/* サービスプリミティブコード */
/* TP-BEGIN-DIALOGUE rsp */

senddata[18] = 0x00 ; /* レングス 02 */
senddata[19] = 0x06 ; /* パラメタ全体の長さ */

senddata[20] = 0x00 ; /* レングス 03 */
senddata[21] = 0x04 ; /* 結果の長さ */

strcpy(senddata + 22, "AC") ; /* 結果(パラメタ 1) */

senddata[24] = 0x00 ; /* レングス 04 */
senddata[25] = 0x02 ; /* ユーザ情報(省略) */

/*      メッセージ送信開始 (sendsync)      */

rtn = dc_mcf_sendsync(action, commform, termnam, resv01,
                      senddata, sdatalelng, resv02, opcd,
                      wachtme);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      メッセージ送信終了 (sendsync)      */

/*      */
/*      TP_DATA受信 (TP-DATA)      */
/*      */
/*      */
/*      recvsync 引数設定 (初期化)      */
action          = DCMCFRST      ;
commform        = DCNOFLAGS     ;

```

```

resv01[0]      =  NULL      ;
rdataleng     =  NULL      ;
inbufleng     =  256      ;
time          =  NULL      ;
wachttime     =  WAITTIME ;

/*  recvdata clear */
memset(recvdata, 0, 256) ;

/*  データ受信開始(recvsync) */

rtn = dc_mcf_recvsync(action, commform, termnam, resv01,
                      recvdata, & rdataleng,
                      inbufleng, & time, wachttime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*  データ受信終了(recvsync) */

/*
/*      ハンドシェイク&制御権譲渡要求受信          */
/*      (TP-HANDSHAKE-AND-GRANT-CONTROL)          */
/*
/*
/*  recvsync引数設定 (初期化) */
action      =  DCMCFRST ;
commform    =  DCNOFLAGS ;
resv01[0]   =  NULL ;
rdataleng   =  NULL ;
inbufleng   =  256 ;
time        =  NULL ;
wachttime   =  WAITTIME ;

/*  recvdata clear */
memset(recvdata, 0, 256) ;

/*  メッセージ受信開始(recvsync) */

rtn = dc_mcf_recvsync(action, commform, termnam, resv01,
                      recvdata, & rdataleng,
                      inbufleng, & time, wachttime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*  メッセージ受信終了(recvsync) */

/*
/*      ハンドシェイク&制御権譲渡応答送信          */
/*      (TP-HANDSHAKE-AND-GRANT-CONTROL)          */
/*
/*
/*  sendsync引数設定 (初期化) */
action      =  DCMCFEMI ;
commform    =  DCNOFLAGS ;

```



```

strcpy(senddata + 10, "OTP1") ; /* 送信手順(OTP1) */
strcpy(senddata + 14, "TDR ") ;
/* サービスプリミティブコード */
/* TP_DATA req */

senddata[18] = 0x00 ; /* レンクス 02 */
senddata[19] = 0x02 ; /* パラメタ(省略) */

senddata[20] = 0x00 ; /* レンクス 03 */
senddata[21] = 0x0b ; /* ユーザ情報の長さ */

senddata[22] = 0x80 ; /* 連結状態 */

strcpy(senddata + 23, "USERDATA") ; /* ユーザデータ */

/* データ送信(sendsync)開始 */

rtn = dc_mcf_sendsync(action, commform, termnam, resv01,
                    senddata, sdataleng, resv02, opcd,
                    wachtime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/* データ送信(sendsync)終了 */

/*
/*   ダイアログ終了要求送受信 (TP-END-DIALOGUE)
/*
/*
/*   sendrecv引数設定
/*
action          = DCMCFEMI ;
commform       = DCMCFIO ;
resv01[0]      = NULL ;
sdataleng      = 18 ;
rdataleng      = NULL ;
inbufleng      = 256 ;
time           = NULL ;
wachtime       = WAITTIME ;

/* senddata clear
memset(senddata, 0, 256) ;
memset(recvdata, 0, 256) ;

/* senddata 設定
/* 先頭8バイト目までは、MCFが使用する
senddata[8] = 0x00 ; /* レンクス 01
senddata[9] = 0x0a ; /* サービスプリミティブの長さ

strcpy(senddata + 10, "OTP1") ; /* 送信手順(OTP1)
strcpy(senddata + 14, "EDR ") ;
/* サービスプリミティブコード
/* TP-END-DIALOGUE req

senddata[18] = 0x00 ; /* レンクス 02
senddata[19] = 0x06 ; /* パラメタ全体の長さ

```

```

senddata[20] = 0x00 ; /* レンダス 03 */
senddata[21] = 0x04 ; /* 確認項目の長さ */

strcpy(senddata + 22, "TR") ; /* 確認項目(パラメタ 1) */

senddata[24] = 0x00 ; /* レンダス 04 */
senddata[25] = 0x02 ; /* ユーザ情報 */

/*      メッセージ送受信開始(sendrecv)      */

rtn = dc_mcf_sendrecv(action, commform, termnam, resv01,
                      senddata, sdataleng,
                      recvdata, & rdataleng,
                      inbufleng, & time, wachtime);

if ( rtn != DCMCFRTN_00000 )
{
    goto ERROR;
}
/*      メッセージ送受信終了(sendrecv)      */

ERROR : ;
}

```

付録 E.5 ユーザアプリケーションプログラムのコーディング例 (半二重での COBOL 言語)

(1) コンテンション勝者側 (SPP)

```

IDENTIFICATION  DIVISION.
PROGRAM-ID.      SEND01.

ENVIRONMENT     DIVISION.

DATA            DIVISION.
WORKING-STORAGE SECTION.

01  SND-S1.
    02  MSG-SNDSYNC  PIC X(8)    VALUE  'SENDSYNC'.
    02  DATA1      PIC X(5) .
    02  FILLER      PIC X(3) .
    02  DATA2      PIC X(4)    VALUE  SPACE.
    02  DATA3      PIC X(4)    VALUE  SPACE.
    02  DATA4      PIC 9(8) .
    02  DATA5      PIC 9(8) .
    02  DATA6      PIC 9(9)    COMP   VALUE  ZERO.
    02  SND-S-SEG   PIC X(4)    VALUE  'EMI'.
    02  DATA7      PIC X(4)    VALUE  SPACE.
    02  DATA8      PIC X(4)    VALUE  SPACE.
    02  DATA9      PIC X(4)    VALUE  SPACE.
    02  DATA10     PIC X(8)    VALUE  SPACE.
    02  DATA11     PIC X(4)    VALUE  SPACE.
    02  DATA12     PIC X(8)    VALUE  SPACE.
    02  DATA13     PIC X(4)    VALUE  SPACE.
    02  DATA14     PIC 9(9)    COMP   VALUE  ZERO.

```

```

02 DATA15 PIC 9(9) COMP VALUE ZERO.
02 DATA16 PIC X(1) VALUE SPACE.
02 DATA17 PIC X(1) VALUE SPACE.
02 DATA18 PIC X(14) VALUE LOW-VALUE.
01 SND-S2.
02 SEG-CODE PIC X(4) VALUE SPACE.
02 TRM-NAME PIC X(8) VALUE 'otm01 '.
02 DATA19 PIC X(8) VALUE SPACE.
02 DATA20 PIC X(8) VALUE SPACE.
02 DATA21 PIC X(28) VALUE LOW-VALUE.
01 SND-S3.
02 DATA22 PIC 9(9) COMP VALUE 54.
02 DATA23 PIC X(8).
02 SEND-DATA.
03 S-DATA1 PIC X(1) VALUE X'00'.
03 S-DATA2 PIC X(1) VALUE X'0a'.
03 S-DATA3 PIC X(4) VALUE 'OTP1'.
03 S-DATA4 PIC X(4) VALUE 'BDR '.
03 S-DATA5 PIC X(1) VALUE X'00'.
03 S-DATA6 PIC X(1) VALUE X'2a'.
03 S-DATA7 PIC X(1) VALUE X'00'.
03 S-DATA8 PIC X(1) VALUE X'0d'.
03 S-DATA8-1 PIC X(2) VALUE 'PR'.
03 S-DATA9 PIC X(9) VALUE 'SEND-TPSU'.
03 S-DATA10 PIC X(1) VALUE X'00'.
03 S-DATA11 PIC X(1) VALUE X'02'.
03 S-DATA12 PIC X(1) VALUE X'00'.
03 S-DATA13 PIC X(1) VALUE X'0d'.
03 S-DATA13-1 PIC X(2) VALUE 'PR'.
03 S-DATA14 PIC X(9) VALUE 'RECV-TPSU'.
03 S-DATA15 PIC X(1) VALUE X'00'.
03 S-DATA16 PIC X(1) VALUE X'04'.
03 S-DATA17 PIC X(1) VALUE X'88'.
03 S-DATA18 PIC X(1) VALUE X'ff'.
03 S-DATA19 PIC X(1) VALUE X'00'.
03 S-DATA20 PIC X(1) VALUE X'02'.
03 S-DATA21 PIC X(1) VALUE X'00'.
03 S-DATA22 PIC X(1) VALUE X'02'.
03 S-DATA23 PIC X(1) VALUE X'00'.
03 S-DATA24 PIC X(1) VALUE X'04'.
03 S-DATA25 PIC X(2) VALUE 'AL'.
03 S-DATA26 PIC X(1) VALUE X'00'.
03 S-DATA27 PIC X(1) VALUE X'02'.
01 SND-S4.
02 MSG-SNDSYNC PIC X(8) VALUE 'SENDSYNC'.
02 DATA25 PIC X(5).
02 FILLER PIC X(3).
02 DATA26 PIC X(4) VALUE SPACE.
02 DATA27 PIC X(4) VALUE SPACE.
02 DATA28 PIC 9(8).
02 DATA29 PIC 9(8).
02 DATA30 PIC 9(9) COMP VALUE ZERO.
02 SND-S-SEG PIC X(4) VALUE 'EMI '.
02 DATA31 PIC X(4) VALUE SPACE.
02 DATA32 PIC X(4) VALUE SPACE.
02 DATA33 PIC X(4) VALUE SPACE.

```

付録 E ユーザアプリケーションプログラムのコーディング例

