

---

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/User Agent 編

解説・手引・文法・操作書

3000-3-D71

マニュアルの購入方法

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

HITACHI

## 対象製品

印の付いているプログラムプロダクトについては、発行時期をご確認ください。

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

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

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

P-F1M64-32311 uCosminexus TP1/NET/User Agent 07-00

・適用 OS : HP-UX 11i

R-18455-41K uCosminexus TP1/Message Control 07-00

R-18456-41K uCosminexus TP1/NET/Library 07-00

R-F18456-411K uCosminexus TP1/NET/User Agent 07-00

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

これらの製品は、ISO9001 および TickIT の認証を受けた品質システムで開発されました。

## 輸出時の注意

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

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

## 商標類

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

HP-UX は、米国 Hewlett-Packard Company のオペレーティングシステムの名称です。

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

## 発行

2006 年 11 月 (第 1 版) 3000-3-D71

## 著作権

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

# はじめに

---

このマニュアルは、OpenTP1 システムの中で、TP1/NET/User Agent を使用した AP 間通信システムの作成方法、および運用方法について説明したものです。  
本文中に記載されている製品のうち、このマニュアルの対象製品ではない製品については、OpenTP1 Version 7 対応製品の発行時期をご確認ください。

## 対象読者

OpenTP1 システムの通信に User Agent プロトコルを使用するシステム管理者、システム設計者、およびプログラマを対象としています。また、オンラインや OpenTP1 システムの基礎的な知識を持っていて、次のマニュアルを理解されていることを前提としています。

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

## マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

### 第 1 章 概要

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

### 第 2 章 機能

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

### 第 3 章 メッセージ送受信インタフェース

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

### 第 4 章 ユーザOWNコーディング、MCF イベントインタフェース

TP1/NET/User Agent 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 5 章 システム定義

OSAS/UA 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/User Agent 固有のシステム定義について説明しています。また、自システム内にある通信管理プログラムと関連づける内容、相手システムの通信定義と関連づける内容、およびシステム定義例について説明しています。

はじめに

## 第 6 章 運用コマンド

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

## 第 7 章 組み込み方法

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

## 第 8 章 障害対策

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

## 付録 A メッセージ送受信の処理の流れ

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

## 付録 B 障害発生時の処理の流れ

障害が発生した場合の処理の流れについて説明しています。

## 付録 C 理由コード一覧

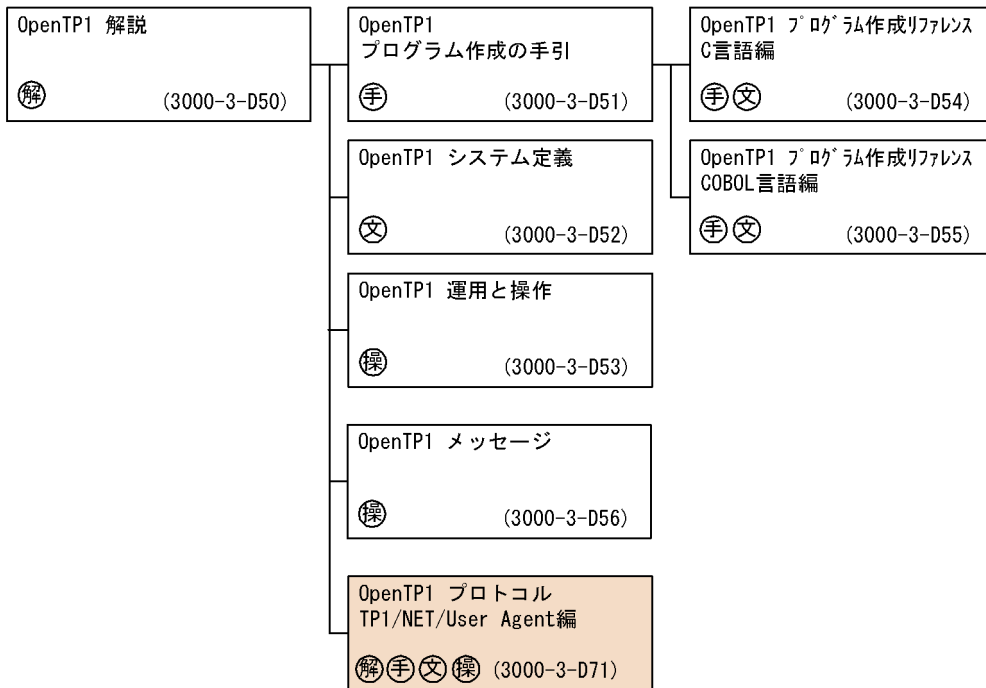
障害通知イベントが発生した場合の理由コードについて説明しています。

## 付録 D 用語解説

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

## 関連マニュアル

## ●OpenTP1 Version 7



&lt;記号&gt;

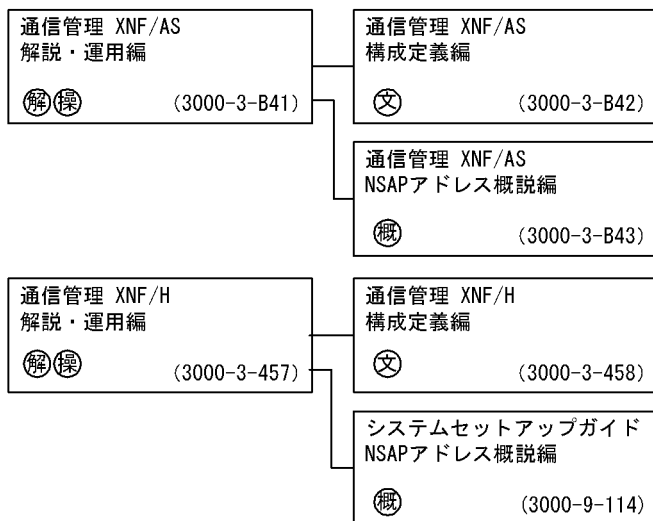
解 : 解説書

手 : 手引書

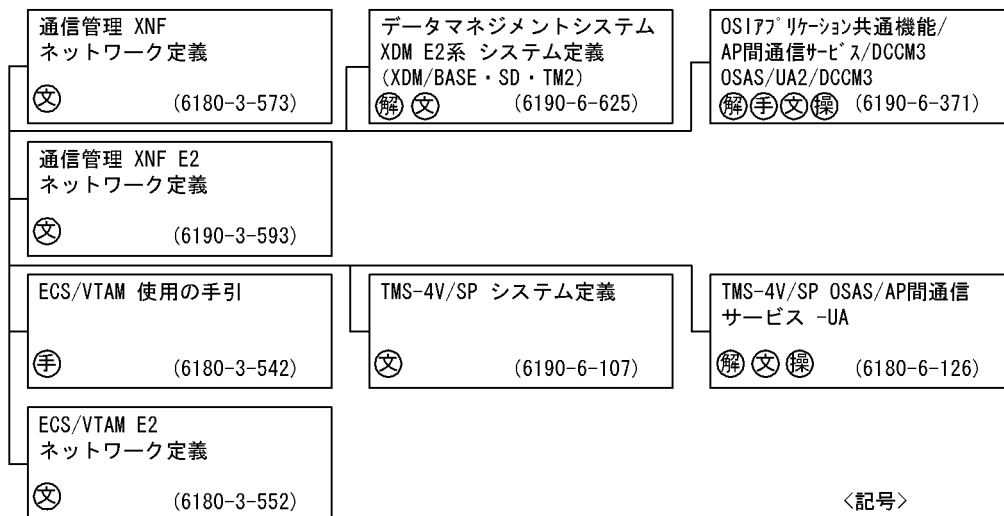
文 : 文法書

操 : 操作書

●通信管理



●通信相手システム (VOS3)

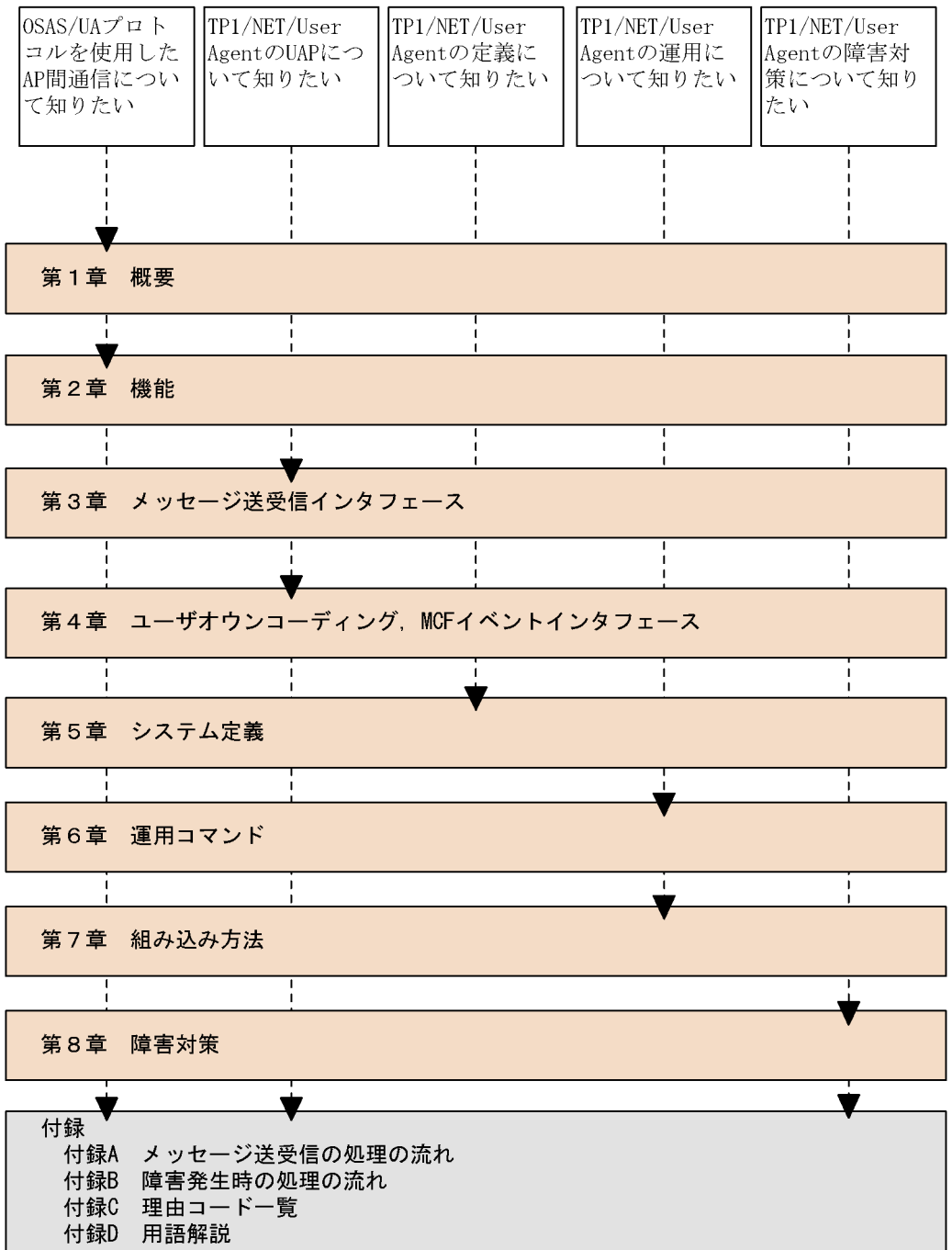


<記号>


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

読書手順

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



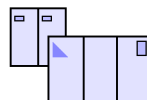
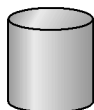
(凡例)

 : 必ず読む項目

## 図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。

- ワークステーション, 端末
- 入出力の動作
- ファイル
- ホストコンピュータ



- 論理端末



- データの流れ



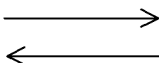
- 通信回線



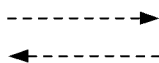
- プログラムの流れ



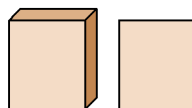
- 制御の流れ



- その他の流れ



- プログラム



- 論理回線



- 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)	この記号で区切られた項目は選択できることを示します。 (例) -t reply   request -t オプションに reply または request を指定できることを示します。 ただし、C 言語のインタフェースの説明でこの記号を使用した場合は、C 言語の文法規則に従います。
{ }	この記号で囲まれている複数の項目のうちから一つを選択できることを示します。 (例) { DCMCFESI   DCMCFEMI } DCMCFESI と DCMCFEMI のうち、どちらかを指定できることを示します。



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

## (2) 属性表示記号

ユーザ指定値の範囲などを説明する記号です。

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

## (3) 構文要素記号

ユーザ指定値の内容を説明する記号です。

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

## このマニュアルでの表記

### (1) 製品名

このマニュアルで使用する製品名称の略称を次に示します。

製品名称	略称
AIX 5L V5.1	AIX
AIX 5L V5.2	
AIX 5L V5.3	
HP-UX 11i	HP-UX
uCosminexus TP1/Message Control	TP1/Message Control
uCosminexus TP1/NET/Library	TP1/NET/Library
uCosminexus TP1/NET/User Agent	TP1/NET/User Agent

- AIX および HP-UX を総称して UNIX と表記しています。

### (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>	' ' (オーバーライン)	'~' (チルド)

## 略語一覧

このマニュアルで使用する英略語の一覧を次に示します。

英略語	英字での表記
ACSE	<u>A</u> ssociation <u>C</u> ontrol <u>S</u> ervice <u>E</u> lement
AP	<u>A</u> pplication <u>P</u> rogram
CCP	<u>C</u> ommunication <u>C</u> ontrol <u>P</u> rocessor
LE	<u>L</u> ogical <u>E</u> ntity
MCF	<u>M</u> essage <u>C</u> ontrol <u>F</u> acility
MHP	<u>M</u> essage <u>H</u> andling <u>P</u> rogram
NIF/OSI	<u>N</u> etwork <u>I</u> nterface <u>F</u> eature/ <u>O</u> pen <u>S</u> ystems <u>I</u> nterconnection
OS	<u>O</u> perating <u>S</u> ystem
OSAS/UA	<u>O</u> pen <u>S</u> ystems <u>I</u> nterconnection <u>A</u> pplication <u>S</u> upport <u>C</u> ommon <u>F</u> acility/ <u>U</u> ser <u>A</u> gent
OSI	<u>O</u> pen <u>S</u> ystems <u>I</u> nterconnection

英略語	英字での表記
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram
UA	<u>U</u> ser <u>A</u> gent
UAM	<u>U</u> A <u>M</u> anager
UAP	<u>U</u> ser <u>A</u> pplication <u>P</u> rogram
UASDU	<u>U</u> ser <u>A</u> gent <u>S</u> ervice <u>D</u> ata <u>U</u> nit
UOC	<u>U</u> ser <u>O</u> wn <u>C</u> oding
UPDU	<u>U</u> ser Agent <u>P</u> rotocol <u>D</u> ata <u>U</u> nit

### 常用漢字以外の漢字の使用について

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

個所（かしょ） 閉塞（へいそく）

### KB（キロバイト）などの単位表記について

1KB（キロバイト）、1MB（メガバイト）、1GB（ギガバイト）、1TB（テラバイト）はそれぞれ 1,024 バイト、1,024<sup>2</sup> バイト、1,024<sup>3</sup> バイト、1,024<sup>4</sup> バイトです。

### 謝辞

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

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

COBOL は産業界の言語であり、特定の団体や組織の所有物ではない。

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

次に示す著作権表示付資料の著作者及び著作権者

はじめに

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

<b>1</b>	<b>概要</b>	<b>1</b>
1.1	AP 間通信の概要	2
1.2	AP 間通信の形態	4
1.3	ソフトウェアの構成	6
1.3.1	前提プログラム	6
1.3.2	ソフトウェア構成の例	6
<b>2</b>	<b>機能</b>	<b>9</b>
2.1	AP 間通信の仕組み	10
2.1.1	コネクションの確立と解放	10
2.1.2	コネクションと論理端末の関係	13
2.1.3	経路交代	14
2.1.4	論理端末とアプリケーションの型の関係	14
2.1.5	メッセージの分割と組み立て	16
2.1.6	UA の開局と閉局	17
2.2	AP 間通信メッセージの送受信	21
2.2.1	問い合わせメッセージの送信	21
2.2.2	応答メッセージの送信	22
2.2.3	一方送信メッセージの送信と受信	23
<b>3</b>	<b>メッセージ送受信インタフェース</b>	<b>25</b>
	メッセージ送受信インタフェースの一覧	27
	dc_mcf_receive - メッセージの受信 (C 言語)	29
	dc_mcf_recvsync - 同期型のメッセージの受信 (C 言語)	33
	dc_mcf_reply - 応答メッセージの送信 (C 言語)	37
	dc_mcf_resend - メッセージの再送 (C 言語)	40
	dc_mcf_send - 一方送信メッセージの送信 (C 言語)	44
	dc_mcf_sendrecv - 同期型のメッセージの送受信 (C 言語)	48
	CBLDCMCF('RECEIVE ') - メッセージの受信 (COBOL 言語)	53
	CBLDCMCF('RECVSYNC') - 同期型のメッセージの受信 (COBOL 言語)	58
	CBLDCMCF('REPLY ') - 応答メッセージの送信 (COBOL 言語)	63
	CBLDCMCF('RESEND ') - メッセージの再送 (COBOL 言語)	68

CBLDCMCF('SEND ') - 一方送信メッセージの送信 (COBOL 言語)	74
CBLDCMCF('SENDRECV') - 同期型のメッセージの送受信 (COBOL 言語)	79
RECEIVE - メッセージの受信 (データ操作言語)	85
SEND - メッセージの送信 (データ操作言語)	88
ユーザアプリケーションプログラム作成例	94

## 4

ユーザオウンコーディング, MCF イベントインタフェース	99
4.1 ユーザオウンコーディングインタフェース	100
4.1.1 入力メッセージの編集とアプリケーション名の決定	100
4.1.2 入力メッセージ編集 UOC インタフェース	102
4.1.3 出力メッセージの編集	107
4.1.4 出力メッセージ編集 UOC インタフェース	108
4.1.5 送信メッセージの通番編集	111
4.1.6 送信メッセージの通番編集 UOC インタフェース	112
4.1.7 UOC 作成上の注意事項	114
4.2 MCF イベントインタフェース	115
4.2.1 MCF イベントの種類	115
4.2.2 MCF イベント通知時のセグメント構成	116
4.2.3 MCF イベント情報の形式 (C 言語)	117
4.2.4 MCF イベント情報の形式 (COBOL 言語)	122

## 5

システム定義	129
TP1/NET/User Agent の定義の概要	130
TP1/NET/User Agent 固有のシステム定義の種類	132
mcfmup (UAP 共通定義)	136
mcftalccn (コネクション定義の開始)	137
mcftalced (コネクション定義の終了)	146
mcftalcle (論理端末定義)	147
mcftalcua (UA 定義)	149
mcftcomn (MCF 通信構成共通定義)	150
システムサービス情報定義	151
システムサービス共通情報定義	152
MCF 定義オブジェクトの生成	155
自システムの通信管理プログラムと関連づける内容	156
相手システムの通信定義と関連づける内容	157

定義例	162
-----	-----

<b>6</b>	<b>運用コマンド</b>	<b>171</b>
	TP1/NET/User Agent の運用コマンド	172
	mcftactcn (コネクションの確立)	173
	mcftactle (論理端末の閉塞解除)	175
	mcftdctcn (コネクションの解放)	177
	mcftdctle (論理端末の閉塞)	179
	mcftlscn (コネクションの状態表示)	181
	mcftlsle (論理端末の状態表示)	184

<b>7</b>	<b>組み込み方法</b>	<b>187</b>
	7.1 TP1/NET/User Agent の組み込みの流れ	188
	7.2 MCF メイン関数の作成	189
	7.3 定義オブジェクトファイルの生成	192

<b>8</b>	<b>障害対策</b>	<b>195</b>
	8.1 障害の種類と対応処理	196
	8.2 コネクション障害	204
	8.3 問い合わせメッセージ, 応答メッセージ障害	206
	8.3.1 問い合わせメッセージ送信時の障害	206
	8.3.2 応答メッセージ送信時の障害	207
	8.4 一方送信メッセージ障害	210
	8.5 論理端末の端末タイプ不正	212

<b>付録</b>	<b>213</b>
付録 A メッセージ送受信の処理の流れ	214
付録 A.1 一方送信メッセージ	214
付録 A.2 問い合わせメッセージ, および応答メッセージ	220
付録 B 障害発生時の処理の流れ	225
付録 B.1 一方送信メッセージ障害	225
付録 B.2 応答メッセージ障害	227
付録 B.3 相手システム (XDM/DCCM3) 接続時の障害	230

付録 C 理由コード一覧	233
付録 D 用語解説	236

---

<b>索引</b>	239
-----------	-----

---



## 目次

図 1-1	OpenTP1 システムを使用したネットワーク構成例	2
図 1-2	TP1/NET/User Agent の OSI7 層構造の中での機能範囲	3
図 1-3	TP1/NET/User Agent を使用した AP 間通信の例	4
図 1-4	TP1/NET/User Agent を組み込んだソフトウェア構成	7
図 2-1	TP1/NET/User Agent からのコネクションの確立	11
図 2-2	相手システムからのコネクションの解放	13
図 2-3	自システムからのコネクションの解放	13
図 2-4	コネクションと論理端末の関係	14
図 2-5	メッセージのデータ形式	17
図 2-6	運用コマンドによる UA の個別閉局	18
図 2-7	運用コマンドによる UA の閉局	19
図 2-8	相手システムからの UA の閉局	20
図 2-9	MCF 検出障害時の UA の閉局	20
図 2-10	問い合わせメッセージの送信と応答メッセージの受信	22
図 2-11	問い合わせメッセージの受信と応答メッセージの送信	23
図 2-12	一方送信メッセージの送信と受信	24
図 3-1	処理の流れ	94
図 4-1	アプリケーション名の決定の処理	101
図 4-2	UOC インタフェース用のパラメタとバッファの関係	106
図 4-3	MCF イベント通知時のセグメント構成	117
図 5-1	TP1/NET/User Agent のプロトコル固有定義コマンドの指定順序	134
図 5-2	プロトコル固有定義の例	135
図 5-3	ネットワーク構成の例 (TMS-4V/SP のシステムと接続する場合)	157
図 5-4	ECS/VTAM ネットワーク定義との関係	158
図 5-5	XNF ネットワーク定義との関係	158
図 5-6	TMS-4V/SP システム定義との関係	159
図 5-7	ネットワーク構成の例 (XDM/DCCM3 のシステムと接続する場合)	160
図 5-8	XDM/DCCM3 システム定義との関係	161
図 5-9	TP1/NET/User Agent のシステム構成例 (新 OSAS/UA プロトコル使用時)	163
図 5-10	TP1/NET/User Agent のシステム構成例 (旧 OSAS/UA プロトコル使用時)	167
図 7-1	MCF メイン関数のコーディング概要 (ANSI C, C++ の場合)	189
図 7-2	MCF メイン関数のコーディング概要 (K&R 版 C の場合)	190
図 7-3	MCF メイン関数のディレクトリへの組み込み方法	191

図 7-4	定義オブジェクトファイルの作成方法の概要	193
図 8-1	コネクション障害の発生個所	204
図 8-2	問い合わせメッセージ送信時，応答メッセージ受信時の障害の発生個所	206
図 8-3	問い合わせメッセージ受信時，応答メッセージ送信時の障害の発生個所	208
図 8-4	一方送信メッセージ障害の発生個所	210
図 8-5	論理端末の端末タイプ不正	212
図 A-1	request 型論理端末からの一方送信メッセージの処理の流れ（UA サービスデータ単位分割なしの場合）	214
図 A-2	request 型論理端末からの一方送信メッセージの処理の流れ（UA サービスデータ単位分割ありの場合）	215
図 A-3	send 型論理端末からの一方送信メッセージの処理の流れ（UA サービスデータ単位分割なしの場合）	216
図 A-4	send 型論理端末からの一方送信メッセージの処理の流れ（UA サービスデータ単位分割ありの場合）	217
図 A-5	receive 型論理端末への一方送信メッセージの処理の流れ（UA サービスデータ単位分割なしの場合）	218
図 A-6	receive 型論理端末への一方送信メッセージの処理の流れ（UA サービスデータ単位分割ありの場合）	219
図 A-7	問い合わせメッセージ送信時の処理の流れ（UA サービスデータ単位分割なしの場合）	220
図 A-8	問い合わせメッセージ送信時の処理の流れ（UA サービスデータ単位分割ありの場合）	221
図 A-9	応答メッセージ送信時の処理の流れ（UA サービスデータ単位分割なしの場合）	222
図 A-10	応答メッセージ送信時の処理の流れ（UA サービスデータ単位分割ありの場合）	224
図 B-1	UINQ 送信時の UA 障害の処理の流れ（一方送信メッセージ障害の場合）	225
図 B-2	入力メッセージ編集 UOC エラーの処理の流れ（一方送信メッセージ障害の場合）	226
図 B-3	UINQ 受信時 UA 障害の処理の流れ（応答メッセージ障害の場合）	227
図 B-4	入力メッセージ編集 UOC エラーの処理の流れ（応答メッセージ障害の場合）	228
図 B-5	UAP 異常終了の処理の流れ（応答メッセージ障害の場合）	229
図 B-6	受信打ち切り受信時の処理の流れ（相手システム接続時の障害の場合）	230
図 B-7	受信拒否・受信拒否解除を受信時の処理の流れ（相手システム接続時の障害の場合）	231
図 B-8	送信中断受信時の処理の流れ（相手システム接続時の障害の場合）	232

## 表目次

表 1-1	TP1/NET/User Agent が適用する AP 間通信のプロトコル	7
表 2-1	TP1/NET/User Agent の UA プロトコルデータ単位	10
表 2-2	UA の数による TP1/NET/User Agent のコネクション確立	12
表 2-3	論理端末，メッセージ，アプリケーションの型，UAP インタフェース，および通信形態の関係	15
表 3-1	メッセージ送受信のライブラリ関数（C 言語）	27
表 3-2	メッセージ送受信の文（COBOL 言語）	27
表 3-3	メッセージ送受信の通信文（データ操作言語）	28
表 4-1	TP1/NET/User Agent が通知する MCF イベントの種類	115
表 4-2	COBOL 言語の MCF イベント情報の内容（ERREVT1）	122
表 4-3	COBOL 言語の MCF イベント情報の内容（ERREVT2）	123
表 4-4	COBOL 言語の MCF イベント情報の内容（ERREVT3）	124
表 4-5	COBOL 言語の MCF イベント情報の内容（ERREVT4）	126
表 4-6	COBOL 言語の MCF イベント情報の内容（GERREVT）	127
表 4-7	COBOL 言語の MCF イベント情報の内容（COPNEVT，CCLSEVT）	128
表 5-1	MCF で使用する定義ファイル	130
表 5-2	TP1/NET/User Agent 固有の定義の一覧	132
表 6-1	TP1/NET/User Agent の運用コマンド	172
表 8-1	コネクション障害発生時の処理	196
表 8-2	UA 障害発生時の処理	197
表 8-3	入力メッセージ編集 UOC の障害発生時の処理	198
表 8-4	入力キュー，スケジュールサービス，ネームサービス，または RPC の障害発生時の処理	199
表 8-5	出力キューおよび出力メッセージ編集 UOC の障害発生時の処理	200
表 8-6	ジャーナル障害発生時の処理	201
表 8-7	送信バッファ障害発生時の処理	201
表 8-8	受信バッファ障害発生時の処理	202
表 8-9	編集バッファ障害発生時の処理	202
表 8-10	UAP 障害発生時の処理	202
表 8-11	MCF の障害発生時の処理	203
表 8-12	OpenTP1 の障害発生時の処理	203
表 8-13	コネクション障害の発生個所に応じた障害処理	204

表 8-14	問い合わせメッセージ送信時，応答メッセージ受信時の障害の発生個所に応じた障害処理	207
表 8-15	問い合わせメッセージ受信時，応答メッセージ送信時の障害の発生個所に応じた障害処理	208
表 8-16	一方送信メッセージ障害の発生個所に応じた障害処理	211
表 C-1	ERREVT2 の理由コード一覧	233
表 C-2	CERREVT の理由コード一覧	233

# 1

## 概要

TP1/NET/User Agent は、OpenTP1 システムの OSAS/UA プロトコルを支援するプログラムです。この章では、TP1/NET/User Agent の概要について説明します。

---

1.1 AP 間通信の概要

---

1.2 AP 間通信の形態

---

1.3 ソフトウェアの構成

---

## 1.1 AP 間通信の概要

---

AP 間通信とは、異なるシステムにあるアプリケーションプログラム間でのメッセージ送受信のことです。

TP1/NET/User Agent は、OpenTP1 システムに組み込まれ、OSAS/UA プロトコルを使用した AP 間通信を提供するプログラムです。TP1/NET/User Agent を使用すると、拡張 HNA の OSAS/UA プロトコルによってほかのオンラインシステムや分散機との水平、垂直分散型ネットワークシステムを構築できます。

OSAS/UA プロトコルとは、OSI 上位層（5 層、6 層、および 7 層の一部）の共通基盤である OSAS を使用して、AP 間通信サービスを提供するプロトコルです。

OpenTP1 システムを使用したネットワーク構成例を次の図に示します。

また、TP1/NET/User Agent の OSI 7 層構造の中での機能範囲を図 1-2 に示します。

図 1-1 OpenTP1 システムを使用したネットワーク構成例

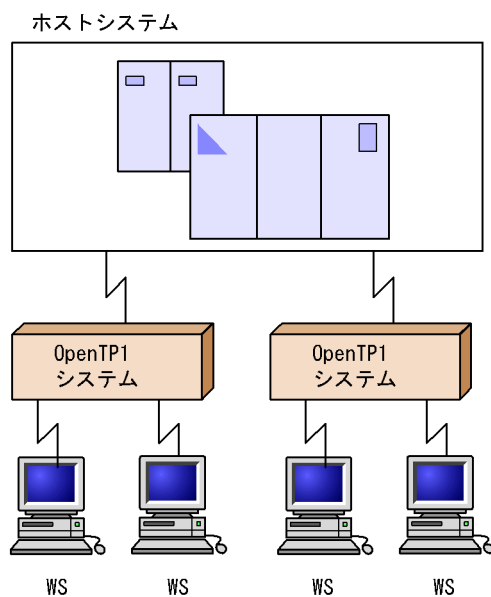
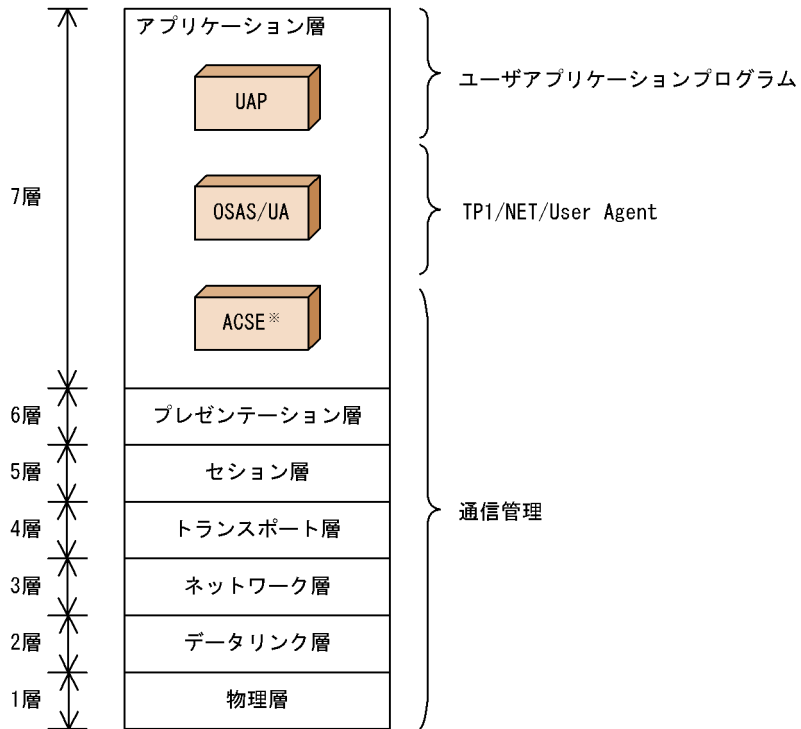


図 1-2 TP1/NET/User Agent の OSI7 層構造の中での機能範囲



注

ACSE : アソシエーション制御サービス要素

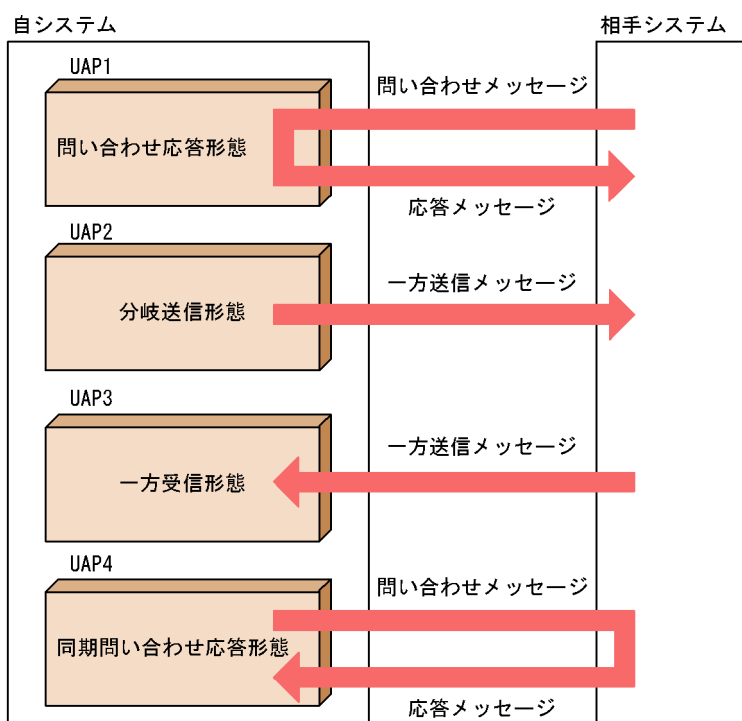
## 1.2 AP 間通信の形態

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

TP1/NET/User Agent を使用した AP 間通信の形態には、問い合わせ応答形態、分岐送信形態、一方受信形態、および同期問い合わせ応答形態があります。

TP1/NET/User Agent 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/User Agent を使用した AP 間通信の例



### (1) 問い合わせ応答形態

問い合わせ応答形態とは、通信相手システムから問い合わせメッセージを受信して、応答メッセージを送信する通信形態です。

### (2) 分岐送信形態

分岐送信形態とは、自システムから一方送信メッセージを通信相手システムへ送信する通信形態です。



(3) 一方受信形態

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

(4) 同期問い合わせ応答形態

同期問い合わせ応答形態とは、相手システムに問い合わせメッセージを送信し、応答メッセージを受信する通信形態です。

## 1.3 ソフトウェアの構成

---

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

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

### 1.3.1 前提プログラム

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

適用 OS : AIX

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

注

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

適用 OS : HP-UX

- R-18452-41K uCosminexus TP1/Server Base 07-00 以降
- R-18141-11K XNF/H/BASE
- R-F18141-112K XNF/H/WAN , または R-F18141-114K XNF/H/OSI Extension

注

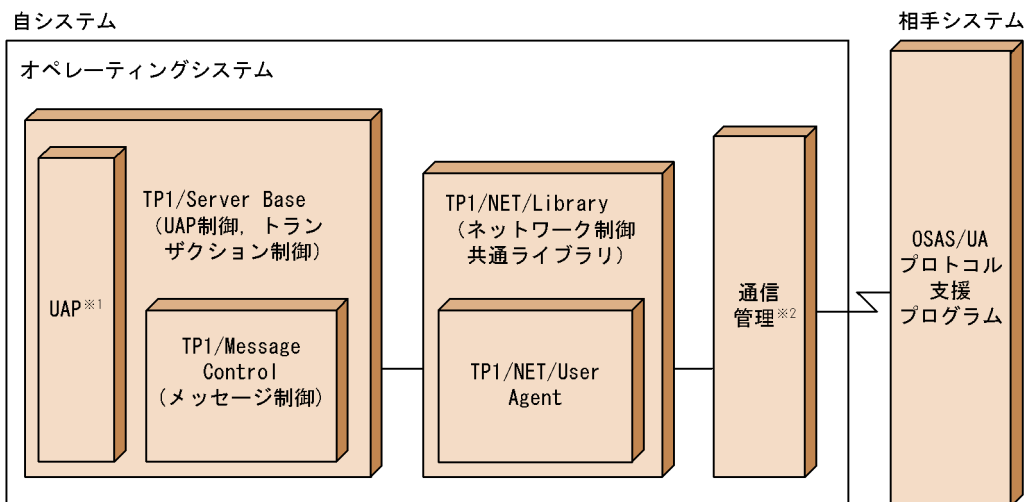
前提となる XNF のバージョンは、HP-UX 11i 以降に対応するバージョン以降です。

### 1.3.2 ソフトウェア構成の例

TP1/NET/User Agent を組み込んだソフトウェア構成の例を次の図に示します。

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

図 1-4 TP1/NET/User Agent を組み込んだソフトウェア構成



## 注 1

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

## 注 2

XNF/H または XNF/AS を使用できます。

表 1-1 TP1/NET/User Agent が適用する AP 間通信のプロトコル

プロトコル	主な通信相手
OSAS/UA プロトコル	VOS3 TMS-4V/SP の OSAS/UA/4VSP
	VOS3 XDM/DCCM3 の OSAS/UA2/DCCM3



# 2

## 機能

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

---

2.1 AP 間通信の仕組み

---

2.2 AP 間通信メッセージの送受信

---

## 2.1 AP 間通信の仕組み

TP1/NET/User Agent は、UA プロトコルデータ単位を最小の単位にして、メッセージの送受信を行います。TP1/NET/User Agent で送受信する UA 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/User Agent の UA プロトコルデータ単位

UA プロトコルデータ単位	機能概要	備考
UINT	初期設定	発呼だけのサポートです。
UOPN	個別開局	発呼だけのサポートです。
UEND	終了	-
UINQ	問い合わせ	UREP の応答が必要です。
UREP	応答	UINQ に対して必要です。
UBRD	一方送信	-
UERR	例外報告	-
UINF	状態通知	-

(凡例)

- : 備考はありません。

TP1/NET/User Agent の AP 間通信の仕組みについて説明します。

### 2.1.1 コネクションの確立と解放

TP1/NET/User Agent は、通信相手システムとの間で、論理的な通信路 (コネクション) を確立してメッセージを送受信します。コネクションは、OSAS/UA プロトコルのアソシエーションに対応します。

OSAS/UA プロトコルでは、データ送受信の多重化は UA で行います。

このため自システムと相手システムの同一 PSAP アドレス間では、1 本のコネクションしか確立できません。

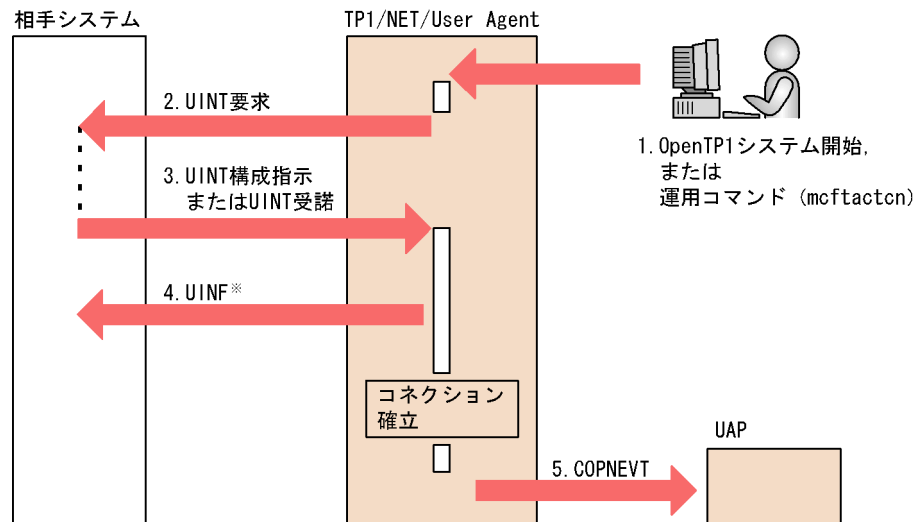
#### (1) コネクションの確立

TP1/NET/User Agent からコネクションを確立します。コネクションの確立方法には、次の二つがあります。

- OpenTP1 システムの開始・再開時の MCF 通信構成定義 (mcftalccn -i auto) による自動確立
- 運用コマンド (mcftactcn) 入力による手動確立

TP1/NET/User Agent からの接続の確立について、次の図に示します。

図 2-1 TP1/NET/User Agent からの接続の確立



注※ 相手システムからUINF構成指示を受信した場合だけ送信します。

1. OpenTP1 システムを開始・再開始します。または運用コマンド ( mcftactcn ) を入力します。
2. TP1/NET/User Agent は、MCF 通信構成定義情報に基づいて UINT 要求を作成し、相手システムに送信します。
3. 相手システムから UINT 構成指示または UINT 受諾を受信します。
4. 相手システムから UINF 構成指示を受信した場合には、処理完了 UINF を相手システムに送信すると、コネクションが確立します。UINF 受諾を受信した場合には、UINF 受諾受信を契機にコネクションが確立します。
5. TP1/NET/User Agent は、コネクションが確立すると状態通知イベント ( COPNEVT ) を発生させます。

コネクション確立時に、自システムと相手システムの UA の数を突き合わせます。UA の数が不一致の場合、一致しない UA を使用不可 ( 閉塞状態 ) として、一致する UA だけが使用可能となります。

コネクション確立時の、UA の数による TP1/NET/User Agent の処理を次の表に示します。

表 2-2 UA の数による TP1/NET/User Agent のコネクション確立

項番	UA の数	TP1/NET/User Agent の処理
1	相手システム < 自システム	不一致となった UA の論理端末を使用不可とし、一致した UA の論理端末を使用可能としてコネクションを確立します。 mfetalccn コマンドに -f オプションを指定した場合は、相手システムの構成に合わせて、自システムを縮退することを通知するメッセージ (KFCA13257-W) が出力されません。
2	相手システム = 自システム	該当するコネクション下の全論理端末を使用してコネクションを確立します。
3	相手システム > 自システム	コネクションの確立を拒否します。
4	相手システム = 0	コネクションの確立を拒否します。

ただし、MCF 通信構成定義で「個別開局」と指定された UA に関しては、該当する UA に関する論理端末を閉塞扱いとし、運用コマンド (mcftactle) で閉塞を解除するまではメッセージ送受信ができません。また、表 2-2 の項番 1 で使用不可となった論理端末については、閉塞を解除できません。

## (2) コネクションの解放

コネクションは、次に示す場合、正常に解放します。

- 相手システムから解放要求された場合
- 運用コマンド (mcftdctn) を入力した場合
- 定義コマンド (mcftcomn) で -x "termrls=nomal" を指定して、MCF を終了した場合

なお、次の場合はコネクションを強制的に解放し、障害通知イベント (CERREVT) を発生させます。

- 運用コマンド (mcftdctn -f) を受け付けた場合
- 送受信バッファ、および編集バッファの資源不足が発生した場合
- TP1/NET/User Agent 内で回復できない論理矛盾が発生した場合
- MCF 通信プロセスが異常終了した場合 (通信管理の機能に依存します。この場合は CERREVT を発生させません。)
- 相手システムに何らかの異常が発生し、コネクションが解放された場合

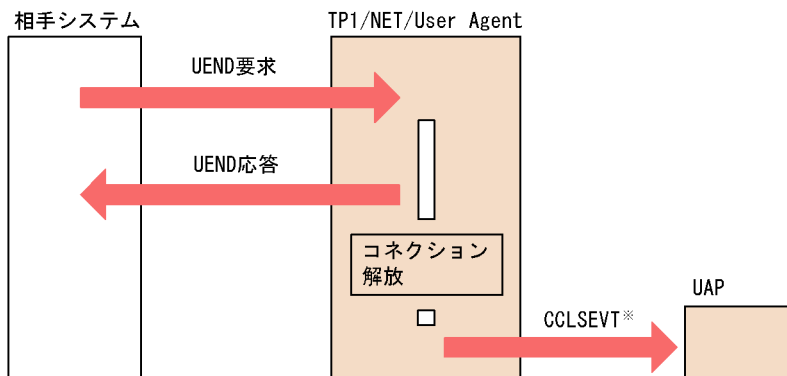
コネクションの正常解放はユーザ間で、データ転送の終了の同期合わせ後に行ってください。

