

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/NCSB 編

解説・手引・文法・操作書

3000-3-D78

## マニュアルの購入方法

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

## 対象製品

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

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

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

P-F1M64-32317 uCosminexus TP1/NET/NCSB 07-00

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

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

## 輸出時の注意

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

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

## 商標類

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

CAFIS は、株式会社 NTT データの登録商標です。

クレジット情報データ通信システムは、日本電信電話（株）のサービス名称です。

## 発行

2009 年 1 月（第 1 版）3000-3-D78

## 著作権

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

# はじめに

---

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

このマニュアルに記載するプログラムプロダクトは次のとおりです。

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

- P-1M64-3131 uCosminexus TP1/Message Control
- P-1M64-3231 uCosminexus TP1/NET/Library
- P-F1M64-32317 uCosminexus TP1/NET/NCSB

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

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、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/NCSB を使用したシステム間の通信 (AP 間通信) の概要について説明しています。

### 第 2 章 機能

TP1/NET/NCSB を使用した場合の、コネクションや論理端末、メッセージ送受信の流れなどについて説明しています。

### 第 3 章 メッセージ送受信インタフェース

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

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

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

NCSB プロトコルを使用するために必要な、OpenTP1 システム定義の中での TP1/NET/NCSB 固有の定義について説明しています。また、自システム内にある通信管理プログラムと関連づける

はじめに

内容、およびシステム定義例について説明しています。

## 第 6 章 運用コマンド

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

## 第 8 章 障害対策

TP1/NET/NCSB の障害時の処理とユーザの対策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、およびジャーナル取得のタイミングについて説明しています。

## 付録 B 障害発生時の処理の流れ

障害が発生した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を、ERREVT2、CERREVT、および MDELEVT に分けて説明しています。