```

02     DATA34     PIC X(8)     VALUE     SPACE.
02     DATA35     PIC X(4)     VALUE     SPACE.
02     DATA36     PIC X(8)     VALUE     SPACE.
02     DATA37     PIC X(4)     VALUE     SPACE.
02     DATA38     PIC 9(9)     COMP     VALUE     ZERO.
02     DATA39     PIC 9(9)     COMP     VALUE     ZERO.
02     DATA40     PIC X(1)     VALUE     SPACE.
02     DATA41     PIC X(1)     VALUE     SPACE.
02     DATA42     PIC X(14)    VALUE     LOW-VALUE.
01     SND-S5.
02     SEG-CODE    PIC X(4)     VALUE     SPACE.
02     TRM-NAME    PIC X(8)     VALUE     'otm01  '.
02     DATA43     PIC X(8)     VALUE     SPACE.
02     DATA44     PIC X(8)     VALUE     SPACE.
02     DATA45     PIC X(28)    VALUE     LOW-VALUE.
01     SND-S6.
02     DATA46     PIC 9(9)     COMP     VALUE     23.
02     DATA47     PIC X(8) .
02     SEND-DATA2.
03     S-DATA281  PIC 9(4)     COMP     VALUE     10.
03     S-DATA282  PIC X(4)     VALUE     'OTP1' .
03     S-DATA283  PIC X(4)     VALUE     'TDR  ' .
03     S-DATA285  PIC 9(4)     COMP     VALUE     2.

03     S-DATA287  PIC 9(4)     VALUE     11.
03     S-DATA289  PIC X(1)     VALUE     X'80' .
03     S-DATA290  PIC X(8)     VALUE     'USERDATA' .
01     SND-S7.
02     MSG-SNDSYNC PIC X(8)     VALUE     'SENDSYNC' .
02     DATA49     PIC X(5) .
02     FILLER      PIC X(3) .
02     DATA50     PIC X(4)     VALUE     SPACE.
02     DATA51     PIC X(4)     VALUE     SPACE.
02     DATA52     PIC 9(8) .
02     DATA53     PIC 9(8) .
02     DATA54     PIC 9(9)     COMP     VALUE     ZERO.
02     SND-S-SEG   PIC X(4)     VALUE     'EMI  ' .
02     DATA55     PIC X(4)     VALUE     SPACE.
02     DATA56     PIC X(4)     VALUE     SPACE.
02     DATA57     PIC X(4)     VALUE     SPACE.
02     DATA58     PIC X(8)     VALUE     SPACE.
02     DATA59     PIC X(4)     VALUE     SPACE.
02     DATA60     PIC X(8)     VALUE     SPACE.
02     DATA61     PIC X(4)     VALUE     SPACE.
02     DATA62     PIC 9(9)     COMP     VALUE     ZERO.
02     DATA63     PIC 9(9)     COMP     VALUE     ZERO.
02     DATA64     PIC X(1)     VALUE     SPACE.
02     DATA65     PIC X(1)     VALUE     SPACE.
02     DATA66     PIC X(14)    VALUE     LOW-VALUE.
01     SND-S8.
02     SEG-CODE    PIC X(4)     VALUE     SPACE.
02     TRM-NAME    PIC X(8)     VALUE     'otm01  '.
02     DATA67     PIC X(8)     VALUE     SPACE.
02     DATA68     PIC X(8)     VALUE     SPACE.
02     DATA69     PIC X(28)    VALUE     LOW-VALUE.
01     SND-S9.

```



```

02 DATA70 PIC 9(9) COMP VALUE 18.
02 DATA71 PIC X(8).
02 SEND-DATA3.
03 S-DATA329 PIC X(1) VALUE X'00'.
03 S-DATA330 PIC X(1) VALUE X'0a'.
03 S-DATA331 PIC X(4) VALUE 'OTP1'.
03 S-DATA332 PIC X(4) VALUE 'HGR '.
03 S-DATA333 PIC X(1) VALUE X'00'.
03 S-DATA334 PIC X(1) VALUE X'06'.
03 S-DATA335 PIC X(1) VALUE X'00'.
03 S-DATA336 PIC X(1) VALUE X'04'.
03 S-DATA337 PIC X(2) VALUE 'UR'.
03 S-DATA338 PIC X(1) VALUE X'00'.
03 S-DATA339 PIC X(1) VALUE X'02'.

01 SND4.
02 MSG-SNDSYNC PIC X(8) VALUE 'SENDSYNC'.
02 DATA100 PIC X(5).
02 FILLER PIC X(3).
02 DATA101 PIC X(4) VALUE SPACE.
02 DATA102 PIC X(4) VALUE SPACE.
02 DATA103 PIC 9(8).
02 DATA104 PIC 9(8).
02 DATA105 PIC 9(9) COMP VALUE ZERO.
02 SND-S-SEG PIC X(4) VALUE 'EMI '.
02 DATA106 PIC X(4) VALUE SPACE.
02 DATA107 PIC X(4) VALUE SPACE.
02 DATA108 PIC X(4) VALUE SPACE.
02 DATA109 PIC X(8) VALUE SPACE.
02 DATA110 PIC X(4) VALUE SPACE.
02 DATA111 PIC X(8) VALUE SPACE.
02 DATA112 PIC X(4) VALUE SPACE.
02 DATA113 PIC 9(9) COMP VALUE ZERO.
02 DATA114 PIC 9(9) COMP VALUE ZERO.
02 DATA115 PIC X(1) VALUE SPACE.
02 DATA116 PIC X(1) VALUE SPACE.
02 DATA117 PIC X(14) VALUE LOW-VALUE.

01 SND5.
02 SEG-CODE PIC X(4) VALUE SPACE.
02 TRM-NAME PIC X(8) VALUE 'otm01 '.
02 DATA120 PIC X(8) VALUE SPACE.
02 DATA121 PIC X(8) VALUE SPACE.
02 DATA122 PIC X(28) VALUE LOW-VALUE.

01 SND6.
02 DATA130 PIC 9(9) COMP VALUE 14.
02 DATA131 PIC X(8).
02 SEND-DATA.
03 S-DATA311 PIC 9(4) COMP VALUE 10.
03 S-DATA312 PIC X(4) VALUE 'OTP1'.
03 S-DATA313 PIC X(4) VALUE 'EDP '.
03 S-DATA314 PIC X(1) VALUE X'00'.
03 S-DATA315 PIC X(1) VALUE X'02'.
03 S-DATA316 PIC X(1) VALUE X'00'.
03 S-DATA317 PIC X(1) VALUE X'02'.

01 RCV-S1.
02 MSG-RCVSYNC PIC X(8) VALUE 'RCVSYNC'.

```

付録 E ユーザアプリケーションプログラムのコーディング例

```

02     DATA73     PIC X(5) .
02     FILLER      PIC X(3) .
02     RCV-S-SEG   PIC X(4)   VALUE 'FRST' .
02     DATA74     PIC X(4)   VALUE SPACE .
02     DATA75     PIC 9(8) .
02     DATA76     PIC 9(8) .
02     RCV-S-LEG   PIC 9(9)   COMP  VALUE 256 .
02     DATA77     PIC X(4)   VALUE SPACE .
02     DATA78     PIC X(4)   VALUE SPACE .
02     DATA79     PIC X(4)   VALUE SPACE .
02     DATA80     PIC X(4)   VALUE SPACE .
02     DATA81     PIC X(8)   VALUE SPACE .
02     DATA82     PIC X(4)   VALUE SPACE .
02     DATA83     PIC X(8)   VALUE SPACE .
02     DATA84     PIC X(4)   VALUE SPACE .
02     DATA85     PIC 9(9)   COMP  VALUE ZERO .
02     DATA86     PIC 9(9)   COMP  VALUE ZERO .
02     DATA87     PIC X(1)   VALUE SPACE .
02     DATA88     PIC X(1)   VALUE SPACE .
02     DATA89     PIC X(14)  VALUE LOW-VALUE .
01    RCV-S2 .
02     DATA90     PIC X(4)   VALUE SPACE .
02     TRM-NAME    PIC X(8)   VALUE 'otm01' .
02     DATA92     PIC X(8)   VALUE SPACE .
02     DATA93     PIC X(8)   VALUE SPACE .
02     DATA94     PIC X(28)  VALUE LOW-VALUE .
01    RCV-S3 .
02     DATA95     PIC 9(9)   COMP .
02     DATA96     PIC X(8) .
02     DATA97     PIC X(1024) .