相手システムからのコネクションの正常解放を図 2-2 に、自システムからのコネクションの解放を図 2-3 に示します。

コネクションが正常に解放されると、TP1/NET/User Agent は、状態通知イベント (CCLSEVT) を発生させます。

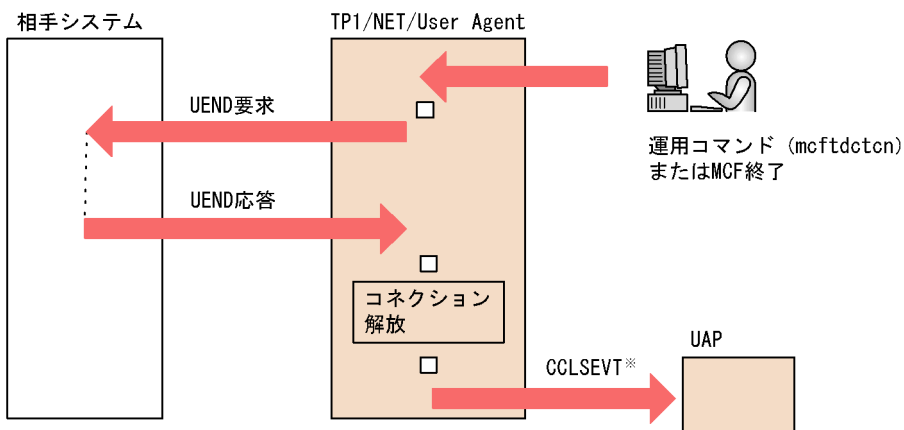


図 2-2 相手システムからの接続の解放



注※ MCF終了時は発生させません。

図 2-3 自システムからの接続の解放



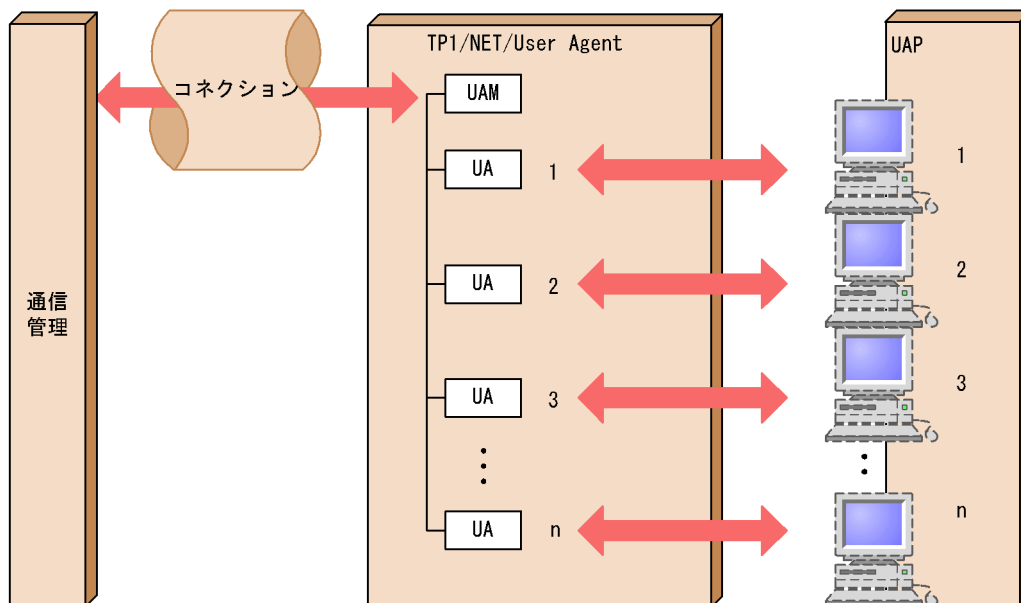
注※ MCF終了時は発生させません。

## 2.1.2 コネクションと論理端末の関係

TP1/NET/User Agent で扱う定義で、ユーザが意識するものとしては、コネクションと論理端末があります。コネクションは OSAS/UA プロトコルのアソシエーションに対応し、論理端末 (LE) は OSAS/UA プロトコルの UA に対応します。

コネクションは MCF の通信管理側の通信接点であり、MCF と通信管理はコネクション単位にメッセージの送受信をします。論理端末は MCF の UAP 側の通信接点であり、MCF と UAP は論理端末単位に送受信をします。コネクションと論理端末の関係を次の図に示します。

図 2-4 コネクションと論理端末の関係



### 2.1.3 経路交代

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

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

### 2.1.4 論理端末とアプリケーションの型の関係

論理端末は、通信相手システムとの間でメッセージを送受信するためのユーザインタフェースであり、OSAS/UA プロトコルの UA に対応します。

論理端末は、送受信するメッセージの形態で次に示す四つの端末タイプがあります。論理端末の端末タイプは、MCF の論理端末属性として、構成定義ファイルで設定します。

- request 型：問い合わせ型論理端末
- reply 型：応答型論理端末
- send 型：送信型論理端末
- receive 型：受信型論理端末

アプリケーションは、ユーザが送受信データの中で指定したアプリケーション名 をキーとして一つの UAP (MHP) プロセスで実行されます。アプリケーションには、処理形

態によって二つの型があり、MCF のアプリケーション属性の一つとして、アプリケーション定義ファイルで設定します。受信メッセージによって起動するアプリケーションの型を次に示します。

- 応答型：受信メッセージに対して応答する MHP の型
- 非応答型：受信メッセージに対して応答しない MHP の型

#### 注

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

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

表 2-3 論理端末、メッセージ、アプリケーションの型、UAP インタフェース、および通信形態の関係

論理端末の端末タイプ	メッセージの種類	受信メッセージのアプリケーションの型	UAP インタフェース	通信形態
reply (応答型論理端末)	問い合わせメッセージ (UINQ)	応答型	receive	問い合わせ応答
	応答メッセージ (UREP)		reply	
request (問い合わせ型論理端末)	一方送信メッセージ (UINQ)	任意	send	分岐送信
	問い合わせメッセージ (UINQ)		sendrecv (セグメント送信時、先頭セグメント受信時)	
	応答メッセージ (UREP)		recvsync (後続セグメント受信時)	
	一方送信メッセージ (UREP)	非応答型	receive	一方受信
send (送信型論理端末)	一方送信メッセージ (UBRD)	任意	send	分岐送信
receive (受信型論理端末)		非応答型	receive	一方受信

注

( ) 内は、UA プロトコルデータ単位を示します。

## 2.1.5 メッセージの分割と組み立て

TP1/NET/User Agent では、次の三つのデータ形式のメッセージを処理します。

- UA プロトコルデータ単位 (UPDU)
- UA サービスデータ単位 (UASDU)
- 論理メッセージ

以上三つのメッセージの分割、組み立てをしてメッセージの送受信をすると、各種資源 (メモリ、回線など) を節約できます。

### (1) UA プロトコルデータ単位

メッセージの最小データ送受信単位で、先頭、中間、最終 UPDU から構成されています。問い合わせ応答、および一方送信のデータ単位と、その他の制御用データ単位に分類されます。

### (2) UA サービスデータ単位

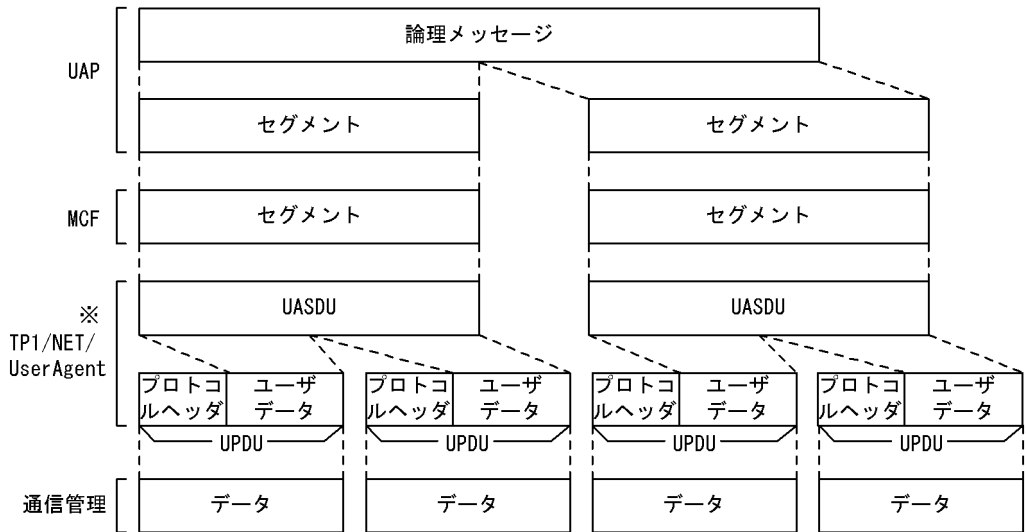
UPDU を組み立てた UA サービス利用者のデータ処理単位、セグメントに相当します。MCF 内のメッセージ送受信の基本単位で、UA サービスプリミティブに対応し、UPDU と同様に、先頭、中間、最終 UASDU で構成されています。

### (3) 論理メッセージ

複数セグメントから構成され、UAP が論理的に処理するデータ単位で、MHP の起動単位となります。入出力メッセージ編集 UOC による変換を伴う場合があります。

メッセージのデータ形式を次の図に示します。

図 2-5 メッセージのデータ形式



注※  
このバージョンでは、UASDUとUPDUの関係は1:1です。

TP1/NET/User Agent では、論理メッセージ単位に次のようなメッセージの送受信をします。

- 論理メッセージ受信  
UA サービスデータ単位に受信したメッセージを 1 論理メッセージにまとめて受信します。
- 論理メッセージ送信  
1 論理メッセージを UA サービスデータ単位に送信します。

## 2.1.6 UA の開局と閉局

TP1/NET/User Agent は、通信相手システムとの接続の確立時、UA を開局してメッセージを送受信します。

### (1) UA の開局

UA の開局方法には、次の二つがあります。

- コネクション確立時の自動開局
- 運用コマンド (mcftactle) による個別開局

コネクション確立時に、「個別開局」指定をした場合、または論理端末に障害があって閉局した場合、運用コマンド (mcftactle) で開局します。

MCF 通信構成定義 (mcftalcn) で「個別開局」指定 (-k each) されている場合、該当

## 2. 機能

するコネクション内の全 UA に関連する論理端末を閉塞扱います。なお、論理端末が閉塞状態であっても、該当する論理端末に対する UAP からの送信要求はできます。ただし、MCF の出力キューにキューイングするので、定義で指定した最大数を超えない範囲が限度となります。

### (a) コネクション確立時の UA の自動開局

MCF 通信構成定義 (mcftalccn) で「一括開局」が指定 (-k together) されている場合、コネクション確立時に、全 UA がメッセージ送受信可能状態になります。

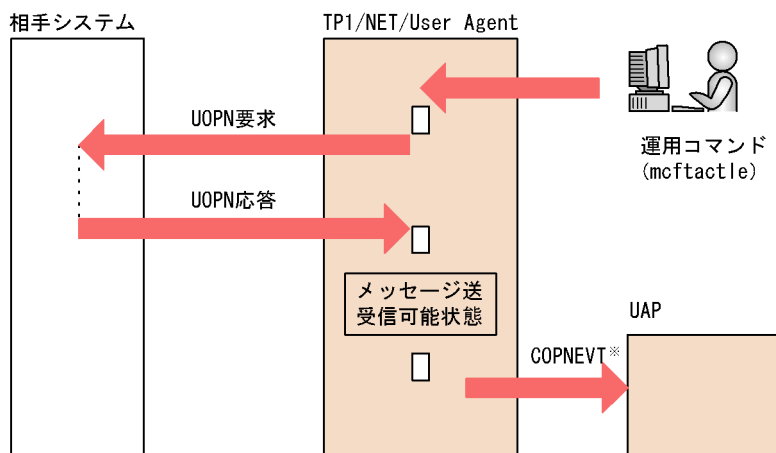
UA が開局すると、該当する UA の論理端末の閉塞状態を解除します。それと同時に、MCF 通信構成定義 (mcftalccn) で「論理端末の閉塞解除時に状態通知イベント (COPNEVT) を起動する」が指定 (-f "leopnevt=use") されている場合は、COPNEVT が起動されます。

### (b) 運用コマンドによる UA の個別開局

TP1/NET/User Agent は、開局されていない UA に対して運用コマンド (mcfactle) を受け付けると、相手システムに UOPN 要求を送信します。そして相手システムから UOPN 応答を受信すると、UA が開局し、メッセージ送受信可能状態になります。

UA が開局すると、該当する UA の論理端末の閉塞状態を解除します。運用コマンドによる UA の個別開局の処理の流れを次の図に示します。

図 2-6 運用コマンドによる UA の個別開局



### 注

COPNEVT は、MCF 通信構成定義 (mcftalccn) で「論理端末の閉塞解除時に状態通知イベント (COPNEVT) を起動する」が指定 (-f "leopnevt=use") されている場合だけ起動されます。

## (2) UA の閉局

UA の閉局方法には、次の三つがあります。

- 運用コマンド (mcftdctl) による強制閉局
- 相手システムからの強制閉局
- MCF 検出障害時の強制閉局

また、TP1/NET/User Agent は UA を閉局するとき、UERR を使用します。

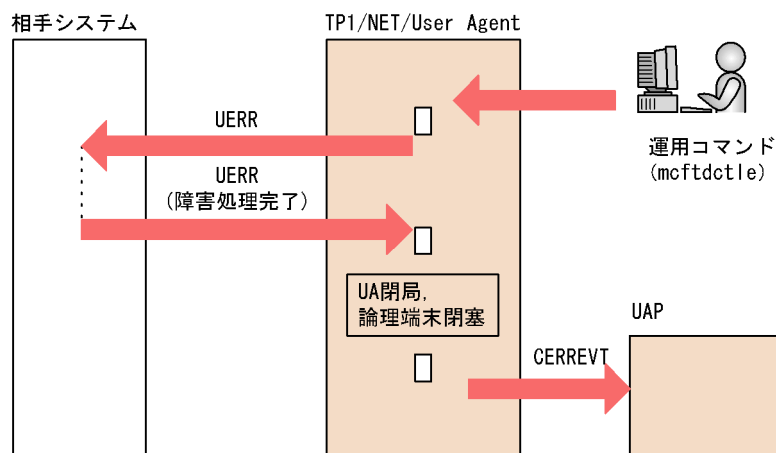
### (a) 運用コマンドによる UA の閉局

TP1/NET/User Agent は、運用コマンド (mcftdctl) を受け付けると、UERR を送信します。そして相手システムからの UERR (障害処理完了) を受信し、UA が閉局します。

UA が閉局すると、TP1/NET/User Agent は関連する論理端末を閉塞すると同時に、障害通知イベント (CERREVT) を起動します。

運用コマンドによる UA の閉局の処理の流れを次の図に示します。

図 2-7 運用コマンドによる UA の閉局

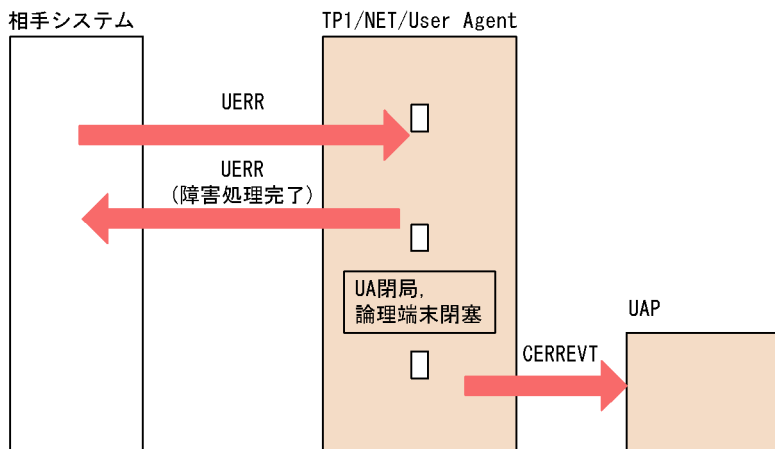


### (b) 相手システムからの UA の閉局

相手システムから UERR を受信すると、TP1/NET/User Agent では UERR (障害処理完了) を送信し、UA を閉局します。UA を閉局すると、TP1/NET/User Agent は関連する論理端末を閉塞すると同時に、障害通知イベント (CERREVT) を起動します。

相手システムからの UA の閉局の処理の流れを次の図に示します。

図 2-8 相手システムからの UA の閉局



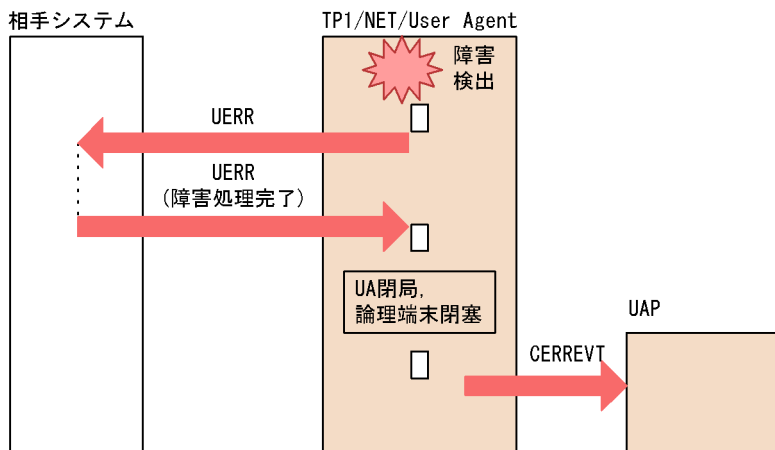
(c) MCF 検出障害時の UA の閉局

TP1/NET/User Agent は、UAP 異常終了、キュー障害などを検出した場合、UERR を送信します。そして相手システムからの UERR (障害処理完了) を受信し、UA が閉局します。

UA が閉局すると、TP1/NET/User Agent は関連する論理端末を閉塞すると同時に、障害通知イベント (CERREVT) を起動します。

MCF 検出障害時の UA の閉局の処理の流れを次の図に示します。

図 2-9 MCF 検出障害時の UA の閉局





## 2.2 AP 間通信メッセージの送受信

---

TP1/NET/User Agent は、AP 間通信メッセージの送受信を制御します。TP1/NET/User Agent で扱う AP 間通信メッセージには次の四つがあります。

- 問い合わせメッセージ
- 応答メッセージ
- 一方送信メッセージ（分岐送信）
- 一方送信メッセージ（一方受信）

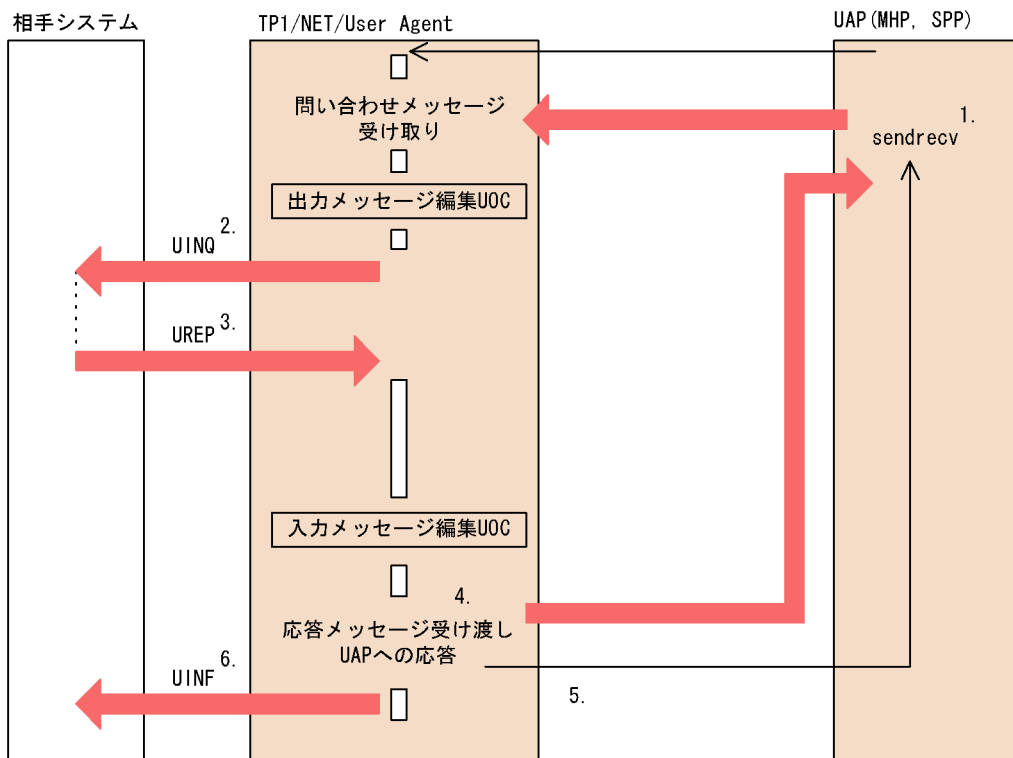
それぞれのメッセージの送受信について説明します。

### 2.2.1 問い合わせメッセージの送信

TP1/NET/User Agent が、相手システムに問い合わせメッセージ（UINQ）を送信し、相手システムからの応答メッセージ（UREP）を受信する処理の流れを説明します。この場合、問い合わせメッセージ関数（sendrecv）を発行した UAP に、応答メッセージが返ります。

自システムからの問い合わせメッセージの送信と、それに対する相手システムからの応答メッセージを受信する処理の流れを次の図に示します。

図 2-10 問い合わせメッセージの送信と応答メッセージの受信



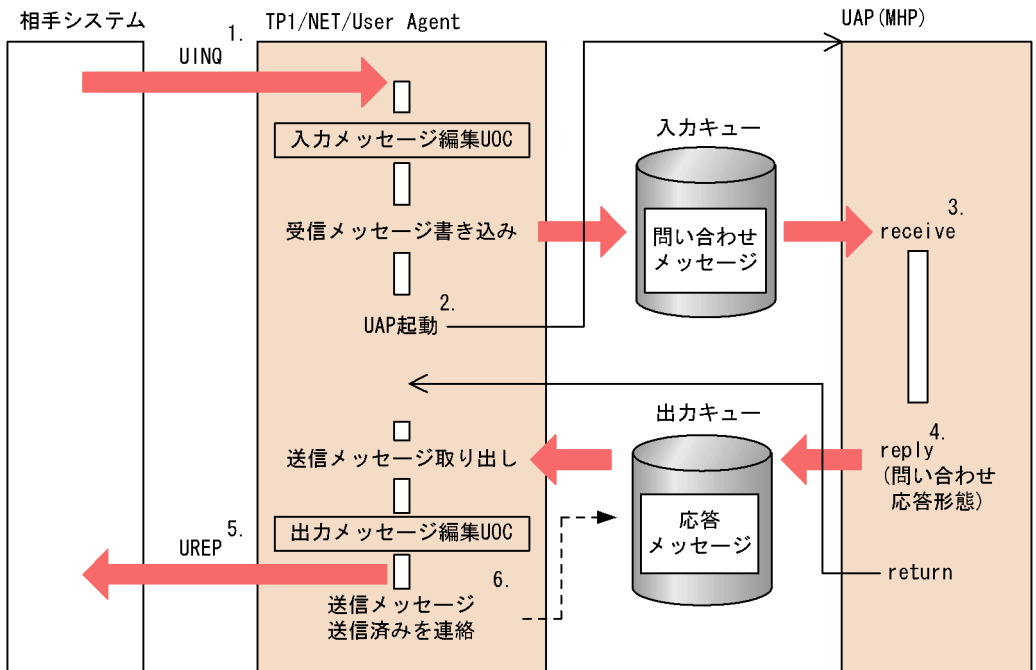
1. MHPまたはSPP から request 型論理端末へ、同期型の問い合わせメッセージを送信します。
2. TP1/NET/User Agent は相手システムに UINQ を送信します。
3. 相手システムからの UREP を受信します。
4. 相手システムからの応答メッセージを UAP に受け渡します。
5. 問い合わせメッセージを送信した UAP に制御が渡され、応答メッセージを受信します。
6. 3. の UREP の処理完了報告が必要な場合、UAP へのリターンが正常に終了した時点で、UINF を送信します。

### 2.2.2 応答メッセージの送信

TP1/NET/User Agent が、相手システムからの問い合わせメッセージ (UINQ) を受信し、相手システムに 応答メッセージ (UREP) を送信する処理の流れを説明します。

相手システムからの問い合わせメッセージの受信と、それに対する 応答メッセージの送信の処理の流れを次の図に示します。

図 2-11 問い合わせメッセージの受信と応答メッセージの送信



1. 相手システムから reply 型論理端末あての UINQ (問い合わせメッセージ) を受信します。
2. TP1/NET/User Agent は相手システムからの問い合わせメッセージを入力キューに書き込み、MHP を起動します。
3. MHP は関数を発行してメッセージを受け取ります。
4. MHP は、応答メッセージを発行します。
5. MHP が正常に return した場合、相手システムに UREP (処理完了報告不要) を送信します。
6. 5. のあと、出力キューの応答メッセージを送信済みにします。

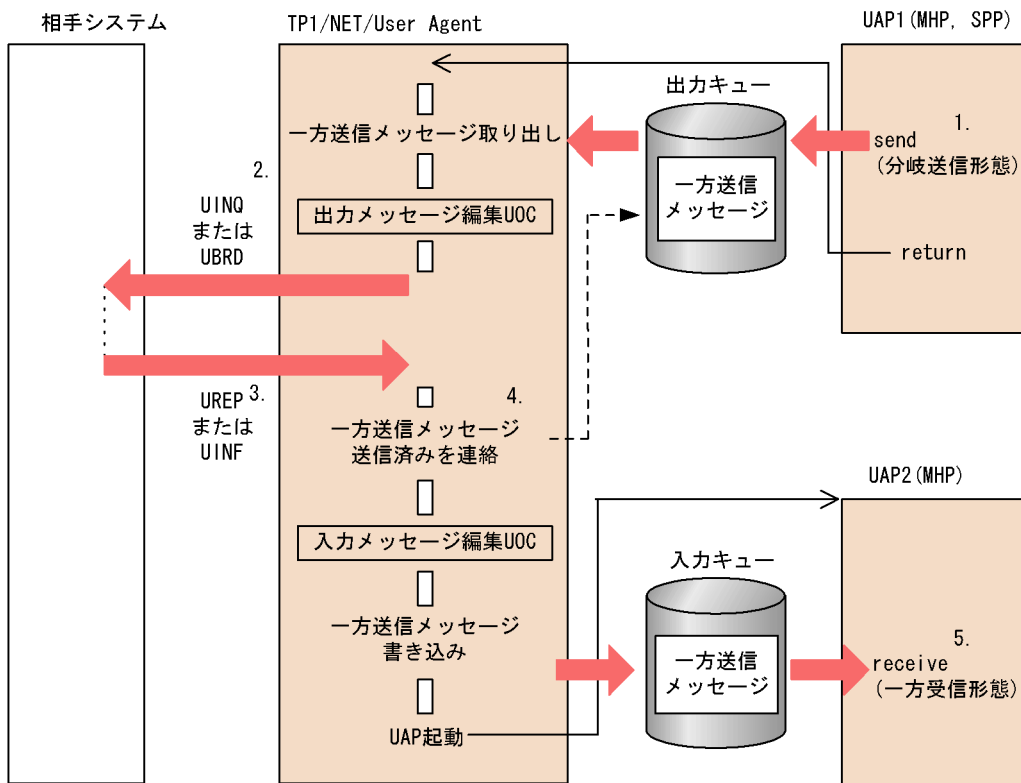
### 2.2.3 一方送信メッセージの送信と受信

TP1/NET/User Agent が、相手システムに一方送信メッセージ (UINQ または UBRD) を送信し、相手システムからの一方送信メッセージ (UREP または UINF) を受信する場合の処理の流れを説明します。

一方送信メッセージの送信と受信の処理の流れを、次の図に示します。

## 2. 機能

図 2-12 一方送信メッセージの送信と受信



1. MHP または SPP から request 型論理端末、または send 型論理端末あてに、一方送信メッセージを送信します。
  2. TP1/NET/User Agent は相手システムに UINQ ( send 型の場合は UBRD ) を送信します。
  3. 相手システムからの UREP ( send 型の場合は UINF ) を受信します。
  4. 3. の時点で、送信メッセージを送信済みとします。
  5. 送信時とは別の MHP に制御が渡され、一方送信メッセージを受信します。
- なお、send 型の場合は、UINF を TP1/NET/User Agent で受信するだけで、MHP での UINF の受信はできません。

# 3

## メッセージ送受信インタフェース

TP1/NET/User Agent を使用してメッセージを送受信する場合、ユーザは個別の業務に対応させるため、UAP を作成します。UAP は、C 言語または COBOL 言語で作成できます。TP1/NET/User Agent は、UAP の作成、運用を支援しています。この章では、TP1/NET/User Agent に関連するメッセージ送受信のユーザアプリケーションプログラムのインタフェースについて説明します。

---

### メッセージ送受信インタフェースの一覧

---

dc\_mcf\_receive - メッセージの受信 (C 言語)

---

dc\_mcf\_recvsync - 同期型のメッセージの受信 (C 言語)

---

dc\_mcf\_reply - 応答メッセージの送信 (C 言語)

---

dc\_mcf\_resend - メッセージの再送 (C 言語)

---

dc\_mcf\_send - 一方送信メッセージの送信 (C 言語)

---

dc\_mcf\_sendrecv - 同期型のメッセージの送受信 (C 言語)

---

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

---

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

---

CBLDCMCF('REPLY ') - 応答メッセージの送信 (COBOL 言語)

---

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

---

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

---

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

---

### 3. メッセージ送受信インタフェース

---

RECEIVE - メッセージの受信 (データ操作言語)

---

SEND - メッセージの送信 (データ操作言語)

---

ユーザアプリケーションプログラム作成例

---

## メッセージ送受信インタフェースの一覧

TP1/NET/User Agent で使用するメッセージ送受信インタフェースについて、C 言語、COBOL 言語、およびデータ操作言語に分けて説明します。

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

### C 言語のメッセージ送受信

C 言語でメッセージ送受信をする場合は、OpenTP1 で提供する関数を使用して UAP を作成します。

メッセージ送受信の関数を次の表に示します。

表 3-1 メッセージ送受信のライブラリ関数 (C 言語)

関数名	機能
dc_mcf_receive	一方送信メッセージの受信
dc_mcf_recvsync	同期型の応答メッセージの受信
dc_mcf_reply	応答メッセージの送信
dc_mcf_resend	メッセージの再送
dc_mcf_send	一方送信メッセージの送信
dc_mcf_sendrecv	同期型のメッセージの送受信

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

### COBOL 言語のメッセージ送受信

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

メッセージ送受信の関数に対応するプログラムを次の表に示します。

表 3-2 メッセージ送受信の文 (COBOL 言語)

プログラム名	データ名	機能
CBLDCMCF	'RECEIVE '	一方送信メッセージの受信
	'RECVSYNC'	同期型の応答メッセージの受信
	'REPLY '	応答メッセージの送信
	'RESEND '	メッセージの再送

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

プログラム名	データ名	機能
	'SEND'	一方送信メッセージの送信
	'SENDRECV'	同期型のメッセージの送受信

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

#### データ操作言語（COBOL 言語）のメッセージ送受信

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

TP1/NET/User Agent で固有の通信文について、次の表に示します。

表 3-3 メッセージ送受信の通信文（データ操作言語）

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

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



## dc\_mcf\_receive - メッセージの受信 (C 言語)

### 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

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

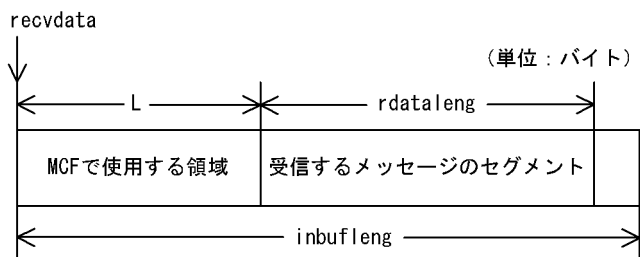
### 機能

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

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

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

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



## UAP で値を設定する引数

action

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

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

DCMCFRST

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

DCMCFSEG

中間セグメントまたは最終セグメントを受信する場合に設定します。

DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

commform

DCNOFLAGS を設定します。

termnam

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

処理終了後、termnam には OpenTP1 から値が返ります。

resv01

ヌル文字を設定します。

recvdata

セグメントを受信する領域を設定します。

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

処理終了後、recvdata には OpenTP1 から値が返ります。

inbufleng

セグメントを受信する領域の長さを設定します。

## OpenTP1 から値が返される引数

termnam

先頭セグメントを受信する場合だけ、入力元の論理端末名称が返されます。

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

recvdata

受信したセグメントの内容が返されます。

rdataleang

受信したセグメントの長さが返されます。

time

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

## リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71000	先頭セグメントを受信する dc_mcf_receive 関数を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、action に DCMCFSEG を設定して dc_mcf_receive 関数を呼び出してください。
DCMCFRTN_71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する dc_mcf_receive 関数を呼び出しています。直前に呼び出した dc_mcf_receive 関数でメッセージはすべて受信しました。このリターン値が返されたあとに、再び dc_mcf_receive 関数を呼び出した場合は、リターン値 DCMCFRTN_72000 が返されます。
DCMCFRTN_71002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。
DCMCFRTN_72000	< MHP の実行でリターンした場合 > <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> < SPP の実行でリターンした場合 > SPP では dc_mcf_receive 関数を呼び出せません。
DCMCFRTN_72001	termnam に設定した論理端末名称が間違っています。
DCMCFRTN_72013	inbufleng の指定値を超えるセグメントを受信しました。inbufleng の指定値を超えた部分は切り捨てられました。

### 3. メッセージ送受信インタフェース

dc\_mcf\_receive - メッセージの受信 (C 言語)

リターン値	意味
DCMCFRTN_72016	action に設定した値が間違っています。
	resv01 に設定した値が間違っています。
	引数に設定した値に間違いがあります。
DCMCFRTN_72024	commform に設定した値が間違っています。
DCMCFRTN_72025	action に設定したセグメント種別 (DCMCFRST または DCMCFSEG) の値が間違っています。
DCMCFRTN_72036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

## dc\_mcf\_recvsync - 同期型のメッセージの受信 (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 resv02)
```

K&R 版 C の形式

```
#include <dcmcf.h>
int dc_mcf_recvsync(action, commform, termnam, resv01,
                   recvdata, rdataleng, inbufleng,
                   time, resv02)
```

```
DCLONG    action;
DCLONG    commform;
char      *termnam;
char      *resv01;
char      *recvdata;
DCLONG    *rdataleng;
DCLONG    inbufleng;
DCLONG    *time;
DCLONG    resv02;
```

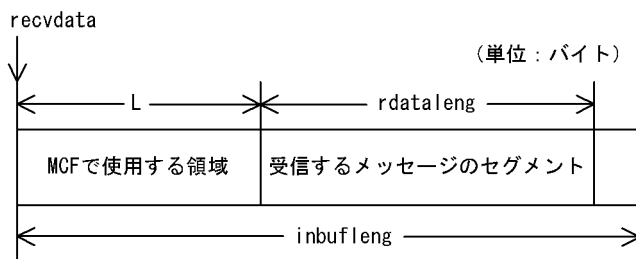
### 機能

相手システムから届いた同期型の受信メッセージのうち、先頭セグメント以外の後続する一つのセグメントを受信します。先頭セグメントを受信する `dc_mcf_sendrecv` 関数のあとに、後続するセグメントの数だけ `dc_mcf_recvsync` 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。

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

### 3. メッセージ送受信インタフェース

dc\_mcf\_recvsync - 同期型のメッセージの受信 (C 言語)



#### UAP で値を設定する引数

action

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

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

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

dc\_mcf\_sendrecv 関数の termnam パラメタで指定した入力元の論理端末名称を設定します。論理端末名称の長さは最大 8 バイトです。論理端末名称の後ろにはヌル文字を付けてください。

resv01

ヌル文字を設定します。

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。

resv02

DCNOFLAGS を設定します。

### 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。

rdata leng

受信したセグメントの長さが返されます。

time

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

### リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71001	メッセージの最終セグメントを受信したあとで、その次のセグメントを受信する 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 関数でその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、再び次のセグメントを受信する 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_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。
DCMCFRTN_72000	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	termnam に設定した論理端末名称が間違っています。 dc_mcf_recvsync 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72013	inbufleng の指定値を超えるセグメントを受信しました。inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRTN_72016	action に設定した値が間違っています。 resv01 または resv02 に設定した値が間違っています。 引数に設定した値に間違いがあります。

### 3. メッセージ送受信インタフェース

dc\_mcf\_recvsync - 同期型のメッセージの受信 (C 言語)

リターン値	意味
DCMCFRTN_72024	commform に設定した値が間違っています。
DCMCFRTN_72025	action に設定したセグメント種別 (DCMCFRST または DCMCFSEG) の値が間違っています。
DCMCFRTN_72036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。



## dc\_mcf\_reply - 応答メッセージの送信 (C 言語)

---

### 形式

ANSI C, C++ の形式

```
#include <dcmcf.h>
int dc_mcf_reply(DCLONG action, DCLONG commform, char *resv01,
                char *resv02, char *senddata, DCLONG sdataleNG,
                char *resv03, DCLONG opcd)
```

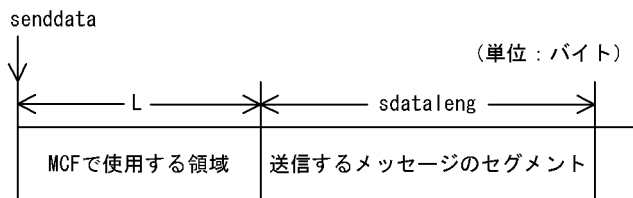
K&R 版 C の形式

```
#include <dcmcf.h>
int dc_mcf_reply(action, commform, resv01, resv02, senddata,
                sdataleNG, resv03, opcd)
DCLONG      action;
DCLONG      commform;
char        *resv01;
char        *resv02;
char        *senddata;
DCLONG      sdataleNG;
char        *resv03;
DCLONG      opcd;
```

### 機能

問い合わせをしたシステムへ送る応答メッセージのうち、一つのセグメントを送信します。セグメントの数だけ dc\_mcf\_reply 関数を呼び出すと、一つの論理メッセージを送信できます。

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



### UAP で値を設定する引数

action

メッセージの最終セグメントを送信するかどうかを、次の形式で設定します。

### 3. メッセージ送受信インタフェース

dc\_mcf\_reply - 応答メッセージの送信 (C 言語)

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

DCMCFESI

先頭セグメントまたは中間セグメントを送信する場合に設定します。

DCMCFEMI

最終セグメントを送信する場合に設定します。

メッセージが単一セグメントの場合も、DCMCFEMI を設定します。

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

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。

resv01 , resv02

ヌル文字を設定します。

senddata

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

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

sdataleug

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

resv03

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

## リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	sdataleng に 32000 バイトを超える値を設定しています。 MCF が終了処理中のため、メッセージの送信を受け付けられません。
DCMCFRTN_71003	メッセージキューが満杯です。
DCMCFRTN_71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRTN_71108	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	< MHP の実行でリターンした場合 > <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、dc_mcf_reply 関数を呼び出しています。</li> <li>非応答型のアプリケーションから dc_mcf_reply 関数を呼び出しています。</li> </ul>
	< SPP の実行でリターンした場合 > SPP では dc_mcf_reply 関数を呼び出せません。
DCMCFRTN_72001	dc_mcf_reply 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72005	先頭セグメントまたは中間セグメントを送信する dc_mcf_reply 関数で、長さが 0 バイトのセグメントを送信しています。
DCMCFRTN_72008	最終セグメントを送信する dc_mcf_reply 関数を呼び出したあとに、再び dc_mcf_reply 関数を呼び出しています。
	dc_mcf_execap 関数を呼び出して応答型のアプリケーションを起動させたあとに、dc_mcf_reply 関数を呼び出しています。
DCMCFRTN_72016	opcd に設定した値が間違っています。
	resv01, resv02, または resv03 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_72026	action に設定したセグメント種別 (DCMCFESI または DCMCFEMI) の値が間違っています。
DCMCFRTN_72041	単一セグメント送信時、sdataleng に 0 以下の値を設定しています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

## dc\_mcf\_resend - メッセージの再送 (C 言語)

---

### 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

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

### 機能

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

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

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

## UAP で値を設定する引数

action

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

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

DCMCFSEQ

再送するメッセージに出力通番を付け直す場合に設定します。

DCMCFNSEQ

再送するメッセージに出力通番を付け直さない場合に設定します。

DCMCFNORM

一般の一方送信メッセージとして再送する場合に設定します。

DCMCFPRIO

優先の一方送信メッセージとして再送する場合に設定します。

DCMCFLAST

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

commform

一方送信を示す、DCMCFOUT を設定します。

rtermnam

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

resv01

ヌル文字を設定します。

oseqid

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

DCMCFRID\_NORM

一般の一方送信メッセージを対象とする場合に設定します。

DCMCFRID\_PRIO

優先の一方送信メッセージを対象とする場合に設定します。

省略した場合は、DCMCFRID\_NORM (一般の一方送信メッセージを対象) が設定

3. メッセージ送受信インタフェース  
 dc\_mcf\_resend - メッセージの再送 (C 言語)

されます。DCMCFRID\_PRIO を設定した場合は、otermnam に出力先の論理端末名称を設定できません。

orgseq

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

otermnam

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

resv02, resv03, resv04

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

### リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_NOMSG	該当するメッセージがありません。
DCMCFRTN_BUF_SHORT	再送するメッセージのセグメントの長さが、UAP 共通定義 (mcfmuap -e) で指定した値を超えています。
DCMCFRTN_71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	MCF が終了処理中のため、メッセージの再送を受け付けられません。
DCMCFRTN_71003	メッセージキューが満杯です。
DCMCFRTN_71004	メッセージキューから取り出したメッセージを格納するバッファ (作業領域) を、メモリ上に確保できませんでした。
DCMCFRTN_71108	メッセージを再送しようとしたのですが、再送先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	< MHP の実行でリターンした場合 > <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、dc_mcf_resend 関数を呼び出しています。</li> <li>非トランザクション属性の MHP から、dc_mcf_resend 関数を呼び出しています。</li> </ul>

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

## 注意事項

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

### -e オプション

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

### -l オプション

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

## dc\_mcf\_send - 一方送信メッセージの送信 (C 言語)

---

### 形式

ANSI C, C++ の形式

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

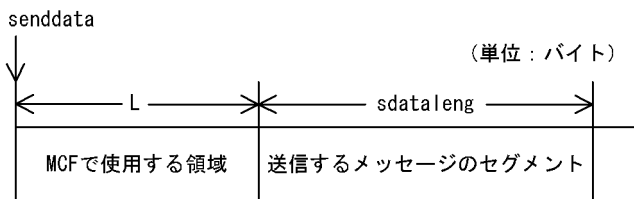
K&R 版 C の形式

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

### 機能

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

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



### UAP で値を設定する引数

action

メッセージの最終セグメントを送信するかどうか、優先か一般か、出力通番を付けるかどうか、および使用するバッファ形式を次の形式で設定します。



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

#### DCMCFESI

先頭セグメントまたは中間セグメントを送信する場合に設定します。

#### DCMCFEMI

最終セグメントを送信する場合に設定します。

メッセージが単一セグメントの場合も、DCMCFEMI を設定します。

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

#### DCMCFNORM

一般の一方送信メッセージとして送信する場合に設定します。

#### DCMCFPRIO

優先の一方送信メッセージとして送信する場合に設定します。

#### DCMCFSEQ

出力通番が必要な場合に設定します。

#### DCMCFNSEQ

出力通番が必要ない場合に設定します。

#### DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

#### DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

commform

一方送信を示す、DCMCFOUT を設定します。

termnam

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

resv01

ヌル文字を設定します。

senddata

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

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

### 3. メッセージ送受信インタフェース

dc\_mcf\_send - 一方送信メッセージの送信 (C 言語)

してください。

sdata Leng

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

resv02

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

#### リターン値

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

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

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

## dc\_mcf\_sendrecv - 同期型の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

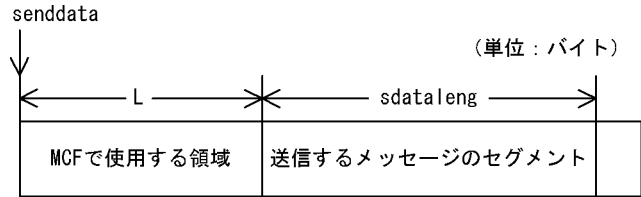
同期型でメッセージを送信したあと、同期型でメッセージを受信します。

dc\_mcf\_sendrecv 関数では、相手システムへ送るメッセージのうち、一つのセグメントを送信できます。セグメントの数だけ dc\_mcf\_sendrecv 関数を呼び出すと、一つの論理メッセージを送信できます。メッセージの最終セグメントを送信すると、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、そのメッセージの先頭セグメントを受信します。

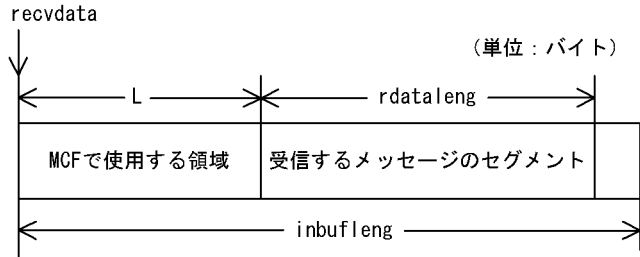
中間セグメントまたは最終セグメントを受信する場合は、dc\_mcf\_recvsync 関数を呼び出してください。

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

メッセージの最終セグメントを送信するかどうか、および使用するバッファ形式を、次の形式で設定します。

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

DCMCFESI

先頭セグメントまたは中間セグメントを送信する場合に設定します。

DCMCFEMI

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

メッセージの送信の終了を連絡するために、最後は必ずこの値を設定してください。この値を設定して dc\_mcf\_sendrecv 関数を呼び出すと、論理端末からの応答を待ちます。

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

同期型のメッセージの送受信を示す、DCMCFIO を設定します。

### 3. メッセージ送受信インタフェース

dc\_mcf\_sendrecv - 同期型のメッセージの送受信 (C 言語)

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

resv01

ヌル文字を設定します。

senddata

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

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

sdatale

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

recvdata

セグメントを受信する領域を設定します。

単一セグメントまたは最終セグメントを送信する dc\_mcf\_sendrecv 関数が終了すると、受信したメッセージの先頭セグメントが返されます。

処理終了後、recvdata には OpenTP1 から値が返ります。

inbufleng

セグメントを受信する領域の長さを設定します。

watchtime

dc\_mcf\_sendrecv 関数を呼び出してから終了するまでの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 (mcfmuap -t) で指定した、同期型送受信監視時間が設定されます。負の値を設定した場合は、時間を監視しません。

#### OpenTP1 から値が返される引数

recvdata

受信したメッセージの先頭セグメントの内容が返されます。

rdatale

受信したメッセージの先頭セグメントの長さが返されます。

time

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

### リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71002	メッセージキューへの出力処理中に障害が発生しました。 メッセージキューが閉塞されています。 メッセージキューが割り当てられていません。 sdataleng に 32000 バイトを超える値を設定しています。 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	メッセージキューが満杯です。
DCMCFRTN_71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRTN_71108	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	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	termnam に設定した論理端末名称が間違っています。 dc_mcf_sendrecv 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72005	先頭セグメントまたは中間セグメント送信時、sdataleng に 0 以下の値を設定しています。
DCMCFRTN_72013	inbufleng の指定値を超えるセグメントを受信しました。inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRTN_72016	action に設定した値が間違っています。 resv01 に設定した値が間違っています。 引数に設定した値に間違いがあります。
DCMCFRTN_72024	commform に設定した値が間違っています。
DCMCFRTN_72026	action に設定したセグメント種別 (DCMCFESI または DCMCFEMI) の値が間違っています。
DCMCFRTN_72036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
DCMCFRTN_72041	単一セグメント送信時、sdataleng に 0 以下の値を設定しています。
DCMCFRTN_73001	出力先の論理端末で障害が発生しました。
DCMCFRTN_73002	MCF 通信サービスで障害が発生しました。
DCMCFRTN_73008	論理端末が閉塞中、または MCF が終了処理中に、dc_mcf_sendrecv 関数を呼び出しました。

### 3. メッセージ送受信インタフェース

dc\_mcf\_sendrecv - 同期型のメッセージの送受信 (C 言語)

リターン値	意味
DCMCFRTN_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。 メッセージの編集エラーが発生しました。
DCMCFRTN_73015	出力先の論理端末は、ほかの UAP で仕掛り中です。
DCMCFRTN_73018	watchtime に設定した値が間違っています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。



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

---

### 形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

### 機能

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

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

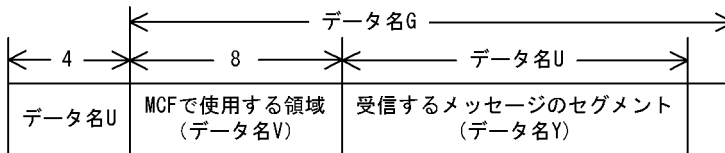
3. メッセージ送受信インタフェース  
 CBLDCMCF('RECEIVE') - メッセージの受信 (COBOL 言語)

- 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

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。

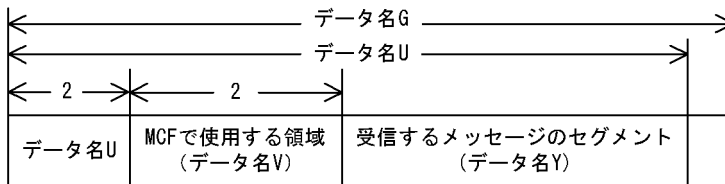
●バッファ形式 1 の場合

(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を「VALUE 'RECEIVE」」と設定します。

データ名 C

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

VALUE 'FRST'

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

VALUE 'SEG」

中間セグメントまたは最終セグメントを受信する場合に設定します。

データ名 D

空白を設定します。

データ名 G

セグメントを受信する領域の長さを設定します。

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

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白

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

データ名 N

MCF で使用する領域です。

データ名 O

空白を設定します。

データ名 P

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

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

データ名 Q

MCF で使用する領域です。

データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

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

データ名 V

【バッファ形式 1 の場合】 PIC X(8)

【バッファ形式 2 の場合】 PIC X(2)

MCF で使用する領域です。

### OpenTP1 から値が返されるデータ領域

データ名 B

ステータスコードが、5 けたの数字で返されます。

データ名 E

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

データ名 F

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

データ名 P

先頭セグメントを受信する場合、入力元の論理端末名称が返されます。

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

データ名 U

【バッファ形式 1 の場合】 PIC 9(9)

受信したセグメントの長さが返されます。

【バッファ形式 2 の場合】 PIC 9(4)

受信したセグメントの長さ + 4 が返されます。

データ名 Y

受信したセグメントの内容が返されます。

### 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。

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

ステータスコード	意味
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	<p>データ名 G の指定値を超えるセグメントを受信しました。データ名 G の指定値を超えた部分は切り捨てられました。</p> <p>32763 バイトを超えるセグメントを受信しました。32763 バイトを超えた部分は切り捨てられました。</p>
72016	<p>データ名 D に設定した値が間違っています。</p> <p>データ名 N またはデータ名 T に設定した値が間違っています。</p> <p>データ名 M7 に設定した値が間違っています。</p>
72024	データ名 O に設定した値が間違っています。
72025	データ名 C に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72036	データ名 G の指定値が不足しています。パッファ形式 1 の場合は 9 バイト以上、パッファ形式 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') - 同期型のメッセージの受信 (COBOL 言語)

---

### 形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

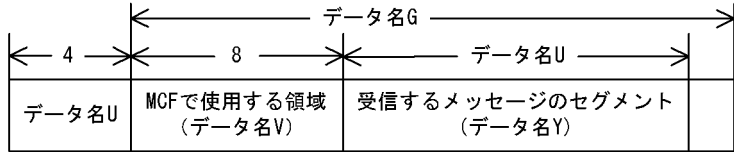
### 機能

相手システムから届いた同期型の受信メッセージのうち、先頭セグメント以外の後続する一つのセグメントを受信します。先頭セグメントを受信する CBLDBMCF('SENDRECV') のあとに、後続するセグメントの数だけ CBLDCMCF('RECVSYNC') を呼び出すと、一つの論理メッセージを受信できます。

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。

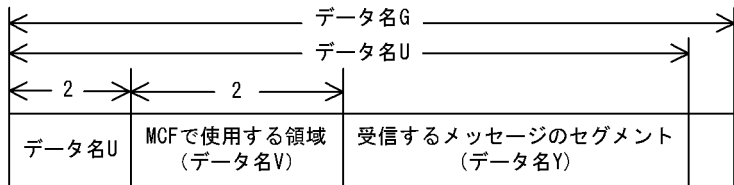
●バッファ形式 1 の場合

(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

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

データ名 C

メッセージの後続セグメントを受信する「VALUE 'SEG '」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, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

### 3. メッセージ送受信インタフェース

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

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

データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 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。相手システムから送信されるセグメントの最大長は 32763 バイトです。

### ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71001	メッセージの最終セグメントを受信したあとで、その次のセグメントを受信する CBLDCMCF('RECVSYNC') を呼び出しています。直前に呼び出した CBLDCMCF('RECVSYNC') でそのメッセージはすべて受信しました。このステータスコードが返されたあとに、再び次のセグメントを受信する 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。
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') を呼び出しています。
72001	データ名 P に設定した論理端末名称が間違っています。 CBLDCMCF('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。

### 3. メッセージ送受信インタフェース

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

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

## CBLDCMCF('REPLY ') - 応答メッセージの送信 (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 'REPLY ' .  
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 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) VALUE SPACE .  
02 データ名Q PIC X(8) VALUE SPACE .  
02 データ名R PIC X(8) VALUE SPACE .  
02 データ名T PIC X(28) VALUE LOW-VALUE .  
01 一意名3.  
02 データ名U PIC 9(x) COMP .  
02 データ名V PIC X(x) .  
02 データ名W PIC X(n) .
```

