

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-D73-10

---

## 前書き

### ■ 対象製品

・適用 OS : AIX V6.1, AIX V7.1

P-1M64-3141 uCosminexus TP1/Message Control 07-51

P-1M64-3241 uCosminexus TP1/NET/Library 07-51

P-F1M64-32414 uCosminexus TP1/NET/OSI-TP 07-50

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。

### ■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、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

HITACHI, DCCM, OpenTP1, OSAS, uCosminexus は、株式会社 日立製作所の商標または登録商標です。

IBM, AIX は、世界の多くの国で登録された International Business Machines Corporation の商標です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

その他記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

### ■ 発行

2017 年 3 月 3000-3-D73-10

### ■ 著作権

All Rights Reserved. Copyright (C) 2009, 2017, Hitachi, Ltd.

## 変更内容

### 変更内容 (3000-3-D73-10) uCosminexus TP1/Message Control 07-51, uCosminexus TP1/NET/Library 07-51, uCosminexus TP1/NET/OSI-TP 07-50

追加・変更内容	変更箇所
<p>コネクションの確立, 解放, および状態取得について, ライブラリ関数および COBOL-UAP 作成用プログラムインタフェースを使用する場合の説明を追加した。</p>	<p>2.1.1, 3. C 言語のライブラリ関数 C 言語のライブラリ関数の一覧, dc_mcf_tactcn, dc_mcf_tdctcn, dc_mcf_tlscn, 4. COBOL-UAP 作成用プログラムインタフェース COBOL-UAP 作成用プログラムインタフェースの一覧, CBLDCMCF('TACTCN '), CBLDCMCF('TDCTCN '), CBLDCMCF('TLSCN ')</p>
<p>NULL またはヌル文字列設定時のコーディング例を追加した。</p>	<p>3. C 言語のライブラリ関数 C 言語のライブラリ関数の一覧</p>
<p>リターン値 DCMCFRTN_73001 およびステータスコード 73001 の説明を変更した。</p>	<p>3. C 言語のライブラリ関数 dc_mcf_recvsync, dc_mcf_sendrecv, dc_mcf_sendsync, 4. COBOL-UAP 作成用プログラムインタフェース CBLDCMCF('RECVSYNC'), CBLDCMCF('SENDRECV'), CBLDCMCF('SENDSYNC'), DISABLE - 同期型メッセージの送信, ENABLE - 同期型メッセージの送信, RECEIVE - メッセージの受信, SEND - メッセージの送信</p>
<p>データ操作言語の通信文と C 言語のライブラリ関数の対応の説明を追加した。</p>	<p>4. COBOL-UAP 作成用プログラムインタフェース COBOL-UAP 作成用プログラムインタフェースの一覧</p>

追加・変更内容	変更箇所
ERREVT3 に設定するトランザクションブランチ ID の形式の説明を追加した。	5.2.3(4)(b), 5.2.4
MCF 性能検証用トレースの説明を追加した。	6. システム定義 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, 付録 G
システムサービス共通情報定義の, 次に示すオペランドの指定値の上限を拡張した。 <ul style="list-style-type: none"> <li>• max_socket_descriptors</li> <li>• max_open_fds</li> </ul>	6. システム定義 システムサービス共通情報定義
定義オブジェクトファイルの出力先に読み取り権限を持つファイルがすでにある場合, 定義オブジェクトファイルを上書きするかどうかを指定できるようにした。	6. システム定義 MCF 定義オブジェクトの生成
MCF 定義オブジェクトの解析コマンド (mcfostpr) を追加した。	6. システム定義 MCF 定義オブジェクトの解析
OpenTP1 システムを変更する場合に, 見直しが必要な定義と, 変更手順について説明を追加した。	6. システム定義 OpenTP1 システムの変更に影響する定義
スタート関数を呼び出したあとの注意事項を追加した。	8.2
次に示すバージョンの変更点を記載した。 <ul style="list-style-type: none"> <li>• TP1/NET/OSI-TP 07-50</li> </ul>	付録 A
バージョン 6 以前のインタフェースの変更一覧を追加した。	付録 C

単なる誤字・脱字などはお断りなく訂正しました。

## 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。

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 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 のコネクション、およびメッセージ送受信に関する機能について説明しています。

#### 第 3 章 C 言語のライブラリ関数

TP1/NET/OSI-TP で使用できる、C 言語のライブラリ関数について説明しています。

#### 第 4 章 COBOL-UAP 作成用プログラムインタフェース

TP1/NET/OSI-TP で使用できる、COBOL-UAP 作成用プログラムインタフェースについて説明しています。

## 第 5 章 ユーザOWNコーディング，MCF イベントインタフェース

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。

## 第 6 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，および定義例について説明しています。

## 第 7 章 運用コマンド

TP1/NET/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 8 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 9 章 障害対策

TP1/NET/OSI-TP 運用中に発生する障害と，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 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 B 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 へ移行する場合の注意事項について説明しています。

## 付録 C インタフェースの変更一覧（バージョン 6 以前から移行する場合）

バージョン 6 以前からバージョン 7 に移行する場合のインタフェースの変更一覧について説明しています。

## 付録 D 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 E 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 F 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 G MCF 性能検証用トレースの取得

MCF 性能検証用トレースの取得について説明しています。

#### 付録 H ユーザアプリケーションプログラムの作成例

TP1/NET/OSI-TP のユーザアプリケーションプログラムの作成例について説明しています。

#### 付録 I 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 J このマニュアルの参考情報

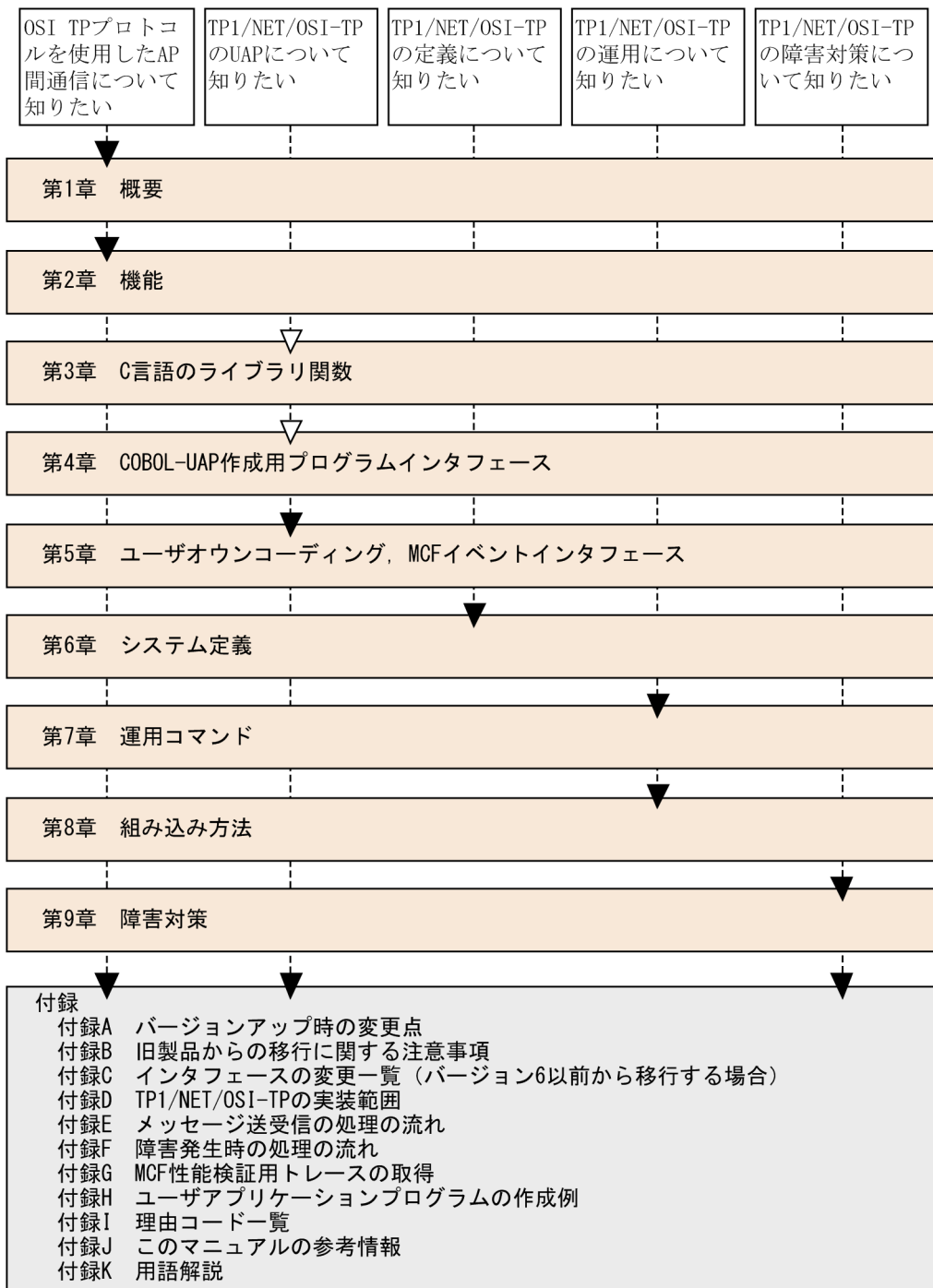
関連マニュアル，このマニュアルで使用している略語の意味などを説明しています。

#### 付録 K 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。

## ■ 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

▽ : 必要に応じて読む項目

## ■ 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



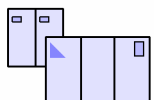
●論理回線



●入出力の動作



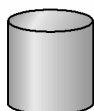
●ホストコンピュータ



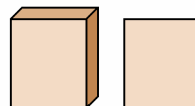
●プログラムの流れ



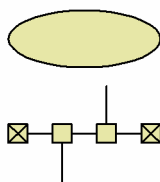
●ファイル



●プログラム



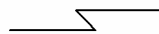
●ネットワーク



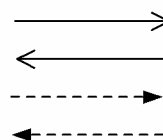
●データの流れ



●通信回線



●制御の流れ



●障害



## ■ 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。

文法記述記号	意味
{ }	(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   mcftalcle   mcftalced}} mcftalccn, mcftalcle および mcftalced 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。

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)

構文要素記号	意味
<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)

## ■ 謝辞

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

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

COBOL は産業界の言語であり, 特定の団体や組織の所有物ではない。

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

次に示す著作権表示付資料の著作者及び著作権者

FLOW-MATIC (Sperry Rand Corporation の商標),

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

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

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

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

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

# 目次

前書き	2
変更内容	3
はじめに	5

<b>1</b>	<b>概要</b>	<b>17</b>
1.1	AP 間通信の概要	18
1.2	AP 間通信の形態	20
1.2.1	通信形態	20
1.2.2	適用範囲	21
1.3	ソフトウェア構成の例	25
<b>2</b>	<b>機能</b>	<b>26</b>
2.1	AP 間通信の仕組み	27
2.1.1	コネクションの確立と解放	27
2.1.2	コネクションと論理端末の関係	35
2.1.3	コネクションと PSAP の関係	35
2.1.4	論理端末とアプリケーションの型の関係	37
2.1.5	論理端末と UAP の関係	37
2.1.6	全二重と半二重	38
2.1.7	メッセージの分割と組み立て	38
2.1.8	アソシエーションプール管理機能	40
2.1.9	サービスプリミティブ	41
2.2	AP 間通信メッセージの送受信	53
2.2.1	非同期型のメッセージの受信	53
2.2.2	同期型のメッセージの送受信	54
2.2.3	制御権	57
2.2.4	ダイアログ	58
2.2.5	アプリケーション名の決定	61
2.2.6	ハンドシェイク	63
2.2.7	データ転送	65
2.2.8	エラー通知	66
2.2.9	APDU 連結	70
<b>3</b>	<b>C 言語のライブラリ関数</b>	<b>71</b>
	C 言語のライブラリ関数の一覧	72

dc_mcf_receive	一方送信メッセージの受信 (C 言語)	73
dc_mcf_recvsync	同期型メッセージの受信 (C 言語)	77
dc_mcf_sendrecv	同期型メッセージの送受信 (C 言語)	82
dc_mcf_sendsync	同期型メッセージの送信 (C 言語)	89
dc_mcf_tactcn	コネクションの確立 (C 言語)	95
dc_mcf_tdctcn	コネクションの解放 (C 言語)	99
dc_mcf_tlscn	コネクションの状態取得 (C 言語)	103

## 4 COBOL-UAP 作成用プログラムインタフェース 108

COBOL-UAP 作成用プログラムインタフェースの一覧		109
CBLDCMCF('RECEIVE')	一方送信メッセージの受信 (COBOL 言語)	112
CBLDCMCF('RECVSYNC')	同期型メッセージの受信 (COBOL 言語)	117
CBLDCMCF('SENDRECV')	同期型メッセージの送受信 (COBOL 言語)	122
CBLDCMCF('SENDSYNC')	同期型メッセージの送信 (COBOL 言語)	129
CBLDCMCF('TACTCN')	コネクションの確立 (COBOL 言語)	135
CBLDCMCF('TDCTCN')	コネクションの解放 (COBOL 言語)	138
CBLDCMCF('TLSCN')	コネクションの状態取得 (COBOL 言語)	142
DISABLE	同期型メッセージの送信 (データ操作言語)	146
ENABLE	同期型メッセージの送信 (データ操作言語)	149
RECEIVE	メッセージの受信 (データ操作言語)	153
SEND	メッセージの送信 (データ操作言語)	158

## 5 ユーザOWNコーディング, MCF イベントインタフェース 164

5.1	ユーザOWNコーディングインタフェース	165
5.1.1	入力メッセージの編集とアプリケーション名の決定	165
5.1.2	入力メッセージ編集 UOC インタフェース	166
5.1.3	出力メッセージの編集	172
5.1.4	出力メッセージ編集 UOC インタフェース	173
5.1.5	UOC 作成上の注意事項	176
5.2	MCF イベントインタフェース	178
5.2.1	MCF イベントの種類	178
5.2.2	MCF イベント通知時のセグメント構成	179
5.2.3	MCF イベント情報の形式 (C 言語)	180
5.2.4	MCF イベント情報の形式 (COBOL 言語)	184

## 6 システム定義 192

TP1/NET/OSI-TP の定義の概要		193
TP1/NET/OSI-TP 固有のシステム定義の種類		195
mcfmuap	(UAP 共通定義)	198
mcftalccn	(コネクション定義の開始)	200
mcftalced	(コネクション定義の終了)	207
mcftalcle	(論理端末定義)	208
mcftgrpcn	(コネクショングループ定義の開始)	209

mcftgrpdc (コネクショングループ定義の終了)	210
システムサービス情報定義	211
システムサービス共通情報定義	213
MCF 定義オブジェクトの生成	216
MCF 定義オブジェクトの解析	217
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	219
相手システムの通信定義と関連づける内容	222
OpenTP1 システムの変更に影響する定義	225
定義例	227

## 7 運用コマンド 231

TP1/NET/OSI-TP の運用コマンド	232
mcftactcn (コネクションの確立)	233
mcftdctcn (コネクションの解放)	236
mcftlscn (コネクションの状態表示)	239

## 8 組み込み方法 243

8.1 TP1/NET/OSI-TP の組み込みの流れ	244
8.1.1 MCF メイン関数の作成	244
8.1.2 MCF サービス名の登録	244
8.1.3 システムサービス情報定義ファイルの作成	244
8.1.4 定義オブジェクトファイルの生成	244
8.2 MCF メイン関数の作成	245
8.3 定義オブジェクトファイルの生成	248

## 9 障害対策 251

9.1 障害の種類と対応処理	252
9.1.1 コネクション障害	252
9.1.2 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)	253
9.1.3 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)	255
9.1.4 UAP 障害	256
9.1.5 ジャーナル障害 (IJ, MJ)	257
9.1.6 関数のリターン処理障害	257
9.1.7 プロシジャ障害	257
9.2 コネクション障害	258
9.3 ダイアログ障害	259
9.4 ユーザアプリケーションプログラム異常終了	260
9.4.1 ダイアログ開始要求受信時の MHP 異常終了	260
9.4.2 ダイアログ中の UAP (MHP, SPP) 終了	260
9.5 ユーザアプリケーションプログラム閉塞	261

## 付録 262

付録 A	バージョンアップ時の変更点	263
付録 A.1	07-50 での変更点	263
付録 A.2	07-00 での変更点	264
付録 B	旧製品からの移行に関する注意事項	265
付録 B.1	ソースの互換性	265
付録 C	インタフェースの変更一覧 (バージョン 6 以前から移行する場合)	266
付録 C.1	メッセージ送受信インタフェース	266
付録 C.2	ユーザOWNコーディング	269
付録 C.3	MCF イベントインタフェース	272
付録 C.4	MCF メイン関数のコーディング概要	273
付録 D	TP1/NET/OSI-TP の実装範囲	275
付録 D.1	実装範囲	275
付録 D.2	サービスプリミティブ	275
付録 D.3	サービス詳細	276
付録 E	メッセージ送受信の処理の流れ	280
付録 F	障害発生時の処理の流れ	282
付録 G	MCF 性能検証用トレースの取得	283
付録 G.1	MCF 固有情報の出力情報	283
付録 G.2	MCF 性能検証用トレースの取得タイミング	283
付録 G.3	MCF 性能検証用トレースの取得量	285
付録 H	ユーザアプリケーションプログラムの作成例	287
付録 H.1	コーディング例	288
付録 H.2	提供するサンプルコーディング	342
付録 I	理由コード一覧	345
付録 J	このマニュアルの参考情報	348
付録 J.1	関連マニュアル	348
付録 J.2	このマニュアルでの表記	349
付録 J.3	英略語	349
付録 J.4	KB (キロバイト) などの単位表記について	350
付録 K	用語解説	351

## 索引 353

# 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

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

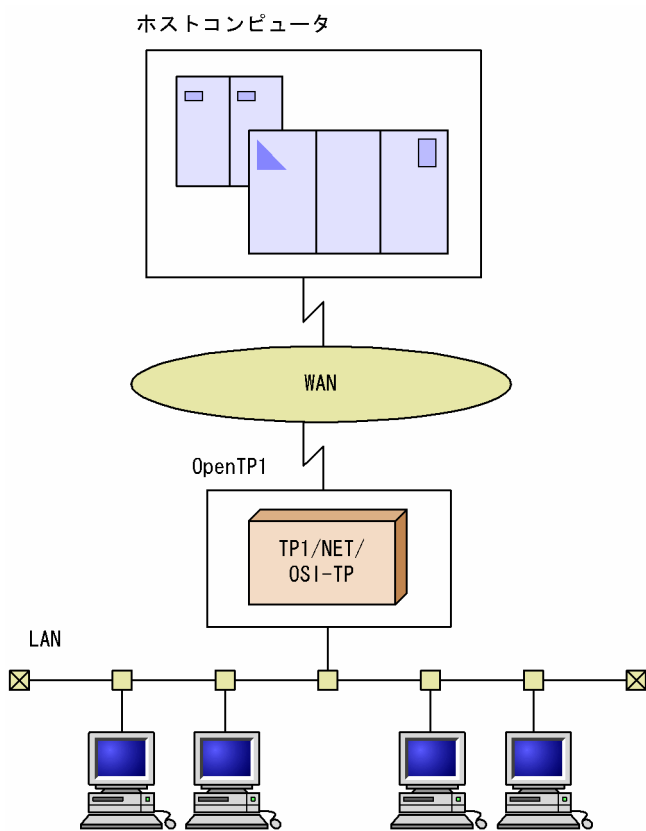
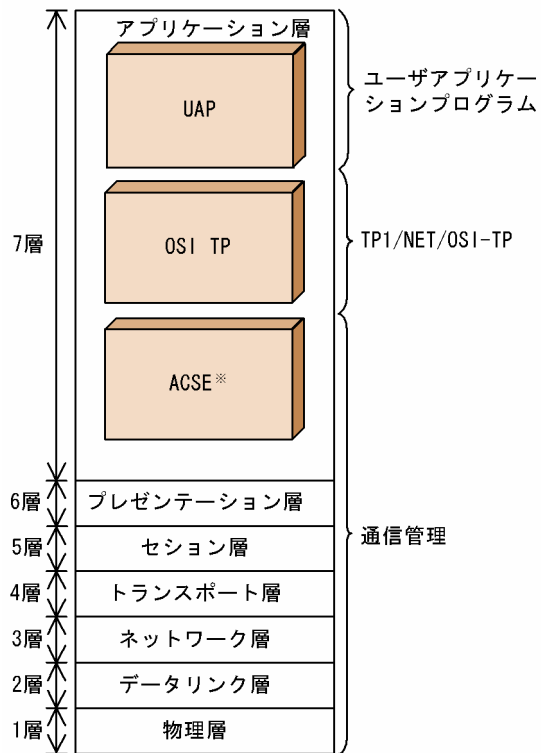


図 1-2 TP1/NET/OSI-TP の OSI7 層構造の中での機能範囲



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

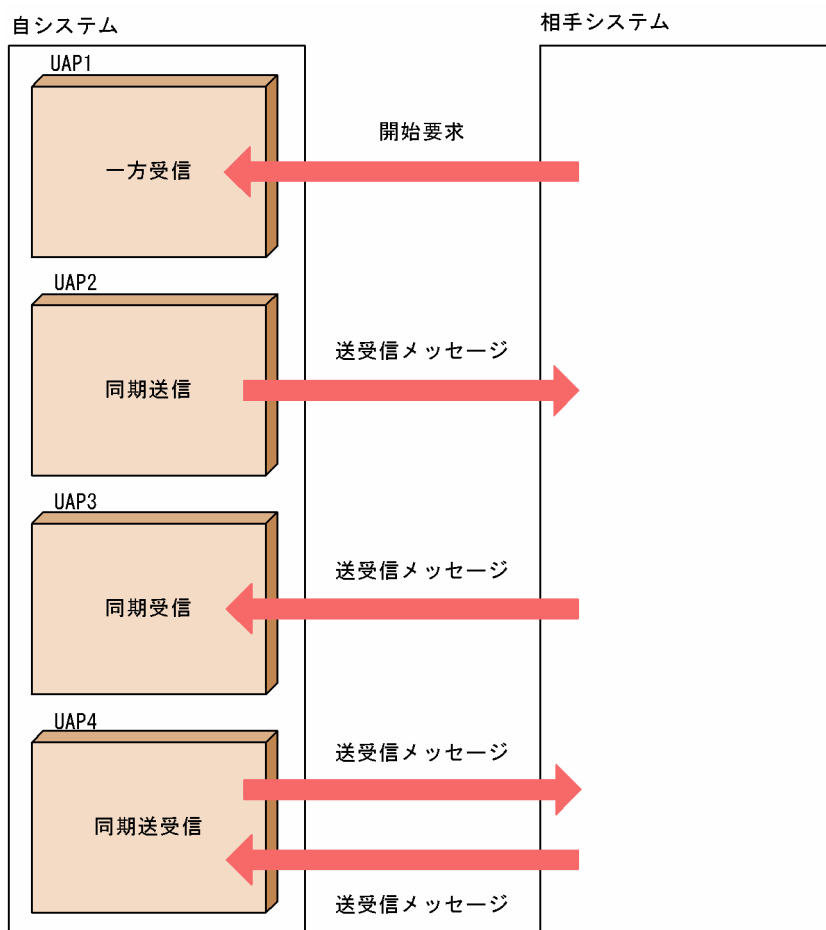
AP 間通信を使用すると、自システムで発生したトランザクションを相手システムで処理したり、その結果を受信したりできます。また、相手システムで発生したトランザクションを自システムで処理したり、その結果を送信したりできます。

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。

## (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 D TP1/NET/OSI-TP の実装範囲](#)」を参照してください。

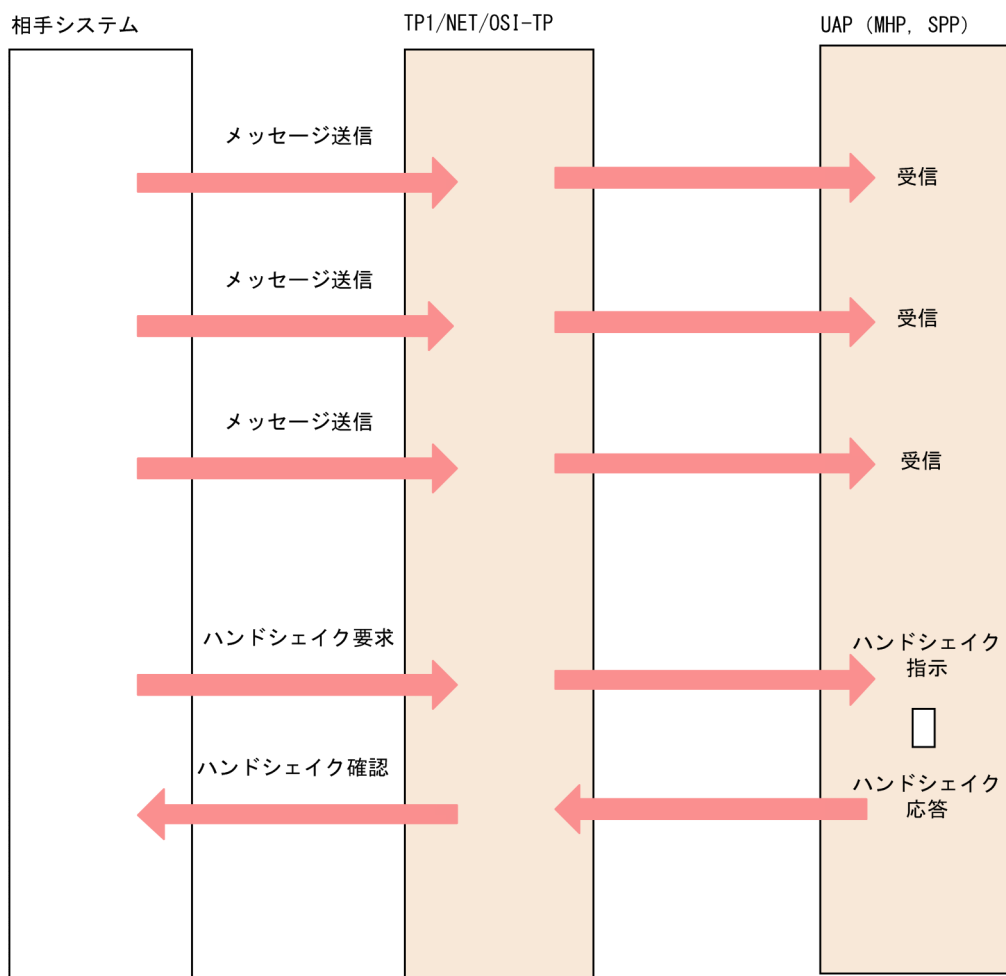
TP1/NET/OSI-TP は、相手システムとの間だけで行われる特別な処理や、メッセージの重複、欠落などのチェックはしません。したがって、AP 間通信をするときには、相手システムと自システムの UAP との間で、次に示す項目に関して事前に取り決めておく必要があります。

## (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

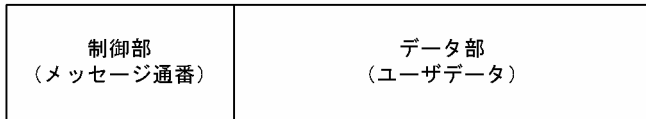


## (2) メッセージの重複、および欠落のチェック

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に分け、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。

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

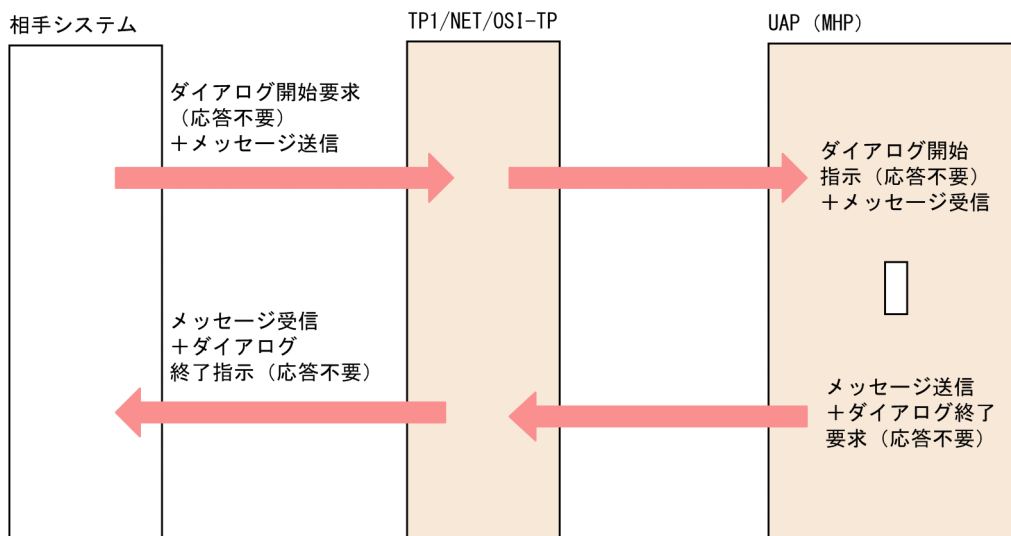
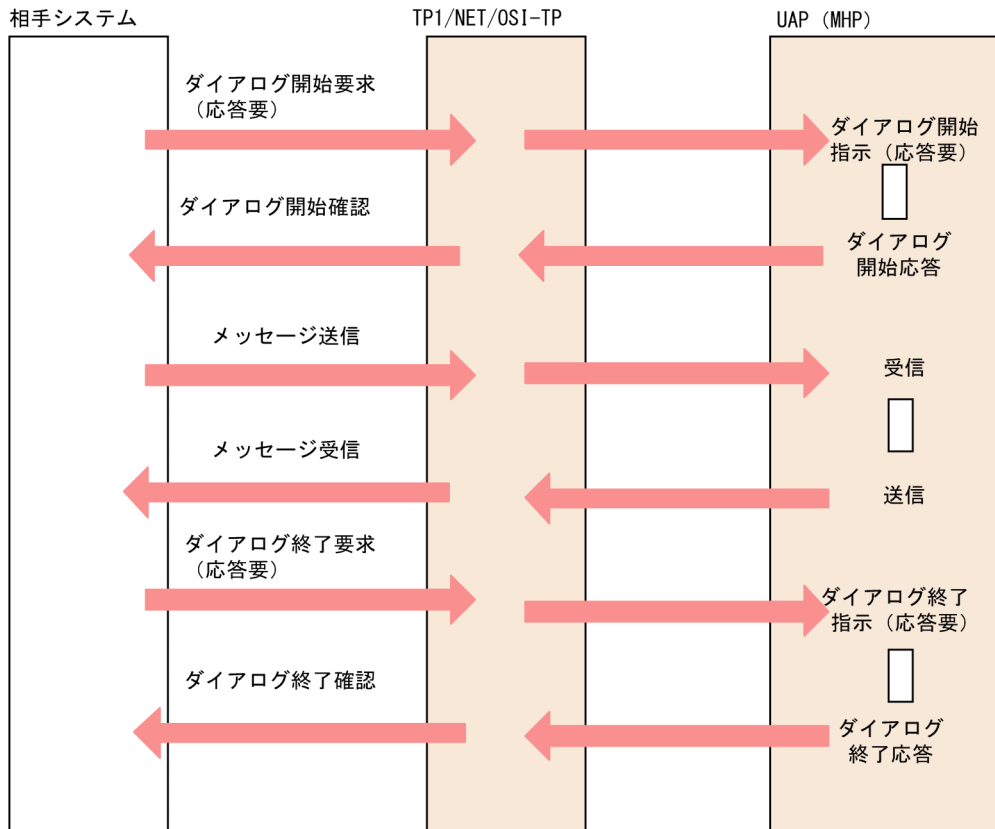


図 1-7 メッセージの応答を必要とした場合



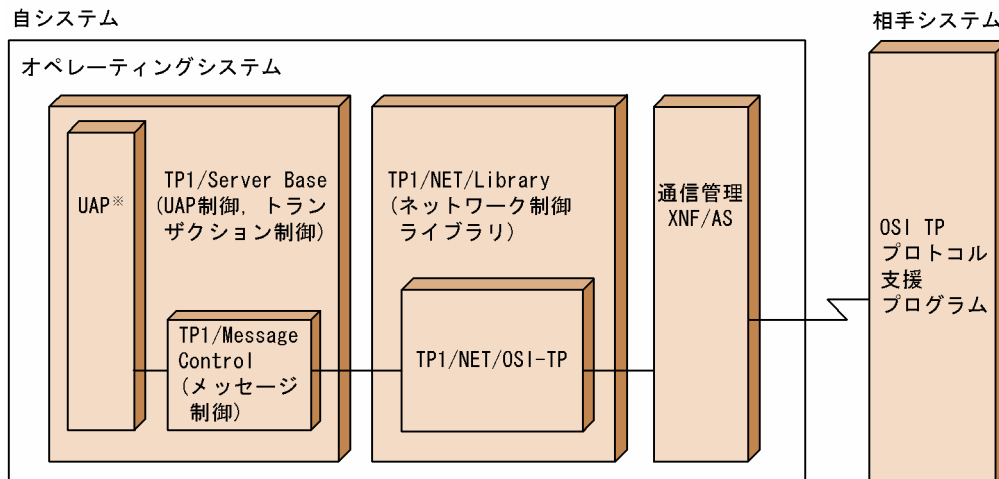
また、ダイアログの開始要求の応答を不要とした場合、それ以後に送信したメッセージが破棄されることがあります。詳細については、「2.2.4 ダイアログ」を参照してください。

## 1.3 ソフトウェア構成の例

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。

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接続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ます。

## 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

コネクションの種別には、次の二つがあります。

- 発呼型
- 着呼型

コネクションの種別は、コネクション定義 (`mcftalccn -t`) で指定します。

##### (a) 発呼型

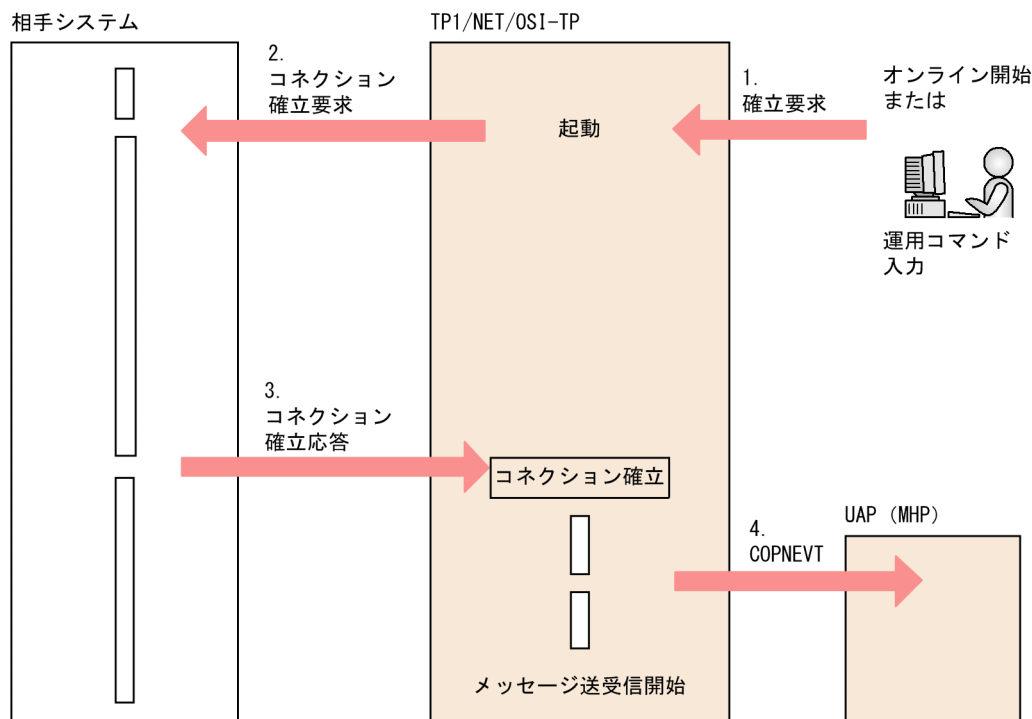
自システムからコネクションの確立を要求します。コネクションの確立方法には、次の三つがあります。

- コネクション定義 (`mcftalccn -i`) に `auto` を指定したことによるオンラインの開始・再開時の自動確立
- 運用コマンド (`mcftactcn`) 入力による手動確立
- API (`dc_mcf_tactcn` 関数または `CBLDCMCF("TACTCN△△")`) の発行による手動確立

TP1/NET/OSI-TP が相手システムにコネクションの確立を要求したあと、相手システムからのコネクション確立応答を受信すると、コネクションが確立し、メッセージの送受信ができる状態になります。

運用コマンド (`mcftactcn`) 入力時またはオンライン開始時の発呼型の確立方法を、次の図に示します。

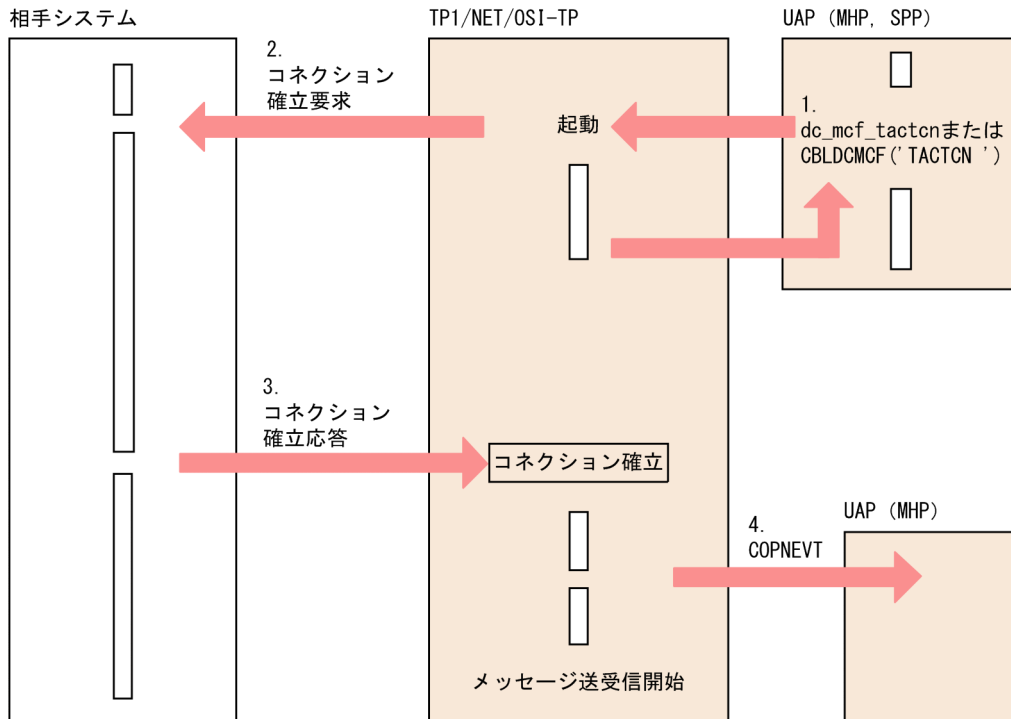
図 2-1 コネクションの確立 (発呼型 (運用コマンド入力時またはオンライン開始時))



1. オンライン開始・再開時、または運用コマンド (`mcftactcn`) を入力します。
2. TP1/NET/OSI-TP は、相手システムにコネクション確立要求を送信します。
3. 相手システムからコネクション確立応答を受信すると、コネクションが確立します。
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) を通知します。

API (`dc_mcf_tactcn` 関数または `CBLDCMCF("TACTCN△△")`) 発行時の発呼型の確立方法を、次の図に示します。

図 2-2 コネクションの確立（発呼型（API 発行時））



1. API (`dc_mcf_tactcn` 関数または `CBLDCMCF("TACTCN△△")`) を発行します。
2. TP1/NET/OSI-TP は、相手システムにコネクション確立要求を送信します。
3. 相手システムからコネクション確立応答を受信すると、コネクションが確立します。
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）を通知します。

## (b) 着呼型

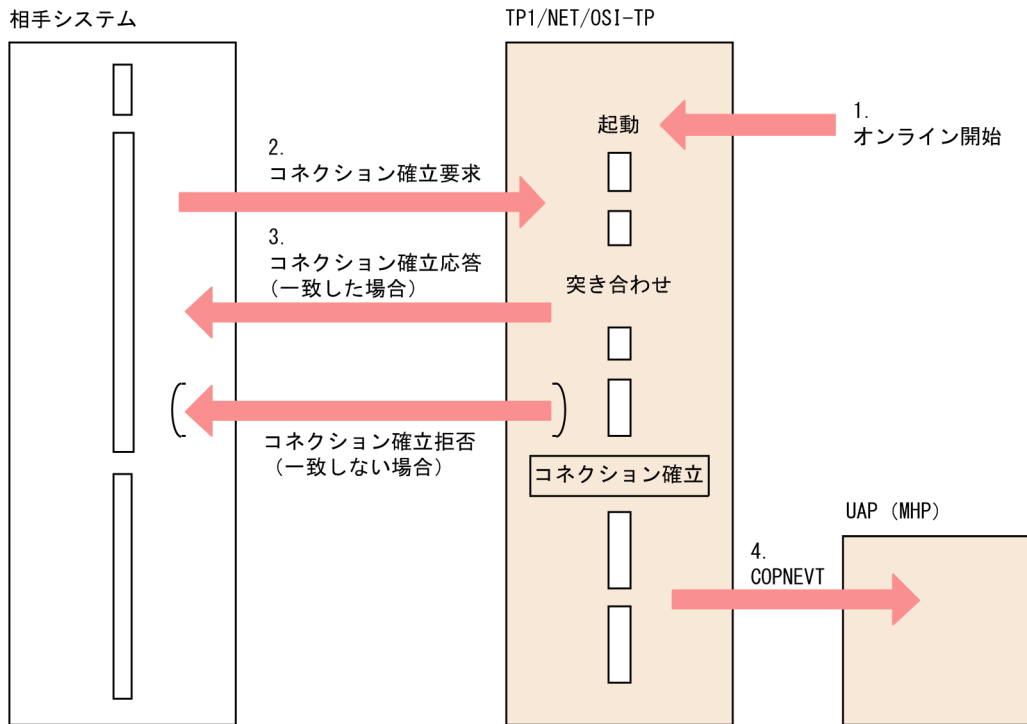
相手システムからのコネクションの確立要求を受け付けます。

オンラインの開始・再開時に相手システムからのコネクション確立要求を待ちます。確立要求を受けると、TP1/NET/OSI-TP は相手システムにコネクション確立応答を返します。応答を返した時点でコネクションが確立し、メッセージの送受信ができる状態になります。

着呼型のコネクションの場合、TP1/NET/OSI-TP では、コネクションが解放されたあと、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 2-3 コネクションの確立（着呼型）



1. オンラインの開始・再開時、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、コネクション確立拒否を相手システムに送信します。
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）を通知します。

注※

定義の突き合わせ項目については、「2.1.9 サービスプリミティブ」を参照してください。

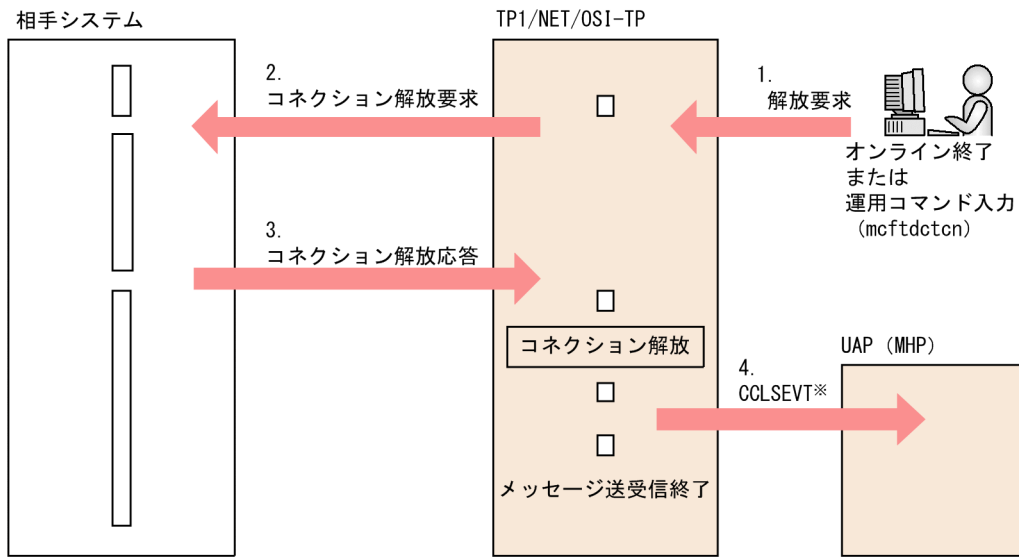
## (2) コネクションの正常解放

コネクションの正常解放には、次の四つがあります。

- オンライン終了時の解放
- 運用コマンド（mcftdctcn）入力による解放
- API（dc\_mcf\_tdctcn 関数または CBLDCMCF('TDCTCN△△'）発行による解放
- 相手システムからの解放要求による解放

オンライン終了時または運用コマンド (mcftdctcn) 入力時の自システムからの接続の解放を図 2-4 に、API (dc\_mcf\_tdctcn 関数または CBLDCMCF('TDCTCN△△')) 発行時の自システムからの接続の解放を図 2-5 に、相手システムからの接続の解放を図 2-6 に示します。

図 2-4 自システムからの接続の解放 (オンライン終了時または運用コマンド入力時)

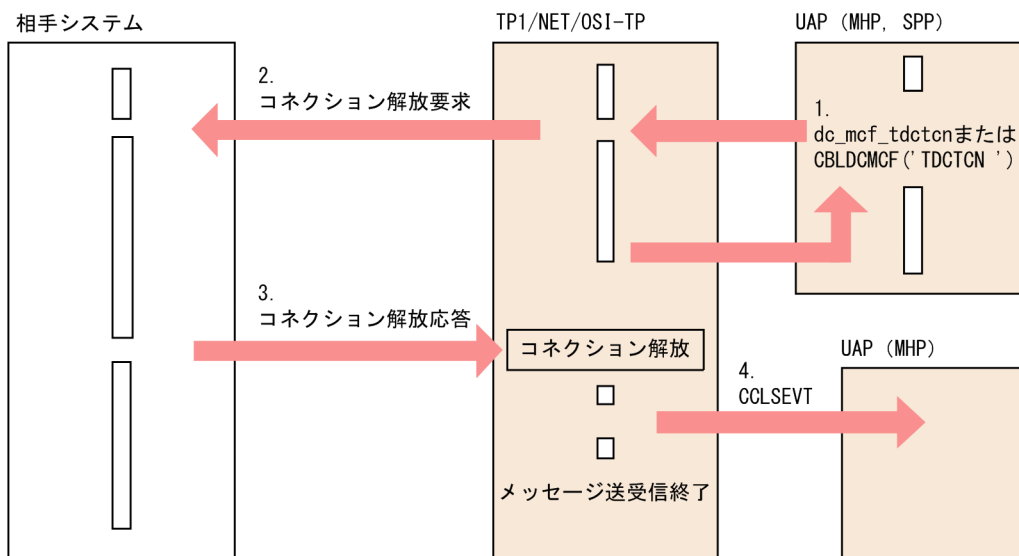


1. オンライン終了, または運用コマンド (mcftdctcn) を入力します。
2. TP1/NET/OSI-TP は, 相手システムに接続解放要求を送信します。
3. 相手システムから, コネクション解放応答を受信します。
4. TP1/NET/OSI-TP は, コネクション解放応答を受信すると, 状態通知イベント (CCLSEVT) を通知します。