PROCEDURE          DIVISION .
CALL  'CBLDCMCF'   USING  SND-S1  SND-S2  SND-S3 .

CALL  'CBLDCMCF'   USING  RCV-S1  RCV-S2  RCV-S3 .

CALL  'CBLDCMCF'   USING  SND-S4  SND-S5  SND-S6 .

CALL  'CBLDCMCF'   USING  SND-S7  SND-S8  SND-S9 .

CALL  'CBLDCMCF'   USING  RCV-S1  RCV-S2  RCV-S3 .

CALL  'CBLDCMCF'   USING  RCV-S1  RCV-S2  RCV-S3 .

CALL  'CBLDCMCF'   USING  RCV-S1  RCV-S2  RCV-S3 .

CALL  'CBLDCMCF'   USING  SND4    SND5    SND6 .

EXIT  PROGRAM .

```

(2) コンテンション敗者側 (MHP)

```

IDENTIFICATION  DIVISION .
PROGRAM-ID.     RECV01 .

ENVIRONMENT     DIVISION .

```

```

DATA          DIVISION.
WORKING-STORAGE SECTION.

01    RCV1.
      02    MSG-RCV      PIC X(8)    VALUE  'RECEIVE '.
      02    DATA1      PIC X(5) .
      02    FILLER      PIC X(3) .
      02    SEG-CODE    PIC X(4)    VALUE  'FRST'.
      02    DATA2      PIC X(4)    VALUE  SPACE.
      02    DATA3      PIC 9(8) .
      02    DATA4      PIC 9(8) .
      02    SEG-LENG    PIC 9(9)    COMP   VALUE  256.
      02    DATA5      PIC X(4)    VALUE  SPACE.
      02    DATA6      PIC X(4)    VALUE  SPACE.
      02    DATA7      PIC X(4)    VALUE  SPACE.
      02    DATA8      PIC X(4)    VALUE  SPACE.
      02    DATA9      PIC X(8)    VALUE  SPACE.
      02    DATA10     PIC X(4)    VALUE  SPACE.
      02    DATA11     PIC X(8)    VALUE  SPACE.
      02    DATA12     PIC X(4)    VALUE  SPACE.
      02    DATA13     PIC 9(9)    COMP   VALUE  ZERO.
      02    DATA13-1   PIC 9(9)    COMP   VALUE  ZERO.
      02    DATA13-2   PIC X(1)    VALUE  SPACE.
      02    DATA13-3   PIC X(1)    VALUE  SPACE.
      02    DATA14     PIC X(14)   VALUE  LOW-VALUE.

01    RCV2.
      02    DATA15     PIC X(4)    VALUE  SPACE.
      02    RCV-TRM-NAM PIC X(8) .
      02    DATA17     PIC X(8)    VALUE  SPACE.
      02    DATA18     PIC X(6)    VALUE  SPACE.
      02    DATA19     PIC X(2)    VALUE  SPACE.
      02    DATA20     PIC X(28)   VALUE  LOW-VALUE.

01    RCV3.
      02    DATA21     PIC 9(9)    COMP.
      02    DATA22     PIC X(8) .
      02    DATA23     PIC X(1024) .

01    SND1.
      02    MSG-SNDSYNC PIC X(8)    VALUE  'SENDSYNC'.
      02    DATA24     PIC X(5) .
      02    FILLER      PIC X(3) .
      02    DATA25     PIC X(4)    VALUE  SPACE.
      02    DATA26     PIC X(4)    VALUE  SPACE.
      02    DATA27     PIC 9(8) .
      02    DATA28     PIC 9(8) .
      02    DATA29     PIC 9(9)    COMP   VALUE  ZERO.
      02    SND-S-SEG   PIC X(4)    VALUE  'EMI '.
      02    DATA30     PIC X(4)    VALUE  SPACE.
      02    DATA31     PIC X(4)    VALUE  SPACE.
      02    DATA32     PIC X(4)    VALUE  SPACE.
      02    DATA33     PIC X(8)    VALUE  SPACE.
      02    DATA34     PIC X(4)    VALUE  SPACE.
      02    DATA35     PIC X(8)    VALUE  SPACE.
      02    DATA36     PIC X(4)    VALUE  SPACE.
      02    DATA37     PIC 9(9)    COMP   VALUE  ZERO.
      02    DATA38     PIC 9(9)    COMP   VALUE  ZERO.
      02    DATA39     PIC X(1)    VALUE  SPACE.

```

付録 E ユーザアプリケーションプログラムのコーディング例

```

02      DATA40      PIC X(1)  VALUE  SPACE.
02      DATA41      PIC X(14) VALUE  LOW-VALUE.
01  SND2.
02      SEG-CODE     PIC X(4)  VALUE  SPACE.
02      S2-TRM-NAM   PIC X(8) .
02      DATA42      PIC X(8)  VALUE  SPACE.
02      DATA43      PIC X(8)  VALUE  SPACE.
02      DATA44      PIC X(28) VALUE  LOW-VALUE.
01  SND3.
02      DATA45      PIC 9(9)  COMP   VALUE  18.
02      DATA46      PIC X(8) .
02      SEND-DATA.
03      S-DATA1      PIC 9(4)  COMP   VALUE  10.
03      S-DATA2      PIC X(4)  VALUE  'OTP1'.
03      S-DATA3      PIC X(4)  VALUE  'BDPA'.
03      S-DATA4      PIC 9(4)  COMP   VALUE  6.
03      S-DATA5      PIC 9(4)  COMP   VALUE  4.
03      S-DATA6      PIC X(2)  VALUE  'AC'.
03      S-DATA7      PIC 9(4)  COMP   VALUE  2.
01  RCV4.
02      MSG-RCVSYNC  PIC X(8)  VALUE  'RCVSYNC'.
02      DATA48      PIC X(5) .
02      FILLER       PIC X(3) .
02      RCV-S-SEG    PIC X(4)  VALUE  'FRST'.
02      DATA49      PIC X(4)  VALUE  SPACE.
02      DATA50      PIC 9(8) .
02      DATA51      PIC 9(8) .
02      RCV-S-LEG    PIC 9(9)  COMP   VALUE  256.
02      DATA52      PIC X(4)  VALUE  SPACE.
02      DATA53      PIC X(4)  VALUE  SPACE.
02      DATA54      PIC X(4)  VALUE  SPACE.
02      DATA55      PIC X(4)  VALUE  SPACE.
02      DATA56      PIC X(8)  VALUE  SPACE.
02      DATA57      PIC X(4)  VALUE  SPACE.
02      DATA58      PIC X(8)  VALUE  SPACE.
02      DATA59      PIC X(4)  VALUE  SPACE.
02      DATA60      PIC 9(9)  COMP   VALUE  ZERO.
02      DATA61      PIC 9(9)  COMP   VALUE  ZERO.
02      DATA62      PIC X(1)  VALUE  SPACE.
02      DATA63      PIC X(1)  VALUE  SPACE.
02      DATA64      PIC X(14) VALUE  LOW-VALUE.
01  RCV5.
02      DATA65      PIC X(4)  VALUE  SPACE.
02      R5-TRM-NAM   PIC X(8) .
02      DATA67      PIC X(8)  VALUE  SPACE.
02      DATA68      PIC X(8)  VALUE  SPACE.
02      DATA69      PIC X(28) VALUE  LOW-VALUE.
01  RCV6.
02      DATA70      PIC 9(9)  COMP.
02      DATA71      PIC X(8) .
02      DATA72      PIC X(1024).
01  SND4.
02      MSG-SNDSYNC  PIC X(8)  VALUE  'SENDSYNC'.
02      DATA100     PIC X(5) .
02      FILLER       PIC X(3) .
02      DATA101     PIC X(4)  VALUE  SPACE.
02      DATA102     PIC X(4)  VALUE  SPACE.

```