### 機能

問い合わせをしたシステムへ送る応答メッセージのうち、一つのセグメントを送信します。セグメントの数だけ CBLDCMCF('REPLY ') を呼び出すと、一つの論理メッセージを送信できます。

セグメントを送信する領域 (一意名 3 で示す領域) の形式を示します。

3. メッセージ送受信インタフェース  
 CBLDCMCF('REPLY') - 応答メッセージの送信 ( COBOL 言語 )

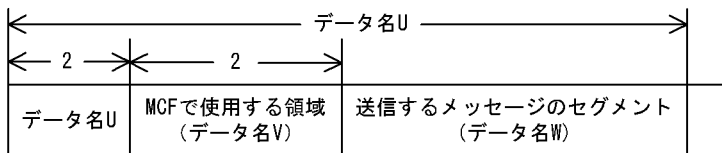
●バッファ形式 1 の場合

(単位 : バイト)



●バッファ形式 2 の場合

(単位 : バイト)



UAP で値を設定するデータ領域

データ名 A

応答メッセージの送信を示す要求コードを「VALUE 'REPLY        '」と設定します。

データ名 C , データ名 D

空白を設定します。

データ名 E , データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H

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

VALUE 'ESI    '

先頭セグメントまたは中間セグメントを送信する場合に設定します。

VALUE 'EMI    '

最終セグメントを送信する場合に設定します。

メッセージが単一セグメントの場合も、「VALUE 'EMI    '」を設定します。

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

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

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白

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

データ名 N

MCF で使用する領域です。

データ名 O, データ名 P, データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 U

【バッファ形式 1 の場合】 PIC 9(9)

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

【バッファ形式 2 の場合】 PIC 9(4)

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

データ名 V

【バッファ形式 1 の場合】 PIC X(8)

【バッファ形式 2 の場合】 PIC X(2)

MCF で使用する領域です。

### 3. メッセージ送受信インタフェース

CBLDCMCF('REPLY') - 応答メッセージの送信 ( COBOL 言語 )

#### データ名 W

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

#### OpenTP1 から値が返されるデータ領域

##### データ名 B

ステータスコードが, 5 けたの数字で返されます。

##### ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	データ名 U に 32000 バイトを超える値を設定しています。
	MCF が終了処理中のため, メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたが, 送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	<p>&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, CBLDCMCF('REPLY ') を呼び出しています。</li> <li>非応答型のアプリケーションから CBLDCMCF('REPLY ') 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('REPLY ') 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。
	CBLDCMCF('REPLY ') を呼び出せない論理端末を設定しています。
72005	先頭セグメントまたは中間セグメントを送信する CBLDCMCF('REPLY ') で, 長さが 0 バイトのセグメントを送信しています。
72008	最終セグメントを送信する CBLDCMCF('REPLY ') を呼び出したあとに, 再び CBLDCMCF('REPLY ') を呼び出しています。
	CBLDCMCF('EXECAP ') を呼び出して応答型のアプリケーションを起動させたあとに, CBLDCMCF('REPLY ') を呼び出しています。
72016	データ名 N またはデータ名 T に設定した値が間違っています。

3. メッセージ送受信インタフェース  
CBLDCMCF('REPLY') - 応答メッセージの送信 (COBOL 言語)

ステータスコード	意味
	データ名 M7 に設定した値が間違っています。
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
72020	データ名 I に設定した値が間違っています。
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 の単一セグメント送信時、データ名 U に 0 以下の値を設定しています。バッファ形式 2 の単一セグメント送信時、データ名 U に 4 以下の値を設定しています。
72045	データ名 R に設定した値が間違っています。
72047	データ名 R に設定した値が間違っています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

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

---

### 形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

### 機能

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



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

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

#### UAP で値を設定するデータ領域

データ名 A

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

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H, データ名 I

空白を設定します。

データ名 J

一般として再送するか優先として再送するかを設定します。

VALUE 'NORM'

一般の一方送信メッセージとして再送する場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージとして再送する場合に設定します。

空白

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

データ名 K

再送するメッセージに出力通番を付け直すかどうかを設定します。

VALUE 'SEQ    '

### 3. メッセージ送受信インタフェース

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

再送するメッセージに出力通番を付け直す場合に設定します。

VALUE 'NSEQ'

再送するメッセージに出力通番を付け直さない場合に設定します。

空白

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

データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6, データ名 M7

空白を設定します。

データ名 N

MCF で使用する領域です。

データ名 O

一方送信を示す「VALUE 'OUT '」を設定します。

データ名 P

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

データ名 Q, データ名 R

空白を設定します。

データ名 S

MCF で使用する領域です。

データ名 T

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

データ名 U

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

#### VALUE 'NORM'

一般の一方送信メッセージを対象とする場合に設定します。

#### VALUE 'PRIO'

優先の一方送信メッセージを対象とする場合に設定します。

#### 空白

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

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

#### データ名 V

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

#### データ名 W

最終出力通番を持つメッセージを再送するかどうかを設定します。

#### VALUE 'LAST'

最終出力通番を持つメッセージを再送する場合に設定します。

この値を設定した場合は、データ名 V に設定した値は無効となります。

#### 空白

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

#### データ名 X

MCF で使用する領域です。

### OpenTP1 から値が返されるデータ領域

#### データ名 B

ステータスコードが、5 けたの数字で返されます。

#### ステータスコード

ステータスコード	意味
00000	正常に終了しました。
70904	該当するメッセージがありません。
70905	再送するメッセージのセグメントの長さが、UAP 共通定義 (mcfmuap -e) で指定した値を超えています。
71002	メッセージキューへの出力処理中に障害が発生しました。 メッセージキューが閉塞されています。

### 3. メッセージ送受信インタフェース

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

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

#### 注意事項

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

#### -e オプション

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

#### -I オプション

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

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

---

### 形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

### 機能

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

セグメントを送信する領域 (一意名 3 で示す領域) の形式を示します。

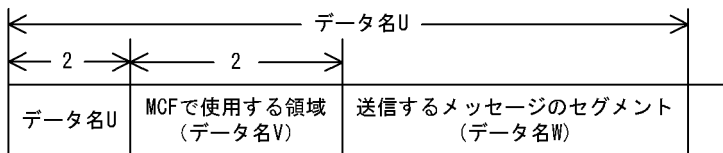
●バッファ形式 1 の場合

(単位: バイト)



●バッファ形式 2 の場合

(単位: バイト)



UAP で値を設定するデータ領域

データ名 A

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

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H

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

VALUE 'ESI    '

先頭セグメントまたは中間セグメントを送信する場合に設定します。

VALUE 'EMI    '

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

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

データ名 I

空白を設定します。

### 3. メッセージ送受信インタフェース

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

データ名 J

一般として送信するか優先として送信するかを設定します。

VALUE 'NORM'

一般の一方送信メッセージとして送信する場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージとして送信する場合に設定します。

空白

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

データ名 K

出力通番を付けるかどうかを設定します。

VALUE 'SEQ'

出力通番が必要な場合に設定します。

VALUE 'NSEQ'

出力通番が必要ない場合に設定します。

空白

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

データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白

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



#### データ名 N

MCF で使用する領域です。

#### データ名 O

一方送信を示す「VALUE 'OUT '」を設定します。

#### データ名 P

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

#### データ名 Q, データ名 R

空白を設定します。

#### データ名 T

MCF で使用する領域です。

#### データ名 U

【バッファ形式 1 の場合】 PIC 9(9)

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

【バッファ形式 2 の場合】 PIC 9(4)

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

#### データ名 V

【バッファ形式 1 の場合】 PIC X(8)

【バッファ形式 2 の場合】 PIC X(2)

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

### OpenTP1 から値が返されるデータ領域

#### データ名 B

ステータスコードが、5 けたの数字で返されます。

### 3. メッセージ送受信インタフェース

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

#### ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	データ名 U に 32000 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する CBLDCMCF('RECEIVE') を呼び出す前に、 CBLDCMCF('SEND') を呼び出しています。
	< SPP の実行でリターンした場合 > トランザクションでない SPP の処理から、CBLDCMCF('SEND') を 呼び出しています。
72001	データ名 P に設定した論理端末名称が間違っています。
	CBLDCMCF('SEND') を呼び出せない論理端末を設定しています。
72005	バッファ形式 1 の先頭セグメントまたは中間セグメント送信時、データ名 U に 0 以下の値を設定しています。または、バッファ形式 2 の先頭セグメントまたは中間セグメント送信時、データ名 U に 4 以下の値を設定しています。
72016	データ名 N またはデータ名 T に設定した値が間違っています。
	データ名 J に設定した値が間違っています。
	データ名 M1 に設定した値が間違っています。
	データ名 M7 に設定した値が間違っています。
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
72020	データ名 I に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72026	データ名 H に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72041	バッファ形式 1 の単一セグメント送信時、データ名 U に 0 以下の値を設定しています。または、バッファ形式 2 の単一セグメント送信時、データ名 U に 4 以下の値を設定しています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

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

---

### 形式

#### PROCEDURE DIVISION の指定

```
CALL 'CBLDCMCF' USING 一意名1 一意名2 一意名3 一意名4
```

#### DATA DIVISION の指定

```
01 一意名1.  
02 データ名A PIC X(8) VALUE 'SENDRECV'.  
02 データ名B PIC X(5).  
02 FILLER PIC X(3).  
02 データ名C PIC X(4) VALUE SPACE.  
02 データ名D PIC X(4) VALUE SPACE.  
02 データ名E PIC 9(8).  
02 データ名F PIC 9(8).  
02 データ名G PIC 9(9) COMP.  
02 データ名H PIC X(4).  
02 データ名I PIC X(4) VALUE SPACE.  
02 データ名J PIC X(4) VALUE SPACE.  
02 データ名K PIC X(4) VALUE SPACE.  
02 データ名L PIC X(8) VALUE SPACE.  
02 データ名M1 PIC X(4) VALUE SPACE.  
02 データ名M2 PIC X(8) VALUE SPACE.  
02 データ名M3 PIC X(4) VALUE SPACE.  
02 データ名M4 PIC 9(9) COMP VALUE ZERO.  
02 データ名M5 PIC 9(9) COMP.  
02 データ名M6 PIC X(1) VALUE SPACE.  
02 データ名M7 PIC X(1).  
02 データ名N PIC X(14) VALUE LOW-VALUE.  
01 一意名2.  
02 データ名O PIC X(4) VALUE 'IO'.  
02 データ名P PIC X(8).  
02 データ名Q PIC X(8) VALUE SPACE.  
02 データ名R PIC X(8) VALUE SPACE.  
02 データ名T PIC X(28) VALUE LOW-VALUE.  
01 一意名3.  
02 データ名U PIC 9(x) COMP.  
02 データ名V PIC X(x).  
02 データ名W PIC X(n).  
01 一意名4.  
02 データ名X PIC 9(x) COMP.  
02 データ名Y1 PIC X(x) VALUE SPACE.  
02 データ名Y2 PIC X(1).  
02 データ名Z PIC X(n).
```

### 機能

同期型でメッセージを送信したあと、同期型でメッセージを受信します。

### 3. メッセージ送受信インタフェース

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

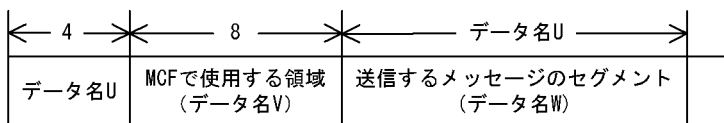
CBLDCMCF('SENDRECV') では、相手システムへ送るメッセージのうち、一つのセグメントを送信できます。セグメントの数だけ CBLDCMCF('SENDRECV') を呼び出すと、一つの論理メッセージを送信できます。

メッセージの最終セグメントを送信すると、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、そのメッセージの先頭セグメントを受信します。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、CBLDCMCF('RECVSYNC') を呼び出してください。

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。

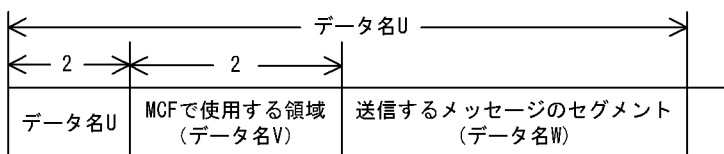
#### ●バッファ形式 1 の場合

(単位: バイト)



#### ●バッファ形式 2 の場合

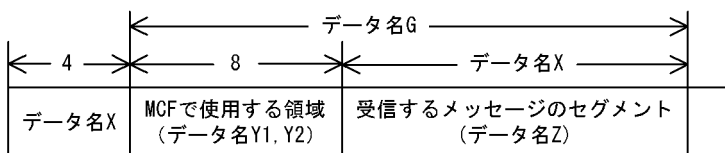
(単位: バイト)



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。

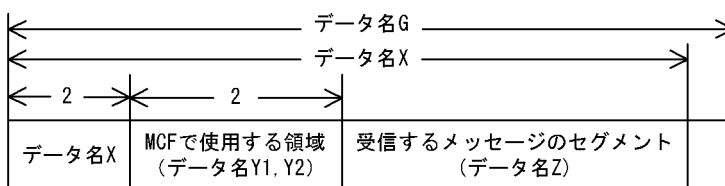
#### ●バッファ形式 1 の場合

(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

同期型のメッセージの送受信を示す要求コードを「VALUE 'SENDRECV'」と設定しま

す。

データ名 C, データ名 D

空白を設定します。

データ名 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

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

VALUE 'ESI' '

先頭セグメントまたは中間セグメントを送信する場合に設定します。

VALUE 'EMI' '

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

メッセージの送信の終了を連絡するために、最後は必ずこの値を設定してください。  
この値を設定して CBLDCMCF('SENDRECV') を呼び出すと、論理端末からの応答  
を待ちます。

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 (mcfmuap -t) で指定した、同期  
型送受信監視時間が設定されます。負の値を設定した場合は、時間を監視しません。

データ名 M6

空白を指定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

### 3. メッセージ送受信インタフェース

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

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

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

データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 U

【バッファ形式 1 の場合】 PIC 9(9)

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

【バッファ形式 2 の場合】 PIC 9(4)

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

データ名 V

【バッファ形式 1 の場合】 PIC X(8)

【バッファ形式 2 の場合】 PIC X(2)

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

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。相手システムから送信されるセグメントの最大長は 32763 バイトです。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの入出力処理時に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	データ名 U に 32000 バイトを超える値を設定しています。

### 3. メッセージ送受信インタフェース

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

ステータスコード	意味
	MCF が終了処理中のため、メッセージの送受信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
72000	先頭セグメントを受信する CBLDCMCF('RECEIVE') を呼び出す前に、CBLDCMCF('SENDRECV') を呼び出します。
72001	データ名 P に設定した論理端末名称が間違っています。 CBLDCMCF('SENDRECV') を呼び出せない論理端末を設定しています。
72005	バッファ形式 1 の先頭セグメントまたは中間セグメント送信時、データ名 U に 0 以下の値を設定しています。または、バッファ形式 2 の先頭セグメントまたは中間セグメント送信時、データ名 U に 4 以下の値を設定しています。
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確保してください。
72041	バッファ形式 1 の単一セグメント送信時、データ名 U に 0 以下の値を設定しています。または、バッファ形式 2 の単一セグメント送信時、データ名 U に 4 以下の値を設定しています。
73001	出力先の論理端末で障害が発生しました。
73002	MCF 通信サービスで障害が発生しました。
73008	論理端末が閉塞中、または MCF が終了処理中に、CBLDCMCF('SENDRECV') を呼び出しました。
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。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。



## RECEIVE - メッセージの受信 (データ操作言語)

---

### 形式

DATA DIVISION (通信記述項) の指定

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

PROCEDURE DIVISION (通信文) の指定

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

### 機能

次に示す CALL インタフェースの機能を実現します。

- 一方送信メッセージの受信 CBLDCMCF('RECEIVE ')

### 通信記述項に設定する項目

FOR 句

次のどちらかの値を指定します。

INPUT

一方送信メッセージの受信

I-O

問い合わせメッセージの受信

STATUS KEY 句

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

SYMBOLIC TERMINAL 句

入力元の論理端末名称を参照するデータ項目を指定します。

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

MESSAGE DATE 句

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

MESSAGE TIME 句

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

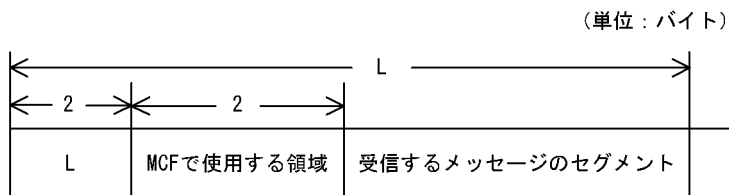
通信文に指定する項目

FIRST

先頭セグメントを受信するときに指定します。

一意名 1

セグメントを受信するデータ項目を指定します。相手システムから送信されるセグメントの最大長は 32763 バイトです。一意名 1 の形式を次に示します。



ステータスコード

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

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

ステータスコード	意味
72000	<p>&lt; MHP の実行でリターンした場合 &gt;</p> <ul style="list-style-type: none"> <li>非同期型の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、そのほかのセグメントを受信する RECEIVE 文を実行しています。</li> <li>ステータスコード 71000 が返されたあとで、再び非同期型のメッセージを受信する RECEIVE 文を実行しています。</li> </ul> <p>&lt; SPP の実行でリターンした場合 &gt;            SPP では非同期型のメッセージを受信する RECEIVE 文を実行できません。</p>
72001	<p>SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。</p> <p>RECEIVE 文を実行できない論理端末を設定しています。</p>
72013	一意名 1 の L の指定値を超えるセグメントを受信しました。一意名 1 の L の指定値を超えた部分は切り捨てられました。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72036	一意名 1 の L の指定値が不足しています。5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

## SEND - メッセージの送信 (データ操作言語)

---

形式 1 (セグメントの内容を設定して送信する場合)

DATA DIVISION (通信記述項) の指定

CD 通信記述名

```
FOR {OUTPUT | I-O}  
[STATUS KEY IS データ名1]  
[SYMBOLIC TERMINAL IS データ名2]  
[SYNCHRONOUS MODE IS {SYNC | ASYNC | データ名6}]  
[SWITCHING MODE IS {NORMAL | PRIOR | データ名7}]  
[DETAIL MODE IS データ名10]  
[WAITING TIME IS データ名11].
```

PROCEDURE DIVISION (通信文) の指定

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

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

DATA DIVISION (通信記述項) の指定

CD 通信記述名

```
FOR {OUTPUT | I-O}  
[STATUS KEY IS データ名1]  
[SYMBOLIC TERMINAL IS データ名2]  
[SWITCHING MODE IS {NORMAL | PRIOR | データ名7}].
```

PROCEDURE DIVISION (通信文) の指定

```
SEND 通信記述名 WITH EMI.
```

### 機能

次に示す CALL インタフェースの機能を実現します。

- 一方送信メッセージの送信 CBLDCMCF('SEND           ')
- 同期型のメッセージの送受信 CBLDCMCF('SENDRECV')
- 応答メッセージの送信 CBLDCMCF('REPLY           ')

同期型のメッセージの送受信では、単一セグメントの論理メッセージだけを扱えます。

## 通信記述項に設定する項目

FOR 句

次のどちらかの値を設定します。

OUTPUT

一方送信メッセージの送信

I-O

応答メッセージの送信, または同期型のメッセージの送受信

STATUS KEY 句

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

SYMBOLIC TERMINAL 句

論理端末名称を設定したデータ項目を設定します。

SYNCHRONOUS MODE 句

メッセージ送信が, 同期型か非同期型かを設定します。

SYNC

同期型のメッセージ送受信

同期型のメッセージ送受信の場合に設定します。

ASYNCR

非同期型のメッセージ送信

一方送信メッセージ送信の場合に設定します。

データ名 6

次の値を設定したデータ項目

'0'...非同期型のメッセージ送信

'1'...同期型のメッセージ送受信

指定を省略した場合は, 非同期型のメッセージ受信 (ASYNCR) が仮定されます。

SWITCHING MODE 句

一方送信メッセージの場合に, 一般か優先かを指定します。

NORMAL

一般の一方送信メッセージ

PRIOR

優先の一方送信メッセージ

データ名 7

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

次の値を設定したデータ項目  
'0'または' '...一般の一方送信メッセージ  
'1'...優先の一方送信メッセージ

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

DETAIL MODE 句

一方送信メッセージの場合に、出力通番を付けるかどうかを指定します。

データ名 10

次の値を設定したデータ項目  
'0'または' '...出力通番を付けます。  
'1'...出力通番を付けません。

省略した場合は、出力通番を付けません。

WAITING TIME 句

同期型のメッセージ送受信の場合に、監視時間の値を指定します。

データ名 11

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

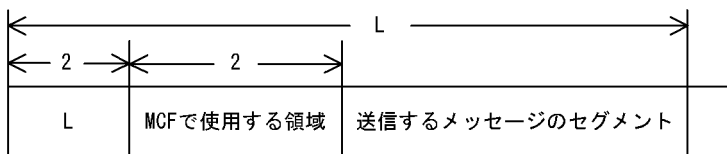
省略した場合、または'00000000'を設定した場合は、MCF マネジャ定義の UAP 共通定義 (mcfmuap -t) で指定した同期型送受信監視時間が設定されます。

通信文に指定する項目

一意名 1

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

(単位: バイト)



WITH 句

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

ESI

先頭セグメントまたは中間セグメントの送信

EMI

最終セグメントまたは単一セグメントの送信

一意名 2

次の値を設定したデータ項目

'1'...ESI (先頭セグメントまたは中間セグメントの送信)

'2'...EMI (最終セグメントまたは単一セグメントの送信)

省略した場合は、EMI (最終セグメントまたは単一セグメントの送信) が設定されます。

なお、同期型のメッセージの送受信 (SENDRECV) の場合は、EMI を指定するか、または指定を省略してください。

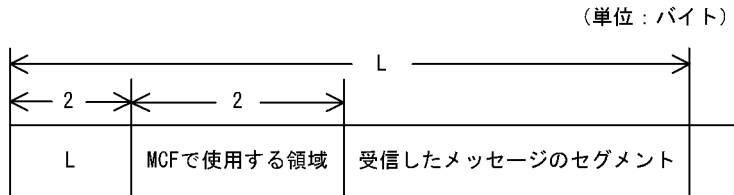
BEFORE 句

同期型のメッセージ送受信の場合に設定します。メッセージ送信後にその端末からメッセージを受信する場合に指定します。

一意名 3

受信メッセージを格納するデータ項目

同期型のメッセージの送受信の場合に、受信するセグメントの領域 (一意名 3) の形式を次に示します。



ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	一意名 1 の L に 32000 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。

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

ステータスコード	意味
	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 RECEIVE 文を実行する前に、SEND 文を実行しています。</li> <li>非応答型の MHP から応答メッセージを送信しています。</li> </ul> <p>&lt; SPP の実行でリターンした場合 &gt;</p> <ul style="list-style-type: none"> <li>トランザクションでない SPP の処理から、SEND 文を実行しています。</li> <li>SPP では、応答メッセージを送信する SEND 文を実行できません。</li> </ul>
72001	<p>SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。</p> <p>SEND 文を実行できない論理端末を設定しています。</p>
72005	先頭セグメントまたは中間セグメント送信時、一意名 1 の L に 4 以下の値を設定しています。
72008	<p>応答メッセージの最終セグメントを送信する SEND 文を実行したあとに、再び応答メッセージを送信する SEND 文を実行しています。</p> <p>SEND 文を実行して応答型のアプリケーションを起動させたあとに、応答メッセージを送信する SEND 文を実行しています。</p>
72013	<p>一意名 3 の L の指定値を超えるセグメントを受信しました。一意名 3 の L の指定値を超えた部分は切り捨てられました。</p> <p>32763 バイトを超えるセグメントを受信しました。32768 バイトを超えた部分は切り捨てられました（バッファ形式 2 の場合）</p>
72017	DETAIL MODE 句に設定した値が間違っています。
72018	SWITCHING MODE 句に設定した値が間違っています。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72026	WITH 句に設定した値が間違っています。
72036	一意名 3 の L の指定値が不足しています。5 バイト以上の領域を確保してください。
72037	BEFORE 句に設定した値が間違っています。
72041	単一セグメント送信時、一意名 1 の L に 4 以下の値を設定しています。
73001	メッセージ出力先の論理端末で障害が発生しました。
73002	MCF 通信サービスで障害が発生しました。
73005	SEND 文（同期型）を呼び出してから、設定した限界経過時間が過ぎても、論理端末からの応答がありません。
73008	論理端末が閉塞中、または MCF が終了処理中に、UAP から SEND 文が呼び出されました。
73010	<p>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。</p> <p>メッセージ編集エラーが発生しました。</p>
73015	メッセージ出力先の論理端末は、ほかの UAP で仕掛り中です。



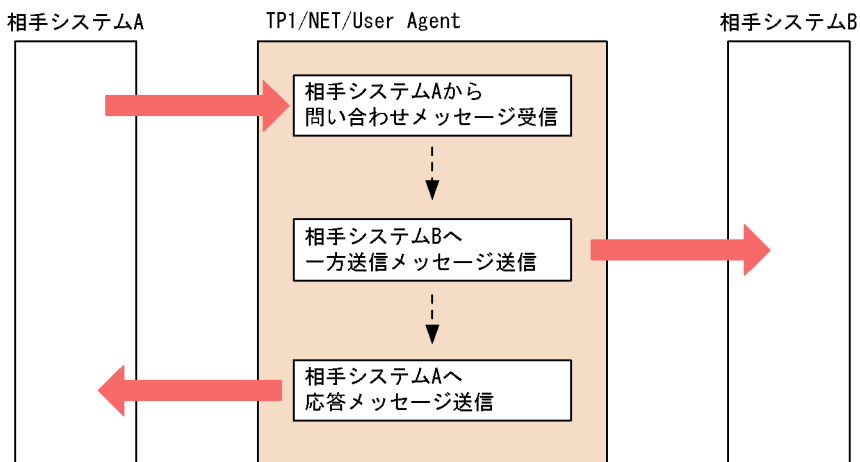
3. メッセージ送受信インタフェース  
SEND - メッセージの送信 (データ操作言語)

ステータスコード	意味
73018	SEND 文 (同期型) を呼び出した場合に、データ名 11 に設定した監視時間が間違っています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

## ユーザアプリケーションプログラム作成例

UAP の作成例として、処理の流れを次の図に示し、その流れに沿ったコーディング例を C 言語 (K&R 版) および COBOL 言語で示します。

図 3-1 処理の流れ



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

C 言語 (K&R 版) のコーディング例

/BeTRAN/examples/mcf/UserAgent/aplib/c/ap.c

COBOL 言語のコーディング例

/BeTRAN/examples/mcf/UserAgent/aplib/cobol/ap.cbl

C 言語 (K&R 版)

```
#include <dcmcf.h>
ex_uapl ()
{
    char termnam[9] ;
    char recvdata[2048] ;
    DCLONG rdataleng ;
    DCLONG time ;
    DCLONG inbufleng=2048 ;
    char senddata[512] ;
    static char resv01[9]="¥0" ;
    static char resv02[9]="¥0" ;
    static char resv03[9]="¥0" ;
    char *workadd = recvdata ;
    /* 問い合わせメッセージの受信 */
    dc_mcf_receive(DCMCFRST,DCNOFLAGS,termnam,resv01,
        workadd,&rdataleng,2048,&time) ;
```

```

workadd = workadd + rdataleng + 8 ;
inbufleng = 2048 - (workadd - recvdata) ;
while(dc_mcf_receive(DCMCFSEG,DCNOFLAGS,termnam,resv01,
                    workadd,&rdataleng,inbufleng,&time)
      == DCMCFRTN_00000)
  {
    workadd = workadd + rdataleng + 8 ;
    inbufleng = inbufleng - (workadd - recvdata) ;
  }

/*          データの処理          */

dc_mcf_send(DCMCFEMI,DCMCFOUT,"HOSTB",
            resv01,senddata,504,resv02,DCNOFLAGS) ;
/*          一方送信メッセージの送信 */
dc_mcf_reply(DCMCFEMI,DCNOFLAGS,resv03,
            resv01,senddata,504,resv02,DCNOFLAGS) ;
/*          応答メッセージの送信 */
}

```

## COBOL 言語

```

IDENTIFICATION DIVISION.
PROGRAM-ID. EXUAP1.

ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01  RCV.
    02  MSG-REC          PIC X(8) VALUE 'RECEIVE '.
    02  STATUS-CODE1    PIC X(5) .
    02  FILLER          PIC X(3) .
    02  SEG-CODE        PIC X(4) VALUE 'FRST'.
    02  RTN-CODE        PIC X(4) VALUE SPACE.
    02  DAY-ID          PIC 9(8) .
    02  TIME-ID         PIC 9(8) .
    02  SEG-LENG        PIC 9(9) COMP VALUE 2048.
    02  MCFUSE1         PIC X(4) VALUE SPACE.
    02  MCFUSE2         PIC X(4) VALUE SPACE.
    02  MCFUSE3         PIC X(4) VALUE SPACE.
    02  MCFUSE4         PIC X(4) VALUE SPACE.
    02  MCFUSE5         PIC X(8) VALUE SPACE.
    02  MCFUSE6         PIC X(4) VALUE SPACE.
    02  MCFUSE7         PIC X(8) VALUE SPACE.
    02  MCFUSE8         PIC X(4) VALUE SPACE.
    02  MCFUSE9         PIC 9(9) COMP VALUE ZERO.
    02  MCFUSE10        PIC 9(9) COMP VALUE ZERO.
    02  MCFUSE11        PIC X(1) VALUE SPACE.
    02  MCFUSE12        PIC X(1) VALUE SPACE.
    02  MCFUSE13        PIC X(14) VALUE LOW-VALUE.
01  CD1.
    02  SEG-CODE1        PIC X(4) VALUE SPACE.
    02  TERM-CODE        PIC X(8) .
    02  MCFUSE14         PIC X(8) VALUE SPACE.
    02  MCFUSE15         PIC X(8) VALUE SPACE.
    02  MCFUSE16         PIC X(28) VALUE LOW-VALUE.

```

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

```

01  DATA1.
02  MSGSEG-LENG1  PIC 9(9) COMP.
02  MCFUSE17      PIC X(4).
02  MCFUSE18      PIC X(2).
02  MCFUSE19      PIC X(1).
02  MCFUSE20      PIC X(1).
02  REC-MSGSEG1   PIC X(2048).
01  SND.
02  MSG-SEND      PIC X(8) VALUE 'SEND  '.
02  STATUS-CODE2  PIC X(5).
02  FILLER        PIC X(3).
02  MCFUSE21      PIC X(4) VALUE SPACE.
02  MCFUSE22      PIC X(4) VALUE SPACE.
02  MCFUSE23      PIC 9(8).
02  MCFUSE24      PIC 9(8).
02  MCFUSE25      PIC 9(9) COMP VALUE ZERO.
02  SEND-SEG      PIC X(4) VALUE 'EMI '.
02  SEND-SYNC     PIC X(4) VALUE 'ASYN'.
02  SEND-NORM     PIC X(4) VALUE 'NORM'.
02  SEND-SEQ      PIC X(4) VALUE 'NSEQ'.
02  MCFUSE26      PIC X(8) VALUE SPACE.
02  SEND-CODE     PIC X(4) VALUE SPACE.
02  MCFUSE27      PIC X(8) VALUE SPACE.
02  MCFUSE28      PIC X(4) VALUE SPACE.
02  MCFUSE29      PIC 9(9) COMP VALUE ZERO.
02  MCFUSE30      PIC 9(9) COMP.
02  MCFUSE31      PIC X(1) VALUE SPACE.
02  MCFUSE32      PIC X(1) VALUE SPACE.
02  MCFUSE33      PIC X(14) VALUE LOW-VALUE.
01  CD2.
02  SENDSEG-CODE2 PIC X(4) VALUE 'OUT '.
02  TERM-CODE2    PIC X(8) VALUE 'HOSTB'.
02  MCFUSE34      PIC X(8) VALUE SPACE.
02  MCFUSE35      PIC X(8) VALUE SPACE.
02  MCFUSE36      PIC X(28) VALUE LOW-VALUE.
01  DATA2.
02  MSGSEG-LENG2  PIC 9(9) COMP VALUE 512.
02  MCFUSE37      PIC X(8).
02  REC-MSGSEG2   PIC X(512).
01  RPLY.
02  MSG-REPLY     PIC X(8) VALUE 'REPLY  '.
02  STATUS-CODE3  PIC X(5).
02  FILLER        PIC X(3).
02  MCFUSE41      PIC X(4) VALUE SPACE.
02  MCFUSE42      PIC X(4) VALUE SPACE.
02  MCFUSE43      PIC 9(8).
02  MCFUSE44      PIC 9(8).
02  MCFUSE45      PIC 9(9) COMP VALUE ZERO.
02  SEND-SEG1     PIC X(4) VALUE 'EMI '.
02  SEND-SYNC1    PIC X(4) VALUE 'ASYN'.
02  MCFUSE46      PIC X(4) VALUE SPACE.
02  SEND-SEQ1     PIC X(4) VALUE SPACE.
02  MCFUSE47      PIC X(8) VALUE SPACE.
02  SEND-CODE1    PIC X(4) VALUE SPACE.
02  MCFUSE48      PIC X(8) VALUE SPACE.
02  MCFUSE49      PIC X(4) VALUE SPACE.
02  MCFUSE50      PIC 9(9) COMP VALUE ZERO.