注※

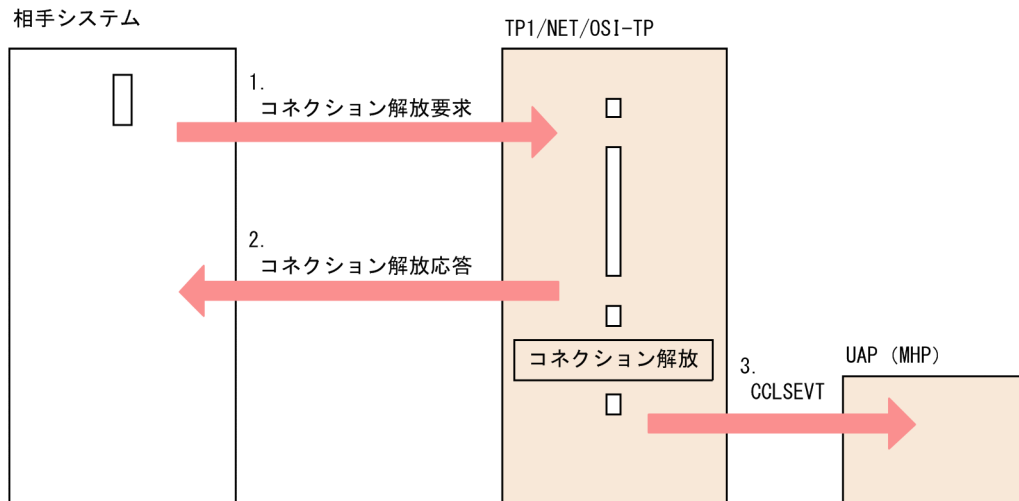
状態通知イベントは, オンライン終了時は通知しません。

図 2-5 自システムからの接続の解放 (API 発行時)



1. API (`dc_mcf_tdctcn` 関数または `CBLDCMCF('TDCTCN△△')`) を発行します。
2. TP1/NET/OSI-TP は、相手システムに接続解放要求を送信します。
3. 相手システムから、接続解放応答を受信します。
4. TP1/NET/OSI-TP は、接続解放応答を受信すると、状態通知イベント (CCLSEVT) を通知します。

図 2-6 相手システムからの接続の解放



1. 相手システムから接続解放要求を受信します。
2. TP1/NET/OSI-TP は、相手システムに接続解放応答を送信します。
3. TP1/NET/OSI-TP は、接続が解放すると状態通知イベント (CCLSEVT) を通知します。

### (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、次の四つがあります。

- 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 (`mcftdctcn -f`) 入力による解放
- 強制解放オプション※を設定した API (`dc_mcf_tdctcn` 関数または `CBLDCMCF('TDCTCN△△')`) 発行による解放
- 相手システムに何らかの異常が発生した場合の解放

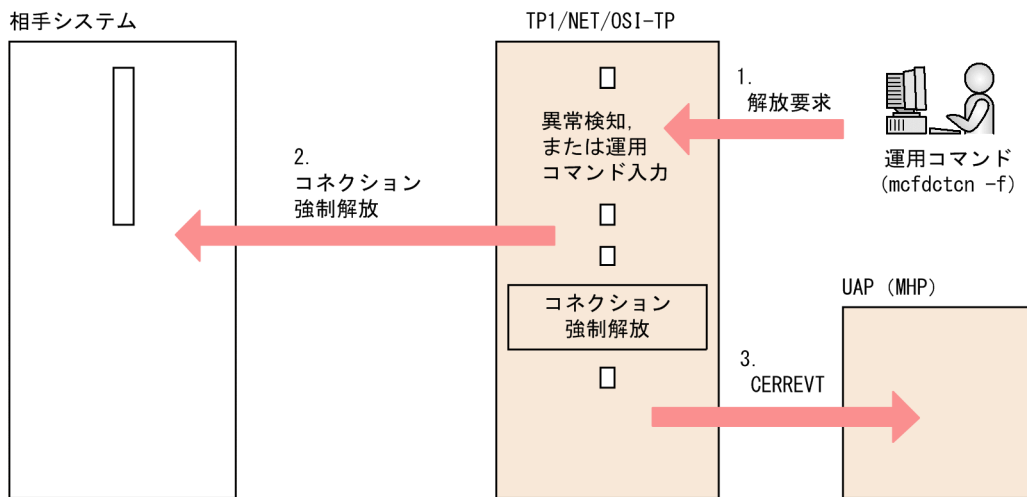
注※

`dc_mcf_tdctcn` 関数の場合、`action` 引数に `DCMCFERC` を指定します。

`CBLDCMCF('TDCTCN△△')` の場合、データ名 `D1` に `'I'` を指定します。

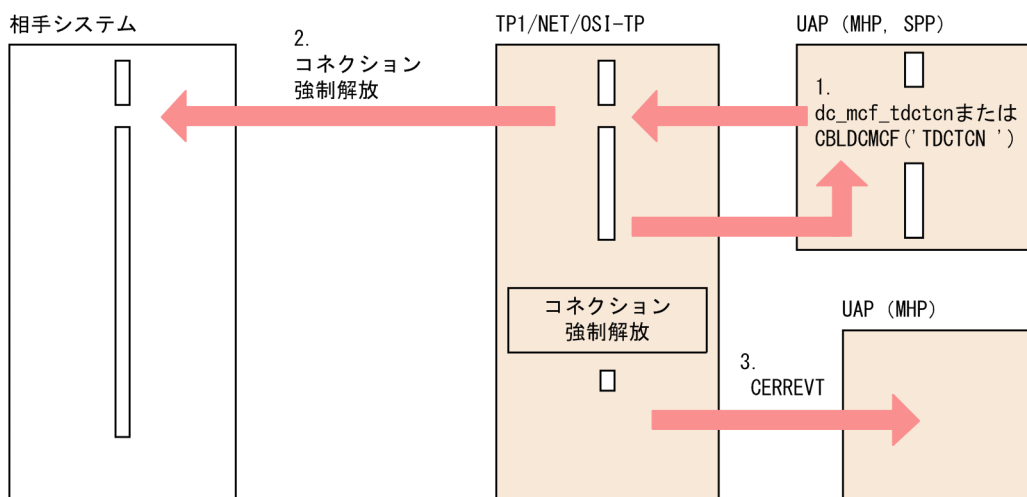
異常検知時または運用コマンド (`mcftdctcn -f`) 入力時の自システムからの接続の強制解放を図 2-7 に、API (`dc_mcf_tdctcn` 関数または `CBLDCMCF('TDCTCN△△')`) 発行時の自システムからの接続の強制解放を図 2-8 に、相手システムの異常による解放を図 2-9 に示します。

図 2-7 自システムからの接続の強制解放（異常検知時または運用コマンド入力時）



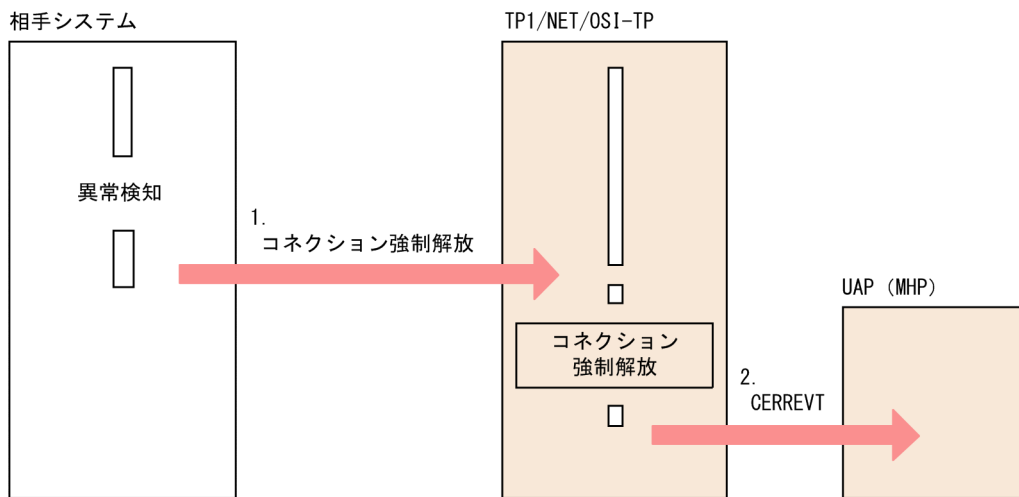
1. TP1/NET/OSI-TP 内部で異常を検出，または運用コマンド (`mcftdctcn -f`) を入力します。
2. TP1/NET/OSI-TP は，相手システムにコネクション強制解放を送信します。
3. TP1/NET/OSI-TP は，コネクションが強制解放すると状態通知イベント (CERREVT) を通知します。

図 2-8 自システムからの接続の強制解放（API 発行時）



1. 強制解放オプションを設定した API (`dc_mcf_tdctcn` 関数または `CBLDCMCF('TDCTCN△△')`) を発行します。
2. TP1/NET/OSI-TP は，相手システムにコネクション強制解放を送信します。
3. TP1/NET/OSI-TP は，コネクションが強制解放すると，状態通知イベント (CERREVT) を通知します。

図 2-9 相手システムの異常による解放

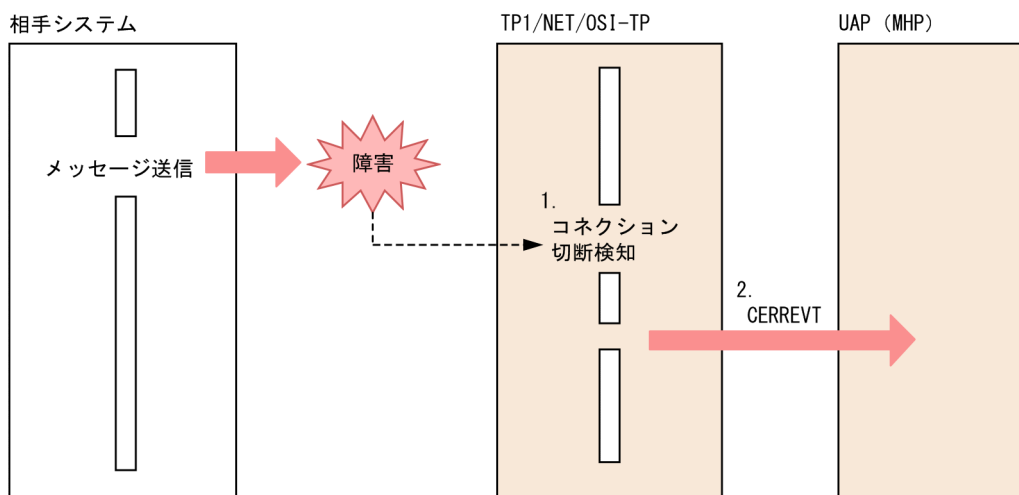


1. 相手システムでの異常，および通信回線の障害によって，相手システムからコネクションの強制解放を受信します。
2. TP1/NET/OSI-TP は，コネクションが強制解放すると状態通知イベント（CERREVT）を通知します。

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，「9.2 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 2-10 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 が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）を通知します。

## 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。

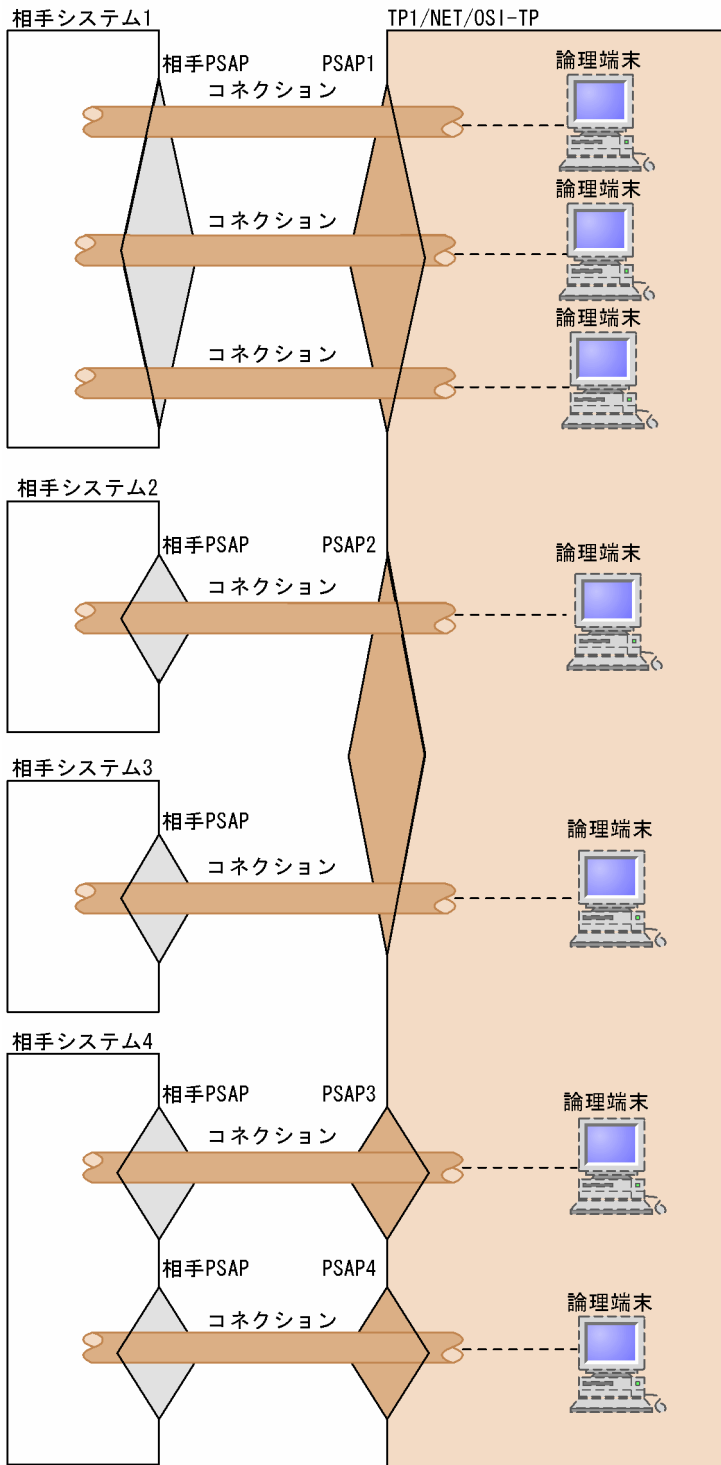
コネクションと論理端末の指定を対応させると、自システムと相手システムとの論理的通信路が確立でき、AP 間通信ができるようになります。コネクションと論理端末は、システム定義で対応させます。

## 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。

図 2-11 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 2-11 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、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、「2.1.8 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 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。

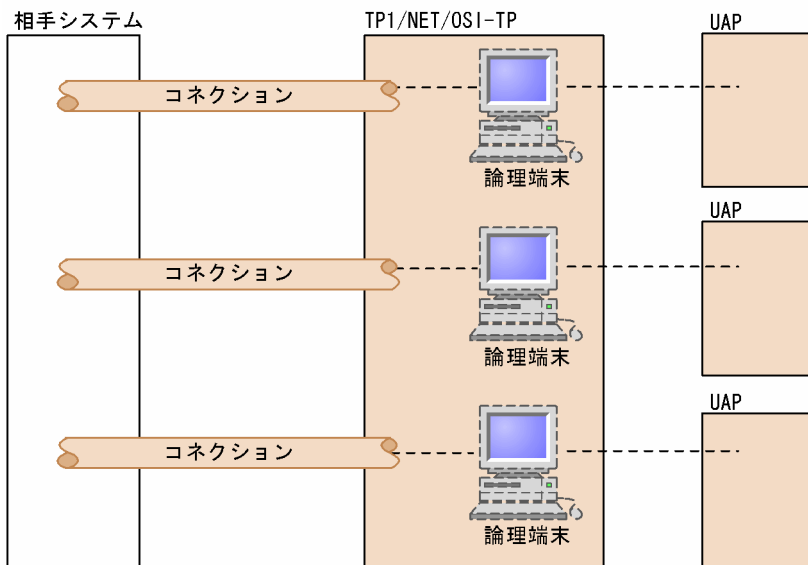
表 2-2 論理端末の端末タイプ、メッセージ、アプリケーションの型、UAP インタフェース、および通信形態の関係

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
			sendrecv	同期送受信

## 2.1.5 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 2-12 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、「2.1.8 アソシエーションプール管理機能」を参照してください。

## 2.1.6 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 (`mcftalccn -d control`) で指定します。

## 2.1.7 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 2-13 に、連結 APDU の場合を図 2-14 に示します。

図 2-13 処理層とデータの関係 (単一 APDU の場合)

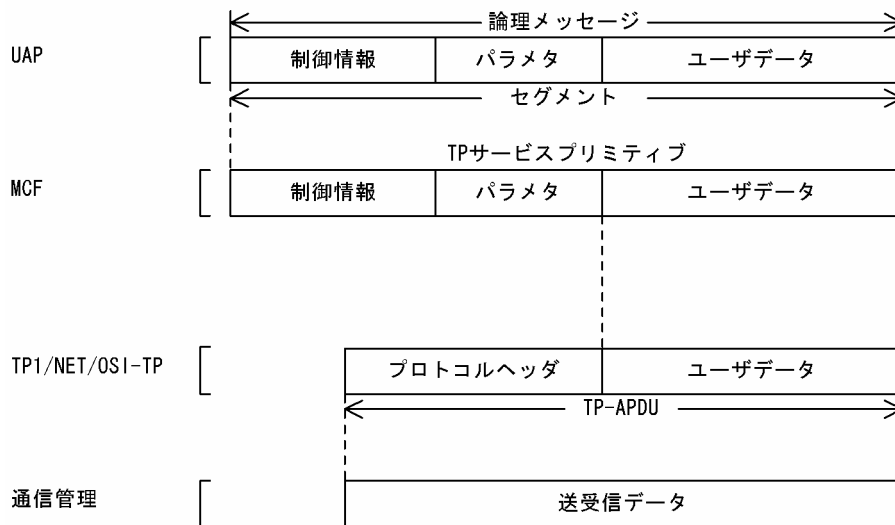
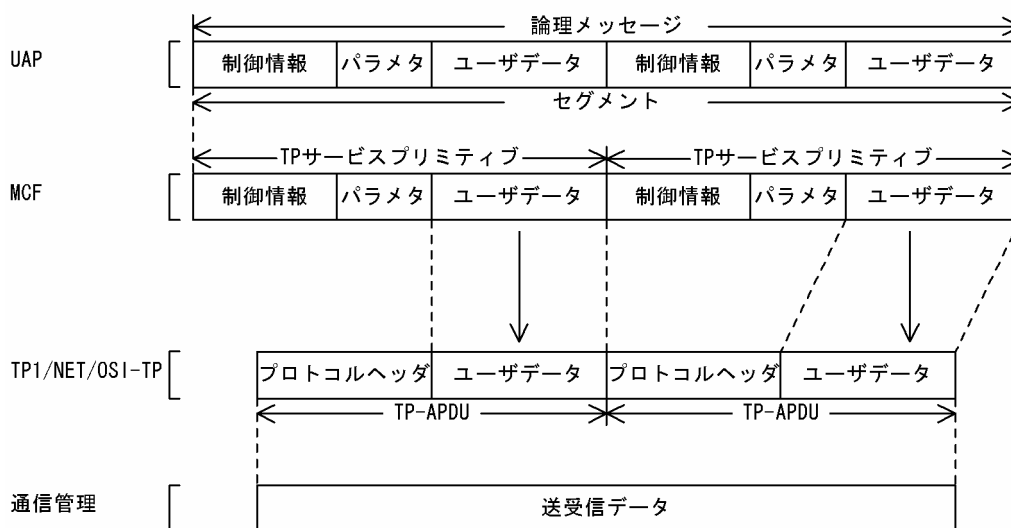


図 2-14 処理層とデータの関係 (連結 APDU の場合)



注

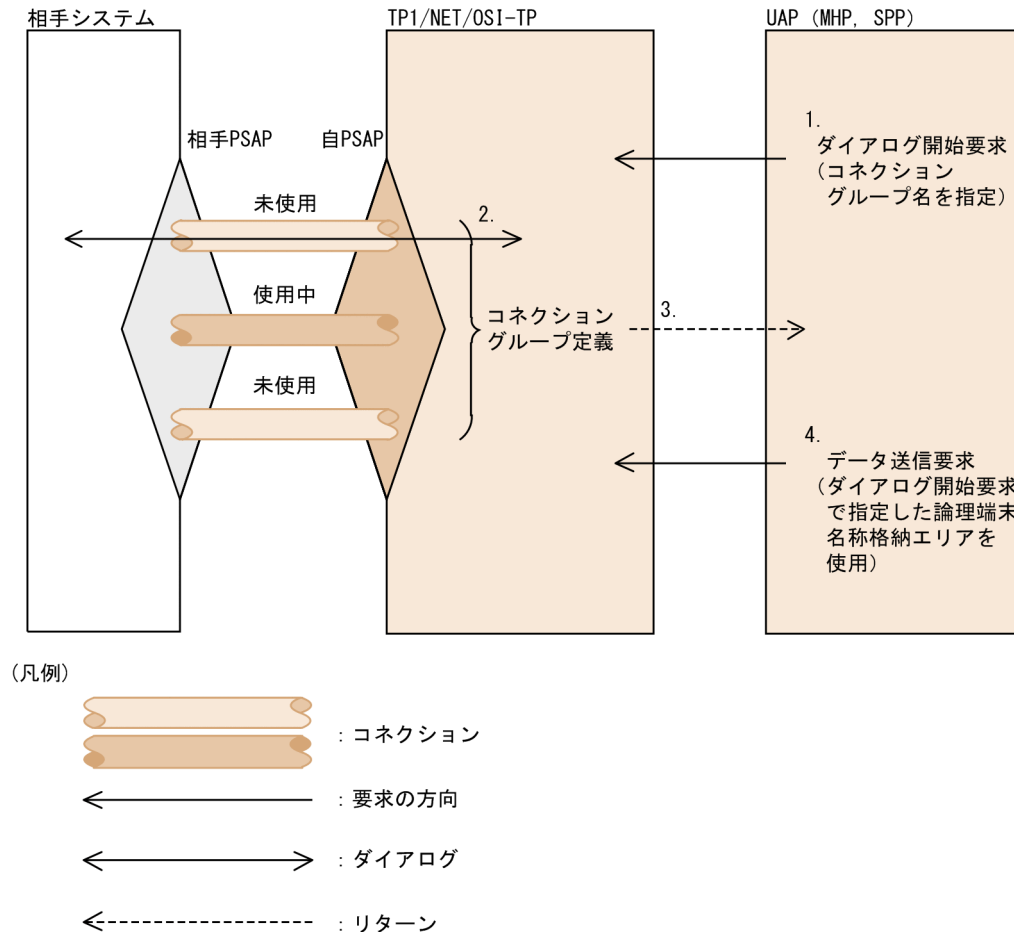
この場合は、2 個連結の例です。

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。

## 2.1.8 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、コネクシヨングループ定義 (mcftgrpcn -g) で指定したコネクシヨングループ名を指定すれば、TP1/NET/OSI-TP がコネクシヨングループ内の使用できるコネクシヨ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、同じコネクシヨングループのコネクシヨンでは、コネクシヨン 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 2-15 アソシエーションプール管理機能の概要



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ヨ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ヨングループ配下の未使用のコネクシヨ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ヨングループ名を、ダイアログ開始時に選択したコネクシヨ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。

## 2.1.9 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 2-3 に、サービスプリミティブと UAP との対応を表 2-4 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 2-5 に示します。

表 2-3 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	エラー発生の通知要求/指示 <ul style="list-style-type: none"> <li>• TP-HANDSHAKEreq 確認型</li> <li>• TP-END-DIALOGUEreq</li> <li>• TP-HANDSHAKE-AND-GRANT-CONTROLreq に対する拒否応答</li> </ul>	○	○	○	○	○
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	○	×	○	○	×
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	×	×	○	○	×

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-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。
- ×
- －：該当しません。

表 2-4 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	<ul style="list-style-type: none"> <li>• sendsync</li> <li>• sendrcv</li> </ul>	<ul style="list-style-type: none"> <li>• SENDSYNC</li> <li>• SENDRECV</li> </ul>	ENABLE	BDR
	ind	receive	RECEIVE	RECEIVE	BDI
	rsp (受諾)	<ul style="list-style-type: none"> <li>• sendsync</li> <li>• sendrcv</li> </ul>	<ul style="list-style-type: none"> <li>• SENDSYNC</li> <li>• SENDRECV</li> </ul>	ENABLE	BDPA
	rsp (拒否)	sendsync	SENDSYNC	DISABLE	BDPN
	cnf	<ul style="list-style-type: none"> <li>• recvsync</li> <li>• sendrcv</li> </ul>	<ul style="list-style-type: none"> <li>• RECVSYNC</li> <li>• SENDRECV</li> </ul>	RECEIVE	BDC
TP-END-DIALOGUE	req	<ul style="list-style-type: none"> <li>• sendsync</li> <li>• sendrcv※2</li> </ul>	<ul style="list-style-type: none"> <li>• SENDSYNC</li> <li>• SENDRECV※2</li> </ul>	DISABLE	EDR
	ind	<ul style="list-style-type: none"> <li>• recvsync</li> <li>• sendrcv</li> </ul>	<ul style="list-style-type: none"> <li>• RECVSYNC</li> <li>• SENDRECV</li> </ul>	RECEIVE	EDI
	rsp	<ul style="list-style-type: none"> <li>• sendsync</li> </ul>	<ul style="list-style-type: none"> <li>• SENDSYNC</li> </ul>	DISABLE	EDP
	cnf	<ul style="list-style-type: none"> <li>• recvsync</li> <li>• sendrcv</li> </ul>	<ul style="list-style-type: none"> <li>• RECVSYNC</li> <li>• SENDRECV</li> </ul>	RECEIVE	EDC
TP-U-ERROR	req	<ul style="list-style-type: none"> <li>• sendsync</li> <li>• sendrcv</li> </ul>	<ul style="list-style-type: none"> <li>• SENDSYNC</li> <li>• SENDRECV</li> </ul>	SEND	UER
	ind	recvsync	RECVSYNC	RECEIVE	UEI
TP-U-ABORT	req	sendsync	SENDSYNC	DISABLE	UAR
	ind	<ul style="list-style-type: none"> <li>• recvsync</li> </ul>	<ul style="list-style-type: none"> <li>• RECVSYNC</li> </ul>	RECEIVE	UAI

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-U-ABORT	ind	• sendrecv	• SENDRECV	RECEIVE	UAI
TP-P-ABORT	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	PAI
TP-HANDSHAKE	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-CONTROL	req	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	GCR
	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	GCI
TP-REQUEST-CONTROL	req	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	RCR
	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	RCI
TP-HANDSHAKE-AND-GRANT-CONTROL	req	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	HGR
	ind	• recvsync • sendrecv	• RECVSYNC • SENDRECV	RECEIVE	HGI
	rsp	• sendsync • sendrecv	• SENDSYNC • SENDRECV	SEND	HGP
	cnf	• recvsync • sendrecv	• 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す。

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 2-5 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"><li>• sendsync</li><li>• sendrecv</li></ul>	<ul style="list-style-type: none"><li>• SENDSYNC</li><li>• SENDRECV</li></ul>	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"><li>• sendsync</li><li>• sendrecv※2</li></ul>	<ul style="list-style-type: none"><li>• SENDSYNC</li><li>• SENDRECV※2</li></ul>	DISABLE
最後が EDR (TDR+EDR,BDPA+EDR,HSP+EDR など)	<ul style="list-style-type: none"><li>• sendsync</li><li>• sendrecv※2</li></ul>	<ul style="list-style-type: none"><li>• SENDSYNC</li><li>• SENDRECV※2</li></ul>	DISABLE
その他の送信 (TDR+HSR,TDR+TDR など)	<ul style="list-style-type: none"><li>• sendsync</li><li>• sendrecv</li></ul>	<ul style="list-style-type: none"><li>• SENDSYNC</li><li>• SENDRECV</li></ul>	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"><li>• recvsync</li><li>• sendrecv</li></ul>	<ul style="list-style-type: none"><li>• RECVSYNC</li><li>• SENDRECV</li></ul>	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す。

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。

注※3

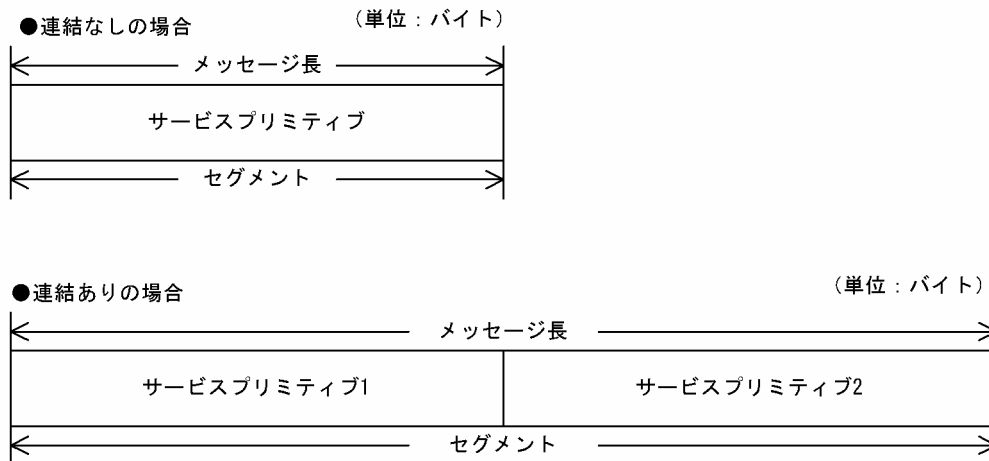
この組み合わせでは BDR は拒否応答型だけ、EDR は確認型だけ使用できます。

## (1) 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 2-16 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。

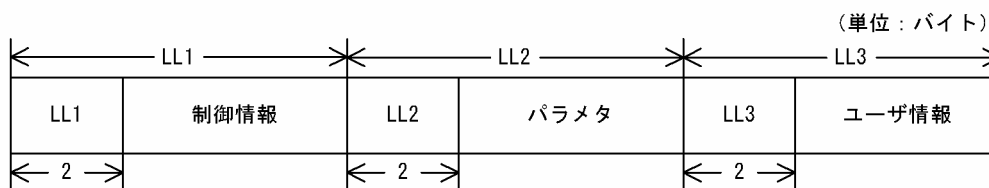
## サービスプリミティブ

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 2-17 サービスプリミティブの形式

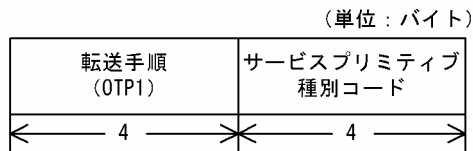


## 制御情報

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 2-4 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。

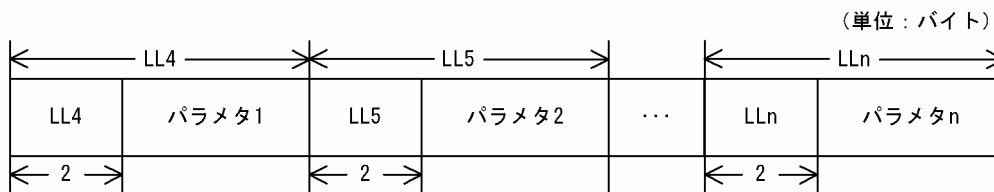
図 2-18 制御情報の形式



## 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)<sub>16</sub>を設定します。

図 2-19 パラメタの形式

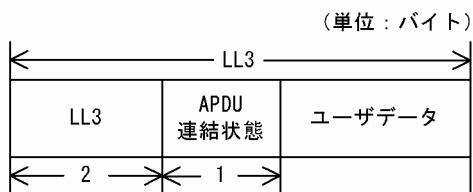


## ユーザ情報

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。

図 2-20 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。

## (2) 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)<sub>16</sub>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 2-6 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	—	—	—	—	—	—	—	×
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	—	—	—	—	—	—	—	×

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-HANDSHAKE-AND-GRANT-CONTROL	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
- (9) : Diagnostic
- (10) : Rollback
- (11) : Confirmation-Urgency
- ◎ : 設定は必須です。
- : 設定できます。
- × : 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、領域だけを確認してください。

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。

### (3) 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)<sub>16</sub> 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。  
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)<sub>16</sub>：受信側 TPSU 名称が定義されていません。  
(0002)<sub>16</sub>：TPSU が利用できません (永続的)。  
(0003)<sub>16</sub>：TPSU が利用できません (一時的)。  
(0004)<sub>16</sub>：受信側 TPSU 名称の指定が必要です。

(0005)<sub>16</sub>：機能単位が提供されていません。

(0006)<sub>16</sub>：機能単位の組み合わせが提供されていません。

(0008)<sub>16</sub>：その他の理由

- **ダイアログ異常終了 (TP-P-ABORT) の場合**

(0001)<sub>16</sub>：永続的障害が発生しました。

(0002)<sub>16</sub>：トランザクションの開始を拒否します。

(0003)<sub>16</sub>：一時的障害が発生しました。

(0004)<sub>16</sub>：プロトコルエラーです。

(0005)<sub>16</sub>：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。

## (4) 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 2-7 および表 2-8 に示します。

表 2-7 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。

突き合わせ項目	チェック内容
contention-winner-assignment	コネクション定義 (mcfalccn -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、「6. 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。

表 2-8 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。

## 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。

- 非同期型の一方送信メッセージの受信
- 同期型の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 2-9 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。

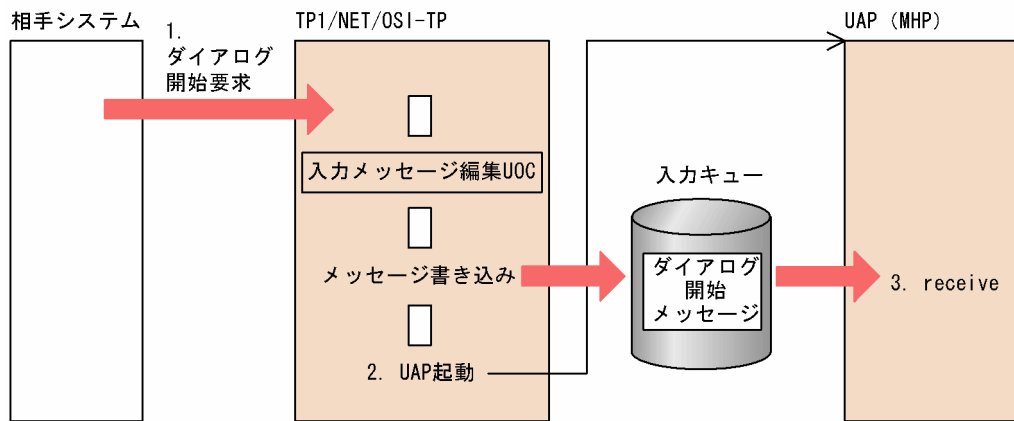
### 2.2.1 非同期型のメッセージの受信

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 2-21 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。

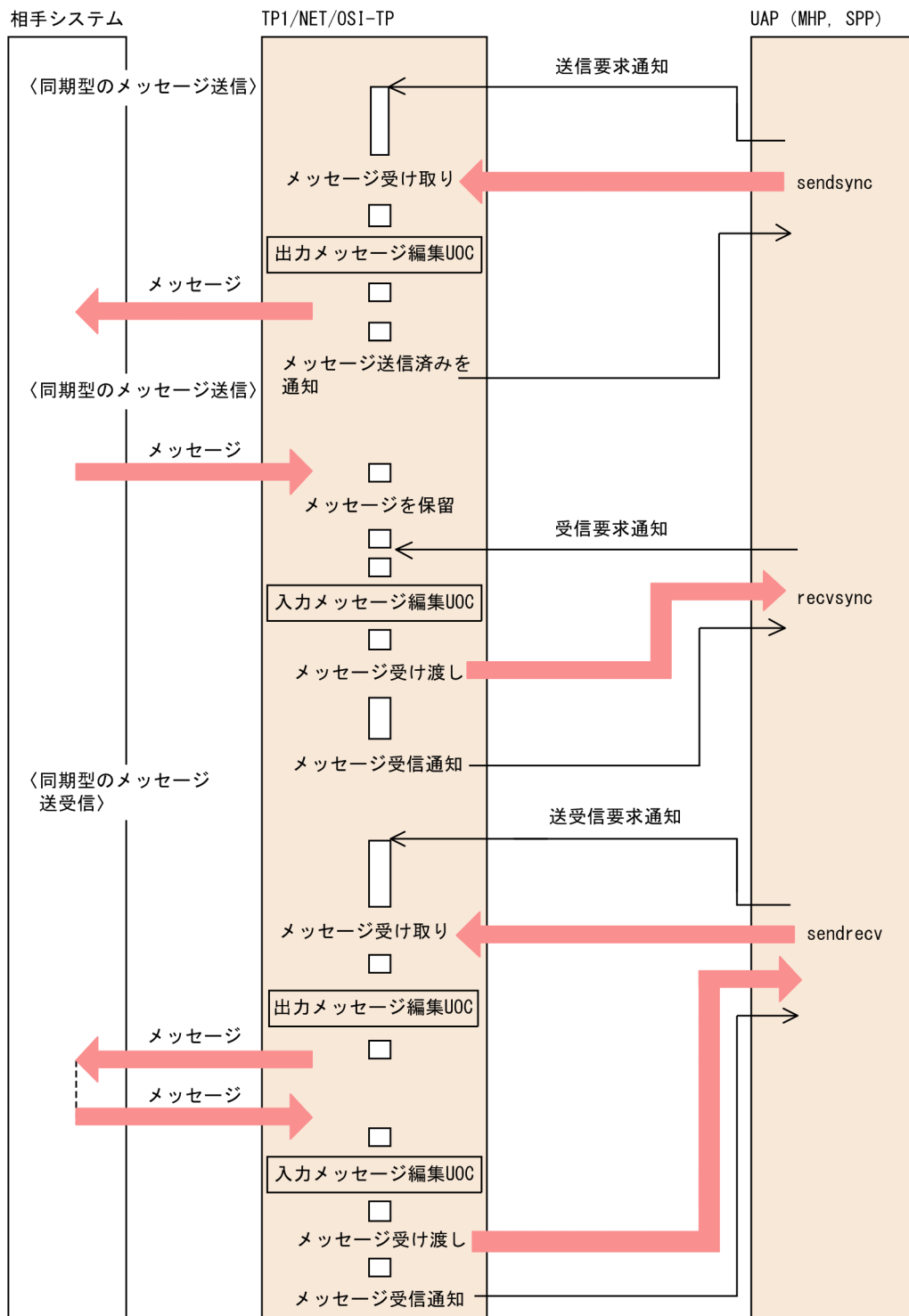
## (3) 同期型のメッセージの送受信

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 2-22 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コネクションの状態をリセットする必要があります。次のどちらかの方法でコネクションを強制解放してください。

- UAP から運用コマンド (`mcftdctcn -f`) を入力する
- 強制解放オプションを設定した API (`dc_mcf_tdctcn` 関数または `CBLDCMCF('TDCTCN△△')`) を発行する

## 2.2.3 制御権

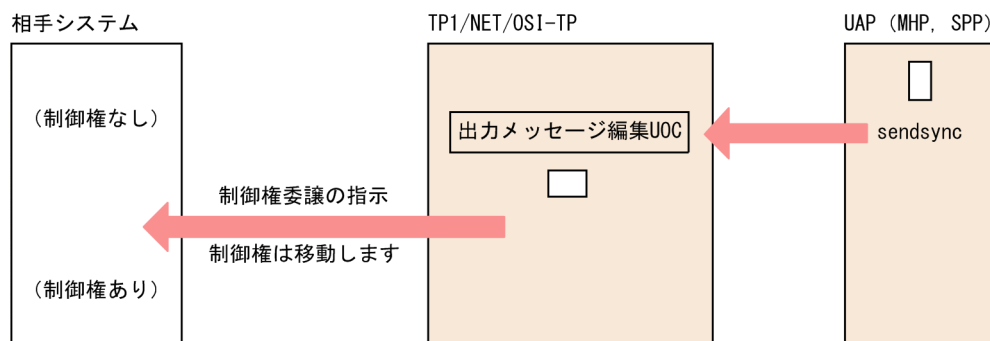
半二重機能を選択した場合、TP1/NET/OSI-TP は**制御権**によって、データの送信を制御します。制御権を持っている場合だけ、自システムからデータを送信要求できます。制御権は一つのダイアログの間にだけ有効です。制御権の受け渡しについては、相手システムとの間で事前に取り決めておく必要があります。

### (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 2-23 制御権委譲

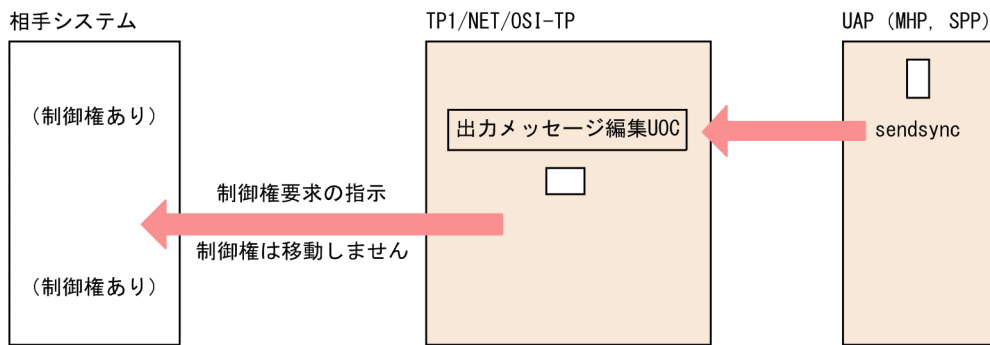


### (2) 制御権の要求

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 2-24 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 2-10 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
	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す。

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。

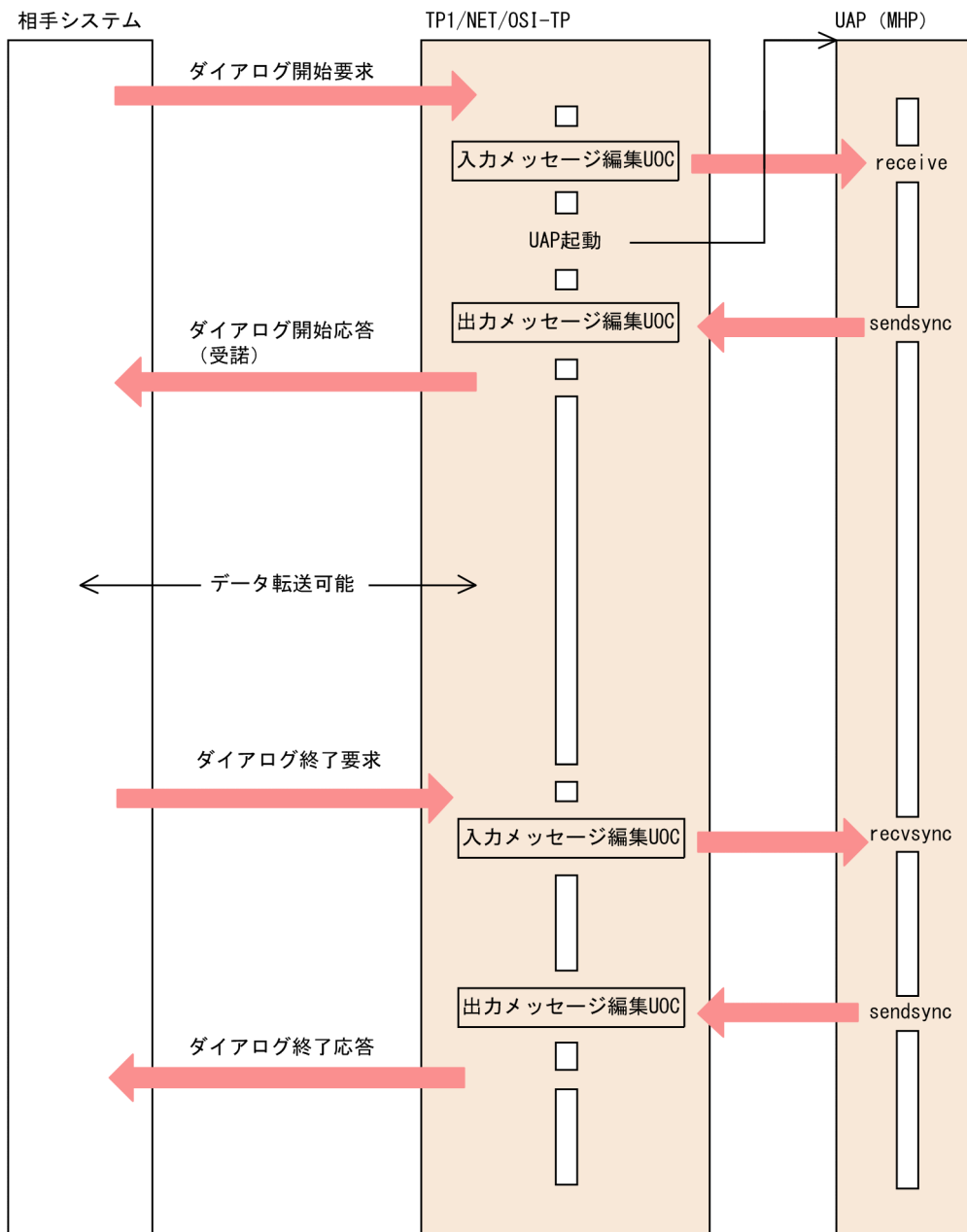
相手システムからダイアログ終了要求を受けた場合は、TP1/NET/OSI-TP は UAP にダイアログ終了要求を渡します。確認型の場合、UAP は SENDSYNC 文、または sendsync 関数でダイアログ終了応答を TP1/NET/OSI-TP に送信します。確認型でダイアログ終了を受け入れない場合は、ダイアログ終了拒否応答を送信します。