```

02 DATA103 PIC 9(8) .
02 DATA104 PIC 9(8) .
02 DATA105 PIC 9(9) COMP VALUE ZERO .
02 SND-S-SEG PIC X(4) VALUE 'EMI ' .
02 DATA106 PIC X(4) VALUE SPACE .
02 DATA107 PIC X(4) VALUE SPACE .
02 DATA108 PIC X(4) VALUE SPACE .
02 DATA109 PIC X(8) VALUE SPACE .
02 DATA110 PIC X(4) VALUE SPACE .
02 DATA111 PIC X(8) VALUE SPACE .
02 DATA112 PIC X(4) VALUE SPACE .
02 DATA113 PIC 9(9) COMP VALUE ZERO .
02 DATA114 PIC 9(9) COMP VALUE ZERO .
02 DATA115 PIC X(1) VALUE SPACE .
02 DATA116 PIC X(1) VALUE SPACE .
02 DATA117 PIC X(14) VALUE LOW-VALUE .
01 SND5 .
02 SEG-CODE PIC X(4) VALUE SPACE .
02 S5-TRM-NAM PIC X(8) .
02 DATA120 PIC X(8) VALUE SPACE .
02 DATA121 PIC X(8) VALUE SPACE .
02 DATA122 PIC X(28) VALUE LOW-VALUE .
01 SND6 .
02 DATA130 PIC 9(9) COMP VALUE 14 .
02 DATA131 PIC X(8) .
02 SEND-DATA .
03 S-DATA11 PIC 9(4) COMP VALUE 10 .
03 S-DATA12 PIC X(4) VALUE 'OTP1' .
03 S-DATA13 PIC X(4) VALUE 'HGP ' .
03 S-DATA14 PIC 9(4) COMP VALUE 2 .
03 S-DATA15 PIC 9(4) COMP VALUE 2 .
01 SND7 .
02 MSG-SNDSYNC PIC X(8) VALUE 'SENDSYNC' .
02 DATA225 PIC X(5) .
02 FILLER PIC X(3) .
02 DATA226 PIC X(4) VALUE SPACE .
02 DATA227 PIC X(4) VALUE SPACE .
02 DATA228 PIC 9(8) .
02 DATA229 PIC 9(8) .
02 DATA230 PIC 9(9) COMP VALUE ZERO .
02 SND-S-SEG PIC X(4) VALUE 'EMI ' .
02 DATA231 PIC X(4) VALUE SPACE .
02 DATA232 PIC X(4) VALUE SPACE .
02 DATA233 PIC X(4) VALUE SPACE .
02 DATA234 PIC X(8) VALUE SPACE .
02 DATA235 PIC X(4) VALUE SPACE .
02 DATA236 PIC X(8) VALUE SPACE .
02 DATA237 PIC X(4) VALUE SPACE .
02 DATA238 PIC 9(9) COMP VALUE ZERO .
02 DATA239 PIC 9(9) COMP VALUE ZERO .
02 DATA240 PIC X(1) VALUE SPACE .
02 DATA241 PIC X(1) VALUE SPACE .
02 DATA242 PIC X(14) VALUE LOW-VALUE .
01 SND8 .
02 SEG-CODE PIC X(4) VALUE SPACE .
02 S8-TRM-NAM PIC X(8) .
02 DATA243 PIC X(8) VALUE SPACE .

```

付録 E ユーザアプリケーションプログラムのコーディング例

```

02      DATA244      PIC X(8)      VALUE      SPACE.
02      DATA245      PIC X(28)     VALUE      LOW-VALUE.
01      SND9.
02      DATA246      PIC 9(9)      COMP      VALUE      23.
02      DATA247      PIC X(8) .
02      SEND-DATA2.
03      S-DATA281     PIC 9(4)      COMP      VALUE      10.
03      S-DATA282     PIC X(4)      VALUE      'OTP1'.
03      S-DATA283     PIC X(4)      VALUE      'TDR'.
03      S-DATA284     PIC 9(4)      COMP      VALUE      2.
03      S-DATA285     PIC 9(4)      COMP      VALUE      11.
03      S-DATA286     PIC X(1)      VALUE      X'80'.
03      S-DATA287     PIC X(8)      VALUE      'USERDATA'.
01      SND-RCV1.
02      MSG-SNDRCV    PIC X(8)      VALUE      'SENDRECV'.
02      DATA301      PIC X(5) .
02      FILLER        PIC X(3) .
02      DATA302      PIC X(4)      VALUE      SPACE.
02      DATA303      PIC X(4)      VALUE      SPACE.
02      DATA304      PIC 9(8) .
02      DATA305      PIC 9(8) .
02      SND-RCV-LEG   PIC 9(9)      COMP      VALUE      256.
02      SND-RCV-SEG   PIC X(4)      VALUE      'EMI'.
02      DATA306      PIC X(4)      VALUE      SPACE.
02      DATA307      PIC X(4)      VALUE      SPACE.
02      DATA308      PIC X(4)      VALUE      SPACE.
02      DATA309      PIC X(8)      VALUE      SPACE.
02      DATA310      PIC X(4)      VALUE      SPACE.
02      DATA311      PIC X(8)      VALUE      SPACE.
02      DATA312      PIC X(4)      VALUE      SPACE.
02      PRO-CD        PIC 9(9)      COMP      VALUE      ZERO.
02      DATA313      PIC 9(9)      COMP      VALUE      ZERO.
02      DATA313-1    PIC X(1)      VALUE      SPACE.
02      DATA313-2    PIC X(1)      VALUE      SPACE.
02      DATA314      PIC X(14)     VALUE      LOW-VALUE.
01      SND-RCV2.
02      MSG           PIC X(4)      VALUE      'IO'.
02      SR-TRM-NAM    PIC X(8) .
02      DATA315      PIC X(8)      VALUE      SPACE.
02      DATA316      PIC X(8)      VALUE      SPACE.
02      DATA318      PIC X(28)     VALUE      LOW-VALUE.
01      SND-RCV3.
02      MSG-SEG-LEG   PIC 9(9)      COMP      VALUE      18.
02      DATA319      PIC X(8) .
02      SNDRCV-DATA.
03      SR-DATA1      PIC 9(4)      COMP      VALUE      10.
03      SR-DATA2      PIC X(4)      VALUE      'OTPL'.
03      SR-DATA3      PIC X(4)      VALUE      'EDR'.
03      SR-DATA4      PIC 9(4)      COMP      VALUE      6.
03      SR-DATA5      PIC 9(4)      COMP      VALUE      4.
03      SR-DATA6      PIC X(2)      VALUE      'TR'.
03      SR-DATA7      PIC 9(4)      COMP      VALUE      2.
01      SND-RCV4.
02      DATA320      PIC 9(9)      COMP.
02      DATA321      PIC X(7)      VALUE      SPACE.
02      DATA322      PIC X(1) .
02      DATA323      PIC X(1024) .

```

```

PROCEDURE          DIVISION.
CALL  'CBLDCMCF'   USING RCV1      RCV2      RCV3 .
MOVE  RCV-TRM-NAM TO  S2-TRM-NAM .
CALL  'CBLDCMCF'   USING SND1      SND2      SND3 .
MOVE  RCV-TRM-NAM TO  R5-TRM-NAM .
CALL  'CBLDCMCF'   USING RCV4      RCV5      RCV6 .
MOVE  RCV-TRM-NAM TO  R5-TRM-NAM .
CALL  'CBLDCMCF'   USING RCV4      RCV5      RCV6 .
MOVE  RCV-TRM-NAM TO  S5-TRM-NAM .
CALL  'CBLDCMCF'   USING SND4      SND5      SND6 .
MOVE  RCV-TRM-NAM TO  S8-TRM-NAM .
CALL  'CBLDCMCF'   USING SND7      SND8      SND9 .
MOVE  RCV-TRM-NAM TO  SR-TRM-NAM .
CALL  'CBLDCMCF'   USING SND-RCV1  SND-RCV2  SND-RCV3  SND-RCV4 .

EXIT      PROGRAM.
    
```

付録 E.6 ユーザアプリケーションプログラムのコーディング例 (半二重でのデータ操作言語)

(1) コンテンション勝者側

```

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

PROGRAM-ID. SEND01.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*   ワーク変数   *
DATA DIVISION.
WORKING-STORAGE SECTION.
*   ダイアログ開始要求送信領域   *
01  SEND-AREA1.
    02  SE-DATALENG1          PIC 9(4)  COMP VALUE 58.
    02  SE-RSV1              PIC X(2) .
    02  SE-DATA1.
        03  S1-DATA1          PIC X(1)  VALUE X'00' .
        03  S1-DATA2          PIC X(1)  VALUE X'0a' .
        03  S1-DATA3          PIC X(4)  VALUE 'OTP1' .
        03  S1-DATA4          PIC X(4)  VALUE 'BDR' .
        03  S1-DATA5          PIC X(1)  VALUE X'00' .
        03  S1-DATA6          PIC X(1)  VALUE X'2a' .
        03  S1-DATA7          PIC X(1)  VALUE X'00' .
        03  S1-DATA8          PIC X(1)  VALUE X'0d' .
        03  S1-DATA8-1        PIC X(2)  VALUE 'PR' .
        03  S1-DATA9          PIC X(9)  VALUE 'SEND_TPSU' .
        03  S1-DATA10         PIC X(1)  VALUE X'00' .
        03  S1-DATA11         PIC X(1)  VALUE X'02' .
        03  S1-DATA12         PIC X(1)  VALUE X'00' .
    