## 付録 D 旧製品からの移行に関する注意事項

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

## 付録 E バージョンアップ時の変更点

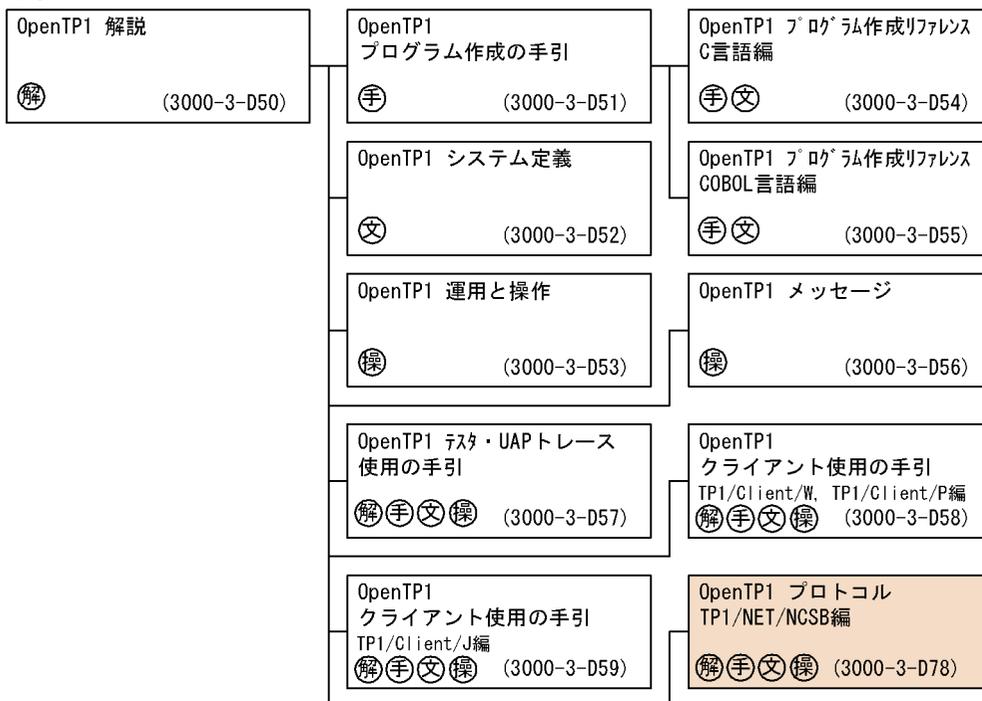
各バージョンでの関数、定義およびコマンドの変更点について説明しています。

## 付録 F 用語解説

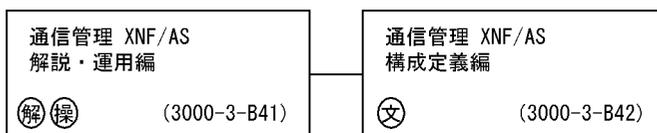
TP1/NET/NCSB で使用する用語について説明しています。

## 関連マニュアル

### ●OpenTP1 Version 7



### ●通信管理

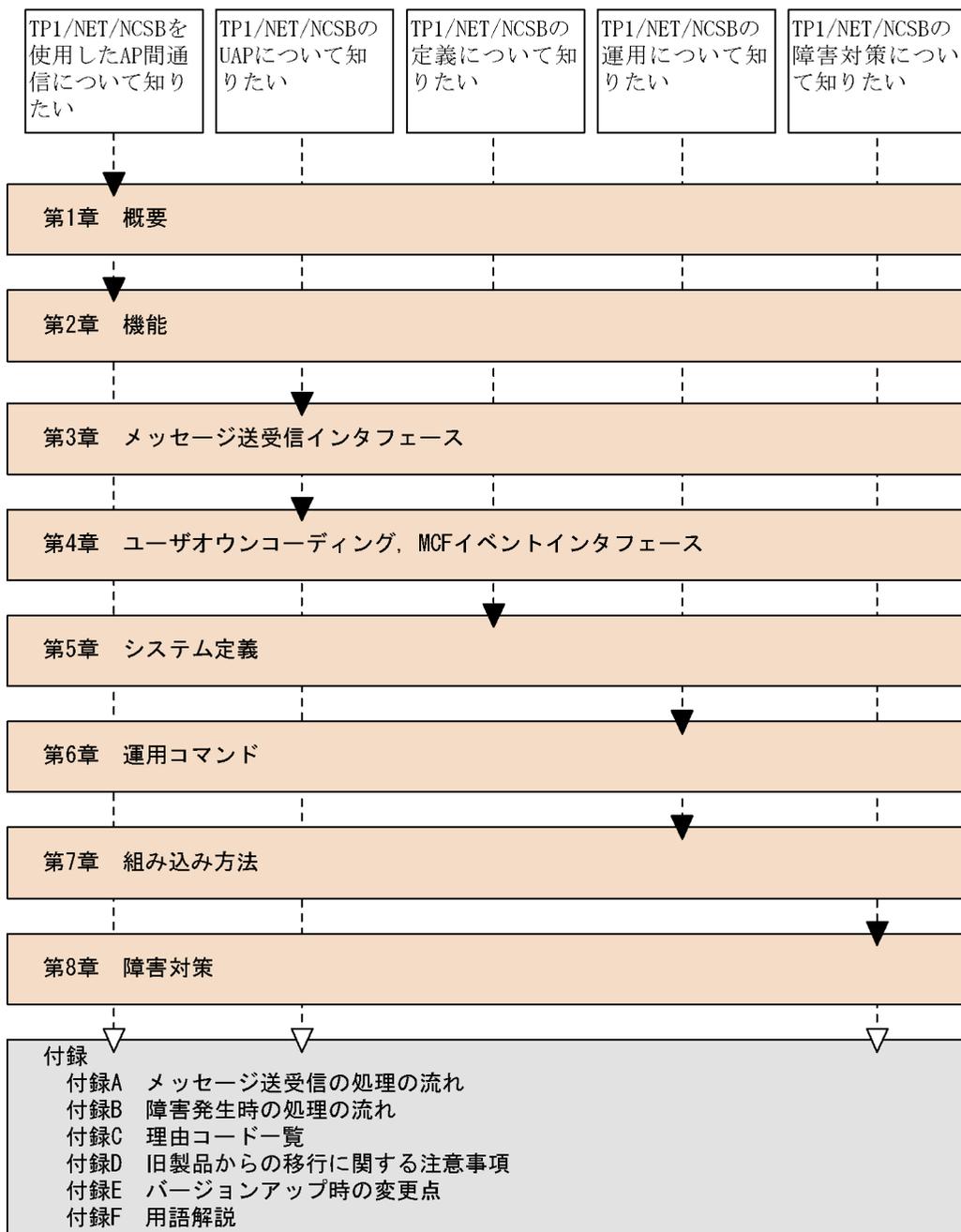


#### <記号>

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

## 読書手順

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



(凡例)



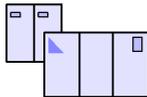
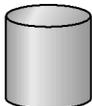
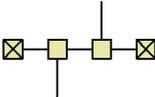
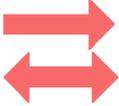
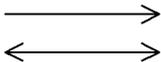
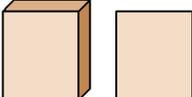
: 必ず読む項目



: 必要に応じて読む項目

## 図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。

●ワークステーション、 ●論理端末 端末	●論理端末 (端末状態)	●論理回線	●ホストコンピュータ
			
●ファイル	●ネットワーク (WAN)	●ネットワーク (LAN)	●通信回線
			
●プログラムの流れ	●データ、メッセージ の流れ	●制御の流れ	●プログラム
			
●入出力の動作	●障害		
			

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

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

コード	JIS コード配列	ASCII コード配列
(5c) <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)

## 文法の記号

このマニュアルで使用する各種の記号を説明します。

## (1) 文法記述記号

文法の記述形式について説明する記号です。

文法記述記号	意味
[ ]	この記号で囲まれている項目は省略してもよいことを示します。 (例) [-s MCF 通信プロセス識別子] -s オプションとそのオペランドを指定するか、何も指定しないことを示します。
 (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) -y winner   loser -y オプションに winner または loser を指定できることを示します。 ただし、C 言語のインタフェースの説明でこの記号を使用した場合は、C 言語の文法規則に従います。
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。 (例) {DCMCFFRST   DCMCFSEG} DCMCFFRST と DCMCFSEG のうち、どちらかを指定することを示します。
<u>      </u> (下線)	この記号で示す項目は、オペランド、オプションまたはコマンド引数を省略した場合の省略時解釈値を示します。 (例) -ai = <u>yes</u>   no -ai オプションを省略した場合、yes 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)

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

### このマニュアルでの表記

このマニュアルでは, 製品名称を省略して表記しています。製品名称と, このマニュアルでの表記を次に示します。

製品名称	このマニュアルでの表記	
AIX 5L V5.1	AIX	
AIX 5L V5.2		
AIX 5L V5.3		
uCosminexus TP1/Message Control	TP1/Message Control	
uCosminexus TP1/Message Control/Tester	TP1/Message Control/Tester	
uCosminexus TP1/NET/Library	TP1/NET/Library	
uCosminexus TP1/NET/NCSB	TP1/NET/NCSB	
uCosminexus TP1/Server Base	TP1/Server Base	
XNF/AS/BASE	XNF/AS	XNF
XNF/AS/NCSB		
XNF/AS/WAN		

### 略語一覧

このマニュアルで使用する英略語の一覧を次に示します。

英略語	英字での表記
AP	<u>A</u> pplication <u>P</u> rogram
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork
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

英略語	英字での表記
OS	<u>O</u> perating <u>S</u> ystem
RPC	<u>R</u> emote <u>P</u> rocedure <u>C</u> all
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram
UAP	<u>U</u> ser <u>A</u> pplication <u>P</u> rogram
UOC	<u>U</u> ser <u>O</u> wn <u>C</u> oding
WAN	<u>W</u> ide <u>A</u> rea <u>N</u> etwork

### 常用漢字以外の漢字の使用について

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

迂回（うかい） 個所（かしよ） 必須（ひつす） 閉塞（へいそく）

### KB（キロバイト）などの単位表記について

1KB（キロバイト）、1MB（メガバイト）、1GB（ギガバイト）、1TB（テラバイト）はそれぞれ1,024バイト、1,024<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