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、「2.1.9 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 2-25 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、接続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、接続も強制解放されます。

## (4) ダイアログと UAP

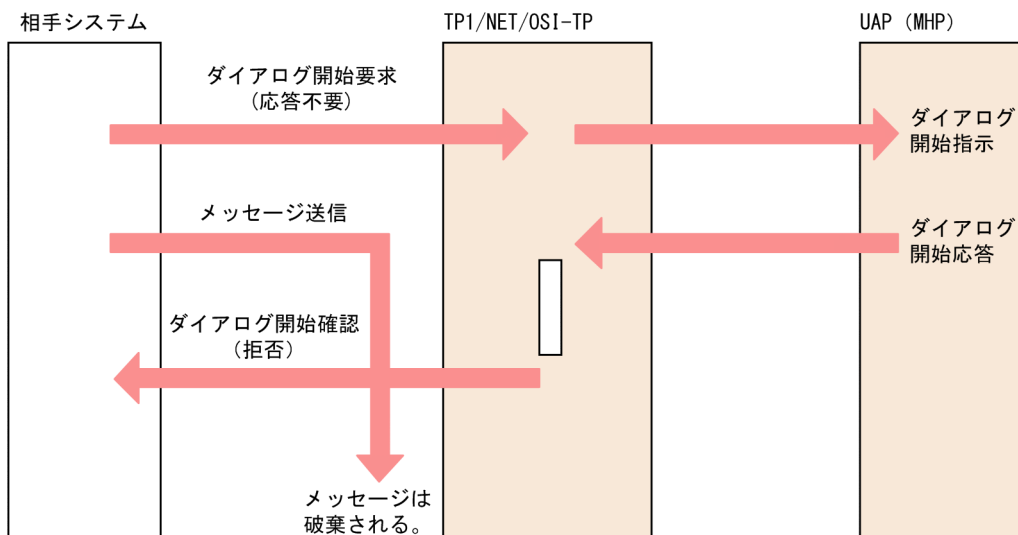
TP1/NET/OSI-TP では、一つの UAP でダイアログの開始から終了までの一連のメッセージ送受信をする必要があります。TP1/NET/OSI-TP は、UAP の終了時に未終了のダイアログを中止します。また、UAP からロールバックの資源回復要求をする場合、その要求はダイアログとは無関係です。ダイアログを継続して使用するか、中止するかは UAP に依存します。

## (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 2-26 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。

- 論理端末定義 (`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

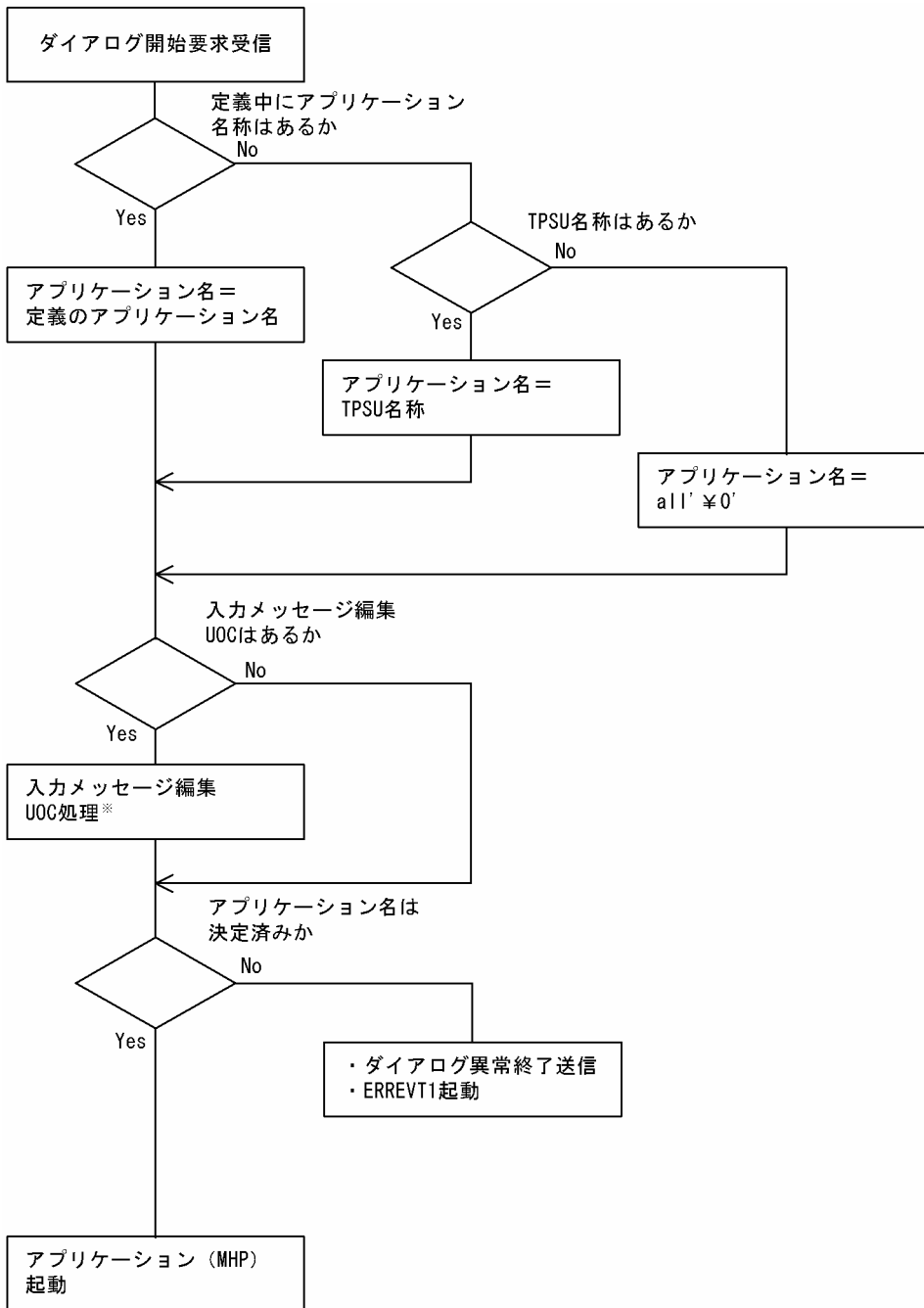
アプリケーション名の決定の優先順位を, 高い順に次に示します。

1. 入力メッセージ編集 UOC での決定
2. 論理端末定義 (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 2-27 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、「5.1.1 入力メッセージの編集とアプリケーション名の決定」を参照してください。

## 2.2.6 ハンドシェイク

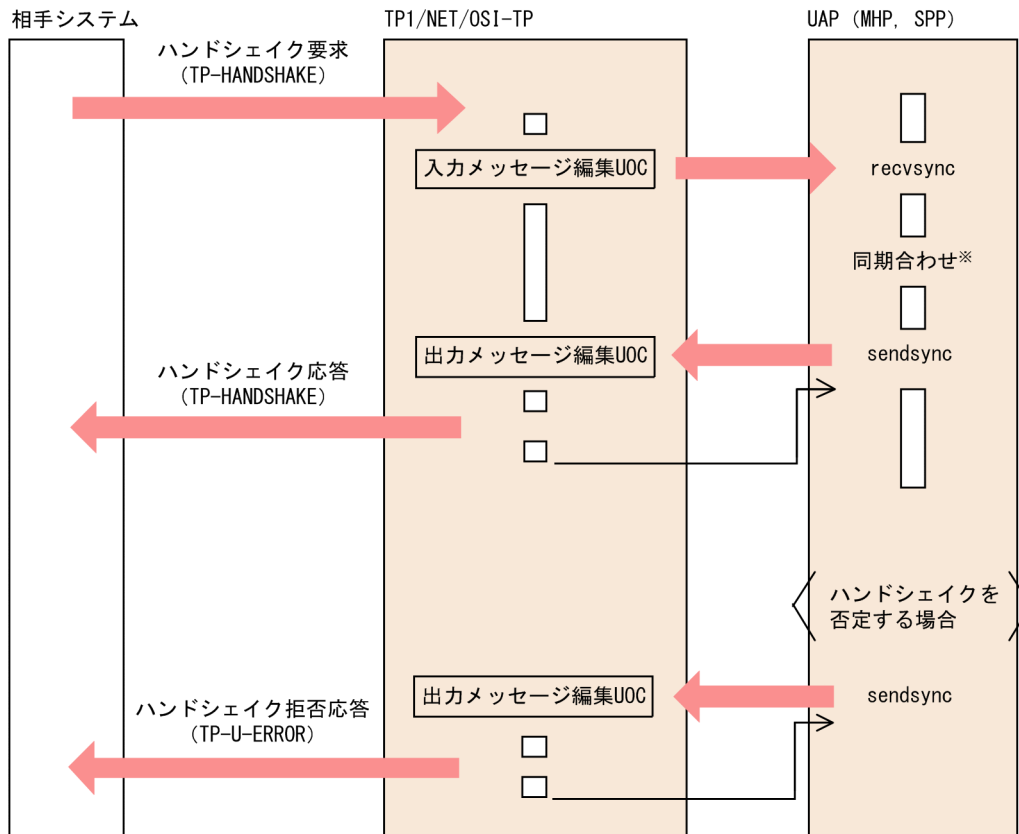
ハンドシェイクとは、システム間で同期を取ってデータを処理するための機能です。同期用の処理をユーザ側で設定する必要がないため、処理を軽減できます。

ハンドシェイク要求は、全二重の場合コンテンツ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 2-28 ハンドシェイク

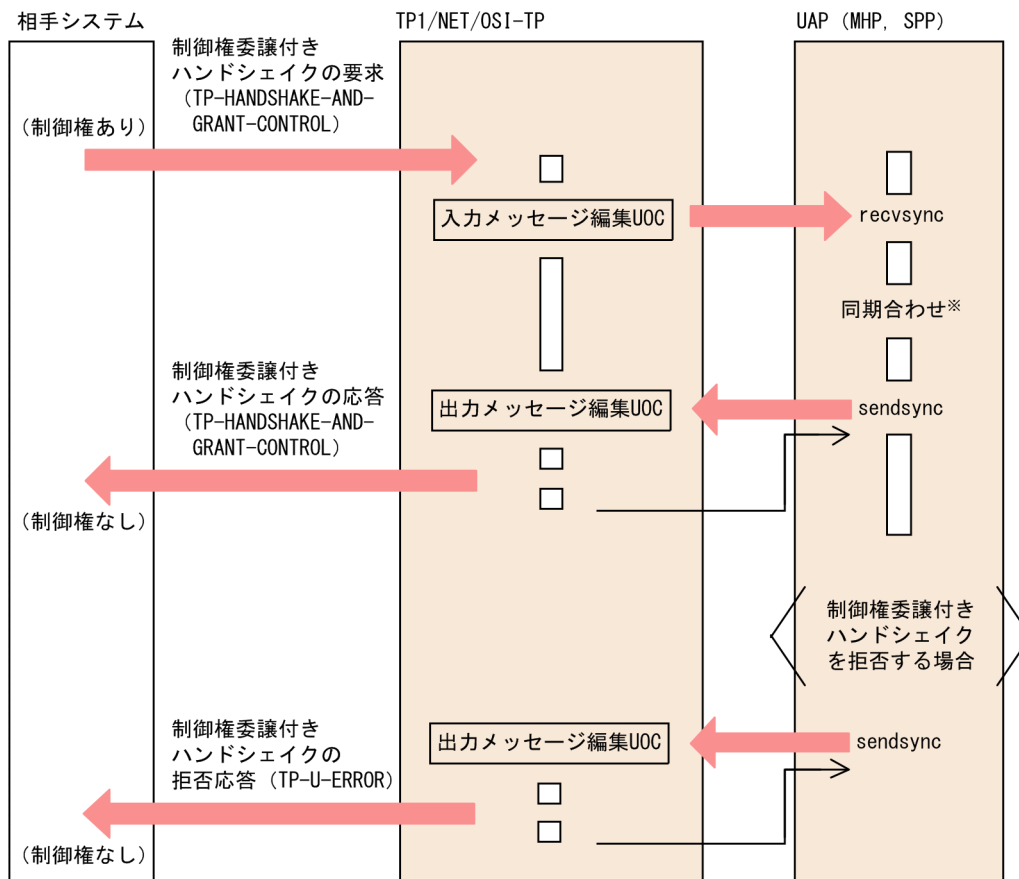


注※  
同期が取れなかった場合、UAP はハンドシェイクを拒否します。TP1/NET/OSI-TP は、相手システムにハンドシェイク拒否応答を返します。

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 2-29 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、自システムの利用者データを相手システムに転送して処理できます。また、相手システムの利用者データも受信できます。

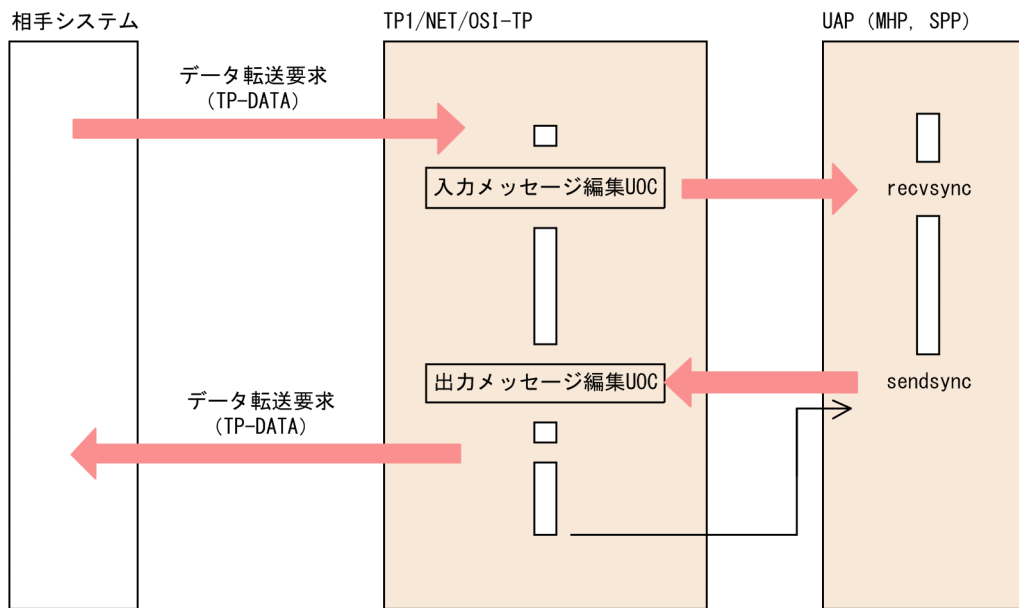
相手システムからデータ転送要求を受けると、TP1/NET/OSI-TP はメッセージを編集し、UAP に渡します。

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 2-30 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とがあります。

### (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 2-31 に、相手システムからのエラー通知受信の処理を図 2-32 に示します。

図 2-31 UAP でのエラー発生時の処理 (全二重の場合)

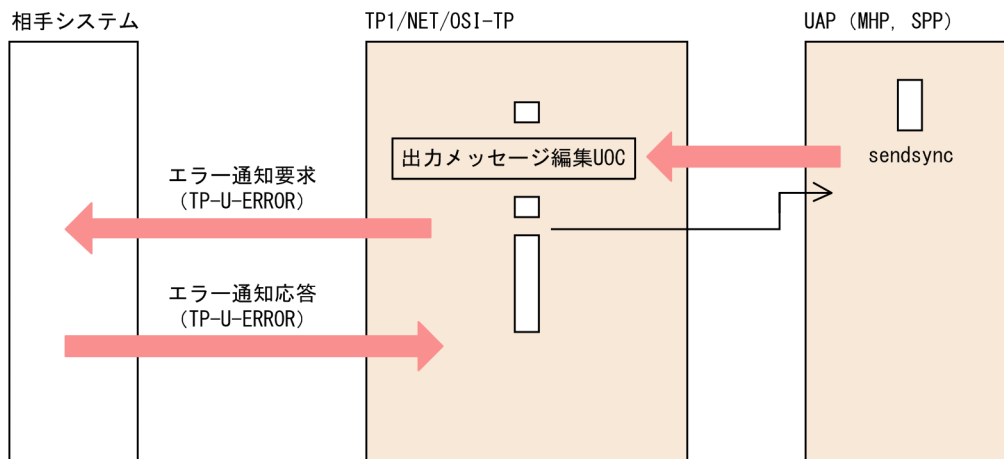
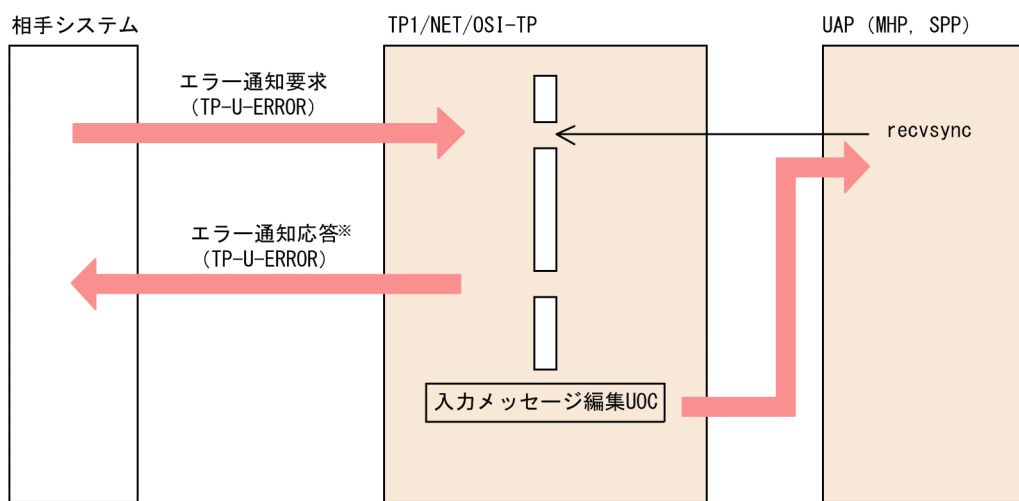


図 2-32 相手システムからのエラー通知受信の処理 (全二重の場合)



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、RCVSYNC 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 2-33 に、相手システムからのエラー受信の処理を図 2-34 に示します。

図 2-33 UAP でのエラー発生時の処理（制御権がある場合）

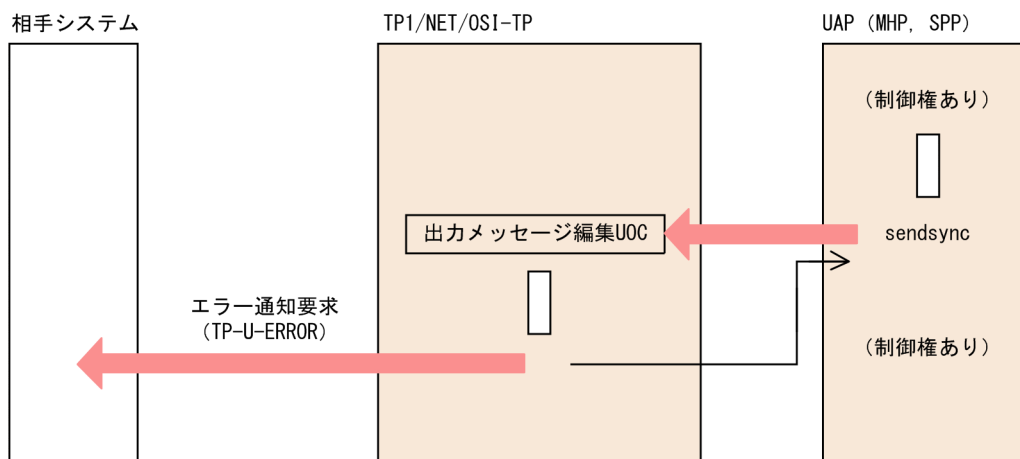
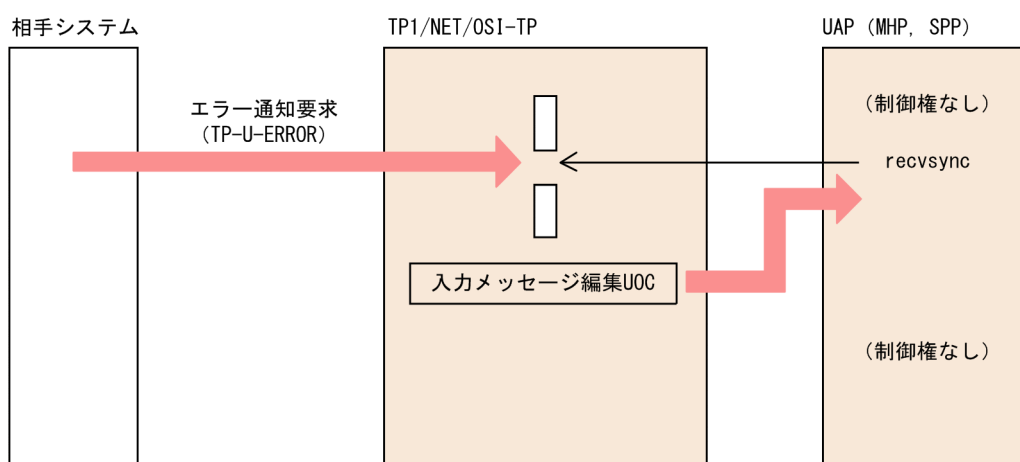


図 2-34 相手システムからのエラー通知受信の処理（制御権がない場合）



## (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 2-35 に、相手システムからのエラー受信の処理を図 2-36 に示します。

図 2-35 UAP でのエラー発生時の処理（制御権がない場合）

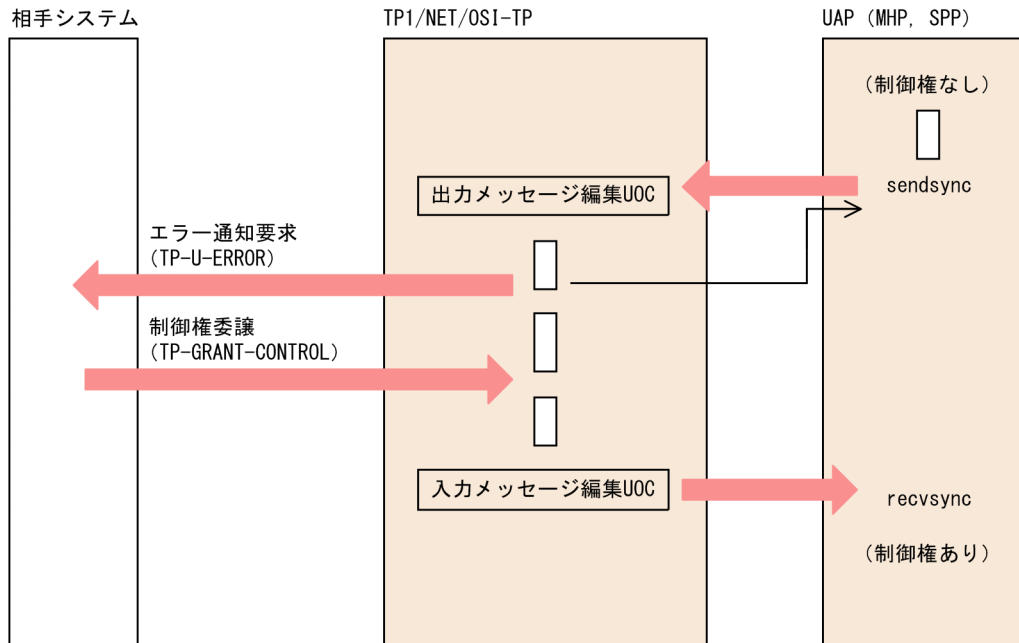
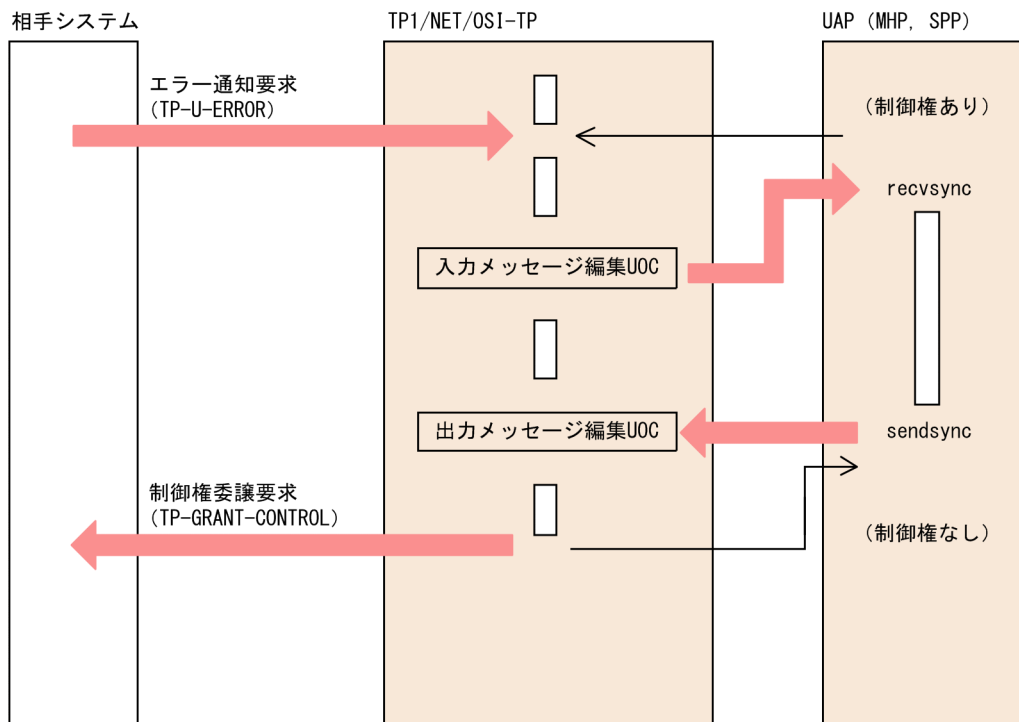


図 2-36 相手システムからのエラー通知受信の処理（制御権がある場合）



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。

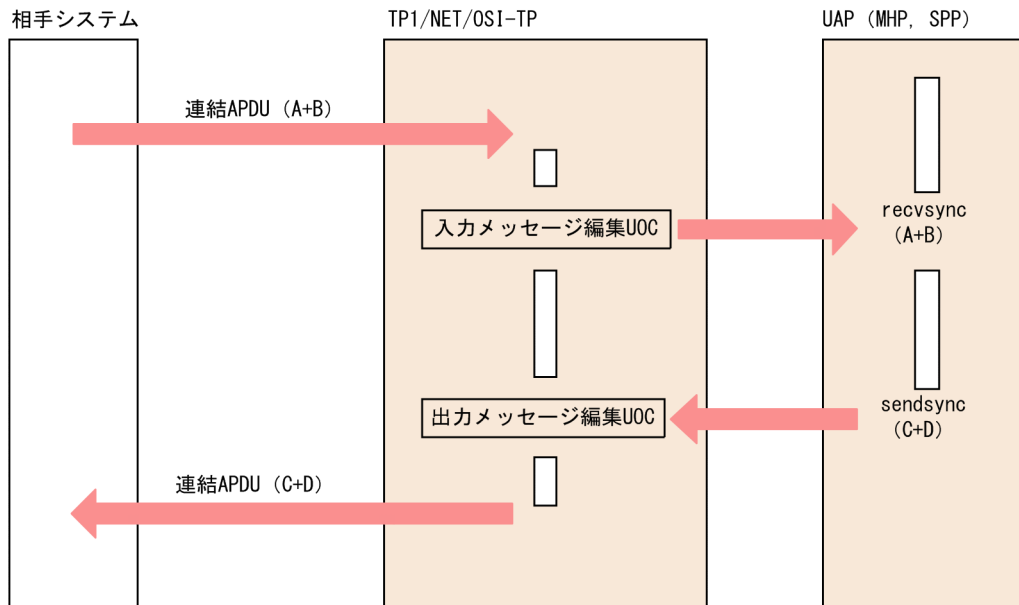
## 2.2.9 APDU 連結

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 2-37 APDU 連結



# 3

## C 言語のライブラリ関数

この章では、TP1/NET/OSI-TP で使用できる、C 言語のライブラリ関数について説明します。

## C 言語のライブラリ関数の一覧

TP1/NET/OSI-TP で使用する C 言語のライブラリ関数の一覧を、次の表に示します。

表 3-1 C 言語のライブラリ関数の一覧

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
dc_mcf_tactcn	接続の確立
dc_mcf_tdctcn	接続の解放
dc_mcf_tlscn	接続の状態取得

なお、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。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。

### NULL またはヌル文字列設定時のコーディング例

C 言語のライブラリ関数の引数に NULL またはヌル文字列を設定する場合のコーディング例を示します。

#### NULL を設定する場合

```
char *resv01=NULL;
dc_mcf_receive(..., resv01, ...);
```

#### ヌル文字列を設定する場合

```
char resv01[1]="\0";
dc_mcf_receive(..., resv01, ...);
```

#### 注

resv01 以外の dc\_mcf\_receive 関数の引数は省略しています。

## 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。

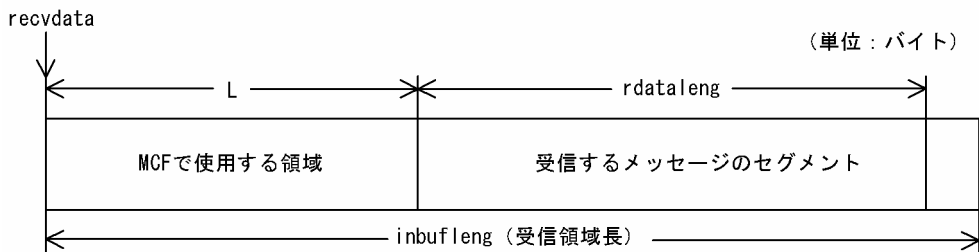
受信できるメッセージの一つのセグメントの最大長は、32000 バイトです。

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、相手システムから受信するメッセージは常に単一セグメントで構成されます。

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



## UAP で値を設定する引数

### ●action

受信するセグメント、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や、論理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

先頭セグメントまたは単一セグメントを受信する場合は、メッセージ入力元の論理端末名称を受け取る領域を設定します。

処理終了後、`termnam` には OpenTP1 から値が返ります。

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、先頭セグメントの受信時に返されたメッセージ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。論理端末名称の最後にはヌル文字を付けてください。

中間セグメントまたは最終セグメントを受信した場合、値は返されません。

### ●resv01

NULL またはヌル文字列を設定します。

## ●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

先頭セグメントまたは単一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。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が付けられます。

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。

### ●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
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71000	-12000	先頭セグメントを受信する dc_mcf_receive 関数を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、action に DCMCFSEG を設定して dc_mcf_receive 関数を呼び出してください。
DCMCFRTN_71001	-12001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する dc_mcf_receive 関数を呼び出しています。直前に呼び出した dc_mcf_receive 関数でメッセージはすべて受信しました。 このリターン値が返されたあとに、再び dc_mcf_receive 関数を呼び出した場合は、リターン値 DCMCFRTN_72000 が返されます。

リターン値	リターン値 (数値)	意味
DCMCFRTN_ 71002	-12002	メッセージキューからの入力処理中に障害が発生しました。
		メッセージキューが閉塞されています。
DCMCFRTN_ 72000	-13000	<p>&lt; MHP の実行でリターンした場合 &gt;</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul>
		<p>&lt; SPP の実行でリターンした場合 &gt;</p> <p>SPP では dc_mcf_receive 関数を呼び出せません。</p>
DCMCFRTN_ 72001	-13001	termnam に設定した論理端末名称が間違っています。
DCMCFRTN_ 72013	-13013	inbufleng の指定値を超えるメッセージ (セグメント) を受信しました。inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRTN_ 72016	-13016	action に設定した値が間違っています。
		resv01 に設定した値が間違っています。
		引数に設定した値が間違っています。
DCMCFRTN_ 72024	-13024	commform に設定した値が間違っています。
DCMCFRTN_ 72025	-13025	action に設定したセグメント種別 (DCMCFRST または DCMCFSEG) の値が間違っています。
DCMCFRTN_ 72036	-13036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
上記以外	—	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

— : 該当しません。

# dc\_mcf\_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)

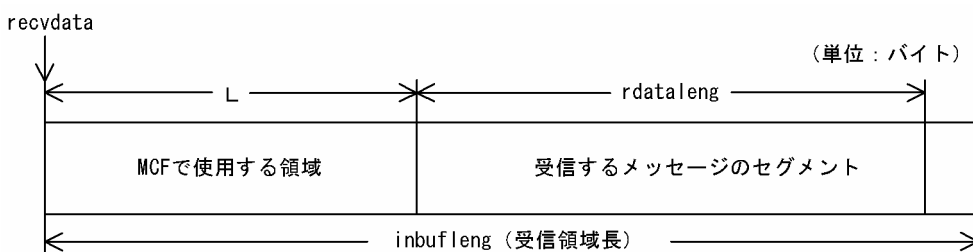
DCLONG      action;
DCLONG      commform;
char        *termnam;
char        *resv01;
char        *recvdata;
DCLONG      *rdataleng;
DCLONG      inbufleng;
DCLONG      *time;
DCLONG      watchtime;
```

## 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。相手システムから受信するメッセージは、常に単一セグメントで構成されます。

受信できるメッセージの一つのセグメントの最大長は、32000 バイトです。

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

受信するセグメント，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 [ | {DCMCFBUF1|DCMCFBUF2} ]
```

### DCMCFRST

単一セグメントを示す DCMCFRST 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。論理端末名称の最後にはヌル文字を付けてください。

### ●resv01

NULL またはヌル文字列を設定します。

### ●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，MCF マネージャ定義の UAP 共通定義で指定した同期型受信監視時間 (mcfmuap -t recvtim) 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。

## 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、設定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、設定した監視時間よりも短い時間で起動することがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は 3 (単位: 秒) 以上の値の設定を推奨します。

## 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。

### ●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
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71001	-12001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する dc_mcf_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_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。
DCMCFRTN_71002	-12002	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。
DCMCFRTN_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。
DCMCFRTN_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_recvsync 関数を呼び出しています。 リターン値 DCMCFRTN_71001 が返されたあとに、再び dc_mcf_recvsync 関数を呼び出しています。
DCMCFRTN_72001	-13001	termnam に設定した論理端末名称が間違っています。 termnam に設定した論理端末名称は、定義されていません。

リターン値	リターン値 (数値)	意味
DCMCFRTN_ 72001	-13001	dc_mcf_recvsync 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_ 72013	-13013	inbufleng の指定値を超えるセグメントを受信しました。 inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRTN_ 72016	-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。
		引数に設定した値が間違っています。
DCMCFRTN_ 72024	-13024	commform に設定した値が間違っています。
DCMCFRTN_ 72025	-13025	action に設定した値が間違っています。
DCMCFRTN_ 72036	-13036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
DCMCFRTN_ 73001	-14001	watchtime に 60 秒を加算した時間が経過しましたが、MCF 通信プロセスからの応答がありません。
		入力元の論理端末で MCF 通信プロセスの内部障害が発生しました。
DCMCFRTN_ 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を受信されません。
DCMCFRTN_ 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。
DCMCFRTN_ 73015	-14015	入力元の論理端末は、ほかの UAP で仕掛り中です。
DCMCFRTN_ 73018	-14018	watchtime に設定した監視時間の値が間違っています。
DCMCFRTN_ 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。
DCMCFRTN_ 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。
DCMCFRTN_ 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。
DCMCFRTN_ 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。
DCMCFRTN_ 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。

リターン値	リターン値 (数値)	意味
DCMCFRTN_ 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。送受信するメッセージは、常に単一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。

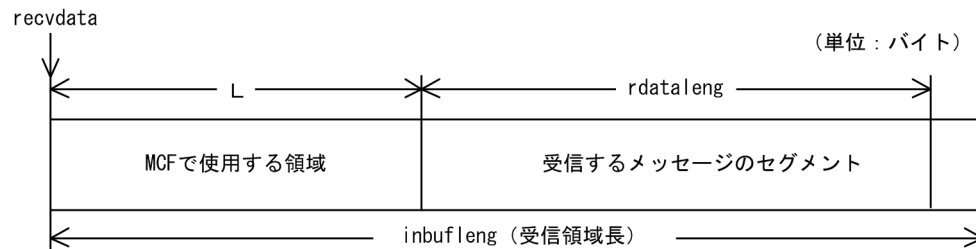
受信できるメッセージの一つのセグメントの最大長は、32000 バイトです。また、送信できるメッセージの一つのセグメントの最大長は、32000 バイトです。

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。

●セグメントを送信する領域



●セグメントを受信する領域



## UAP で値を設定する引数

### ●action

送受信するセグメント、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

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_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

NULL またはヌル文字列を設定します。

### ●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返されます。

処理終了後、recvdata には OpenTP1 から値が返されます。

### ●inbufleng

セグメントを受信する領域の長さを設定します。

### ●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型送受信監視時間 (mcfmuap -t sndrcvtim) が設定されます。

負の値を設定した場合は、時間を監視しません。

## ■ 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、設定した監視時間と実

際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、設定した監視時間よりも短い時間で起動することがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は3（単位：秒）以上の値の設定を推奨します。

## 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。

### ●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
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71002	-12002	メッセージキューへの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。
		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、メッセージの送受信を受け付けられません。	
DCMCFRTN_71003	-12003	メッセージキューが満杯です。
DCMCFRTN_71004	-12004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRTN_71108	-12108	メッセージを送信しようとしたのですが、送信先の管理テーブルが確保できませんでした。
		プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	-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。
DCMCFRTN_72001	-13001	termnam に設定した論理端末名称が間違っています。
		termnam に設定した論理端末名称は、定義されていません。
		dc_mcf_sendrecv 関数を呼び出せない論理端末を設定しています。

リターン値	リターン値 (数値)	意味
DCMCFRTN_ 72013	-13013	inbufleng の指定値を超えるセグメントを受信しました。 inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRTN_ 72016	-13016	action に設定した値が間違っています。
		resv01 に設定した値が間違っています。
		引数に設定した値に間違いがあります。
DCMCFRTN_ 72024	-13024	commform に設定した値が間違っています。
DCMCFRTN_ 72026	-13026	action に設定した値が間違っています。
DCMCFRTN_ 72036	-13036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
DCMCFRTN_ 72041	-13041	sdataleang に 0 バイト、またはマイナス値を設定しています。
DCMCFRTN_ 73001	-14001	watchtime に 60 秒を加算した時間が経過しましたが、MCF 通信プロセスからの応答がありません。
		送信処理中にコネクション障害が発生しました。
		出力先の論理端末で MCF 通信プロセスの内部障害が発生しました。
DCMCFRTN_ 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も、端末からの応答がありません。
DCMCFRTN_ 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。
DCMCFRTN_ 73015	-14015	出力先の論理端末は、ほかの UAP で仕掛り中です。
DCMCFRTN_ 73018	-14018	watchtime に設定した値が間違っています。
DCMCFRTN_ 73020	-14020	メッセージ出力先の論理端末は、停止しています。
DCMCFRTN_ 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。
DCMCFRTN_ 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。
DCMCFRTN_ 77302	-18302	制御情報のサイズの設定が間違っています。

リターン値	リターン値 (数値)	意味
DCMCFRTN_ 77303	-18303	転送手順の設定が間違っています。
DCMCFRTN_ 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。
DCMCFRTN_ 77310	-18310	パラメタ情報サイズの設定が間違っています。
DCMCFRTN_ 77311	-18311	Initiating-TPSU-Title に設定した値が間違っています。
DCMCFRTN_ 77312	-18312	Recipient-AP-Title に設定した値が間違っています。
DCMCFRTN_ 77313	-18313	Recipient-TPSU-Title に設定した値が間違っています。
DCMCFRTN_ 77314	-18314	Functional-Units に設定した値が間違っています。
DCMCFRTN_ 77315	-18315	Quality-of-Service に設定した値が間違っています。
DCMCFRTN_ 77316	-18316	Application-Context-Name に設定した値が間違っています。
DCMCFRTN_ 77317	-18317	Confirmation に設定した値が間違っています。
DCMCFRTN_ 77318	-18318	Result に設定した値が間違っています。
DCMCFRTN_ 77321	-18321	Confirmation-Urgency に設定した値が間違っています。
DCMCFRTN_ 77330	-18330	ユーザデータのサイズが間違っています。
DCMCFRTN_ 77331	-18331	APDU の連結状態が間違っています。
DCMCFRTN_ 77332	-18332	ユーザデータのサイズが設定されていません。
DCMCFRTN_ 77333	-18333	ユーザデータが設定できません。
DCMCFRTN_ 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。
DCMCFRTN_ 77351	-18351	送信バッファが確保できません。

リターン値	リターン値 (数値)	意味
DCMCFRTN_ 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。
DCMCFRTN_ 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。
DCMCFRTN_ 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。
DCMCFRTN_ 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)

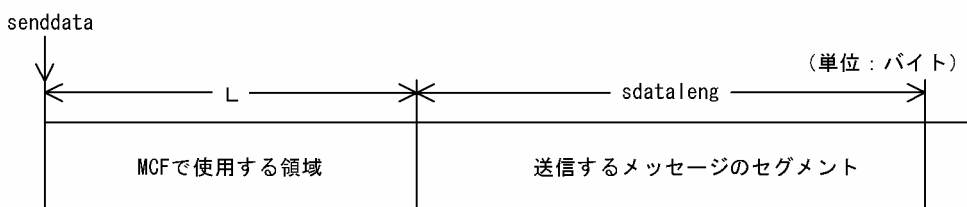
DCLONG      action;
DCLONG      commform;
char        *termnam;
char        *resv01;
char        *senddata;
DCLONG      sdataleng;
char        *resv02;
DCLONG      opcd;
DCLONG      watchtime;
```

## 機能

相手システムへ同期型でメッセージを送信します。送信するメッセージは、常に単一セグメントで構成されます。

送信できるメッセージの一つのセグメントの最大長は、32000 バイトです。

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

送信するセグメント、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

NULL またはヌル文字列を設定します。

## ●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。

## ●resv02

NULL またはヌル文字列を設定します。

## ●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、MCF マネージャ定義の UAP 共通定義で指定した同期型送信監視時間 (mcfmuap -t sndtim) 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。

### 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcftim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、設定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、設定した監視時間よりも短い時間で起動することがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は 3 (単位: 秒) 以上の値の設定を推奨します。

## リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71002	-12002	sdata leng に 32000 バイトを超える値を設定しています。 MCF が終了処理中のため、メッセージの送信を受け付けられません。
DCMCFRTN_71003	-12003	メッセージキューが満杯です。
DCMCFRTN_71004	-12004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRTN_71108	-12108	メッセージを送信しようとしたのですが、送信先の管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	-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。
DCMCFRTN_72001	-13001	termnam に設定した論理端末名称が間違っています。 termnam に設定した論理端末名称は、定義されていません。 dc_mcf_sendsync 関数を呼び出せない論理端末を設定しています。

リターン値	リターン値 (数値)	意味
DCMCFRTN_ 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。
DCMCFRTN_ 72024	-13024	commform に設定した値が間違っています。
DCMCFRTN_ 72026	-13026	action に設定したセグメント種別 (DCMCFEMI) の値が間違っています。
DCMCFRTN_ 72041	-13041	sdata leng に 0 バイト, またはマイナス値を設定しています。
DCMCFRTN_ 73001	-14001	watchtime に 60 秒を加算した時間が経過しましたが, MCF 通信プロセスからの応答がありません。
		送信処理中にコネクション障害が発生しました。
		出力先の論理端末で MCF 通信プロセスの内部障害が発生しました。
DCMCFRTN_ 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。
DCMCFRTN_ 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。
DCMCFRTN_ 73015	-14015	出力先の論理端末は, ほかの UAP で仕掛り中です。
DCMCFRTN_ 73018	-14018	watchtime に設定した監視時間が間違っています。
DCMCFRTN_ 73020	-14020	出力先の論理端末は停止しています。
DCMCFRTN_ 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。
DCMCFRTN_ 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。
DCMCFRTN_ 77302	-18302	制御情報のサイズの設定が間違っています。
DCMCFRTN_ 77303	-18303	転送手順の設定が間違っています。
DCMCFRTN_ 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。

リターン値	リターン値 (数値)	意味
DCMCFRTN_ 77310	-18310	パラメタ情報サイズの設定が間違っています。
DCMCFRTN_ 77311	-18311	Initiating-TPSU-Title に設定した値が間違っています。
DCMCFRTN_ 77312	-18312	Recipient-AP-Title に設定した値が間違っています。
DCMCFRTN_ 77313	-18313	Recipient-TPSU-Title に設定した値が間違っています。
DCMCFRTN_ 77314	-18314	Functional-Units に設定した値が間違っています。
DCMCFRTN_ 77315	-18315	Quality-of-Service に設定した値が間違っています。
DCMCFRTN_ 77316	-18316	Application-Context-Name に設定した値が間違っています。
DCMCFRTN_ 77317	-18317	Confirmation に設定した値が間違っています。
DCMCFRTN_ 77318	-18318	Result に設定した値が間違っています。
DCMCFRTN_ 77321	-18321	Confirmation-Urgency に設定した値が間違っています。
DCMCFRTN_ 77330	-18330	ユーザデータのサイズが間違っています。
DCMCFRTN_ 77331	-18331	APDU の連結状態が間違っています。
DCMCFRTN_ 77332	-18332	ユーザデータのサイズが設定されていません。
DCMCFRTN_ 77333	-18333	ユーザデータが設定できません。
DCMCFRTN_ 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。
DCMCFRTN_ 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。
DCMCFRTN_ 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。
DCMCFRTN_ 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。

リターン値	リターン値 (数値)	意味
DCMCFRTN_ 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\_tactcn – コネクションの確立 (C 言語)

## 形式

### ANSI C, C++の形式

```
#include <dcpcf.h>
int dc_mcf_tactcn (DCLONG action, dcmcf_tactcnopt *cnopt,
                  char *proinf, DCLONG *resv02, char *resv03,
                  char *resv04)
```

### K&R 版 C の形式