```

付録E ユーザアプリケーションプログラムのコーディング例

```

03 S1-DATA13          PIC X(1)  VALUE X'0d'.
03 S1-DATA13-1        PIC X(2)  VALUE 'PR'.
03 S1-DATA14          PIC X(9)  VALUE 'RECV_TPSU'.
03 S1-DATA15          PIC X(1)  VALUE X'00'.
03 S1-DATA16          PIC X(1)  VALUE X'04'.
03 S1-DATA17          PIC X(1)  VALUE X'88'.
03 S1-DATA18          PIC X(1)  VALUE X'ff'.
03 S1-DATA19          PIC X(1)  VALUE X'00'.
03 S1-DATA20          PIC X(1)  VALUE X'02'.
03 S1-DATA21          PIC X(1)  VALUE X'00'.
03 S1-DATA22          PIC X(1)  VALUE X'02'.
03 S1-DATA23          PIC X(1)  VALUE X'00'.
03 S1-DATA24          PIC X(1)  VALUE X'04'.
03 S1-DATA25          PIC X(2)  VALUE 'AL'.
03 S1-DATA26          PIC X(1)  VALUE X'00'.
03 S1-DATA27          PIC X(1)  VALUE X'02'.
*   ダイアログ開始応答受信領域   *
01   RECV-AREA1.
    02 RE-DATALENG1      PIC 9(4)  COMP VALUE 1028.
    02 RE-RSV1          PIC X(2).
    02 RE-DATA1         PIC X(1024).
*   データ転送要求送信領域   *
01   SEND-AREA2.
    02 SE-DATALENG2      PIC 9(4)  COMP VALUE 27.
    02 SE-RSV2          PIC X(2).
    02 SE-DATA2.
      03 S2-DATA1        PIC X(1)  VALUE X'00'.
      03 S2-DATA2        PIC X(1)  VALUE X'0a'.
      03 S2-DATA3        PIC X(4)  VALUE 'OTP1'.
      03 S2-DATA4        PIC X(4)  VALUE 'TDR '.
      03 S2-DATA5        PIC X(1)  VALUE X'00'.
      03 S2-DATA6        PIC X(1)  VALUE X'02'.
      03 S2-DATA7        PIC X(1)  VALUE X'00'.
      03 S2-DATA8        PIC X(1)  VALUE X'0b'.
      03 S2-DATA9        PIC X(1)  VALUE X'80'.
      03 S2-DATA10       PIC X(8)  VALUE 'USERDATA'.
*   ハンドシェイク&制御権譲渡要求送信領域   *
01   SEND-AREA3.
    02 SE-DATALENG3      PIC 9(4)  COMP VALUE 22.
    02 SE-RSV3          PIC X(2).
    02 SE-DATA3.
      03 S3-DATA1        PIC X(1)  VALUE X'00'.
      03 S3-DATA2        PIC X(1)  VALUE X'0a'.
      03 S3-DATA3        PIC X(4)  VALUE 'OTP1'.
      03 S3-DATA4        PIC X(4)  VALUE 'HGR '.
      03 S3-DATA5        PIC X(1)  VALUE X'00'.
      03 S3-DATA6        PIC X(1)  VALUE X'06'.
      03 S3-DATA7        PIC X(1)  VALUE X'00'.
      03 S3-DATA8        PIC X(1)  VALUE X'04'.
      03 S3-DATA9        PIC X(2)  VALUE 'UR'.
      03 S3-DATA10       PIC X(1)  VALUE X'00'.
      03 S3-DATA11       PIC X(1)  VALUE X'02'.
*   ハンドシェイク&制御権譲渡確認受信領域   *
01   RECV-AREA3.
    02 RE-DATALENG3      PIC 9(4)  COMP VALUE 1028.
    02 RE-RSV3          PIC X(2).
    02 RE-DATA3         PIC X(1024).

```



```

*   データ転送指示受信領域   *
01  RECV-AREA2.
    02  RE-DATALENG2          PIC  9(4)  COMP  VALUE  1028.
    02  RE-RSV2              PIC  X(2).
    02  RE-DATA2             PIC  X(1024).
*   ダイアログ終了指示受信領域 *
01  RECV-AREA4.
    02  RE-DATALENG4          PIC  9(4)  COMP  VALUE  1028.
    02  RE-RSV4              PIC  X(2).
    02  RE-DATA4             PIC  X(1024).
*   ダイアログ終了応答送信領域 *
01  SEND-AREA4.
    02  SE-DATALENG4          PIC  9(4)  COMP  VALUE  18.
    02  SE-RSV4              PIC  X(2).
    02  SE-DATA4.
        03  S4-DATA1          PIC  X(1)  VALUE  X'00'.
        03  S4-DATA2          PIC  X(1)  VALUE  X'0a'.
        03  S4-DATA3          PIC  X(4)  VALUE  'OTP1'.
        03  S4-DATA4          PIC  X(4)  VALUE  'EDP '.
        03  S4-DATA5          PIC  X(1)  VALUE  X'00'.
        03  S4-DATA6          PIC  X(1)  VALUE  X'02'.
        03  S4-DATA7          PIC  X(1)  VALUE  X'00'.
        03  S4-DATA8          PIC  X(1)  VALUE  X'02'.
*   通信記述項   *
COMMUNICATION SECTION.
*   ダイアログ開始要求の送信   *
CD  SEND-OT1
    FOR I-O
        STATUS KEY IS          SE-STATUS1
        SYMBOLIC TERMINAL IS   SE-TERMNAM1
        SYNCHRONOUS MODE IS    SYNC.
*   ダイアログ開始指示の受信   *
CD  RECV-IN1
    FOR I-O
        STATUS KEY IS          RE-STATUS1
        SYMBOLIC TERMINAL IS   RE-TERMNAM1
        MESSAGE DATE IS        RE-DATE1
        MESSAGE TIME IS        RE-TIME1
        SYNCHRONOUS MODE IS    SYNC
        WAITING TIME IS RE-WAIT1.
*   データ転送要求の送信   *
CD  SEND-OT2
    FOR I-O
        STATUS KEY IS          SE-STATUS2
        SYMBOLIC TERMINAL IS   SE-TERMNAM2
        SYNCHRONOUS MODE IS    SYNC.
*   ハンドシェイク&制御権譲渡要求の送信 *
CD  SEND-OT3
    FOR I-O
        STATUS KEY IS          SE-STATUS3
        SYMBOLIC TERMINAL IS   SE-TERMNAM3
        SYNCHRONOUS MODE IS    SYNC.
*   ハンドシェイク&制御権譲渡確認の受信 *
CD  RECV-IN3
    FOR I-O
        STATUS KEY IS          RE-STATUS3
        SYMBOLIC TERMINAL IS   RE-TERMNAM3

```

```

MESSAGE DATE IS          RE-DATE3
MESSAGE TIME IS          RE-TIME3
SYNCHRONOUS MODE IS SYNC
WAITING TIME IS RE-WAIT3.
*   データ転送指示の受信   *
CD  RECV-IN2
    FOR  I-O
      STATUS KEY IS          RE-STATUS2
      SYMBOLIC TERMINAL IS  RE-TERMNAM2
      MESSAGE DATE IS       RE-DATE2
      MESSAGE TIME IS       RE-TIME2
      SYNCHRONOUS MODE IS SYNC
      WAITING TIME IS RE-WAIT2.
*   ダイアログ終了指示の受信   *
CD  RECV-IN4
    FOR  I-O
      STATUS KEY IS          RE-STATUS4
      SYMBOLIC TERMINAL IS  RE-TERMNAM4
      MESSAGE DATE IS       RE-DATE4
      MESSAGE TIME IS       RE-TIME4
      SYNCHRONOUS MODE IS SYNC
      WAITING TIME IS RE-WAIT4.
*   ダイアログ終了応答の送信   *
CD  SEND-OT4
    FOR  I-O
      STATUS KEY IS          SE-STATUS4
      SYMBOLIC TERMINAL IS  SE-TERMNAM4.
*   通信文   *
PROCEDURE DIVISION.
*   ダイアログ開始要求の送信   *
MOVE 'otm01' TO SE-TERMNAM1.
SEND SEND-OT1
    FROM SEND-AREA1.
*   ダイアログ開始確認の受信   *
MOVE 'otm01' TO RE-TERMNAM1.
MOVE '00010000' TO RE-WAIT1.
RECEIVE RECV-IN1
    SEGMENT
    INTO RECV-AREA1.
*   データ転送要求の送信   *
MOVE 'otm01' TO SE-TERMNAM2.
SEND SEND-OT2
    FROM SEND-AREA2.
*   ハンドシェイク&制御権譲渡要求の送信   *
MOVE 'otm01' TO SE-TERMNAM3.
SEND SEND-OT3
    FROM SEND-AREA3.
*   ハンドシェイク&制御権譲渡確認の受信   *
MOVE 'otm01' TO RE-TERMNAM3.
MOVE '00010000' TO RE-WAIT3.
RECEIVE RECV-IN3
    SEGMENT
    INTO RECV-AREA3.
*   データ転送指示の受信   *
MOVE 'otm01' TO RE-TERMNAM2.
MOVE '00010000' TO RE-WAIT2.
RECEIVE RECV-IN2

```

```

SEGMENT
INTO RECV-AREA2.
*   ダイアログ終了指示の受信   *
    MOVE 'otm01' TO RE-TERMNAM4.
    MOVE '00010000' TO RE-WAIT4.
    RECEIVE RECV-IN4
      SEGMENT
INTO RECV-AREA4.
*   ダイアログ終了応答の送信   *
    MOVE 'otm01' TO SE-TERMNAM4.
    DISABLE SEND-OT4
      WITH SEND-AREA4.
*   終了処理   *
    EXIT PROGRAM.

```

(2) コンテンション敗者側

```

*
*****
*   MHPサーブプログラム   *
*****
*
IDENTIFICATION DIVISION.