```

```
02 MCFUSE51 PIC 9(9) COMP.
02 MCFUSE52 PIC X(1) VALUE SPACE.
02 MCFUSE53 PIC X(1) VALUE SPACE.
02 MCFUSE54 PIC X(14) VALUE LOW-VALUE.
01 CD3.
02 SEND-CODE3 PIC X(4) VALUE SPACE.
02 TERM-CODE3 PIC X(8) VALUE SPACE.
02 MCFUSE55 PIC X(8) VALUE SPACE.
02 MCFUSE56 PIC X(8) VALUE SPACE.
02 MCFUSE57 PIC X(28) VALUE LOW-VALUE.
```

```
PROCEDURE DIVISION.
CALL 'CBLDCMCF'USING RCV CD1 DATA1.
*(問い合わせメッセージの受信)
*
*   処理 1
*
CALL 'CBLDCMCF'USING SND CD2 DATA2.
*(一方送信メッセージの送信)
*
*   処理 2
*
CALL 'CBLDCMCF'USING RPLY CD3 DATA2.
*(応答メッセージの送信)
*
*   処理 3
*
EXIT PROGRAM.
```



# 4

## ユーザOWNコーディング， MCF イベントインタフェース

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

---

4.1 ユーザOWNコーディングインタフェース

---

4.2 MCF イベントインタフェース

---

## 4.1 ユーザOWNコーディングインタフェース

---

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

TP1/NET/User Agent で使用できる UOC を次に示します。

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

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

### 4.1.1 入力メッセージの編集とアプリケーション名の決定

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

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

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

#### (1) 入力メッセージの編集

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

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

#### (2) アプリケーション名の決定

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

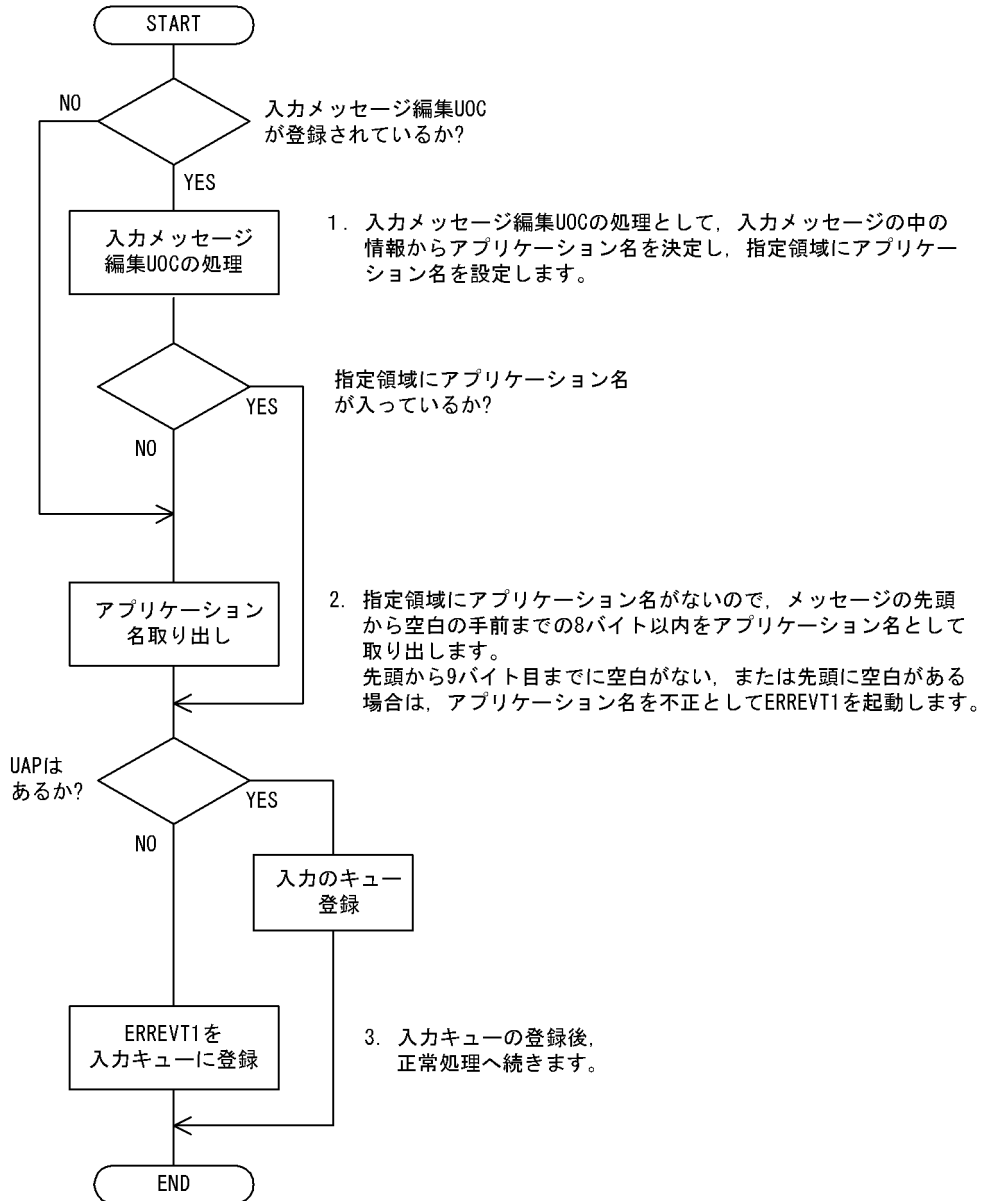
UOC でアプリケーション名を決定する場合, アプリケーション名の形式は, アプリケー



ション名格納領域の先頭から, '¥0' の手前までの 1 ~ 8 バイトの英数字です。先頭から 9 バイト目までに '¥0' がないときは, アプリケーション名を不正とし, ERREVT1 を起動します。

アプリケーション名の決定の処理を次の図に示します。

図 4-1 アプリケーション名の決定の処理



#### 4. ユーザOWNコーディング，MCF イベントインタフェース

##### (3) UOC エラーリターン処理

UOC から DCMCF\_UOC\_MSG\_NG でリターンした場合，MCF はメッセージログを出力し，相手システムに UERR を送信します。相手システムからの UERR に対する応答を受信したあとに，MCF は障害通知イベント (CERREVT) を起動します。

UOC で障害を検出し，エラー処理 UAP を起動したい場合は，ユーザ任意のエラー処理 UAP のアプリケーション名を設定します。このとき，MCF には DCMCF\_UOC\_MSG\_OK (\_RCV) でリターンします。

##### (4) UOC パラメタ不正の場合の処理

UOC で設定したパラメタに不正があった場合，MCF はメッセージログを出力し，相手システムに UERR を送信します。相手システムからの UERR に対する応答を受信したあとに，MCF は障害通知イベント (CERREVT) を起動します。

##### (5) OpenTP1 への組み込み方法

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

### 4.1.2 入力メッセージ編集 UOC インタフェース

入力メッセージ編集 UOC は，次に示す形式で呼び出します。

#### (1) 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

```
#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];         ...予備
    DCLONG rcv_prim;          ...受信サ - ビスプリミティブ
    dcmcf_uocbuff_list_n *buflist_adr;
                               ...受信バッファリストアドレス
    dcmcf_uocbuff_list_n *ebuflist_adr;
                               ...編集バッファリストアドレス
    char aplname[9];          ...アプリケーション名称
    char reserve2[7];         ...予備
    char *pro_indv_ifa;       ...MCF使用領域
    DCLONG rtn_detail;        ...詳細リターンコード
    char reserve3[8];         ...予備
} dcmcf_uoc_min_n;
```

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

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

#### (c) dcmcf\_uocbufinf\_n (バッファ情報) の内容

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

### (4) MCF が値を設定する項目

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

pro\_kind

プロトコル種別として, 次の値が設定されます。

DCMCF\_UOC\_PRO\_OSAS

OSAS/UA プロトコル

#### 4. ユーザOWNコーディング, MCF イベントインタフェース

le\_name

メッセージを入力した論理端末名称が設定されます。

rcv\_prim

受信サービスプリミティブとして, 次の値が設定されます。

DCMCF\_UOC\_RCV\_INQ

UINQ

DCMCF\_UOC\_RCV\_REP

UREP

DCMCF\_UOC\_RCV\_BRD

UBRD

DCMCF\_UOC\_RCV\_REP\_SR

sendrecv 時の UREP 受信

この場合, アプリケーション名は無効です。

buflist\_adr

受信用バッファリストのアドレスが設定されます。

ebuflist\_adr

編集用バッファリストのアドレスが設定されます。

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

aplname

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 -19000 ~ -19999 の範囲で指定してください。

### (b) dcmcf\_uocbuff\_list\_n (バッファリスト)

used\_buf\_num

使用したバッファ情報の数を設定します。

### (c) dcmcf\_uocbufinf\_n (バッファ情報)

seg\_size

バッファの使用長を設定します。

## (6) リターン値

uoc\_func() は次のコードでリターンしてください。

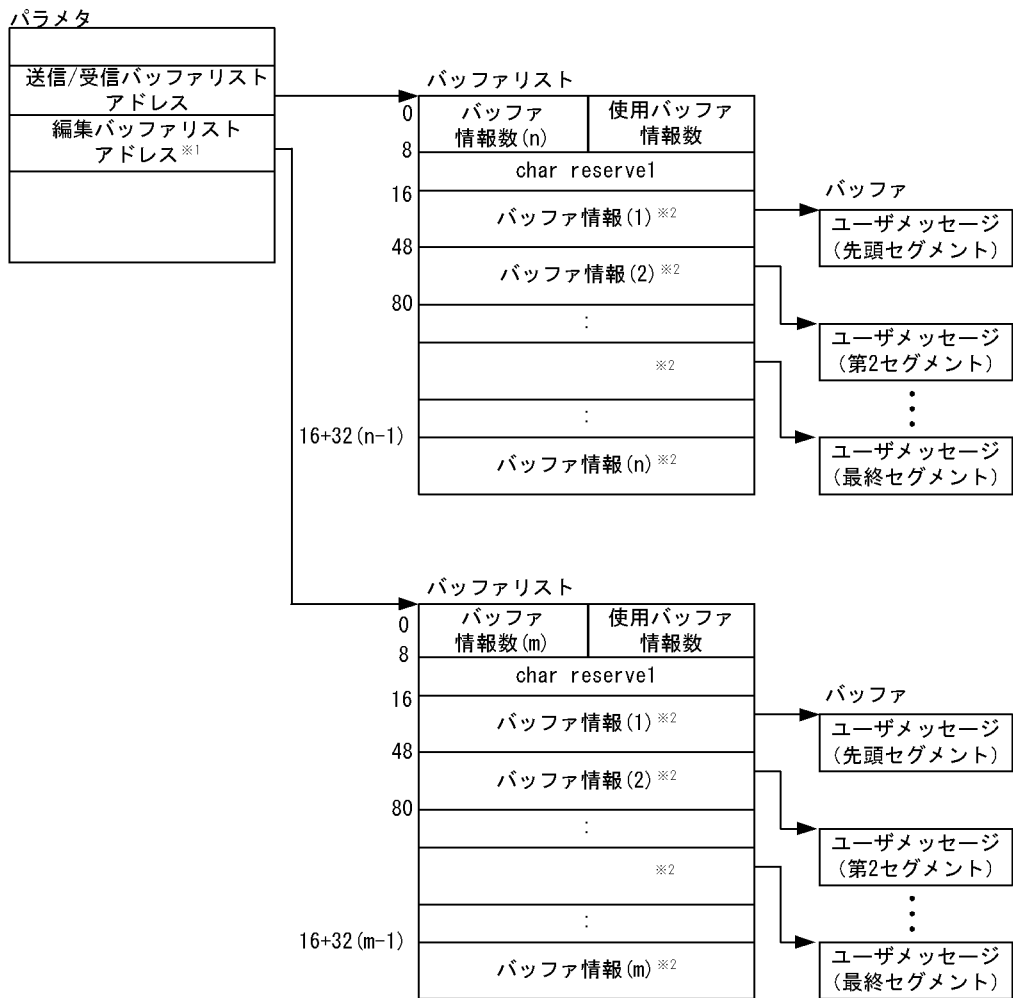
リターン値	意味
DCMCF_UOC_MSG_OK	正常リターン (編集バッファでスケジューリング)
DCMCF_UOC_MSG_OK_RCV	正常リターン (受信バッファでスケジューリング)
DCMCF_UOC_MSG_NG	メッセージ編集エラー

## (7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係を次の図に示します。

4. ユーザOWNコーディング, MCF イベントインタフェース

図 4-2 UOC インタフェース用のパラメタとバッファの関係



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

注 2  
バッファ情報は次の形式をしています。

0	バッファアドレス
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	
32	予備

### 4.1.3 出力メッセージの編集

出力メッセージの編集 UOC は、応答メッセージまたは一方送信メッセージの編集をする UOC です。出力メッセージの編集 UOC は、UAP が発行した送信メッセージを相手システムに実際に送信する前に処理するように位置させます。出力キューから全セグメントを読み出すと起動します。

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

#### (1) 出力メッセージの編集

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

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

#### (2) UOC エラーリターン処理

UOC から DCMCF\_UOC\_MSG\_NG でリターンした場合、MCF は次に示す処理をします。

- reply 型論理端末の場合  
メッセージログを出力し、相手システムに UERR を送信します。相手システムからの UERR に対する応答を受信したあとに、MCF は障害通知イベント (CERREVT) を起動します。
- request 型論理端末または send 型論理端末の場合  
メッセージログを出力し、該当するメッセージを破棄します。

### (3) UOC パラメタ不正の場合の処理

UOC で設定した値に不正があった場合, MCF は次に示す処理をします。

- reply 型論理端末の場合  
メッセージログを出力し, 相手システムに UERR を送信します。相手システムからの UERR に対する応答を受信したあとに, MCF は障害通知イベント (CERREVT) を起動します。
- request 型論理端末または send 型論理端末の場合  
メッセージログを出力し, 該当するメッセージを破棄します。

### (4) OpenTP1 への組み込み方法

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

## 4.1.4 出力メッセージ編集 UOC インタフェース

出力メッセージ編集 UOC は, 次に示す形式で呼び出します。

### (1) 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

```
#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];         ...予備
    dcmcf_uocbuff_list_n *buflist_adr;
                                ...送信バッファリストアドレス
};
```



```

dcmcf_uocbuff_list_n *ebuflist_adr;
                                ...編集バッファリストアドレス
DCLONG output_no;                ...メッセージ出力通番
char msg_type;                    ...メッセージ種別
char outputno_flag;              ...メッセージ出力通番有効フラグ
char resend_flag;                ...再送フラグ
char reserve2[1];                ...予備
char *pro_indv_ifa;              ...MCF使用領域
DCLONG rtn_detail;               ...詳細リターンコード
char reserve3[20];               ...予備
} dcmcf_uoc_mout_n;

```

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

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

#### (4) MCF が値を設定する項目

(a) dcmcf\_uoc\_mout\_n

pro\_kind

プロトコル種別として, 次の値が設定されます。

DCMCF\_UOC\_PRO\_OSAS

OSAS/UA プロトコル

le\_name

メッセージを入力した論理端末名称が設定されます。

buflist\_adr

送信用バッファリストのアドレスが設定されます。

ebuflist\_adr

編集用バッファリストのアドレスが設定されます。

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

output\_no

メッセージ出力通番が設定されます。ただし, outputno\_flag が DCMCF\_UOC\_OUTPUTNO\_OK のときだけ有効です。

msg\_type

メッセージ種別として, 次の値が設定されます。

'o'

問い合わせ応答メッセージ

'n'

一般一方送信メッセージ

'p'

#### 4. ユーザOWNコーディング, MCF イベントインタフェース

優先一方送信メッセージ

's'

同期送信メッセージ

outputno\_flag

メッセージ出力通番の有効フラグとして, 次のどちらかの値が設定されます。

DCMCF\_UOC\_OUTPUTNO\_OK

メッセージ出力通番を有効にします。

DCMCF\_UOC\_OUTPUTNO\_NG

メッセージ出力通番を無効にします。

resend\_flag

再送メッセージフラグとして, 次のどちらかの値が設定されます。

'r'

再送メッセージです。

'n'

再送メッセージではありません。

pro\_indv\_ifa

MCF で使用するパラメタです。

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

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

#### (5) ユーザが値を設定する項目

(a) dcmcf\_uoc\_mout\_n

rtn\_detail

詳細リターンコードを設定します。

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

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

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

#### (6) リターン値

uoc\_func() は次のコードでリターンしてください。

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

### (7) パラメタとバッファの関係

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

## 4.1.5 送信メッセージの通番編集

### (1) 通番編集

送信メッセージの通番編集 UOC では、受け取った通番を基に、ユーザ独自の処理ができます。例えば、通番を使用して編集した送信メッセージを再送要求します。

送信メッセージの通番編集 UOC を起動する場合、メッセージ送信の関数で、出力通番を付けるように設定してください。UOC は、UAP が send 関数または resend 関数を発行したときに、MCF によって起動されます。したがって、この UOC でメッセージを編集する場合は、先頭セグメントしか編集できません。

### (2) OpenTP1 への組み込み方法

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

#### (a) 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

#### (b) ユーザが値を設定する引数

flags

#### 4. ユーザOWNコーディング, MCF イベントインタフェース

DCMCF\_SEND\_UOC を指定します。

uoc\_addr

flags に対応する UOC のアドレスを指定します。

##### (c) リターン値

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

##### (d) メイン関数への登録例

- MHP の場合

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

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

- SPP の場合

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

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

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

### 4.1.6 送信メッセージの通番編集 UOC インタフェース

送信メッセージの通番編集 UOC は, 次を示す形式で, send\_uoc 関数として作成します。  
なお, UOC の関数名称はユーザの任意です。

## (1) 形式

### ANSI C, C++ の形式

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

### K&R 版 C の形式

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

## (2) MCF が値を設定する項目

### flags

送信メッセージの通番編集 UOC がいつ呼ばれたかが設定されます。

### DCMCF\_SEND\_DML

メッセージを送信する関数または命令文が呼び出されたときを意味します。

### DCMCF\_RESEND\_DML

メッセージを再送する関数または命令文が呼び出されたときを意味します。

### termname

送信先の論理端末名称が設定されます。

### sendno

送信メッセージの通番が設定されます。

### sendid

送信するメッセージ種別が設定されます。

### DCMCF\_SEND\_PRIO

優先の一方送信メッセージ

### DCMCF\_SEND\_NORM

一般の一方送信メッセージ

### dataleng

送信メッセージ長が設定されます。

### senddata

セグメントの内容が設定されます。

### (3) リターン値

リターン値	意味
DC_OK	正常に終了しました。

## 4.1.7 UOC 作成上の注意事項

UOC 作成上の注意事項を次に示します。

### (1) UOC の構造

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

### (2) UOC で使用できる関数

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

メモリ操作をする関数

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

時間取得関数

### (3) UOC の異常処理

TP1/NET/User Agent の UOC で異常を検知した場合、MCF の所定のリターンコードを使用して MCF に異常の発生を通知してください。UOC でプロセス終了となるシグナル、または abort() を呼び出すと、MCF が異常終了します。

### (4) UOC の実行タイミング

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

## 4.2 MCF イベントインタフェース

TP1/NET/User Agent でメッセージ送受信をすると, OpenTP1 の各種システム情報が MHP に通知されます。これを MCF イベントといいます。メッセージ送受信処理でエラーや障害が発生した場合, システム内で何が起きているのが MCF イベントの内容でわかります。MCF イベントに対応する MHP を MCF イベント処理用 MHP といいます。

MCF イベントは入力キューに渡されて, MCF イベント処理用 MHP で回復処理をします。なお, MCF イベントの発生時は入力メッセージ編集 UOC は呼び出しません。詳細についてはマニュアル「OpenTP1 プログラム作成の手引」を参照してください。

### 4.2.1 MCF イベントの種類

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

表 4-1 TP1/NET/User Agent が通知する MCF イベントの種類

MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP の処理の例
不正アプリケーション名検出通知イベント	ERREVT1	メッセージのアプリケーション名が MCF アプリケーション定義にありません。	該当するアプリケーション名がなかったことを報告します。問い合わせメッセージの場合は, 応答メッセージを出力できません。
メッセージ廃棄通知イベント	ERREVT2	次の理由で受信メッセージを破棄しました。 <ul style="list-style-type: none"> <li>• 入力キューに障害が発生しました。</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ました。</li> <li>• MHP のセグメント受信関数に, セグメントを渡す前に MHP の異常終了が発生しました。</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>• アプリケーションの即時起動時に障害が発生しました。</li> <li>• MHP のアプリケーション, サービスグループがセキュア状態です。</li> </ul>	メッセージを廃棄したことを報告します。問い合わせメッセージの場合は, 応答メッセージを出力できません。

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP の処理の例
UAP 異常終了通知イベント	ERREVT3	MHP のセグメント受信関数に、セグメントを渡したあとに MHP の異常終了が発生しました。	UAP 異常終了時の対処障害メッセージを送信します。 問い合わせメッセージの場合は、応答メッセージを出力できません。
タイマ起動メッセージ廃棄通知イベント	ERREVT4	アプリケーションのタイマ起動時に障害が発生しました。	メッセージを廃棄したことを報告します。
未処理送信メッセージ廃棄通知イベント	ERREVT4	次の理由で未処理送信メッセージを破棄しました。 <ul style="list-style-type: none"> <li>• MCF の正常終了処理時に、未処理送信メッセージの滞留時間監視の時間切れ（タイムアウト）が発生しました。</li> <li>• 運用コマンド（mcftdlqle）の入力によって、出力キューが削除されました。</li> </ul>	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。または，論理端末が閉塞解除しました。	コネクションが確立したことまたは論理端末が閉塞解除したことを報告します。
	CCLSEVT	コネクションが正常に解放されました。	コネクションが解放されたことを報告します。

#### 注

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

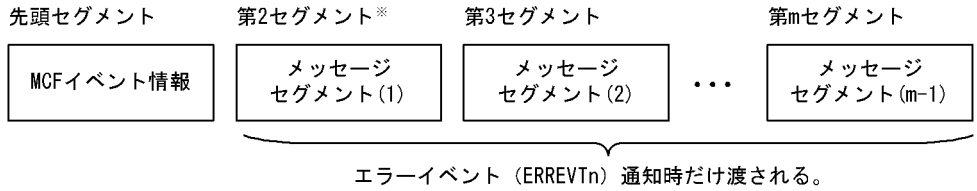
### 4.2.2 MCF イベント通知時のセグメント構成

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

MCF イベント通知時のセグメント構成を次の図に示します。



図 4-3 MCF イベント通知時のセグメント構成



注※ CERREVTの場合，利用者情報を設定します。

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

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

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

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

## 4.2.3 MCF イベント情報の形式 (C 言語)

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

エラーイベント (ERREVTn) で使用する構造体は，<dcmcf.h> で定義してあります。なお，<dcmoum\_statevt> は，<dcmoum.h> で定義されています。<dcmoum.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の場合)
                                   出力先論理端末名称
                                   (ERREVT4の場合)
    char cn_name[9];              ...コネクション名
    unsigned char format_kind;    ...MCF使用領域
    char reserve01;               ...予備
    DCLONG time;                  ...メッセージ入力時刻
}
```

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

```
};
```

##### (b) MCF イベントとして設定される項目

le\_name

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

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

- SPP がアプリケーション起動をした MHP で，障害が発生した場合
- 上記のエラーイベントとして起動された MHP が，さらにアプリケーション起動をした MHP で，障害が発生した場合

ERREVT4 では，メッセージを出力する論理端末名称が設定されます。

CERREVT では，論理端末障害が発生した論理端末名称が設定されます。コネクション障害時は無効です。

COPNEVT では，閉塞解除した論理端末名称が設定されます。コネクション確立時は無効です。

CCLSEVT では無効です。

cn\_name

コネクション名が設定されます。

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

- SPP がアプリケーション起動をした MHP で，障害が発生した場合
- 上記のエラーイベントとして起動された MHP が，さらにアプリケーション起動をした MHP で，障害が発生した場合

time

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

## (2) ERREVT1

### (a) 形式

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

## (b) MCF イベントとして設定される項目

ap\_name

次に示すどちらかが設定されます。

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

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

MHP 以外から送信された場合は, ヌル文字が設定されます。

## (3) ERREVT2

## (a) 形式

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

## (b) MCF イベントとして設定される項目

ap\_name

エラーになった UAP のアプリケーション名が設定されます。

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

MHP 以外から送信された場合は, ヌル文字が設定されます。

reason\_code

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

## (4) ERREVT3

## (a) 形式

```
struct dc_mcf_evt3_type {
    struct dc_mcf_evtheader  evtheader ;
                                ...MCFイベント共通ヘッダ
    char reserve01[12] ;        ...予備
    char map_name[10] ;         ...MCF使用領域
    char reserve03[2] ;         ...予備
    char ap_name[10] ;          ...アプリケーション名
                                (異常が発生したメッセージの
                                アプリケーション名)
    char reserve04[2] ;         ...予備
    char service_name[32] ;     ...サービス名
};
```

#### 4. ユーザOWNコーディング, MCF イベントインタフェース

```
char serv_grp_name[32] ;    ...サービスグループ名
char bid[36] ;            ...トランザクションブランチID領域
};
```

##### (b) MCF イベントとして設定される項目

###### ap\_name

異常が発生した MHP のアプリケーション名が設定されます。

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

MHP 以外から送信された場合は, ヌル文字が設定されます。

###### service\_name

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

###### serv\_grp\_name

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

###### bid

トランザクションのブランチ ID が設定されます。

#### (5) ERREVTA

##### (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 ;     ...MCF使用領域
    char rname[16] ;           ...MCF使用領域
    char reserve1[26] ;        ...予備
} dcmoum_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

(00000031)<sub>16</sub>

論理端末障害発生

err\_reason1, err\_reason2

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

## (7) COPNEVT, CCLSEVT

### (a) 形式

```
typedef struct {
    struct dc_mcf_evtheader header ;
                                ...MCFイベント共通ヘッダ
    DCLONG notice_fact ;       ...通知要因コード
    char reserve1[12] ;         ...予備
    char rname[16] ;           ...MCF使用領域
    char reserve2[26] ;        ...予備
} dcmoum_statevt ;
```

### (b) MCF イベントとして設定される項目

notice\_fact

COPNEVT の場合，通知された要因が次に示す値で設定されます。

(00000030)<sub>16</sub>

コネクション確立

(00000031)<sub>16</sub>

## 論理端末閉塞解除

## 4.2.4 MCF イベント情報の形式（COBOL 言語）

COBOL 言語の場合はセグメントの並びとして渡されます。COBOL 言語の UAP の場合，MCF イベント情報の内容を表 4-2 ～ 表 4-7 に示します。

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

項目	位置（バイト）		長さ （バイト）	属性	内容
	形式 1	形式 2			
予備（形式 1 のときだけ）	0	-	2	-	-
予備（形式 1 のときだけ）	2	-	2	-	-
エラーイベントコード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT1 を示す '1' が設定されます。
入力元論理端末名称	12	8	8	英数字	メッセージを入力した論理端末名称です。
予備	20	16	20	-	-
アプリケーション名	40	36	8	英数字	次に示すどれかが設定されます。 <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 進数字	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日

## 4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
メッセージが入力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
予備	112	108	16	-	-

(凡例)

- : 該当しません。または、使用されません。

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベントコード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT2 を示す '2' が設定されます。
入力元論理端末名称	12	8	8	英数字	メッセージを入力した論理端末名称です。次に示す場合には、'*' が設定されます。 <ul style="list-style-type: none"> <li>SPP がアプリケーション起動をした MHP で障害が発生した場合</li> <li>上記の障害発生時に MCF イベントとして起動された MHP によって、さらにアプリケーション起動された MHP で、障害が発生した場合</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	-	-

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
コネクション名	72	68	8	英数字	コネクション名です。次に示す場合には、'*' が設定されます。 <ul style="list-style-type: none"> <li>SPP がアプリケーション起動をした MHP で障害が発生した場合</li> <li>上記の障害発生時に MCF イベントとして起動された MHP によって、さらにアプリケーション起動された MHP で、障害が発生した場合</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 進数字	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
理由コード	112	108	4	外部 10 進数字	理由コードが設定されます。
予備	116	112	12	-	-

(凡例)

-：該当しません。または、使用されません。

注

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

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

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



## 4. ユーザOWNコーディング, MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
	10	6	2	英数字	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 がアプリケーション起動をした MHP で, 障害が発生した場合</li> <li>上記の障害発生時に MCF イベントとして起動された MHP が, さらにアプリケーション起動をした MHP で, 障害が発生した場合</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。
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 がアプリケーション起動をした MHP で, 障害が発生した場合</li> <li>上記の障害発生時に MCF イベントとして起動された MHP が, さらにアプリケーション起動をした MHP で, 障害が発生した場合</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 進数字	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY: 西暦の年 MM: 月 DD: 日
メッセージが入 力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH: 時 MM: 分 SS: 秒 00 は固定です。
予備	112	108	16	-	-
サービス名	128	124	31	英数字	異常が発生した UAP のアプリケーション名に対応するサービス名です。
予備	159	155	1	-	-

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
サービスグループ名	160	156	31	英数字	異常が発生した UAP のサービスグループ名です。
予備	191	187	1	-	-
トランザクション ID ( BID )	192	188	36	2 進数字	異常が発生したトランザクションの BID です。
予備	228	224	28	-	-

( 凡例 )

- : 該当しません。または，使用されません。

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベントコード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	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	-	-
予備	80	76	16	-	-

4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
メッセージが入力された日付	96	92	8	外部 10 進数字	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
予備	112	108	16	-	-
予備	128	124	31	-	-
予備	159	155	1	-	-
予備	160	156	31	-	-
予備	191	187	1	-	-
予備	192	188	36	-	-
予備	228	224	28	-	-

(凡例)

- : 該当しません。または、使用されません。

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

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	イベントコード「CERREVT」が設定されます。
入力元論理端末名称	8	8	英数字	障害の発生した論理端末名称が設定されます。コネクション障害時は無効です。
予備	16	8	-	-
入力元コネクション名	24	8	英数字	コネクション名が設定されます。
メッセージ入力日付	32	8	外部 10 進数字	CERREVT を入力した日付です。
メッセージ入力時刻	40	8	外部 10 進数字	CERREVT を入力した時刻です。
障害要因コード	48	4	2 進数字	障害要因が設定されます。 (00000030) <sub>16</sub> ：コネクション障害 (00000031) <sub>16</sub> ：論理端末障害

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)	長さ (バイト)	属性	内容
理由コード 1	52	4	2 進数字	理由コード 1 が設定されます。
理由コード 2	56	4	2 進数字	理由コード 2 が設定されます。
予備	60	48	-	-

(凡例)

- : 該当しません。または，使用されません。

注

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

注

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

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

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	イベントコード「COPNEVT」，または「CCLSEVT」が設定されます。
入力元論理端末名称	8	8	英数字	論理端末の閉塞解除を通知する COPNEVT の場合は，閉塞解除した論理端末名称が設定されます。 コネクション確立を通知する COPNEVT および CCLSEVT では無効です。
予備	16	8	-	-
入力元コネクション名	24	8	英数字	コネクション名が設定されます。
メッセージ入力日付	32	8	外部 10 進数字	COPNEVT，CCLSEVT を入力した日付です。
メッセージ入力時刻	40	8	外部 10 進数字	COPNEVT，CCLSEVT を入力した時刻です。
通知要因	48	4	2 進数字	COPNEVT の場合に，イベントが通知された要因が設定されます。 (00000030) <sub>16</sub> : コネクション確立 (00000031) <sub>16</sub> : 論理端末閉塞解除
予備	52	56	-	-

(凡例)

- : 該当しません。または，使用されません。

# 5

## システム定義

この章では，OSAS/UA 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/User Agent 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/User Agent の定義の概要

---

TP1/NET/User Agent 固有のシステム定義の種類

---

mcfmuap ( UAP 共通定義 )

---

mcftalccn ( コネクション定義の開始 )

---

mcftalced ( コネクション定義の終了 )

---

mcftalcle ( 論理端末定義 )

---

mcftalcua ( UA 定義 )

---

mcftcomn ( MCF 通信構成共通定義 )

---

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

---

MCF 定義オブジェクトの生成

---

自システムの通信管理プログラムと関連づける内容

---

相手システムの通信定義と関連づける内容

---

定義例

---

## TP1/NET/User Agent の定義の概要

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

表 5-1 MCF で使用する定義ファイル

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

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

- MCF マネージャ定義
- MCF 通信構成定義

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

#### TP1/NET/User Agent 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/User Agent 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ティリティの起動コマンドについて説明します。TP1/NET/User Agent を組み込む方法については、「7. 組み込み方法」を参照してください。

### 通信定義の内容の関連づけ

OSAS/UA 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/User Agent 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、自システムの通信管理プログラム（XNF/H、またはXNF/AS）と関連づける内容を示します。また、相手システム（TMS-4V/SP またはXDM/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/User Agent 固有のシステム定義の種類

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

表 5-2 TP1/NET/User Agent 固有の定義の一覧

定義名	コマンド	オプション・オペランド		定義内容	指定値 (( 値範囲 )) 《省略時解釈値》
MCF マネージャ定義	mcfmuap	-t	sndrevertim	同期送受信監視時間 (SENDRECV)	符号なし整数 (単位: 秒) ((0 ~ 65535)) 《0》
MCF 共通定義	mcftcomn	-x	termrls	MCF 終了時の接続の解放方法	nomal   error   《no》
MCF 通信構成定義	プロトコル固有定義 (接続定義の開始) 指定数: 1 ~ 512	-c	-	接続 ID	1 ~ 8 文字の識別子
		-p	-	プロトコルの種別	ua
		-n	-	自システムの PSAP アドレス	1 ~ 142 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	xnfh   xnfas
		-i	-	システム開始時および再開時に接続を自動的に確立するかどうか	auto   《manual》



定義名	コマンド	オプション・オペランド	定義内容	指定値 (( 値範囲 )) 《省略時解釈値》	
		-o	-	OSAS/UA プロトコル種別	old   new
		-u	-	UINT を重畳する MPDU	ht   ws
		-w	nomltim	プロトコル監視時間	符号なし整数 ( 単位 : 秒 ) ((0, 10 ~ 65535)) 《60》
			usertim	問い合わせ応答監視時間	符号なし整数 ( 単位 : 秒 ) ((0, 10 ~ 65535)) 《60》
		-b	bretry	コネクション確立時に障害が発生した場合にコネクション確立再試行をするかどうか	《yes》   no
			bretrycnt	コネクション確立再試行の回数	符号なし整数 ( 単位 : 回 ) ((0 ~ 65535)) 《0》
			bretryint	コネクション確立再試行の間隔	符号なし整数 ( 単位 : 秒 ) ((0 ~ 2550)) 《60》
		-k	-	全 UA の開局モード	《together》   each
		-d	-	端末識別子の文字数	符号なし整数 ((0, 5 ~ 125)) 《5》
		-y	-	制御 UA の端末識別子	5 ~ 125 文字の英数字記号または 10 ~ 250 けた ( 偶数けた ) の 16 進数字
		-q	-	通信相手システムの PSAP アドレス	1 ~ 186 けたの 16 進数字
		-z	dtea	現用相手 DTE アドレス	1 ~ 32 けたの 16 進数字
			slot	現用スロット番号	符号なし整数 ((0 ~ 65535))
			altn_dtea	交代用相手 DTE アドレス	1 ~ 32 けたの 16 進数字
			altn_slot	交代用スロット番号	符号なし整数 ( n : 1 ~ 7 を昇順に指定 ) ((0 ~ 65535))
		-l	-	TL クラス	0   《2》
		-f	leopnevt	論理端末の閉塞解除時に、状態通知イベント ( COPNEVT ) を起動するかどうか	use   《nose》
			negomsg	システム構成の縮退を通知するメッセージを出力するかどうか	use   《nose》
	mcfalcle ( 論理端末定義 )	-l	-	論理端末名称	1 ~ 8 文字の識別子

## 5. システム定義

### TP1/NET/User Agent 固有のシステム定義の種類

定義名	コマンド	オプション・オペランド	定義内容	指定値 (( 値範囲 )) 《省略時解釈値》	
	指定数：1 ～ 2048	-t	-	この論理端末のタイプ	reply   request   send   receive
		-m	mmsgent	メモリキュー内の出力メッセージの最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
			dmsgent	ディスクキュー内の出力メッセージの最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
		-k	quekind	出力キューの媒体の種類	《memory》   disk
			quegrpid	キューグループ ID	1 ~ 8 文字の識別子
		-o	aj	送信完了時の情報を取得するかどうかの指定	《yes》   no
mcftalcua (UA 定義)	-u	-	UA 番号	符号なし整数 ((1 ~ 2048))	
指定数：1 ～ 2048	-y	-	この UA の端末識別子	5 ~ 125 文字の英数字記号または 10 ~ 250 けた (偶数けた) の 16 進数字	
mcftalced (コネクション定義の終了) 指定数： mcftalccn と同数	-	-	コネクション定義の終了	-	

(凡例)

- : 該当する内容がないことを表します。

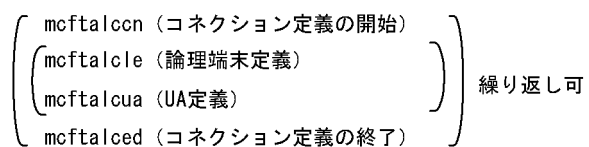
注

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

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

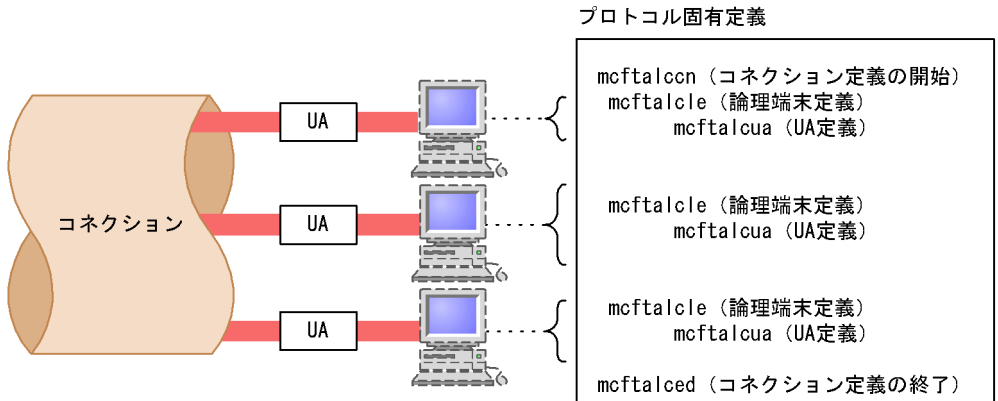


TP1/NET/User Agent のプロトコル固有定義の指定には次の条件があります。

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

コネクション 1 個に対し、論理端末を 3 個定義した場合のプロトコル固有定義の例を次の図に示します。

図 5-2 プロトコル固有定義の例



## mcfmuap (UAP 共通定義)

---

### 形式

```
mcfmuap      :  
             [-t "[sndrcvtim=同期送受信監視時間]"]  
             :
```

### 機能

同期型のメッセージの送受信の監視時間を定義します。

### 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)

sndrcvtim= 同期送受信監視時間 ~ 符号なし整数 ((0 ~ 65535)) 《0》(単位: 秒)

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

このオペランドは、mcftalccn 定義コマンドの -w オプションの usertim オペランドで指定する問い合わせ監視時間より、60 秒多い値を指定してください。

## mcftalccn ( コネクション定義の開始 )

### 形式

```
mcftalccn  -c コネクションID
           -p ua
           -n x' 自システムの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
              rcvbuf=メッセージ受信用バッファグループ番号"
           [-e " {msgbuf=メッセージ編集用バッファグループ番号}
              {count=メッセージ編集用バッファ数} " ]
           -m "mode=xnfh | xnfas"
           [-i auto | manual ]
           -o old | new
           -u ht | ws
           [-w " {nomltim=プロトコル監視時間}
              {usertim=問い合わせ監視時間} " ]
           [-b " {bretry=yes | no}
              {bretrycnt=コネクション確立時再試行回数}
              {bretryint=コネクション確立時再試行間隔} " ]
           [-k together | each]
           [-d 端末識別子の文字数]
           -y e' 制御UAの端末識別子' | x' 制御UAの端末識別子'
           -q x' 通信相手システムのPSAPアドレス'
           -z " {dtea=x' 現用相手DTEアドレス' }
              slot=現用スロット番号
              {altn_dtea=x' 交代用相手DTEアドレス' }
              {altn_slot=交代用スロット番号} "
           [-l 0 | 2]
           [-f " {leopnevt=use | nouse }
              {negomsg=use | nouse } " ]
```

### 機能

コネクションに関する環境を定義します。

### オプション

-c コネクション ID ~ 1 ~ 8 文字の識別子

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

-p ua

プロトコルの種別を指定します。

ua

OSAS/UA 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

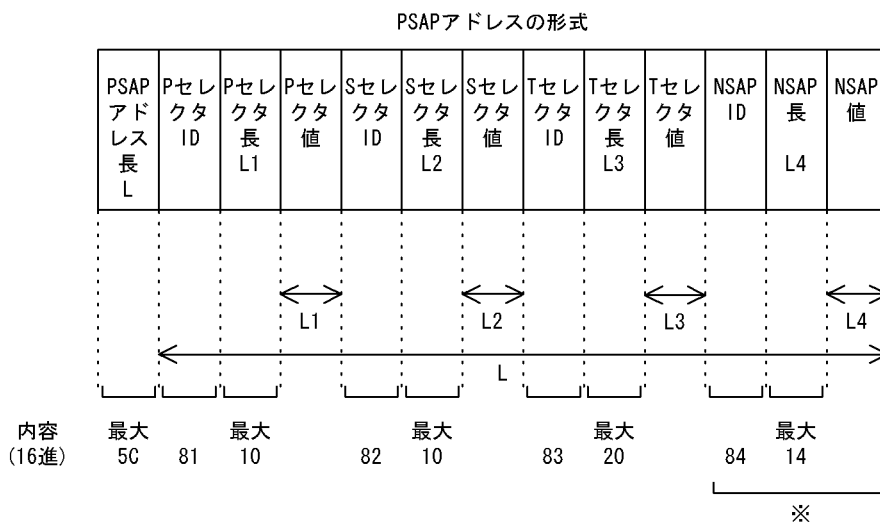
## 5. システム定義

mcftalccn (コネクション定義の開始)

ます。

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「システムセットアップガイド NSAP アドレス概説編」、「通信管理 XNF/H 構成定義編」、「通信管理 XNF/AS NSAP アドレス概説編」、または「通信管理 XNF/AS 構成定義編」を参照してください。

自システムの PSAP アドレスは、他の通信構成定義ファイルで指定する値と重複しないよう指定してください。



### 注

自システムの場合、NSAP アドレス (NSAP ID, NSAP 長, NSAP 値) は指定しません。

-g

(オペランド)

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

メッセージ送信用バッファのグループ番号を指定します。

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

-n オプションで指定する自システムの PSAP アドレスの値が、ほかの mcftalccn 定義コマンドの値と同じ場合、このオペランドもほかの mcftalccn 定義コマンドと同じ値を指定してください。

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

メッセージ受信用バッファのグループ番号を指定します。

mcftbuf コマンドの -g オプションの groupno オペランドに指定されているバッファ

グループ番号を指定してください。

-n オプションで指定する自システムの PSAP アドレスの値が、ほかの mcftalccn 定義コマンドの値と同じ場合、このオペランドもほかの mcftalccn 定義コマンドと同じ値を指定してください。

-e

( オペランド )

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

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

このオペランドを省略した場合は、メッセージ編集用バッファは確保されません。メッセージ編集用バッファグループ番号は、mcftbuf コマンドの -g オプションの groupno オペランドで指定するバッファグループ番号を指定してください。

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

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

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

-e オプションの msgbuf オペランドを省略した場合は、このオペランドの指定は無効です。

-m

( オペランド )

mode=xnfh | xnfas

使用する通信管理を指定します。

xnfh

XNF/H を使用します。

xnfas

XNF/AS を使用します。

このオペランドは必ず指定してください。

-i auto | manual ~ 《manual》

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

auto

## 5. システム定義

mcftalccn (コネクション定義の開始)

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

manual

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

-o old | new

OSAS/UA プロトコルの種別を指定します。XDM/DCCM3 のシステムと接続するときは new を指定します。

old

旧 OSAS/UA プロトコル

new

新 OSAS/UA プロトコル

-u ht | ws

初期設定メッセージ (UINT) を重畳する NIF/OSI 層の MPDU を指定します。

ht

UINT を Invoke (問い合わせ) MPDU に重畳します。

ws

UINT を Invoke (初期設定要求) MPDU および ACSE 層の初期設定要求に重畳します。

-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)

usertim オペランド以外の OSAS/UA プロトコルで規定する監視時間を指定します (初期設定監視時間 (T12), 個別開局監視時間 (T11), 後続送信指示監視時間 (T7), 後続送信直後の監視時間 (T9), 例外報告監視時間 (T11))。0 を指定した場合は、時間監視をしません。

usertim= 問い合わせ監視時間 ~ 符号なし整数 ((0, 10 ~ 65535)) 《60》(単位: 秒)

問い合わせメッセージ送信後、応答メッセージを受信するまでの監視時間を指定します。

0 を指定した場合は、時間監視をしません。また、このオペランドは、mcfmuap 定義コマンドの -t オプションの sndrcvtim オペランドで指定する同期送受信監視時間より少ない値を必ず指定してください。



-b

(オペランド)

bretry=yes | no ~ 《yes》

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

yes

コネクション確立再試行をします。

no

コネクション確立再試行をしません。

bretrycnt= コネクション確立時再試行回数 ~ 符号なし整数 ((0 ~ 65535)) 《0》

(単位：回)

コネクション確立時に障害が発生した場合、MCFが行う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= コネクション確立時再試行間隔 ~ 符号なし整数 ((0 ~ 2550)) 《60》(単位：秒)

(単位：秒)

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

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

-k together | each ~ 《together》

コネクション確立時、このコネクションのすべての UA の開局モードを指定します。

together

コネクション確立と同時にすべての UA を開局します (一括開局)。

each

すべての UA を閉局した状態でコネクションを確立します (個別開局)。

-d 端末識別子の文字数 ~ 符号なし整数 ((0, 5 ~ 125)) 《5》

mcfatalccn コマンドおよび mcfatalcua コマンドの -y オプションで指定する、端末識別子の文字数を指定します。XDM/DCCM3 のシステムと接続する場合 (mcfatalccn -o new 指

## 5. システム定義

mcftalccn (コネクション定義の開始)

定), 0 を指定してください。

-y e' 制御 UA の端末識別子' | x' 制御 UA の端末識別子'

該当する制御 UA の端末識別子を指定します。制御 UA とは、コネクションの確立と解放に使用する UA です。XDM/DCCM3 のシステムと接続する場合、このオプションを省略します。

この -y オプションで指定する端末識別子の文字数は -d オプションで指定した文字数と一致しなければなりません。

なお、端末識別子は、ほかの mcftalccn 定義コマンドで指定する端末識別子と重複して指定できません。

e ~ 5 ~ 125 文字の英数字記号

EBCDIC コードに変換することを意味します。

x ~ 10 ~ 250 けた (偶数けた) の 16 進数字

16 進数形式で指定することを意味します。

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

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

-n オプションで指定する自システムの PSAP アドレスと、-q オプションで指定する通信相手システムの PSAP アドレスの組み合わせは、ほかの mcftalccn 定義コマンドで指定する値と重複しないようにします。

PSAP アドレスの形式については、mcftalccn コマンドの -n オプションを参照してください。

-z

(オペランド)

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

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

slot= 現用スロット番号 ~ 符号なし整数 ((0 ~ 65535))

現用のスロット番号を指定します。

通信管理の定義で指定したスロット番号を指定してください。

altn\_dtea=x' 交代用 DTE アドレス' (n: 1 ~ 7) ~ 1 ~ 32 けたの 16 進数字

経路交代用の通信相手システムの DTE アドレスを指定してください。x は、16 進数字で指定することを意味します。

n は交代の順序であり、1 から昇順に指定する必要があります。

alt\_n\_slot= 交代用スロット番号 (n: 1 ~ 7) ~ 符号なし整数 ((0 ~ 65535))  
経路交代用のスロット番号を指定してください。

-l 0 | 2 ~ 《2》

使用する TL クラスを指定します。

0

TL クラス 0 を使用します。

2

TL クラス 2 を使用します。

RFC 1006 接続 (XNF/AS/OSI Extension または XNF/H/OSI Extension を使用) の場合は TL クラス 0, WAN 接続または LAN 接続の場合は TL クラス 2 を指定してください。

-f

(オペランド)

leopnevt=use | nouse ~ 《nouse》

論理端末が閉塞解除したときに、状態通知イベント (COPNEVT) を起動するかどうかを指定します。

このオプションの設定値にかかわらず、コネクション確立時の自動開局によって閉塞解除した場合は、COPNEVT は起動しません。

use

論理端末の閉塞解除時に COPNEVT を起動します。

nouse

論理端末の閉塞解除時に COPNEVT を起動しません。

negomsg=use | nouse ~ 《nouse》

相手システムの構成と自システムの構成が不一致 (相手システムの UA < 自システムの UA) であった場合に、相手システムの構成に合わせて自システムの構成を縮退することを通知するメッセージ (KFCA13257-W) を出力するかどうかを指定します。

use

自システムの構成縮退を通知するメッセージを出力します。

nouse

自システムの構成縮退を通知するメッセージを出力しません。

## 注意事項

-g オプションおよび -e オプションで指定するバッファグループ番号に対応するバッファグループ定義の mcfbuf コマンドでは、1 コネクション単位に次の表に示す資源が必要です。バッファグループ定義については、マニュアル「OpenTP1 システム定義」を参照

## 5. システム定義

mcftalccn (コネクション定義の開始)

してください。

バッファ種別	length オペランド	count オペランド
sndbuf	<ul style="list-style-type: none"> <li>最大セグメント長</li> <li>または</li> <li>( 端末識別子の文字数 + 14 ) × UA 数の最大値 以上</li> </ul>	( 最大セグメント 分割数 + 1 ) × UA 数 以上
revbuf	sndbuf と同じ	同上
msgbuf	最大セグメント長以上	同上

注

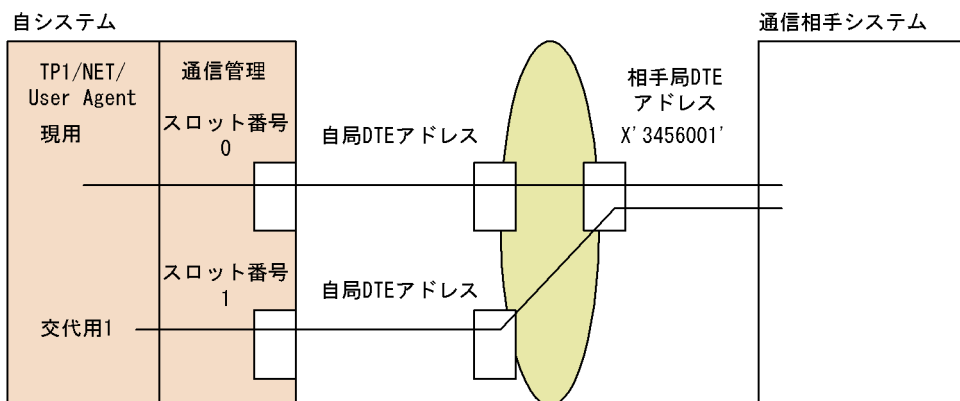
mcftalccn コマンドの -d オプションで指定します。

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、相手システムのバッファ資源として、次に示す長さで定義する場合があるので注意してください。

( 最大セグメント長 ) + ( プロトコルヘッダ長 )

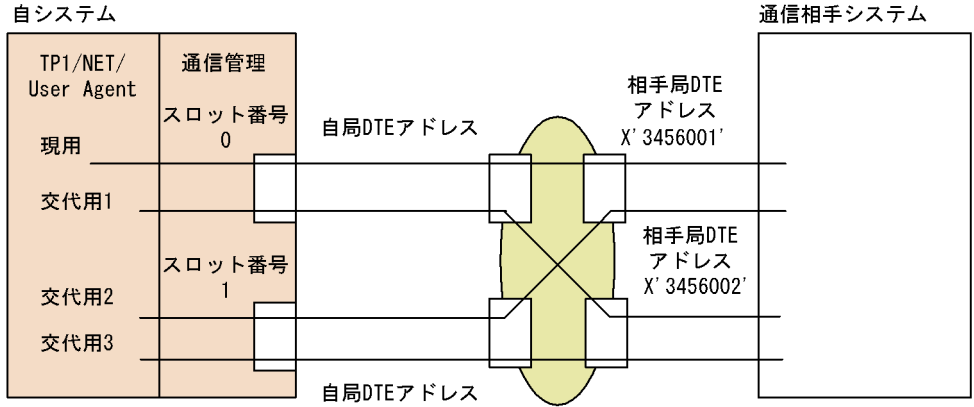
-z オプションで経路交代の定義をする場合、次のように指定します。

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

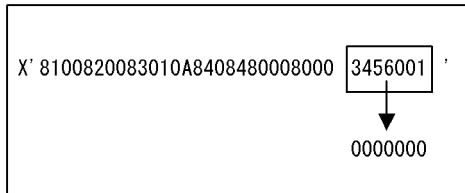