```
#include <dcpcf.h>
int dc_mcf_tactcn (action, cnopt, proinf, resv02, resv03, resv04)
DCLONG          action;
dcmcf_tactcnopt *cnopt;
char            *proinf;
DCLONG          *resv02;
char            *resv03;
char            *resv04;
```

## 機能

コネクションを確立します。

なお、dc\_mcf\_tactcn 関数の正常終了は、コネクション確立要求を TP1/NET/OSI-TP が正常に受け付けたことを意味します。このため、相手システムとのコネクションの確立が正常に完了したことを示すものではありません。

dc\_mcf\_tactcn 関数の呼び出し後にコネクションに関する何らかの処理をする場合は、dc\_mcf\_tlscn 関数を用いてコネクションの状態を確認してください。

## UAP で値を設定する引数

### ●action

確立するコネクションの指定方法を次の形式で設定します。

```
{DCMCFLE | DCMFCFN}
```

### DCMCFLE

確立するコネクションを論理端末名称で指定するときに設定します。

### DCMFCFN

確立するコネクションをコネクション ID で指定するときに設定します。

## ●cnopt

この関数の対象となった接続の情報を、構造体 dcmcf\_tactcnopt に設定します。

構造体の形式を次に示します。

```
typedef struct {
    DCLONG    mcfid;           …MCF通信プロセス識別子
    char      resv01[4];       …予備領域
    char      idnam[9];        …論理端末名称, コネクションID
    char      resv02[7];       …予備領域
    char      resv03[112];     …予備領域
    char      scnam[9];        …MCF使用領域
    char      resv04[7];       …予備領域
    char      yournam[9];      …MCF使用領域
    char      resv05[7];       …予備領域
    char      hostnam[143];    …MCF使用領域
    char      resv06[17];     …予備領域
    char      resv07[184];    …予備領域
} dcmcf_tactcnopt;
```

### • mcfid

処理対象の接続を持つ MCF 通信サービスの MCF 通信プロセス識別子<sup>※</sup>を設定します。設定できる範囲は 0~239 です。

論理端末名称を使用して接続の確立を要求する場合は、無効となります。

0 を指定すると、該当する接続 ID が属する MCF 通信サービスを検索します。MCF 通信サービスが多い構成や UAP からこの関数を多数発行する場合は、MCF 通信プロセス識別子の指定をお勧めします。

注※

MCF 環境定義 (mcftenv -s) で指定する MCF 通信プロセス識別子は 16 進数とみなしてください。

例えば、MCF 通信プロセス識別子が 10 の場合、16 を設定してください。

### • resv01

領域をヌル文字で埋めます。

### • idnam

確立する接続の論理端末名称、または接続 ID を設定します。論理端末名称、または接続 ID は最大 8 バイトの長さです。論理端末名称、または接続 ID の最後にはヌル文字を付けてください。

### • resv02, resv03, scnam, resv04, yournam, resv05, hostnam, resv06, resv07

領域をヌル文字で埋めます。

## ●proinf, resv02, resv03, resv04

NULL を設定します。

## リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71001	-12001	MCF が開始処理中のため、dc_mcf_tactcn 関数が受け付けられません。
DCMCFRTN_71002	-12002	MCF が終了処理中のため、dc_mcf_tactcn 関数が受け付けられません。
DCMCFRTN_71004	-12004	dc_mcf_tactcn 関数の処理中にメモリ不足が発生しました。
DCMCFRTN_71005	-12005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71006	-12006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71007	-12007	指定されたコネクション ID は登録されていません。
DCMCFRTN_71008	-12008	指定された論理端末名称は登録されていません。
DCMCFRTN_71009	-12009	dc_mcf_tactcn 関数が、該当する MCF 通信プロセスではサポートされていません。
DCMCFRTN_71010	-12010	MCF 通信プロセスにコネクションの確立を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71011	-12011	コネクションが削除されているため、dc_mcf_tactcn 関数が受け付けられません。
DCMCFRTN_71014	-12014	TP1/NET/NCSB, または TP1/NET/X25-Extended の論理端末名称を指定しています。または TP1/NET/OSI-TP のコネクショングループ名を指定しています。
DCMCFRTN_72050	-13050	action に未サポートのフラグを設定しています。
DCMCFRTN_72051	-13051	cnopt に NULL が設定されています。
DCMCFRTN_72052	-13052	proinf に NULL が設定されていません。
DCMCFRTN_72053	-13053	resv02 に NULL が設定されていません。
DCMCFRTN_72054	-13054	resv03 に NULL が設定されていません。
DCMCFRTN_72055	-13055	resv04 に NULL が設定されていません。
DCMCFRTN_72060	-13060	action には DCMCFLE と DCMCFCN を同時に設定できません。
DCMCFRTN_72061	-13061	dcmcf_tactcnopt の mcfid に 0 未満または 240 以上の値が設定されています。
DCMCFRTN_72062	-13062	dcmcf_tactcnopt の resv01 がヌル文字で埋められていません。
DCMCFRTN_72063	-13063	dcmcf_tactcnopt の idnam の先頭がヌル文字です。

リターン値	リターン値 (数値)	意味
DCMCFRTN_72064	-13064	dcmcf_tactcnopt の resv02 がヌル文字で埋められていません。
DCMCFRTN_72065	-13065	dcmcf_tactcnopt の resv03 がヌル文字で埋められていません。
DCMCFRTN_72066	-13066	dcmcf_tactcnopt の scnnam がヌル文字で埋められていません。
DCMCFRTN_72067	-13067	dcmcf_tactcnopt の resv04 がヌル文字で埋められていません。
DCMCFRTN_72068	-13068	dcmcf_tactcnopt の yournam がヌル文字で埋められていません。
DCMCFRTN_72069	-13069	dcmcf_tactcnopt の resv05 がヌル文字で埋められていません。
DCMCFRTN_72070	-13070	dcmcf_tactcnopt の hostnam がヌル文字で埋められていません。
DCMCFRTN_72071	-13071	dcmcf_tactcnopt の resv06 がヌル文字で埋められていません。
DCMCFRTN_72072	-13072	dcmcf_tactcnopt の resv07 がヌル文字で埋められていません。
DCMCFRTN_72073	-13073	dcmcf_tactleopt の idnam に設定された文字数が9以上です。
DCMCFRTN_72074	-13074	dcmcf_tactcnopt の idnam に設定された文字列中に不正な文字があります。

## dc\_mcf\_tdctcn – コネクションの解放 (C 言語)

### 形式

#### ANSI C, C++の形式

```
#include <dcpcf.h>
int dc_mcf_tdctcn (DCLONG action, dcmcf_tdctcnopt *cnopt,
                  char *proinf, DCLONG *resv02, char *resv03,
                  char *resv04)
```

#### K&R 版 C の形式

```
#include <dcpcf.h>
int dc_mcf_tdctcn (action, cnopt, proinf, resv02, resv03, resv04)
DCLONG          action;
dcmcf_tdctcnopt *cnopt;
char            *proinf;
DCLONG          *resv02;
char            *resv03;
char            *resv04;
```

### 機能

コネクションを解放します。

なお、dc\_mcf\_tdctcn 関数の正常終了は、コネクション解放要求を TP1/NET/OSI-TP が正常に受け付けたことを意味します。このため、相手システムとのコネクションの解放が正常に完了したことを示すものではありません。

dc\_mcf\_tdctcn 関数の呼び出し後にコネクションに関する何らかの処理をする場合は、dc\_mcf\_tlscn 関数を用いてコネクションの状態を確認してください。

### UAP で値を設定する引数

#### ●action

解放するコネクションの指定方法を次の形式で設定します。

```
{DCMCFLE | DCMFCFN} [ | DCMCFFRC]
```

#### DCMCFLE

解放するコネクションを論理端末名称で指定するときに設定します。

#### DCMFCFN

解放するコネクションをコネクション ID で指定するときに設定します。

#### DCMCFFRC

コネクションを強制的に解放するときに設定します。

## ●cnopt

この関数の対象となったコネクションの情報を、構造体 dcmcf\_tdctcnopt に設定します。

構造体の形式を次に示します。

```
typedef struct {
    DCLONG    mcfid;           …MCF通信プロセス識別子
    char      resv01[4];       …予備領域
    char      idnam[9];        …論理端末名称, コネクションID
    char      resv02[7];       …予備領域
    char      resv03[112];     …予備領域
    char      scnam[9];        …MCF使用領域
    char      resv04[7];       …予備領域
    char      resv05[360];     …予備領域
} dcmcf_tdctcnopt;
```

### • mcfid

処理対象のコネクションを持つ MCF 通信サービスの MCF 通信プロセス識別子<sup>※</sup>を設定します。設定できる範囲は 0~239 です。

論理端末名称を使用してコネクションの解放を要求する場合は、無効となります。

0 を指定すると、該当するコネクション ID が属する MCF 通信サービスを検索します。MCF 通信サービスが多い構成や UAP からこの関数を多数発行する場合は、MCF 通信プロセス識別子の指定をお勧めします。

注※

MCF 環境定義 (mcftenv -s) で指定する MCF 通信プロセス識別子は 16 進数とみなしてください。

例えば、MCF 通信プロセス識別子が 10 の場合、16 を設定してください。

### • resv01

領域をヌル文字で埋めます。

### • idnam

解放するコネクションの論理端末名称、またはコネクション ID を設定します。論理端末名称、またはコネクション ID は最大 8 バイトの長さです。論理端末名称、またはコネクション ID の最後にはヌル文字を付けてください。

### • resv02, resv03, scnam, resv04, resv05

領域をヌル文字で埋めます。

## ●proinf, resv02, resv03, resv04

NULL を設定します。

## リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71001	-12001	MCF が開始処理中のため、dc_mcf_tdctcn 関数が受け付けられません。
DCMCFRTN_71002	-12002	MCF が終了処理中のため、dc_mcf_tdctcn 関数が受け付けられません。
DCMCFRTN_71004	-12004	dc_mcf_tdctcn 関数の処理中にメモリ不足が発生しました。
DCMCFRTN_71005	-12005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71006	-12006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71007	-12007	指定されたコネクション ID は登録されていません。
DCMCFRTN_71008	-12008	指定された論理端末名称は登録されていません。
DCMCFRTN_71009	-12009	dc_mcf_tdctcn 関数が、該当する MCF 通信プロセスではサポートされていません。
DCMCFRTN_71010	-12010	MCF 通信プロセスにコネクションの解放を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71011	-12011	コネクションが削除されているため、dc_mcf_tdctcn 関数が受け付けられません。
DCMCFRTN_71014	-12014	TP1/NET/NCSB, または TP1/NET/X25-Extended の論理端末名称を指定しています。または TP1/NET/OSI-TP のコネクショングループ名を指定しています。
DCMCFRTN_72050	-13050	action に未サポートのフラグを設定しています。
DCMCFRTN_72051	-13051	cnopt に NULL が設定されています。
DCMCFRTN_72052	-13052	proinf に NULL が設定されていません。
DCMCFRTN_72053	-13053	resv02 に NULL が設定されていません。
DCMCFRTN_72054	-13054	resv03 に NULL が設定されていません。
DCMCFRTN_72055	-13055	resv04 に NULL が設定されていません。
DCMCFRTN_72060	-13060	action には DCMCFLE と DCMCFCN を同時に設定できません。
DCMCFRTN_72061	-13061	dcmcf_tdctcnopt の mcfid に 0 未満または 240 以上の値が設定されています。
DCMCFRTN_72062	-13062	dcmcf_tdctcnopt の resv01 がヌル文字で埋められていません。
DCMCFRTN_72063	-13063	dcmcf_tdctcnopt の idnam の先頭がヌル文字です。

リターン値	リターン値 (数値)	意味
DCMCFRTN_72064	-13064	dcmcf_tdctcnopt の resv02 がヌル文字で埋められていません。
DCMCFRTN_72065	-13065	dcmcf_tdctcnopt の resv03 がヌル文字で埋められていません。
DCMCFRTN_72066	-13066	dcmcf_tdctcnopt の scnam がヌル文字で埋められていません。
DCMCFRTN_72067	-13067	dcmcf_tdctcnopt の resv04 がヌル文字で埋められていません。
DCMCFRTN_72069	-13069	dcmcf_tdctcnopt の resv05 がヌル文字で埋められていません。
DCMCFRTN_72073	-13073	dcmcf_tdctcnopt の idnam に設定された文字数が9以上です。
DCMCFRTN_72074	-13074	dcmcf_tdctcnopt の idnam に設定された文字列中に不正な文字があります。

## dc\_mcf\_tlscn – コネクションの状態取得 (C 言語)

### 形式

#### ANSI C, C++の形式

```
#include <dcpcf.h>
int dc_mcf_tlscn (DCLONG action, dcmcf_tlscnopt *cnopt,
                 char *resv01, DCLONG *resv02,
                 char *resv03, DCLONG *infcnt,
                 dcmcf_cninf *inf, char *resv04)
```

#### K&R 版 C の形式

```
#include <dcpcf.h>
int dc_mcf_tlscn (action, cnopt, resv01, resv02, resv03, infcnt,
                 inf, resv04)
DCLONG          action;
dcmcf_tlscnopt  *cnopt;
char            *resv01;
DCLONG          *resv02;
char            *resv03;
DCLONG          *infcnt;
dcmcf_cninf     *inf;
char            *resv04;
```

### 機能

コネクションの状態を取得します。

### UAP で値を設定する引数

#### ●action

状態を取得するコネクションの指定方法を次の形式で設定します。

```
{DCMCFLE | DCMCFCN}
```

#### DCMCFLE

状態を取得するコネクションを論理端末名称で指定するときに設定します。

#### DCMCFCN

状態を取得するコネクションをコネクション ID で指定するときに設定します。

#### ●cnopt

この関数の対象となったコネクションの情報を、構造体 dcmcf\_tlscnopt に設定します。

構造体の形式を次に示します。

```

typedef struct {
    DCLONG    mcfid;        …MCF通信プロセス識別子
    char      resv01[4];    …予備領域
    char      idnam[9];    …論理端末名称, コネクションID
    char      resv02[7];    …予備領域
    char      resv03[112]; …予備領域
    char      resv04[376]; …予備領域
} dcmcf_tlscnopt;

```

- **mcfid**

処理対象のコネクションを持つ MCF 通信サービスの MCF 通信プロセス識別子※を設定します。設定できる範囲は 0~239 です。

論理端末名称を使用してコネクションの状態取得を要求する場合は、無効となります。

0 を指定すると、該当するコネクション ID が属する MCF 通信サービスを検索します。MCF 通信サービスが多い構成や UAP からこの関数を多数発行する場合は、MCF 通信プロセス識別子の指定をお勧めします。

注※

MCF 環境定義 (mcftenv -s) で指定する MCF 通信プロセス識別子は 16 進数とみなしてください。

例えば、MCF 通信プロセス識別子が 10 の場合、16 を設定してください。

- **resv01**

領域をヌル文字で埋めます。

- **idnam**

状態を取得するコネクションの論理端末名称、またはコネクション ID を設定します。論理端末名称、またはコネクション ID は最大 8 バイトの長さです。論理端末名称、またはコネクション ID の最後にはヌル文字を付けてください。

- **resv02, resv03, resv04**

領域をヌル文字で埋めます。

- resv01, resv02, resv03**

NULL を設定します。

- infcnt**

コネクション状態を格納する領域 dcmcf\_cninf の個数として、1 を設定します。

処理終了後は、該当するコネクションの個数が返されます。

- inf**

コネクション状態を格納する領域 dcmcf\_cninf を設定します。

「構造体 dcmcf\_cninf のサイズ×infcnt」バイト数分の領域が必要です。

## ●resv04

NULL を設定します。

## OpenTP1 から値が返される引数

### ●infcnt

この関数の対象となった接続の個数が返されます。

### ●inf

この関数の対象となった接続の情報が、構造体 dcmcf\_cninf で返されます。

構造体の形式を次に示します。

```
typedef struct {
    char    idnam[9];      …接続ID
    char    resv01[7];    …予備領域
    char    pnam[4];      …プロトコル種別
    DCLONG  status;       …接続状態
    char    resv02[40];   …予備領域
} dcmcf_cninf;
```

- idnam

要求した接続の接続 ID が設定されます。接続 ID は最大 8 バイトの長さです。接続 ID の最後にはヌル文字が付けられます。

- resv01

領域をヌル文字で埋めます。

- pnam

要求した接続のプロトコル種別が設定されます。プロトコル種別の最後にはヌル文字が付けられます。

TP△

OSI TP プロトコル

- status

要求した接続の状態として、次の値が設定されます。

DCMCF\_CNST\_ACT

確立状態

DCMCF\_CNST\_ACT\_B

確立処理中状態

DCMCF\_CNST\_DCT

解放状態

## DCMCF\_CNST\_DCT\_B

解放処理中状態

- resv02

領域をヌル文字で埋めます。

### リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71001	-12001	MCF が開始処理中のため、dc_mcf_tlscn 関数が受け付けられません。
DCMCFRTN_71004	-12004	dc_mcf_tlscn 関数の処理中にメモリ不足が発生しました。
DCMCFRTN_71005	-12005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71006	-12006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71007	-12007	指定されたコネクション ID は登録されていません。
DCMCFRTN_71008	-12008	指定された論理端末名称は登録されていません。
DCMCFRTN_71009	-12009	dc_mcf_tlscn 関数が、該当する MCF 通信プロセスではサポートされていません。
DCMCFRTN_71010	-12010	MCF 通信プロセスにコネクションの状態取得を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71011	-12011	コネクションが削除されているため、dc_mcf_tlscn 関数が受け付けられません。
DCMCFRTN_71014	-12014	TP1/NET/NCSB, または TP1/NET/X25-Extended の論理端末名称を指定しています。または TP1/NET/OSI-TP のコネクショングループ名を指定しています。
DCMCFRTN_72050	-13050	action に未サポートのフラグを設定しています。
DCMCFRTN_72051	-13051	cnopt に NULL が設定されています。
DCMCFRTN_72052	-13052	resv01 に NULL が設定されていません。
DCMCFRTN_72053	-13053	resv02 に NULL が設定されていません。
DCMCFRTN_72054	-13054	resv03 に NULL が設定されていません。
DCMCFRTN_72055	-13055	resv04 に NULL が設定されていません。
DCMCFRTN_72056	-13056	infcnt に NULL が設定されています。
DCMCFRTN_72057	-13057	inf に NULL が設定されています。
DCMCFRTN_72060	-13060	action には DCMCFLE と DCMCFCN を同時に設定できません。

リターン値	リターン値 (数値)	意味
DCMCFRTN_72061	-13061	dcmcf_tlscnopt の mcfid に 0 未満または 240 以上の値が設定されています。
DCMCFRTN_72062	-13062	dcmcf_tlscnopt の resv01 がヌル文字で埋められていません。
DCMCFRTN_72063	-13063	dcmcf_tlscnopt の idnam の先頭がヌル文字です。
DCMCFRTN_72064	-13064	dcmcf_tlscnopt の resv02 がヌル文字で埋められていません。
DCMCFRTN_72065	-13065	dcmcf_tlscnopt の resv03 がヌル文字で埋められていません。
DCMCFRTN_72067	-13067	dcmcf_tlscnopt の resv04 がヌル文字で埋められていません。
DCMCFRTN_72073	-13073	dcmcf_tlscnopt の idnam に設定された文字数が 9 以上です。
DCMCFRTN_72074	-13074	dcmcf_tlscnopt の idnam に設定された文字列中に不正な文字があります。
DCMCFRTN_72076	-13076	incnt に 1 が設定されていません。

# 4

## COBOL-UAP 作成用プログラムインタフェース

この章では、TP1/NET/OSI-TP で使用できる、COBOL-UAP 作成用プログラムインタフェースについて説明します。

# COBOL-UAP 作成用プログラムインタフェースの一覧

TP1/NET/OSI-TP で使用する COBOL-UAP 作成用プログラムインタフェースについて、COBOL 言語、およびデータ操作言語に分けて説明します。

なお、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。

## COBOL 言語のプログラムインタフェース

COBOL 言語で UAP を作成する場合、OpenTP1 システムの関数に対応しているプログラムを、CALL 文で呼び出して UAP を作成します。

COBOL 言語のプログラムインタフェースの一覧を、次の表に示します。

表 4-1 COBOL 言語のプログラムインタフェースの一覧

プログラム名	データ名 A に設定する要求コード	機能
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
	'TACTCN△△'	コネクションの確立
	'TDCTCN△△'	コネクションの解放
	'TLSCN△△△'	コネクションの状態取得

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

## データ操作言語のプログラムインタフェース

データ操作言語（DML）を使用した、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。

データ操作言語のプログラムインタフェースの一覧を、次の表に示します。

表 4-2 データ操作言語のプログラムインタフェースの一覧

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	メッセージの受信	「表 4-4 RECEIVE 文（データコミュニケーション機能）の指定と C 言語のライブラリ関数との対応」、および「表 4-5 SEND 文（データコミュニ
	SEND	メッセージの送信	

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	SEND	メッセージの送信	ケーション機能) の指定と C 言語のライブラリ関数との対応」を参照してください。

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

## 通信記述項について

TP1/NET/OSI-TP のメッセージ送受信の通信文で、通信記述項に指定できる句の指定要否を、次の表に示します。

表 4-3 通信記述項に指定できる句の指定要否

データ名を指定する句	データ領域の値の設定元						
	1.	2.	3.	4.	5.	6.	7.
STATUS KEY	B	B	B	B	B	B	B
SYMBOLIC TERMINAL	U <sub>1</sub>	U <sub>1</sub>	B	U <sub>1</sub> *	U <sub>1</sub>	U <sub>1</sub>	U <sub>1</sub>
MESSAGE DATE	—	—	B	B	B	—	—
MESSAGE TIME	—	—	B	B	B	—	—
SYNCHRONOUS MODE	—	U <sub>1</sub>	U <sub>2</sub>	U <sub>2</sub>	U <sub>1</sub>	U <sub>1</sub>	U <sub>1</sub>
WAITING TIME	—	—	—	—	U <sub>2</sub>	U <sub>2</sub>	U <sub>2</sub>

(凡例)

1. : 単一セグメントの同期送信 (DISABLE)
  2. : 単一セグメントの同期送信 (ENABLE)
  3. : 先頭セグメントの非同期受信 (RECEIVE)
  4. : 中間, 最終セグメントの非同期受信 (RECEIVE)
  5. : 単一セグメントの同期受信 (RECEIVE)
  6. : 単一セグメントの同期送信 (SEND)
  7. : 単一セグメントの同期送受信 (SEND)
- B : OpenTP1 から値が返されます。省略できます。  
U<sub>1</sub> : UAP で値を設定します。省略できません。  
U<sub>2</sub> : UAP で値を設定します。省略できます。  
— : 該当しません。設定しても無効です。

注※

先頭メッセージ受信時の RECEIVE 文と同一の CD 句を用いた場合は省略できます。

## データコミュニケーション機能と C 言語のライブラリ関数の対応

RECEIVE 文 (データコミュニケーション機能) の指定と C 言語のライブラリ関数との対応を、次の表に示します。

表 4-4 RECEIVE 文（データコミュニケーション機能）の指定と C 言語のライブラリ関数との対応

FOR 句		SYNCHRONOUS MODE 句		対応する C 言語のライブラリ関数
INPUT	I-O	SYNC, または'1'	ASYN, '0', '△', または省略	
○	—	—	○	dc_mcf_receive
—	○	—	○	
—	○	○	—	dc_mcf_recvsync

(凡例)

○：指定あり

—：指定なし

SEND 文（データコミュニケーション機能）の指定と C 言語のライブラリ関数との対応を、次の表に示します。

表 4-5 SEND 文（データコミュニケーション機能）の指定と C 言語のライブラリ関数との対応

FOR 句		SYNCHRONOUS MODE 句		BEFORE 句	対応する C 言語のライブラリ関数
OUTPUT	I-O	SYNC, または'1'	ASYN, '0', '△', または省略		
○	—	—	○	—	dc_mcf_send*
—	○	—	○	—	dc_mcf_reply*
—	○	○	—	—	dc_mcf_sendsync
—	○	○	—	○	dc_mcf_sendrecv

(凡例)

○：指定あり

—：指定なし

注※

TP1/NET/OSI-TP ではサポートしていない関数です。

# 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 CBLDMCF('RECEIVE△') を呼び出すと、一つの論理メッセージを受信できます。

受信できるメッセージの一つのセグメントの最大長は、32000 バイトです。

CBLDMCF('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

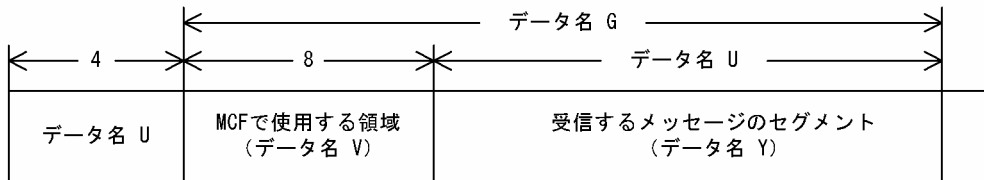
- MCF イベント
- アプリケーション起動で渡されたメッセージ

TP1/NET/OSI-TP を使用して通信する場合、相手システムから受信するメッセージは、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。

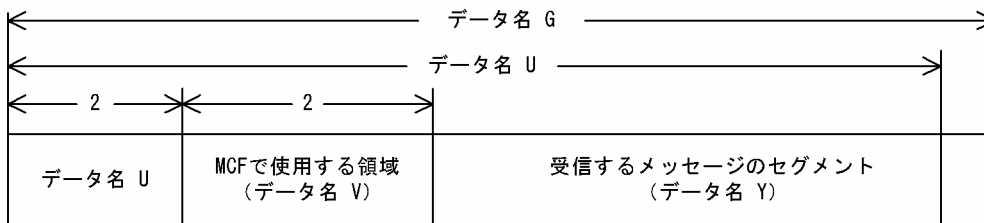
●バッファ形式 1 の場合

(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

受信するセグメントを設定します。次のどちらかを設定します。

#### VALUE 'FRST'

先頭セグメントを受信する場合や、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。

### ●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は、入力元の論理端末名称を設定します。先頭セグメントの受信時に返された論理端末名称を設定してください。論理端末名称は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称の後ろを空白で埋めてください。

先頭セグメントまたは単一セグメントの受信処理終了後、データ名 P には OpenTP1 から値が返されます。

●データ名 Q, 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

先頭セグメントまたは単一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。論理端末名  
称は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称の後ろが空白で埋められます。

中間セグメントまたは最終セグメントを受信する場合、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。
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。

ステータスコード	意味
71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する CBLDCMCF('RECEIVE△')を呼び出しています。直前に呼び出した CBLDCMCF('RECEIVE△')でメッセージはすべて受信しました。 このステータスコードが返されたあとに、再び CBLDCMCF('RECEIVE△')を呼び出した場合は、ステータスコード 72000 が返されます。
71002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。
71108	プロセスのローカルメモリが不足しています。
72000	<p>&lt; MHP の実行でリターンした場合 &gt;</p> <ul style="list-style-type: none"> <li>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。</li> <li>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。</li> </ul> <p>&lt; SPP の実行でリターンした場合 &gt;</p> <p>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。
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).
```

### 注※

負の値を設定する場合、「PIC S9(9) COMP」としてください。

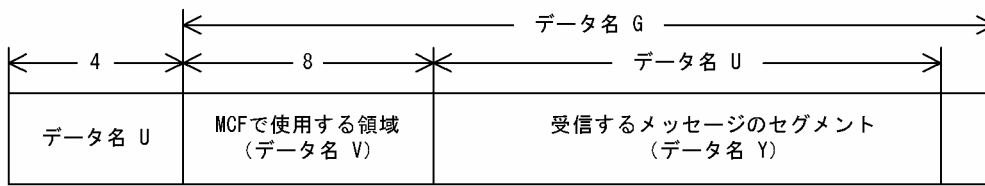
## 機能

相手システムから同期型でメッセージを受信します。相手システムから受信するメッセージは、常に単一セグメントで構成されます。

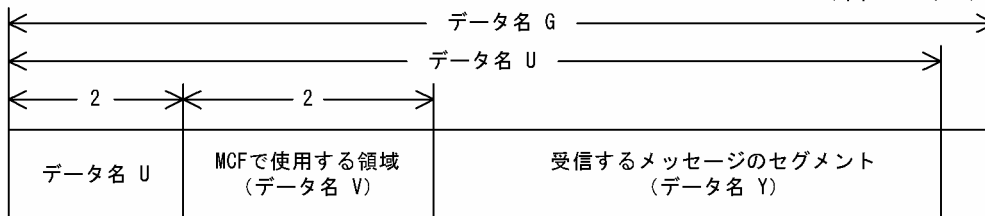
受信できるメッセージの一つのセグメントの最大長は、32000 バイトです。

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

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

先頭セグメントまたは単一セグメントの受信を示す「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、MCF マネージャ定義の UAP 共通定義で指定した同期型受信監視時間 (mcfmuap -t recvtim) 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。

## 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、設定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、設定した監視時間よりも短い時間で起動することがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は 3 (単位: 秒) 以上の値の設定を推奨します。

### ●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

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 バイトに満たない場合、論理端末名称の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。

### ●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。

## 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('RECVSYNC')を呼び出しています。 直前に呼び出した CBLDCMCF('RECVSYNC')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('RECVSYNC')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	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	先頭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 が返されたあとに、再び CBLDCMCF('RECVSYNC')を呼び出していません。

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。
	データ名 P に設定した論理端末名称は、定義されていません。
	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。
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	データ名 M5 に 60 秒を加算した時間が経過しましたが、MCF 通信プロセスからの応答がありません。
	入力元の論理端末で MCF 通信プロセスの内部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).
```

### 注※

負の値を設定する場合、「PIC S9(9) COMP」としてください。

## 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。送受信するメッセージは、常に単一セグメントで構成されます。

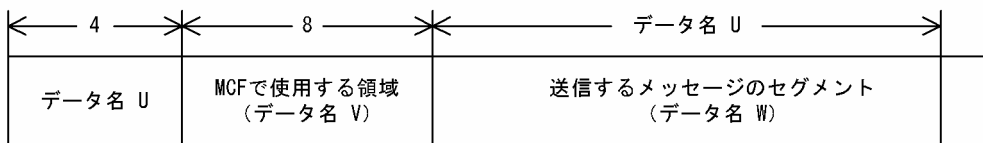
メッセージを受信すると、CBLDCMCF('SENDRECV')は相手システムからの応答を待ちます。応答が届くと、そのメッセージを受信します。

受信できるメッセージの一つのセグメントの最大長は、32000 バイトです。また、送信できるメッセージの一つのセグメントの最大長は、32000 バイトです。

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。

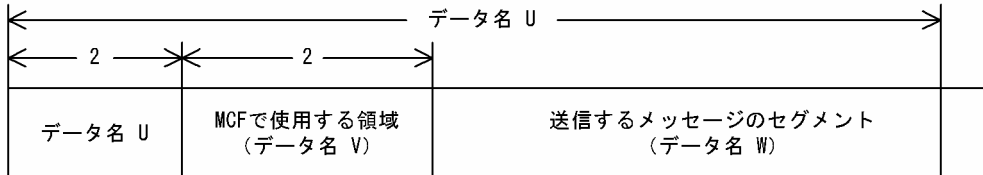
### ●バッファ形式 1 の場合

(単位: バイト)



### ●バッファ形式 2 の場合

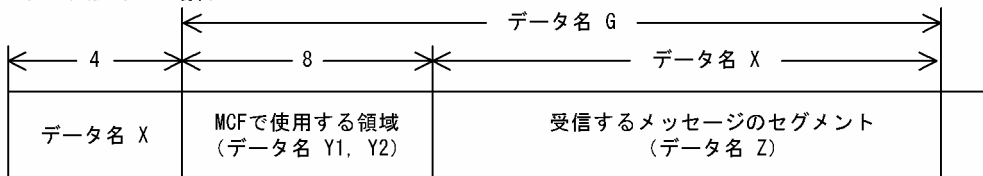
(単位: バイト)



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。

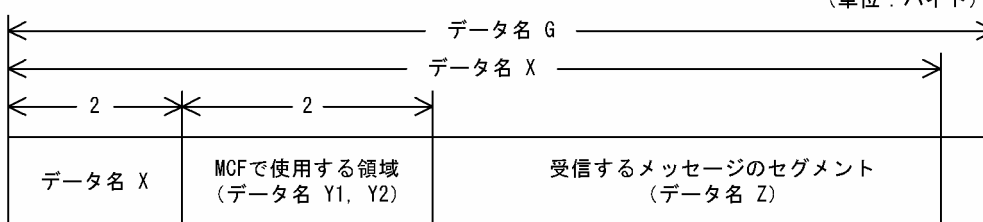
### ●バッファ形式 1 の場合

(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 'SENDRECV」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 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('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、同期型送受信監視時間 (mcfmuap -t sndrcvtim) が設定されます。

負の値を設定した場合は、時間を監視しません。

### ■ 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、設定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、設定した監視時間よりも短い時間で起動することがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は 3 (単位：秒) 以上の値の設定を推奨します。

## ●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 'IO△△」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 バイトに満たない場合、論理端末名称の後ろを空白で埋めてください。

#### ●データ名 Q, データ名 R

空白を設定します。

#### ●データ名 T

MCF で使用する領域です。

#### ●データ名 U

【バッファ形式 1 の場合】 PIC 9(9)

送信するセグメントの長さを設定します。

【バッファ形式 2 の場合】 PIC 9(4)

送信するセグメントの長さ + 4 を設定します。

#### ●データ名 V

【バッファ形式 1 の場合】 PIC X(8)

【バッファ形式 2 の場合】 PIC X(2)

MCF で使用する領域です。

#### ●データ名 W

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

#### ●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。

## 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。

## 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。
	バッファ形式 1 の場合はデータ名 U に 32000 バイトを超える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 32004 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のですが、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。

ステータスコード	意味
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。
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 に設定した論理端末名称は、定義されていません。
	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。
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 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。
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 バイト、またはマイナス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 0 から 4 バイト、またはマイナス値を設定しています。
73001	データ名 M5 に 60 秒を加算した時間が経過しましたが、MCF 通信プロセスからの応答がありません。
	送信処理中にコネクション障害が発生しました。
	出力先の論理端末で MCF 通信プロセスの内部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。

ステータスコード	意味
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。
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).
```

### 注※

負の値を設定する場合、「PIC S9(9) COMP」としてください。

## 機能

相手システムへ同期型でメッセージを送信します。送信するメッセージは、常に単一セグメントで構成されます。

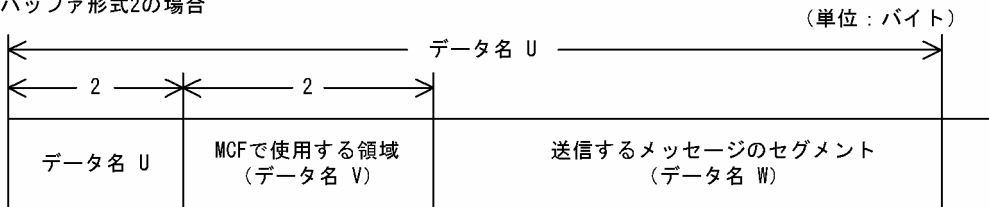
送信できるメッセージの一つのセグメントの最大長は、32000 バイトです。

セグメントを送信する領域（一意名 3 で示す領域）の形式を次に示します。

●バッファ形式1の場合



●バッファ形式2の場合



## UAP で値を設定するデータ領域

●データ名 A

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、MCF マネージャ定義の UAP 共通定義で指定した同期型送信監視時間 (mcfmuap -t sndtim) 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。

## 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、設定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、設定した監視時間よりも短い時間で起動することがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は 3 (単位: 秒) 以上の値の設定を推奨します。

### ●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

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 バイトに満たない場合、論理端末名称の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)

## 【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

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

## OpenTP1 から値が返されるデータ領域

### ●データ名 B

ステータスコードが、5 けたの数字で返されます。

## ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	バッファ形式 1 の場合はデータ名 U に 32000 バイトを超える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 32004 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のですが、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 に設定した論理端末名称は、定義されていません。 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。

ステータスコード	意味
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 バイト、またはマイナス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 0 から 4 バイト、またはマイナス値を設定しています。
73001	データ名 M5 に 60 秒を加算した時間が経過しましたが、MCF 通信プロセスからの応答がありません。
	送信処理中に接続障害が発生しました。
	出力先の論理端末で MCF 通信プロセスの内部障害が発生しました。
73002	同期送信時、メッセージ送信先の論理端末と TP1/NET/OSI-TP との間の接続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 の間の接続が切断されました。
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	指定した接続グループ内に使用できる接続がありません。
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。

ステータスコード	意味
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。
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('TACTCN ') - コネクションの確立 (COBOL 言語)

## 形式

### PROCEDURE DIVISION の指定

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

### DATA DIVISION の指定

```
01 一意名1.  
02 データ名A PIC X(8) VALUE 'TACTCN '.  
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D1 PIC X(1) VALUE SPACE.  
02 データ名D2 PIC X(1) VALUE SPACE.  
02 データ名D3 PIC X(26) VALUE SPACE.  
02 データ名E PIC 9(9) COMP.  
02 データ名F1 PIC X(8).  
02 データ名F2 PIC X(56) VALUE SPACE.  
02 データ名G PIC X(8) VALUE SPACE.  
02 データ名H PIC X(8) VALUE SPACE.  
02 データ名I PIC X(144) VALUE SPACE.  
02 データ名J PIC X(184) VALUE SPACE.  
02 データ名K PIC 9(9) COMP VALUE ZERO.  
  
01 一意名2.  
02 データ名L PIC 9(9) COMP VALUE ZERO.
```

## 機能

コネクションを確立します。

なお、CBLDCMCF('TACTCN△△')の正常終了は、コネクション確立要求を TP1/NET/OSI-TP が正常に受け付けたことを意味します。このため、相手システムとのコネクションの確立が正常に完了したことを示すものではありません。

CBLDCMCF('TACTCN△△')の呼び出し後にコネクションに関する何らかの処理をする場合は、CBLDCMCF('TLSCN△△△')を用いてコネクションの状態を確認してください。

## UAP で値を設定するデータ領域

### ●データ名 A

コネクション確立を示す要求コード「VALUE 'TACTCN△△'」を設定します。

### ●データ名 C

確立するコネクションの指定方法を設定します。

VALUE 'LE△△'

確立するコネクションを論理端末名称で指定するときに設定します。

VALUE 'CN△△'

確立するコネクションをコネクション ID で指定するときに設定します。

空白

省略されたものとして、「VALUE 'LE△△」(論理端末名称を指定)が設定されます。

### ●データ名 D1, データ名 D2, データ名 D3

空白を設定します。

### ●データ名 E

処理対象のコネクションを持つ MCF 通信サービスの MCF 通信プロセス識別子<sup>※</sup>を設定します。設定できる範囲は、0~239 です。

論理端末名称を使用してコネクションの確立を要求する場合は、無効となります。

0 を指定すると、該当するコネクション ID が属する MCF 通信サービスを検索します。MCF 通信サービスが多い構成や UAP からこの命令文を多数発行する場合は、MCF 通信プロセス識別子の指定をお勧めします。

注※

MCF 環境定義 (mcftenv -s) で指定する MCF 通信プロセス識別子は 16 進数とみなしてください。

例えば、MCF 通信プロセス識別子が 10 の場合、16 を設定してください。

### ●データ名 F1

確立するコネクションの論理端末名称、またはコネクション ID を設定します。論理端末名称、またはコネクション ID は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称、またはコネクション ID の後ろを空白で埋めてください。

### ●データ名 F2, データ名 G, データ名 H, データ名 I, データ名 J

空白を設定します。

### ●データ名 K, データ名 L

0 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。
71001	MCF が開始処理中のため、CBLDCMCF('TACTCN△△')が受け付けられません。
71002	MCF が終了処理中のため、CBLDCMCF('TACTCN△△')が受け付けられません。
71004	CBLDCMCF('TACTCN△△')の処理中にメモリ不足が発生しました。
71005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
71006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
71007	指定されたコネクション ID は登録されていません。
71008	指定された論理端末名称は登録されていません。
71009	CBLDCMCF('TACTCN△△')が、該当する通信プロセスではサポートされていません。
71010	MCF 通信プロセスにコネクションの確立を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
71011	コネクションが削除されているため、CBLDCMCF('TACTCN△△')が受け付けられません。
71014	TP1/NET/NCSB, または TP1/NET/X25-Extended の論理端末名称を指定しています。または TP1/NET/OSI-TP のコネクショングループを指定しています。
72028	データ名 A に設定した値が間違っています。
72052	データ名 K に 0 でない値が設定されています。
72053	データ名 L に 0 でない値が設定されています。
72058	データ名 C に 'LE△△', 'CN△△', または空白以外の値が設定されています。
72059	データ名 D1, データ名 D2, またはデータ名 D3 に空白でない値が設定されています。
72061	データ名 E に 0 未満, または 240 以上の値が設定されています。
72063	データ名 F1 に空白が設定されています。
72065	データ名 F2 に空白でない値が設定されています。
72066	データ名 G に空白でない値が設定されています。
72068	データ名 H に空白でない値が設定されています。
72070	データ名 I に空白でない値が設定されています。
72072	データ名 J に空白でない値が設定されています。
72074	データ名 F1 に設定された文字列中に不正な文字があります。

# CBLDCMCF('TDCTCN ') – コネクションの解放 (COBOL 言語)

## 形式

### PROCEDURE DIVISION の指定

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

### DATA DIVISION の指定

```
01 一意名1.  
  02 データ名A PIC X(8) VALUE 'TDCTCN'.  
  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D1 PIC X(1).  
  02 データ名D2 PIC X(1) VALUE SPACE.  
  02 データ名D3 PIC X(26) VALUE SPACE.  
  02 データ名E PIC 9(9) COMP.  
  02 データ名F1 PIC X(8).  
  02 データ名F2 PIC X(56) VALUE SPACE.  
  02 データ名G PIC X(8) VALUE SPACE.  
  02 データ名H PIC X(8) VALUE SPACE.  
  02 データ名I PIC X(144) VALUE SPACE.  
  02 データ名J PIC X(184) VALUE SPACE.  
  02 データ名K PIC 9(9) COMP VALUE ZERO.  
  
01 一意名2.  
  02 データ名L PIC 9(9) COMP VALUE ZERO.
```

## 機能

コネクションを解放します。

なお、CBLDCMCF('TDCTCN△△')の正常終了は、コネクション解放要求を TP1/NET/OSI-TP が正常に受け付けたことを意味します。このため、相手システムとのコネクションの解放が正常に完了したことを示すものではありません。

CBLDCMCF('TDCTCN△△')の呼び出し後にコネクションに関する何らかの処理をする場合は、CBLDCMCF('TLSCN△△△')を用いてコネクションの状態を確認してください。

## UAP で値を設定するデータ領域

### ●データ名 A

コネクション解放を示す要求コード「VALUE 'TDCTCN△△」を設定します。

### ●データ名 C

解放するコネクションの指定方法を設定します。

VALUE 'LE△△'

解放するコネクションを論理端末名称で指定するときに設定します。

VALUE 'CN△△'

解放するコネクションをコネクション ID で指定するときに設定します。

空白

省略されたものとして、「VALUE 'LE△△」(論理端末名称を指定)が設定されます。

### ●データ名 D1

解放するコネクションの指定方法を設定します。

VALUE '1'

コネクションを強制的に解放します。

VALUE '0'

コネクションを正常に解放します。

空白

省略されたものとして、'0' (正常解放) が設定されます。

### ●データ名 D2, データ名 D3

空白を設定します。

### ●データ名 E

処理対象のコネクションを持つ MCF 通信サービスの MCF 通信プロセス識別子<sup>※</sup>を設定します。設定できる範囲は 0~239 です。

論理端末名称を使用してコネクションの解放を要求する場合は、無効となります。

0 を指定すると、該当するコネクション ID が属する MCF 通信サービスを検索します。MCF 通信サービスが多い構成や UAP からこの命令文を多数発行する場合は、MCF 通信プロセス識別子の指定をお勧めします。

注※

MCF 環境定義 (mcftenv -s) で指定する MCF 通信プロセス識別子は 16 進数とみなしてください。

例えば、MCF 通信プロセス識別子が 10 の場合、16 を設定してください。

### ●データ名 F1

解放するコネクションの論理端末名称、またはコネクション ID を設定します。論理端末名称、またはコネクション ID は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称、またはコネクション ID の後ろを空白で埋めてください。

### ●データ名 F2, データ名 G, データ名 H, データ名 I, データ名 J

空白を設定します。

## ●データ名 K, データ名 L

0 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。
71001	MCF が開始処理中のため、CBLDCMCF('TDCTCN△△')が受け付けられません。
71002	MCF が終了処理中のため、CBLDCMCF('TDCTCN△△')が受け付けられません。
71004	CBLDCMCF('TDCTCN△△')の処理中にメモリ不足が発生しました。
71005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
71006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
71007	指定されたコネクション ID は登録されていません。
71008	指定された論理端末名称は登録されていません。
71009	CBLDCMCF('TDCTCN△△')が、該当する通信プロセスではサポートされていません。
71010	MCF 通信プロセスにコネクションの解放を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
71011	コネクションが削除されているため、CBLDCMCF('TDCTCN△△')が受け付けられません。
71014	TP1/NET/NCSB, または TP1/NET/X25-Extended の論理端末名称を指定しています。または TP1/NET/OSI-TP のコネクショングループ名を指定しています。
72028	データ名 A に設定した値が間違っています。
72052	データ名 K に 0 でない値が設定されています。
72053	データ名 L に 0 でない値が設定されています。
72058	データ名 C に 'LE△△', 'CN△△', または空白以外の値が設定されています。
72059	データ名 D2, またはデータ名 D3 に空白でない値が設定されています。
72061	データ名 E に 0 未満または 240 以上の値が設定されています。
72063	データ名 F1 に空白が設定されています。
72065	データ名 F2 に空白でない値が設定されています。
72066	データ名 G に空白でない値が設定されています。
72068	データ名 H に空白でない値が設定されています。

ステータスコード	意味
72070	データ名 I に空白でない値が設定されています。
72072	データ名 J に空白でない値が設定されています。
72074	データ名 F1 に設定された文字列中に不正な文字があります。
72075	データ名 D1 に 1, 0, または空白以外の値が設定されています。

# CBLDCMCF('TLSCN ') - コネクションの状態取得 (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 'TLSCN ' .  
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(28) VALUE SPACE.  
02 データ名E PIC 9(9) COMP.  
02 データ名F1 PIC X(8).  
02 データ名F2 PIC X(56) VALUE SPACE.  
02 データ名G PIC X(8) VALUE SPACE.  
02 データ名H PIC X(8) VALUE SPACE.  
02 データ名I PIC X(144) VALUE SPACE.  
02 データ名J PIC X(184) VALUE SPACE.  
02 データ名K PIC 9(9) COMP VALUE ZERO.  
  
01 一意名2.  
02 データ名L PIC 9(9) COMP VALUE ZERO.  
  