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	3
1.2.1	分岐送信形態	3
1.2.2	一方受信形態	3
1.3	ソフトウェアの構成	4
1.3.1	前提プログラム	4
1.3.2	ソフトウェア構成の例	4
<b>2</b>	<b>機能</b>	<b>7</b>
2.1	AP 間通信の仕組み	8
2.1.1	コネクションとコネクショングループ	8
2.1.2	コネクショングループと論理端末	12
2.1.3	メッセージ送信時のコネクション選択	13
2.1.4	論理端末とアプリケーションの型との関係	14
2.1.5	メッセージの分割と組み立て	15
2.2	AP 間通信メッセージの送受信	16
2.2.1	一方送信メッセージの送信	16
2.2.2	一方送信メッセージの受信	17
<b>3</b>	<b>メッセージ送受信インタフェース</b>	<b>19</b>
	メッセージ送受信インタフェースの一覧	20
	dc_mcf_receive - 一方送信メッセージの受信 (C 言語)	22
	dc_mcf_resend - メッセージの再送 (C 言語)	27
	dc_mcf_send - 一方送信メッセージの送信 (C 言語)	32
	CBLDCMCF('RECEIVE ') - 一方送信メッセージの受信 (COBOL 言語)	36
	CBLDCMCF('RESEND ') - メッセージの再送 (COBOL 言語)	42
	CBLDCMCF('SEND ') - 一方送信メッセージの送信 (COBOL 言語)	48
	RECEIVE - メッセージの受信 (データ操作言語)	55
	SEND - 一方送信メッセージの送信 (データ操作言語)	58
	ユーザアプリケーションプログラムの作成例	62

<b>4</b>	<b>ユーザOWNコーディング，MCF イベントインタフェース</b>	<b>67</b>
4.1	ユーザOWNコーディングインタフェース	68
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	68
4.1.2	入力メッセージ編集 UOC インタフェース	71
4.1.3	出力メッセージの編集	76
4.1.4	出力メッセージ編集 UOC インタフェース	77
4.1.5	送信メッセージの迂回判定	80
4.1.6	送信メッセージの迂回判定 UOC インタフェース	82
4.1.7	送信メッセージの通番編集	85
4.1.8	送信メッセージの通番編集 UOC インタフェース	87
4.1.9	UOC 作成上の注意事項	88
4.2	MCF イベントインタフェース	90
4.2.1	MCF イベントの種類	90
4.2.2	MCF イベント通知時のセグメント構成	91
4.2.3	MCF イベント情報の形式 (C 言語)	92
4.2.4	MCF イベント情報の形式 (COBOL 言語)	96
<b>5</b>	<b>システム定義</b>	<b>107</b>
	TP1/NET/NCSB の定義の概要	108
	TP1/NET/NCSB 固有のシステム定義の種類	110
	mcftalccn (コネクション定義)	113
	mcftalcle (論理端末定義)	118
	mcftgrpcn (コネクショングループ定義の開始)	120
	mcftgrped (コネクショングループ定義の終了)	121
	システムサービス情報定義	122
	システムサービス共通情報定義	123
	MCF 定義オブジェクトの生成	125
	自システムの通信管理プログラムと関連づける内容	126
	定義例	127
<b>6</b>	<b>運用コマンド</b>	<b>131</b>
	TP1/NET/NCSB の運用コマンド	132
	mcftactcn (コネクションの確立)	133

mcftactle (論理端末の閉塞解除)	136
mcftdctcn (コネクションの解放)	138
mcftdctle (論理端末の閉塞)	141
mcftlscn (コネクションの状態表示)	143
mcftlslc (論理端末の状態表示)	146

## 7

組み込み方法	149
7.1 TP1/NET/NCSB の組み込みの流れ	150
7.2 MCF メイン関数の作成	151
7.3 定義オブジェクトファイルの生成	154

## 8

障害対策	157
8.1 障害の種類と対応処理	158

## 付録

付録 A メッセージ送受信の処理の流れ	166
付録 A.1 メッセージの送信	166
付録 A.2 メッセージの受信	167
付録 B 障害発生時の処理の流れ	170
付録 C 理由コード一覧	175
付録 C.1 ERREVT2 の理由コード	175
付録 C.2 CERREVT の理由コード	175
付録 C.3 MDELEVT の理由コード	177
付録 D 旧製品からの移行に関する注意事項	178
付録 E バージョンアップ時の変更点	179
付録 E.1 07-00 での変更点	179
付録 F 用語解説	180

## 索引

索引	183
----	-----

## 図目次

図 1-1	TP1/NET/NCSB を使用したネットワーク構成の例	2
図 1-2	分岐送信形態の AP 間通信の例	3
図 1-3	一方受信形態の AP 間通信の例	3
図 1-4	TP1/NET/NCSB を組み込んだソフトウェア構成の例	4
図 2-1	コネクション，コネクショングループ，および論理端末の関係	8
図 2-2	コネクション確立の流れ	10
図 2-3	コネクション解放の流れ	11
図 2-4	コネクション切断の流れ	12
図 2-5	迂回不可メッセージ廃棄通知イベントが通知される場合の処理の流れ	14
図 2-6	メッセージ分割の例	15
図 2-7	メッセージ送信の流れ	16
図 2-8	メッセージ受信の流れ	17
図 3-1	UAP の作成例の処理の流れ	62
図 4-1	アプリケーション名の決定の処理	70
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	75
図 4-3	送信メッセージの迂回判定の処理	81
図 4-4	MCF イベント通知時のセグメント構成	91
図 5-1	TP1/NET/NCSB のシステム構成例	126
図 7-1	MCF メイン関数のコーディング概要 (ANSI C, C++ の場合)	151
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)	152
図 7-3	MCF メイン関数のディレクトリへの組み込み方法	153
図 7-4	定義オブジェクトファイルの作成方法の概要	155
図 A-1	メッセージ送信の処理の流れ (単一セグメントの場合)	166
図 A-2	メッセージ送信の処理の流れ (複数セグメントの場合)	167
図 A-3	メッセージ受信の処理の流れ (単一セグメントの場合)	168
図 A-4	メッセージ受信の処理の流れ (複数セグメントの場合)	169
図 B-1	コネクション障害	170
図 B-2	メッセージ送信中の回線障害 (送信メッセージの迂回判定 UOC で迂回する場合)	171
図 B-3	メッセージ送信中の回線障害 (送信メッセージの迂回判定 UOC で迂回しない場合)	172
図 B-4	入力メッセージ編集 UOC エラーリターン	173
図 B-5	入力メッセージ編集 UOC パラメタ不正	174