```
-z "dtea=x'3456001'  
slot=0  
alt1_dtea=x'3456002'  
alt1_slot=0  
alt2_dtea=x'3456001'  
alt2_slot=1  
alt3_dtea=x'3456002'"
```

alt3\_slot=1"



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



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

## 5. システム定義

mcftalced (コネクション定義の終了)

# mcftalced (コネクション定義の終了)

---

### 形式

mcftalced

### 機能

一つのコネクション定義の終了を示します。

### オプション

ありません。

## mcftalcle (論理端末定義)

---

### 形式

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

### 機能

論理端末に関する環境を定義します。

### オプション

-l 論理端末名称     ~   1 ~ 8 文字の識別子

論理端末名称を指定します。

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

-t reply | request | send | receive

この論理端末の端末タイプを指定します。

#### reply

応答型論理端末

#### request

問い合わせ型論理端末

#### send

送信型論理端末

#### receive

受信型論理端末

-m

### (オペランド)

mmsgcnt=メモリ出力メッセージ最大格納数     ~   符号なし整数 ((0 ~ 65535)) 《0》  
メモリキューで待ち合わせをする出力メッセージの最大格納数を指定します。  
出力メッセージの待ち合わせ数が指定した最大数になると、それ以後 UAP からの送信要求 (dc\_mcf\_send 関数または SEND 文) はエラーリターンとなります。

5. システム定義  
mcftalcle (論理端末定義)

0 を指定した場合、メモリキューで待ち合わせをする出力メッセージの数は無制限となります。ただし、実際に待ち合わせをできる出力メッセージ数は動的共用メモリの容量に依存します。

dmsgcnt= ディスク出力メッセージ最大格納数 ~ 符号なし整数 ((0 ~ 65535)) 《0》  
ディスクキューで待ち合わせをする出力メッセージの最大格納数を指定します。  
出力メッセージの待ち合わせ数が指定した最大数になると、それ以後 UAP からの送信要求 (dc\_mcf\_send 関数または SEND 文) はエラーリターンとなります。  
0 を指定した場合、ディスクキューで待ち合わせをする出力メッセージの数は無制限となります。ただし、実際に待ち合わせをできる出力メッセージ数はメッセージキューファイルの容量に依存します。

-k

(オペランド)

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

memory

メモリキューだけに割り当てます。

disk

ディスクキューおよびメモリキューに割り当てます。

disk を指定した場合、必ず quegrpID オペランドを指定してください。

quegrpID= キューグループ ID ~ 1 ~ 8 文字の識別子  
ディスクキューで待ち合わせをする出力メッセージに使用するキューグループ ID を指定します。MCF マネージャ定義の、mcfmqgid コマンドで指定するキューグループ ID (キューグループ種別は otq) のどれかを指定してください。  
この quegrpID オペランドは、quekind オペランドで disk を指定した場合だけ指定します。

-o

(オペランド)

aj=yes | no ~ 《yes》  
メッセージ送信が完了した場合に、メッセージ送信完了ジャーナル (AJ) を取得するかどうかを指定します。

yes

メッセージ送信完了ジャーナルを取得します。

no

メッセージ送信完了ジャーナルを取得しません。



## mcftalcua (UA 定義)

---

### 形式

```
mcftalcua -u UA番号  
          [-y e'UA端末識別子' | x'UA端末識別子']
```

### 機能

UA に関する環境を定義します。

### オプション

-u UA 番号     ~ 符号なし整数 ((1 ~ 2048))

UA 番号を指定します。

UA 番号には、一つのコネクション (mcftalcen 定義) に対して 1 から始まる昇順でかつ連続した番号を指定してください。

-y e'UA 端末識別子' | x'UA 端末識別子'

該当する UA の端末識別子を指定します。この場合の UA は、制御 UA に対する一般 UA で、一般 UA の開局、閉局、およびデータの送受信に使用します。

XDM/DCCM3 のシステムと接続する場合、このオプションを省略します。

この -y オプションで指定する端末識別子の文字数は、mcftalcen 定義コマンドの -d オプションで指定した文字数と一致しなければなりません。

なお、端末識別子は、ほかの mcftalcua 定義コマンドで指定する端末識別子と重複して指定できません。

e     ~ 5 ~ 125 文字の英数字記号)  
      EBCDIC コードに変換することを意味します。

x     ~ 10 ~ 250 けた (偶数けた) の 16 進数字  
      16 進数形式で指定することを意味します。

## mcftcomn (MCF 通信構成共通定義)

---

### 形式

```
mcftcomn      :
               [-x "termrls=nomal | error | no"]
```

### 機能

MCF 終了時の接続の解放方法を定義します。

### 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。

-x

(オペランド)

```
termrls=nomal | error | no    ~ 《no》
```

MCF 終了時 (正常終了, 計画停止 A, 計画停止 B) の接続の解放方法を指定します。

接続確立, 解放処理中に MCF 終了要求があった場合, 異常解放 (UERR 送信) します。ただし, 接続確立再試行をしている場合に, このオプションで no を指定していると, 再試行の処理を継続する場合があります。接続確立再試行を行う運用の場合には, このオプションで no 以外を指定してください。なお, このオプションは TP1/NET/User Agent 固有です。

nomal

正常解放 (UEND 送信)

error

異常解放 (UERR 送信)

no

解放処理はしません。MCF 通信プロセスの終了で解放します。

## システムサービス情報定義

---

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

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

システムサービス情報定義は、テキストエディタを使用して作成します。

システムサービス情報定義の完全パス名を次に示します。

```
$DCDIR/lib/sysconf/定義ファイル名
```

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

### 形式

```
set module="TP1/NET/User Agentの実行形式プログラム名"
```

### 機能

プロセスサービスが MCF 通信サービスを起動するための環境を定義します。

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

### オペランド

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

MCF 通信サービスを起動するための実行形式プログラム名を指定します。

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

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

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

## システムサービス共通情報定義

---

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

システムサービス共通情報定義の完全パス名を次に示します。

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

### 形式

set 形式

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

### 機能

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

### 説明

#### set 形式のオペランド

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

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

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

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

注 1

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

注 2

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

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

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

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

ファイル記述子の最大数を求める計算式を次に示します。

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

注 1

TP1/NET/User Agent の場合、コネクションの総数を 2 倍した値になります。

注 2

MCF 通信プロセスが扱う定義ファイルなどの数の最大値です。

自 OpenTP1 内の各 MCF 通信プロセスごとに計算し、その結果の中で最大値が 500 より大きい場合は、その値を指定します。500 以下の場合は、500 を指定します。

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

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

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

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

2048 - (同定義内の「max\_socket\_descriptors オペランドの指定値」)

## 注意事項

max\_socket\_descriptors オペランドの指定値と max\_open\_fds オペランドの指定値の合

## 5. システム定義

### システムサービス共通情報定義

計は、OSのシステムパラメタで指定する「1プロセスでオープンできるファイル数」を超えないようにする必要があります。システム定義の変更などによって、上記のオペランドの指定値の合計が増加する場合は、OSのシステムパラメタの指定を変更してください。

## 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

```
mcfosua -i {パス名} 入力ファイル名  
        -o {パス名} 出力オブジェクトファイル名
```

### 機能

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

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

TP1/NET/User Agent のプロトコル固有定義オブジェクトファイル以外の生成ユーティリティについては、マニュアル「OpenTP1 システム定義」を参照してください。

### オプション

-i {パス名} 入力ファイル名    ~    パス名    1 ~ 8 文字の識別子

定義ソースが格納されているファイル名を指定します。

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

定義オブジェクトを格納するファイル名を指定します。

## 自システムの通信管理プログラムと関連づける内容

---

TP1/NET/User Agent の定義には、通信管理と関連づける項目があります。

通信管理が XNF/H、または XNF/AS の場合 ( `mcftalccn -m "mode=xnfh"` ,  
"mode=xnfas" )

MCF 通信構成定義の `mcftalccn -z` で指定するスロット番号 ( 0 ~ 65535 ) と、通信管理  
の構成定義で指定したスロット番号を一致させます。

各通信管理の定義については、次に示すマニュアルを参照してください。

- XNF/H の場合  
マニュアル「通信管理 XNF/H 構成定義編」
- XNF/AS の場合  
マニュアル「通信管理 XNF/AS 構成定義編」

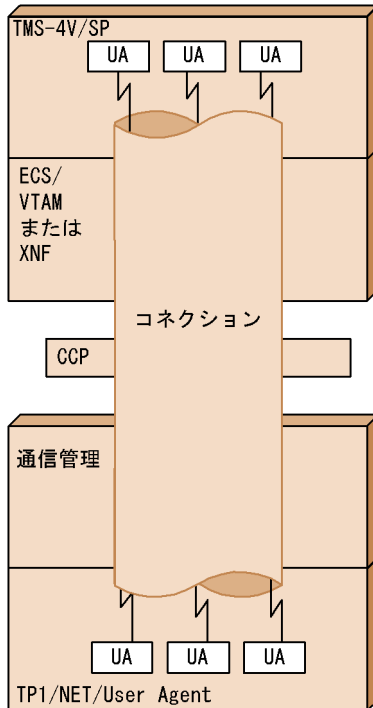


## 相手システムの通信定義と関連づける内容

### TMS-4V/SP のシステムとの接続

TMS-4V/SP のシステムと接続して TP1/NET/User Agent を使用する場合のネットワーク構成の例を次の図に示します。

図 5-3 ネットワーク構成の例 (TMS-4V/SP のシステムと接続する場合)



TP1/NET/User Agent は、図 5-3 の例のような場合、ECS/VTAM または XNF、および TMS-4V/SP と定義を関連づける必要があります。

通信相手システムが ECS/VTAM、または XNF と TMS-4V/SP の場合の TP1/NET/User Agent の定義と関連づける内容を示します。

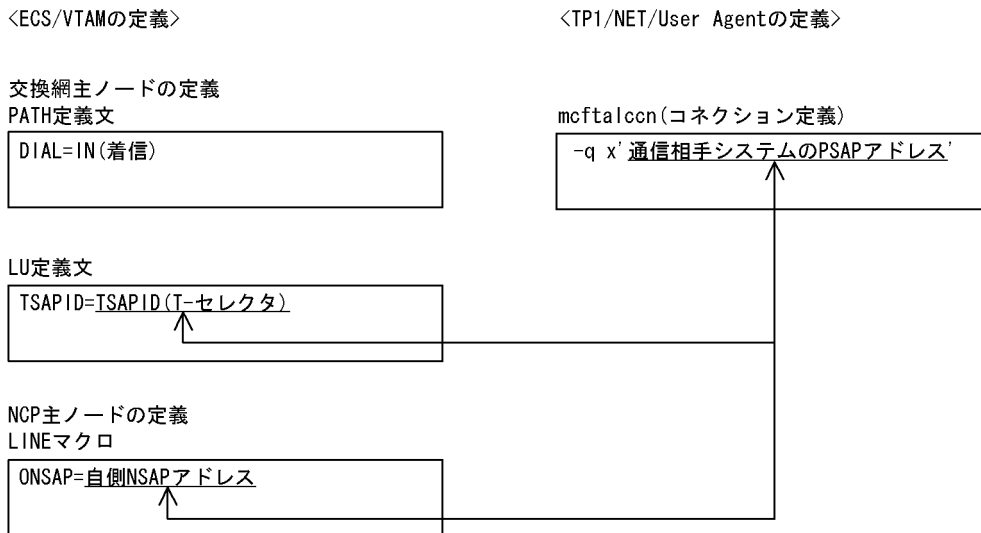
### ECS/VTAM ネットワーク定義と関連づける内容

TP1/NET/User Agent のシステム定義と ECS/VTAM ネットワーク定義との関係を次の図に示します。ECS/VTAM の定義については、マニュアル「VOS3 ECS/VTAM 使用の手引」を参照してください。

## 5. システム定義

相手システムの通信定義と関連づける内容

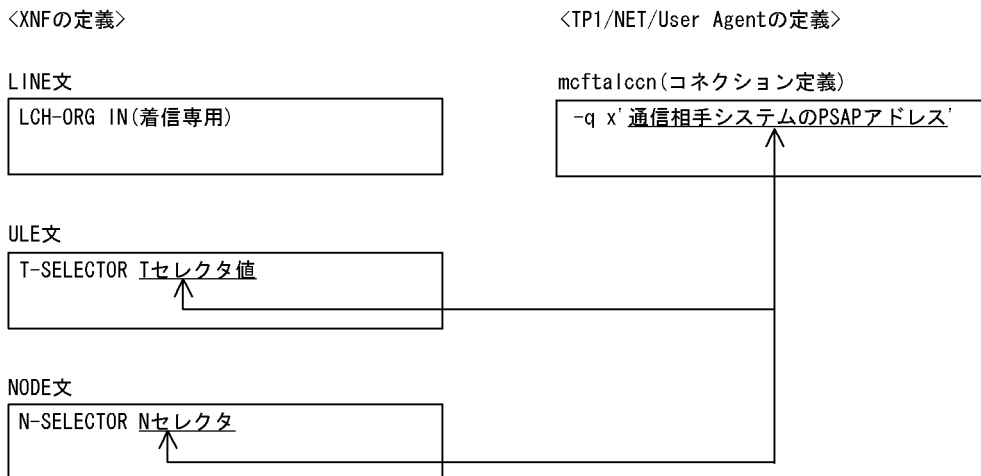
図 5-4 ECS/VTAM ネットワーク定義との関係



XNF ネットワーク定義と関連づける内容

TP1/NET/User Agent のシステム定義と XNF ネットワーク定義との関係を次の図に示します。XNF の定義については、マニュアル「VOS3 XNF ネットワーク定義」を参照してください。

図 5-5 XNF ネットワーク定義との関係



TMS-4V/SP システム定義と関連づける内容

TP1/NET/User Agent のシステム定義と TMS-4V/SP システム定義との関係を次の図に示します。TMS-4V/SP の定義については、マニュアル「VOS3 TMS-4V/SP OSAS/AP 間通信サービス-UA」を参照してください。

図 5-6 TMS-4V/SP システム定義との関係

<TMS-4V/SPの定義>

T4LD123-NODEマクロ  
 (ノード管理テーブルゼネレーション)

TLIDNO=端末識別子のバイト数  
 UAVRCN=0

TMCDUA-TERMINALマクロ

TLID=端末識別子  
 UATYP=UAM

TMCDUA-TERMINALマクロ

TLID=端末識別子  
 UATYP=UA

<TP1/NET/User Agentの定義>

mcftalccn(コネクション定義)

-o old  
 -d 端末識別子の文字数  
 -y e' 制御UAの端末識別子'

mcftalqua(UA定義)

-y e' UA端末識別子'

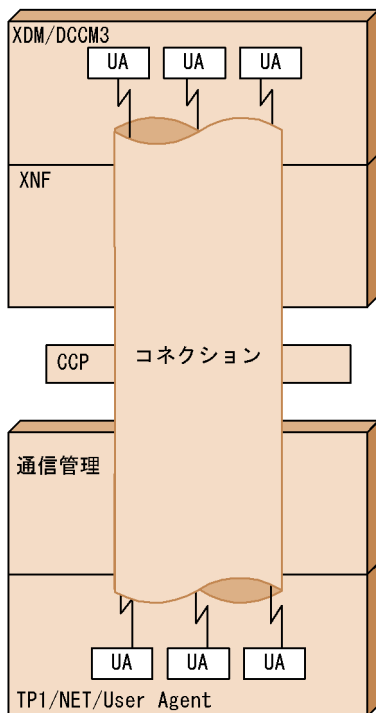
### XDM/DCCM3 のシステムとの接続

XDM/DCCM3 のシステムと接続して TP1/NET/User Agent 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。

## 5. システム定義

### 相手システムの通信定義と関連づける内容

図 5-7 ネットワーク構成の例（XDM/DCCM3 のシステムと接続する場合）



TP1/NET/User Agent は、図 5-7 の例のような場合、XNF および XDM/DCCM3 と定義を関連づける必要があります。

通信相手システムが XNF、および XDM/DCCM3 の場合の TP1/NET/User Agent の定義と関連づける内容を示します。

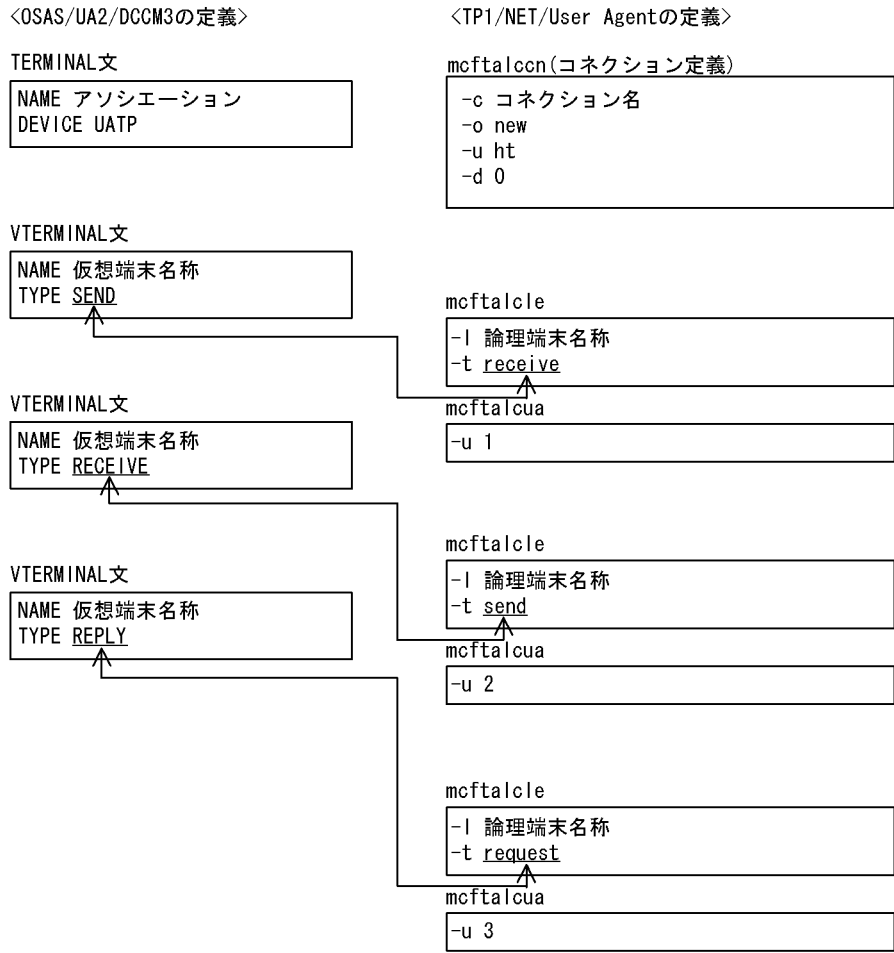
#### XNF ネットワーク定義と関連づける内容

TP1/NET/User Agent のシステム定義と XNF ネットワーク定義との関係については、図 5-5 を参照してください。

#### XDM/DCCM3 システム定義と関連づける内容

TP1/NET/User Agent のシステム定義と XDM/DCCM3 システム定義との関係を次の図に示します。XDM/DCCM3 の定義については、マニュアル「VOS3 OSAS/UA2/DCCM3」を参照してください。

図 5-8 XDM/DCCM3 システム定義との関係



## 定義例

---

ここでは、次に示す 2 種類の定義例を示します。

- 新 OSAS/UA プロトコルを使用する場合の定義例
- 旧 OSAS/UA プロトコルを使用する場合の定義例

### 新 OSAS/UA プロトコルを使用する場合の定義例

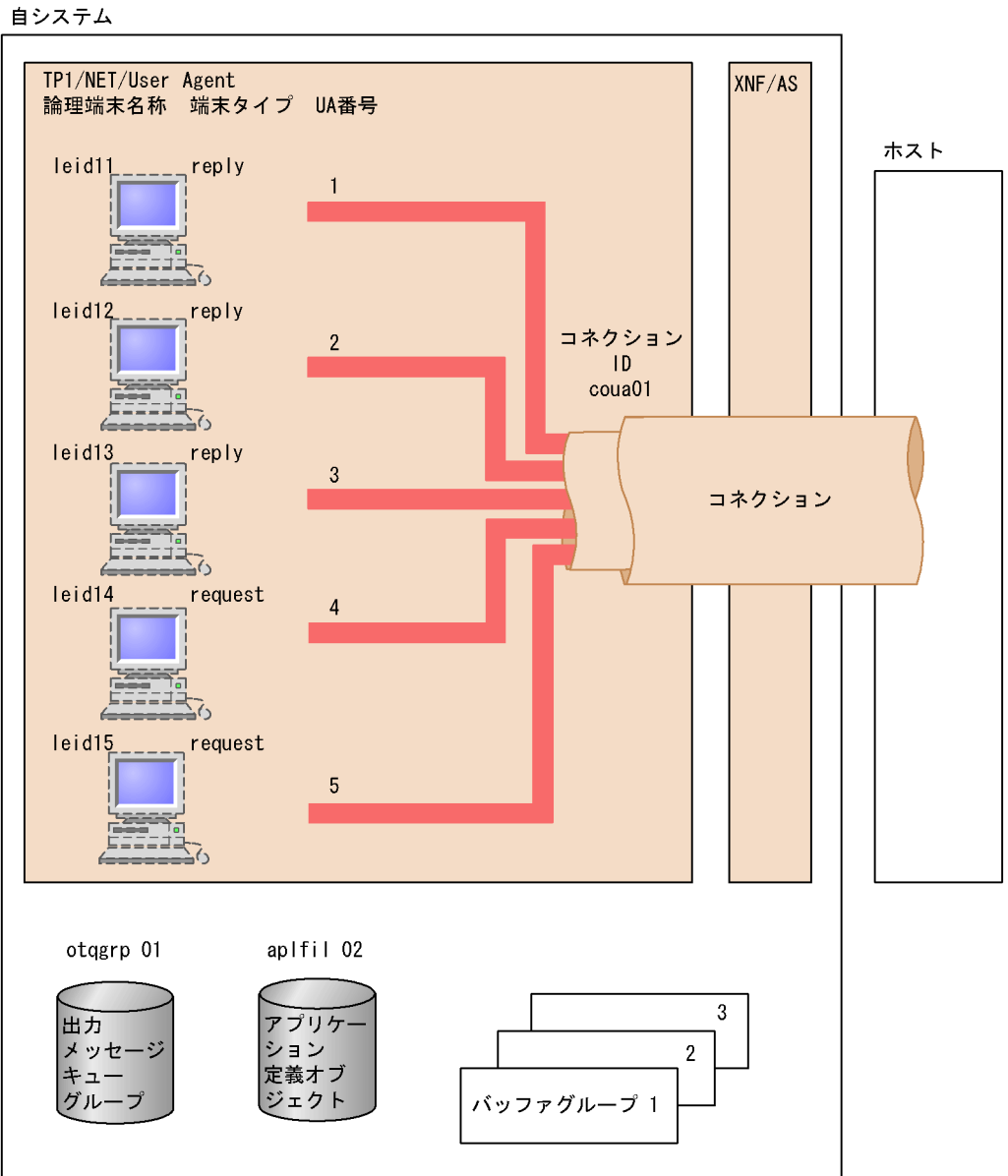
新 OSAS/UA プロトコルを使用した場合に、TP1/NET/User Agent を使用したシステム定義の例を示します。

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

- /BeTRAN/examples/mcf/UserAgent/conf/com\_c2
- /BeTRAN/examples/mcf/UserAgent/conf/com\_d2

図 5-9 TP1/NET/User Agent のシステム構成例 (新 OSAS/UA プロトコル使用時)



コーディング例 (新 OSAS/UA プロトコルで XNF/AS 使用時の TP1/NET/User Agent の共通定義)

```

1 #####
2 ### MCF通信構成定義 共通定義 #####
3 #####
4 #
5 ### MCF環境定義
6 mcftenv -s 01 ¥

```

5. システム定義  
定義例

```

7          -a      aplfil01
8 #
9 ### MCF通信構成共通定義
10 mcftcomn  -j      32768                ¥
11          -x      "termrls=no"
12 #
13 ### トレース環境定義
14 mcfttrc   -t      "size=204800        ¥
                  disk=yes"
15 #
16 ### バッファグループ定義 (メッセージ送信バッファグループ)
17 mcftbuf   -g      "groupno=1          ¥
                  length=4096          ¥
                  count=256            ¥
                  extend=256"
21 #
22 ### バッファグループ定義 (メッセージ受信バッファグループ)
23 mcftbuf   -g      "groupno=2          ¥
                  length=4096          ¥
                  count=256            ¥
                  extend=256"
27 #
28 ### バッファグループ定義 (メッセージ編集用バッファグループ)
29 mcftbuf   -g      "groupno=3          ¥
                  length=4096          ¥
                  count=256            ¥
                  extend=256"

```

コーディング例 (新 OSAS/UA プロトコルで XNF/AS 使用時の TP1/NET/User Agent の  
プロトコル固有定義)

```

1 #####
2 ### MCF通信構成定義 TP1/NET/User Agentプロトコル固有定義###
3 #####
4 #
5 ### コネクション定義の開始 (coua01)
6 mcftalccn -c coua01                ¥
7          -p      ua                  ¥
8          -n      x'0a81008202001283020012' ¥
9          -g      "sndbuf=1            ¥
                  rcvbuf=2"           ¥
10         -e      "msgbuf=3            ¥
                  count=5"           ¥
11         -m      "mode=xnfas"         ¥
12         -i      auto                 ¥
13         -o      new                  ¥
14         -u      ht                   ¥
15         -w      "nomltim=60          ¥
                  usertim=60"        ¥
16         -b      "bretry=yes          ¥
                  bretrycnt=10       ¥
                  bretryint=60"      ¥
17         -k      each                 ¥
18         -d      0                    ¥
19         -q      x'15810082008302ffff840b49000142010300005ffe01' ¥
20         -z      "slot=103"          ¥
21
22
23
24
25

```



```

26 ### 論理端末/UA定義 (reply型:leid11)
27     mcftalcle -l leid11
28             -t reply
29             -m "mmsgcnt=20
30                dmsgcnt=20"
31             -k "quekind=disk
32                quegrpid=otqgrp01"
33             -o "aj=no"
34     mcftalcua -u 1
35 ### 論理端末/UA定義 (reply型:leid12)
36     mcftalcle -l leid12
37             -t reply
38             -m "mmsgcnt=20
39                dmsgcnt=20"
40             -k "quekind=disk
41                quegrpid=otqgrp01"
42             -o "aj=no"
43     mcftalcua -u 2
44 ### 論理端末/UA定義 (reply型:leid13)
45     mcftalcle -l leid13
46             -t reply
47             -m "mmsgcnt=20
48                dmsgcnt=20"
49             -k "quekind=disk
50                quegrpid=otqgrp01"
51             -o "aj=no"
52     mcftalcua -u 3
53 ### 論理端末/UA定義 (request型:leid14)
54     mcftalcle -l leid14
55             -t request
56             -m "mmsgcnt=20
57                dmsgcnt=20"
58             -k "quekind=disk
59                quegrpid=otqgrp01"
60             -o "aj=no"
61     mcftalcua -u 4
62 ### 論理端末/UA定義 (request型:leid15)
63     mcftalcle -l leid15
64             -t request
65             -m "mmsgcnt=20
66                dmsgcnt=20"
67             -k "quekind=disk
68                quegrpid=otqgrp01"
69             -o "aj=no"
70     mcftalcua -u 5
71 ### コネクション定義の終了 (coua01)
72     mcftalced

```

## 注

AIX用に提供するファイルの内容です。

HP-UX用に使用する場合は次のように変更してください。

```
-m "mode=xnfh"
```

## 5. システム定義 定義例

### 旧 OSAS/UA プロトコルを使用する場合の定義例

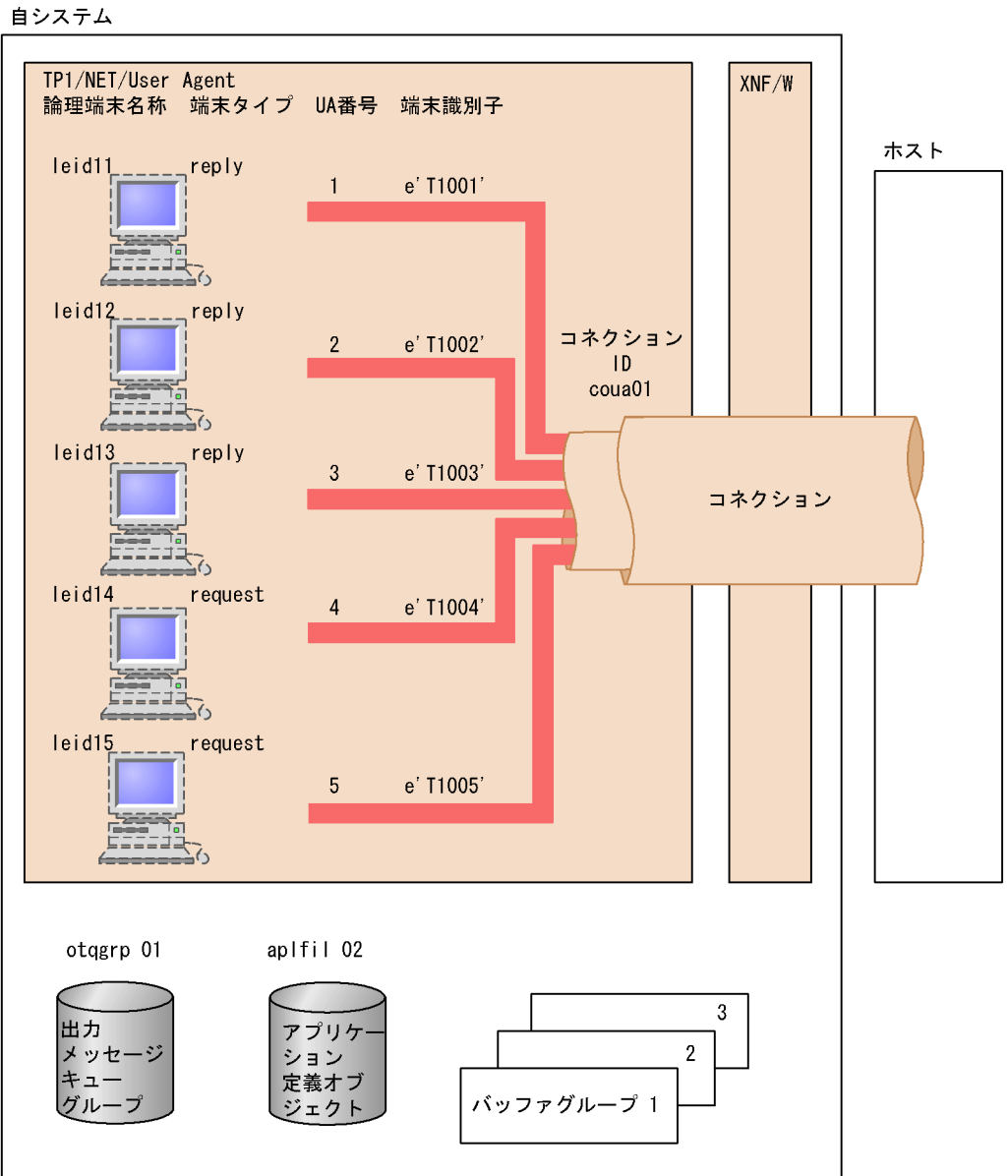
旧 OSAS/UA プロトコルを使用した場合に、TP1/NET/User Agent を使用したシステム定義の例を示します。

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

- /BeTRAN/examples/mcf/UserAgent/conf/com\_c1
- /BeTRAN/examples/mcf/UserAgent/conf/com\_d1

図 5-10 TP1/NET/User Agent のシステム構成例 (旧 OSAS/UA プロトコル使用時)



コーディング例 (旧 OSAS/UA プロトコルで XNF/AS 使用時の TP1/NET/User Agent の共通定義)

```

1 #####
2 ### MCF通信構成定義 共通定義 #####
3 #####
4 #
5 ### MCF環境定義
6 mcftenv -s 01 ¥

```

5. システム定義  
定義例

```

7          -a      aplfil01
8 #
9 ### MCF通信構成共通定義
10 mcftcomn  -j      32768                ¥
11          -x      "termrls=no"
12 #
13 ### トレース環境定義
14 mcfttrc   -t      "size=204800        ¥
                  disk=yes"
15 #
16 ### バッファグループ定義 (メッセージ送信バッファグループ)
17 mcftbuf   -g      "groupno=1         ¥
                  length=4096         ¥
                  count=256           ¥
                  extend=256"
21 #
22 ### バッファグループ定義 (メッセージ受信バッファグループ)
23 mcftbuf   -g      "groupno=2         ¥
                  length=4096         ¥
                  count=256           ¥
                  extend=256"
27 #
28 ### バッファグループ定義 (メッセージ編集用バッファグループ)
29 mcftbuf   -g      "groupno=3         ¥
                  length=4096         ¥
                  count=256           ¥
                  extend=256"

```

コーディング例 (旧 OSAS/UA プロトコルで XNF/AS 使用時の TP1/NET/User Agent の  
プロトコル固有定義)

```

1 #####
2 ### MCF通信構成定義 TP1/NET/User Agentプロトコル固有定義###
3 #####
4 #
5 ### コネクション定義の開始 (coua01)
6 mcftalccn -c coua01                ¥
7          -p ua                      ¥
8          -n x'0a81008202001283020012' ¥
9          -g "sndbuf=1                ¥
                  rcvbuf=2"          ¥
10         -e "msgbuf=3                ¥
                  count=5"          ¥
11         -m "mode=xnfas"             ¥
12         -i auto                     ¥
13         -o old                       ¥
14         -u ht                        ¥
15         -w "nomltim=60              ¥
                  usertim=60"       ¥
16         -b "bretry=yes              ¥
                  bretrycnt=10      ¥
                  bretryint=60"     ¥
17         -k each                      ¥
18         -d 5                         ¥
19         -y e'T1000'                  ¥
20         -q x'15810082008302ffff840b49000142010300005ffe01' ¥
21         -z "slot=3"

```

```

27 ### 論理端末/UA定義 (reply型:leid11)
28     mcftalcle -l leid11
29             -t reply
30             -m "mmsgcnt=20
31                dmsgcnt=20"
32             -k "quekind=disk
33                quegrpId=otqgrp01"
34             -o "aj=no"
35     mcftalcua -u 1
36             -y e'T1001'
37 ### 論理端末/UA定義 (reply型:leid12)
38     mcftalcle -l leid12
39             -t reply
40             -m "mmsgcnt=20
41                dmsgcnt=20"
42             -k "quekind=disk
43                quegrpId=otqgrp01"
44             -o "aj=no"
45     mcftalcua -u 2
46             -y e'T1002'
47 ### 論理端末/UA定義 (reply型:leid13)
48     mcftalcle -l leid13
49             -t reply
50             -m "mmsgcnt=20
51                dmsgcnt=20"
52             -k "quekind=disk
53                quegrpId=otqgrp01"
54             -o "aj=no"
55     mcftalcua -u 3
56             -y e'T1003'
57 ### 論理端末/UA定義 (request型:leid14)
58     mcftalcle -l leid14
59             -t request
60             -m "mmsgcnt=20
61                dmsgcnt=20"
62             -k "quekind=disk
63                quegrpId=otqgrp01"
64             -o "aj=no"
65     mcftalcua -u 4
66             -y e'T1004'
67 ### 論理端末/UA定義 (request型:leid15)
68     mcftalcle -l leid15
69             -t request
70             -m "mmsgcnt=20
71                dmsgcnt=20"
72             -k "quekind=disk
73                quegrpId=otqgrp01"
74             -o "aj=no"
75     mcftalcua -u 5
76             -y e'T1005'
77 ### コネクション定義の終了 (coua01)
78     mcftalced

```

## 注

AIX 用に提供するファイルの内容です。

HP-UX 用に使用する場合は次のように変更してください。

5. システム定義  
定義例

```
-m "mode=xfh"
```

# 6

## 運用コマンド

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

---

TP1/NET/User Agent の運用コマンド

---

mcfactcn ( コネクションの確立 )

---

mcftactle ( 論理端末の閉塞解除 )

---

mcfdctcn ( コネクションの解放 )

---

mcfdctle ( 論理端末の閉塞 )

---

mcftlscn ( コネクションの状態表示 )

---

mcftlsle ( 論理端末の状態表示 )

---

## 6. 運用コマンド

TP1/NET/User Agent の運用コマンド

# TP1/NET/User Agent の運用コマンド

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

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

表 6-1 TP1/NET/User Agent の運用コマンド

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

(凡例)

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



## mcfactcn (コネクションの確立)

---

### 形式

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

### 機能

コネクションを確立します。

### オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID ~ 1 ~ 8 文字の識別子

確立するコネクションのコネクション ID を指定します。

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

同一コネクション ID は、重複して指定できません。また、コネクション ID は、\* を使って一括指定ができます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定するときは、引用符 (") で囲んで指定します。

\* : すべてのコネクションを確立します。

先行文字列 \* : 先行文字列で始まるすべてのコネクションを確立します。

複数指定の例 cnn1, cnn2, cnn3 を指定する場合

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

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

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

### 出力メッセージ

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

## 6. 運用コマンド

mcftactcn (コネクションの確立)

出力メッセージ ID	内容	出力先
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftactcn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftactcn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftactcn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10500-I	ヘルプメッセージ	標準出力
KFCA13130-E	TP1/NET/User Agent で論理エラーが発生しました。	標準エラー出力
KFCA13132-E	コネクション確立済みです。	標準エラー出力
KFCA13133-E	コネクション確立処理中です。	標準エラー出力
KFCA13134-E	コネクション解放処理中です。	標準エラー出力
KFCA13143-E	コネクション確立が着呼指定です。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

## mcfactle (論理端末の閉塞解除)

---

### 形式

```
mcfactle [-s MCF通信プロセス識別子] [-c コネクションID]
          -l 論理端末名称
```

### 機能

論理端末の閉塞を解除します。

### オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

`-c コネクション ID`        ~    1 ~ 8 文字の識別子

閉塞解除したい論理端末に対応するコネクションのコネクション ID を指定します。

コネクション ID は複数指定できません。また、一括指定もできません。

`-l 論理端末名称`        ~    1 ~ 8 文字の識別子

閉塞解除する論理端末の名称を指定します。

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

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

同一コネクション ID は、重複して指定できません。また、コネクション ID は、\* を使って一括指定ができます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定するときは、引用符 (") で囲んで指定します。

\* : すべての論理端末の閉塞を解除します。

先行文字列 \* : 先行文字列で始まるすべての論理端末の閉塞を解除します。

複数指定の例    len1 , len2 , len3 を指定する場合

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

一括指定の例    len で始まるすべての論理端末を指定する場合

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

## 6. 運用コマンド

mcftactle (論理端末の閉塞解除)

### 出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftactle コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftactle コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftactle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftactle コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定されたコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10503-I	ヘルプメッセージ	標準出力
KFCA13131-E	コネクションが確立されていません。	標準エラー出力
KFCA13136-E	構成合わせで使用できなくなった論理端末です。	標準エラー出力
KFCA13137-E	閉塞解除済みです。	標準エラー出力
KFCA13138-E	閉塞解除処理中です。	標準エラー出力
KFCA13139-E	論理端末の閉塞処理中です。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

## mcftdctcn (コネクションの解放)

---

### 形式

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

### 機能

コネクションを解放します。

このコマンド入力時、該当するコネクションに仕掛り中の UA がある場合、このコマンド処理は異常終了します。仕掛り中の UA がないことを確認してからコネクションを正常解放してください。

コマンド入力時に送信中であったメッセージの扱いについては、「8.1 障害の種類と対応処理」を参照してください。なお、コネクションが強制的に解放されても、UAP からの送信要求はできません。

### オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID    ~    1 ~ 8 文字の識別子

解放するコネクションのコネクション ID を指定します。

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

同一コネクション ID は、重複して指定できません。また、コネクション ID は、\* を使って一括指定ができます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定するときは、引用符 (") で囲んで指定します。

\* : すべてのコネクションを解放します。

先行文字列 \* : 先行文字列で始まるすべてのコネクションを解放します。

複数指定の例    cnn1 , cnn2 , cnn3 を指定する場合

```
-c "cnn1    cnn2    cnn3"
```

一括指定の例    cnn で始まるすべてのコネクションを指定する場合

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

## 6. 運用コマンド

mcftdctcn (コネクションの解放)

-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けり中の場合、仕掛けり中の処理を終了しないで無条件に解放します。

このオプションの指定を省略すると、該当するコネクションが仕掛けり中の場合、このコマンドは異常終了します。

### 出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	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。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftdctcn 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	mcftdctcn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftdctcn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10501-I	ヘルプメッセージ	標準出力
KFCA13130-E	TP1/NET/User Agent で論理エラーが発生しました。	標準エラー出力
KFCA13131-E	コネクションが確立されていません。	標準エラー出力
KFCA13133-E	コネクション確立処理中です。	標準エラー出力
KFCA13134-E	コネクション解放処理中です。	標準エラー出力
KFCA13141-E	論理端末が仕掛けり中です。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

## mcftdctl (論理端末の閉塞)

---

### 形式

```
mcftdctl [-s MCF通信プロセス識別子] [-c コネクションID]
          -l 論理端末名称
```

### 機能

論理端末を閉塞します。

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

### オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID ~ 1 ~ 8 文字の識別子

閉塞したい論理端末に対応するコネクションのコネクション ID を指定します。

コネクション ID は複数指定できません。また、一括指定もできません。

-l 論理端末名称 ~ 1 ~ 8 文字の識別子

閉塞する論理端末の名称を指定します。

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

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

同一コネクション ID は、重複して指定できません。また、コネクション ID は、\* を使って一括指定ができます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定するときは、引用符 (") で囲んで指定します。

\* : すべての論理端末を閉塞します。

先行文字列 \* : 先行文字列で始まるすべての論理端末を閉塞します。

複数指定の例 len1, len2, len3 を指定する場合

## 6. 運用コマンド

mcftdctl (論理端末の閉塞)

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

一括指定の例 len で始まるすべての論理端末を指定する場合

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

### 出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftdctl コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftdctl コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftdctl コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftdctl コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定されたコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10504-I	ヘルプメッセージ	標準出力
KFCA13131-E	コネクションが確立されていません。	標準エラー出力
KFCA13135-E	論理端末は閉塞解除されていません。	標準エラー出力
KFCA13136-E	構成合わせで使用できなくなった論理端末です。	標準エラー出力
KFCA13138-E	閉塞解除処理中です。	標準エラー出力
KFCA13139-E	論理端末の閉塞処理中です。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

### 注意事項

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

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



## mcftlscn ( コネクションの状態表示 )

---

### 形式

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

### 機能

コネクションの状態を標準出力に出力します。

### オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID    ~    1 ~ 8 文字の識別子

状態を表示するコネクションのコネクション ID を指定します。

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

同一コネクション ID は、重複して指定できません。また、コネクション ID は、\* を使って一括指定ができます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定するときは、引用符 ( " ) で囲んで指定します。

\* : すべてのコネクションの状態を表示します。

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

複数指定の例    cnn1 , cnn2 , cnn3 を指定する場合