01 一意名3.  
02 データ名M PIC 9(9) COMP.  
02 一意名4.  
03 データ名N PIC X(8).  
03 データ名O PIC X(4).  
03 データ名P PIC X(4).  
03 データ名Q PIC X(40) VALUE LOW-VALUE.
```

## 機能

コネクションの状態を取得します。

### UAP で値を設定するデータ領域

#### ●データ名 A

コネクション状態取得を示す要求コード「VALUE 'TLSCN△△△」を設定します。

#### ●データ名 C

状態を取得するコネクションの指定方法を設定します。

VALUE 'LE△△'

状態を取得するコネクションを論理端末名称で指定するときに設定します。

VALUE 'CN△△'

状態を取得するコネクションをコネクション ID で指定するときに設定します。

空白

省略されたものとして、「VALUE 'LE△△」(論理端末名称を指定)が設定されます。

### ●データ名 D

空白を設定します。

### ●データ名 E

処理対象のコネクションを持つ MCF 通信サービスの MCF 通信プロセス識別子<sup>※</sup>を設定します。設定できる範囲は 0~239 です。

論理端末名称を使用してコネクションの状態取得を要求する場合は、無効となります。

0 を指定すると、該当するコネクション ID が属する MCF 通信サービスを検索します。MCF 通信サービスが多い構成や UAP からこの命令文を多数発行する場合は、MCF 通信プロセス識別子の指定をお勧めします。

注※

MCF 環境定義 (mcftenv -s) で指定する MCF 通信プロセス識別子は 16 進数とみなしてください。

例えば、MCF 通信プロセス識別子が 10 の場合、16 を設定してください。

### ●データ名 F1

状態を取得するコネクションの論理端末名称、またはコネクション ID を設定します。論理端末名称、またはコネクション ID は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称、またはコネクション ID の後ろを空白で埋めてください。

### ●データ名 F2, データ名 G, データ名 H, データ名 I, データ名 J

空白を設定します。

### ●データ名 K, データ名 L

0 を設定します。

### ●データ名 M

一意名 4 から一意名 n の数 (データ名 N, データ名 O, データ名 P, およびデータ名 Q の組の数) として、1 を設定します。

処理終了後は、該当するコネクションの個数が返されます。

### ●データ名 Q

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 M

この命令文の対象となった接続の個数が返されます。

### ●データ名 N

要求した接続の接続 ID が設定されます。8 バイトに満たない場合、接続 ID の後ろが空白で埋められます。

### ●データ名 O

要求した接続のプロトコル種別が設定されます。

VALUE 'TP△△'

OSI TP プロトコル

### ●データ名 P

要求した接続の状態として、次の値が設定されます。

VALUE 'ACT△'

確立状態

VALUE 'ACTB'

確立処理中状態

VALUE 'DCT△'

解放状態

VALUE 'DCTB'

解放処理中状態

## 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	MCF が開始処理中のため、CBLDCMCF('TLSCN△△△')が受け付けられません。
71004	CBLDCMCF('TLSCN△△△')の処理中にメモリ不足が発生しました。
71005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
71006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
71007	指定された接続 ID は登録されていません。

ステータスコード	意味
71008	指定された論理端末名称は登録されていません。
71009	CBLDCMCF('TLSCN△△△')が、該当する通信プロセスではサポートされていません。
71010	MCF 通信プロセスに接続の状態取得を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
71011	接続が削除されているため、CBLDCMCF('TLSCN△△△')が受け付けられません。
71014	TP1/NET/NCSB, または TP1/NET/X25-Extended の論理端末名称を指定しています。または TP1/NET/OSI-TP の接続グループ名を指定しています。
72028	データ名 A に設定した値が間違っています。
72052	データ名 K に 0 でない値が設定されています。
72053	データ名 L に 0 でない値が設定されています。
72058	データ名 C に 'LE△△', 'CN△△', または空白以外の値が設定されています。
72059	データ名 D に空白でない値が設定されています。
72061	データ名 E に 0 未満または 240 以上の値が設定されています。
72063	データ名 F1 に空白が設定されています。
72065	データ名 F2 に空白でない値が設定されています。
72066	データ名 G に空白でない値が設定されています。
72068	データ名 H に空白でない値が設定されています。
72070	データ名 I に空白でない値が設定されています。
72072	データ名 J に空白でない値が設定されています。
72074	データ名 F1 に設定された文字列中に不正な文字があります。
72076	データ名 M に 1 以外の値が設定されています。

# 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-0
  [STATUS KEY IS データ名1]
SYMBOLIC TERMINAL IS データ名2.
```

### DATA DIVISION（データ記述項）の指定

```
01 一意名1.
02 データ名A PIC 9(4) COMP.
02 データ名B PIC X(2).
02 データ名C PIC X(n).
```

### PROCEDURE DIVISION（通信文）の指定

```
DISABLE 通信記述名 WITH 一意名1.
```

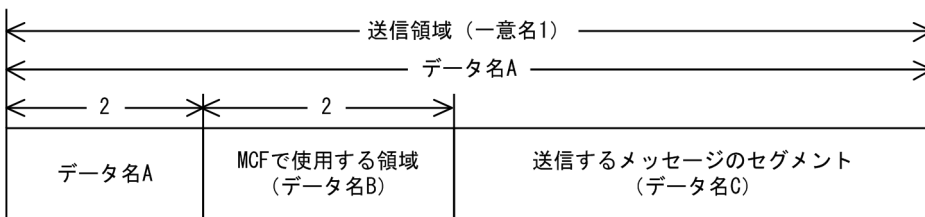
## 機能

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')

セグメントを送信する領域（一意名 1 で示す領域）の形式を次に示します。

（単位：バイト）



## UAP で値を設定する項目

### ●FOR 句

次の値を設定します。

### I-O

同期型メッセージの送信

## ●SYMBOLIC TERMINAL 句

論理端末名称を設定したデータ項目を設定します。データ名 2 に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 バイトに満たない場合、論理端末名称の後ろを空白で埋めてください。

## ●データ名 A

送信するセグメントの長さ + 4 を設定します。

## ●データ名 B

MCF で使用する領域です。

## ●データ名 C

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

## OpenTP1 から値が返される項目

### ●STATUS KEY 句

ステータスコードを受け取りたい場合に設定します。省略した場合は、ステータスコードを受け取りません。データ名 1 にステータスコードが返されます。

## ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	データ名 A に 32004 バイトを超える値を設定しています。 MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたが、送信先の管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
72000	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	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。 SYMBOLIC TERMINAL 句に設定した論理端末名称は、定義されていません。
72014	DISABLE 文を実行できない論理端末を設定しています。
72024	FOR 句に設定した値が間違っています。
72041	データ名 A に、0 から 4 バイト、またはマイナス値を設定しています。
73001	同期型送信監視時間に 60 秒を加算した時間が経過しましたが、MCF 通信プロセスからの応答がありません。

ステータスコード	意味
73001	送信処理中にコネクション障害が発生しました。
	出力先の論理端末で MCF 通信プロセスの内部障害が発生しました。
73005	同期型送信監視時間に指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。
73015	出力先論理端末は、ほかの UAP で仕掛り中です。
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。
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-0
  [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} .
```

### DATA DIVISION（データ記述項）の指定

```
01 一意名1.
02 データ名A PIC 9(4) COMP.
02 データ名B PIC X(2).
02 データ名C PIC X(n).
```

### PROCEDURE DIVISION（通信文）の指定

```
ENABLE 通信記述名 WITH 一意名1.
```

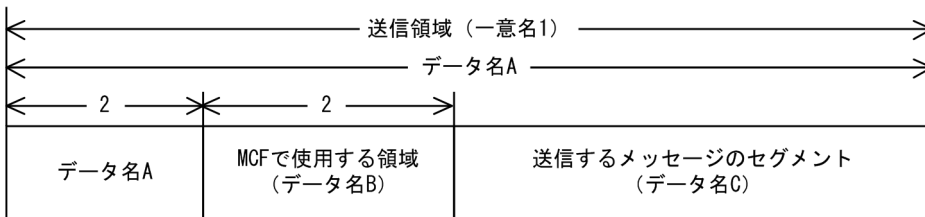
## 機能

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')

セグメントを送信する領域（一意名 1 で示す領域）の形式を次に示します。

（単位：バイト）



## UAP で値を設定する項目

### ●FOR 句

次の値を設定します。

### I-O

同期型メッセージの送信

## ●SYMBOLIC TERMINAL 句

論理端末名称を設定したデータ項目を設定します。データ名 2 に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 バイトに満たない場合、論理端末名称の後ろを空白で埋めてください。

## ●SYNCHRONOUS MODE 句

次のどちらかの値を設定します。

### SYNC

同期型メッセージの送信

### データ名 3

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

## ●データ名 A

送信するセグメントの長さ + 4 を設定します。

## ●データ名 B

MCF で使用する領域です。

## ●データ名 C

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

## OpenTP1 から値が返される項目

## ●STATUS KEY 句

ステータスコードを受け取りたい場合に設定します。省略した場合は、ステータスコードを受け取りません。データ名 1 にステータスコードが返されます。

## ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	データ名 A に 32004 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のですが、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。

ステータスコード	意味
72000	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	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。
	SYMBOLIC TERMINAL 句に設定した論理端末名称は、定義されていません。
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。
72041	データ名 A に、0 から 4 バイト、またはマイナス値を設定しています。
73001	同期型送信監視時間に 60 秒を加算した時間が経過しましたが、MCF 通信プロセスからの応答がありません。
	送信処理中にコネクション障害が発生しました。
	出力先の論理端末で MCF 通信プロセスの内部障害が発生しました。
73005	同期型送信監視時間に指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。
73015	出力先論理端末は、ほかの UAP で仕掛り中です。
73016	ENABLE 文を実行できない論理端末を設定しています。
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。

ステータスコード	意味
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-0}
  [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|ASYN|データ名6} ]
  [WAITING TIME IS データ名11] .
```

### DATA DIVISION (データ記述項) の指定

```
01 一意名1.
02 データ名A PIC 9(4) COMP.
02 データ名B PIC X(2).
02 データ名C PIC X(n).
```

### PROCEDURE DIVISION (通信文) の指定

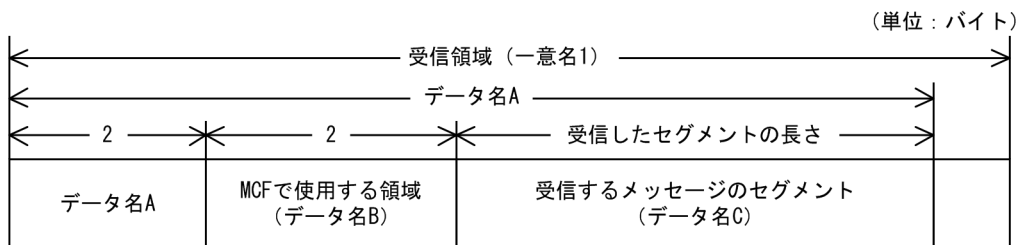
```
RECEIVE 通信記述名
  [FIRST] SEGMENT
  INTO 一意名1.
```

## 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')

セグメントを受信する領域 (一意名 1 で示す領域) の形式を次に示します。



## UAP で値を設定する項目

### ●FOR 句

次のどちらかの値を設定します。

## 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

### ●SYMBOLIC TERMINAL 句

非同期型メッセージ（中間セグメントまたは最終セグメント）の受信，または，同期型メッセージの受信の場合，データ名 2 に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。論理端末名称は最大 8 バイトの長さです。8 バイトに満たない場合，論理端末名称の後ろを空白で埋めてください。

非同期型メッセージ（先頭セグメント）の受信処理終了後，SYMBOLIC TERMINAL 句には OpenTP1 から値が返されます。

### ●SYNCHRONOUS MODE 句

非同期型でメッセージを受信するか，同期型でメッセージを受信するかを設定します。

## SYNC

同期型メッセージの受信

同期型メッセージの受信のとき設定します。

## ASYNC

非同期型メッセージの受信

一方送信メッセージの受信のとき設定します。

## 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

'0'または'△'：非同期型メッセージの受信

省略した場合は，ASYNC（非同期型メッセージの受信）が設定されます。

### ●WAITING TIME 句

RECEIVE 文を実行してから終了するまでの，最大時間を設定したデータ項目を設定します。同期型メッセージを受信する場合に設定します。

## データ名 11

監視時間を HHMMSS00（HH：時 MM：分 SS：秒 00 は固定）の形式で設定したデータ項目を設定します。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。

## 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、設定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、設定した監視時間よりも短い時間で起動することがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は 3 秒以上の値の設定を推奨します。

### ●データ名 B

MCF で使用する領域です。

### ●FIRST

先頭セグメントを受信する場合に設定します。

## OpenTP1 から値が返される項目

### ●STATUS KEY 句

ステータスコードを受け取りたい場合に設定します。省略した場合は、ステータスコードを受け取りません。データ名 1 にステータスコードが返されます。

### ●SYMBOLIC TERMINAL 句

非同期型メッセージ (先頭セグメント) の受信の場合、入力元の論理端末名称を受け取りたいときに設定します。省略した場合は、論理端末名称を受け取れません。データ名 2 にメッセージ入力元の論理端末名称が返されます。論理端末名称は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称の後ろが空白で埋められます。

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 SYMBOLIC TERMINAL 句に設定します。

### ●MESSAGE DATE 句

メッセージを受信した日付を受け取りたい場合に設定します。省略した場合は、メッセージを受信した日付を受け取れません。データ名 3 にメッセージを受信した日付が YYMMDD (YY: 西暦下 2 けた MM: 月 DD: 日) の形式で返されます。

### ●MESSAGE TIME 句

メッセージを受信した時刻を受け取りたい場合に設定します。省略した場合は、メッセージを受信した時刻を受け取れません。データ名 4 に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 A

受信したセグメントの長さ + 4 が返されます。

## ●データ名 C

受信したセグメントの内容が返されます。

## ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する 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"><li>非同期型メッセージの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 を設定して RECEIVE 文を実行してください。</li><li>ステータスコード 71001 が返されたあとで、再び RECEIVE 文を実行しています。</li></ul>
	< SPP の実行でリターンした場合 > SPP では非同期型メッセージを受信する 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。
	SYMBOLIC TERMINAL 句に設定した論理端末名称は、定義されていません。
	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 のサイズを超えるセグメントを受信しました。 一意名 1 のサイズを超える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	データ名 C のサイズが不足しています。5 バイト以上の領域を確保してください。
73001	データ名 11 に 60 秒を加算した時間が経過しましたが、MCF 通信プロセスからの応答がありません。

ステータスコード	意味
73001	入力元の論理端末で MCF 通信プロセスの内部障害が発生しました。
73005	データ名 11 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の編集 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。
73018	データ名 11 に設定した監視時間が間違っています。
73020	RECEIVE 文で設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 - 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-0
   [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] .
```

### DATA DIVISION (データ記述項) の指定

```
01 一意名1.
   02 データ名A PIC 9(4) COMP.
   02 データ名B PIC X(2).
   02 データ名C PIC X(n).
01 一意名3.
   02 データ名D PIC 9(4) COMP.
   02 データ名E PIC X(2).
   02 データ名F PIC X(n).
```

### PROCEDURE DIVISION (通信文) の指定

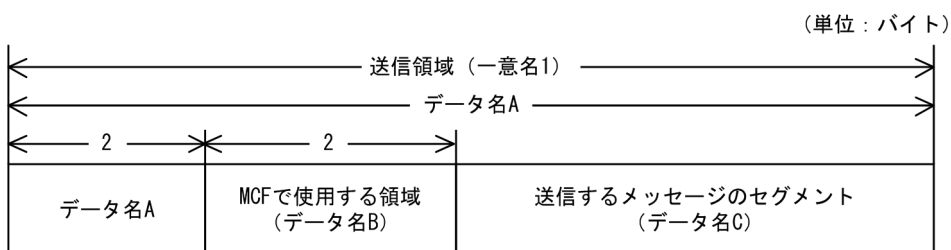
```
SEND 通信記述名 FROM 一意名1
     [WITH {EMI | 一意名2}]
     [BEFORE RECEIVING MESSAGE INTO 一意名3] .
```

## 機能

次に示す CALL インタフェースの機能を実現します。

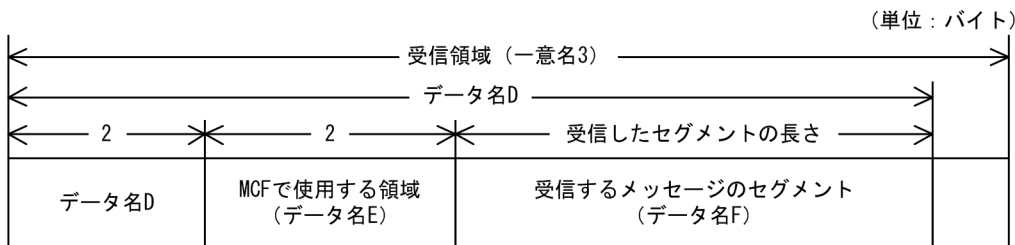
- 同期型メッセージの送信 CBLDCMCF('SENDSYNC')
- 同期型メッセージの送受信 CBLDCMCF('SENDRECV')

セグメントを送信する領域 (一意名 1 で示す領域) の形式を次に示します。



同期型メッセージの送受信の場合、セグメントを受信する領域 (一意名 3 で示す領域) も設定します。

セグメントを受信する領域の形式を次に示します。



## UAP で値を設定する項目

### ●FOR 句

次の値を設定します。

#### 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

### ●SYMBOLIC TERMINAL 句

論理端末名称を設定したデータ項目を設定します。データ名 2 に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 バイトに満たない場合, 論理端末名称の後ろを空白で埋めてください。

### ●SYNCHRONOUS MODE 句

次のどちらかの値を設定します。

#### SYNC

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

#### 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, または同期型メッセージの送受信

### ●WAITING TIME 句

SEND 文を実行してから終了するまでの, 最大時間を設定したデータ項目を設定します。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に設定します。

#### データ名 11

監視時間の値を, HHMMSS00 (HH:時 MM:分 SS:秒 00 は固定) の形式で設定したデータ項目を設定します。

省略した場合, またはデータ名 11 に'00000000'を設定した場合, 同期型メッセージの送信と, 同期型メッセージの送受信とは設定される値が次のとおり異なります。

- 同期型メッセージの送信の場合

MCF マネジャ定義の UAP 共通定義で指定した同期型送信監視時間 (mcfmuap -t sndtim) が設定されます。

- 同期型メッセージの送受信の場合

MCF マネジャ定義の UAP 共通定義で指定した同期型送受信監視時間 (mcfmuap -t sndrcvtim) が設定されます。

## ■ 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、設定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、設定した監視時間よりも短い時間で起動することがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は 3 秒以上の値の設定を推奨します。

## ●データ名 A

送信するセグメントの長さ + 4 を設定します。

## ●データ名 B

MCF で使用する領域です。

## ●データ名 C

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

## ●データ名 E

MCF で使用する領域です。

## ●WITH 句

次のどちらかの値を設定します。

EMI

単一セグメント

## 一意名 2

次の値を設定したデータ項目

'2': EMI (単一セグメント)

省略した場合は、EMI (単一セグメント) が設定されます。

## ●BEFORE 句

同期型メッセージ送受信の場合に、セグメントを受信するデータ項目 (一意名 3) を設定します。同期型メッセージの送信の場合、BEFORE 句を省略します。

## OpenTP1 から値が返される項目

### ●STATUS KEY 句

ステータスコードを受け取りたい場合に設定します。省略した場合は、ステータスコードを受け取れません。データ名 1 にステータスコードが返されます。

### ●データ名 D

受信したセグメントの長さ + 4 が返されます。

### ●データ名 F

受信したセグメントの内容が返されます。

## ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	データ名 A に 32004 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	SYNCHRONOUS MODE 句が省略されています。
	SYNCHRONOUS MODE 句に ASYNC を設定しています。
	SYNCHRONOUS MODE 句に 0 または空白を設定したデータ名 6 を設定しています。
	< 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。
72001	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。
	SYMBOLIC TERMINAL 句に設定した論理端末名称は、定義されていません。
	SEND 文を実行できない論理端末を設定しています。
72013	データ名 F のサイズを超えるセグメントを受信しました。 データ名 F のサイズ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。

ステータスコード	意味
72024	FOR 句に設定した値が間違っています。
72036	データ名 F のサイズが不足しています。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。
72041	データ名 A に 0 から 4 バイト、またはマイナス値を設定しています。
72045	継続問い合わせ応答型のアプリケーションはサポートしていません。
72047	継続問い合わせ応答型のアプリケーションはサポートしていません。
73001	データ名 11 に 60 秒を加算した時間が経過しましたが、MCF 通信プロセスからの応答がありません。
	送信処理中にコネクション障害が発生しました。
	出力先の論理端末で MCF 通信プロセスの内部障害が発生しました。
73005	データ名 11 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。

ステータスコード	意味
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。

# 5

## ユーザOWNコーディング，MCF イベントインタフェース

この章では，TP1/NET/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。

## 5.1 ユーザOWNコーディングインタフェース

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

TP1/NET/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。

### 5.1.1 入力メッセージの編集とアプリケーション名の決定

入力メッセージ編集 UOC は、受信した論理メッセージの編集をする 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、必要に応じてコネクション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、およびコネクション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 バイトの識別子です。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、「[8.2 MCF メイン関数の作成](#)」を参照してください。

## 5.1.2 入力メッセージ編集 UOC インタフェース

入力メッセージ編集 UOC は、次に示す形式で呼び出します。

### (1) 形式

ANSI C, C++の場合

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

K&R 版 C の場合

```
#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
    DCULONG buf_size;      ...バッファ最大長
    DCULONG seg_size;      ...バッファ使用長
    char reserve1[4];
    dcmcfuoc_w_type buff_id; ...MCF内部情報1
    DCMLONG 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、コネクション定義 (mcftalccn) 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  
論理端末定義 (mcftalcle) 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 の数だけ設定されます。

### (c) dcmcf\_uocbufinf\_n (バッファ情報)

- buf\_adr

バッファのアドレスが設定されます。

- `buf_size`

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

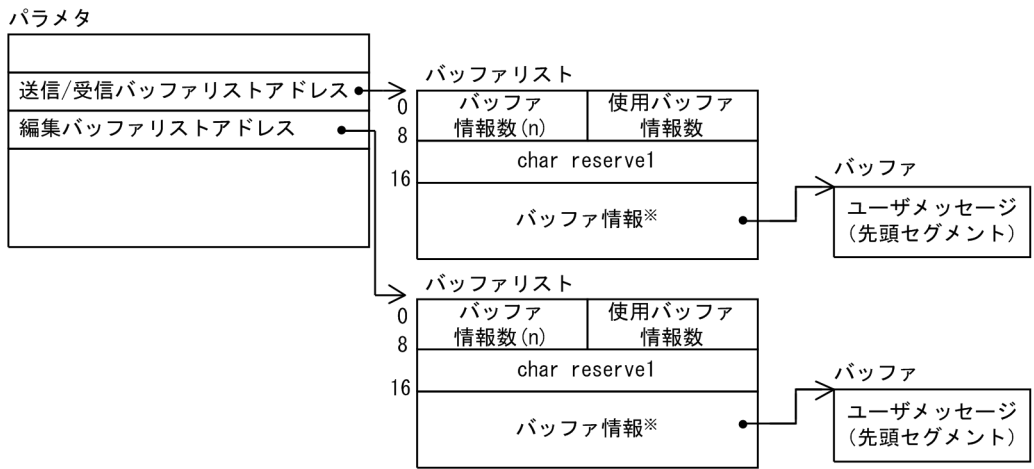
`uoc_func()` は次のコードでリターンしてください。

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

## (7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係を次の図に示します。

図 5-1 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

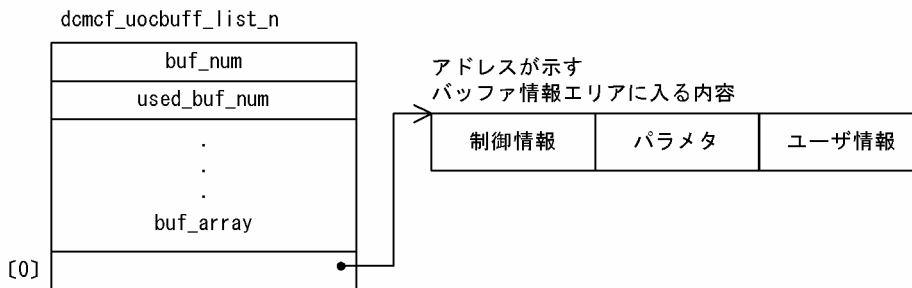
入力メッセージ編集 UOC を使用する場合、APDU 連結があるかどうかによって、バッファのインタフェースが異なります。

### (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。

図 5-2 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)

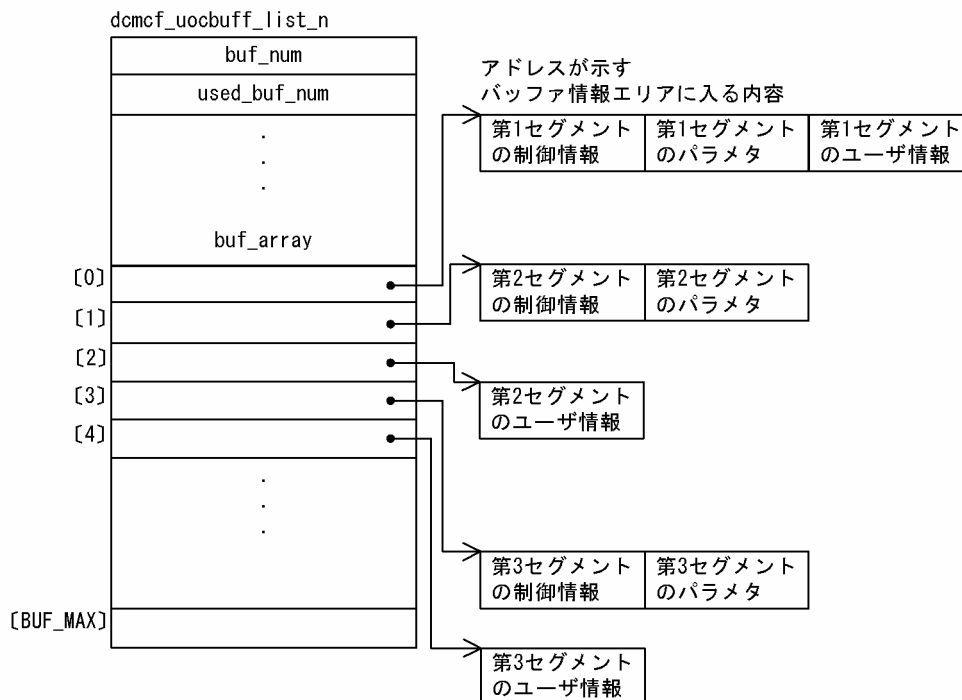


**(b) APDU 連結がある場合**

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。

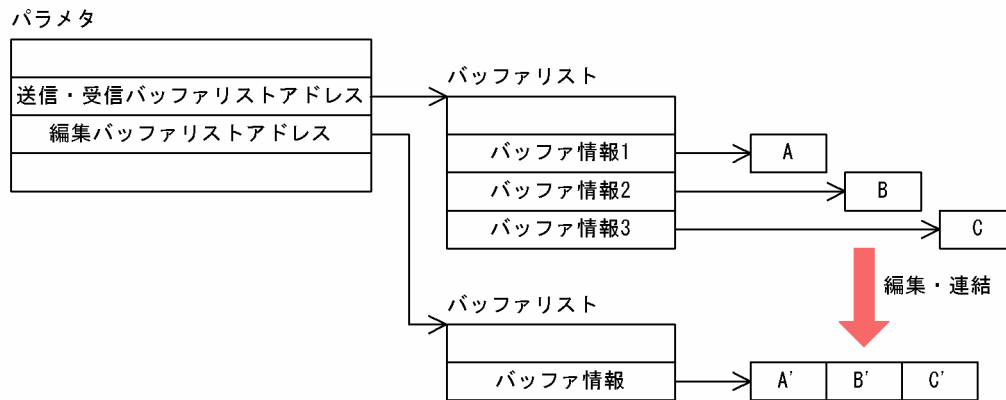
APDU 連結がある場合のバッファインタフェースを次の図に示します。

図 5-3 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 5-4 受信メッセージの連結



### 5.1.3 出力メッセージの編集

出力メッセージの編集 UOC は、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、必要に応じてコネクション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、およびコネクション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 (CERREVT) を通知します。該当するメッセージは破棄します。

## (4) OpenTP1 への組み込み方法

入力メッセージ編集 UOC の組み込み方法と同じです。「5.1.1(5) OpenTP1 への組み込み方法」を参照してください。

### 5.1.4 出力メッセージ編集 UOC インタフェース

出力メッセージ編集 UOC は、次に示す形式で呼び出します。

#### (1) 形式

ANSI C, C++の場合

```
#include <dcmcf.h>
#include <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。  
 「5.1.2(3)(b) dcmcf\_uocbuff\_list\_n (バッファリスト) の内容」および「5.1.2(3)(c) dcmcf\_uocbufinf\_n (バッファ情報) の内容」を参照してください。

## (4) MCF が値を設定する項目

### (a) dcmcf\_uoc\_mout\_n

- pro\_kind  
 プロトコル種別として、次の値が設定されます。  
 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、コネクション定義 (mcftalccn) 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。  
 '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。

## 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。

「5.1.2(4)(b) dcmcf\_uocbuff\_list\_n (バッファリスト)」および「5.1.2(4)(c) dcmcf\_uocbufinf\_n (バッファ情報)」を参照してください。

## (5) ユーザが値を設定する項目

### (a) dcmcf\_uoc\_mout\_n

- **rtn\_detail**

詳細リターンコードを設定します。

このコードは、UOC が DCMCF\_UOC\_MSG\_NG をリターンしたときに、MCF に渡されます。

MCF は、詳細リターンコードをメッセージログファイルに出力します。

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

### (b) dcmcf\_uocbuff\_list\_n (バッファリスト), dcmcf\_uocbufinf\_n (バッファ情報)

入力メッセージ編集 UOC インタフェースのバッファリストおよびバッファ情報の内容と同じです。

「5.1.2(5)(b) dcmcf\_uocbuff\_list\_n (バッファリスト)」および「5.1.2(5)(c) dcmcf\_uocbufinf\_n (バッファ情報)」を参照してください。

## (6) リターン値

uoc\_func()は次のコードでリターンしてください。

リターン値	意味
DCMCF_UOC_MSG_OK	正常リターン (編集バッファでスケジューリング)
DCMCF_UOC_MSG_OK_SND	正常リターン (送信バッファでスケジューリング)

リターン値	意味
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。  
 「5.1.2(7) パラメタとバッファの関係」を参照してください。

## (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

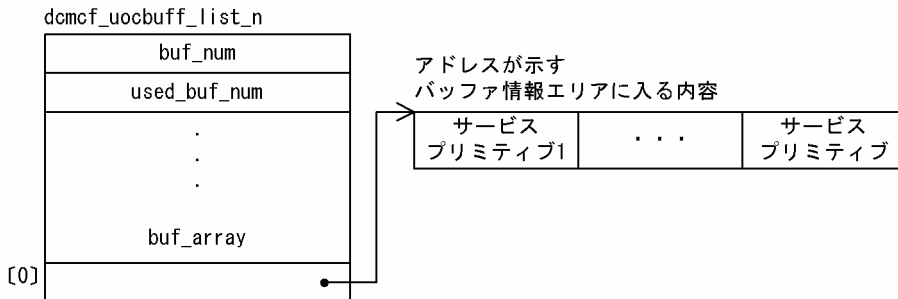
APDU 連結がない場合、入力メッセージ編集 UOC と同様に、セグメント数は 1 になります。ユーザセグメントは、バッファ情報エリアの 1 番目を使用されます。「5.1.2(8) バッファインタフェース」を参照してください。

### (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 番目を使用されます。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 5-5 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)



## 5.1.5 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 の中で関数の再呼び出しはしないでください。

## (2) UOC で使用できる関数

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

- メモリ操作をする関数
  - データ領域管理 (例: malloc, free)
  - 共有メモリ管理関数/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。

## 5.2 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。

### 5.2.1 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 5-1 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"><li>入力キューに障害が発生しました。</li><li>入力メッセージ最大格納数を超過しました。</li><li>動的共用メモリが不足しました。</li><li>キューファイルが満杯になりました。</li><li>MHP のサービス、サービスグループ、またはアプリケーションが閉塞しています。</li><li>スケジュール閉塞されているサービスグループの入力キューに未処理受信メッセージが残った状態で、OpenTP1 を正常終了または計画停止 A で終了しました。</li><li>MHP のサービスグループ、またはアプリケーションがセキュア状態です。</li><li>MHP で呼び出す receive 関数にセグメントを渡す前に、MHP が異常終了しました。</li><li>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。</li><li>ユーザサーバ未起動などによって、MHP の起動に失敗しました。</li><li>DBMS の障害などによって、トランザクションの開始に失敗しました。</li></ul>	メッセージを廃棄したことを報告します。

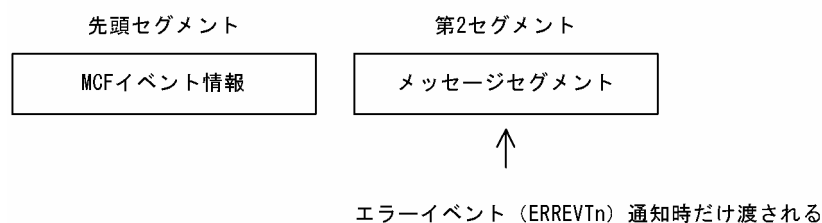
MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP での処理の例
UAP 異常終了通知イベント	ERREVT3	MHP の先頭セグメントの受信文に、セグメントを渡したあとに、MHP の異常終了が発生しました。	UAP 異常終了時の対処障害メッセージを送信します。
タイマ起動メッセージ廃棄通知イベント	ERREVT4	アプリケーションのタイマ起動時に障害が発生しました。	メッセージを廃棄したことを報告します。
未処理送信メッセージ廃棄通知イベント	ERREVT4	MCF の正常終了処理時の、未処理送信メッセージの滞留時間監視の時間切れ（タイムアウト）でメッセ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。
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。

## 5.2.2 MCF イベント通知時のセグメント構成

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

MCF イベント通知時のセグメント構成を次の図に示します。

図 5-6 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。「5.2.4 MCF イベント情報の形式 (COBOL 言語)」のエラーイベントの表では ERREVT1, ERREVT2, ERREVT3, および 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。

## 5.2.3 MCF イベント情報の形式 (C 言語)

MCF イベント情報は構造体で、MCF イベント処理用 MHP に渡されます。MHP に渡される構造体の形式は、MCF イベントの種類によって異なります。ただし、MCF イベント情報の先頭部分 (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。<dcmotm.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
                                   (ERREVTa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。

ただし、ERREVT2 および ERREVT3 で、次に示す場合は、'\*'が設定されます。

- SPP からアプリケーション起動機能で起動した MHP で、障害が発生した場合
- 上記の障害が発生したあとに、MCF イベントとして起動した MHP からさらにアプリケーション起動機能で起動した MHP で、障害が発生した場合

ERREVTa 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。

ただし、ERREVT2 および ERREVT3 で、次に示す場合は、'\*'が設定されます。

- SPP からアプリケーション起動機能で起動した MHP で、障害が発生した場合
- 上記の障害が発生したあとに、MCF イベントとして起動した MHP からさらにアプリケーション起動機能で起動した MHP で、障害が発生した場合

- time

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

## (2) ERREVT1

### (a) 形式

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

### (b) MCF イベントとして設定される項目

- ap\_name

次に示すどちらかが設定されます。

- 形式不正の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 I 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] ;              ... トランザクションブランチ
                                ID領域
};

```

### (b) MCF イベントとして設定される項目

- **ap\_name**

異常が発生した MHP のアプリケーション名が設定されます。

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

- **service\_name**

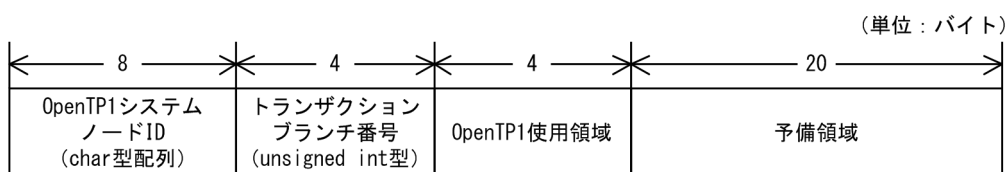
異常が発生した MHP のアプリケーション名に対応するサービス名が設定されます。

- **serv\_grp\_name**

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

- **bid**

トランザクションブランチ ID が次の形式で設定されます。



## (5) ERREVT

### (a) 形式

```
struct dc_mcf_evta_type {
    struct dc_mcf_evtheader  evtheader ;
                                ... MCFイベント共通ヘッダ
    char  reserve01[12] ;      ... 予備
    char  map_name[10] ;      ... MCF使用領域
    char  reserve03[2] ;      ... 予備
    char  ap_name[10] ;       ... アプリケーション名
                                (正常終了したメッセージ
                                のアプリケーション名)
    char  reserve04[2] ;      ... 予備
    char  reserve05[32] ;     ... 予備
    char  reserve06[32] ;     ... 予備
    DCLONG user_leng ;        ... 他プロトコルの場合の使用領域
    char  user_data[16] ;     ... 他プロトコルの場合の使用領域
    char  reserve07[16] ;     ... 予備
};
```

### (b) MCF イベントとして設定される項目

- ap\_name

正常終了したメッセージのアプリケーション名が設定されます。

アプリケーション名は、MHP から送信されたメッセージの場合に設定されます。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)<sub>16</sub>

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 I 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

## 5.2.4 MCF イベント情報の形式 (COBOL 言語)

COBOL 言語の場合はセグメントの並びとして渡されます。

COBOL 言語の UAP の場合, MCF イベント情報の内容を以降の表に示します。

表 5-2 COBOL 言語の MCF イベント情報の内容 (ERREVT1)

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	—	2	—	—
予備 (形式 1 のときだけ)	2	—	2	—	—
エラーイベントコード	4	0	3	英数字	'ERR'が設定されます。
	7	3	3	—	—
	10	6	2	英数字	ERREVT1 を示す'l△'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"> <li>形式不正となったアプリケーション名</li> <li>定義されていないアプリケーション名</li> </ul>
予備	48	44	8	—	—
予備	56	52	8	—	—
予備	64	60	8	—	—
コネクション名	72	68	8	英数字	コネクション名です。
予備	80	76	16	—	—
メッセージが入力された日付	96	92	8	外部 10 進	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 5-3 COBOL 言語の MCF イベント情報の内容 (ERREVT2)

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	—	2	—	—
予備 (形式 1 のときだけ)	2	—	2	—	—
エラーイベントコード	4	0	3	英数字	'ERR'が設定されます。
	7	3	3	—	—
	10	6	2	英数字	ERREVT2 を示す'2△'が設定されます。
入力元論理端末名称	12	8	8	英数字	<p>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。</p> <ul style="list-style-type: none"> <li>• SPP からアプリケーション起動機能で起動した MHP で障害が発生した場合</li> <li>• 上記の障害が発生したあとに、MCF イベントとして起動した MHP からさらにアプリケーション起動機能で起動した MHP で、障害が発生した場合</li> </ul>
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	<p>コネクション名です。 次に示す場合は、 '*'が設定されます。</p> <ul style="list-style-type: none"> <li>• SPP からアプリケーション起動機能で起動した MHP で障害が発生した場合</li> <li>• 上記の障害が発生したあとに、MCF イベントとして起動した MHP からさらにアプリケーション起動機能で起動した MHP で、障害が発生した場合</li> </ul>
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	<p>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</p>

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
メッセージが入力された時刻	104	100	8	外部 10 進	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
理由コード※	112	108	4	外部 10 進	理由コードが設定されます。
予備	116	112	12	—	—

(凡例)

—：該当しません。または、使用されません。

注※

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

表 5-4 COBOL 言語の MCF イベント情報の内容 (ERREVT3)

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	—	2	—	—
予備 (形式 1 のときだけ)	2	—	2	—	—
エラーイベントコード	4	0	3	英数字	'ERR'が設定されます。
	7	3	3	—	—
	10	6	2	英数字	ERREVT3 を示す'3△'が設定されます。
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"> <li>SPP からアプリケーション起動機能で起動した MHP で、障害が発生した場合</li> <li>上記の障害が発生したあとに、MCF イベントとして起動した MHP からさらにアプリケーション起動機能で起動した MHP で、障害が発生した場合</li> </ul>
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。

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備	64	60	8	—	—
コネクション名	72	68	8	英数字	コネクション名です。 次に示す場合は、'*'が設定されます。 <ul style="list-style-type: none"> <li>• SPP からアプリケーション起動機能で起動した MHP で、障害が発生した場合</li> <li>• 上記の障害が発生したあとに、MCF イベントとして起動した MHP からさらにアプリケーション起動機能で起動した MHP で、障害が発生した場合</li> </ul>
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	—	—
トランザクションブランチ ID (BID)	192	188	36	英数字	異常が発生したトランザクションの BID です。 トランザクションブランチ ID の形式については、表 5-5 を参照してください。
予備	228	224	28	—	—

(凡例)

—：該当しません。または、使用されません。

表 5-5 トランザクションブランチ ID の形式

項目	位置 (バイト)	長さ (バイト)	属性
OpenTP1 システムノード ID	0	8	英数字
トランザクションブランチ番号	8	4	2 進数字
OpenTP1 使用領域	12	4	—
予備	16	20	—

表 5-6 COBOL 言語の MCF イベント情報の内容 (ERREVTA)

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
メッセージが入力された時刻	104	100	8	外部 10 進	SS: 秒 00 は固定です。
予備	112	108	16	—	—
予備	128	124	31	—	—
予備	159	155	1	—	—
予備	160	156	31	—	—
予備	191	187	1	—	—
予備	192	188	36	—	—
予備	228	224	28	—	—

(凡例)

—: 該当しません。または、使用されません。

表 5-7 COBOL 言語の MCF イベント情報の内容 (CERREVT)

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	イベントコード「CERREVT」が設定されます。
入力元論理端末名称	8	8	英数字	障害の発生した論理端末名称が設定されます
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) <sub>16</sub> : 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) <sub>16</sub> : コマンド入力による手動回復 (00000000) <sub>16</sub> : システムによる自動回復

項目	位置 (バイト)	長さ (バイト)	属性	内容
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。

注※

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

表 5-8 COBOL 言語の MCF イベント情報の内容 (COPNEVT, CCLSEVT)

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	イベントコード「COPNEVT」, または「CCLSEVT」が設定されます。
入力元論理端末名称	8	8	英数字	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) <sub>16</sub> : コマンド入力による手動回復 (00000000) <sub>16</sub> : 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。

# 6

## 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 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 6-1 MCF で使用する定義ファイル

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

定義のソースファイルは、定義コマンド、オプション、およびオペランドを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。次の定義には、TP1/NET/OSI-TP に固有の定義があります。

- MCF マネージャ定義
- MCF 通信構成定義

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、`mcftbuf` (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、「`mcftalccn` (コネクション定義の開始)」の「注意事項」を参照してください。

#### 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、「8. [組み込み方法](#)」を参照してください。

## 通信定義の内容の関連づけ

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。

## 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 6-2 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	同期型送信監視時間	符号なし整数((0~65535)) 《0》(単位:秒)	
			sndrcvtim	同期型送受信監視時間	符号なし整数((0~65535)) 《0》(単位:秒)	
			recvtim	同期型受信監視時間	符号なし整数((0~65535)) 《0》(単位:秒)	
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	mcftgrpcn (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
		mcftalccn (コ ネクション定義 の開始) 指定 数:1~512	-c	—	コネクション ID	1~8 文字の識別子
			-p	—	プロトコルの種別	tp
			-n	—	自システムの PSAP ア ドレス	1~142 けたの 16 進数字
			-q	—	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	符号なし整数((1~512))
			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))
		-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	—		コネクションの確立 方法	auto   《manual》		
-u	—		相手システムの種別	ht		

定義名		コマンド	オプション・オペランド		定義内容	指定値((値範囲))《省略時解釈値》
MCF 通信構成定義	プロトコル固有定義	mcftalccn (コネクション定義の開始) 指定数:1~512	-w	nomltim	プロトコル監視時間	符号なし整数((0, 10~65535))《60》(単位:秒)
			-b	bretry	コネクション確立障害時の確立再試行をするかどうかを指定	《yes》   no
				bretrycnt	コネクション確立障害時の確立再試行回数	符号なし整数((0~65535))《0》(単位:回)
				bretryint	コネクション確立障害時の確立再試行間隔	符号なし整数((0~2550))《60》(単位:秒)
			-t	—	コネクションの種別	int   rsp
			-z	slot	仮想スロット番号	符号なし整数((0~65535))
			-l	—	TL クラス	0
			-y	—	コンテンション勝者の割り当て先	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を指定	《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を指定	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	—	論理端末の端末タイプ	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	—
		mcftgrpded (コネクショングループ定義の終了)	—	—	コネクショングループ定義の終了	—

定義名		コマンド	オプション・オペランド		定義内容	指定値((値範囲))《省略時解釈値》
MCF 通信構成定義	プロトコル固有定義	終了) 指定数:mcftgrpcnと同数	-	-	コネクシヨ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 6-1 TP1/NET/OSI-TP のプロトコル固有定義コマンドの指定順序

(1) コネクシヨングループを指定しない場合

mcftalccn	(コネクシヨ定義の開始)	}	※
mcftalcle	(論理端末定義)		
mcftalced	(コネクシヨ定義の終了)		
⋮	繰り返し指定可能		
mcftalccn	(コネクシヨ定義の開始)	}	※
mcftalcle	(論理端末定義)		
mcftalced	(コネクシヨ定義の終了)		

(2) コネクシヨングループを指定する場合

{	mcftgrpcn	(コネクシヨングループ定義の開始)	}	※
	mcftalccn	(コネクシヨ定義の開始)		
	mcftalcle	(論理端末定義)		
	mcftalced	(コネクシヨ定義の終了)		
	⋮	繰り返し指定可能		
	⋮			
	mcftalccn	(コネクシヨ定義の開始)		
	mcftalcle	(論理端末定義)		
	mcftalced	(コネクシヨ定義の終了)		
	mcftgrped	(コネクシヨ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=同期型送信監視時間]  
        [sndrcvtim=同期型送受信監視時間]  
        [recvtim=同期型受信監視時間] "]  
      :
```

## 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=同期型送信監視時間**    ~<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。

#### 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、このオペランドで指定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、指定した監視時間よりも短い時間でタイムアウトすることがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は 3 (単位：秒) 以上の値の設定を推奨します。

**sndrcvtim=同期型送受信監視時間**    ~<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。

#### 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、このオペランドで指定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、指定した監視時間よりも短い時間でタイムアウトすることがあります。監視時間

が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は3（単位：秒）以上の値の設定を推奨します。

recvtim=同期型受信監視時間 ~<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。

#### 注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcftim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、このオペランドで指定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、指定した監視時間よりも短い時間でタイムアウトすることがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は3（単位：秒）以上の値の設定を推奨します。

# mcftalccn (コネクション定義の開始)

## 形式