## 表目次

表 2-1	論理端末の端末タイプ，アプリケーションの型，通信形態，メッセージの種類，および UAP インタフェースの関係	15
表 3-1	メッセージ送受信の関数（C 言語）	20
表 3-2	メッセージ送受信の関数に対応するプログラム（COBOL 言語）	20
表 3-3	メッセージ送受信の通信文（データ操作言語）	21
表 4-1	MCF イベント一覧	90
表 4-2	COBOL 言語の MCF イベント情報の内容（ERREVT1）	96
表 4-3	COBOL 言語の MCF イベント情報の内容（ERREVT2）	98
表 4-4	COBOL 言語の MCF イベント情報の内容（ERREVT3）	99
表 4-5	COBOL 言語の MCF イベント情報の内容（ERREVT4）	101
表 4-6	COBOL 言語の MCF イベント情報の内容（ERREVT5）	102
表 4-7	COBOL 言語の MCF イベント情報の内容（COPNEVT，CCLSEVT）	103
表 4-8	COBOL 言語の MCF イベント情報の内容（MDELEVT）	104
表 5-1	MCF で使用する定義ファイル	108
表 5-2	TP1/NET/NCSB に固有の定義の一覧	110
表 6-1	TP1/NET/NCSB で使用する運用コマンド	132
表 8-1	コネクション障害の種類と対応処理	158
表 8-2	受信スケジュール関係の障害の種類と対応処理	159
表 8-3	送信スケジュール関係の障害の種類と対応処理	161
表 8-4	UAP 障害の種類と対応処理	163
表 8-5	MCF 障害の種類と対応処理	163
表 C-1	ERREVT2 の理由コード	175
表 C-2	コネクション障害の場合の理由コード	176
表 C-3	論理端末障害の場合の理由コード	177
表 C-4	MDELEVT の理由コード	177
表 D-1	バージョン 6 以前で使用していたソースファイルの互換性	178
表 E-1	TP1/NET/NCSB 07-00 での動作の変更点	179



# 1

## 概要

TP1/NET/NCSB は、OpenTP1 システムを構成するプログラムの一つです。ホストコンピュータと端末を論理的に接続し、銀行間提携システムなどで使用される伝送制御手順でメッセージを送受信します。この章では、TP1/NET/NCSB を使用したシステム間の通信（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ます。

---

1.1 AP 間通信の概要

---

1.2 AP 間通信の形態

---

1.3 ソフトウェアの構成

---

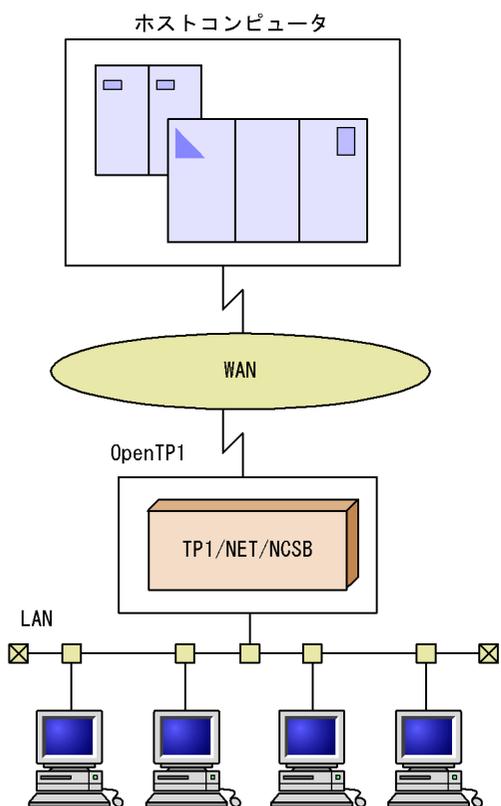
## 1.1 AP 間通信の概要

AP 間通信とは、異なるシステムにあるアプリケーションプログラム間でのメッセージの送受信をいいます。

TP1/NET/NCSB は、OpenTP1 システムと各種コンピュータシステムを接続し、銀行間提携システムなどで使用される伝送制御手順で AP 間通信をするプログラムです。TP1/NET/NCSB を使用した AP 間通信では、相手システムで発生したトランザクションを自システムで処理したり、その結果を相手システムへ送ったりできます。

TP1/NET/NCSB を使用したネットワーク構成の例を次の図に示します。

図 1-1 TP1/NET/NCSB を使用したネットワーク構成の例



## 1.2 AP 間通信の形態

TP1/NET/NCSB を使用した AP 間通信の形態には、次の二つがあります。

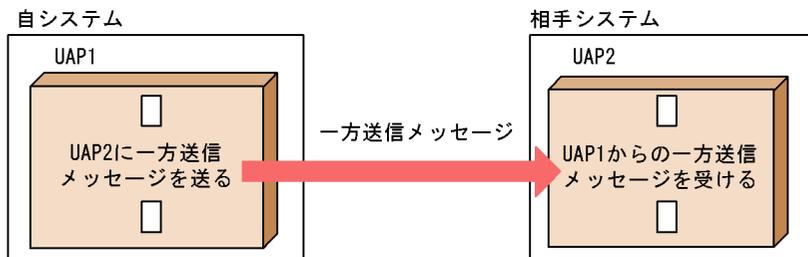
- 分岐送信形態
- 一方受信形態

### 1.2.1 分岐送信形態

自システムのユーザアプリケーションプログラム (UAP) から別のシステムの UAP へメッセージを送信する形態です。自システムから相手システムへ送信するメッセージを、一方送信メッセージといいます。

分岐送信形態の AP 間通信の例を次の図に示します。

図 1-2 分岐送信形態の AP 間通信の例



### 1.2.2 一方受信形態

自システムの UAP が相手システムの UAP からメッセージを受信する形態です。相手システムから受信するメッセージも、分岐送信形態の場合と同様に一方送信メッセージといいます。

一方受信形態の AP 間通信の例を次の図に示します。

図 1-3 一方受信形態の AP 間通信の例



## 1.3 ソフトウェアの構成

TP1/NET/NCSB 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/NCSB を使用した MCF を実現するための前提プログラムと、TP1/NET/NCSB 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/NCSB 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-F1M14-5114 XNF/AS/NCSB

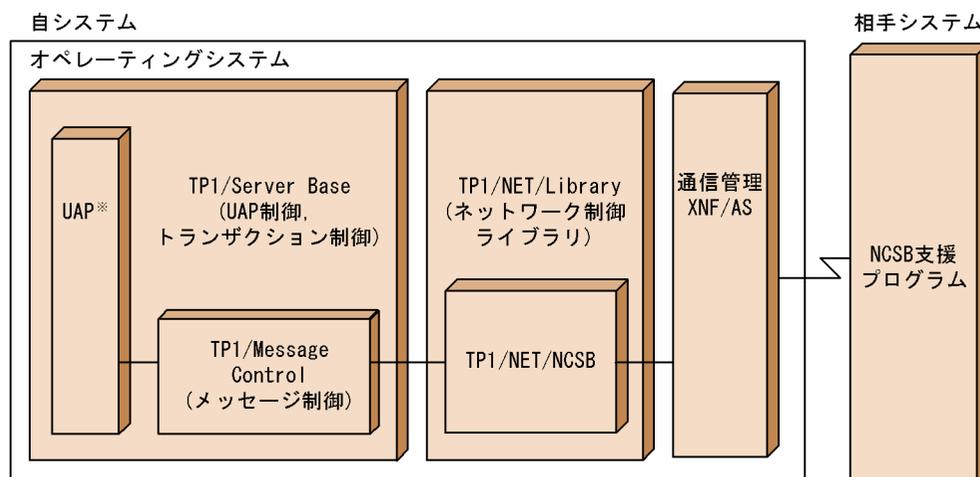
注

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

### 1.3.2 ソフトウェア構成の例

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

図 1-4 TP1/NET/NCSB を組み込んだソフトウェア構成の例



注

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



# 2

## 機能

TP1/NET/NCSB は AP 間通信機能を提供し、メッセージを送受信します。この章では AP 間通信に必要な知識として、コネクションや論理端末、メッセージ送受信の流れなどを説明します。