```
-c "cnn1    cnn2    cnn3"
```

一括指定の例    cnn で始まるすべてのコネクションを指定する場合

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

```
-d
```

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

このオプションの指定を省略すると、コネクションの状態だけを表示します。

### 出力形式

## 6. 運用コマンド

mcftlscn (コネクションの状態表示)

```
mmm cccccc ppp sssss dddd  
lllllllll ttt uuuu xxxx
```

### 注

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

- mmm : MCF 識別子
- cccccc : コネクション ID
- ppp : プロトコル種別  
UA...OSAS/UA 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)
- lllllll : 論理端末名称
- ttt : 論理端末タイプ  
REQ...request 型  
RLY...reply 型  
SND...send 型  
RCV...receive 型
- uuuu : UA 番号
- xxxx : UA ステータス (保守情報)

### 出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftlscn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftlscn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10360-I	状態表示を開始します。	標準出力

出力メッセージ ID	内容	出力先
KFCA10361-I	標準情報を表示します。	標準出力
KFCA10362-I	詳細情報を表示します。	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10371-I	mcftlscn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftlscn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10502-I	ヘルプメッセージ	標準出力
KFCA13412-E	資源不足です。	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

## mcftlsle (論理端末の状態表示)

---

### 形式

```
mcftlsle [-s MCF通信プロセス識別子] [-c コネクションID]
         -l 論理端末名称 [-q] [-t]
```

### 機能

論理端末の状態を標準出力に出力します。

### オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

**-c** コネクション ID    ~    1 ~ 8 文字の識別子

状態を表示したい論理端末に対応するコネクションのコネクション ID を指定します。コネクション ID は複数指定できません。また、一括指定もできません。

**-l** 論理端末名称    ~    1 ~ 8 文字の識別子

状態を表示する論理端末の名称を指定します。

**-c** オプションを指定した場合、指定したコネクション ID に対応する論理端末名称を指定します。

論理端末名称は、一度に 8 個まで指定できます。複数指定するときは、引用符 (") で囲んで、論理端末名称と論理端末名称との間を空白で区切ります。同一コネクション ID は、重複して指定できません。また、コネクション ID は、\* を使って一括指定ができません。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定するときは、引用符 (") で囲んで指定します。

**\***: すべての論理端末の状態を表示します。

先行文字列 **\***: 先行文字列で始まるすべての論理端末の状態を表示します。

複数指定の例    len1, len2, len3 を指定する場合

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

一括指定の例    len で始まるすべての論理端末を指定する場合

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

-q

指定した論理端末に対応する出力キューの保留状態を表示します。

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

-t

指定した論理端末がセキュア状態かどうかを表示します。

## 出力形式

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

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

## 6. 運用コマンド

mcftlsle (論理端末の状態表示)

### 出力メッセージ

出力メッセージID	内容	出力先
KFCA10350-I	mcftlsle コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftlsle コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10360-I	状態表示を開始します。	標準出力
KFCA10364-I	表示情報	標準出力
KFCA10365-I	表示情報	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10371-I	mcftlsle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftlsle コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10395-E	指定したコネクションには未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10505-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

# 7

## 組み込み方法

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

---

7.1 TP1/NET/User Agent の組み込みの流れ

---

7.2 MCF メイン関数の作成

---

7.3 定義オブジェクトファイルの生成

---

## 7.1 TP1/NET/User Agent の組み込みの流れ

---

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

### (1) MCF メイン関数の作成

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

### (2) MCF サービス名の登録

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

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

詳細は、マニュアル「OpenTP1 システム定義」を参照してください。

### (3) システムサービス情報定義ファイルの作成

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

### (4) 定義オブジェクトファイルの生成

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



## 7.2 MCF メイン関数の作成

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

TP1/NET/User Agent を起動するためには、ユーザが MCF メイン関数を作成し、コンパイル、およびリンケージを行って TP1/NET/User Agent 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、mcfplua コマンドを使用します。

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

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

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

なお、これらのコーディング例は、次のファイルで提供しています。

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

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

```

#include <dcmosas.h>                                /*TP1/NET/User Agent用ヘッダファイル */ 1.
extern DCLONG msgrcv01(dcmcf_uoc_min_n *);          /*入力メッセージ編集UOC関数extern宣言 */ 2.
extern DCLONG msgsend01(dcmcf_uoc_mout_n *);        /*出力メッセージ編集UOC関数extern宣言 */ 2.
extern dcmcf_uoc_t dcmcf_uoctbl;                   /*UOCテーブルextern宣言 */ 3.

int main()
{
    dcmcf_uoctbl.msgrcv = (dcmcf_uocfunc)msgrcv01;  /*入力メッセージ編集UOCアドレス設定 */ 4.
    dcmcf_uoctbl.msgsend = (dcmcf_uocfunc)msgsend01; /*出力メッセージ編集UOCアドレス設定 */ 4.
    dc_mcf_svstart();                               /*スタート関数の呼び出し*/ 5.
    return 0;
}

```

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

```

#include <dcmosas.h>                /*TP1/NET/User Agent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/User Agent で提供するヘッダファイルを取り込みます。
2. 使用する UOC 関数を extern 宣言します。UOC のリターン値は DCLONG 型にしてください。

UOC をまったく使用しない場合、このコーディングは必要ありません。

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

UOC をまったく使用しない場合、このコーディングは必要ありません。

4. 各 UOC 関数のアドレスを、次に示すシステム提供変数に設定します。使用する UOC だけコーディングしてください。

```

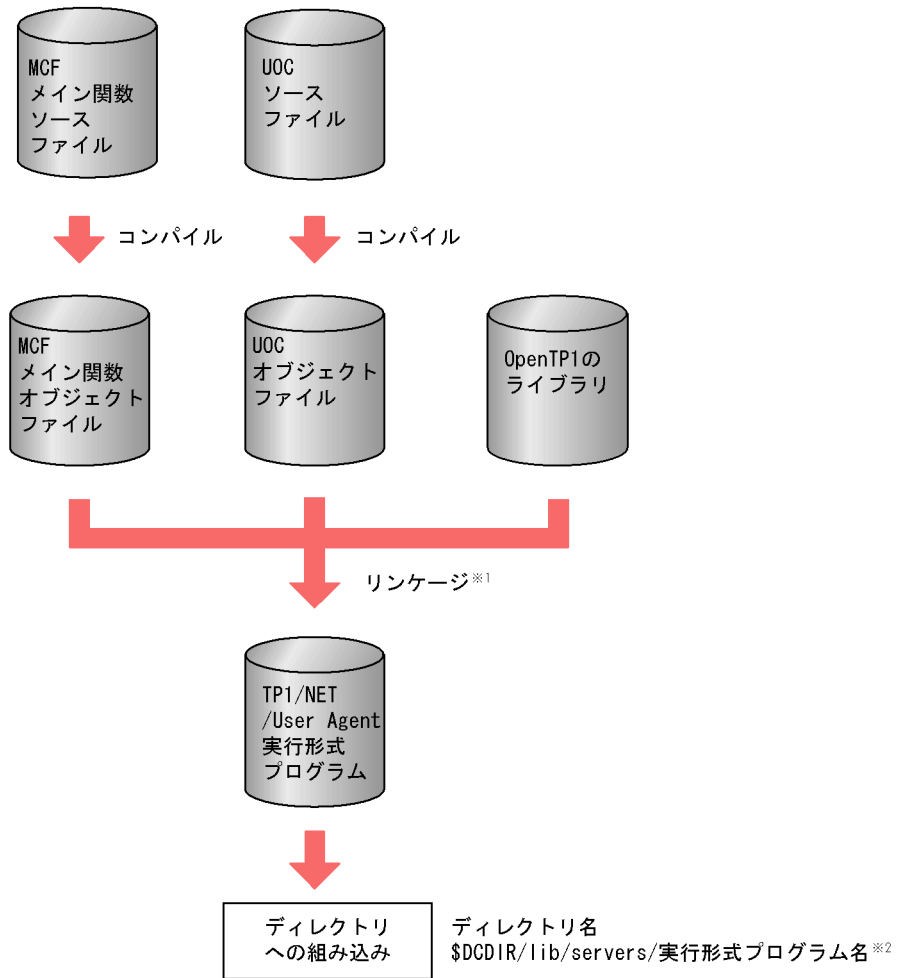
dcmcf_uoctbl.msgrcv /*入力メッセージ編集UOCアドレス*/
dcmcf_uoctbl.msgsend /*出力メッセージ編集UOCアドレス*/

```

UOC をまったく使用しない場合、このコーディングは必要ありません。

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

図 7-3 MCF メイン関数のディレクトリへの組み込み方法



注 1  
mcfplua コマンドでリンケージします。

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

## 7.3 定義オブジェクトファイルの生成

---

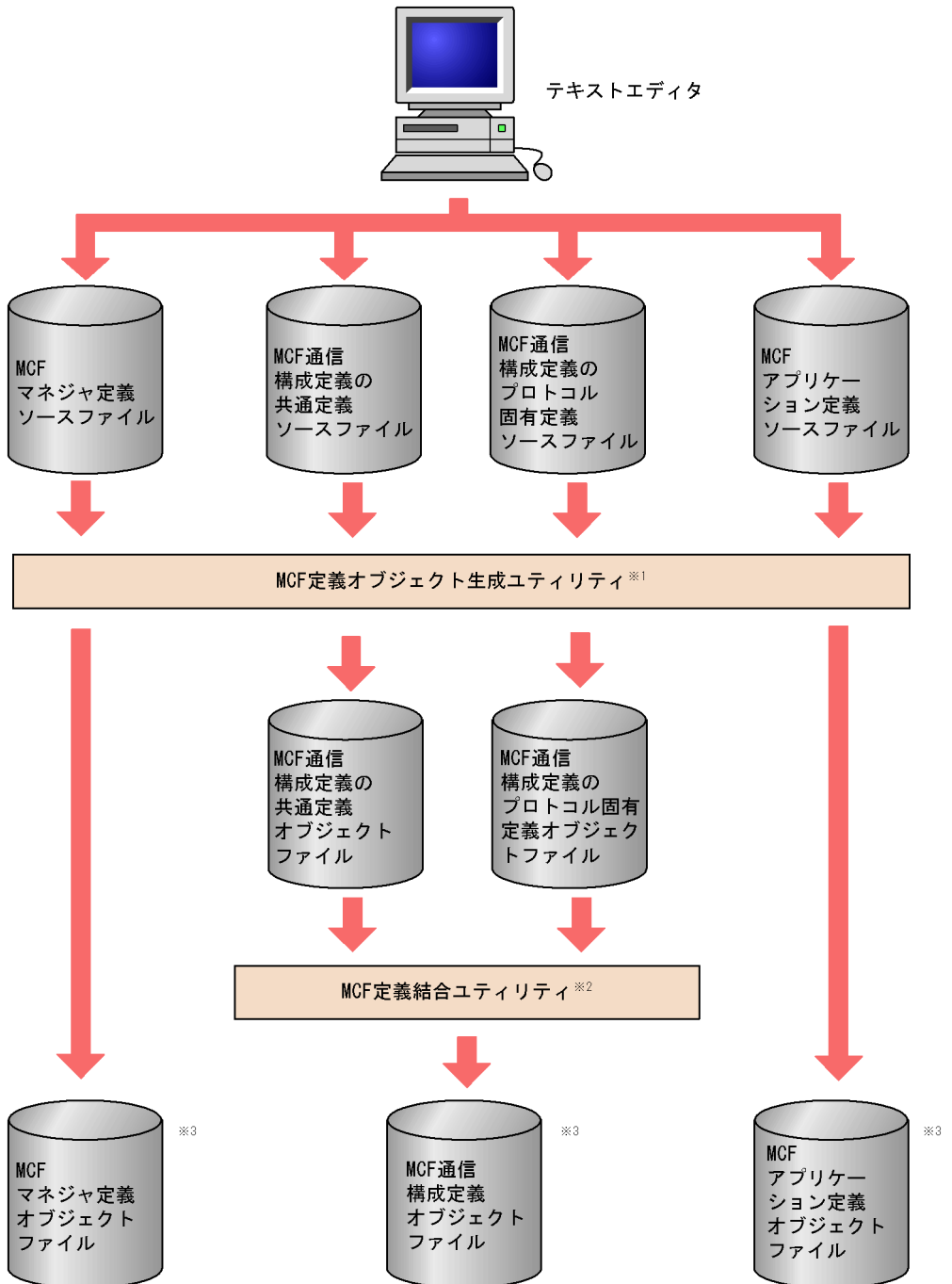
定義オブジェクトファイルを次の手順で生成します。

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

1. テキストエディタを使用して、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/User Agent 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/User Agent のプロトコル固有定義オブジェクトファイル
  - MCF アプリケーション定義オブジェクトファイル
3. MCF 定義結合ユーティリティを使用して、MCF 通信構成定義の共通定義とプロトコル固有定義のオブジェクトファイルを結合し、次に示すオブジェクトファイルを作成します。
  - MCF 通信構成定義オブジェクトファイル

定義オブジェクトファイルの作成方法の概要を次の図に示します。

図 7-4 定義オブジェクトファイルの作成方法の概要



注 1  
次に示すコマンドで生成します。

## 7. 組み込み方法

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

mcfXXXX は、ソースファイルごとに異なります。

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

### 注 2

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

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

### 注 3

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

# 8

## 障害対策

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

---

8.1 障害の種類と対応処理

---

8.2 コネクション障害

---

8.3 問い合わせメッセージ、応答メッセージ障害

---

8.4 一方送信メッセージ障害

---

8.5 論理端末の端末タイプ不正

---

## 8.1 障害の種類と対応処理

TP1/NET/User Agent 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（アソシエーション）障害
- UA（論理端末）障害
- 入力メッセージ編集 UOC の障害
- 入力キュー、スケジュールサービス、ネームサービス、または RPC の障害
- 出力キューおよび出力メッセージ編集 UOC の障害
- ジャーナル障害
- 送信バッファ障害
- 受信バッファ障害
- 編集バッファ障害
- UAP 障害
- MCF の障害
- OpenTP1 の障害

### （1）コネクション（アソシエーション）障害

コネクション障害の発生個所に応じた TP1/NET/User Agent の障害処理については、「8.2 コネクション障害」を参照してください。

表 8-1 コネクション障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
<ul style="list-style-type: none"> <li>• UA 層論理矛盾</li> <li>• 下位層障害</li> <li>• リトライオーバー</li> <li>• mcftdctn -f コマンド入力</li> </ul> など	<ol style="list-style-type: none"> <li>1. 各 UA の障害処理をします（「(2) UA（論理端末）障害」参照）。</li> <li>2. CERREVT（コネクション障害）を起動します。</li> <li>3. コネクション障害を通知するメッセージログ（KFCA13102-E）を出力します。</li> <li>4. コネクションを解放します。</li> </ol>	運用コマンド（mcftactcn）を入力して、再びコネクションを確立します。



## (2) UA (論理端末) 障害

表 8-2 UA 障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
<ul style="list-style-type: none"> <li>• UA 障害</li> <li>• UA 層検出異常</li> <li>• タイムアウト</li> <li>• mcftdctle コマンド入力</li> </ul> など	<ol style="list-style-type: none"> <li>1. UA 障害を通知するメッセージログ (KFCA13103-E), および UA 閉局を通知するメッセージログ (KFCA13115-I) を出力します。</li> <li>2. CERREVT (UA 障害) を起動します。</li> <li>3. 論理端末を閉塞します。</li> <li>4. UA (論理端末) の端末タイプに従って処理します。<sup>1</sup></li> </ol>	運用コマンド (mcftactle) を入力して, UA 閉局処理をします。
受信メッセージと論理端末の端末タイプの不一致	<ol style="list-style-type: none"> <li>1. 受信メッセージを破棄します。<sup>2</sup></li> <li>2. 相手システムに UERR を送信します。</li> <li>3. UA 障害を通知するメッセージログ (KFCA13103-E), および UA 閉局を通知するメッセージログ (KFCA13115-I) を出力します。</li> <li>4. CERREVT (UA 障害) を起動します。</li> <li>5. 論理端末を閉塞します。</li> </ol>	相手システムとの構成を見直してください。
送信メッセージと論理端末の端末タイプの不一致	dc_mcf_send, dc_mcf_reply, dc_mcf_sendrecv, dc_mcf_recvsync がエラーリターンします。	UAP で後処理をする必要があります。
UA 再開局処理の失敗	<ol style="list-style-type: none"> <li>1. 該当する UA を縮退させます。</li> <li>2. UA 障害を通知するメッセージログ (KFCA13103-E), および UA 閉局を通知するメッセージログ (KFCA13115-I) を出力します。</li> <li>3. CERREVT (UA 障害) を起動します。</li> <li>4. 論理端末を閉塞します。</li> </ol>	障害要因を取り除いたあと, 運用コマンド (mcftactn) を入力して, 再びコネクションを確立します。

## 注 1

論理端末の端末タイプごとの処理内容を次の表に示します。

論理端末の端末タイプ	TP1/NET/User Agent の処理
reply	<p>問い合わせメッセージ (UINQ) 受信障害の場合は, 受信メッセージを無効にします。</p> <p>応答メッセージ (UREP) 送信障害の場合は, 次に示す処理をします。</p> <ol style="list-style-type: none"> <li>1. 送信メッセージを送信済み扱いにします。</li> <li>2. 送信破棄を通知するメッセージログ (KFCA10607-W) を出力します。</li> </ol>
request	<p>一方送信メッセージ (UINQ) 送信障害, および一方送信メッセージ (UREP) 受信障害の場合は, 次に示す処理をします。</p> <ol style="list-style-type: none"> <li>1. 一方送信メッセージ (UINQ) の先頭セグメントからの再送準備をします。</li> <li>2. 送信中断を通知するメッセージログ (KFCA10608-W) を出力します。</li> </ol> <p>UA 再開局後, 一方送信メッセージ (UINQ) を再送します。</p> <p>送信完了通知 (UINF) 送信障害です。</p>

## 8. 障害対策

論理端末の端末タイプ	TP1/NET/User Agent の処理
request (send-recv 時)	一方送信メッセージ (UINQ) 送信障害, および一方送信メッセージ (UREP) 受信障害の場合は, 問い合わせメッセージ (UINQ) を送信済みにし, UAP にエラーリターンします。 送信完了通知 (UINF) 送信障害です。
send	一方送信メッセージ (UBRD) 送信障害, および送信完了通知 (UINF) 受信障害の場合は, 次に示す処理をします。 1. 一方送信メッセージ (UBRD) の先頭セグメントからの再送準備をします。 2. 送信中断を通知するメッセージログ (KFCA13108-W) を出力します。 UA 再開局後, 一方送信メッセージ (UBRD) を再送します。
receive	一方送信メッセージ (UBRD) 受信障害の場合は, 一方送信メッセージを無効にします。 処理完了通知 (UINF) 送信障害です。

### 注 2

1 論理メッセージの組み立て以前に不一致が検出されるため, ERREVT は起動しません。

### (3) 入力メッセージ編集 UOC の障害

表 8-3 入力メッセージ編集 UOC の障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
<ul style="list-style-type: none"> <li>• UOC エラーリターン</li> <li>• UOC 指定誤り</li> </ul>	<ol style="list-style-type: none"> <li>1. UOC エラーを通知するメッセージログ (KFCA10611-E), UOC 指定誤りを通知するメッセージログ (KFCA10620-E), およびアプリケーション不正を通知するメッセージログ (KFCA10610-E) を出力します。</li> <li>2. UA (論理端末) の端末タイプに従って処理します。</li> </ol>	UOC の処理を見直してください。

### 注

論理端末の端末タイプごとの処理内容を次の表に示します。

論理端末の端末タイプ	TP1/NET/User Agent の処理
reply	<ol style="list-style-type: none"> <li>1. 相手システムに UERR を送信します。</li> <li>2. CERREVT (UOC 障害) を起動します。</li> <li>3. UA 障害を通知するメッセージログ (KFCA13103-E), および UA 閉局を通知するメッセージログ (KFCA13115-I) を出力します。</li> <li>4. 論理端末を閉塞します。</li> </ol>
request	<ol style="list-style-type: none"> <li>1. 相手システムに UERR を送信します。</li> <li>2. CERREVT (UOC 障害) を起動します。</li> <li>3. UA 障害を通知するメッセージログ (KFCA13103-E), および UA 閉局を通知するメッセージログ (KFCA13115-I) を出力します。</li> <li>4. 送信メッセージ (UINQ) は送信済み扱いとします。</li> <li>5. 論理端末を閉塞します。</li> </ol>
request (send-recv 時)	<ol style="list-style-type: none"> <li>1. 相手システムに UERR を送信します。</li> <li>2. CERREVT (UOC 障害) を起動します。</li> <li>3. UAP にエラーリターンします。</li> <li>4. UA 障害を通知するメッセージログ (KFCA13103-E), および UA 閉局を通知するメッセージログ (KFCA13115-I) を出力します。</li> <li>5. 送信メッセージ (UINQ) は送信済み扱いとします。</li> <li>6. 論理端末を閉塞します。</li> </ol>
send	該当しません。
receive	<ol style="list-style-type: none"> <li>1. 相手システムに UERR を送信します。</li> <li>2. CERREVT (UOC 障害) を起動します。</li> <li>3. UA 障害を通知するメッセージログ (KFCA13103-E), および UA 閉局を通知するメッセージログ (KFCA13115-I) を出力します。</li> <li>4. 論理端末を閉塞します。</li> </ol>

#### (4) 入力キュー, スケジュールサービス, ネームサービス, または RPC の障害

表 8-4 入力キュー, スケジュールサービス, ネームサービス, または RPC の障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
<ul style="list-style-type: none"> <li>• アプリケーション閉塞</li> <li>• サービス閉塞</li> <li>• アプリケーション名不正<sup>1</sup></li> <li>• 入力キュー書き込み障害 (メモリ含む)</li> </ul>	<ol style="list-style-type: none"> <li>1. 入力キュー障害を通知するメッセージログ (KFCA10604-E) を出力します。</li> <li>2. 入力メッセージの種類に従って次に示す処理をします。通信メッセージ入力障害の場合は, ERREVT1 または ERREVT2 を起動します。ERREVT1, ERREVT2, ERREVT3, ERREVT4, または ERREVT4 の場合は, メッセージを破棄します。CERREVT, COPNEVT, または CCLSEVT の場合は, イベントを破棄します。</li> </ol>	障害要因を取り除いてください。

8. 障害対策

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
アプリケーション名の形式不正	<ol style="list-style-type: none"> <li>1. アプリケーション不正を通知するメッセージログ (KFCA10610-E) を出力します。</li> <li>2. 相手システムに UERR を送信します。</li> </ol>	アプリケーション名の形式を見直してください。
同期メッセージリターン時障害 <sup>2</sup>	<ol style="list-style-type: none"> <li>1. 同期エラーを通知するメッセージログ (KFCA10606-E) を出力します。</li> <li>2. UAP ではタイムアウトが発生します。</li> <li>3. リターンの種類に従って次に示す処理をします。 正常にリターンした場合は、次のように処理します。 <ol style="list-style-type: none"> <li>1. 相手システムに UERR を送信します。</li> <li>2. CERREVT を起動します。</li> </ol> </li> <li>3. UA 障害を通知するメッセージログ (KFCA13103-E), および UA 閉局を通知するメッセージログ (KFCA13115-I) を出力します。</li> <li>4. 論理端末を閉塞します。 エラーリターンした場合は、無視して処理を続けます。</li> </ol>	障害要因を取り除いたあと、運用コマンド (mcftactcn) を入力して、再び接続を確立します。

注 1

アプリケーション名の形式が不正の場合、ERREVT1 を起動します。この ERREVT1 に対する応答メッセージを送信しなかった場合、相手システムに UERR を送信後、CERREVT を起動します。

注 2

RPC だけで発生する障害です。

(5) 出力キューおよび出力メッセージ編集 UOC の障害

表 8-5 出力キューおよび出力メッセージ編集 UOC の障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
<ul style="list-style-type: none"> <li>• 出力キュー読み込み障害 (メモリ含む)</li> <li>• UOC エラーリターン</li> <li>• UOC 指定誤り</li> </ul>	<ol style="list-style-type: none"> <li>1. UOC エラーを通知するメッセージログ (KFCA10611-E), UOC 指定誤りを通知するメッセージログ (KFCA10620-E), および出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。</li> <li>2. UA (論理端末) の端末タイプに従って処理します。</li> </ol>	障害要因を取り除いてください。
<ul style="list-style-type: none"> <li>• メッセージ送信済み障害</li> <li>• メッセージリセット障害</li> </ul>	メッセージ送信後の障害を通知するメッセージログ (KFCA10617-E) を出力します。	処理を続けます。

注

論理端末の端末タイプごとの処理内容を次の表に示します。

論理端末の端末タイプ	TP1/NET/User Agent の処理
reply	<ol style="list-style-type: none"> <li>1. 該当するメッセージを送信済み扱いとし、送信破棄を通知するメッセージログ (KFCA10607-W) を出力します。</li> <li>2. 相手システムに UERR を送信します。</li> <li>3. UA 障害を通知するメッセージログ (KFCA13103-E), および UA 閉局を通知するメッセージログ (KFCA13115-I) を出力します。</li> <li>4. CERREVT (出力キュー障害) を起動します。</li> <li>5. 論理端末を閉塞します。</li> </ol>
request	<ol style="list-style-type: none"> <li>1. 該当するメッセージを送信済み扱いとし、送信破棄を通知するメッセージログ (KFCA10607-W) を出力します。</li> <li>2. 該当するメッセージを破棄し、次メッセージのサービスをします。</li> </ol>
request (send-recv 時)	<ol style="list-style-type: none"> <li>1. 該当するメッセージを送信済み扱いとし、送信破棄を通知するメッセージログ (KFCA10607-W) を出力します。</li> <li>2. UAP にエラーリターンします。</li> <li>3. 該当するメッセージを破棄し、次メッセージのサービスをします。</li> </ol>
send	<ol style="list-style-type: none"> <li>1. 該当するメッセージを送信済み扱いとし、送信破棄を通知するメッセージログ (KFCA10607-W) を出力します。</li> <li>2. 該当するメッセージを破棄し、次メッセージのサービスをします。</li> </ol>
receive	該当しません。

## (6) ジャーナル障害

表 8-6 ジャーナル障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
ジャーナル取得障害 (AJ/IJ/OJ)	ジャーナル障害を通知するメッセージログ (KFCA10609-E) を出力します。	処理を続けます。

## (7) 送信バッファ障害

表 8-7 送信バッファ障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
サイズ不足	出力キュー読み込み障害として扱います。 <ol style="list-style-type: none"> <li>1. 出力障害を通知するメッセージログ (KFCA10605-E), および送信破棄を通知するメッセージログ (KFCA10607-W) を出力します。</li> <li>2. 論理端末を閉塞します。</li> <li>3. CERREVT (メッセージ出力障害) を起動します。</li> </ol>	システム定義を修正してください。
バッファ数不足	出力キュー読み込み障害として扱います。 <ol style="list-style-type: none"> <li>1. バッファ不足を通知するメッセージログ (KFCA10618-E), および送信破棄を通知するメッセージログ (KFCA10607-E) を出力します。</li> <li>2. コネクションを解放します。</li> <li>3. CERREVT (バッファ取得障害) を起動します。</li> </ol>	システム定義を修正してください。

## 8. 障害対策

### (8) 受信バッファ障害

表 8-8 受信バッファ障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
サイズ不足	<ol style="list-style-type: none"> <li>不正データ受信を通知するメッセージログ (KFCA13274-E) を出力します。</li> <li>コネクションを解放します。</li> <li>CERREVT (受信障害) を起動します。</li> </ol>	相手システムとの構成を見直してください。
バッファ数不足	<ol style="list-style-type: none"> <li>バッファ不足を通知するメッセージログ (KFCA13213-E) を出力します。</li> <li>コネクションを解放します。</li> <li>CERREVT (受信障害) を起動します。</li> </ol>	システム定義を修正してください。

### (9) 編集バッファ障害

表 8-9 編集バッファ障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
サイズ不足	ありません。	必要に応じて、UOC でエラーリターンしてください。
バッファ数不足	<ol style="list-style-type: none"> <li>バッファ不足を通知するメッセージログ (KFCA10618-E) を出力します。</li> <li>障害の発生した処理に応じて、次に示す処理をします。 送信処理の場合は、次のように処理します。 <ol style="list-style-type: none"> <li>送信中断を通知するメッセージログ (KFCA13608-W) を出力します。</li> <li>送信メッセージを再送します。</li> </ol> 受信処理の場合は、受信メッセージを破棄します。 </li> <li>コネクションを解放します。</li> <li>CERREVT (バッファ取得障害) を起動します。</li> </ol>	システム定義を修正してください。

### (10) UAP 障害

表 8-10 UAP 障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
<ul style="list-style-type: none"> <li>UAP 異常終了</li> <li>rollback (UAP 終了指定) 呼び出し</li> </ul>	reply 型の論理端末を使用している場合は、相手システムに UERR を送信します。	UAP を見直してください。

## ( 11 )MCF の障害

表 8-11 MCF の障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
<ul style="list-style-type: none"> <li>• 内部論理矛盾</li> <li>• 他 PP とのインタフェースエラー</li> </ul>	<ol style="list-style-type: none"> <li>1. KFCA13197-E, KFCA13198-E, または KFCA13199-E を出力します。</li> <li>2. メモリダンプを出力します。</li> <li>3. 必要に応じて, MCF のプロセスを異常終了します ( 回復できる場合は終了しません )</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。
<ul style="list-style-type: none"> <li>• MCF プログラムエラー</li> <li>• UOC プログラムエラー</li> </ul>	OS によるコアダンプを出力します。	保守情報 ( \$DCDIR/spool ディレクトリ以下) を退避してください。

## ( 12 )OpenTP1 の障害

表 8-12 OpenTP1 の障害発生時の処理

障害の内容	TP1/NET/User Agent の処理	ユーザの処理
OpenTP1 異常終了	OS によるコアダンプを出力します。	保守情報 ( \$DCDIR/spool ディレクトリ以下) を退避してください。

## 8.2 コネクション障害

コネクションの処理中に、障害の発生する個所の区分を次の図に示します。また、障害発生個所に応じた TP1/NET/User Agent の障害処理について、表 8-13 に示します。

図 8-1 コネクション障害の発生個所

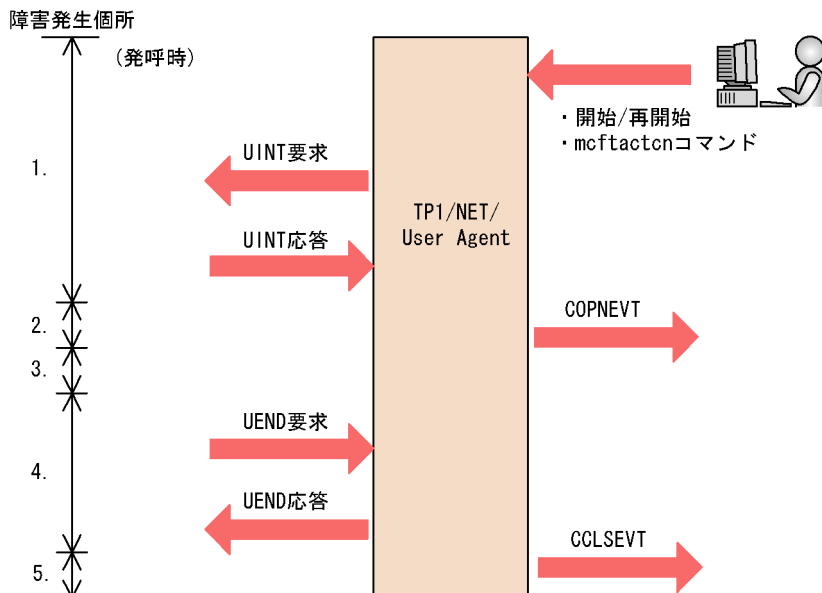


表 8-13 コネクション障害の発生個所に応じた障害処理

障害発生個所	内容	メッセージログの出力内容	TP1/NET/User Agent の処理
1.	コネクション確立時のリトライオーバー（下位層の障害，またはタイムアウトによるリトライ）	コネクション障害	<ul style="list-style-type: none"> <li>• CERREVT を起動します。</li> <li>• コネクションを解放します。</li> </ul> 解放後は運用コマンド（mcfactcn）で再確立できます。
2.	入力キュー書き込みエラー（COPNEVT）	メッセージ入力障害	<ul style="list-style-type: none"> <li>• メッセージを破棄します。</li> </ul>
3.	コネクション確立後のコネクション障害	コネクション障害	<ul style="list-style-type: none"> <li>• UA 障害を処理します。</li> <li>• CERREVT を起動します。</li> <li>• コネクションを解放します。</li> </ul> 解放後は運用コマンド（mcfactcn）で再確立できます。



障害発生箇所	内容	メッセージログの出力内容	TP1/NET/User Agent の処理
4.	コネクション解放中のコネクション障害	コネクション障害	<ul style="list-style-type: none"><li>• CERREVT を起動します。</li><li>• コネクションを解放します。 解放後は運用コマンド ( mcfactcn ) で再確立できます。</li></ul>
5.	入力キュー書き込みエラー ( CCLSEVT )	メッセージ入力障害	<ul style="list-style-type: none"><li>• メッセージを破棄します。</li></ul>

## 8.3 問い合わせメッセージ，応答メッセージ障害

### 8.3.1 問い合わせメッセージ送信時の障害

TP1/NET/User Agent からの問い合わせメッセージ，およびそれに対する相手システムからの応答メッセージ処理中に，障害の発生する個所の区分を次の図に示します。また，障害発生個所に応じた TP1/NET/User Agent の障害処理について，表 8-14 に示します。

図 8-2 問い合わせメッセージ送信時，応答メッセージ受信時の障害の発生個所

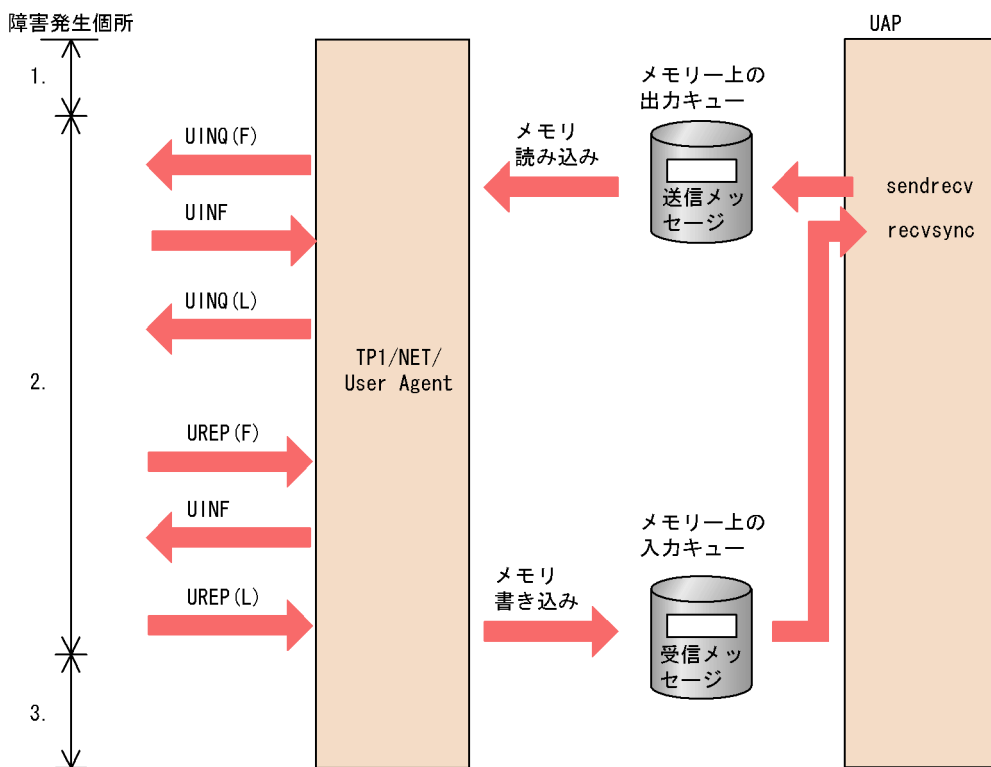


表 8-14 問い合わせメッセージ送信時，応答メッセージ受信時の障害の発生個所に応じた障害処理

障害発生個所	内容	メッセージログの出力内容	TP1/NET/User Agent の処理	メッセージの扱い	UAP のできる処理
1.	メモリ読み込みエラー	メッセージ出力障害	UAP ヘエラーリターンします。	送信メッセージを破棄します。	UAP ヘリターン後に再送できます。または障害処理ができます。
2.	問い合わせ応答時の UA 障害 (UA 障害通知, コネクション障害通知)	UA 障害 (コネクション障害時を除く)	<ul style="list-style-type: none"> <li>UAP ヘエラーリターンします。</li> <li>CERREVT を起動します。</li> <li>論理端末を閉塞します。</li> </ul>	送信メッセージを破棄します。	<ul style="list-style-type: none"> <li>CERREVT 処理</li> <li>UAP ヘリターン後に再送できます。または障害処理ができます。</li> </ul>
	受信打ち切り受信時	送信メッセージの受信打ち切り	<ul style="list-style-type: none"> <li>ERREVT を起動します。</li> <li>論理端末の閉塞処理はしません。</li> </ul>	送信メッセージを破棄します。	ERREVT 処理
	受信拒否受信時	送信メッセージの受信拒否	論理端末を閉塞します。	相手局からの受信拒否解除受信後，メッセージを再送します。	ありません。
3.	応答エラー	メッセージ入力障害	<ul style="list-style-type: none"> <li>UAP ヘエラーリターンします。</li> <li>論理端末を閉塞します。</li> </ul>	受信メッセージを破棄します。	UAP ヘリターン後に再送できます。または障害処理ができます。

### 8.3.2 応答メッセージ送信時の障害

相手システムからの問い合わせメッセージ，および TP1/NET/User Agent からの応答メッセージ処理中に，障害の発生する個所の区分を次の図に示します。また，障害発生個所に応じた TP1/NET/User Agent の障害処理について，表 8-15 に示します。

8. 障害対策

図 8-3 問い合わせメッセージ受信時，応答メッセージ送信時の障害の発生個所

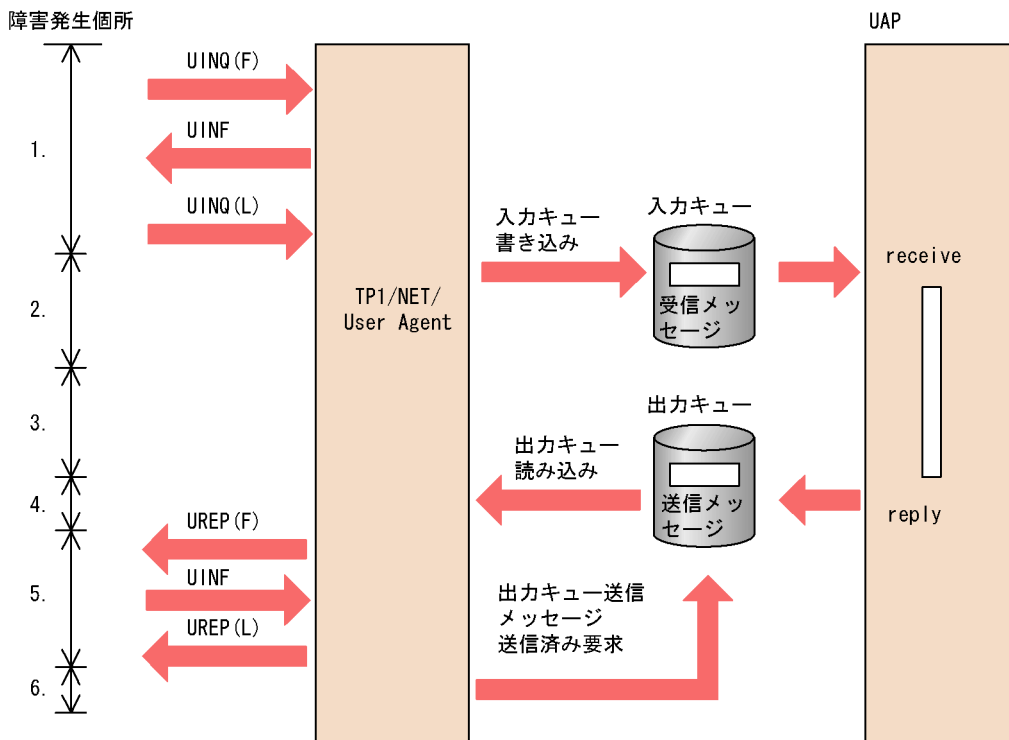


表 8-15 問い合わせメッセージ受信時，応答メッセージ送信時の障害の発生個所に応じた障害処理

障害発生個所	内容	メッセージログの出力内容	TP1/NET/User Agent の処理	メッセージの扱い	UAP ができる処理
1.	問い合わせメッセージ受信時 UA 障害 (コネクション障害 (UA 障害通知またはコネクション障害通知))	UA 障害 (コネクション障害時を除く)	<ul style="list-style-type: none"> <li>• CERREVT を起動します。</li> <li>• 論理端末を閉塞します。</li> </ul>	該当しません。	CERREVT 処理
	送信中断受信時	相手局がメッセージの送信を中断	論理端末の閉塞処理はしません。	受信メッセージを破棄します。	ありません。
2.	入力キュー書き込みエラー	メッセージ入力障害	<ul style="list-style-type: none"> <li>• ERREVT1 または ERREVT2 を起動します。</li> <li>• 論理端末の閉塞処理はしません。</li> </ul>	ERREVT1, または ERREVT2 で受信メッセージの障害を通知します。	ERREVT1, または ERREVT2 で応答メッセージを送信できるようになります。

障害発生箇所	内容	メッセージログの出力内容	TP1/NET/User Agent の処理	メッセージの扱い	UAP でできる処理
3.	UAP 異常終了	ありません。	ERREVT3 を起動します。	該当しません。	ERREVT3 で応答メッセージを送信できるようになります。
4.	出力キュー読み込みエラー	メッセージ出力障害	<ul style="list-style-type: none"> <li>• CERREVT を起動します。</li> <li>• 論理端末を閉塞します。</li> </ul>	送信済みとします。	CERREVT 処理
5.	応答メッセージ送信時 UA 障害 (UA 障害通知, コネクション障害通知)	UA 障害 (コネクション障害時を除く)	<ul style="list-style-type: none"> <li>• CERREVT を起動します。</li> <li>• 論理端末を閉塞します。</li> </ul>	送信済みとします。	CERREVT 処理
6.	出力キュー送信メッセージ送信済み要求エラー	メッセージ送信後処理障害	処理を続けます。	送信済みとします。	ありません。

## 8.4 一方送信メッセージ障害

TP1/NET/User Agent からの一方送信メッセージ，および相手システムからの一方送信メッセージの処理中に，障害の発生する個所の区分を次の図に示します。また，障害発生個所に応じた TP1/NET/User Agent の障害処理について，表 8-16 に示します。