```
mcftalccn -c コネクションID
          -p tp
          -n x' 自システムのPSAPアドレス'
          -q x' 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=コネクション確立障害時の確立再試行回数] ※1
      [bretryint=コネクション確立障害時の確立再試行間隔] " ] ※1
          -t int | rsp ※1, ※2, ※3
          -z "slot=仮想スロット番号"※1
          -l 0※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 (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。

## 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。

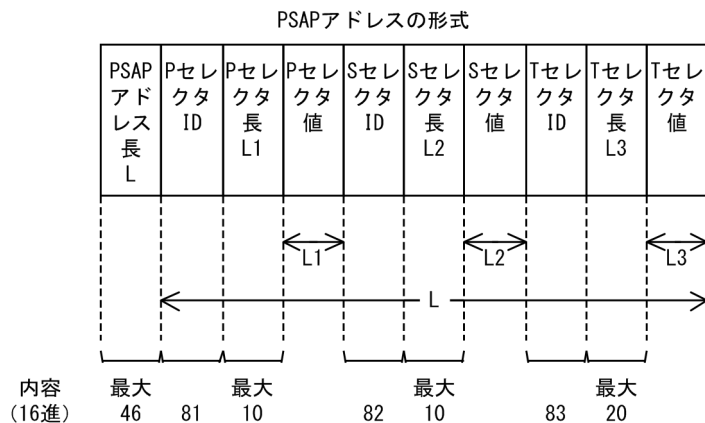
tp

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 アドレスの形式を次に示します。

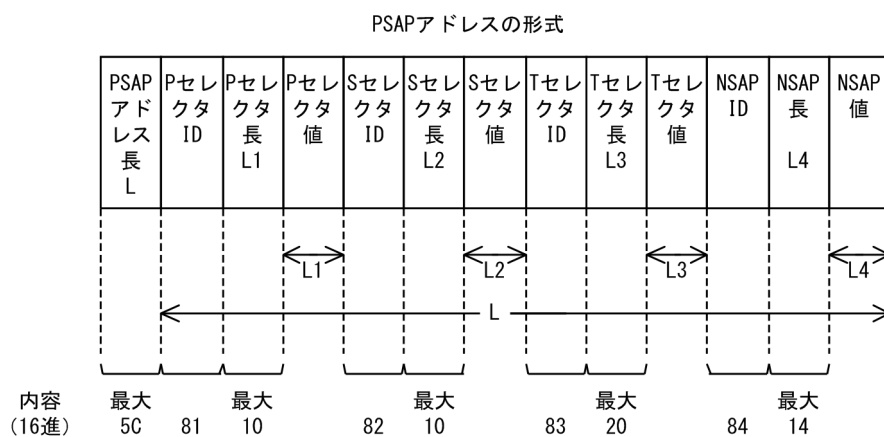


ほかのプロセスが使用する PSAP アドレスと重複しない PSAP アドレスを指定してください。

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

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。

相手システムの PSAP アドレスの形式を次に示します。なお、NSAP 値の形式については、マニュアル「通信管理 XNF/AS NSAP アドレス概説編」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、出力メッセージ編集 UOC 呼び出し時に、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、出力メッセージ編集 UOC 呼び出し時に、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。

メッセージ編集用バッファグループ番号に対応する mcftbuf の count オペランドおよび extend オペランドには、すべての論理端末で入力メッセージ編集 UOC および出力メッセージ編集 UOC が、同時に動作した場合を考慮した面数を指定してください。

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。

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。

msgbuf オペランドを指定した場合、このオペランドを省略できません。省略した場合、定義オブジェクト生成時に KFCA11519-E メッセージを出力してエラーとなります。

## ●-m

(オペランド)

mode=xnfas

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》

接続の確立方法を指定します。

auto

オンライン開始・再開始時に接続を自動的に確立します。

manual

MCF 起動後、接続を確立します。接続の確立は、運用コマンド (mcftactcn) の入力、または API (dc\_mcf\_tactcn 関数もしくは CBLDCMCF('TACTCN△△')) の発行で行います。

●-u ht

相手システムの種別を指定します。

ht

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)

接続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。

注意事項

監視時間の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔でタイムアウトが発生したかどうかを監視しています。このため、このオペランドで指定した監視時間と実際にタイムアウトを検出する時間には秒単位の誤差が生じます。そのため、タイミングによっては、指定した監視時間よりも短い時間でタイムアウトすることがあります。監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、監視時間は 3 (単位: 秒) 以上の値の設定を推奨します。

●-b

(オペランド)

bretry=yes | no ~ 《yes》

接続確立時に障害が発生した場合、接続の確立再試行をするかどうかを指定します。

yes

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

コネクションの確立再試行をする場合の確立再試行回数を指定します。

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

コネクションの確立再試行をする場合の確立再試行間隔を指定します。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。

注意事項

再試行間隔の精度は秒単位です。また、タイマ定義 (mcfttim -t) の btim オペランドで指定する時間の間隔で再試行するかどうかをチェックします。このため、このオペランドで指定した再試行間隔と実際に再試行する時間には秒単位の誤差が生じます。

## ●-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)

slot=仮想スロット番号 ～<符号なし整数>((0~65535))

仮想スロット番号を指定します。

通信管理の定義で指定した仮想スロット番号を指定してください。

## ●-l 0

使用する TL クラスを指定します。

0

TL クラス 0

### ●-y int | rsp

コンテンション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 を通知します。

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。

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"><li>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</li><li>ダイアログ開始要求以外 ユーザデータ長 以上</li></ul>	2 以上
rcvbuf	sndbuf と同じ	<ul style="list-style-type: none"><li>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</li><li>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</li></ul>
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。

## 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 (mcfaalcap -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-DIALODUE) 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。

## 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 mcfmcname コマンドに指定します。

### 形式

#### set 形式

```
set module="TP1/NET/OSI-TPの実行形式プログラム名"  
[set mcf_prf_trace=Y|N]
```

### 機能

プロセスサービスが MCF 通信サービスを起動するための環境を定義します。

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

### 説明

set 形式のオペランド

#### ●module="TP1/NET/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せん。

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。

## ●mcf\_prf\_trace=Y|N ~ 〈Y〉

MCF 通信サービスごとに、MCF 性能検証用トレース情報を取得するかどうかを指定します。このオペランドの指定値を有効にするには、システムサービス共通情報定義の mcf\_prf\_trace\_level オペランドに 00000001 を指定してください。

Y

MCF 性能検証用トレース情報を取得します。

N

MCF 性能検証用トレース情報を取得しません。

MCF 通信サービスでの MCF 性能検証用トレース情報取得有無とオペランドの指定値の関係を、次の表に示します。

システムサービス共通情報定義 mcf_prf_trace_level オペランドの指定値	システムサービス情報定義 mcf_prf_trace オペランドの指定値	
	Y	N
00000000	取得しない	取得しない
00000001	取得する	取得しない

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

# 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
[set mcf_prf_trace_level=MCF性能検証用トレース情報の取得レベル]
```

## 機能

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

## 説明

### set 形式のオペランド

この定義には、ほかにもオペランドがあります。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。

### ●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〉((64~3596))

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値を指定します。

OpenTP1 制御下のプロセスでは、システムサーバやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数などによって、ソケット用のファイル記述子の最大数を変更する必要があります。

各 MCF 通信プロセスまたはアプリケーション起動プロセスが使用する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。

```
↑ (このMCF通信プロセスに対してメッセージ送信要求を行うUAPプロセス数※1
  +システムサービスプロセス数※2
  +このMCF通信プロセスまたはアプリケーション起動プロセスに対して同時に処理要求を行う運用コマンド数
  ) / 0.8 ↑
```

(凡例)

↑↑：小数点以下を切り上げます。

注※1

アプリケーション起動プロセス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。自 OpenTP1 内のシステムサービスプロセスは、rpcstat コマンドで表示されるサーバ名をカウントすることで求められます。rpcstat コマンドで表示されるサーバ名のうち、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造に記載されているシステムサービスプロセスをカウントしてください。

自 OpenTP1 内の各 MCF 通信プロセスおよびアプリケーション起動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。

このオペランドの指定値が小さいと、OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため、プロセスが KFCA00307-E メッセージを出力して異常終了します。

●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〉 ((500~4032))

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。

MCF 通信プロセスが行うメッセージの送受信にもファイル記述子が使われます。この数が不足すると、コネクションの確立ができないなどの障害が発生するため、事前に必要となるファイル記述子の数を設定しておく必要があります。

各 MCF 通信プロセスが使用するファイル記述子の最大数を求める計算式を次に示します。

(プロトコル制御で使用するファイル記述子数※1)  
+MCFメイン関数でユーザが使用するファイル記述子数  
+30※2

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。実際に通信を行う論理端末の総数ではありません。

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-max\_open\_fds オペランドの指定値の超過分」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。

max\_socket\_descriptors オペランドと max\_open\_fds 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。

(「max\_socket\_descriptorsオペランドの指定値」  
+「max\_open\_fdsオペランドの指定値」) ≤4096

ただし、TP1/NET/OSI-TP の MCF 通信プロセスで使用できるファイル記述子の最大数は 2048 です。

TP1/NET/OSI-TP の MCF 通信プロセスで、max\_socket\_descriptors オペランドと max\_open\_fds オペランドの和が 1 プロセスで使用できるファイル記述子の最大数を超過している場合、TP1/NET/OSI-TP の MCF 通信プロセスで使用できるファイル記述子数は、1 プロセスで使用できるファイル記述子の最大数に強制的に補正されます。

max\_socket\_descriptors オペランドと max\_open\_fds オペランドの和が 1 プロセス当たりでオープンできるファイル数の物理限界値（ハードリミット）を超過していたとき、MCF の開始を中断します。

●mcf\_prf\_trace\_level=MCF 性能検証用トレース情報の取得レベル    ~((00000000~00000001))  
《00000001》

MCF 性能検証用トレース情報の取得レベルを指定します。MCF 性能検証用トレースを取得する場合は、システム共通定義の prf\_trace オペランドに Y を指定するか、または省略してください。

00000000

MCF 性能検証用トレース情報を取得しません。

00000001

MCF 性能検証用トレース情報（イベント ID：0xa000~0xa0ff）を取得します。イベント ID の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。また、TP1/NET/OSI-TP 固有の出力情報や取得タイミングについては、「付録 G MCF 性能検証用トレースの取得」を参照してください。

オペランドの指定に誤りがある場合は、OpenTP1 開始処理中に OpenTP1 が異常終了します。

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

## 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
         [-r {no | rep} ]
```

## 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、開始から再開の間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、再開時に正常に動作しないこと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。

次に示す条件を満たした名称を指定してください。

- 先頭 3 文字が `_mu` で始まる最大 8 文字の名称
- 通信サービス定義 (mcfmcname -s) の mcfsvname オペランドで指定する MCF 通信サーバ名

### ●-r {no | rep} ～<no>

定義オブジェクトファイルの出力先に読み取り権限を持つファイルがすでに存在する場合、定義オブジェクトファイルを上書きするかどうかを指定します。

#### no

定義オブジェクトファイルを上書きしないで、KFCA10332-E メッセージを出力します。

#### rep

定義オブジェクトファイルを上書きします。

## MCF 定義オブジェクトの解析

対応する定義ソースが不明となった MCF 定義オブジェクトファイルの内容を知りたい場合に、MCF 定義オブジェクトの解析をします。ここでは、MCF 定義オブジェクト解析コマンドについて説明します。

### 形式

```
mcfostrpr -i [パス名] 解析対象オブジェクトファイル名
```

### 機能

TP1/NET/OSI-TP のプロトコル固有定義オブジェクトファイル（または、これと共通定義を結合した MCF 通信構成定義オブジェクトファイル）を解析し、定義ソースの形式で標準出力します。

### オプション

●-i [パス名] 解析対象オブジェクトファイル名 ~ <1~8 文字の英数字>

定義オブジェクトが格納されているファイル名を指定します。

### 出力例

```
#####  
MCF communication configuration definition  
OSI-TP definition  
#####  
OBJECT FILE NAME : xxxxxxxx  
VV-RR             : vv-rr  
DATE              : yyyy-mm-dd hh:mm:ss  
#####  
  
mcftalccn  
-c                = cn01  
-p                = tp  
  
.  
.  
.  
  
mcftalcle  
-l                = le01  
-t                = any  
  
.  
.  
.  
  
mcftalced  
  
##### End Of File #####
```

## 解析結果

定義オブジェクト解析コマンドは、その解析結果を定義ソースの形式で出力します。出力される内容は解析結果であり、記述形式は元の定義ソースの記述形式とは一致しません。定義ソースと定義オブジェクト解析結果の差異を次の表に示します。

表 6-3 定義ソースと定義オブジェクト解析結果の差異

項目	定義ソース	定義オブジェクト解析結果
注釈文	書き込みできる。	出力しない。
省略値の扱い	省略できる。	限定公開部分も含めて、省略値を出力する。
限定公開部分の表記方法	一般公開部分と差異なし。	バージョン 7 での限定公開機能の行の先頭に、"*"を付与する。
定義コマンド名とオプションの表記方法	1 行に表記できる。 (例) mcftalccn -c cn01	定義コマンド名を表記後、改行する。また、オプションに"="を付記する。 (例) mcftalccn -c = cn01
1 定義コマンドが複数の行にわたる場合	継続記号"¥"を付与する。 (例) mcftalccn -c cn01 ¥ -p tp	継続記号は出力しない。 (例) mcftalccn -c = cn01 -p = tp
1 定義オプションに複数のオペランドを指定する場合	複数のオペランドをまとめて二重引用符(")で囲む。 (例) mcftalccn -g "sndbuf=1 rcvbuf=2"	個々のオペランドに対してオプションを付記する。 (例) mcftalccn -g sndbuf = 1 -g rcvbuf = 2
その他	なし	<ul style="list-style-type: none"><li>ファイル名などを記したタイトルが出力される。</li><li>定義オブジェクト生成時の補正によって、実際の指定値とは異なる内容が出力される場合がある。</li><li>定義ソースと該当コマンドのバージョンの差異によって、解析結果にサポート内容の過不足がある場合がある。</li></ul>

## 注意事項

解析対象が不正であった場合は、正常に動作しないことがあります。

# 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とおり異なります。

- OSI 拡張機能を使用する場合  
XNF/AS/BASE  
XNF/AS/OSI Extension
- 自局 IP アドレス指定機能を使用する場合  
XNF/AS/BASE  
XNF/AS/OSI Extension  
XNF/AS/OSI Extension/Cluster
- OSI 拡張高信頼化機能を使用する場合  
XNF/AS/BASE  
XNF/AS/OSI Extension  
XNF/AS/Host Adaptor

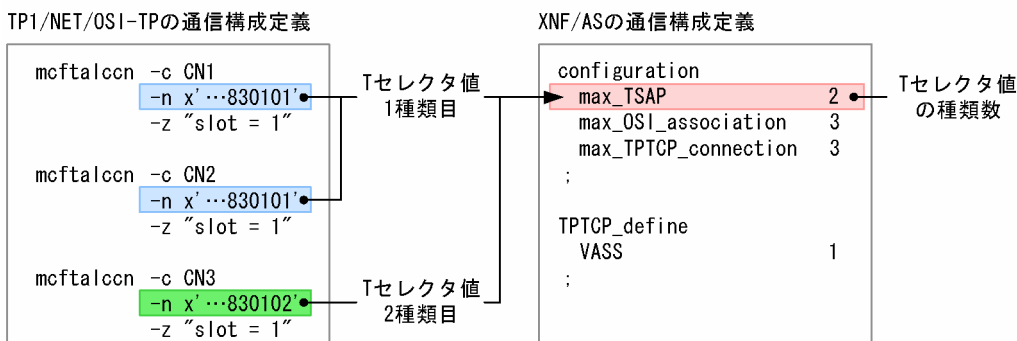
通信管理プログラムの詳細については、マニュアル「通信管理 XNF/AS 解説・運用編」およびマニュアル「通信管理 XNF/AS 構成定義編」を参照してください。

どの通信管理プログラムの機能を使用する場合でも、接続定義 (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。

- コネクション数
- T セレクタ値の種類数※
- 仮想スロット番号

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 セレクタ値の種類数である 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、使用する機能によって異なります。使用する機能別に、次に示します。

## 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 6-4 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	mcftalccn	コネクション数 (定義数)	configuration	max_OSI_association max_TPTCP_connection
2		T セレクタ値の種類数 (-n オプションの T セレクタ値)		max_TSAP
3		仮想スロット番号 (-z オプションの slot オペランド)	TPTCP_define	VASS

## 自局 IP アドレス指定機能を使用する場合

自局 IP アドレス指定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 6-5 自局 IP アドレス指定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	mcftalccn	コネクション数 (定義数)	configuration	max_OSI_association max_TPTCP_connection
2		T セレクタ値の種類数 (-n オプションの T セレクタ値)		max_TSAP
3		仮想スロット番号 (-z オプションの slot オペランド)	TPTCP_slot	VASS

## OSI 拡張高信頼化機能を使用する場合

OSI 拡張高信頼化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 6-6 OSI 拡張高信頼化機能を使用する場合の XNF/AS の定義との関連づけ

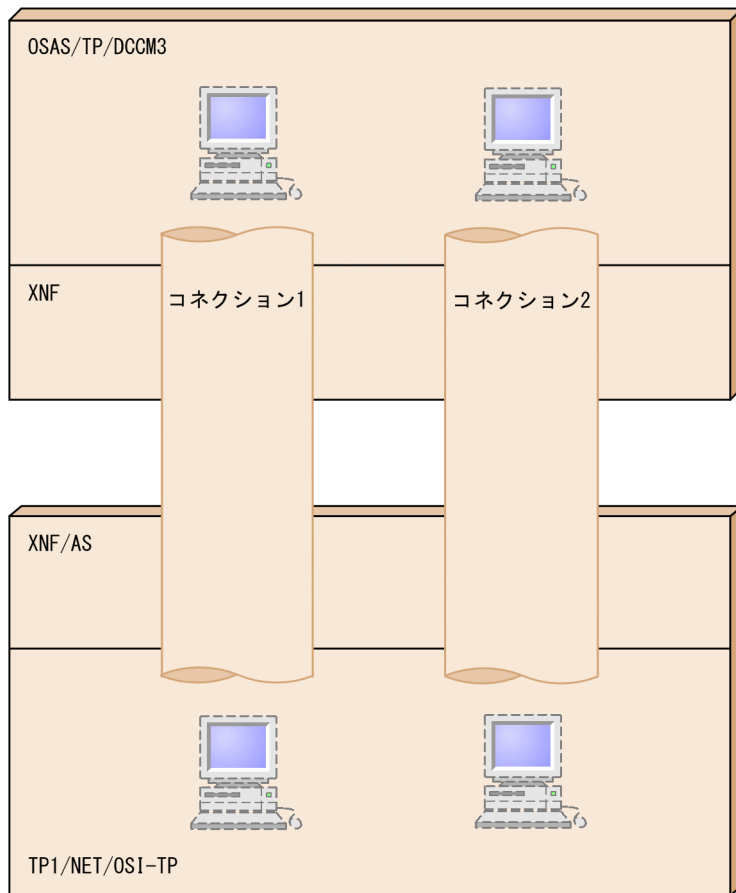
項番	TP1/NET/OSI-TP での定義内容		XNF/AS での定義内容	
	コマンド	内容	定義文	オペランド
1	mcftalccn	コネクション数 (定義数)	configuration	max_OSI_association max_TPTCP_connection
2		T セレクタ値の種類数 (-n オプションの T セレクタ値)		max_TSAP

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
3	mcftalccn	仮想スロット番号 (-z オプションの slot オペランド)	TPTCP_VC	VASS

## 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 6-2 ネットワーク構成の例



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

### XNF の定義と関連づける内容

TP1/NET/OSI-TP のシステム定義と XNF の定義との関係を以降の図に示します。XNF の定義については、マニュアル「XNF TCP/IP 接続機能 XNF/TCP 定義とコマンド」を参照してください。

図 6-3 XNF の定義との関係 (OSI 拡張機能または自局 IP アドレス指定機能を使用する場合)

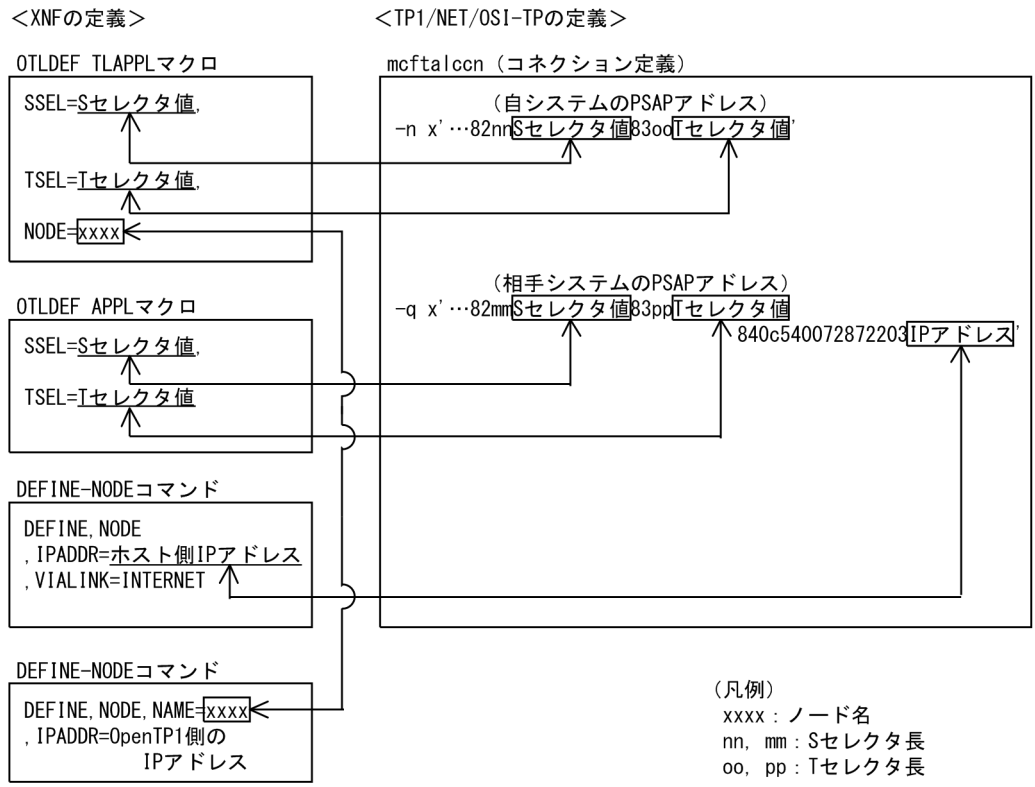
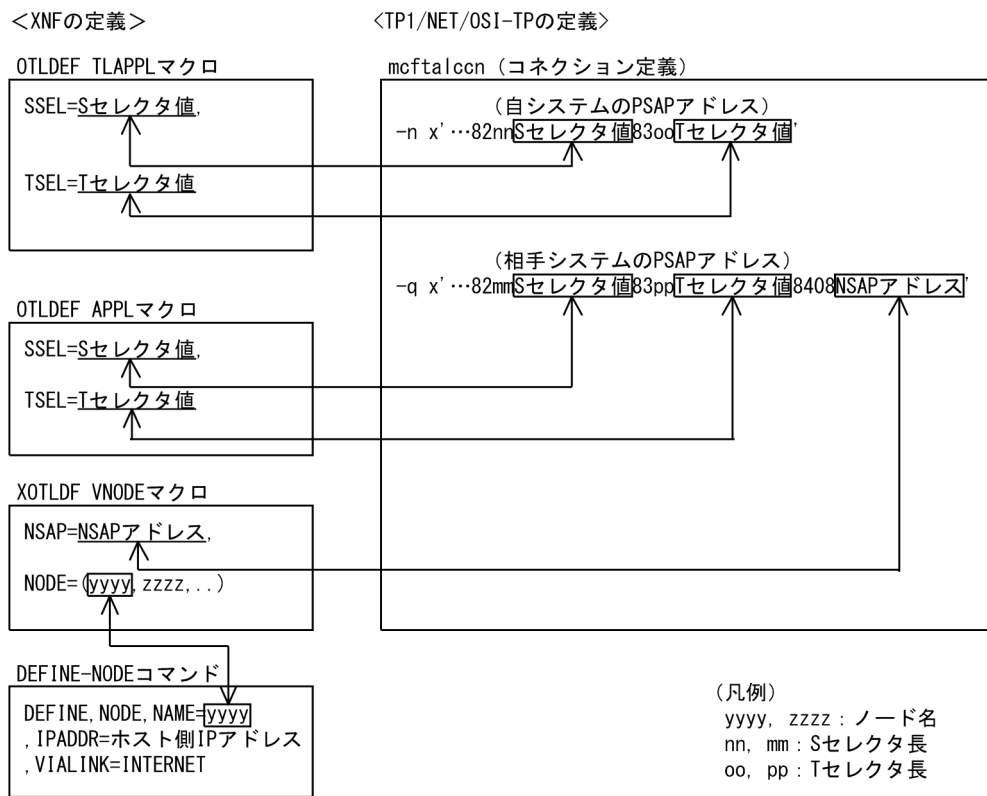


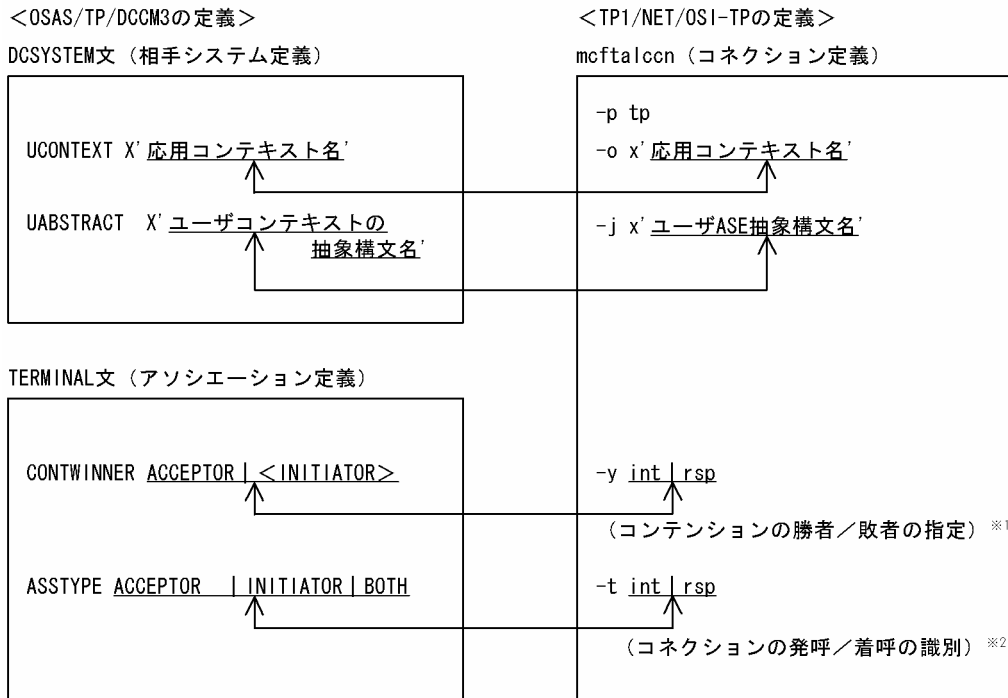
図 6-4 XNF の定義との関係 (OSI 高信頼化機能を使用する場合)



## XDM/DCCM3 システム定義と関連づける内容

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「OSI アプリケーション共通機能/OSI トランザクション処理/DCCM3 OSAS/TP/DCCM3」を参照してください。

図 6-5 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。

# OpenTP1 システムの変更に影響する定義

OpenTP1 システムの変更に伴って見直しが必要となる定義および OpenTP1 ファイルについて説明します。

## IP アドレスの変更

IP アドレスを変更する場合に、見直しが必要な定義および変更手順について説明します。

### IP アドレスを変更する場合に見直しが必要な定義

IP アドレスを変更する場合、見直す必要のある定義の一覧と発生する条件を次の表に示します。

表 6-7 IP アドレスを変更する場合に見直しが必要な定義の一覧

定義ファイル名	定義	見直しが必要な条件
MCF 通信構成定義	mcftalccn -q <sup>*1</sup>	OSI 拡張機能を使用する場合（OSI 拡張高信頼化機能を除く）
XNF の構成定義文	TPTCP_slot の IP_address <sup>*2</sup>	自局 IP アドレス指定機能を使用する場合

注※1

OSI 拡張機能を使用する場合、NSAP アドレスに IP アドレスが含まれています。NSAP アドレスの形式については、マニュアル「通信管理 XNF/AS NSAP アドレス概説編」またはマニュアル「通信管理 XNF/LS 使用の手引」を参照してください。

注※2

詳細については、マニュアル「通信管理 XNF/AS 構成定義編」またはマニュアル「通信管理 XNF/LS 使用の手引」を参照してください。

## IP アドレスの変更手順

IP アドレスは、次の手順で変更してください。

1. OpenTP1 を正常停止します。
2. XNF/AS を停止します。
3. MCF 通信構成定義および XNF の構成定義文について、変更前の IP アドレスを grep コマンドで検索します。
4. 検索の結果、変更前の IP アドレスが見つかった場合には、変更します。
5. MCF 通信構成定義を変更した場合、MCF 通信構成定義の定義オブジェクトファイルを再作成します。XNF の構成定義文を変更した場合、ゼネレーションを行います。

## コネクション（論理端末）の追加

コネクション（論理端末）を追加する場合に見直す必要のある定義の一覧、および再見積もりが発生する条件を次の表に示します。

表 6-8 コネクション（論理端末）を追加する場合に見直しが必要な定義の一覧

定義ファイル名	定義	再見積もりが発生する条件
システム環境定義	static_shmpool_size <sup>*1</sup>	無条件に再見積もりが必要
	dynamic_shmpool_size <sup>*1</sup>	同時に送受信するメッセージ数が増加する場合
MCF マネージャ定義	mcfmcomn -n	メッセージ出力通番を使用する場合
	mcfmcomn -p <sup>*2</sup>	無条件に再見積もりが必要
	mcfmexp -l <sup>*3</sup>	拡張予約定義を定義している場合
MCF 通信構成定義	mcftbuf -g count <sup>*4</sup>	無条件に再見積もりが必要
	mcftsts -l	状態を引き継ぐ論理端末が増える場合
システムサービス共通情報定義	max_open_fds <sup>*5</sup>	無条件に再見積もりが必要

注※1

詳細については、マニュアル「OpenTP1 システム定義」の「MCF サービス用の共用メモリの見積もり式」の説明を参照してください。

注※2

詳細については、マニュアル「OpenTP1 システム定義」の「mcfmcomn」と「MCF サービス用の共用メモリの見積もり式」の説明を参照してください。

注※3

詳細については、マニュアル「OpenTP1 システム定義」の「mcfmexp」の説明を参照してください。

注※4

詳細については、「mcftalccn（コネクション定義の開始）」の注意事項とマニュアル「OpenTP1 システム定義」の「mcftbuf」の説明を参照してください。

注※5

詳細については、「システムサービス共通情報定義」を参照してください。

見直す必要のある OpenTP1 ファイルの一覧、および再見積もりが発生する条件を次の表に示します。

表 6-9 コネクション（論理端末）を追加する場合に見直しが必要な OpenTP1 ファイルの一覧

OpenTP1 ファイル	再見積もりが発生する条件
ステータスファイル	次に示すどれかの条件の場合、再見積もりが必要 <ul style="list-style-type: none"> <li>メッセージ出力通番を使用する場合</li> <li>拡張予約定義を定義している場合</li> <li>状態を引き継ぐ論理端末が増える場合</li> </ul>
メッセージキューファイル	入力キューまたは出力キューにディスクキューを割り当てていて、同時に送受信するメッセージ数が増加する場合

また、TP1/NET/OSI-TP の「リリースノート」を参照し、MCF 通信プロセスが使用するローカルメモリのメモリ所要量も見直してください。

## 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次の図に、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

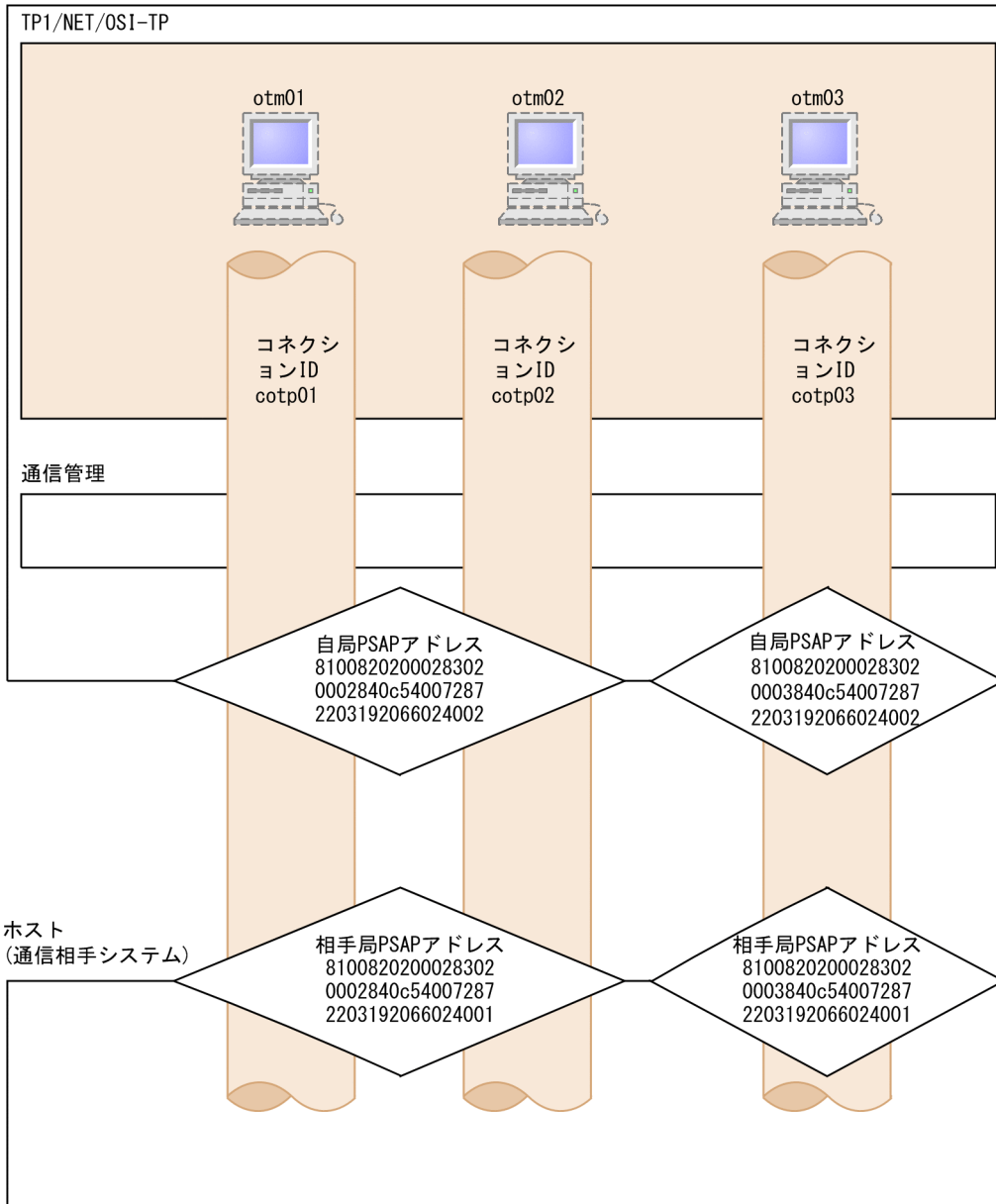
- /BeTRAN/examples/mcf/OSITP/conf/mngr1
- /BeTRAN/examples/mcf/OSITP/conf/com\_c1
- /BeTRAN/examples/mcf/OSITP/conf/com\_d1

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 mcftalccn の定義に次の指定を追加します。

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

図 6-6 TP1/NET/OSI-TP のシステム構成例

自システム



## TP1/NET/OSI-TP の MCF マネジャ環境定義例

```
# ----- ファイル1 MCFマネジャ環境定義 ----- #
mcfmenv -m "name =TPmngr"
#
mcfmcomn -p 100
#
mcfmcname -s "mcfsvname =_muTPrsp ¥
             syssvname =mcfutp"
mcfmcname -s "mcfsvname =_muTPint ¥
             syssvname =mcfutp"
#
mcfmuap
# ----- ファイル1 MCFマネジャ環境定義 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 ----- #
mcftenv   -s 01                               ￥
          -a _APCNF01
mcftcomm
mcfttrc
mcftbuf   -g "groupno=1                       ￥
          length =2048                         ￥
          count  =35"
mcftbuf   -g "groupno=2                       ￥
          length =2048                         ￥
          count  =35"
mcftbuf   -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###
mcftalccn -c cotp01                            ￥
          -p tp                                ￥
          -n x'0a81008202000283020002'        ￥
          -q x'1881008202000283020002840c540072872203192066024001' ￥
          -g "sndbuf =1                        ￥
          rcvbuf =2"                          ￥
          -e "msgbuf=4                        ￥
          count=1"                            ￥
          -m "mode=xnfas"                     ￥
          -i auto                              ￥
          -u ht                                ￥
          -w "nomltim=0"                      ￥
          -b "bretrycnt=2                     ￥
          bretryint=10"                       ￥
          -t int                              ￥
          -z "slot=3"                         ￥
          -l 0                                 ￥
          -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'1881008202000283020002840c540072872203192066024001' ¥
-g "sndbuf =1 ¥
   rcvbuf =2" ¥
-e "msgbuf=4 ¥
   count=1" ¥
-m "mode=xfas" ¥
-i auto ¥
-u ht ¥
-w "nomltim=0" ¥
-b "bretrycnt=2 ¥
   bretryint=10" ¥
-t int ¥
-z "slot=3" ¥
-l 0 ¥
-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'1881008202000283020003840c540072872203192066024001' ¥
          -g "sndbuf =1 ¥
             rcvbuf =2" ¥
          -e "msgbuf=4 ¥
             count=1" ¥
          -m "mode=xfas" ¥
          -i auto ¥
          -u ht ¥
          -w "nomltim=0" ¥
          -b "bretrycnt=2 ¥
             bretryint=10" ¥
          -t int ¥
          -z "slot=3" ¥
          -l 0 ¥
          -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 ----- #

```

# 7

## 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

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 7-1 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	<a href="#">mcftactcn</a>	×	×	○	○
	コネクションの解放	<a href="#">mcftdctcn</a>	×	×	○	○
	コネクション、および論理端末の状態表示	<a href="#">mcftlscn</a>	×	×	○	○

(凡例)

- ：組み込み、または実行ができます。
- ×

# mcftactcn (コネクションの確立)

## 形式

```
mcftactcn [-s MCF通信プロセス識別子]
           {-c コネクションID | -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 ～<数字 (0~9), a~f >((01~ef))

処理対象のコネクションを制御する MCF 通信サービスの 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 通信サービスに対して、mcftactcn コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

MCF 通信サービスが多い構成や運用コマンドを多数入力する運用を行う場合は、-s オプションで、MCF 通信プロセス識別子を指定する運用設計を行ってください。

### ●-c コネクション ID ～<1~8 文字の識別子>

確立するコネクションのコネクション ID を指定します。

コネクション ID は、一度に 8 個まで指定できます。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

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。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

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。	メッセージログファイル、または標準エラー出力
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

メッセージID	内容	出力先
KFCA10358-E	内部関数のエラーが発生しました。	メッセージログファイル, または標準エラー出力
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
KFCA10391-E	mcftactcn コマンドはサポートされていません。	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
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	mcftactcn 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 (コネクシヨンの解放)

## 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

コネクシヨンを解放します。

## オプション

### ●-s MCF 通信プロセス識別子 ~<数字 (0~9), a~f >((01~ef))

処理対象のコネクシヨンを制御する MCF 通信サービスの 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 通信サービスに対して、mcftdctcn コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

MCF 通信サービスが多い構成や運用コマンドを多数入力する運用を行う場合は、-s オプションで、MCF 通信プロセス識別子を指定する運用設計を行ってください。

### ●-c コネクション ID ~<1~8 文字の識別子>

解放するコネクシヨンのコネクション 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。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

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。

\*: すべてのコネクシヨンを解放します。

先行文字列\*: 先行文字列で始まるすべてのコネクシヨンを解放します。

〈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。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

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	mcftdctcn 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。	メッセージログファイル、または標準エラー出力

メッセージID	内容	出力先
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。	メッセージログファイル、または標準エラー出力
KFCA10359-W	mcftdctn 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	mcftdctn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftdctn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定した接続は登録されていません。	標準エラー出力
KFCA10391-E	mcftdctn コマンドはサポートされていません。	標準エラー出力
KFCA10501-I	ヘルプメッセージ	標準出力
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	接続が未確立のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13933-E	接続確立処理中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13934-E	接続解放処理中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13941-E	接続使用中のため、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接続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 ～<数字 (0～9), a～f >((01～ef))

処理対象のコネクションを制御する MCF 通信サービスの 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 通信サービスに対して、mcftlscn コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

MCF 通信サービスが多い構成や運用コマンドを多数入力する運用を行う場合は、-s オプションで、MCF 通信プロセス識別子を指定する運用設計を行ってください。

### ●-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。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

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 Δcnn2 Δcnn3"
```

〈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。

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。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

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 ccccccc 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 ccccccc ppp sssss dddd gggggggg」の行だけ出力されます。

- mmm : MCF 識別子
- ccccccc : 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	mcftlscn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	メッセージログファイル, または標準エラー出力
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。	メッセージログファイル, または標準エラー出力
KFCA10359-W	mcftlscn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10360-I	状態表示を開始します。	標準出力
KFCA10361-I	標準情報を表示します。	標準出力
KFCA10362-I	詳細情報を表示します。	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10373-E	mcftlscn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力

メッセージID	内容	出力先
KFCA10381-E	指定した接続は登録されていません。	標準エラー出力
KFCA10391-E	mcftlscn コマンドはサポートされていません。	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	mcftlscn コマンドで指定した接続グループ名は登録されていません。	標準エラー出力

# 8

## 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。

## 8.1 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。

### 8.1.1 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、「[8.2 MCF メイン関数の作成](#)」を参照してください。

### 8.1.2 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。

### 8.1.3 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、「[システムサービス情報定義](#)」を参照してください。

### 8.1.4 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、「[8.3 定義オブジェクトファイルの生成](#)」を参照してください。

## 8.2 MCF メイン関数の作成

TP1/NET/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 8-1、図 8-2 に示します。また、ディレクトリへの組み込み方法を図 8-3 に示します。

なお、こ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 8-1 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(dc_mcf_uoc_min_n *): /*入力メッセージ編集UOC関数extern宣言 */
extern DCLONG msgsend01(dc_mcf_uoc_mout_n *): /*出力メッセージ編集UOC関数extern宣言 */ 2.

extern dc_mcf_uoc_t  dc_mcf_uoctbl;          /*UOCテーブルextern宣言          */ 3.

int main()
{
    dc_mcf_uoctbl.msgrcv = (dc_mcf_uocfunc)msgrcv01;
                                /*入力メッセージ編集UOCアドレス設定          */
    dc_mcf_uoctbl.msgsend = (dc_mcf_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 8-2 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。