---

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

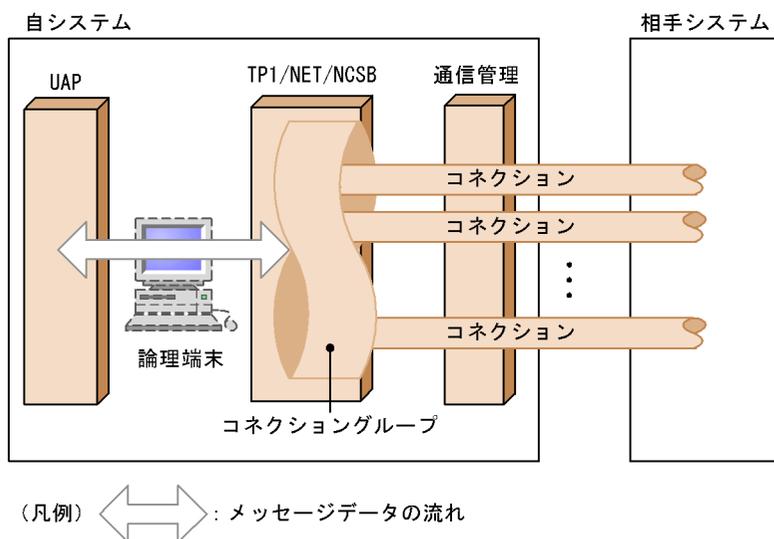
TP1/NET/NCSB を使用した AP 間通信では、通信相手システムを表す論理的な概念として、次の三つがあります。

- コネクション
- コネクショングループ
- 論理端末 (LE)

コネクションは、通信管理が相手システムに接続するときの論理的な通信路です。コネクショングループは、幾つかのコネクションをまとめたものです。論理端末は、UAP と TP1/NET/NCSB との接点です。

コネクション、コネクショングループ、および論理端末の関係を次の図に示します。

図 2-1 コネクション、コネクショングループ、および論理端末の関係



### 2.1.1 コネクションとコネクショングループ

コネクションは、通信相手システムへの論理的な通信路を示します。コネクションを確立・解放することで、相手システムに接続したり、相手システムから切り離したりします。コネクショングループは、同じ目的で使用される幾つかのコネクションをまとめたものです。メッセージを送受信するには、コネクショングループ内のコネクションが一つ以上確立されている必要があります。

コネクションとコネクショングループとの関係は、MCF 通信構成定義で指定します。なお、同じコネクショングループに属するコネクションでは、定義の指定値を一致させる必要があるものがあります。詳細は「5. システム定義」を参照してください。

## (1) コネクションおよびコネクショングループの確立

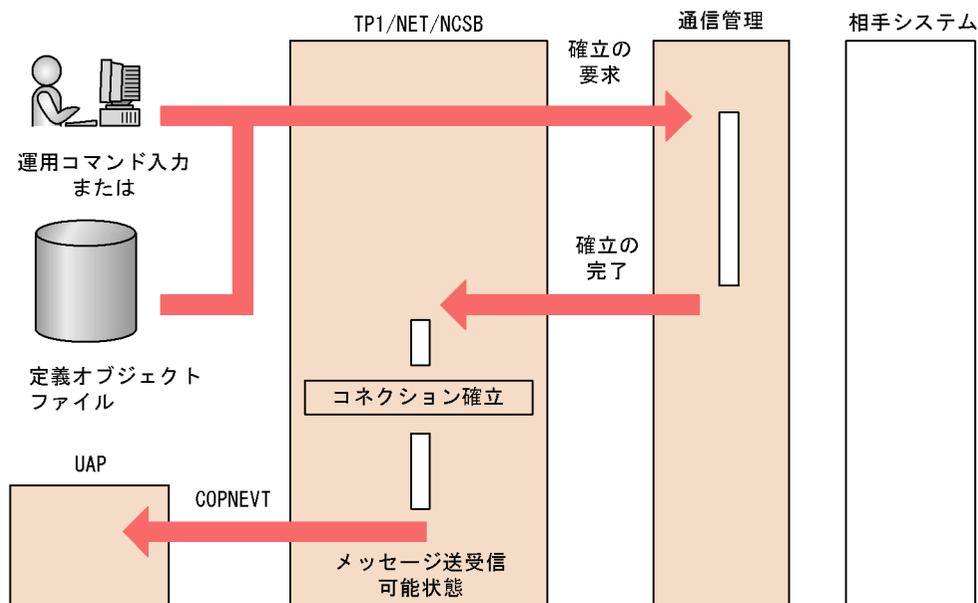
相手システムと通信するには、コネクションを確立する必要があります。コネクショングループ内のコネクションが一つ以上確立されている状態を、そのコネクショングループの確立状態といいます。

コネクションの確立には、次の方法があります。

- オンライン開始時の自動確立  
ユーザが MCF 通信構成定義 (mcftalccn -i) でコネクションの自動確立を指定しておきます。そのコネクションは、オンライン開始時に、TP1/NET/NCSB によって自動的に確立されます。  
コネクションが確立されると、状態通知イベント (COPNEVT) が UAP に通知されます。ただし、MCF アプリケーション定義で COPNEVT を定義した場合だけです。
- 運用コマンドによる手動確立  
ユーザがオンライン中に運用コマンド (mcftactcn) を入力して確立します。運用コマンドでは、コネクションまたはコネクショングループを指定します。コネクショングループを指定すると、コネクショングループ内のすべてのコネクションに対して確立を試みられます。このとき、確立済みのコネクションに対しても確立が試みられます。mcftactcn コマンドはエラーとなりますが、処理は続行されます。  
コネクションが確立されると、状態通知イベント (COPNEVT) が UAP に通知されます。ただし、MCF アプリケーション定義で COPNEVT を定義した場合だけです。  
コネクショングループを指定して確立した場合は、グループ内のコネクションごとにイベントが通知されます。