PROGRAM-ID. RECV01.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*   ワーク変数   *
DATA DIVISION.
WORKING-STORAGE SECTION.
*   ダイアログ開始指示受信領域   *
01  RECV-AREA1.
    02  RE-DATALENG1      PIC  9(4)  COMP  VALUE  1028.
    02  RE-RSV1          PIC  X(2).
    02  RE-DATA1         PIC  X(1024).
*   ダイアログ開始応答送信領域   *
01  SEND-AREA1.
    02  SE-DATALENG1     PIC  9(4)  COMP  VALUE  22.
    02  SE-RSV1          PIC  X(2).
    03  S1-DATA1         PIC  X(1)  VALUE  X'00'.
    03  S1-DATA2         PIC  X(1)  VALUE  X'0a'.
    03  S1-DATA3         PIC  X(4)  VALUE  'OTP1'.
    03  S1-DATA4         PIC  X(4)  VALUE  'BDPA'.
    03  S1-DATA5         PIC  X(1)  VALUE  X'00'.
    03  S1-DATA6         PIC  X(1)  VALUE  X'06'.
    03  S1-DATA7         PIC  X(1)  VALUE  X'00'.
    03  S1-DATA8         PIC  X(1)  VALUE  X'04'.
    03  S1-DATA9         PIC  X(2)  VALUE  'AC'.
    03  S1-DATA10        PIC  X(1)  VALUE  X'00'.
    03  S1-DATA11        PIC  X(1)  VALUE  X'02'.
*   データ転送指示受信領域   *
01  RECV-AREA2.
    02  RE-DATALENG2     PIC  9(4)  COMP  VALUE  1028.
    02  RE-RSV2          PIC  X(2).

```

付録E ユーザアプリケーションプログラムのコーディング例

```

02 RE-DATA2 PIC X(1024).
* ハンドシェイク&制御権譲渡指示受信領域 *
01 RECV-AREA3.
02 RE-DATALENG3 PIC 9(4) COMP VALUE 1028.
02 RE-RSV3 PIC X(2).
02 RE-DATA3 PIC X(1024).
* ハンドシェイク&制御権譲渡応答送信領域 *
01 SEND-AREA3.
02 SE-DATALENG3 PIC 9(4) COMP VALUE 18.
02 SE-RSV3 PIC X(2).
02 SE-DATA3.
03 S3-DATA1 PIC X(1) VALUE X'00'.
03 S3-DATA2 PIC X(1) VALUE X'0a'.
03 S3-DATA3 PIC X(4) VALUE 'OTP1'.
03 S3-DATA4 PIC X(4) VALUE 'HGP '.
03 S3-DATA5 PIC X(1) VALUE X'00'.
03 S3-DATA6 PIC X(1) VALUE X'02'.
03 S3-DATA7 PIC X(1) VALUE X'00'.
03 S3-DATA8 PIC X(1) VALUE X'02'.
* データ転送要求送信領域 *
01 SEND-AREA2.
02 SE-DATALENG2 PIC 9(4) COMP VALUE 27.
02 SE-RSV2 PIC X(2).
02 SE-DATA2.
03 S2-DATA1 PIC X(1) VALUE X'00'.
03 S2-DATA2 PIC X(1) VALUE X'0a'.
03 S2-DATA3 PIC X(4) VALUE 'OTP1'.
03 S2-DATA4 PIC X(4) VALUE 'TDR '.
03 S2-DATA5 PIC X(1) VALUE X'00'.
03 S2-DATA6 PIC X(1) VALUE X'02'.
03 S2-DATA7 PIC X(1) VALUE X'00'.
03 S2-DATA8 PIC X(1) VALUE X'0b'.
03 S2-DATA9 PIC X(1) VALUE X'80'.
03 S2-DATA10 PIC X(8) VALUE 'USERDATA'.
* ダイアログ終了要求送信領域 *
01 SEND-AREA4.
02 SE-DATALENG4 PIC 9(4) COMP VALUE 22.
02 SE-RSV4 PIC X(2).
02 SE-DATA4.
03 S4-DATA1 PIC X(1) VALUE X'00'.
03 S4-DATA2 PIC X(1) VALUE X'0a'.
03 S4-DATA3 PIC X(4) VALUE 'OTP1'.
03 S4-DATA4 PIC X(4) VALUE 'EDR '.
03 S4-DATA5 PIC X(1) VALUE X'00'.
03 S4-DATA6 PIC X(1) VALUE X'06'.
03 S4-DATA7 PIC X(1) VALUE X'00'.
03 S4-DATA8 PIC X(1) VALUE X'04'.
03 S4-DATA9 PIC X(2) VALUE 'TR'.
03 S4-DATA10 PIC X(1) VALUE X'00'.
03 S4-DATA11 PIC X(1) VALUE X'02'.
* ダイアログ終了確認受信領域 *
01 RECV-AREA4.
02 RE-DATALENG4 PIC 9(4) COMP VALUE 1028.
02 RE-RSV4 PIC X(2).
02 RE-DATA4 PIC X(1024).
* 通信記述項 *
COMMUNICATION SECTION.

```

```

*   ダイアログ開始要求の受信   *
    CD  RECV-IN1
        FOR  INPUT
            STATUS KEY IS          RE-STATUS1
            SYMBOLIC TERMINAL IS    RE-TERMNAM1
            MESSAGE DATE IS         RE-DATE1
            MESSAGE TIME IS         RE-TIME1.
*   ダイアログ開始応答の送信   *
    CD  SEND-OT1
        FOR  I-O
            STATUS KEY IS          SE-STATUS1
            SYMBOLIC TERMINAL IS    SE-TERMNAM1
            SYNCHRONOUS MODE IS     SYNC.
*   データ転送指示の受信       *
    CD  RECV-IN2
        FOR  I-O
            STATUS KEY IS          RE-STATUS2
            SYMBOLIC TERMINAL IS    RE-TERMNAM2
            MESSAGE DATE IS         RE-DATE2
            MESSAGE TIME IS         RE-TIME2
            SYNCHRONOUS MODE IS     SYNC
            WAITING TIME IS         RE-WAIT2.
*   ハンドシェイク&制御権譲渡指示の受信   *
    CD  RECV-IN3
        FOR  I-O
            STATUS KEY IS          RE-STATUS3
            SYMBOLIC TERMINAL IS    RE-TERMNAM3
            MESSAGE DATE IS         RE-DATE3
            MESSAGE TIME IS         RE-TIME3
            SYNCHRONOUS MODE IS     SYNC
            WAITING TIME IS         RE-WAIT3.
*   ハンドシェイク&制御権譲渡応答の送信   *
    CD  SEND-OT3
        FOR  I-O
            STATUS KEY IS          SE-STATUS3
            SYMBOLIC TERMINAL IS    SE-TERMNAM3
            SYNCHRONOUS MODE IS     SYNC.
*   データ転送要求の送信       *
    CD  SEND-OT2
        FOR  I-O
            STATUS KEY IS          SE-STATUS2
            SYMBOLIC TERMINAL IS    SE-TERMNAM2
            SYNCHRONOUS MODE IS     SYNC.
*   ダイアログ終了要求の送信   *
    CD  SEND-OT4
        FOR  I-O
            STATUS KEY IS          SE-STATUS4
            SYMBOLIC TERMINAL IS    SE-TERMNAM4.
*   ダイアログ終了確認の受信   *
    CD  RECV-IN4
        FOR  I-O
            STATUS KEY IS          RE-STATUS4
            SYMBOLIC TERMINAL IS    RE-TERMNAM4
            MESSAGE DATE IS         RE-DATE4
            MESSAGE TIME IS         RE-TIME4
            SYNCHRONOUS MODE IS     SYNC
            WAITING TIME IS         RE-WAIT4.

```