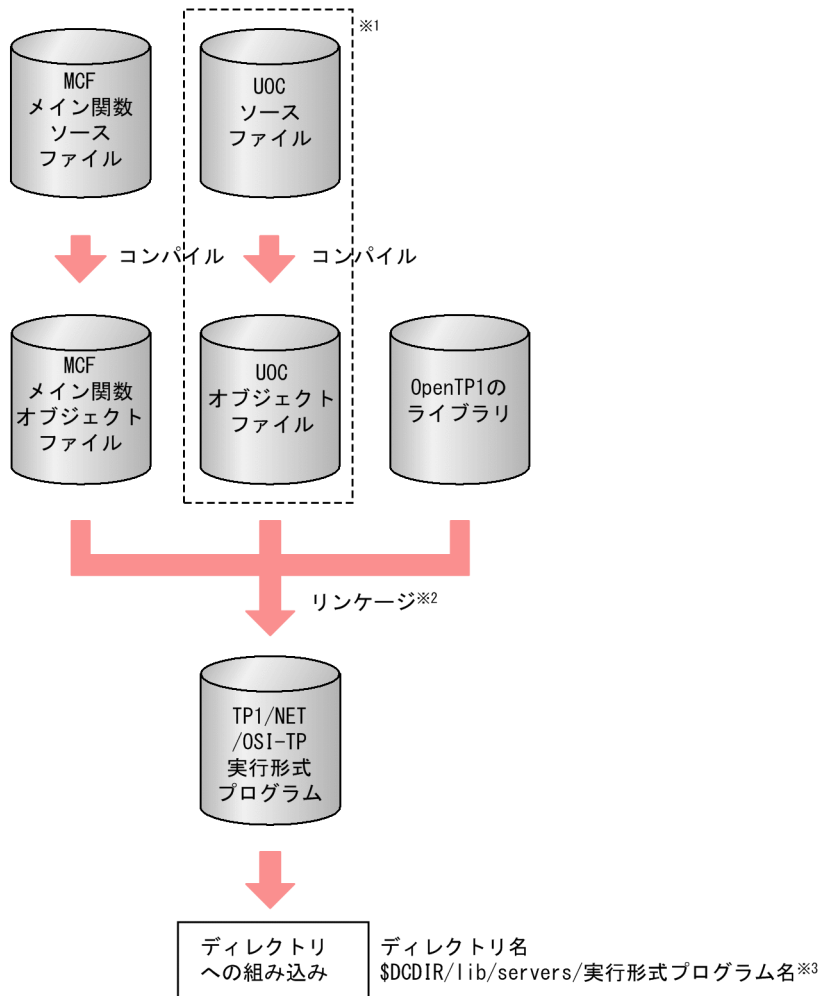
```

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。  
スタート関数を呼び出したあとは、MCF メイン関数に制御が戻りません。そのため、スタート関数のあとにコーディングした処理は実行されませんので注意してください。

図 8-3 MCF メイン関数のディレクトリへの組み込み方法の概要



注※1

UOC を使用しない場合は、必要ありません。

注※2

mcfpltp コマンドでリンケージします。

mcfpltp コマンドの詳細については、TP1/NET/OSI-TP の「リリースノート」を参照してください。

注※3

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。

## 8.3 定義オブジェクトファイルの生成

---

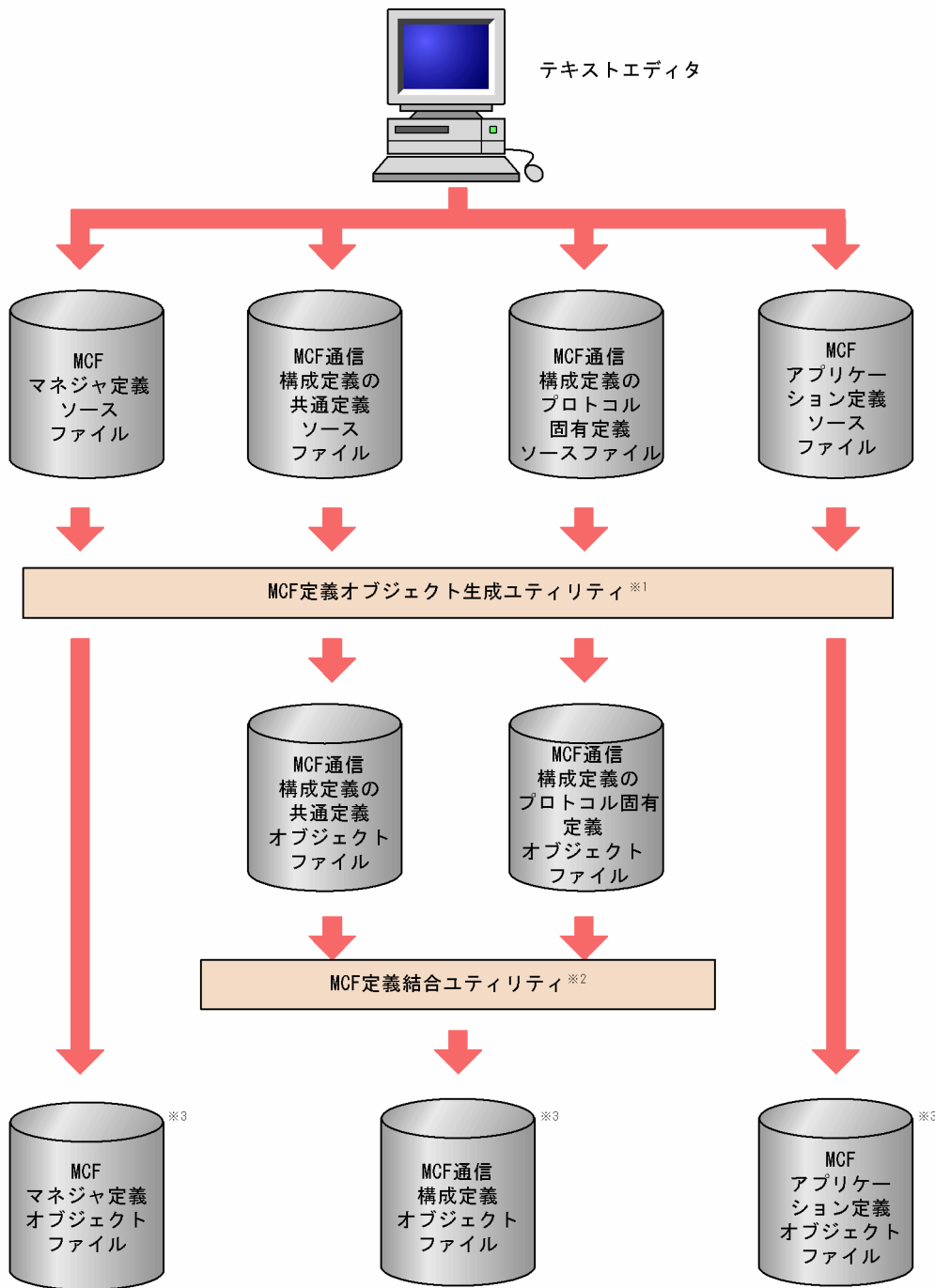
定義オブジェクトファイルを次の手順で生成します。

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

1. 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 8-4 定義オブジェクトファイルの作成方法の概要



注※1

次に示すコマンドで生成します。

```

mcfXXXXΔ-iΔ [パス名] 入力ファイル名
          Δ-oΔ [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。

- mcfmng: MCF マネージャ定義のソースファイル
- mcfcom: 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「[MCF 定義オブジェクトの生成](#)」を、その他のコマンドについてはマニュアル「[OpenTP1 システム定義](#)」を参照してください。

#### 注※2

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

```
mcf link  $\Delta$ -i  $\Delta$  共通定義オブジェクトファイル名  
           $\Delta$  TP1/NET/OSI-TP 定義オブジェクトファイル名  
           $\Delta$ -o  $\Delta$  出力オブジェクトファイル名
```

#### 注※3

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

# 9

## 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。

## 9.1 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 I 理由コード一覧」を参照してください。

### 9.1.1 コネクション障害

表 9-1 コネクション障害と対応処理

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"> <li>1. TP-P-ABORTind (recvsync 中) を通知します。</li> <li>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。</li> <li>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。</li> </ol>	<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> <p>発呼時</p> <p>運用コマンド (mcftactcn) を入力するか，API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△')) を発行するかしてコネクションを確立します。</p> <p>着呼時</p> <p>コネクションの確立を待ちます。</p>
TP プロトコル 障害	<ol style="list-style-type: none"> <li>1. TP-P-ABORTind (recvsync 中) を通知します。</li> <li>2. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> </ol>	<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> <p>発呼時</p> <p>運用コマンド (mcftactcn) を入力するか，API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△')) を発行するかしてコネクションを確立します。</p>

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
TP プロトコル障害	<ol style="list-style-type: none"> <li>1. TP-P-ABORTind (recvsync 中) を通知します。</li> <li>2. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> </ol>	<p>着呼時</p> <p>コネクションの確立を待ちます。</p>

## 9.1.2 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 9-2 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"> <li>1. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> <li>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。</li> </ol>	<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> <p>発呼時</p> <p>運用コマンド (mcfactcn) を入力するか、API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。</p> <p>着呼時</p> <p>コネクションの確立を待ちます。</p>
TP-BEGIN-DIALOGUEind 以外の UOC エラーリターン	<ol style="list-style-type: none"> <li>1. TP-P-ABORTind (recvsync 中) を通知します。</li> <li>2. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> <li>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。</li> </ol>	<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> <p>発呼時</p> <p>運用コマンド (mcfactcn) を入力するか、API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。</p> <p>着呼時</p> <p>コネクションの確立を待ちます。</p>
UOC パラメタ指定誤り	<ol style="list-style-type: none"> <li>1. TP-P-ABORTind (recvsync 中) を通知します。</li> <li>2. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> <li>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。</li> </ol>	<p>UOC を修正後、次に示す処理をします。</p> <p>発呼時</p> <p>運用コマンド (mcfactcn) を入力するか、API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。</p> <p>着呼時</p> <p>コネクションの確立を待ちます。</p>

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
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"> <li>1. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> <li>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。</li> </ol>	<p>システム定義を修正後、次に示す処理をします。</p> <p>発呼時</p> <p>運用コマンド (mcftactcn) を入力するか、API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。</p> <p>着呼時</p> <p>コネクションの確立を待ちます。</p>
MCF イベント名未定義 ・ ERREVTn ・ CXXX EVT	処理を続行します。	ありません。
入力キュー書き込み障害 スケジュール障害 アプリケーション閉塞 MHP サービス、サービスグループ閉塞	<ol style="list-style-type: none"> <li>1. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> <li>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。</li> <li>5. ERREVT2 を起動します。</li> </ol>	<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> <p>発呼時</p> <p>運用コマンド (mcftactcn) を入力するか、API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。</p> <p>着呼時</p> <p>コネクションの確立を待ちます。</p>
受信バッファオーバーフロー	<ol style="list-style-type: none"> <li>1. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>3. UAP にエラーリターンします。</li> <li>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。</li> </ol>	<p>システム定義を修正後、次に示す処理をします。</p> <p>発呼時</p> <p>運用コマンド (mcftactcn) を入力するか、API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。</p> <p>着呼時</p> <p>コネクションの確立を待ちます。</p>
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"> <li>1. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> </ol>	<p>システム定義を修正後、次に示す処理をします。</p> <p>発呼時</p> <p>運用コマンド (mcftactcn) を入力するか、API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。</p>

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
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 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。	△)) を発行するか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。

### 9.1.3 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 9-3 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	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。	障害の要因を取り除いたあと, 次に示す処理をします。 発呼時 運用コマンド (mcftactcn) を入力するか, API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するか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	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。	UOC を修正後, 次に示す処理をします。 発呼時 運用コマンド (mcftactcn) を入力するか, API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するか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フロー	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。	システム定義を修正後, 次に示す処理をします。 発呼時 運用コマンド (mcftactcn) を入力するか, API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。

障害の内容	TP1/NET/OSI-TP の処理	ユーザの処理
送信バッファオーバーフロー	<ol style="list-style-type: none"> <li>1. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> </ol>	<p>着呼時</p> <p>コネクションの確立を待ちます。</p>
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"> <li>1. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> </ol> <p>処理 2</p> <ol style="list-style-type: none"> <li>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。</li> <li>2. MCF プロセスを異常終了します。</li> </ol>	<p>システム定義を修正後、次に示す処理をします。</p> <p>発呼時</p> <p>運用コマンド (mcfactcn) を入力するか、API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。</p> <p>着呼時</p> <p>コネクションの確立を待ちます。</p>

## 9.1.4 UAP 障害

表 9-4 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)	<ol style="list-style-type: none"> <li>1. TP-ABORT-RI (provider) を送信します。</li> <li>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。</li> <li>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。</li> <li>4. ERREVT3 を起動します。</li> </ol>	<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> <p>発呼時</p> <p>運用コマンド (mcfactcn) を入力するか、API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。</p> <p>着呼時</p> <p>コネクションの確立を待ちます。</p>
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。

## 9.1.5 ジャーナル障害 (IJ, MJ)

表 9-5 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。

## 9.1.6 関数のリターン処理障害

表 9-6 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	<ol style="list-style-type: none"><li>1. TP-ABORT-RI (provider) を送信します。</li><li>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。</li><li>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。</li><li>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。</li><li>5. UAP (XXXXsync()) にエラーリターンします。</li></ol>	<p>発呼時</p> <p>運用コマンド (mcftactcn) を入力するか、API (dc_mcf_tactcn 関数または CBLDCMCF('TACTCN△△')) を発行するかしてコネクションを確立します。</p> <p>着呼時</p> <p>コネクションの確立を待ちます。</p>

## 9.1.7 プロシジャ障害

表 9-7 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	<ol style="list-style-type: none"><li>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。</li><li>2. 必要に応じて MCF を異常終了させます。</li></ol>	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	<ol style="list-style-type: none"><li>1. OS によるコア出力をします。</li><li>2. 通信管理によるコネクション解放をします。</li></ol>	

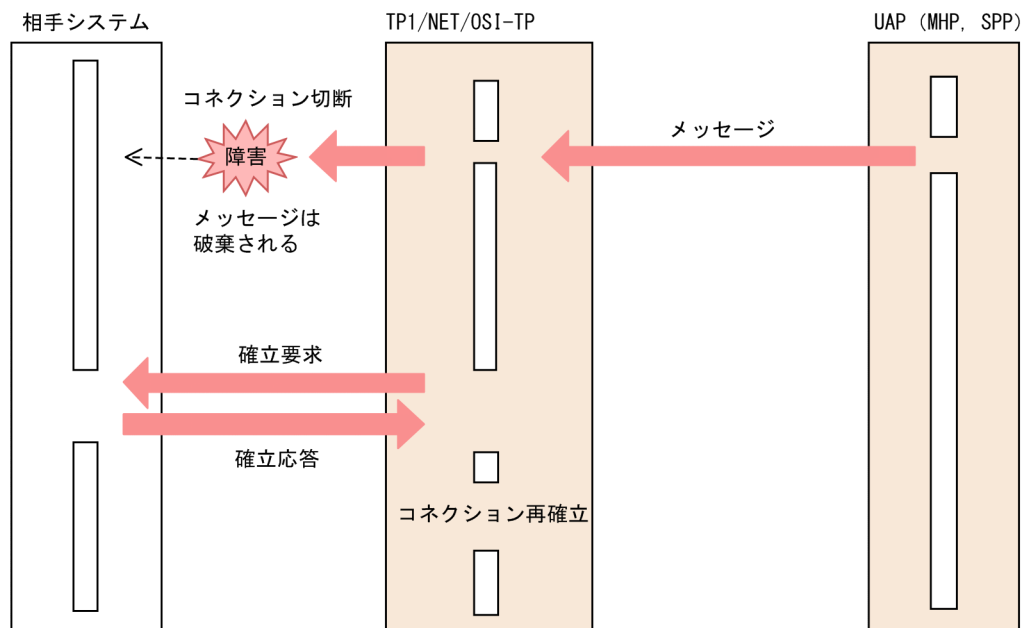
## 9.2 コネクション障害

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 (mcftactcn) を入力するか、API (dc\_mcf\_tactcn 関数または CBLDCMCF("TACTCN△△")) を発行するか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 9-1 メッセージ送受信時のコネクション障害

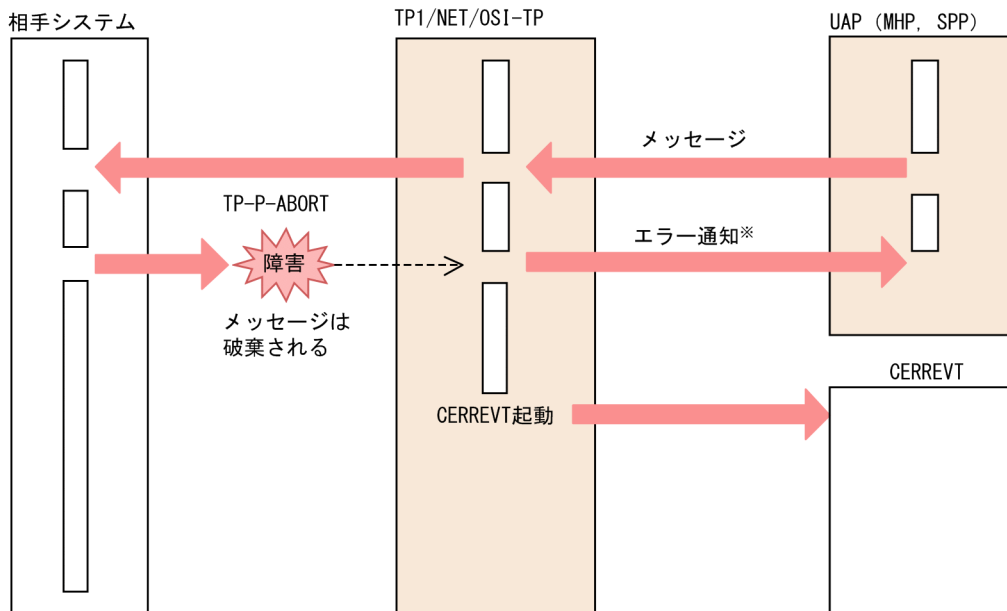


## 9.3 ダイアログ障害

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 9-2 ダイアログ障害



注※

エラー通知 (TP-P-ABORT ind) 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。

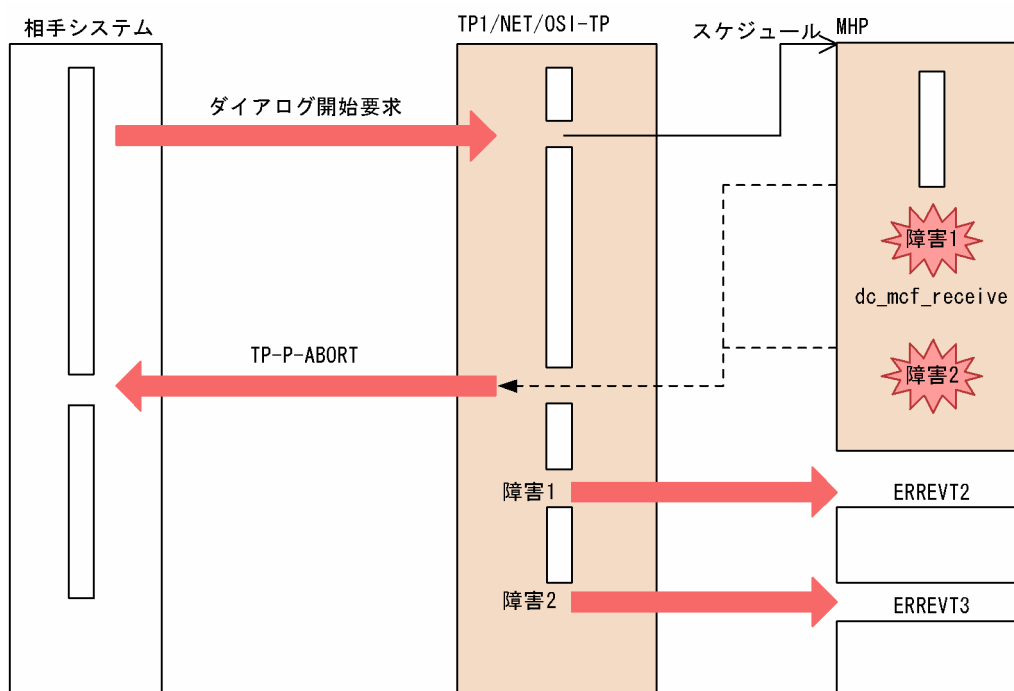
## 9.4 ユーザアプリケーションプログラム異常終了

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。

### 9.4.1 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 9-3 ダイアログ開始要求受信時の MHP 異常終了



### 9.4.2 ダイアログ中の UAP (MHP, SPP) 終了

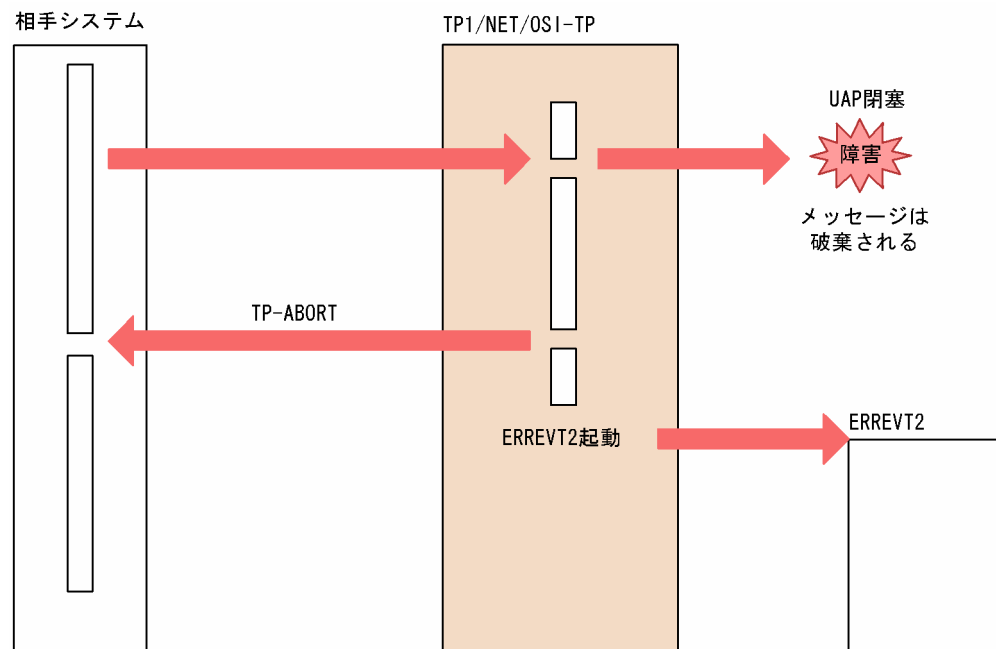
ダイアログ中に UAP (MHP, SPP) が終了 (正常・異常) した場合、TP1/NET/OSI-TP は接続を切断 (TP-P-ABORT) し、CERREVT を起動します。

## 9.5 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 9-4 UAP 閉塞による障害



# 付録

## 付録 A 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・変更・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 A.1 07-50 での変更点

TP1/NET/OSI-TP 07-50 での関数，定義およびコマンドの追加・変更・削除を次の表に示します。

表 A-1 TP1/NET/OSI-TP 07-50 での関数，定義およびコマンドの追加・変更・削除

種別	分類	内容
追加	関数	RECEIVE – メッセージの受信 <ul style="list-style-type: none"> <li>• SYNCHRONOUS MODE 句の設定値に ASYNC を追加</li> <li>• SYNCHRONOUS MODE 句のデータ名 6 の設定値に 0，空白を追加</li> </ul>
		SEND – メッセージの送信 <ul style="list-style-type: none"> <li>• WITH 句</li> </ul>
	定義	なし
	コマンド	mcfostp コマンド <ul style="list-style-type: none"> <li>• -r オプション</li> </ul>
変更		なし
削除	関数	なし
	定義	コネクション定義の開始 (mcftalccn) <ul style="list-style-type: none"> <li>• -z オプションの dtea オペランド</li> <li>• -z オプションの alt1_dtea オペランド</li> <li>• -z オプションの alt2_dtea オペランド</li> <li>• -z オプションの alt3_dtea オペランド</li> <li>• -z オプションの alt4_dtea オペランド</li> <li>• -z オプションの alt5_dtea オペランド</li> <li>• -z オプションの alt6_dtea オペランド</li> <li>• -z オプションの alt7_dtea オペランド</li> <li>• -z オプションの alt1_slot オペランド</li> <li>• -z オプションの alt2_slot オペランド</li> <li>• -z オプションの alt3_slot オペランド</li> <li>• -z オプションの alt4_slot オペランド</li> <li>• -z オプションの alt5_slot オペランド</li> <li>• -z オプションの alt6_slot オペランド</li> </ul>

種別	分類	内容
削除	定義	• -z オプションの alt7_slot オペランド
	コマンド	なし

TP1/NET/OSI-TP 07-50 での動作の変更点はありません。

TP1/NET/OSI-TP 07-50 でのデフォルト値の変更はありません。

## 付録 A.2 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・変更・削除はありません。

TP1/NET/OSI-TP 07-00 での動作の変更点はありません。

TP1/NET/OSI-TP 07-00 でのデフォルト値の変更はありません。

## 付録 B 旧製品からの移行に関する注意事項

旧製品から移行する場合の注意事項を示します。

### 付録 B.1 ソースの互換性

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

表 B-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

ソースファイルの種類	ソースファイルを作成した言語	互換性
UAP	C 言語	ソースファイルを変更しないで使用できます。*
	COBOL 言語	ソースファイルを変更しないで使用できます。
UOC	C 言語	ソースファイルを変更しないで使用できます。*
MCF 通信構成定義 (プロトコル固有の定義)	—	ソースファイルを変更しないで使用できます。

(凡例)

— : 該当する内容がないことを表します。

注※

バージョン 7 では、メッセージ送受信インタフェース、UOC、および MCF イベントインタフェースのそれぞれの引数ならびにパラメタの型が変更されていますが、この変更による UAP や UOC の処理への影響はありません。

詳細については、「付録 C インタフェースの変更一覧 (バージョン 6 以前から移行する場合)」を参照してください。

## 付録 C インタフェースの変更一覧 (バージョン 6 以前から移行する場合)

バージョン 6 以前のインタフェースの変更一覧を示します。

ここで説明するインタフェースを次に示します。

表 C-1 インタフェースの変更一覧

変更されたインタフェース	バージョン 7 のマニュアルの該当箇所
メッセージ送受信インタフェース	dc_mcf_receive 3. C 言語のライブラリ関数 <a href="#">dc_mcf_receive</a> – 一方送信メッセージの受信 (C 言語)
	dc_mcf_recvsync 3. C 言語のライブラリ関数 <a href="#">dc_mcf_recvsync</a> – 同期型メッセージの受信 (C 言語)
	dc_mcf_sendrecv 3. C 言語のライブラリ関数 <a href="#">dc_mcf_sendrecv</a> – 同期型メッセージの送受信 (C 言語)
	dc_mcf_sendsync 3. C 言語のライブラリ関数 <a href="#">dc_mcf_sendsync</a> – 同期型メッセージの送信 (C 言語)
ユーザOWNコーディング	入力メッセージ編集 UOC 5.1.2 入力メッセージ編集 UOC インタフェース
	出力メッセージ編集 UOC 5.1.4 出力メッセージ編集 UOC インタフェース
MCF イベントインタフェース	5.2.3 MCF イベント情報の形式 (C 言語)
MCF メイン関数のコーディング概要	8.2 MCF メイン関数の作成

以降、バージョン 6 以前のインタフェースと、バージョン 7 のインタフェースの変更一覧を示します。変更箇所には、下線を付与しています。

### 付録 C.1 メッセージ送受信インタフェース

#### (1) dc\_mcf\_receive – 一方送信メッセージの受信

##### (a) ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre>#include &lt;dcpcf.h&gt; int dc_mcf_receive(long action,                   long commform,                   char *termnam,                   char *resv01,                   char *recvdata,                   long *rdataleng,                   long inbufleng,                   long *time)</pre>	<pre>#include &lt;dcpcf.h&gt; int dc_mcf_receive(<u>DCLONG</u> action,                   <u>DCLONG</u> commform,                   char *termnam,                   char *resv01,                   char *recvdata,                   <u>DCLONG</u> *rdataleng,                   <u>DCLONG</u> inbufleng,                   <u>DCLONG</u> *time)</pre>

## (b) K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre>#include &lt;dcmf.h&gt; int dc_mcf_receive(action,                   commform,                   termnam,                   resv01,                   recvdata,                   rdataleng,                   inbufleng,                   time)  long    action; long    commform; char    *termnam; char    *resv01; char    *recvdata; long    *rdataleng; long    inbufleng; long    *time;</pre>	<pre>#include &lt;dcmf.h&gt; 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;</pre>

## (2) dc\_mcf\_recvsync – 同期型メッセージの受信

### (a) ANSI C, C++ の形式

バージョン 6 以前	バージョン 7
<pre>#include &lt;dcmf.h&gt; int dc_mcf_recvsync(long action,                    long commform,                    char *termnam,                    char *resv01,                    char *recvdata,                    long *rdataleng,                    long inbufleng,                    long *time,                    long watchtime)</pre>	<pre>#include &lt;dcmf.h&gt; int dc_mcf_recvsync(DCLONG action,                    DCLONG commform,                    char *termnam,                    char *resv01,                    char *recvdata,                    DCLONG *rdataleng,                    DCLONG inbufleng,                    DCLONG *time,                    DCLONG watchtime)</pre>

### (b) K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre>#include &lt;dcmf.h&gt; int dc_mcf_recvsync(action,                   commform,                   termnam,                   resv01,                   recvdata,                   rdataleng,                   inbufleng,                   time,                   watchtime)  long    action; long    commform; char    *termnam; char    *resv01; char    *recvdata; long    *rdataleng; long    inbufleng;</pre>	<pre>#include &lt;dcmf.h&gt; 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;</pre>

バージョン 6 以前	バージョン 7
<pre> long      *time; long      watchtime; </pre>	<pre> DCLONG   *time; DCLONG   watchtime; </pre>

### (3) dc\_mcf\_sendrecv – 同期型メッセージの送受信

#### (a) ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre> #include &lt;dcmcf.h&gt; int dc_mcf_sendrecv(long action,                     long commform,                     char *termnam,                     char *resv01,                     char *senddata,                     long sdataleng,                     char *recvdata,                     long *rdataleng,                     long inbufleng,                     long *time,                     long watchtime) </pre>	<pre> #include &lt;dcmcf.h&gt; 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) </pre>

#### (b) K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre> #include &lt;dcmcf.h&gt; int dc_mcf_sendrecv(action,                     commform,                     termnam,                     resv01,                     senddata,                     sdataleng,                     recvdata,                     rdataleng,                     inbufleng,                     time,                     watchtime)  long      action; long      commform; char      *termnam; char      *resv01; char      *senddata; long      sdataleng; char      *recvdata; long      *rdataleng; long      inbufleng; long      *time; long      watchtime; </pre>	<pre> #include &lt;dcmcf.h&gt; 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; </pre>

## (4) dc\_mcf\_sendsync – 同期型メッセージの送信

### (a) ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre>#include &lt;dcpcf.h&gt; int dc_mcf_sendsync(long action,                     long commform,                     char *termnam,                     char *resv01,                     char *senddata,                     long sdataleng,                     char *resv02,                     long opcd,                     long watchtime)</pre>	<pre>#include &lt;dcpcf.h&gt; int dc_mcf_sendsync(DCLONG action,                     DCLONG commform,                     char *termnam,                     char *resv01,                     char *senddata,                     DCLONG sdataleng,                     char *resv02,                     DCLONG opcd,                     DCLONG watchtime)</pre>

### (b) K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre>#include &lt;dcpcf.h&gt; int dc_mcf_sendsync(action,                     commform,                     termnam,                     resv01,                     senddata,                     sdataleng,                     resv02,                     opcd,                     watchtime)  long    action; long    commform; char    *termnam; char    *resv01; char    *senddata; long    sdataleng; char    *resv02; long    opcd; long    watchtime;</pre>	<pre>#include &lt;dcpcf.h&gt; 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;</pre>

## 付録 C.2 ユーザOWNコーディング

### (1) 入力メッセージ編集 UOC

#### (a) 形式

- ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre>#include &lt;dcpcf.h&gt; #include &lt;dcmotm.h&gt; #include &lt;dcpcfuo.h&gt; long uoc_func(dcpcf_uoc_min_n *parm)</pre>	<pre>#include &lt;dcpcf.h&gt; #include &lt;dcmotm.h&gt; #include &lt;dcpcfuo.h&gt; DCLONG uoc_func(dcpcf_uoc_min_n *parm)</pre>

- K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre>#include &lt;dcpcf.h&gt; #include &lt;dcmotm.h&gt; #include &lt;dcpcfuc.h&gt; long uoc_func(parm)  dcpcf_uoc_min_n *parm ;</pre>	<pre>#include &lt;dcpcf.h&gt; #include &lt;dcmotm.h&gt; #include &lt;dcpcfuc.h&gt; DCLONG uoc_func(parm)  dcpcf_uoc_min_n *parm ;</pre>

## (b) パラメタの内容

- dcmcf\_uoc\_min\_n の内容

バージョン 6 以前	バージョン 7
<pre>typedef struct {     long pro_kind;     char le_name[9];     char reserve1[7];     long rcv_prim;     dcmcf_uocbuff_list_n *buflist_adr;     dcmcf_uocbuff_list_n *ebuflist_adr;     char aplname[9];     char reserve2[7];     char *pro_indv_ifa;     long rtn_detail;     char reserve3[8]; } dcmcf_uoc_min_n;</pre>	<pre>typedef struct {     DCLONG pro_kind;     char le_name[9];     char reserve1[7];     DCLONG rcv_prim;     dcmcf_uocbuff_list_n *buflist_adr;     dcmcf_uocbuff_list_n *ebuflist_adr;     char aplname[9];     char reserve2[7];     char *pro_indv_ifa;     DCLONG rtn_detail;     char reserve3[8]; } dcmcf_uoc_min_n;</pre>

- dcmcf\_uocbuff\_list\_n (バッファリスト) の内容

バージョン 6 以前	バージョン 7
<pre>typedef struct {     long buf_num;     long used_buf_num;     char reserve1[8];     dcmcf_uocbufinf_n         buf_array[DCMCF_UOC_BUFF_MAX]; } dcmcf_uocbuff_list_n;</pre>	<pre>typedef struct {     DCLONG buf_num;     DCLONG used_buf_num;     char reserve1[8];     dcmcf_uocbufinf_n         buf_array[DCMCF_UOC_BUFF_MAX]; } dcmcf_uocbuff_list_n;</pre>

- dcmcf\_uocbufinf\_n (バッファ情報) の内容

バージョン 6 以前	バージョン 7
<pre>typedef struct {     char *buf_adr;     unsigned long buf_size;     unsigned long seg_size;     char reserve1[4];     dcmcfuc_w_type buff_id;     long buff_addr;     char reserve2[4]; } dcmcf_uocbufinf_n;</pre>	<pre>typedef struct {     char *buf_adr;     DCULONG buf_size;     DCULONG seg_size;     char reserve1[4];     dcmcfuc_w_type buff_id;     DCMLONG buff_addr;     char reserve2[4]; } dcmcf_uocbufinf_n;</pre>

## (2) 出力メッセージ編集 UOC

### (a) 形式

- ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre>#include &lt;dcmcf.h&gt; #include &lt;dcmotm.h&gt; #include &lt;dcmcfuoc.h&gt; long uoc_func(dcmcf_uoc_mout_n *parm)</pre>	<pre>#include &lt;dcmcf.h&gt; #include &lt;dcmotm.h&gt; #include &lt;dcmcfuoc.h&gt; DCLONG uoc_func(dcmcf_uoc_mout_n *parm)</pre>

- K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre>#include &lt;dcmcf.h&gt; #include &lt;dcmotm.h&gt; #include &lt;dcmcfuoc.h&gt; long uoc_func(parm)  dcmcf_uoc_mout_n *parm ;</pre>	<pre>#include &lt;dcmcf.h&gt; #include &lt;dcmotm.h&gt; #include &lt;dcmcfuoc.h&gt; DCLONG uoc_func(parm)  dcmcf_uoc_mout_n *parm ;</pre>

### (b) パラメタの内容

- dcmcf\_uoc\_mout\_n の内容

バージョン 6 以前	バージョン 7
<pre>typedef struct {     long pro_kind;     char le_name[9];     char reserve1[7];     dcmcf_uocbuff_list_n *buflist_adr;     dcmcf_uocbuff_list_n *ebuflist_adr;     long output_no;     char msg_type;     char outputno_flag;     char resend_flag;     char reserve2[1];     char *pro_indv_ifa;     long rtn_detail;     char reserve3[20]; } dcmcf_uoc_mout_n;</pre>	<pre>typedef struct {     DCLONG pro_kind;     char le_name[9];     char reserve1[7];     dcmcf_uocbuff_list_n *buflist_adr;     dcmcf_uocbuff_list_n *ebuflist_adr;     DCLONG output_no;     char msg_type;     char outputno_flag;     char resend_flag;     char reserve2[1];     char *pro_indv_ifa;     DCLONG rtn_detail;     char reserve3[20]; } dcmcf_uoc_mout_n;</pre>

- 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 のパラメタの内容と同じです。「付録 C.2(1)(b) パラメタの内容」を参照してください。

## 付録 C.3 MCF イベントインタフェース

### (1) MCF イベントの共通ヘッダの形式

バージョン 6 以前	バージョン 7
<pre>struct dc_mcf_evtheader {     char mcfevt_name[9];     char le_name[16];     char cn_name[9];     unsigned char format_kind;     char reserve01;     long time; };</pre>	<pre>struct dc_mcf_evtheader {     char mcfevt_name[9];     char le_name[16];     char cn_name[9];     unsigned char format_kind;     char reserve01;     DCLONG time; };</pre>

### (2) ERREVT1 の形式

バージョン 6 以前とバージョン 7 で、差異はありません。

### (3) ERREVT2 の形式

バージョン 6 以前とバージョン 7 で、差異はありません。

### (4) ERREVT3 の形式

バージョン 6 以前とバージョン 7 で、差異はありません。

### (5) ERREVTA の形式

バージョン 6 以前	バージョン 7
<pre>struct dc_mcf_evta_type {     struct dc_mcf_evtheader evtheader;     char reserve01[12];     char reserve02[10];     char reserve03[2];     char ap_name[10];     char reserve04[2];     char reserve05[32];     char reserve06[32];     long user_leng;     char user_data[16];     char reserve07[16]; };</pre>	<pre>struct dc_mcf_evta_type {     struct dc_mcf_evtheader evtheader;     char reserve01[12];     char map_name[10];     char reserve03[2];     char ap_name[10];     char reserve04[2];     char reserve05[32];     char reserve06[32];     DCLONG user_leng;     char user_data[16];     char reserve07[16]; };</pre>

### (6) CERREVT の形式

バージョン 6 以前	バージョン 7
<pre>typedef struct {     struct dc_mcf_evtheader header;     long err_fact;     long err_reason1; };</pre>	<pre>typedef struct {     struct dc_mcf_evtheader header;     DCLONG err_fact;     DCLONG err_reason1; };</pre>

バージョン 6 以前	バージョン 7
<pre> long err_reason2; long err_rcv_action; char group_name[16]; char reserve1[26]; } dcmotm_cerrevt; </pre>	<pre> DCLONG err_reason2; DCLONG err_rcv_action; char group_name[16]; char reserve1[26]; } dcmotm_cerrevt; </pre>

## (7) COPNEVT, CCLSEVT の形式

バージョン 6 以前	バージョン 7
<pre> typedef struct {     struct dc_mcf_evtheader header;     long cls_rcv_action;     char reserve1[12];     char group_name[16];     char reserve2[26]; } dcmotm_statevt; </pre>	<pre> typedef struct {     struct dc_mcf_evtheader header;     DCLONG cls_rcv_action;     char reserve1[12];     char group_name[16];     char reserve2[26]; } dcmotm_statevt; </pre>

## 付録 C.4 MCF メイン関数のコーディング概要

MCF メイン関数のコーディング概要の変更一覧を示します。変更箇所は、図中の網掛け部分です。

### (1) ANSI C, C++の場合

#### (a) バージョン 6 以前

```

#include <dcmotp.h>          /*TP1/NET/OSI-TP用ヘッダファイル      */
extern long  msgrcv01(dcmcf_uoc_min_n *); /*入力メッセージ編集UOC関数extern宣言 */
extern long  msgsend01(dcmcf_uoc_mout_n *); /*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      */

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

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

```

## (b) バージョン7

```
#include <dcmotp.h>                /*TP1/NET/OSI-TP用ヘッダファイル */
extern DCLONG msgrcv01(dcmcf_uoc_min_n *); /*入力メッセージ編集UOC関数extern宣言 */
extern DCLONG msgsend01(dcmcf_uoc_mout_n *); /*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 */

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

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

## (2) K&R 版 C の場合

### (a) バージョン6以前

```
#include <dcmotp.h>                /*TP1/NET/OSI-TP用ヘッダファイル */
extern long msgrcv01();           /*入力メッセージ編集UOC関数extern宣言 */
extern long 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 */