図 8-4 一方送信メッセージ障害の発生個所

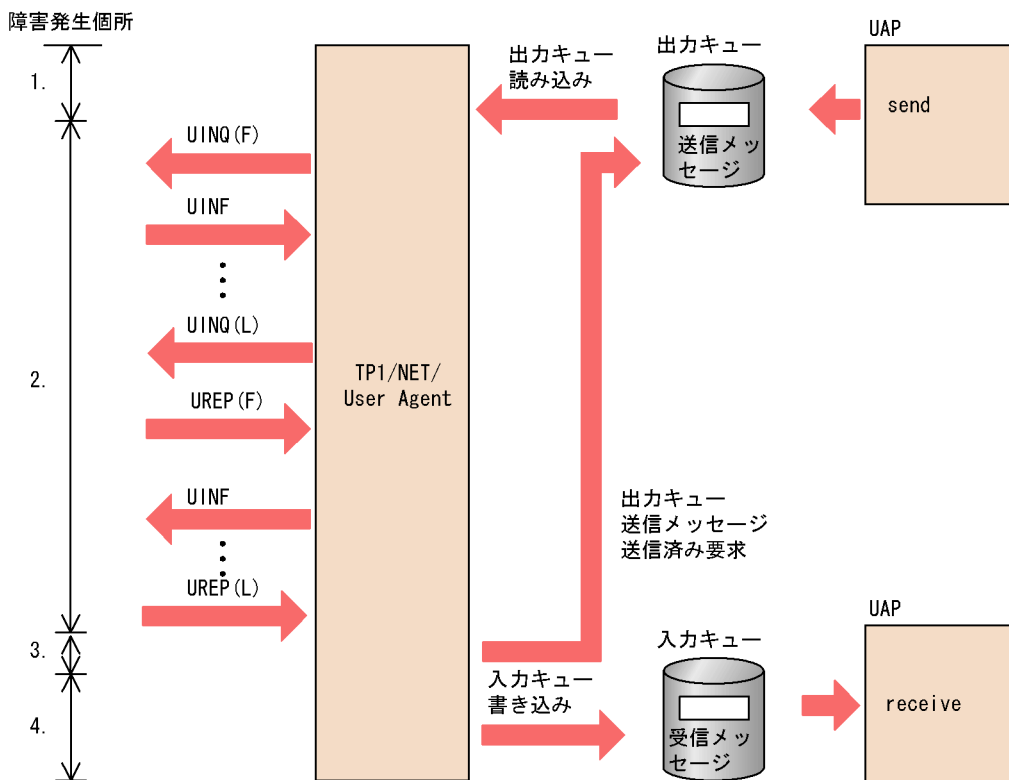


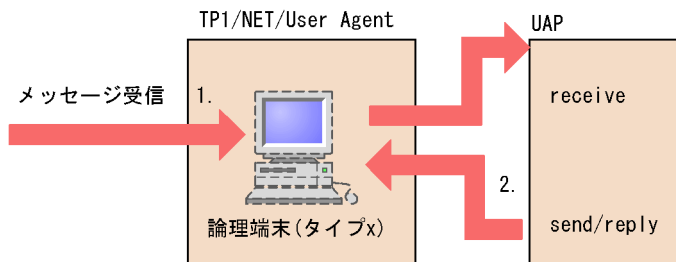
表 8-16 一方送信メッセージ障害の発生個所に応じた障害処理

障害発生個所	内容	メッセージログの出力内容	TP1/NET/User Agent の処理	メッセージの扱い	UAP ができる処理
1.	出力キュー読み込みエラー	メッセージ出力障害	次の問い合わせメッセージの送信処理をします。	送信済みとします。	ありません。
2.	一方送信メッセージ送受信時 UA 障害 (UA 障害通知, コネクション障害通知)	UA 障害 (コネクション障害時を除く)	<ul style="list-style-type: none"> <li>• CERREVT を起動します。</li> <li>• 論理端末を閉塞します。</li> </ul>	論理端末の閉塞を解除したあと, メッセージを再送します。	CERREVT 処理
	受信打ち切り受信時	送信メッセージの受信打ち切り	<ul style="list-style-type: none"> <li>• ERREVT A を起動します。</li> </ul>	送信メッセージを破棄します。	ERREVT A 処理
	受信拒否受信時	送信メッセージの受信拒否	論理端末を閉塞します。	相手局からの受信拒否解除受信後, メッセージを再送します。	ありません。
3.	出力キュー送信メッセージ送信済み要求エラー	メッセージ送信後処理障害	処理を続けます。	送信済みとします。	ありません。
4.	入力キュー書き込みエラー	メッセージ入力障害	<ul style="list-style-type: none"> <li>• ERREVT1 または ERREVT2 を起動します。</li> <li>• CERREVT を起動します。</li> <li>• 論理端末を閉塞します。</li> </ul>	ERREVT1, または ERREVT2 で受信メッセージを通知します。	<ul style="list-style-type: none"> <li>• ERREVT1 処理または ERREVT2 処理</li> <li>• CERREVT 処理</li> </ul>

## 8.5 論理端末の端末タイプ不正

論理端末ごとに端末タイプを定義できるため、次の図の 1. と 2. に示す個所で論理端末の端末タイプ不正が発生することがあります。

図 8-5 論理端末の端末タイプ不正



- (凡例) 1. : 受信メッセージ不正  
2. : 送信先論理端末の端末タイプ不正

### (1) 受信メッセージ不正

図 8-5 の 1. の個所では、論理端末で受信できるメッセージと、実際に受信したメッセージが整合しない場合に、受信メッセージ不正が発生します。

例えば、request 型の論理端末で問い合わせメッセージ (UINQ) を受信した場合、受信メッセージが該当する論理端末でサービスできないときは、UERR を送信し、論理端末を閉塞します。

### (2) 送信先論理端末の端末タイプ不正

図 8-5 の 2. の個所では、送信先の論理端末の端末タイプと送信種別が整合しない場合に、送信先論理端末の端末タイプ不正が発生します。

例えば、reply 型の論理端末に対する一方送信要求では、送信先論理端末の端末タイプ不正の発生時、UAP にエラーリターンします。



# 付録

---

付録 A メッセージ送受信の処理の流れ

---

付録 B 障害発生時の処理の流れ

---

付録 C 理由コード一覧

---

付録 D 用語解説

---

## 付録 A メッセージ送受信の処理の流れ

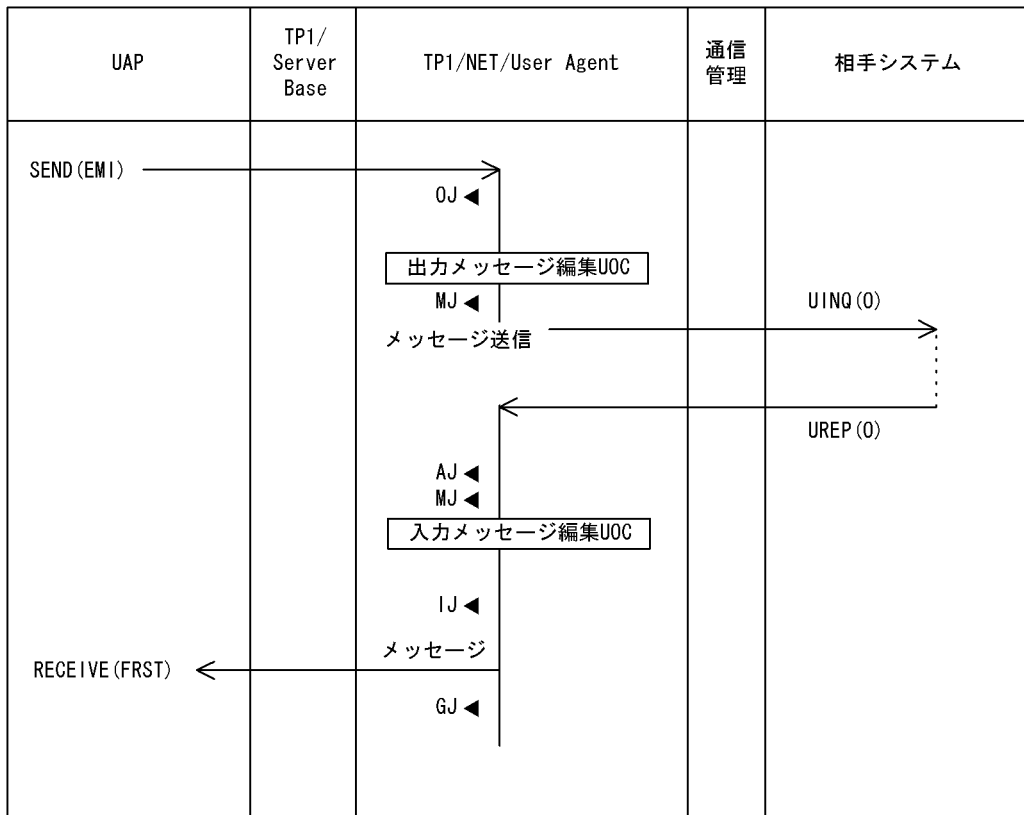
メッセージを送受信するときのデータの流れ，ジャーナルの取得タイミングなどを，メッセージの種類ごとに示します。

### 付録 A.1 一方送信メッセージ

(1) 一方送信メッセージ送信時の処理の流れ (request 型論理端末)

(a) UA サービスデータ単位分割なしの場合

図 A-1 request 型論理端末からの一方送信メッセージの処理の流れ (UA サービスデータ単位分割なしの場合)

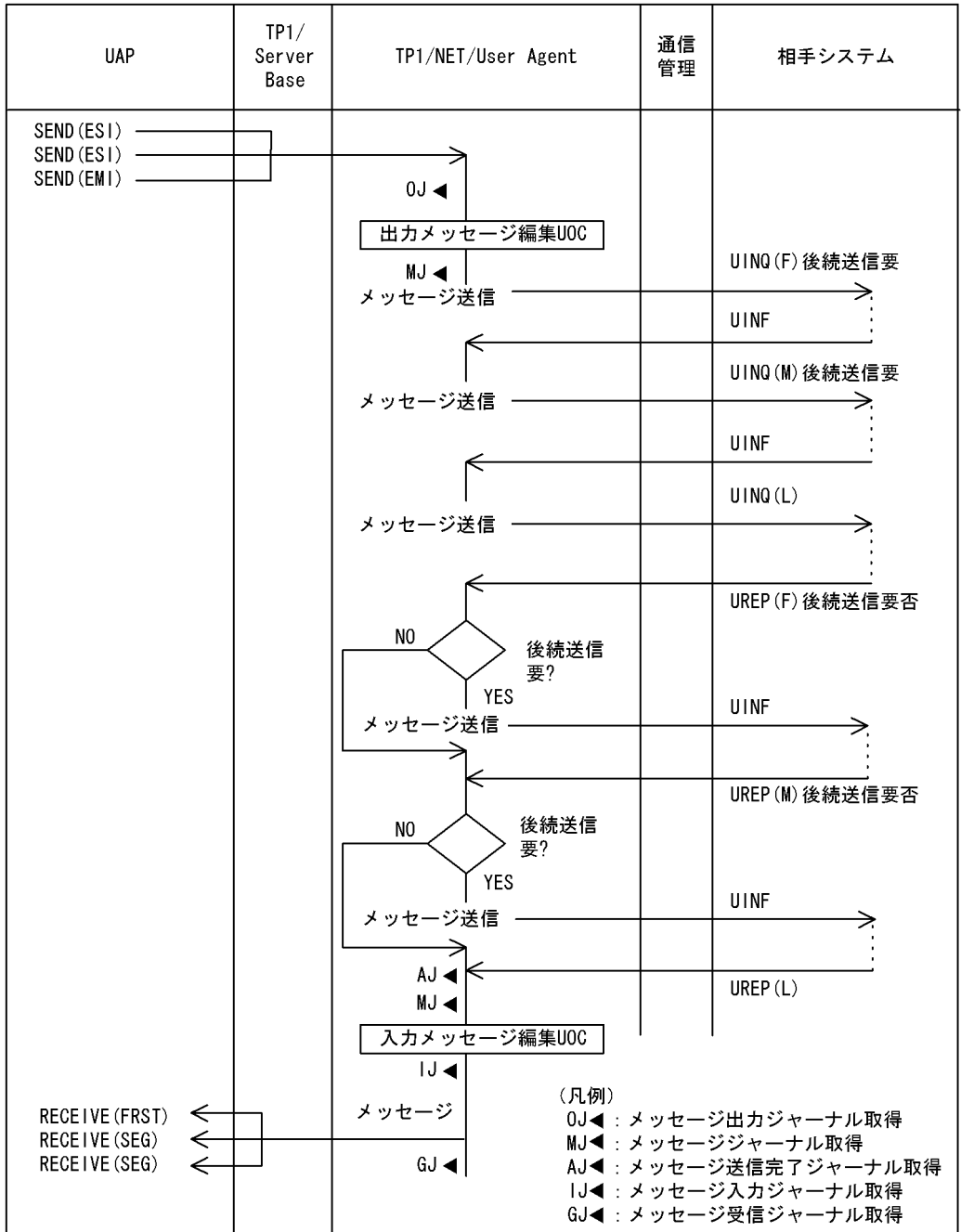


(凡例)

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

(b) UA サービスデータ単位分割ありの場合

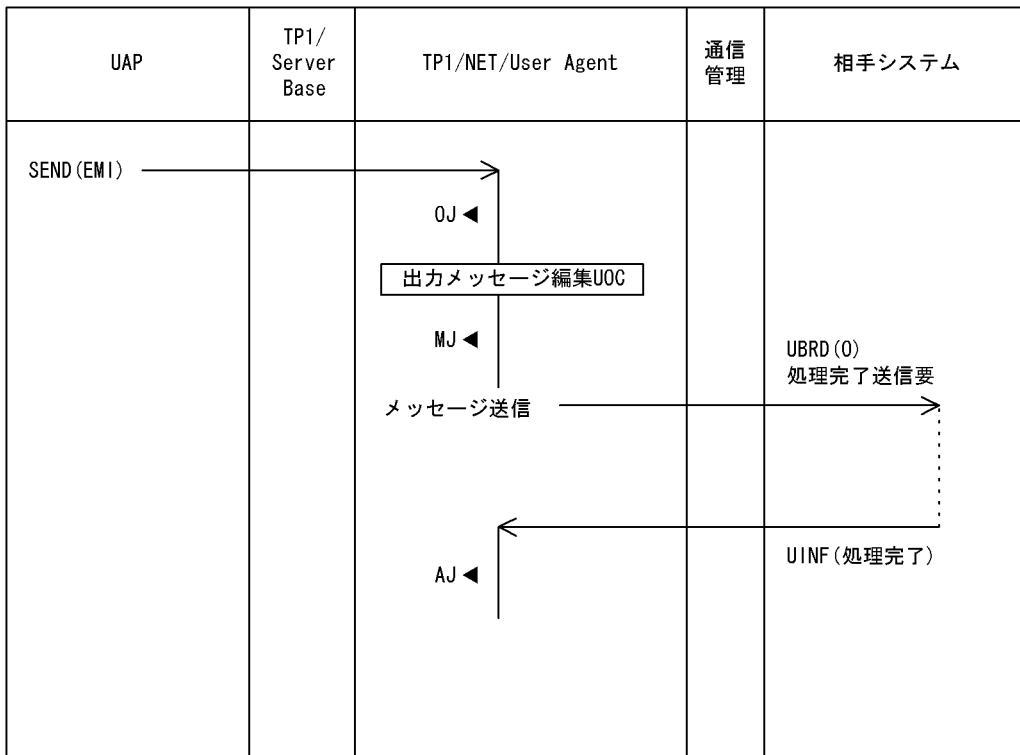
図 A-2 request 型論理端末から的一方送信メッセージの処理の流れ (UA サービスデータ単位分割ありの場合)



(2) 一方送信メッセージ送信時の処理の流れ (send 型論理端末)

(a) UA サービスデータ単位分割なしの場合

図 A-3 send 型論理端末からの一方送信メッセージの処理の流れ (UA サービスデータ単位分割なしの場合)

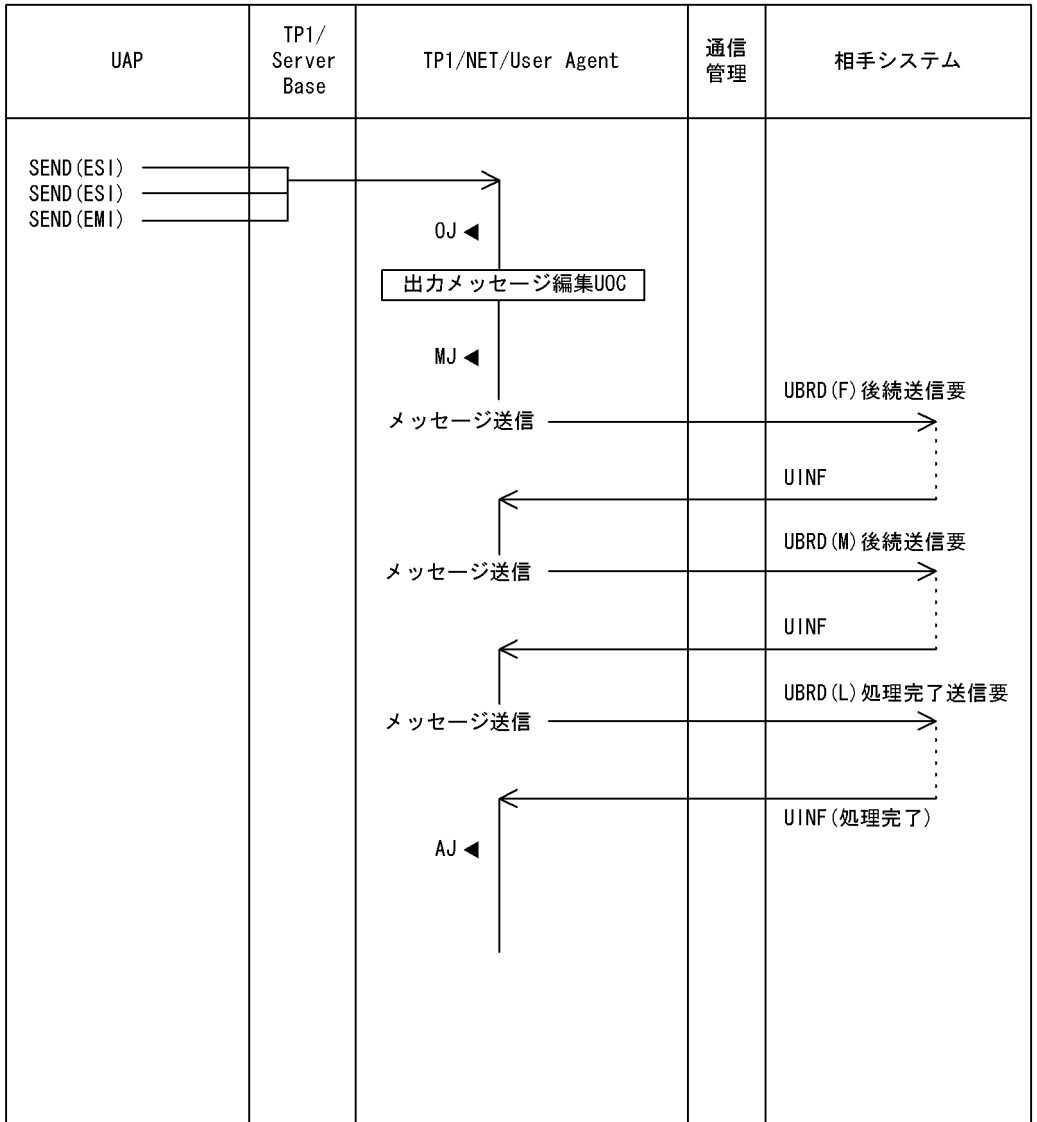


(凡例)

- OJ ◀: メッセージ出カジャーナル取得
- MJ ◀: メッセージジャーナル取得
- AJ ◀: メッセージ送信完了ジャーナル取得

(b) UA サービスデータ単位分割ありの場合

図 A-4 send 型論理端末から的一方送信メッセージの処理の流れ (UA サービスデータ単位分割ありの場合)



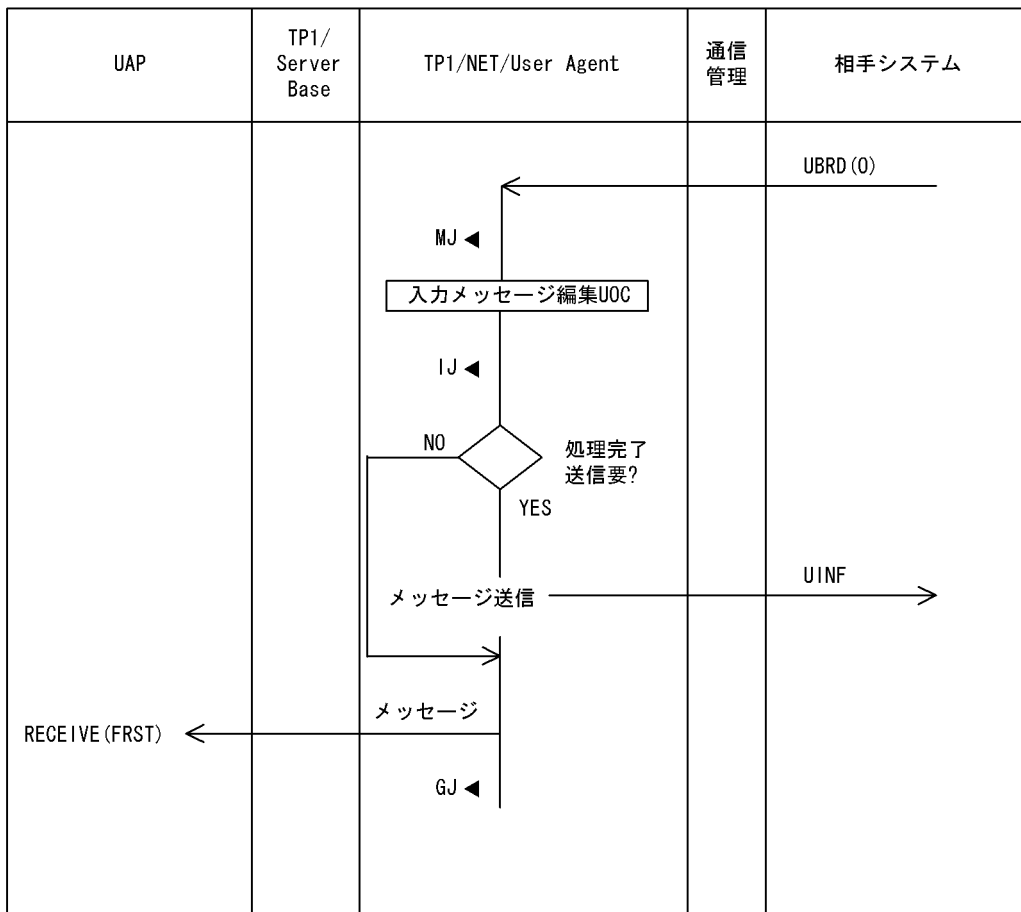
(凡例)

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

(3) 一方送信メッセージ受信時の処理の流れ

(a) UA サービスデータ単位分割なしの場合

図 A-5 receive 型論理端末への一方送信メッセージの処理の流れ (UA サービスデータ単位分割なしの場合)

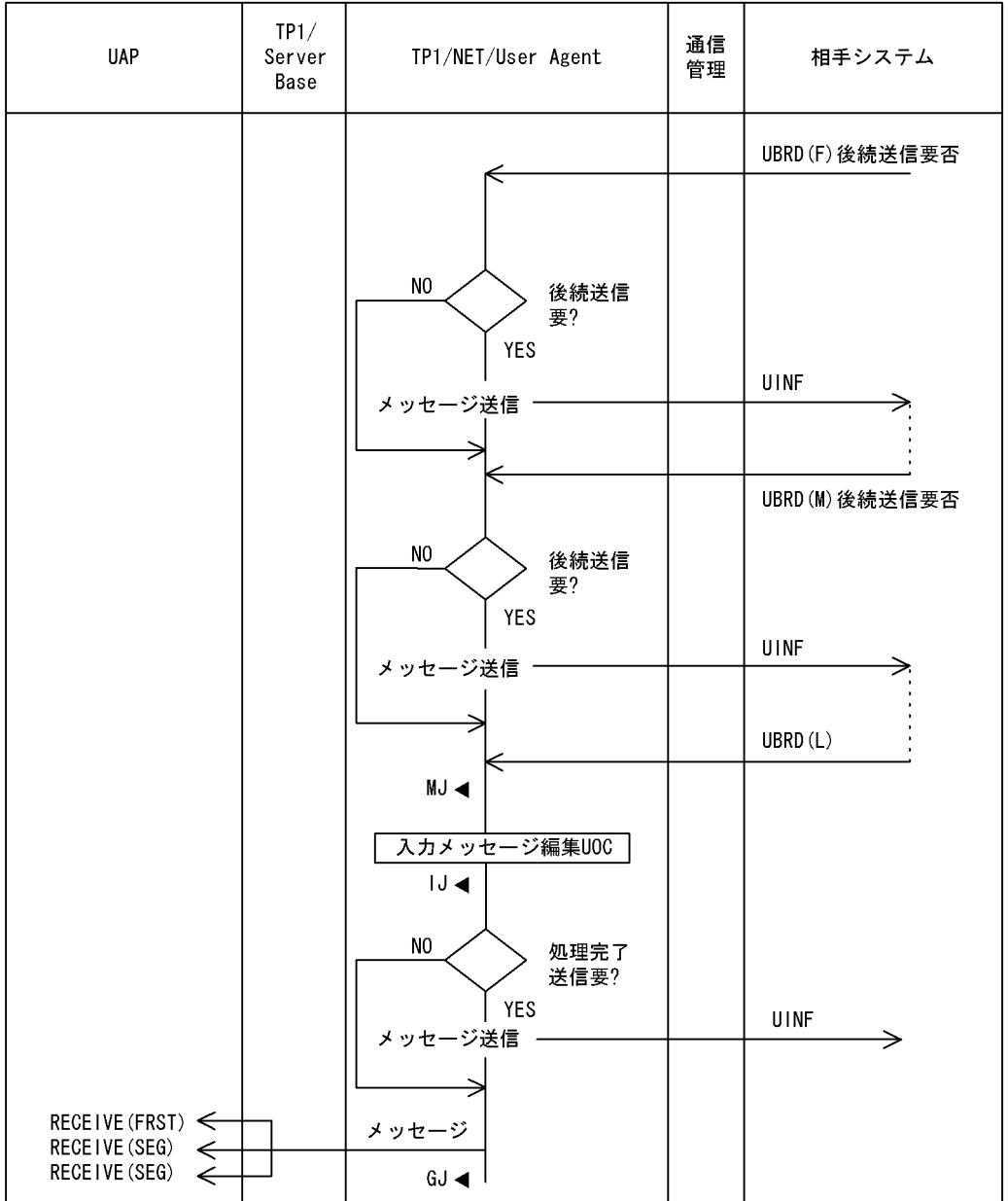


(凡例)

- 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

(b) UA サービスデータ単位分割ありの場合

図 A-6 receive 型論理端末へ的一方送信メッセージの処理の流れ (UA サービスデータ単位分割ありの場合)



(凡例)

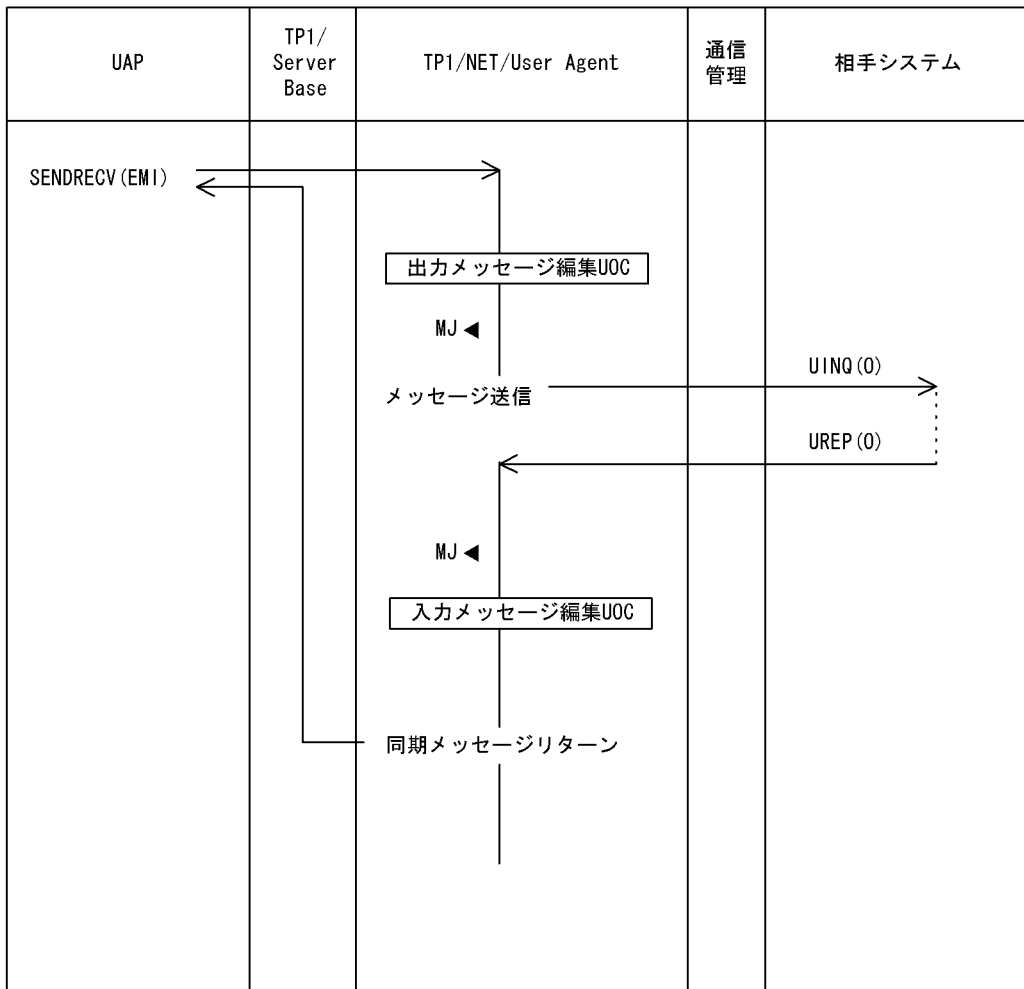
- MJ ◀: メッセージジャーナル取得
- IJ ◀: メッセージ入力ジャーナル取得
- GJ ◀: メッセージ受信ジャーナル取得

## 付録 A.2 問い合わせメッセージ，および応答メッセージ

### (1) 問い合わせメッセージ送信時の処理の流れ

#### (a) UA サービスデータ単位分割なしの場合

図 A-7 問い合わせメッセージ送信時の処理の流れ (UA サービスデータ単位分割なしの場合)



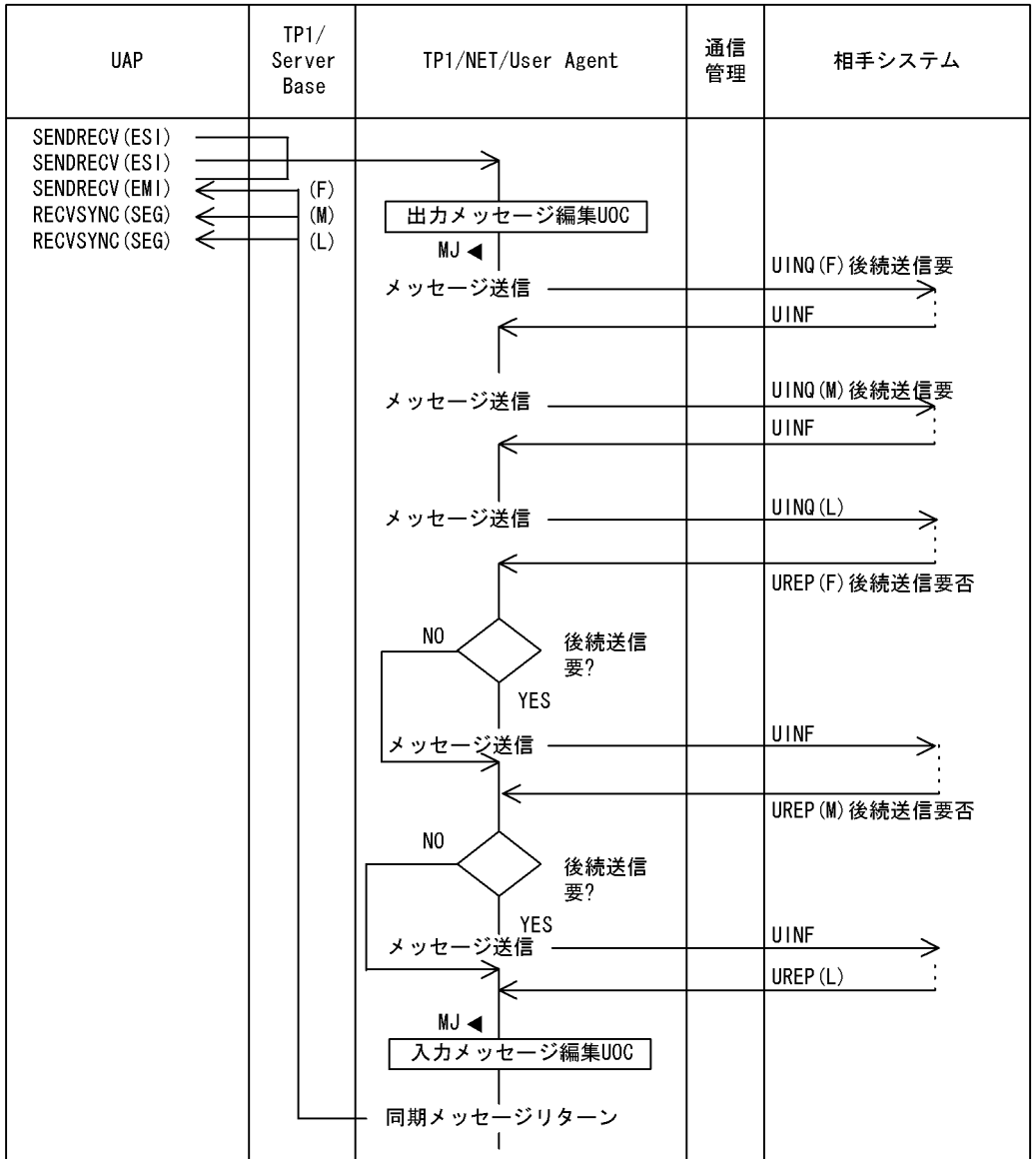
(凡例)

MJ ◀ : メッセージジャーナル取得



(b) UA サービスデータ単位分割ありの場合

図 A-8 問い合わせメッセージ送信時の処理の流れ (UA サービスデータ単位分割ありの場合)



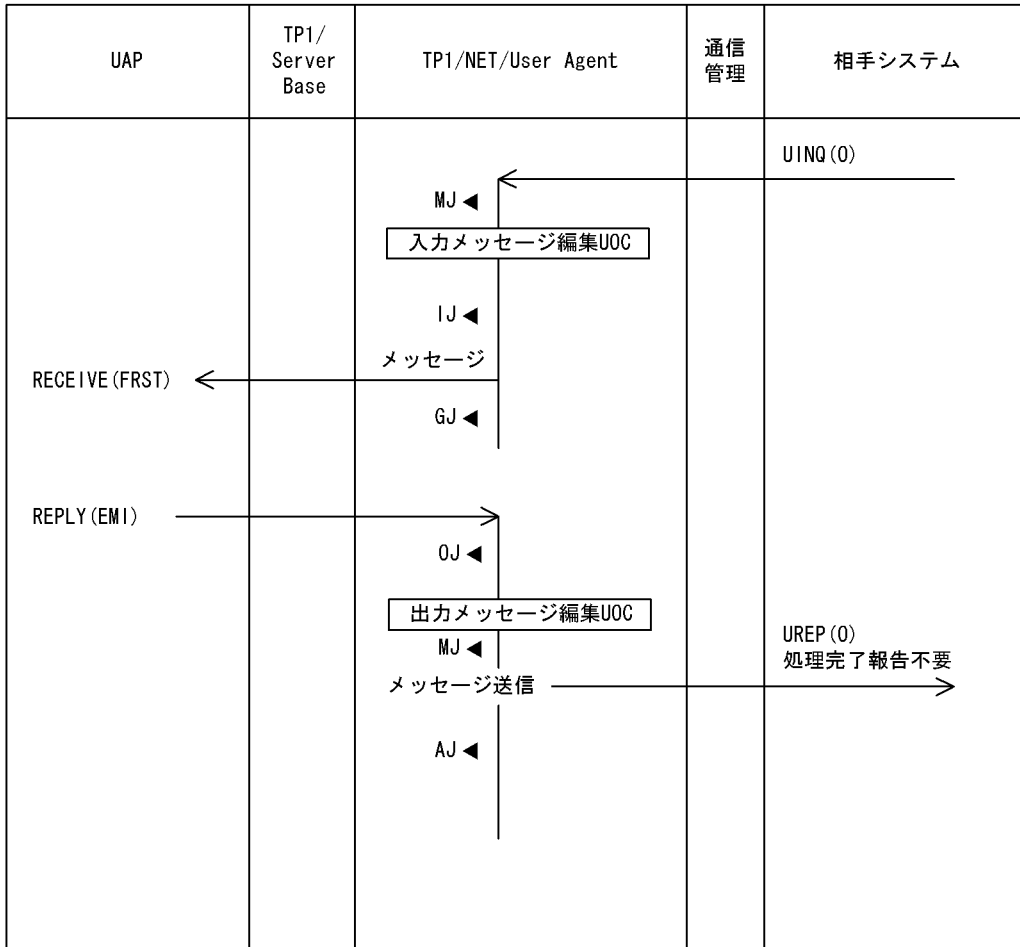
(凡例)

MJ ◀: メッセージジャーナル取得

(2) 応答メッセージ送信時の処理の流れ

(a) UA サービスデータ単位分割なしの場合

図 A-9 応答メッセージ送信時の処理の流れ (UA サービスデータ単位分割なしの場合)

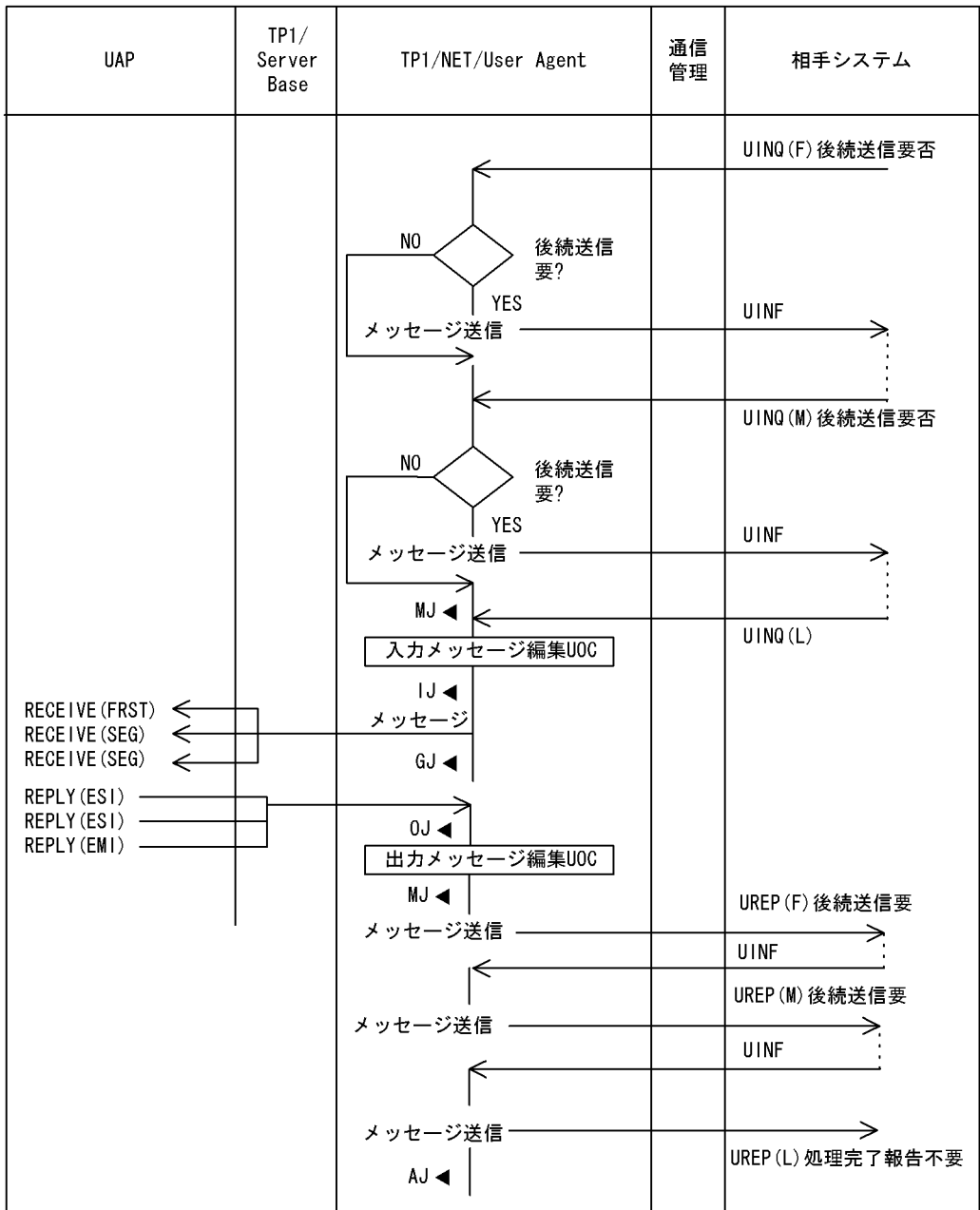


(凡例)

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

(b) UA サービスデータ単位分割ありの場合

図 A-10 応答メッセージ送信時の処理の流れ (UA サービスデータ単位分割ありの場合)



(凡例)

- OJ ◀ : メッセージ出カジャーナル取得
- MJ ◀ : メッセージジャーナル取得
- AJ ◀ : メッセージ送信完了ジャーナル取得
- IJ ◀ : メッセージ入カジャーナル取得
- GJ ◀ : メッセージ受信ジャーナル取得

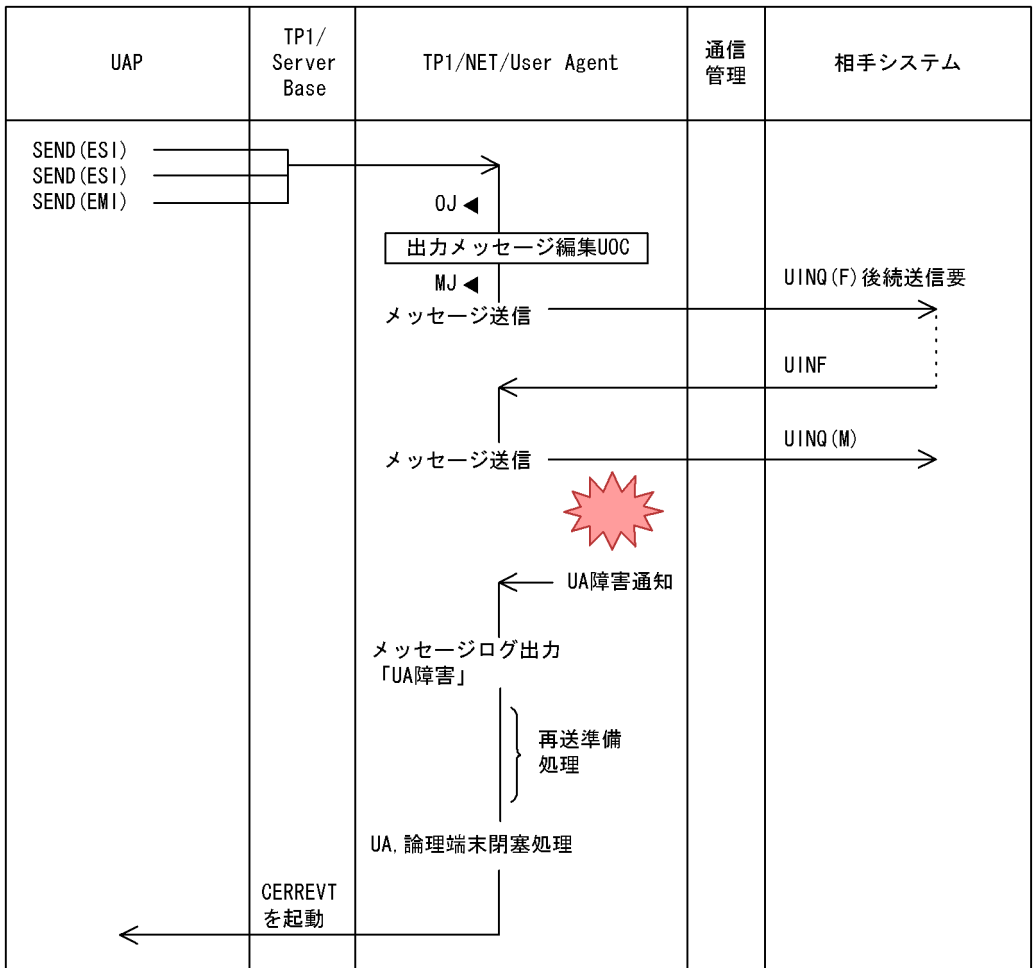
## 付録 B 障害発生時の処理の流れ

メッセージを送受信するときに発生した障害処理の流れについて示します。

### 付録 B.1 一方送信メッセージ障害

#### (1) UINQ 送信時の UA 障害

図 B-1 UINQ 送信時の UA 障害の処理の流れ (一方送信メッセージ障害の場合)