コネクション確立の流れを次の図に示します。

図 2-2 コネクション確立の流れ



コネクション確立時に障害が発生した場合、MCF 通信構成定義 (mcftalccn -b) で "bretry=yes" を指定、またはそのオペランドを省略しているときは、TP1/NET/NCSB はコネクションの再確立を試みます。MCF 通信構成定義 (mcftalccn -b) の bretrycnt オペランドで指定した回数を繰り返しても確立できなかったときは、メッセージログ (KFCA15603-E) を出力してコネクションの確立処理を終了します。このとき、TP1/NET/NCSB は、MCF アプリケーション定義で CERREVT を定義していると、UAP に対して CERREVT を通知します。

再度コネクションの確立を試みたい場合、メッセージログおよび CERREVT で通知された内容を基に障害の要因を取り除いてから、運用コマンド (mcftacten) を使用してください。

## (2) コネクションおよびコネクショングループの解放

相手システムとの通信を終了するために、コネクションを解放します。コネクショングループ内のコネクションが一つも確立されていない状態を、そのコネクショングループの解放状態といいます。

コネクションが解放されるときに、受信メッセージが破棄される場合があります。このとき、障害通知イベント (CERREVT) は通知されません。

コネクションの解放には、次の方法があります。

- 運用コマンドによる手動解放  
ユーザがオンライン中に運用コマンド (mcftdeten) を入力して解放します。運用コマ

ンドでは、コネクションまたはコネクショングループを指定します。コネクショングループを指定すると、コネクショングループ内のすべてのコネクションに対して解放が試みられます。このとき、確立されていないコネクションに対しても解放が試みられます。mcfddetcn コマンドはエラーとなりますが、処理は続行されます。

コネクションが解放されると、状態通知イベント（CCLSEVT）がUAPに通知されます。ただし、MCFアプリケーション定義でCCLSEVTを定義した場合だけです。コネクショングループを指定して解放した場合は、グループ内のコネクションごとにイベントが通知されます。また、運用コマンド（mcfddetcn -f）を入力してコネクションを強制解放した場合、コネクションレベルの障害通知イベント（CERREVT）がUAPに通知されます。

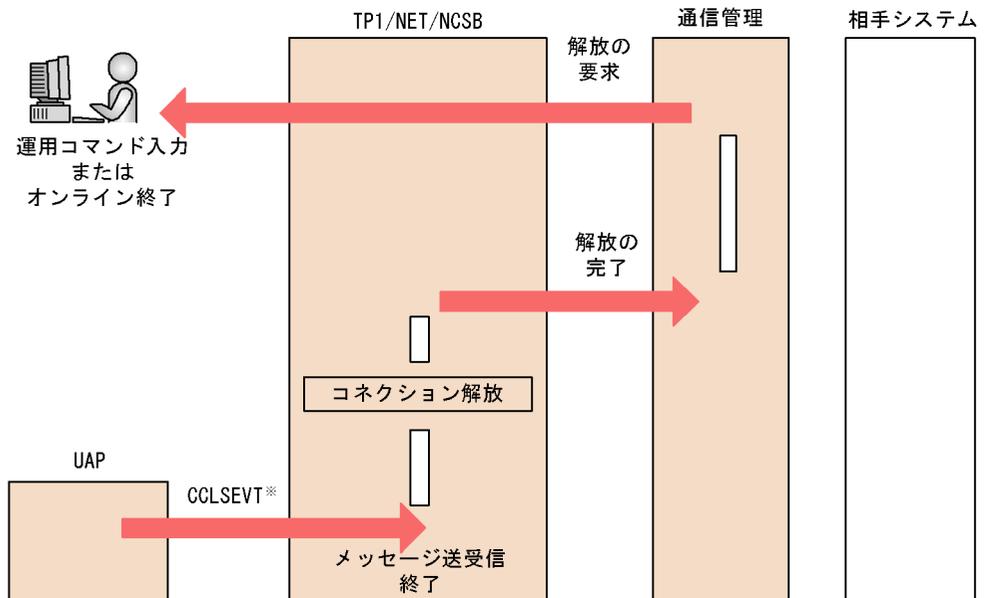
- オンライン終了時の自動解放

すべてのコネクションは、オンライン終了時に、TP1/NET/NCSBによって自動的に解放されます。このとき、状態通知イベント（CCLSEVT）は通知されません。

コネクションの解放は、コネクションごとに状態通知イベント（CCLSEVT）で通知されます。コネクショングループに対しては通知されません。したがって、ユーザはコネクショングループ内のそれぞれのコネクションの状態を把握して、コネクショングループが解放状態になったかどうかを判断してください。

コネクション解放の流れを次の図に示します。

図 2-3 コネクション解放の流れ



注

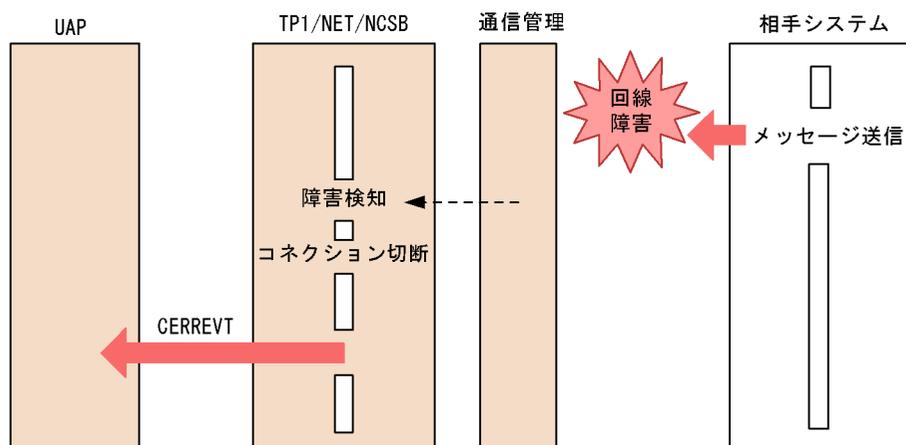
オンライン終了時は通知されません。

### (3) コネクションの切断

障害によってコネクションに対する回線がすべて使用できなくなったとき、TP1/NET/NCSB は、そのコネクションを切断状態にします。コネクションが切断状態になると、TP1/NET/NCSB は、状態通知イベント (CERREVT) を UAP に通知します。コネクションが切断された場合の対策については、「8. 障害対策」を参照してください。