```

*   通信文   *
PROCEDURE DIVISION.
*   ダイアログ開始指示の受信   *
    RECEIVE  RECV-IN1
           FIRST SEGMENT
           INTO  RECV-AREA1.
*   ダイアログ開始応答の送信   *
    MOVE RE-TERMNAM1 TO SE-TERMNAM1.
    SEND  SEND-OT1
           FROM  SEND-AREA1.
*   データ転送指示の受信   *
    MOVE RE-TERMNAM1 TO RE-TERMNAM2.
    MOVE '00010000' TO RE-WAIT2.
    RECEIVE  RECV-IN2
           SEGMENT
           INTO  RECV-AREA2.
*   ハンドシェイク&制御権譲渡指示の受信   *
    MOVE RE-TERMNAM1 TO RE-TERMNAM3.
    MOVE '00010000' TO RE-WAIT3.
    RECEIVE  RECV-IN3
           SEGMENT
           INTO  RECV-AREA3.
*   ハンドシェイク&制御権譲渡応答の送信   *
    MOVE RE-TERMNAM1 TO SE-TERMNAM3.
    SEND  SEND-OT3
           FROM  SEND-AREA3.
*   データ転送要求の送信   *
    MOVE RE-TERMNAM1 TO SE-TERMNAM2.
    SEND  SEND-OT2
           FROM  SEND-AREA2.
*   ダイアログ終了要求の送信   *
    MOVE RE-TERMNAM1 TO SE-TERMNAM4.
    DISABLE SEND-OT4
           WITH SEND-AREA4.
*   ダイアログ終了確認の受信   *
    MOVE RE-TERMNAM1 TO RE-TERMNAM4.
    MOVE '00010000' TO RE-WAIT4.
    RECEIVE  RECV-IN4
           SEGMENT
           INTO  RECV-AREA4.
*   終了処理   *
    EXIT PROGRAM.

```

付録 F 旧製品からの移行に関する注意事項

旧製品からの移行に関する注意事項について説明します。

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

表 F-1 バージョン 6 以前で使用していたソースファイルの互換性

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

付録 G バージョンアップ時の変更点

各バージョンでの変更点を次に示す分類ごとに示します。

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

付録 G.1 07-00 での変更点

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

TP1/NET/OSI-TP 07-00 での動作の変更点を次に示します。

表 G-1 TP1/NET/OSI-TP 07-00 での動作の変更点

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

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

付録 H 用語解説

(英字)

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。論理端末名称、および入力メッセージの情報で構成されます。

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

MJ (メッセージジャーナル)

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。

OSI TP プロトコル

OSI 参照モデルの階層のうち、第 7 層 (アプリケーション層) について規定したプロトコルです。分散トランザクション処理について規定しています。

(カ行)

経路交代

コネクション確立要求でネットワーク層より以下の層に障害が発生した場合、経路を切り替えてコネクションを確立できます。この機能が経路交代です。MCF 通信構成定義で現用相手 DTE アドレスと現用スロット番号のほかに、交代用相手 DTE アドレスと交代用スロット番号の指定が必要です。

コネクション

TP1/NET/OSI-TP 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。

コンテンション (競合)

ダイアログ開始時に、両方のシステムから同時に開始要求が出されている状態です。コンテンション発生時は、定義によって優先権を持っているシステムが、開始要求を送信できます。

(サ行)

サービスプリミティブ

OSI 参照モデルで、それぞれの階層が利用するサービスです。TP1/NET/OSI-TP では、プレゼンテーション層が提供するサービスプリミティブを利用して、メッセージの送受信をします。

制御権

半二重の場合に使用します。半二重ではどちらか一方からしか送信できません。どちらが送信するかを明確にするために設けたもので、これを持っている方だけが送信できます。

全二重

送信動作と受信動作は互いに独立していて、同時に送信動作と受信動作ができます。

(夕行)

ダイアログ

TP1/NET/OSI-TP が OSI TP プロトコルに準拠して提供する、メッセージ送受信のサービスです。

(八行)

ハンドシェイク

メッセージの送受信時に、システム間で同期を取って処理をするための機能です。

半二重

送信動作と受信動作は独立していません。このため送信動作をするときは受信動作ができないし、受信動作をするときは送信動作ができません。

(ラ行)

論理端末

TP1/NET/OSI-TP と UAP との通信接点です。TP1/NET/OSI-TP と UAP は、論理端末単位にメッセージの送受信をします。

索引

A

altn_dtea 165
altn_slot 166
APDU 連結 47
AP 間通信 2
AP 間通信の概要 2
AP 間通信の形態 4
AP 間通信の仕組み 14
AP 間通信メッセージの送受信 28

B

bretry 164
bretryent 165
bretryint 165

C

CBLDCMCF('RECEIVE') 72
CBLDCMCF('RECVSYNC') 77
CBLDCMCF('SENDRECV') 82
CBLDCMCF('SENDSYNC') 89
CCLSEVT 144, 152
CERREVT 144, 151
CERREVT の理由コード一覧 232
cnflevt 167
COBOL 言語のメッセージ送受信 51
control 167
COPNEVT 144, 152
count 163
C 言語のメッセージ送受信 51

D

dc_mcf_receive 53
dc_mcf_recvsync 57
dc_mcf_sendrecv 61
dc_mcf_sendsync 67
DISABLE 95
dtea 165

E

ENABLE 98
ERREVT1 141, 145
ERREVT2 142, 146
ERREVT2 の理由コード一覧 232
ERREVT3 143, 148
ERREVT4 143, 150

G

GJ (メッセージ受信ジャーナル) 291

H

handshake 167

I

IJ (メッセージ入力ジャーナル) 291

M

max_open_fds 176
max_socket_descriptors 175
MCF 10
mcfmuap 160
mftactcn 191
mftalcen 161
mftalced 170
mftalcle 171
mftbuf 163, 167
mftdetcn 194
mftgrpcn 172
mftgrped 173
mftlscn 197
MCF アプリケーション定義 154
MCF イベント 139
MCF イベントインタフェース 139
MCF イベント処理用 MHP 139, 140
MCF イベント通知時のセグメント構成 140
MCF イベントの共通ヘッダ 141
MCF イベントの種類 139

MCF サービス名の登録 202
 MCF 通信構成定義 154
 MCF 通信プロセスでアクセスするファイルの最大数 176
 MCF 定義オブジェクトの生成 177
 MCF で使用する定義ファイル 154
 MCF マネージャ定義 154
 MCF メイン関数の作成 203
 MJ (メッセージジャーナル) 291
 module 174
 msgbuf 163

N

nomltim 164

O

OSAS/TP/DCCM3 11
 OSAS/TP/DCCM3 の定義との関係 182
 OSI TP プロトコル 291

P

PSAP 21

R

revbuf 163
 RECEIVE 101
 recvtim 160

S

SEND 105
 slot 165
 sndbuf 163
 sndrcvtim 160
 sndtim 160

T

TL クラス 166
 TP1/Message Control 10
 TP1/NET/OSI-TP からのコネクションの解放 17

TP1/NET/OSI-TP からのコネクションの強制解放 19
 TP1/NET/OSI-TP の OSI7 層構造の中での機能範囲 3
 TP1/NET/OSI-TP の TP プロトコルデータ単位 14
 TP1/NET/OSI-TP の運用コマンド 190
 TP1/NET/OSI-TP のシステム構成例 184
 TP1/NET/OSI-TP の実行形式プログラム名 174
 TP1/NET/OSI-TP の実装範囲 222
 TP1/NET/OSI-TP を組み込んだソフトウェア構成の例 11
 TP1/NET/OSI-TP を使用したネットワーク構成の例 2

U

UAP 異常終了通知イベント 139
 UAP 共通定義 160
 UAP でのエラー発生時の処理 (制御権がある場合) 44
 UAP でのエラー発生時の処理 (制御権がない場合) 46
 UAP でのエラー発生時の処理 (全二重の場合) 43
 UAP 閉塞による障害 219
 UOC 作成上の注意事項 137