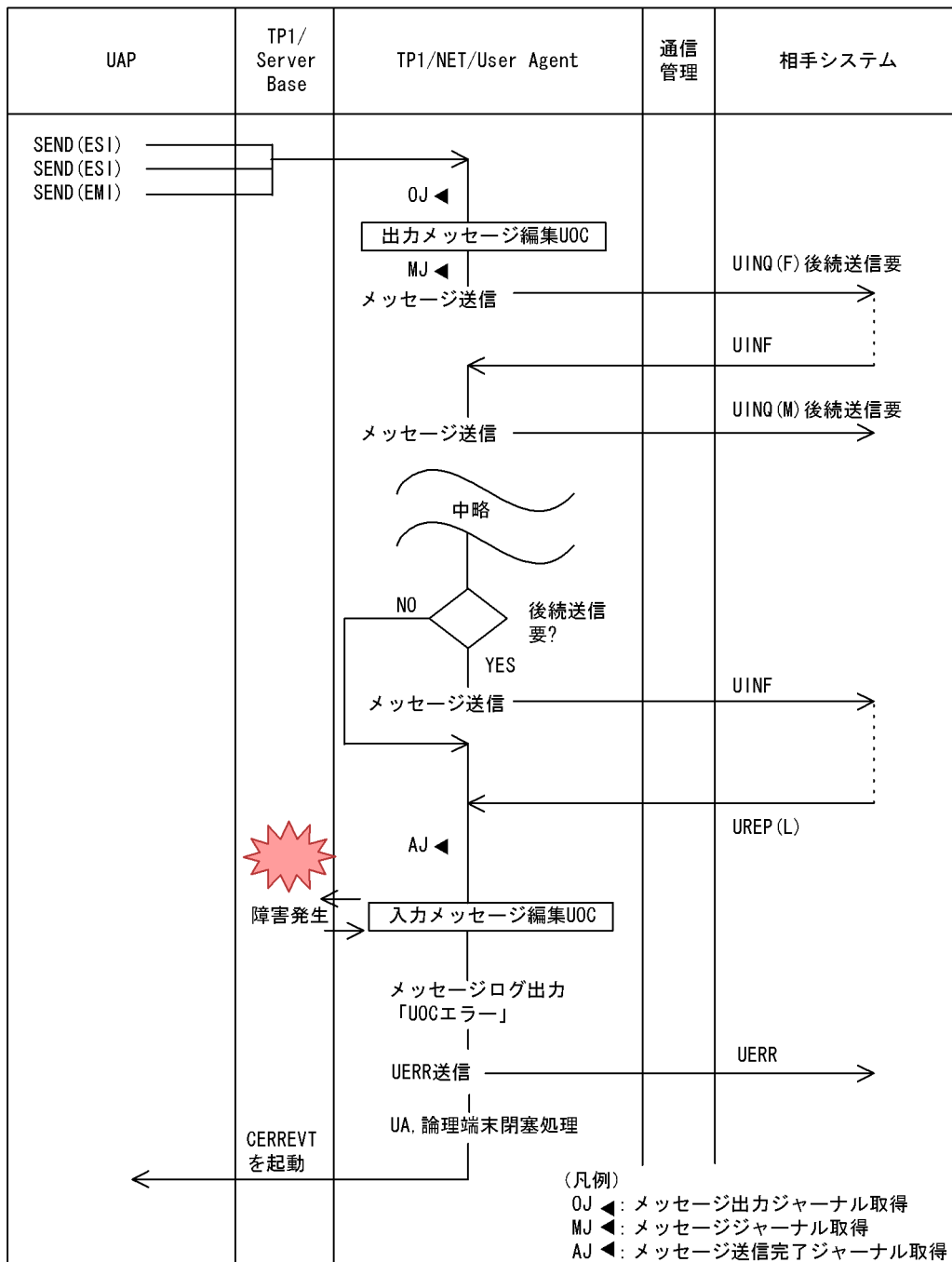
(凡例)

OJ ◀: メッセージ出力ジャーナル取得

MJ ◀: メッセージジャーナル取得

(2) 入力メッセージ編集 UOC エラー

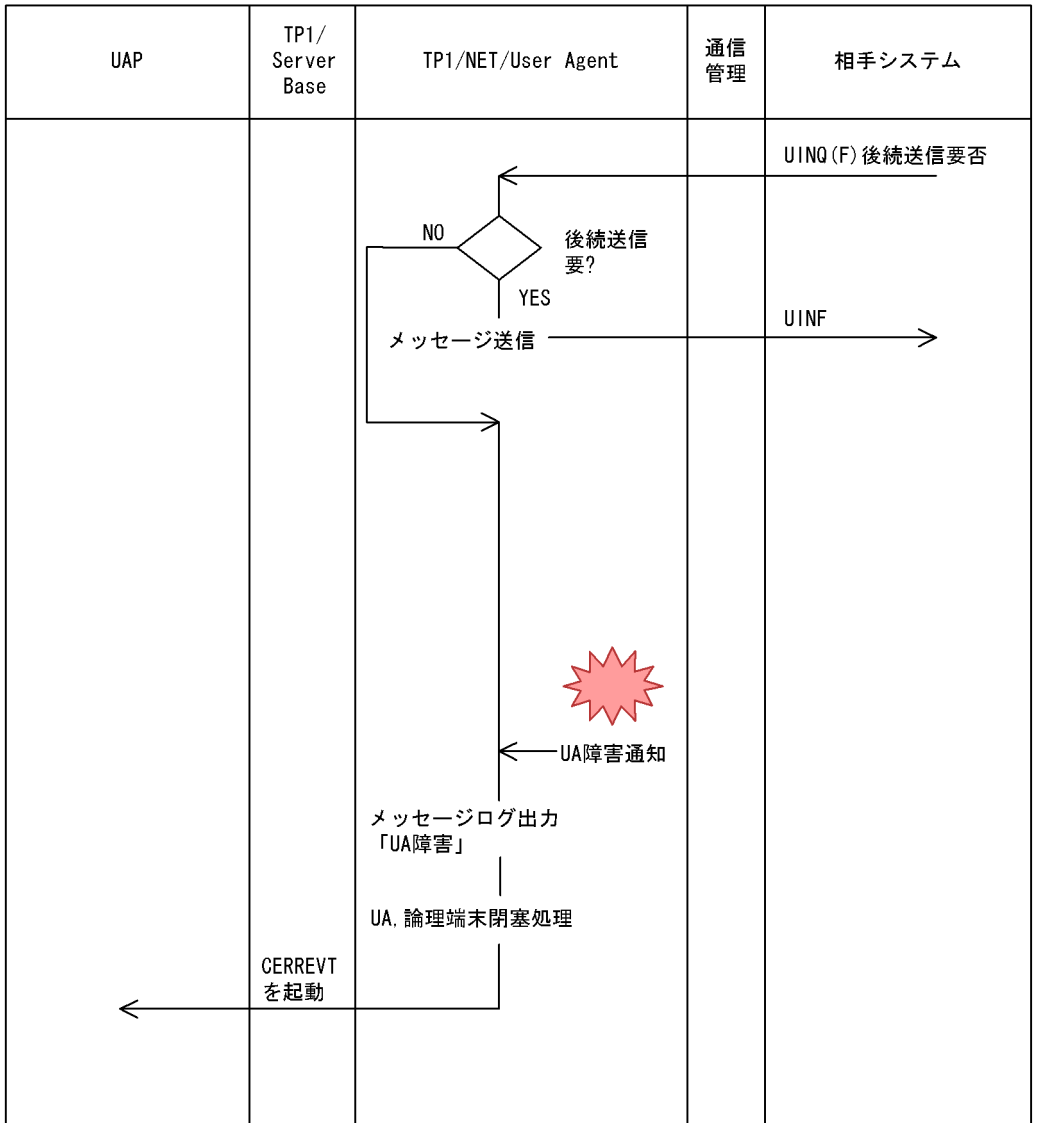
図 B-2 入力メッセージ編集 UOC エラーの処理の流れ (一方送信メッセージ障害の場合)



## 付録 B.2 応答メッセージ障害

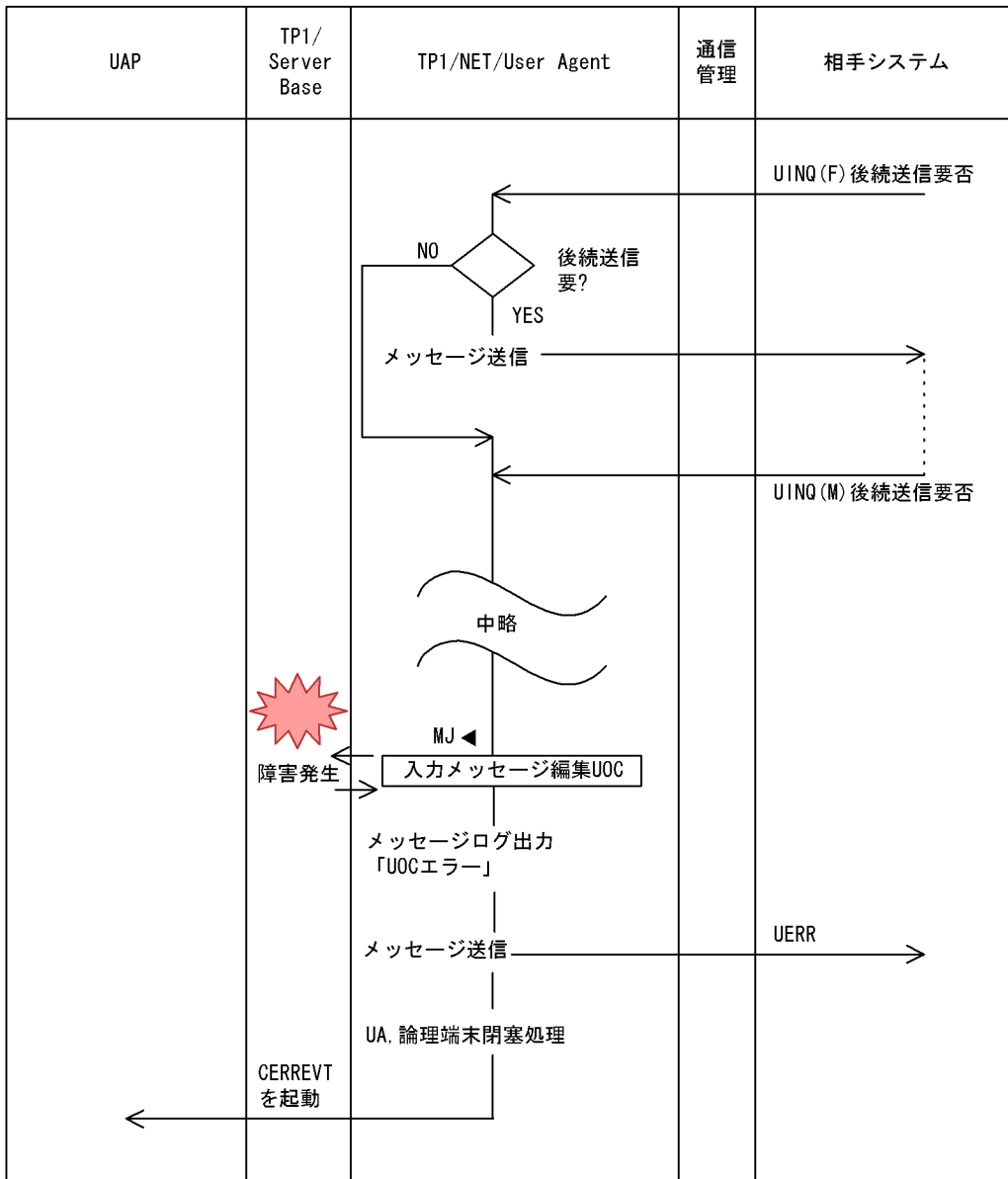
### (1) UINQ 受信時の UA 障害

図 B-3 UINQ 受信時 UA 障害の処理の流れ ( 応答メッセージ障害の場合 )



(2) 入力メッセージ編集 UOC エラー

図 B-4 入力メッセージ編集 UOC エラーの処理の流れ ( 応答メッセージ障害の場合 )



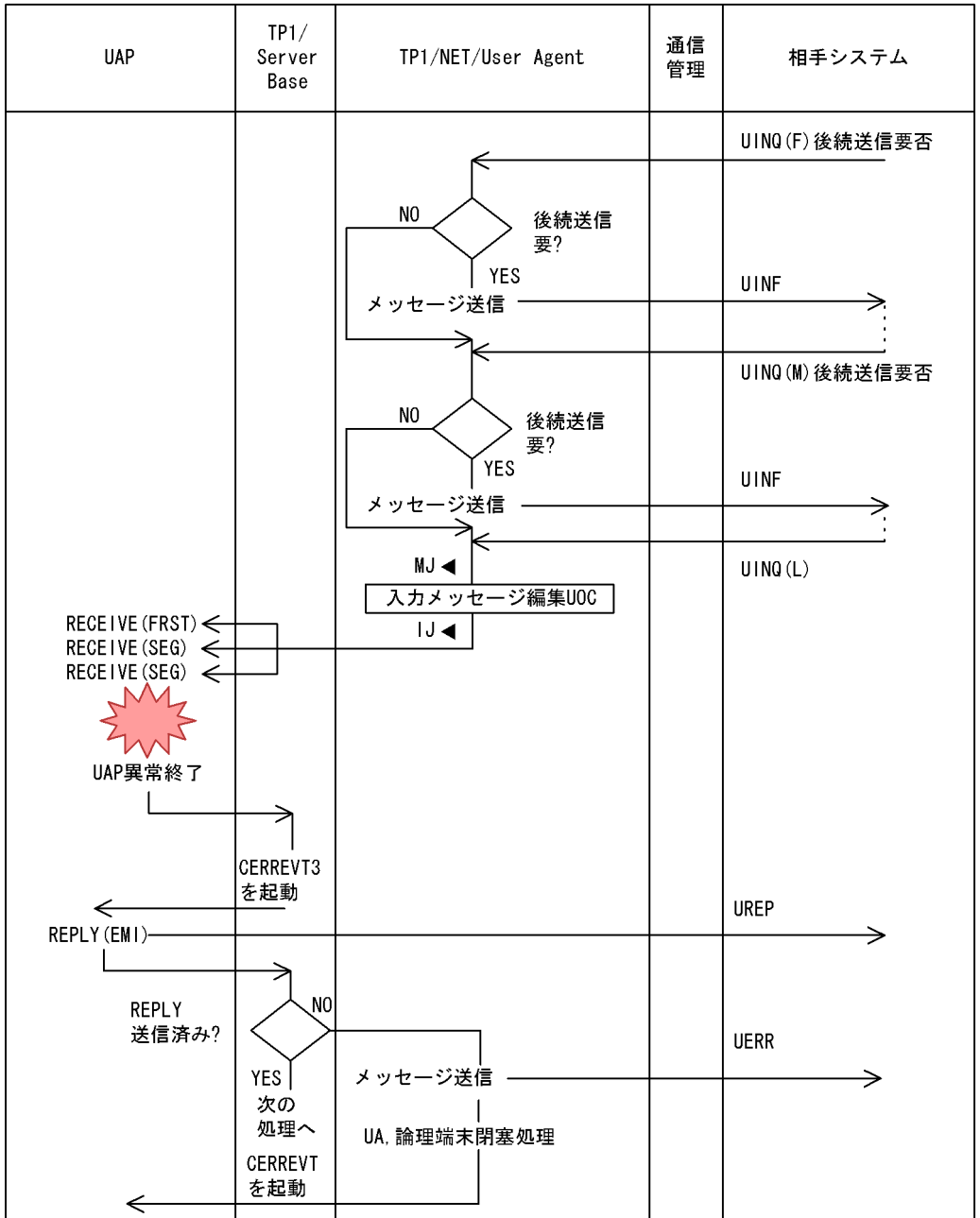
(凡例)

MJ ◀: メッセージジャーナル取得



(3) UAP 異常終了

図 B-5 UAP 異常終了の処理の流れ (応答メッセージ障害の場合)



(凡例)

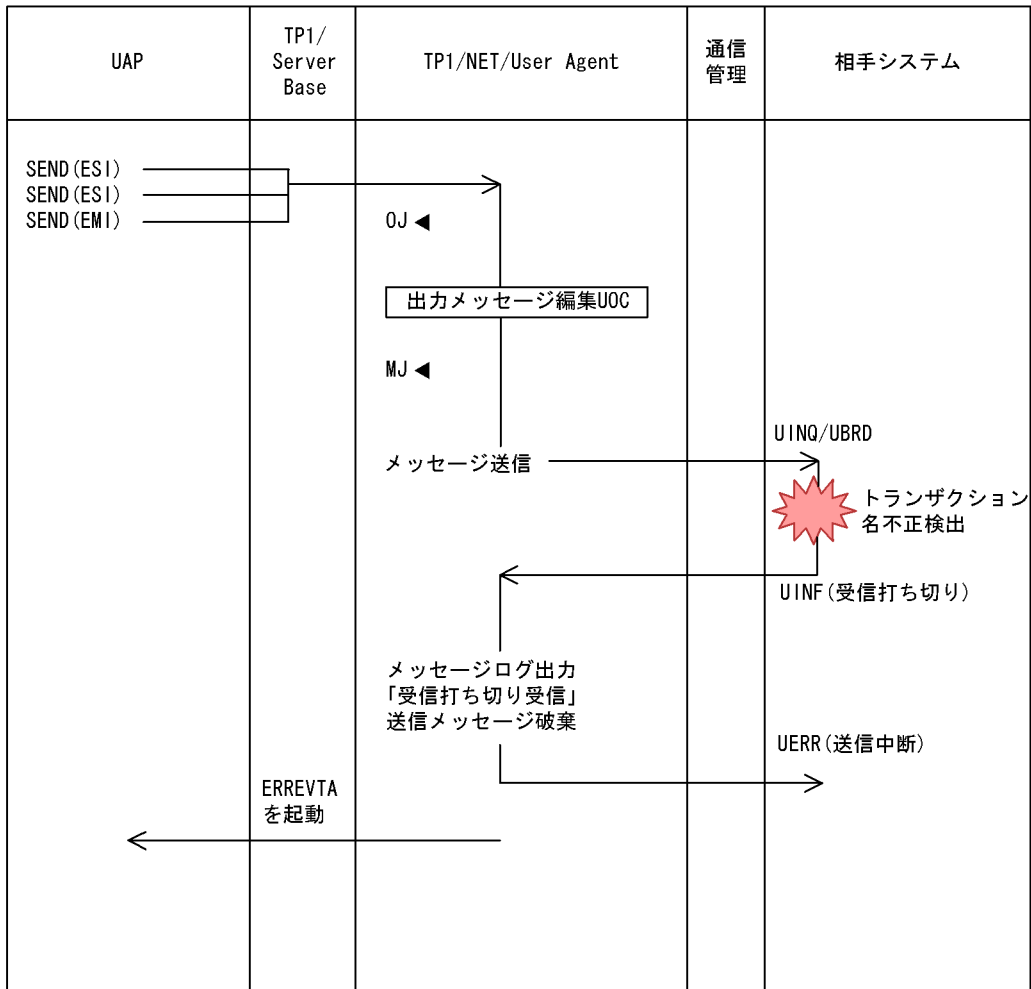
MJ ◀: メッセージジャーナル取得

IJ ◀: メッセージ入力ジャーナル取得

### 付録 B.3 相手システム (XDM/DCCM3) 接続時の障害

#### (1) 受信打ち切りを受信

図 B-6 受信打ち切り受信時の処理の流れ (相手システム接続時の障害の場合)

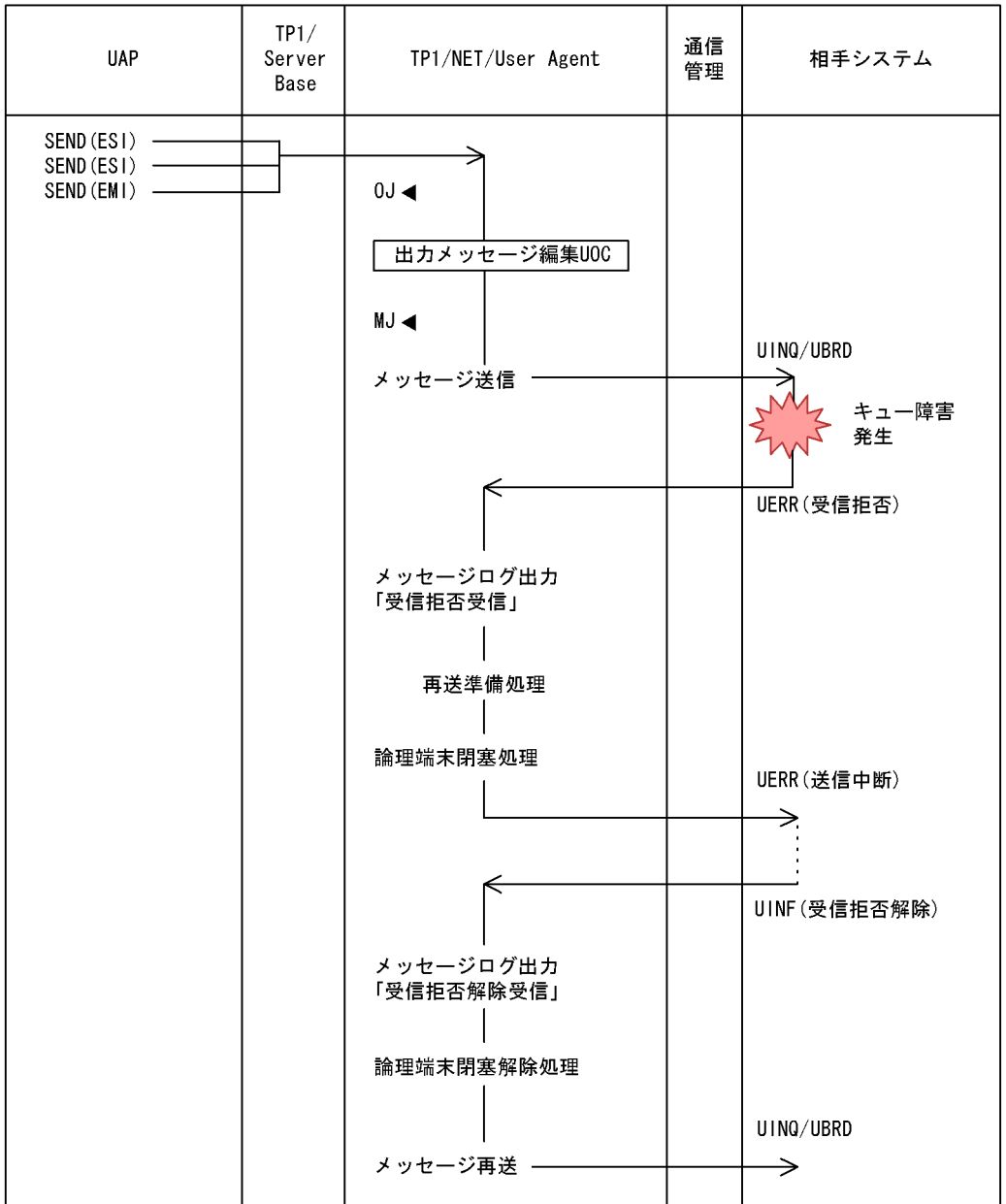


(凡例)

- 0J ◀: メッセージ出カジャーナル取得
- MJ ◀: メッセージジャーナル取得

(2) 受信拒否・受信拒否解除を受信

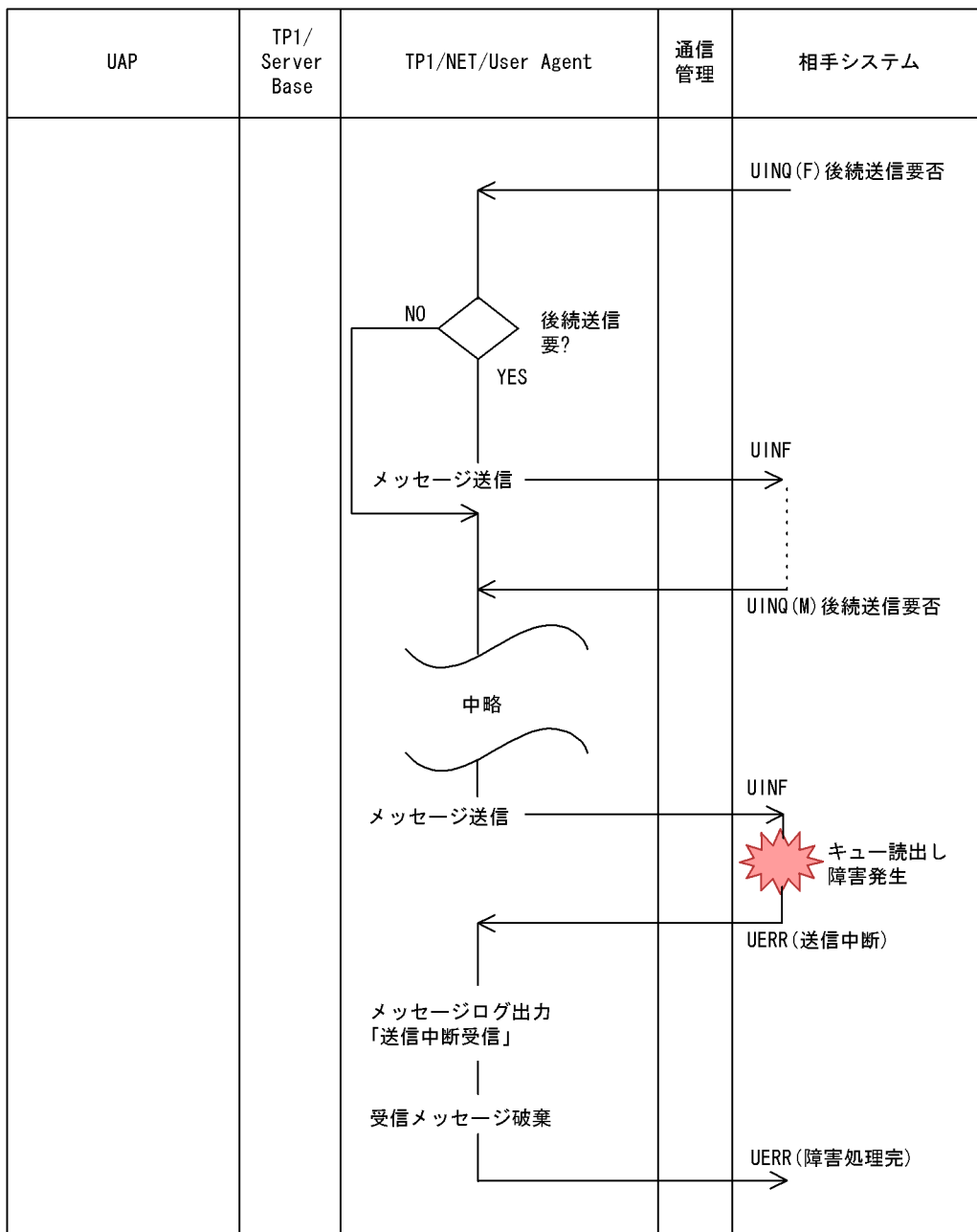
図 B-7 受信拒否・受信拒否解除を受信時の処理の流れ (相手システム接続時の障害の場合)



(凡例)  
 OJ ◀ : メッセージ出力ジャーナル取得  
 MJ ◀ : メッセージジャーナル取得

(3) 送信中断を受信

図 B-8 送信中断受信時の処理の流れ (相手システム接続時の障害の場合)



## 付録 C 理由コード一覧

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

表 C-1 ERREVT2 の理由コード一覧

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

表 C-2 CERREVT の理由コード一覧

理由コード 1 (16 進数)	理由コード 2 (16 進数)	発生条件	発生箇所
DCMOUM_RSN1_MCF (00000001)	DCMOUM_RSN2_ITQ (00000000)	メッセージ入力障害	UA, 論理端末
	DCMOUM_RSN2_APL (00000001)	アプリケーション名取得障害	UA, 論理端末
	DCMOUM_RSN2_OTGET (00000002)	メッセージ出力障害	UA, 論理端末

理由コード 1 (16進数)	理由コード 2 (16進数)	発生条件	発生箇所
	DCMOUM_RSN2_OTC MP (00000003)	メッセージ送信完了処理 障害	UA, 論理端 末
	DCMOUM_RSN2_DCTL E (00000004)	mcftdctle による UA, 論理端末の閉塞	UA, 論理端 末
	DCMOUM_RSN2_UAPA B (00000005)	reply 型 UAP 異常終了 による UERR 送信	UA, 論理端 末
	DCMOUM_RSN2_SYCE R (00000006)	UAP への同期リターン 失敗	UA, 論理端 末
	DCMOUM_RSN2_ENDE R (00000007)	終了処理中のメッセージ 受信による UERR 送信 (入力抑止)	UA, 論理端 末
	DCMOUM_RSN2_OPNR J (00000008)	UA 開局要求拒否	UA, 論理端 末
	(11ff000a)	UINQ を受信できない端 末タイプ (MCF 通信構 成定義 (mcftalcle) の -t オプションで指定) の論 理端末において UINQ を受信	論理端末
	(11ff000e)	UBRD を受信できない 端末タイプ (MCF 通信 構成定義 (mcftalcle) の -t オプションで指定) の論理端末において UBRD を受信	論理端末
	DCMOUM_RSN2_DCTC N_F (20000000)	mcftdcten -f による強制 解放	コネクショ ン
	DCMOUM_RSN2_NOB UF (20000002)	バッファ取得失敗による 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)	UA, 論理端 末, コネク ション
DCMOUM_RSN1_UERR (00000002)	RRRRDDDD (受信障害コード)	UERR 送受信, または 下位層障害	UA, 論理端 末, コネク ション
DCMOUM_RSN1_UOC (00000003)	詳細リターンコード	ユーザ (UOC) 検出障 害	論理端末, コネクショ ン

理由コード 1 (16 進数)	理由コード 2 (16 進数)	発生条件	発生個所
DCMOUM_RSN1_UAP (00000004)	RRRRDDDD (送信障害コード)	send (エラー送信) による閉局	UA, 論理端末
DCMOUM_RSN1_ACTER (00000005)	RRRRDDDD (受信障害コード)	コネクションの確立失敗 (UERR 送受信, または下位層障害)	コネクション
DCMOUM_RSN1_ACTLERJ (00000006)	RRRR0000 (拒否理由)	個別開局の拒否応答の受信	UA, 論理端末
その他	不定	上記以外の障害 (理由コード 1, 2 は保守情報)	UA, 論理端末, コネクション

## (凡例)

RRRR : 障害理由または拒否理由 (下位層障害時は 0000 を設定します。)

DDDD : OSAS/UA プロトコルで規定される障害理由詳細 (下位層障害時は 0000 を設定します。)

---

## 付録 D 用語解説

### (英字)

---

#### AJ (メッセージ送信完了ジャーナル)

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

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

#### GJ (メッセージ受信ジャーナル)

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

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

#### IJ (メッセージ入力ジャーナル)

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

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

#### MJ (メッセージジャーナル)

メッセージに関する情報です。端末名称、メッセージ種別、入力メッセージ編集前のデータ、出力メッセージ編集後のデータなどで構成されます。

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

#### OJ (メッセージ出力ジャーナル)

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

OJ の取得タイミングは UAP から send を受け付けた時です。

#### OSAS/UA プロトコル

拡張 HNA のアプリケーションプロファイルの一つとして規定した、分散処理ネットワークでのトランザクション処理を実現するためのプロトコルです。

#### UA

OSAS/UA プロトコルで指定された手順を実行するエンティティです。データの送受信をします。

UA には、一般 UA と制御 UA があります。一般 UA は、一般 UA の開局、閉局、およびデータの送受信に使用します。制御 UA は、コネクションの確立と解放に使用します。

### (力行)

---

#### コネクション

MCF の通信管理側の通信接点であり、MCF と通信管理はコネクション単位に送受信をします。



OSAS/UA プロトコルのアソシエーションに対応します。

## (ラ行)

---

### 論理端末

TP1/NET/User Agent と UAP との通信接点です。TP1/NET/User Agent と UAP は、論理端末単位にメッセージの送受信をします。



---

# 索引

## A

---

AJ (メッセージ送信完了ジャーナル) 236  
AP 間通信 2  
AP 間通信の概要 2  
AP 間通信の形態 4  
AP 間通信の仕組み 10  
AP 間通信メッセージの送受信 21

## C

---

CBLDCMCF('RECEIVE') 53  
CBLDCMCF('RECVSYNC') 58  
CBLDCMCF('REPLY') 63  
CBLDCMCF('RESEND') 68  
CBLDCMCF('SEND') 74  
CBLDCMCF('SENDRECV') 79  
CCLSEVT 121, 128  
CERREVT 121, 127  
CERREVT の理由コード一覧 233  
COBOL 言語のメッセージ送受信 27  
COPNEVT 121, 128  
C 言語のメッセージ送受信 27

## D

---

dc\_mcf\_receive 29  
dc\_mcf\_recvsync 33  
dc\_mcf\_reply 37  
dc\_mcf\_resend 40  
dc\_mcf\_send 44  
dc\_mcf\_sendrecv 48

## E

---

ECS/VTAM ネットワーク定義と関連づける  
内容 157  
ERREVT1 118, 122  
ERREVT2 119, 123  
ERREVT2 の理由コード一覧 233  
ERREVT3 119, 124  
ERREVT4 120, 126

## G

---

GJ (メッセージ受信ジャーナル) 236

## I

---

IJ (メッセージ入力ジャーナル) 236

## M

---

max\_open\_fds 153  
max\_socket\_descriptors 152  
MCF 6  
mcfmuap 136  
mftacten 173  
mftactle 175  
mftalcen 137  
mftalced 146  
mftalcle 147  
mftalcua 149  
mftbuf 138, 139, 143  
mftcomn 150  
mftdeten 177  
mftdetle 179  
mftlscn 181  
mftlsle 184  
MCF アプリケーション定義 130  
MCF イベント 115  
MCF イベントインタフェース 115  
MCF イベント処理用 MHP 117  
MCF イベント通知時のセグメント構成  
116, 117  
MCF イベントの共通ヘッダ 117  
MCF イベントの種類 115  
MCF 検出障害時の UA の閉局 20  
MCF サービス名の登録 188  
MCF 通信構成共通定義 150  
MCF 通信構成定義 130  
MCF 通信プロセスでアクセスするファイル  
の最大数 153  
MCF 定義オブジェクトの生成 155  
MCF で使用する定義ファイル 130

MCF の障害 203  
 MCF マネージャ定義 130  
 MCF メイン関数の作成 188, 189  
 MJ (メッセージジャーナル) 236

## O

---

OJ (メッセージ出力ジャーナル) 236  
 OpenTP1 システムを使用したネットワーク  
 構成例 2  
 OpenTP1 の障害 203  
 OSAS/UA プロトコル 236  
 OSAS/UA プロトコルの種別 140

## R

---

RECEIVE 85  
 receive 型論理端末への一方送信メッセージ  
 の処理の流れ (UA サービスデータ単位分割  
 ありの場合) 219  
 receive 型論理端末への一方送信メッセージ  
 の処理の流れ (UA サービスデータ単位分割  
 なしの場合) 218  
 request 型論理端末からの一方送信メッセ  
 ージの処理の流れ (UA サービスデータ単位分  
 割ありの場合) 215  
 request 型論理端末からの一方送信メッセ  
 ージの処理の流れ (UA サービスデータ単位分  
 割なしの場合) 214

## S

---

SEND 88  
 send 型論理端末からの一方送信メッセ  
 ージの処理の流れ (UA サービスデータ単位分割  
 ありの場合) 217  
 send 型論理端末からの一方送信メッセ  
 ージの処理の流れ (UA サービスデータ単位分  
 割なしの場合) 216

## T

---

TMS-4V/SP システム定義と関連づける内容  
 158  
 TMS-4V/SP のシステムとの接続 157

TP1/Message Control 6  
 TP1/NET/Library 6  
 TP1/NET/User Agent が適用する AP 間通信  
 のプロトコル 7  
 TP1/NET/User Agent からのコネクションの  
 確立 11  
 TP1/NET/User Agent の OSI7 層構造の中  
 の機能範囲 3  
 TP1/NET/User Agent の UA プロトコルデー  
 タ単位 10  
 TP1/NET/User Agent の運用コマンド 172  
 TP1/NET/User Agent の組み込みの流れ  
 188  
 TP1/NET/User Agent の定義の概要 130  
 TP1/NET/User Agent を組み込んだソフト  
 ウェア構成 7  
 TP1/NET/User Agent を使用した AP 間通信  
 の例 4

## U

---

UA 236  
 UA (論理端末) 障害 197  
 UAP 異常終了 229  
 UAP 異常終了通知イベント 116  
 UAP 共通定義 136  
 UAP 障害 202  
 UA サービスデータ単位 16  
 UA 定義 149  
 UA の開局 17  
 UA の開局と閉局 17  
 UA の開局モード 141  
 UA の数による TP1/NET/User Agent のコネ  
 クション確立 12  
 UA の端末識別子 149  
 UA の閉局 19  
 UA 番号 149  
 UA プロトコルデータ単位 16  
 UINQ 受信時の UA 障害 227  
 UINQ 送信時の UA 障害 225  
 UOC 作成上の注意事項 114  
 UOC の異常処理 114  
 UOC の構造 114  
 UOC の実行タイミング 114

**X**

XDM/DCCM3 システム定義と関連づける内容 160  
 XDM/DCCM3 のシステムとの接続 159  
 XNF ネットワーク定義と関連づける内容 158, 160

**あ**

相手システム (XDM/DCCM3) 接続時の障害 230  
 相手システムからの UA の閉局 20  
 相手システムからのコネクションの解放 13  
 相手システムの通信定義と関連づける内容 157  
 アプリケーションの型 15  
 アプリケーション名 14  
 アプリケーション名の決定 100  
 アプリケーション名の決定の処理 101

**い**

一方受信形態 5  
 一方送信メッセージ 23, 214  
 一方送信メッセージ受信時の処理の流れ 218  
 一方送信メッセージ障害 210, 225  
 一方送信メッセージ送信時の処理の流れ (request 型論理端末) 214  
 一方送信メッセージ送信時の処理の流れ (send 型論理端末) 216  
 一方送信メッセージの送信 (COBOL 言語) 74  
 一方送信メッセージの送信 (C 言語) 44  
 一方送信メッセージの送信と受信 23, 24

**う**

運用コマンド 171  
 運用コマンドによる UA の個別閉局 18  
 運用コマンドによる UA の閉局 19

**お**

応答メッセージ 22  
 応答メッセージ障害 227

応答メッセージ送信時の障害 207  
 応答メッセージ送信時の処理の流れ 222  
 応答メッセージ送信時の処理の流れ (UA サービスデータ単位分割ありの場合) 224  
 応答メッセージ送信時の処理の流れ (UA サービスデータ単位分割なしの場合) 222  
 応答メッセージの送信 22  
 応答メッセージの送信 (COBOL 言語) 63  
 応答メッセージの送信 (C 言語) 37

**か**

概要 1

**き**

機能 9  
 キューグループ ID 148

**く**

組み込み方法 187

**け**

経路交代 14  
 現用相手 DTE アドレス 142  
 現用スロット番号 142

**こ**

交代用 DTE アドレス 142  
 交代用スロット番号 143  
 コネクション 10, 13, 236  
 コネクション (アソシエーション) 障害 196  
 コネクション ID 137  
 コネクション確立再試行 141  
 コネクション確立時再試行回数 141  
 コネクション確立時再試行間隔 141  
 コネクション障害 204  
 コネクション定義の開始 137  
 コネクション定義の終了 146  
 コネクションと論理端末の関係 13  
 コネクションの解放 12, 177  
 コネクションの解放方法 150  
 コネクションの確立 10, 173

コネクションの確立と解放 10  
コネクションの状態表示 181

## し

---

自システムからのコネクションの解放 13  
自システムの PSAP アドレス 137  
システムサービス共通情報定義 152  
システムサービス情報定義 151  
システムサービス情報定義ファイルの作成  
188  
システム定義 129  
ジャーナル障害 201  
受信打ち切りを受信 230  
受信拒否・受信拒否解除を受信 231  
受信バッファ障害 202  
受信メッセージ不正 212  
出力キューおよび出力メッセージ編集 UOC  
の障害 200  
出力メッセージの編集 107  
出力メッセージの編集 UOC 107  
出力メッセージの割り当て先 148  
出力メッセージ編集 UOC インタフェース  
108  
障害対策 195  
障害通知イベント 116  
障害の種類と対応処理 196  
障害発生時の処理の流れ 225  
状態通知イベント 116  
新 OSAS/UA プロトコルを使用する場合の定  
義例 162

## せ

---

制御 UA の端末識別子 142

## そ

---

送信先論理端末の端末タイプ不正 212  
送信中断を受信 232  
送信バッファ障害 201  
送信メッセージの通番編集 111  
送信メッセージの通番編集 UOC インタ  
フェース 112  
ソケット用ファイル記述子の最大数 152

ソフトウェア構成の例 6  
ソフトウェアの構成 6

## た

---

タイマ起動メッセージ廃棄通知イベント 116  
端末識別子の文字数 141  
端末タイプ 147

## つ

---

通信相手システムの PSAP アドレス 142  
通信管理プログラムと関連づける内容 156

## て

---

定義オブジェクトファイルの生成 188, 192  
定義の指定順序 134  
定義例 162  
ディスク出力メッセージ最大格納数 148  
データ操作言語 (COBOL 言語) のメッセー  
ジ送受信 28

## と

---

問い合わせ応答形態 4  
問い合わせ監視時間 140  
問い合わせメッセージ 22  
問い合わせメッセージ、応答メッセージ障害  
206  
問い合わせメッセージ、および応答メッセー  
ジ 220  
問い合わせメッセージ送信時の障害 206  
問い合わせメッセージ送信時の処理の流れ  
220  
問い合わせメッセージ送信時の処理の流れ  
(UA サービスデータ単位分割ありの場合)  
221  
問い合わせメッセージ送信時の処理の流れ  
(UA サービスデータ単位分割なしの場合)  
220  
問い合わせメッセージの受信と応答メッセー  
ジの送信 23  
問い合わせメッセージの送信 21

問い合わせメッセージの送信と応答メッセージの受信 22  
 同期型のメッセージの受信 (COBOL 言語) 58  
 同期型のメッセージの受信 (C 言語) 33  
 同期型のメッセージの送受信 (COBOL 言語) 79  
 同期型のメッセージの送受信 (C 言語) 48  
 同期送受信監視時間 136  
 同期問い合わせ応答形態 5

## に

入力キュー, スケジュールサービス, ネームサービス, または RPC の障害 199  
 入力メッセージの編集 100  
 入力メッセージの編集とアプリケーション名の決定 100  
 入力メッセージ編集 UOC 100  
 入力メッセージ編集 UOC インタフェース 102  
 入力メッセージ編集 UOC エラー 226, 228  
 入力メッセージ編集 UOC の障害 198

## ね

ネットワーク構成の例 (TMS-4V/SP のシステムと接続する場合) 157  
 ネットワーク構成の例 (XDM/DCCM3 のシステムと接続する場合) 160

## ふ

不正アプリケーション名検出通知イベント 115  
 プロトコル監視時間 140  
 分岐送信形態 4

## へ

編集バッファ障害 202

## み

未処理送信メッセージ廃棄通知イベント 116

## め

メッセージ受信用バッファグループ番号 138  
 メッセージ制御機能 6  
 メッセージ送受信インタフェース 25  
 メッセージ送受信インタフェースの一覧 27  
 メッセージ送受信の処理の流れ 214  
 メッセージ送受信の通信文 (データ操作言語) 28  
 メッセージ送受信の文 (COBOL 言語) 27  
 メッセージ送受信のライブラリ関数 (C 言語) 27  
 メッセージ送信用バッファグループ番号 138  
 メッセージの再送 (COBOL 言語) 68  
 メッセージの再送 (C 言語) 40  
 メッセージの受信 (COBOL 言語) 53  
 メッセージの受信 (C 言語) 29  
 メッセージの受信 (データ操作言語) 85  
 メッセージの送信 (データ操作言語) 88  
 メッセージのデータ形式 17  
 メッセージの分割と組み立て 16  
 メッセージ廃棄通知イベント 115  
 メッセージ編集用バッファグループ番号 139  
 メッセージ編集用バッファ数 139  
 メモリ出力メッセージ最大格納数 147

## ゆ

ユーザアプリケーションプログラム作成例 94  
 ユーザOWNコーディング 100  
 ユーザOWNコーディングインタフェース 100

## り

理由コード一覧 233

## ろ

論理端末 13, 237  
 論理端末, メッセージ, アプリケーションの型, UAP インタフェース, および通信形態の関係 15  
 論理端末定義 147

- 論理端末とアプリケーションの型の関係 14
- 論理端末の状態表示 184
- 論理端末の端末タイプ 14
- 論理端末の端末タイプ不正 212
- 論理端末の閉塞 179
- 論理端末の閉塞解除 175
- 論理端末名称 147
- 論理メッセージ 16



# ソフトウェアマニュアルのサービス ご案内

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

## 1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しております。

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

ホームページのメニューは次のとおりです。

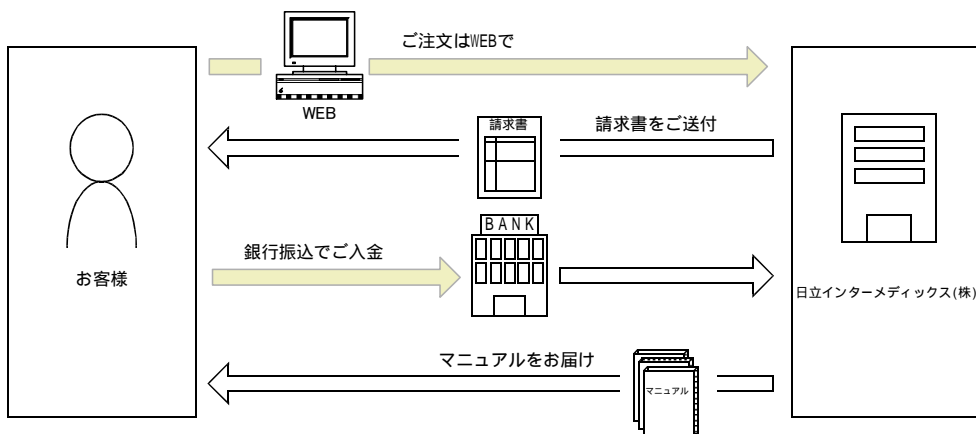
マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル情報	複数マニュアルを格納したCD-ROMマニュアルを提供しています。どの製品に対応したCD-ROMマニュアルがあるか、を参照できます。
マニュアルのご購入	日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からお申し込みできます。 (詳細は「3. マニュアルのご注文」を参照してください。)
Web提供マニュアル一覧	インターネットで参照できるマニュアルの一覧を提供しています。 (詳細は「2. インターネットからのマニュアル参照」を参照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

## 2. インターネットからのマニュアル参照(ソフトウェアサポートサービス)

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

## 3. マニュアルのご注文

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



下記 URL にアクセスして必要事項を入力してください。

URL [http://www2.himdx.net/manual/privacy.asp?purchase\\_flag=1](http://www2.himdx.net/manual/privacy.asp?purchase_flag=1)

ご注文いただいたマニュアルについて、請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。なお、送料は弊社で負担します。

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