コネクション切断の流れを次の図に示します。

図 2-4 コネクション切断の流れ



## 2.1.2 コネクショングループと論理端末

論理端末 (LE) は、UAP と TP1/NET/NCSB との接点です。UAP は、特定の通信相手システムを示す論理端末からメッセージを受信し、同じまたは別の論理端末へメッセージを送信します。

論理端末は、コネクショングループと 1 対 1 に対応します。コネクショングループと論理端末の対応は、MCF 通信構成定義で指定します。

### (1) 論理端末の閉塞

メッセージを送信できない論理端末の状態を、論理端末の閉塞状態といいます。このとき、送信メッセージは、論理端末に対応する出力キューに残ります。

論理端末が閉塞状態であっても、UAP から送信要求はできます。また、相手システムから送信されたメッセージを閉塞状態の論理端末で受け取ることもできます。

論理端末に対応するコネクショングループが解放状態になると、TP1/NET/NCSB は、その論理端末を自動的に閉塞します。また、ユーザが運用コマンド (mcftdetle) で論理端末を閉塞することもできます。ただし、使用中の論理端末は閉塞できません。なお、閉塞状態の論理端末を閉塞しようとする、警告を示すメッセージログが出力されます。

## (2) 論理端末の閉塞解除

論理端末に対応するコネクショングループが確立状態になると、TP1/NET/NCSB は、その論理端末の閉塞状態を自動的に解除します。また、ユーザが運用コマンド (mcftactle) で論理端末を閉塞解除することもできます。ただし、論理端末に対応するコネクショングループが確立状態である必要があります。閉塞されていない論理端末を閉塞解除しようとする、警告を示すメッセージログが出力されます。

### 2.1.3 メッセージ送信時のコネクション選択

UAP が論理端末を指定してメッセージを送信したときに、実際に使用されるコネクションは次のようになります。

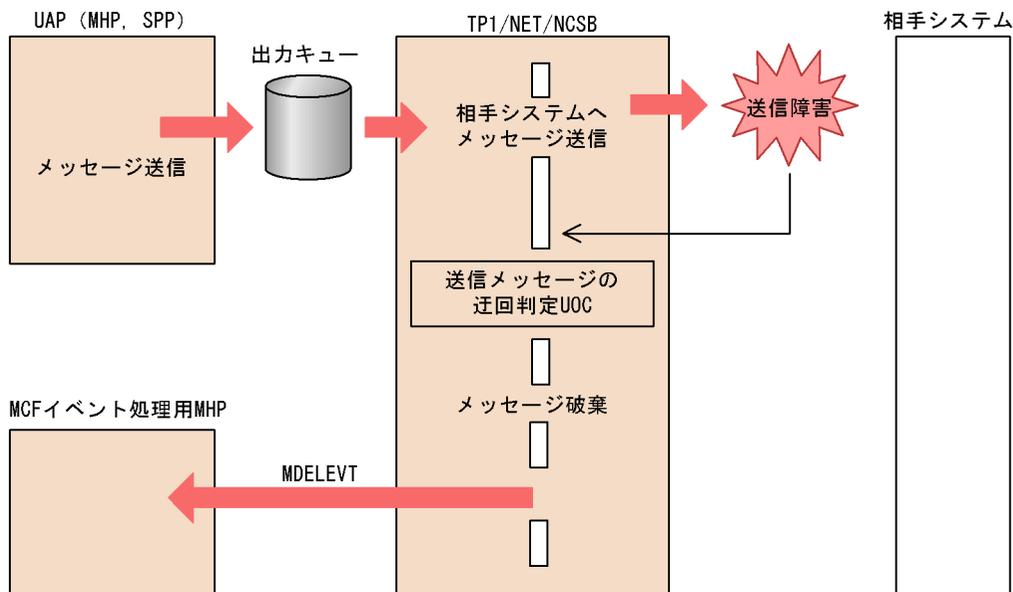
使用されるコネクションは、指定した論理端末に対応するコネクショングループ内から、TP1/NET/NCSB によってラウンドロビン方式で選択されます。ただし、確立状態でないもの、ほかのメッセージの送受信で使用されているものは除かれます。

コネクション障害などでメッセージが正しく送信されなかった場合、TP1/NET/NCSB は、送信メッセージの迂回判定 UOC を呼び出します。送信メッセージの迂回判定 UOC では、送信メッセージと通信管理から渡された理由コードを基に、ほかのコネクションに迂回してメッセージを再送するかどうかを判定します。判定の結果、迂回する場合はほかの正常なコネクションを使用して再びメッセージ送信を試みます。迂回しない場合は、迂回不可メッセージ廃棄通知イベント (MDELEVT) を通知して、送信メッセージを廃棄します。

送信障害時に迂回不可メッセージ廃棄通知イベント (MDELEVT) が通知される場合の処理の流れを次の図に示します。

## 2. 機能

図 2-5 迂回不可メッセージ廃棄通知イベントが通知される場合の処理の流れ



送信メッセージの迂回判定 UOC が登録されていない場合、迂回しない場合と同じに処理します。

送信メッセージの迂回判定 UOC のインターフェースの詳細は、「4.1.6 送信メッセージの迂回判定 UOC インタフェース」を参照してください。

### 2.1.4 論理端末とアプリケーションの型との関係

TP1/NET/NCSB で使用する論理端末の端末タイプを示します。論理端末定義 (mcftalcle -t) で、この端末タイプを指定してください。

any : 任意型論理端末

次に、TP1/NET/NCSB で使用するアプリケーションの型を示します。アプリケーション属性定義 (mcfaalcap -n) で、この型を指定してください。

noans : 非応答型

論理端末定義については「5. システム定義」を、アプリケーション属性定義については、マニュアル「OpenTP1 システム定義」を参照してください。

論理端末の端末タイプ、アプリケーションの型、通信形態、メッセージの種類、および UAP インタフェースの関係を次の表に示します。

表 2-1 論理端末の端末タイプ，アプリケーションの型，通信形態，メッセージの種類，および UAP インタフェースの関係

論理端末の 端末タイプ	アプリケー ションの型	通信形態	メッセージの種類	UAP インタフェース
any	noans	一方受信形態	一方送信メッセージ	receive
		分岐送信形態	一方送信メッセージ	send
				resend