 UOC で使用できる関数 137
 UOC の異常処理 138
 UOC の構造 137
 UOC の実行タイミング 138

あ

相手システムからのエラー通知受信の処理 (制御権がある場合) 46
 相手システムからのエラー通知受信の処理 (制御権がない場合) 45
 相手システムからのエラー通知受信の処理 (全二重の場合) 43
 相手システムからのコネクションの解放 18
 相手システムの異常による解放 19

相手システムの通信定義と関連づける内容
181
 アソシエーションプール管理機能 26
 アソシエーションプール管理機能の概要 27
 アプリケーション名 171
 アプリケーション名の決定 38, 126

い

一方受信 5
 一方送信メッセージの受信 (COBOL 言語)
72
 一方送信メッセージの受信 (C 言語) 53

う

運用コマンド 189

え

エラー通知 42

お

応答を不要とした場合のメッセージ破棄の例
37
 応用コンテキスト名 166

か

概要 1
 確認型 34

き

機能 13
 基本サービスプリミティブ 110
 旧製品からの移行に関する注意事項 289
 拒否応答型 34

く

組み込み方法 201

け

経路交代 23, 291

経路交代用のスロット番号 166
 経路交代用の通信相手システムの DTE アド
レス 166
 現用スロット番号 165
 現用の相手システムの DTE アドレス 165

こ

コネクション 14, 291
 コネクション ID 162
 コネクション確立再試行回数 165
 コネクション確立再試行間隔 165
 コネクショングループ定義の開始 172
 コネクショングループ定義の終了 173
 コネクショングループ名 172
 コネクション障害 216
 コネクション定義の開始 161
 コネクション定義の終了 170
 コネクションと PSAP の関係 21
 コネクションと論理端末の関係 20
 コネクションの解放 194
 コネクションの確立 14, 191
 コネクションの確立 (着呼型) 16
 コネクションの確立 (発呼型) 15
 コネクションの確立再試行 164
 コネクションの確立と解放 14
 コネクションの確立方法 164
 コネクションの起動, または受信の種別を指
定 165
 コネクションの強制解放 18
 コネクションの状態表示 197
 コネクションの正常解放 16
 コネクションの切断 20
 コンテンション 34
 コンテンション (競合) 291
 コンテンションの勝者または敗者の種別 166

さ

サービス詳細 223
 サービスプリミティブ 110, 222, 292
 サービスプリミティブと UAP との対応 111
 サービスプリミティブとパラメタとの対応
116

サービスプリミティブの形式 115
 サービスプリミティブの突き合わせ項目（コ
 ネクション確立時） 121
 サービスプリミティブの突き合わせ項目（ダ
 イアログ開始時） 122
 サービスプリミティブの連結パターンと
 UAP との対応 113

し

自システムの PSAP アドレス 162
 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 178
 システムサービス共通情報定義 175
 システムサービス情報定義 174
 システム定義 153
 実装範囲 222
 出力メッセージの編集 133
 出力メッセージの編集 UOC 133
 出力メッセージ編集 UOC インタフェース
 134
 障害対策 209
 障害通知イベント 140
 障害の種類と対応処理 210
 障害発生時の処理の流れ 230
 状態通知イベント 140
 処理層とデータの関係(単一 APDU の場合)
 25
 処理層とデータの関係(連結 APDU の場合)
 26

せ

制御権 32, 292
 制御権委譲付きハンドシェイク 41
 制御権が移動する契機 33
 制御権の委譲 32
 制御権の要求 33
 制御情報の形式 115
 全二重 292
 全二重機能 167
 全二重と半二重 24

そ

送受信メッセージの形式 114
 ソケット用ファイル記述子の最大数 175
 ソフトウェアの構成 10

た

ダイアログ 34, 292
 ダイアログ開始方向 37
 ダイアログ開始要求受信時の MHP 異常終了
 218
 ダイアログ障害 217
 ダイアログ中の UAP（MHP，SPP）終了
 218
 ダイアログと UAP 37
 ダイアログの異常終了 36
 ダイアログの開始 34
 ダイアログの開始と終了 36
 ダイアログの終了 34
 タイマ起動メッセージ廃棄通知イベント 139
 端末タイプ 171

ち

着呼型 15

つ

通信相手システムの PSAP アドレス 163
 通信相手システムの種別 164
 通信形態 4

て

定義オブジェクトファイルの生成 206
 定義の指定順序 159
 定義例 183
 データ操作言語（COBOL 言語）のメッセー
 ジ送受信 52
 データ転送 42
 適用範囲 6

と

同期型 29

同期型のメッセージの受信 29
 同期型のメッセージの送受信 29, 30
 同期型のメッセージの送信 29
 同期型のメッセージの送信, 受信, および送
 受信 31
 同期型メッセージの受信 (COBOL 言語) 77
 同期型メッセージの受信 (C 言語) 57
 同期型メッセージの送受信 (COBOL 言語)
 82
 同期型メッセージの送受信 (C 言語) 61
 同期型メッセージの送受信 (データ操作言
 語) 105
 同期型メッセージの送信 (COBOL 言語) 89
 同期型メッセージの送信 (C 言語) 67
 同期型メッセージの送信 (データ操作言語)
 95, 98
 同期受信 5
 同期受信監視時間 160
 同期送受信 6
 同期送受信監視時間 160
 同期送信 5
 同期送信監視時間 160

に

入力メッセージの編集 126
 入力メッセージの編集とアプリケーション名
 の決定 126
 入力メッセージ編集 UOC 126
 入力メッセージ編集 UOC インタフェース
 127

ね

ネットワーク構成の例 181

は

発呼型 14
 バッファ形式 1 26
 バッファ形式 2 26
 パラメタの形式 115
 パラメタの詳細 118
 ハンドシェイク 40, 292
 ハンドシェイクを使用 167

半二重 292
 半二重機能 167

ひ

非同期型 28
 非同期型のメッセージの受信 28

ふ

不正アプリケーション名検出通知イベント
 139
 プロトコル監視時間 164
 プロトコルの種別 162

み

未処理送信メッセージ廃棄通知イベント 139

め

メッセージ受信用バッファグループ番号 163
 メッセージ制御機能 10
 メッセージ送受信インタフェース 49
 メッセージ送受信インタフェースの一覧 51
 メッセージ送受信時のコネクション障害 216
 メッセージ送受信の関数 (C 言語) 51
 メッセージ送受信の関数に対応するプログラ
 ム (COBOL 言語) 51
 メッセージ送受信の処理の流れ 226
 メッセージ送受信の通信文 (データ操作言
 語) 52
 メッセージ送信用バッファグループ番号 163
 メッセージの応答の要否 8, 37
 メッセージの応答を必要とした場合 9
 メッセージの応答を不要とした場合 8
 メッセージの形式 114
 メッセージの形式の例 8
 メッセージの受信 29
 メッセージの受信 (データ操作言語) 101
 メッセージの送達確認の例 7
 メッセージの送達の確認 6
 メッセージの重複, および欠落のチェック 7
 メッセージの分割と組み立て 25
 メッセージ廃棄通知イベント 139

メッセージ編集用バッファグループ番号 163

メッセージ編集用バッファ数 163

ゆ

ユーザ ASE 抽象構文名 166

ユーザアプリケーションプログラム異常終了
218

ユーザアプリケーションプログラム作成例
123

ユーザアプリケーションプログラムのコー
ディング例 234

ユーザアプリケーションプログラムのコー
ディング例 (全二重での C 言語) 234

ユーザアプリケーションプログラムのコー
ディング例 (全二重でのデータ操作言語)
253

ユーザアプリケーションプログラムのコー
ディング例 (半二重での COBOL 言語) 272

ユーザアプリケーションプログラムのコー
ディング例 (半二重でのデータ操作言語)
281

ユーザアプリケーションプログラム閉塞 219

ユーザOWNコーディング 126

ユーザOWNコーディングインタフェース
126

ユーザ情報の形式 116

ユーザ情報の操作方法 138

り

理由コード一覧 232

ろ

論理端末 20, 292

論理端末定義 171

論理端末と UAP の関係 24

論理端末とアプリケーションの型の関係 23

論理端末の端末タイプ, 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 23

論理端末名称 171

ソフトウェアマニュアルのサービス ご案内

1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

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

ホームページのメニューは次のとおりです。

マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

2. インターネットでのマニュアル公開

2種類のマニュアル公開サービスを実施しています。

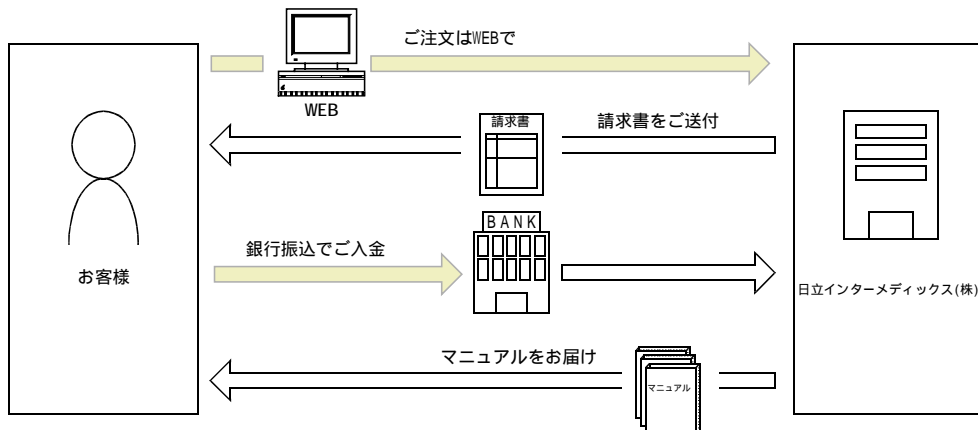
(1) マニュアル情報ホームページ「オンラインマニュアル」での公開

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

(2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開

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

3. マニュアルのご注文



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

ご注文いただいたマニュアルについて請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。

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