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

### (b) バージョン7

```
#include <dcmotp.h>                /*TP1/NET/OSI-TP用ヘッダファイル */
extern DCLONG msgrcv01();         /*入力メッセージ編集UOC関数extern宣言 */
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 */

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

## 付録 D 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。

### 付録 D.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

### 付録 D.2 サービスプリミティブ

表 D-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	○	○	○	○	• TP-BEGIN-DIALOGUE-RI • TP-BEGIN-DIALOGUE-RC
TP-END-DIALOGUE	○	○	○	○	• TP-END-DIALOGUE-RI • TP-END-DIALOGUE-RC
TP-U-ERROR	○	○	—	—	• TP-U-ERROR-RI • TP-U-ERROR-RC
TP-U-ABORT	○	○	—	—	TP-U-ABORT-RI
TP-P-ABORT	—	○	—	—	TP-P-ABORT-RI
TP-HANDSHAKE	○	○	○	○	• 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	○	○	○	○	• 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。

## 付録 D.3 サービス詳細

表 D-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-AP-Title	－	－	－	－	B-Dreq 送信元の AEI に関する情報
	Initiating-API-Identifier	－	－	－	－	
	Initiating-AE-Qualifier	－	－	－	－	
	Initiating-AEI-Identifier	－	－	－	－	
	Initiating-TPSU-Title	●	△(=)	－	－	B-Dreq 送信元の TPSUI 識別情報
	Recipient-AP-Title	○	－	－	－	リモート TPSUI が存在する AEI に関する情報
	Recipient-API-Identifier	－	－	－	－	
	Recipient-AE-Qualifier	－	－	－	－	
	Recipient-AEI-Identifier	－	－	－	－	
	Recipient-TPSU-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-Service	○	－	－	－	サービス品質に関する情報
	Application-Context-name	○	－	－	－	アプリケーションコンテキスト名に関する情報

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)	Confirmation	◎	◎(=)	—	—	"always" (応答要) "negative" (拒否時だけ応答要)
	Result	—	—	◎*2	◎	"accepted" "rejected(provider)" "rejected(user)"
	Diagnostic	—	—	—	△*3	"recipient-unknown"*4 "recipient-tpsu-title-unknown" "tpsu-not-available(permanent)" "tpsu-not-available(transient)" "recipient-tpsu-title-required" "functional-unit-not-supported" "functional-unit-combination-not-supported" "no-reason-given"
	Rollback	—	—	—	◎	"false"
	User-Data	●	△(=)	●	△	ユーザ情報
TP-END-DIALOGUE (E-D)	Confirmation	◎	◎(=)	—	—	"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-reject" "transient-failure" "protocol-error" "end-dialogue-collision"
	Rollback	—	◎	—	—	"false"
TP-HANDSHAKE (TP-H)	Confirmation-Urgency	△*5	—	—	—	"urgent" (即時応答要) "normal" (遅延応答可)

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-GRANT-CONTROL(G-C)	—	—	—	—	—	—
TP-REQUEST-CONTROL(R-C)	—	—	—	—	—	—
TP-HANDSHAKE-AND-GRANT-CONTROL(H-G)	Confirmation-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

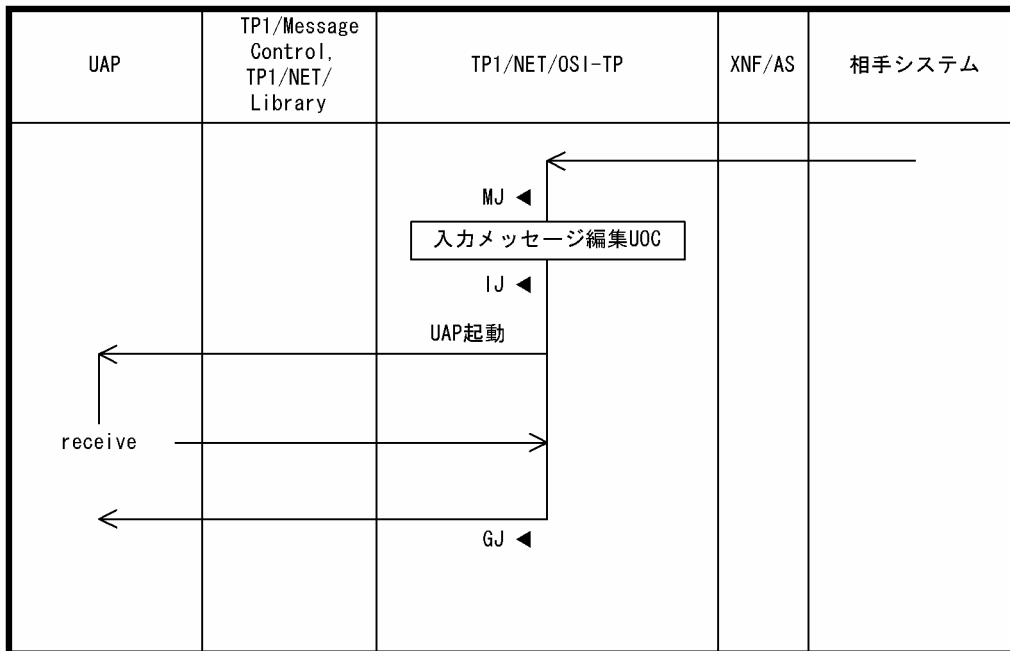
User-Data：ユーザデータ

Confirmation-Urgency：確認緊急性

## 付録 E 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。

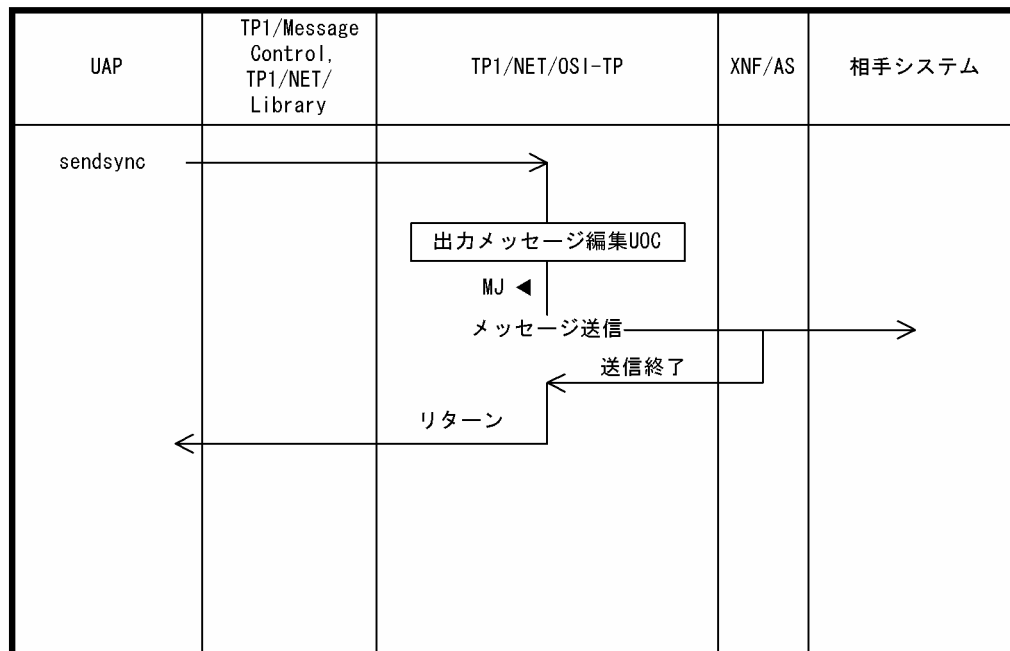
図 E-1 一方送信メッセージの受信



(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

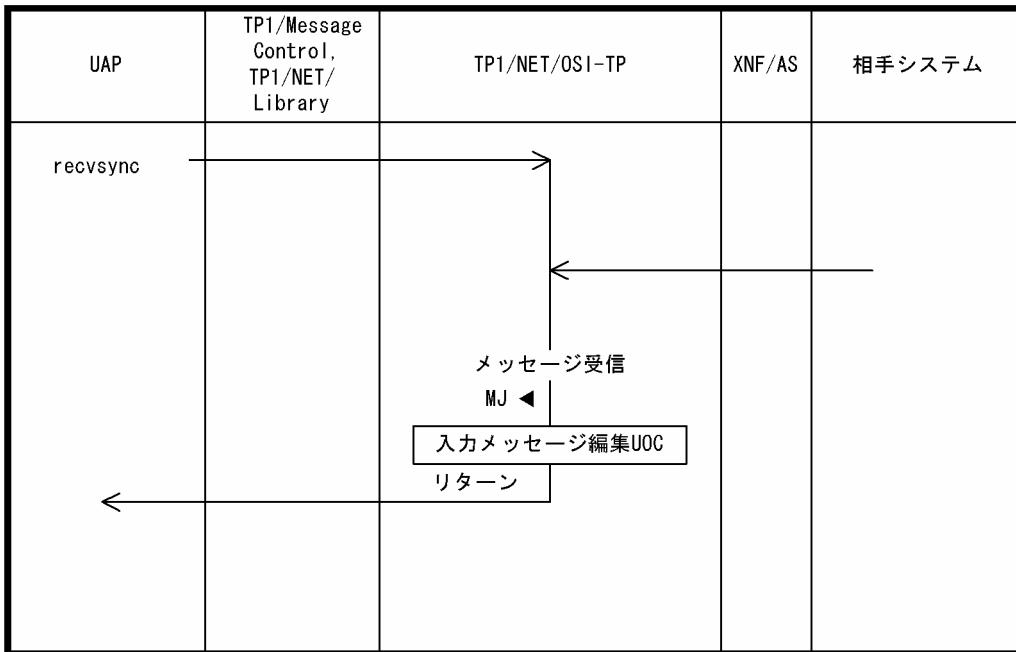
図 E-2 同期型のメッセージの送信



(凡例)

- MJ ◀: メッセージジャーナル取得

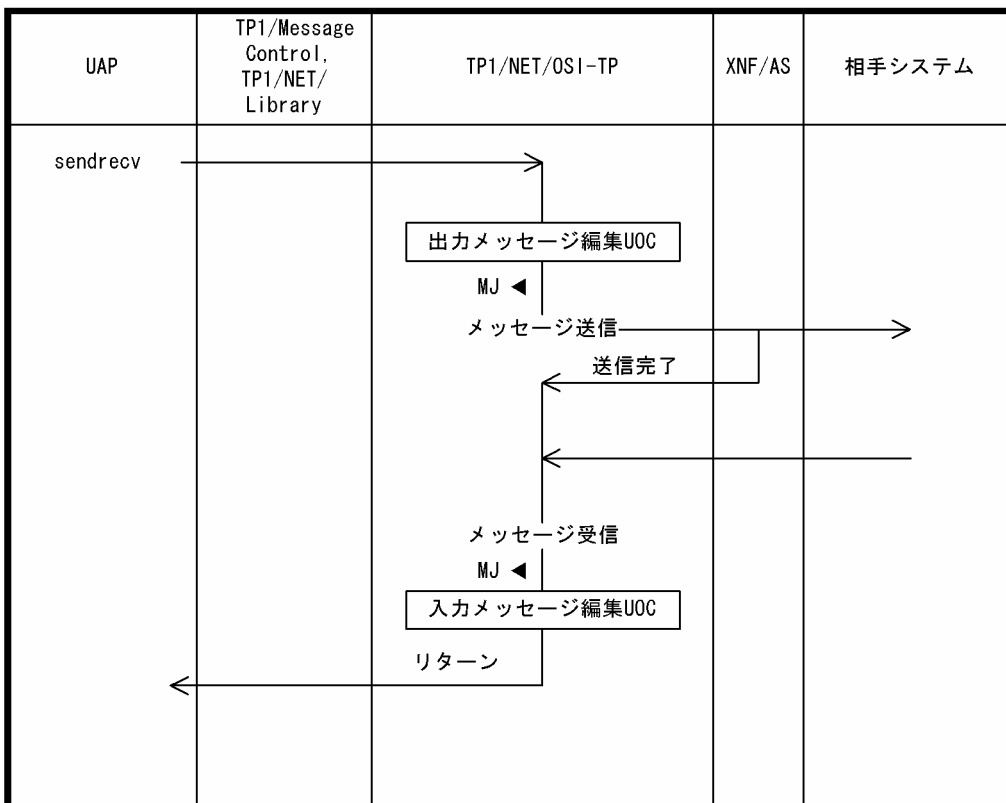
図 E-3 同期型のメッセージの受信



(凡例)

MJ ◀: メッセージジャーナル取得

図 E-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 F 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 F-1 通信回線障害

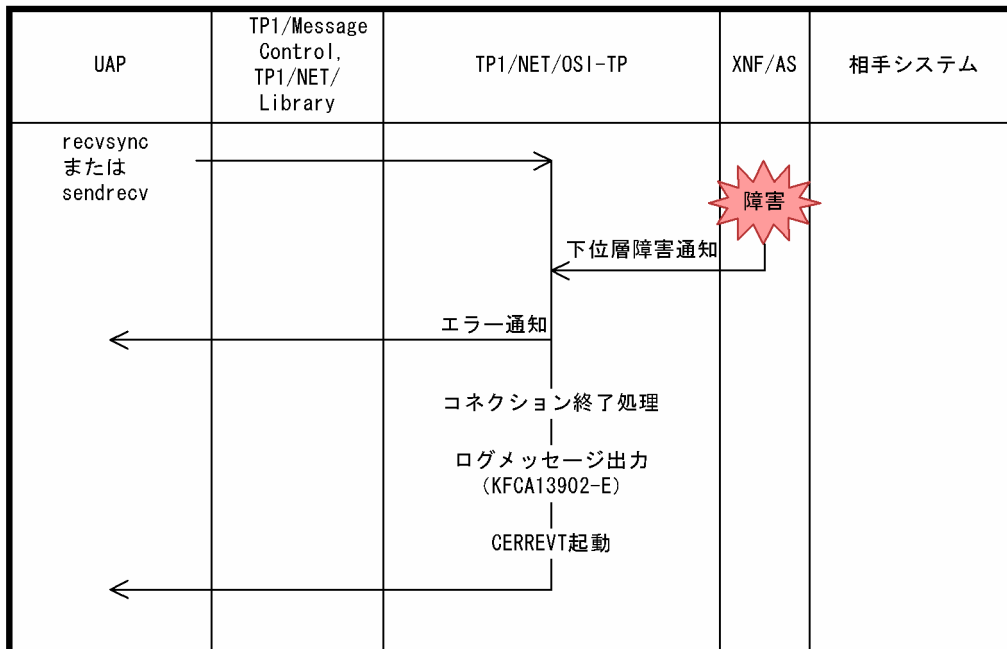
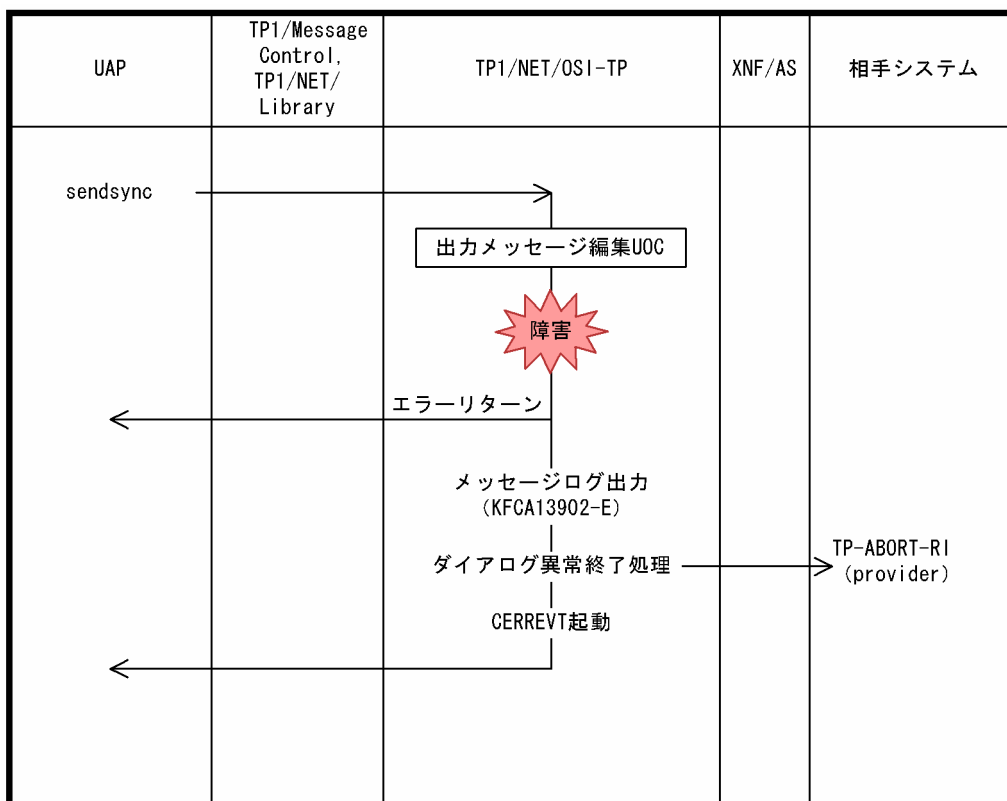


図 F-2 TP1/NET/OSI-TP での障害



## 付録 G MCF 性能検証用トレースの取得

TP1/Message Control を使用したメッセージ送受信での主なイベントで、MCF 識別子などのトレース情報を取得しています。これを MCF 性能検証用トレースと呼びます。

ここでは、MCF 性能検証用トレースの MCF 固有情報の出力情報、取得タイミング、および取得量について説明します。

### 付録 G.1 MCF 固有情報の出力情報

ここでは、UOC 呼び出し時の MCF 性能検証用トレースのダンプ出力情報について説明します。

#### (1) UOC 呼び出し時

TP1/NET/OSI-TP で使用する UOC の情報を取得します。MCF 固有情報のダンプ出力情報、および UOC 名称の出力情報を、以降の表に示します。

表 G-1 UOC 呼び出し時の MCF 固有情報のダンプ出力情報

イベント ID	オフセット				
	0x0000~ 0x0003	0x0004~ 0x0007	0x0008~ 0x000f	0x0010~ 0x0017	0x0018~0x001f
0xa070	MCF 識別子	スレッド ID	—	—	UOC 名称
0xa071			—	—	

(凡例)

—：情報を取得しません。

表 G-2 UOC 名称の出力情報

UOC の種類	UOC 名称出力情報
入力メッセージ編集 UOC	"MSGRCV"
出力メッセージ編集 UOC	"MSGSEND"

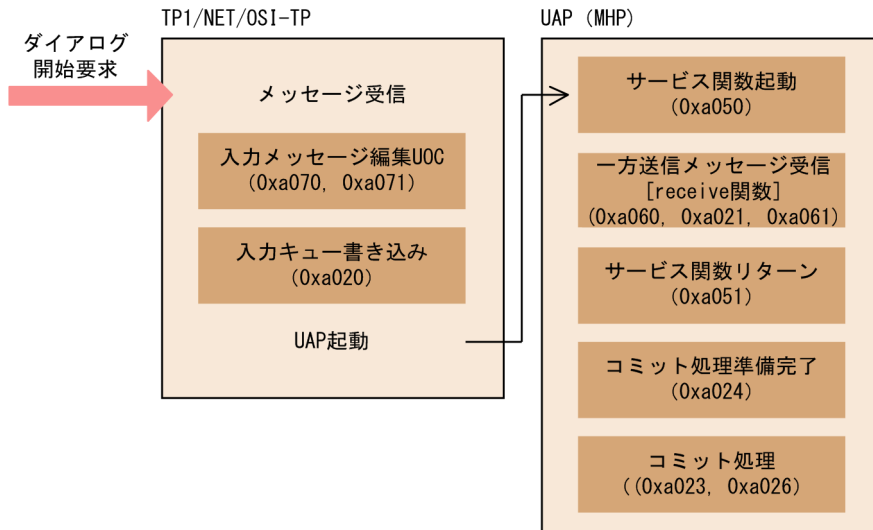
### 付録 G.2 MCF 性能検証用トレースの取得タイミング

MCF 性能検証用トレースの取得タイミングを、使用する送受信形態別に説明します。

#### (1) 一方送信メッセージ受信時

一方送信メッセージ受信時の MCF 性能検証用トレースの取得タイミングについて、次の図に示します。

図 G-1 一方送信メッセージ受信時の MCF 性能検証用トレースの取得タイミング

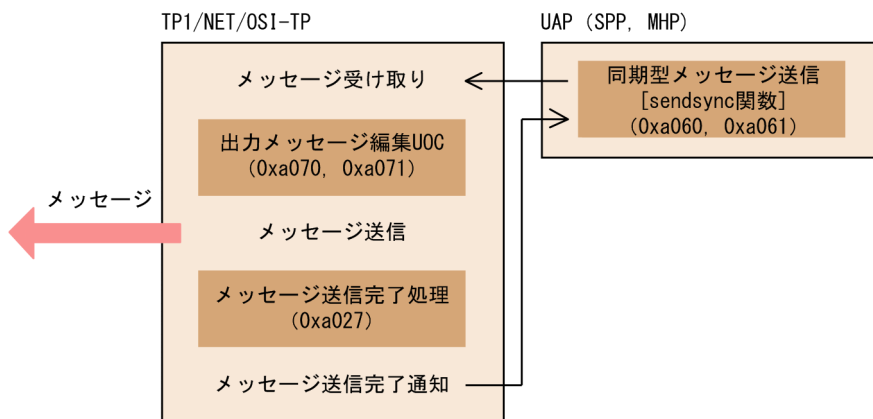


(凡例)  
 : MCF性能検証用トレースの取得タイミング  
 ( ) : イベントID

## (2) 同期型メッセージ送信時

同期型メッセージ送信時の MCF 性能検証用トレースの取得タイミングについて、次の図に示します。

図 G-2 同期型メッセージ送信時の MCF 性能検証用トレースの取得タイミング

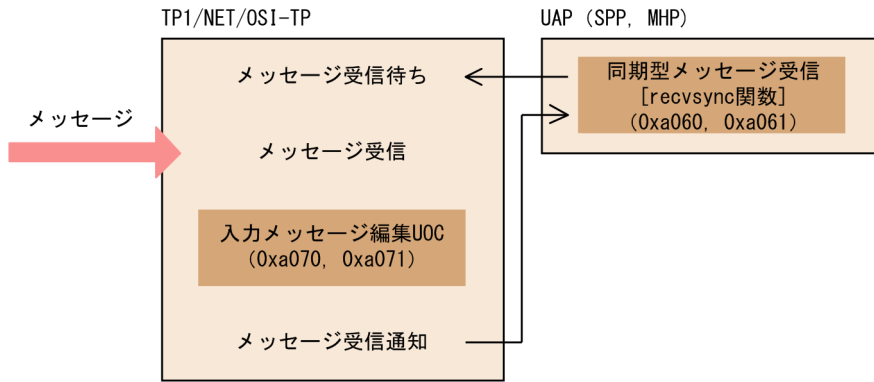


(凡例)  
 : MCF性能検証用トレースの取得タイミング  
 ( ) : イベントID

## (3) 同期型メッセージ受信時

同期型メッセージ受信時の MCF 性能検証用トレースの取得タイミングについて、次の図に示します。

図 G-3 同期型メッセージ受信時の MCF 性能検証用トレースの取得タイミング



(凡例)

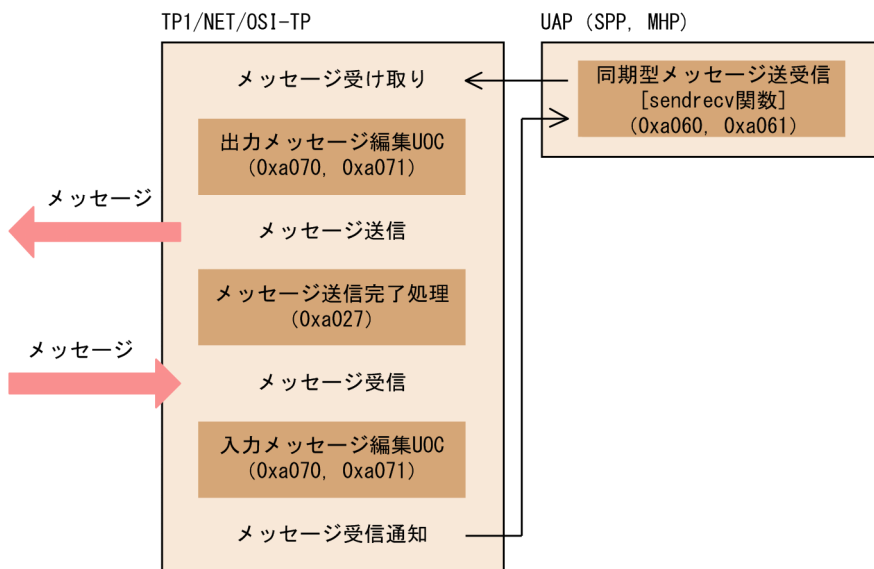
■ : MCF性能検証用トレースの取得タイミング

( ) : イベントID

#### (4) 同期型メッセージ送受信時

同期型メッセージ送受信時の MCF 性能検証用トレースの取得タイミングについて、次の図に示します。

図 G-4 同期型メッセージ送受信時の MCF 性能検証用トレースの取得タイミング



(凡例)

■ : MCF性能検証用トレースの取得タイミング

( ) : イベントID

### 付録 G.3 MCF 性能検証用トレースの取得量

1 回のメッセージ送受信で取得する MCF 性能検証用トレースのトレース取得量を、次の表に示します。

表 G-3 MCF 性能検証用トレースの取得量

メッセージの送受信形態	トレース取得量 (単位：キロバイト)
一方送信メッセージの受信	2.0
同期型メッセージの送信	0.6
同期型メッセージの受信	0.5
同期型のメッセージの送受信	0.9

# 付録 H ユーザアプリケーションプログラムの作成例

メッセージ送受信の処理の流れを次の図に示します。

図 H-1 処理の流れ (全二重の場合)

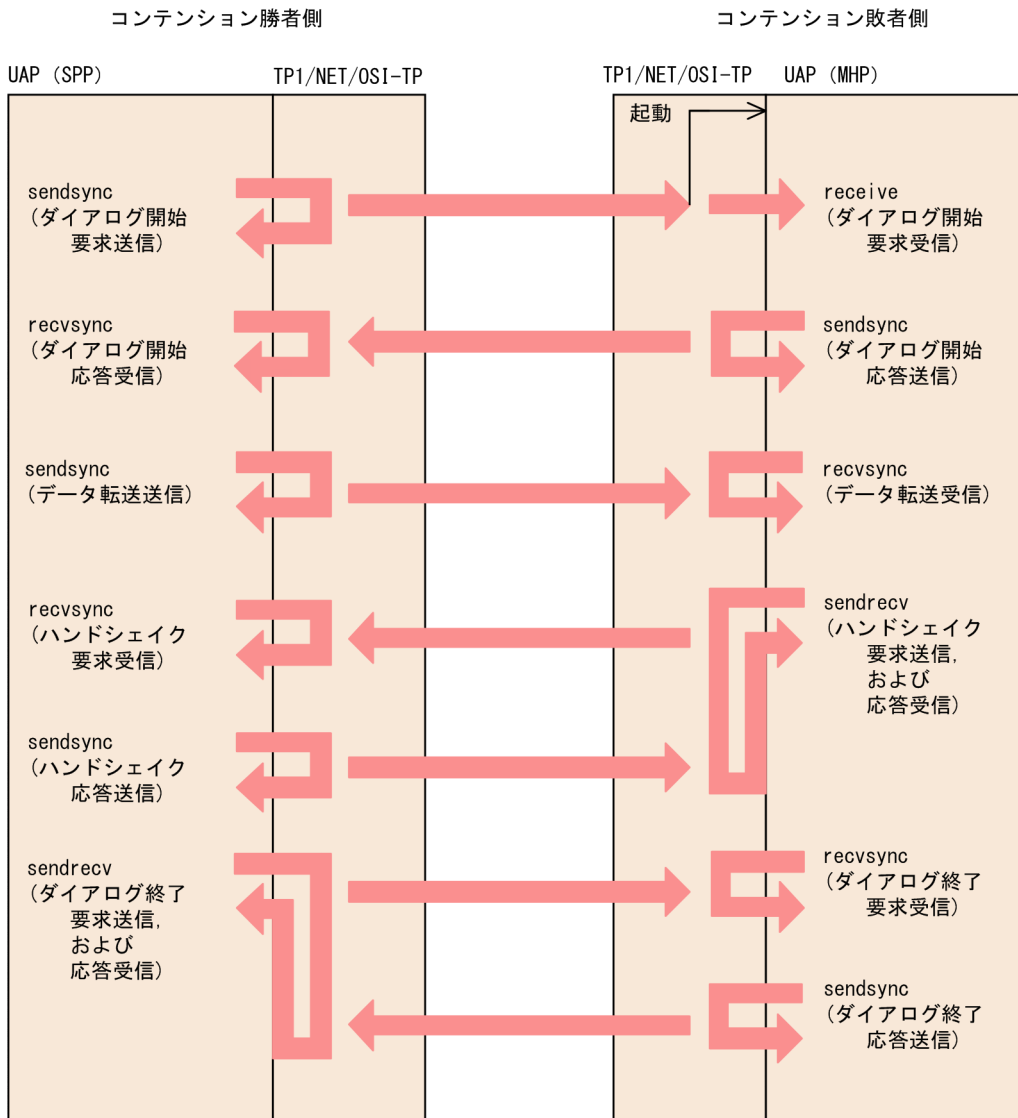
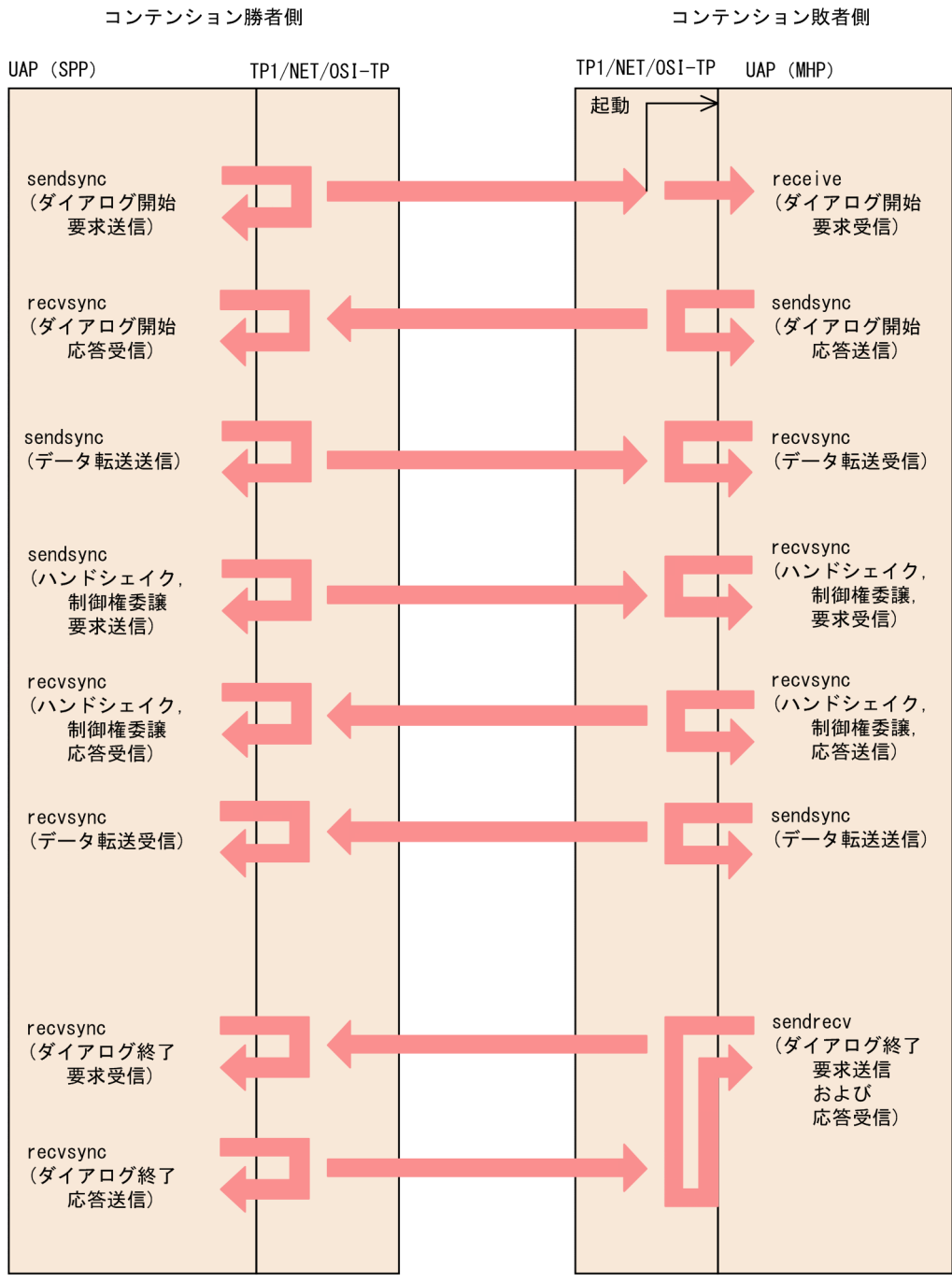


図 H-2 処理の流れ (半二重の場合)



## 付録 H.1 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。



```

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     ; /* パラメタフィールドのトータル */

senddata[20] = 0x00     ; /* レングス 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     ; /* レングス 04 */
senddata[34] = 0x02     ; /* 受信側AP(パラメタ 2省略) */

senddata[35] = 0x00     ; /* レングス 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     ; /* レングス 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     ; /* レングス 07 */
senddata[53] = 0x02     ; /* サービス品質(パラメタ5省略) */

senddata[54] = 0x00     ; /* レングス 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     ; /* レングス 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 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, "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,
                    &dataleng, inbufleng, &time,
                    wachtime);

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

ERROR : ;
}

```

## (b) 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)

        unsigned      char      *in      ;
        unsigned      DCLONG    in_len  ;
        unsigned      char      *out     ;
        unsigned      DCLONG    out_len  ;
{

```









```

/*★ 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 : ;
}

```

## (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)

### (a) 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'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 '0TP1'.
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'.
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 'SENDRCV'.
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.
02 DATA117 PIC X(28) VALUE LOW-VALUE.
01 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.

```

## (b) コンテンション敗者側 (MHP)

```

IDENTIFICATION DIVISION.
PROGRAM-ID. REC01.

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(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.
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.

```

### (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)

#### (a) コンテンション勝者側 (SPP)

```

*
*****
* 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'.
        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=0
        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=0
        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-0T2
  FOR I-0
  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-0
  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-0T3
  FOR I-0
  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-0T4
  FOR I-0
  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-0
  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-0T1
  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-0T2
  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.

```

## (b) コンテンション敗者側 (MHP)

```

*
*****
*   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).
      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'.
        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-0
    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-0
    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-0
        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-0
        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-0
        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-0
        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.

```

## (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)

### (a) コンテンション勝者側 (SPP)

```

#include <dcpcf.h>
#include <string.h>

#define    WAITTIME    60

void    SEND01(in, in_len,  out, out_len)

    unsigned    char        *in        ;
    unsigned    DCLONG      in_len    ;
    unsigned    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      recvddata[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(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, "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 */

/*▼▼ 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 : ;
}

```

## (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)

### (a) 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.
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'.
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.
```

## (b) 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.	
	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.
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 'OTP1'.
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.

```

## (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)

### (a) コンテンション勝者側 (SPP)

```

*
*****
* 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'.
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-0
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-0
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-0
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-0
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-0
    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-0
    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-0
    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-0T4
    FOR I-0
    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-0T1
    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-0T2
    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-0T3
    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.

```

## (b) コンテンション敗者側 (MHP)

```

*
*****
*   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).
        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&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-0
        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-0
        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-0
        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-0
        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-0
        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-0
        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-0
        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 H.2 提供するサンプルコーディング

ここでは、TP1/NET/OSI-TP が提供するサンプルコーディングの格納場所について言語ごとに示します。

## (1) C 言語

### (a) 全二重の場合

- 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
```

### (b) 半二重の場合

- 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
```

## (2) COBOL 言語

### (a) 全二重の場合

- 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
```

### (b) 半二重の場合

- 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
```

## (3) データ操作言語

### (a) 全二重の場合

- 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
```

## (b) 半二重の場合

- 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I 理由コード一覧

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

表 I-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	ユーザサーバ未起動などによって、MHP の起動に失敗しました。
DCMCF_QUE_BUF_ERR (0030)	0030	動的共用メモリが不足しました。
DCMCF_QUE_FIL_OVER (0031)	0031	キューファイルが満杯になりました。
DCMCF_QUE_LIMIT_OVER (0032)	0032	入力メッセージ最大格納数を超過しました。
DCMCF_QUE_IO_ERR (0033)	0033	入力キュー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しています。
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	<ul style="list-style-type: none"> <li>MHP のサービスまたはサービスグループが閉塞しています。</li> <li>スケジュール閉塞されているサービスグループの入力キューに未処理受信メッセージが残った状態で、OpenTP1 を正常終了または計画停止 A で終了しました。</li> </ul>
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	<ul style="list-style-type: none"> <li>MHP で呼び出す receive 関数にセグメントを渡す前に、MHP が異常終了しました。</li> <li>DBMS の障害などによって、トランザクションの開始に失敗しました。</li> </ul>

表 I-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
	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)	運用コマンドまたは API 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(UOC 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
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

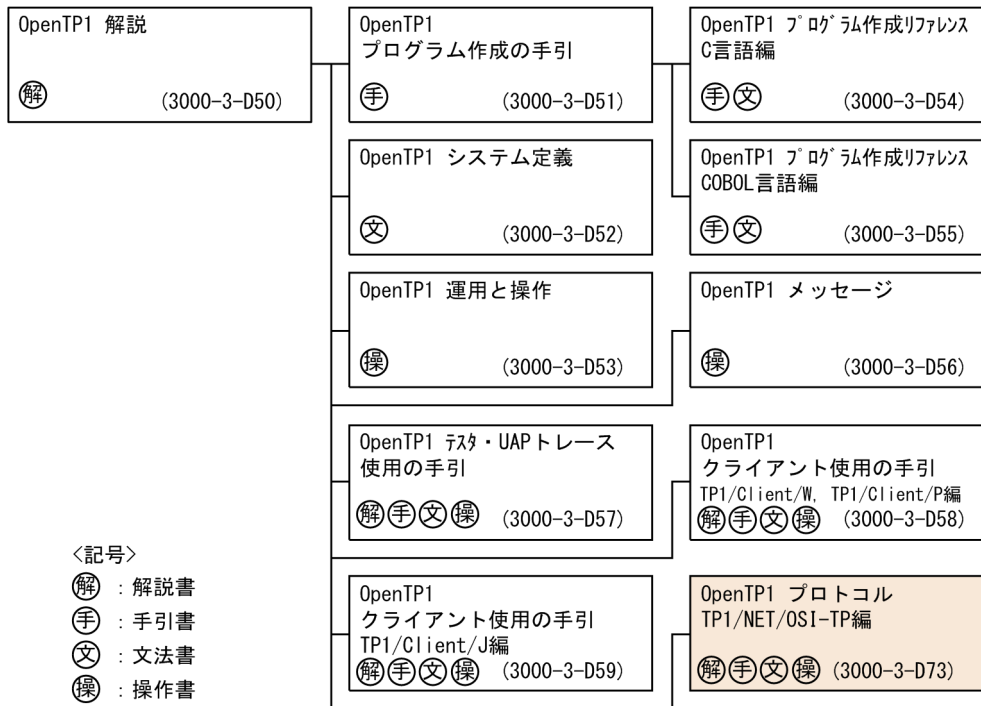
理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件	障害レベル
その他	不定	上記以外の障害 (理由コード 1, 2 は保守情報)	コネクション

## 付録 J.1 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

### (1) OpenTP1 関連

●OpenTP1 Version 7



### (2) 通信管理関連

- 通信管理 XNF/LS 使用の手引 (3000-3-B51)
- 通信管理 XNF/AS 解説・運用編 (3000-3-B61)
- 通信管理 XNF/AS 構成定義編 (3000-3-B62)
- 通信管理 XNF/AS NSAP アドレス概説編 (3000-3-B43)

### (3) 相手システム (VOS3) 関連

- OSI アプリケーション共通機能/OSI トランザクション処理/DCCM3 OSAS/TP/DCCM3 (6190-6-372)
- XNF TCP/IP 接続機能 XNF/TCP 定義とコマンド (6190-3-584)

## 付録 J.2 このマニュアルでの表記

### (1) 製品名

このマニュアルでの表記と正式名称を次の表に示します。

表記		製品名
TP1/Message Control		uCosminexus TP1/Message Control
TP1/NET/Library		uCosminexus TP1/NET/Library
TP1/NET/OSI-TP		uCosminexus TP1/NET/OSI-TP
UNIX	AIX	AIX V6.1
		AIX V7.1

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

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

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

## 付録 J.3 英略語

このマニュアルで使用する英略語を次に示します。

英略語	英字での表記
ACSE	Association Control Service Element
AP	Application Program
LAN	Local Area Network
MCF	Message Control Facility
MHP	Message Handling Program
OS	Operating System
OSI TP	Open Systems Interconnection Transaction Processing

英略語	英字での表記
PSAP	Presentation Service Access Point
RPC	Remote Procedure Call
SPP	Service Providing Program
UAP	User Application Program
UOC	User Own Cording

## 付録 J.4 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,  $1,024^2$  バイト,  $1,024^3$  バイト,  $1,024^4$  バイトです。

### (英字)

#### GJ (メッセージ受信ジャーナル)

MCF が取得する統計用の履歴情報の一つです。入力メッセージサイズ、入力論理端末名などで構成されます。

GJ の取得タイミングは、非同期受信関数を発行して、MHP が受信メッセージを入力キューから取り出した直後です。

#### IJ (メッセージ入力ジャーナル)

MCF が取得する統計用の履歴情報の一つです。メッセージ入力通番、入力論理端末名、入力メッセージ種別、および入力メッセージなどで構成されます。

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格納する直前です。

#### MJ (メッセージジャーナル)

MCF が取得する統計用の履歴情報の一つです。入力論理端末名または出力論理端末名、メッセージジャーナル種別、入力または出力メッセージ（入力メッセージ編集前のデータ、または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、入力メッセージの場合、入力メッセージ編集 UOC を呼び出す直前です。また、出力メッセージの場合、出力メッセージ編集 UOC を呼び出した直後です。

#### 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。

### (力行)

#### 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。

## (タ行)

### 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。

## (ハ行)

### 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

## 数字

07-00 での変更点 264

07-50 での変更点 263

## A

APDU 連結 70

AP 間通信 18

AP 間通信の概要 18

AP 間通信の形態 20

AP 間通信の仕組み 27

AP 間通信メッセージの送受信 53

## B

bretry 203

bretrycnt 204

bretryint 204

## C

CBLDCMCF('RECEIVE ') 112

CBLDCMCF('RECVSYNC') 117

CBLDCMCF('SENDRECV') 122

CBLDCMCF('SENDSYNC') 129

CBLDCMCF('TACTCN ') 135

CBLDCMCF('TDCTCN ') 138

CBLDCMCF('TLSCN ') 142

CCLSEVT 184, 191

CERREVT 183, 190

CERREVT の理由コード一覧 346

cnflevt 206

COBOL-UAP 作成用プログラムインタフェース 108

COBOL-UAP 作成用プログラムインタフェースの  
一覧 109

control 205

COPNEVT 184, 191

count 202

C 言語のライブラリ関数 71

C 言語のライブラリ関数の一覧 72

## D

dc\_mcf\_receive 73

dc\_mcf\_recvsync 77

dc\_mcf\_sendrecv 82

dc\_mcf\_sendsync 89

dc\_mcf\_tactcn 95

dc\_mcf\_tdctcn 99

dc\_mcf\_tlscn 103

DISABLE 146

## E

ENABLE 149

ERREVT1 181, 185

ERREVT2 181, 186

ERREVT2 の理由コード一覧 345

ERREVT3 182, 187

ERREVTA 183, 189

## G

GJ (メッセージ受信ジャーナル) 351

## H

handshake 205

## I

IJ (メッセージ入力ジャーナル) 351

## M

max\_open\_fds 214

max\_socket\_descriptors 213

MCF 25

mcfmuap 198

mcfostp 216

mcfostpr 217

mctactcn 233

mcftalccn 200

mcftalced 207

mcftalcle 208  
mcftbuf 202, 206  
mcftdctcn 236  
mcftgrpcn 209  
mcftgrped 210  
mcftlscn 239  
MCF アプリケーション定義 193  
MCF イベント 178  
MCF イベントインタフェース 178, 272  
MCF イベント処理用 MHP 178, 180  
MCF イベント通知時のセグメント構成 179  
MCF イベントの共通ヘッダ 180  
MCF イベントの種類 178  
MCF 固有情報の出力情報 283  
MCF サービス名の登録 244  
MCF 性能検証用トレースの取得 283  
MCF 性能検証用トレースの取得タイミング 283  
MCF 性能検証用トレースの取得量 285  
MCF 通信構成定義 193  
MCF 通信プロセスでアクセスするファイルの最大数  
214  
MCF 定義オブジェクトの解析 217  
MCF 定義オブジェクトの生成 216  
MCF で使用する定義ファイル 193  
MCF マネージャ定義 193  
MCF メイン関数のコーディング概要 273  
MCF メイン関数の作成 245  
MJ (メッセージジャーナル) 351  
module 211  
msgbuf 202

## N

nomltim 203

## O

OpenTP1 システムの変更に影響する定義 225  
OSAS/TP/DCCM3 25  
OSAS/TP/DCCM3 の定義との関係 224  
OSI TP プロトコル 351

## P

PSAP 35

## R

rcvbuf 202  
RECEIVE 153  
recvtim 199

## S

SEND 158  
slot 204  
sndbuf 202  
sndrcvtim 198  
sndtim 198

## T

TL クラス 204  
TP1/Message Control 25  
TP1/NET/OSI-TP の OSI7 層構造の中での機能範囲  
19  
TP1/NET/OSI-TP の TP プロトコルデータ単位 27  
TP1/NET/OSI-TP の運用コマンド 232  
TP1/NET/OSI-TP のシステム構成例 228  
TP1/NET/OSI-TP の実行形式プログラム名 211  
TP1/NET/OSI-TP の実装範囲 275  
TP1/NET/OSI-TP を組み込んだソフトウェア構成  
の例 25  
TP1/NET/OSI-TP を使用したネットワーク構成の例  
18

## U

UAP 異常終了通知イベント 179  
UAP 共通定義 198  
UAP でのエラー発生時の処理 (制御権がある場合) 68  
UAP でのエラー発生時の処理 (制御権がない場合) 69  
UAP でのエラー発生時の処理 (全二重の場合) 67  
UAP 閉塞による障害 261  
UOC 作成上の注意事項 176  
UOC で使用できる関数 177  
UOC の異常処理 177

UOC の構造 176  
UOC の実行タイミング 177  
UOC 名称の出力情報 283  
UOC 呼び出し時の MCF 固有情報のダンプ出力情報  
283

## 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) 69  
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) 68  
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) 67  
相手システムからの接続の解放 32  
相手システムの異常による解放 34  
相手システムの通信定義と関連づける内容 222  
アソシエーションプール管理機能 40  
アソシエーションプール管理機能の概要 40  
アプリケーション名 208  
アプリケーション名の決定 61, 165

## い

一方受信 21  
一方送信メッセージの受信 (COBOL 言語) 112  
一方送信メッセージの受信 (C 言語) 73  
インタフェースの変更一覧 (バージョン 6 以前から移行する場合) 266

## う

運用コマンド 231

## え

エラー通知 66

## お

応答を不要とした場合のメッセージ破棄の例 61  
応用コンテキスト名 205

## か

概要 17

確認型 59  
仮想スロット番号 204  
関連マニュアル 348

## き

機能 26  
基本サービスプリミティブ 41  
旧製品からの移行に関する注意事項 265  
拒否応答型 59

## <

組み込み方法 243

## こ

コーディング例 288  
接続 27, 351  
接続 ID 200  
接続確立再試行回数 204  
接続確立再試行間隔 204  
接続グループ定義の開始 209  
接続グループ定義の終了 210  
接続グループ名 209  
接続障害 258  
接続定義の開始 200  
接続定義の終了 207  
接続と PSAP の関係 35  
接続と論理端末の関係 35  
接続の解放 236  
接続の解放 (COBOL 言語) 138  
接続の解放 (C 言語) 99  
接続の確立 27, 233  
接続の確立 (COBOL 言語) 135  
接続の確立 (C 言語) 95  
接続の確立 (着呼型) 30  
接続の確立 (発呼型 (運用コマンド入力時またはオンライン開始時)) 28  
接続の確立再試行 203  
接続の確立と解放 27  
接続の確立方法 203

コネクションの起動, または受信の種別を指定 204  
コネクションの強制解放 32  
コネクションの状態取得 (COBOL 言語) 142  
コネクションの状態取得 (C 言語) 103  
コネクションの状態表示 239  
コネクションの正常解放 30  
コネクションの切断 34  
コンテンション 59  
コンテンション (競合) 351  
コンテンションの勝者または敗者の種別 205

## さ

サービス詳細 276  
サービスプリミティブ 41, 275, 352  
サービスプリミティブと UAP との対応 42  
サービスプリミティブとパラメタとの対応 47  
サービスプリミティブの形式 45  
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) 51  
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) 52  
サービスプリミティブの連結パターンと UAP との対応 44

## し

自システムからのコネクションの解放 (オンライン終了時または運用コマンド入力時) 31  
自システムからのコネクションの強制解放 (異常検知時または運用コマンド入力時) 33  
自システムの PSAP アドレス 201  
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 219  
システムサービス共通情報定義 213  
システムサービス情報定義 211  
システム定義 192  
実装範囲 275  
出力メッセージの編集 172  
出力メッセージの編集 UOC 172  
出力メッセージ編集 UOC インタフェース 173  
障害対策 251

障害通知イベント 179  
障害の種類と対応処理 252  
障害発生時の処理の流れ 282  
状態通知イベント 179  
処理層とデータの関係 (単一 APDU の場合) 39  
処理層とデータの関係 (連結 APDU の場合) 39

## せ

制御権 57, 352  
制御権委譲付きハンドシェイク 65  
制御権が移動する契機 58  
制御権の委譲 57  
制御権の要求 57  
制御情報の形式 46  
全二重 352  
全二重機能 205  
全二重と半二重 38

## そ

送受信メッセージの形式 45  
ソケット用ファイル記述子の最大数 213  
ソフトウェア構成の例 25

## た

ダイアログ 58, 352  
ダイアログ開始方向 61  
ダイアログ開始要求受信時の MHP 異常終了 260  
ダイアログ障害 259  
ダイアログ中の UAP (MHP, SPP) 終了 260  
ダイアログと UAP 61  
ダイアログの異常終了 60  
ダイアログの開始 59  
ダイアログの開始と終了 60  
ダイアログの終了 59  
タイマ起動メッセージ廃棄通知イベント 179  
端末タイプ 208

## ち

着呼型 29

## つ

- 通信相手システムの PSAP アドレス 201
- 通信相手システムの種別 203
- 通信形態 20

## て

- 定義オブジェクトファイルの生成 248
- 定義の指定順序 197
- 提供するサンプルコーディング 342
- 定義例 227
- データ転送 65
- 適用範囲 21

## と

- 同期型 54
- 同期型のメッセージの受信 55
- 同期型のメッセージの送受信 54, 55
- 同期型のメッセージの送信 54
- 同期型のメッセージの送信, 受信, および送受信 56
- 同期型メッセージの受信 (COBOL 言語) 117
- 同期型メッセージの受信 (C 言語) 77
- 同期型メッセージの送受信 (COBOL 言語) 122
- 同期型メッセージの送受信 (C 言語) 82
- 同期型メッセージの送信 (COBOL 言語) 129
- 同期型メッセージの送信 (C 言語) 89
- 同期型メッセージの送信 (データ操作言語) 146, 149
- 同期受信 21
- 同期受信監視時間 199
- 同期送受信 21
- 同期送受信監視時間 198
- 同期送信 21
- 同期送信監視時間 198

## に

- 入力メッセージの編集 165
- 入力メッセージの編集とアプリケーション名の決定 165
- 入力メッセージ編集 UOC 165
- 入力メッセージ編集 UOC インタフェース 166

## ね

- ネットワーク構成の例 222

## は

- バージョンアップ時の変更点 263
- 発呼型 27
- バッファ形式 1 39
- バッファ形式 2 39
- パラメタの形式 46
- パラメタの詳細 48
- ハンドシェイク 63, 352
- ハンドシェイクを使用 205
- 半二重 352
- 半二重機能 205

## ひ

- 非同期型 53
- 非同期型のメッセージの受信 53

## ふ

- 不正アプリケーション名検出通知イベント 178
- プロトコル監視時間 203
- プロトコルの種別 201

## み

- 未処理送信メッセージ廃棄通知イベント 179

## め

- メッセージ受信用バッファグループ番号 202
- メッセージ制御機能 25
- メッセージ送受信インタフェース 266
- メッセージ送受信時のコネクション障害 258
- メッセージ送受信の処理の流れ 280
- メッセージ送信用バッファグループ番号 202
- メッセージの応答の要否 23, 61
- メッセージの応答を必要とした場合 24
- メッセージの応答を不要とした場合 23
- メッセージの形式 44
- メッセージの形式の例 23

- メッセージの受信 54
- メッセージの受信 (データ操作言語) 153
- メッセージの送信 (データ操作言語) 158
- メッセージの送達確認の例 22
- メッセージの送達の確認 22
- メッセージの重複, および欠落のチェック 22
- メッセージの分割と組み立て 38
- メッセージ廃棄通知イベント 178
- メッセージ編集用バッファグループ番号 202
- メッセージ編集用バッファ数 202

## ゆ

- ユーザ ASE 抽象構文名 205
- ユーザアプリケーションプログラム異常終了 260
- 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) 289
- 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) 307
- 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) 326
- 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) 335
- ユーザアプリケーションプログラムの作成例 287
- ユーザアプリケーションプログラム閉塞 261
- ユーザオウンコーディング 165, 269
- ユーザオウンコーディングインタフェース 165
- ユーザ情報の形式 46
- ユーザ情報の操作方法 177

## り

- 理由コード一覧 345

## ろ

- 論理端末 35, 352
- 論理端末定義 208
- 論理端末と UAP の関係 37
- 論理端末とアプリケーションの型の関係 37
- 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 37