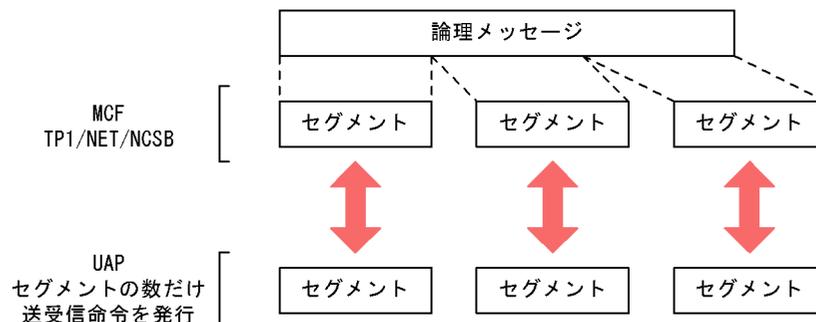
## 2.1.5 メッセージの分割と組み立て

TP1/NET/NCSB と UAP との間では，メッセージを，セグメントと呼ばれる単位に分割して扱います。一つの業務で処理するメッセージを，論理メッセージといいます。論理メッセージは，一つまたは複数のセグメントで構成されます。

UAP で送受信命令を呼び出すと，一つのセグメントを送信または受信します。一つの論理メッセージが複数のセグメントで構成される場合，セグメントの数だけ送受信命令を呼び出してください。

メッセージの分割の例を次の図に示します。

図 2-6 メッセージ分割の例



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

## 2.2 AP 間通信メッセージの送受信

TP1/NET/NCSB は、AP 間通信メッセージを送受信します。TP1/NET/NCSB には、次のメッセージの送受信があります。

- 一方送信メッセージの送信
- 一方送信メッセージの受信

### 2.2.1 一方送信メッセージの送信

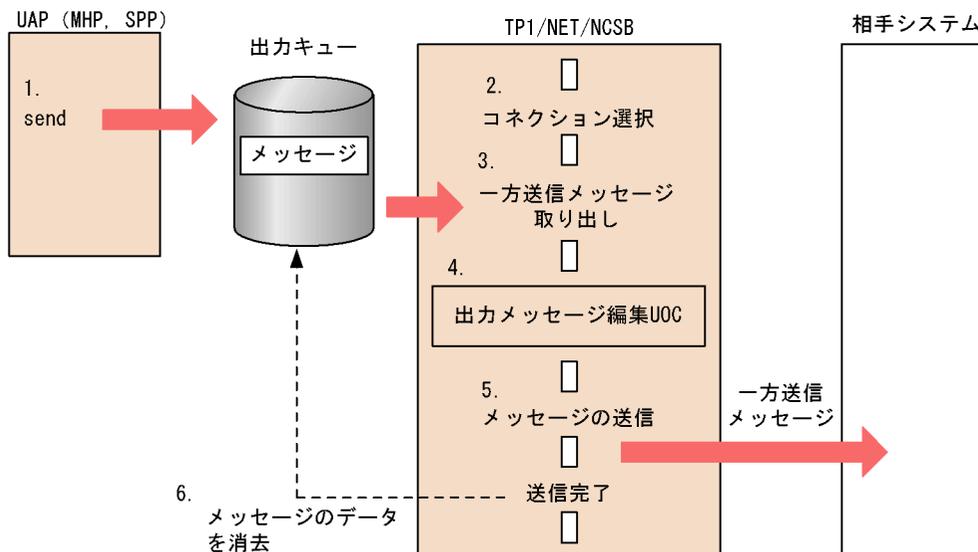
自システムの UAP から相手システムへメッセージを送る形態です。

TP1/NET/NCSB が、通信相手システムに一方送信メッセージを送信する手順を次に示します。

1. UAP は、SEND 文、または dc\_mcf\_send 関数で TP1/NET/NCSB にメッセージの送信を要求します。
2. コネクショングループ内の使用できる空きコネクションを選択します。
3. TP1/NET/NCSB は、出力キューに書き込まれたメッセージを取り出します。
4. 出力メッセージ編集 UOC でメッセージを編集します。
5. TP1/NET/NCSB は、相手システムへメッセージを送信します。
6. 送信が完了すると、TP1/NET/NCSB は出力キューにある送信済みのメッセージを消去します。

メッセージ送信の流れを次の図に示します。

図 2-7 メッセージ送信の流れ



## 2.2.2 一方送信メッセージの受信

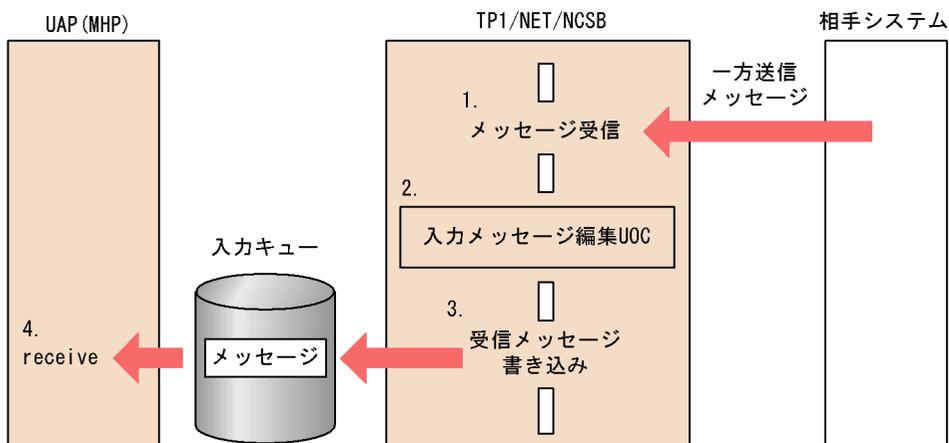
相手システムから一方的に送られたメッセージを受ける形態です。

TP1/NET/NC SB が、通信相手システムから一方送信メッセージを受信する手順を次に示します。

1. TP1/NET/NC SB が、相手システムから any 型論理端末あての一方送信メッセージを受信します。
2. 入力メッセージ編集 UOC でメッセージを編集します。
3. 受信したメッセージを入力キューに書き込みます。
4. UAP は、RECEIVE 文、または dc\_mcf\_receive 関数で受信メッセージを受け取ります。

メッセージ受信の流れを次の図に示します。

図 2-8 メッセージ受信の流れ





# 3

## メッセージ送受信インタフェース

ユーザは、相手システムとのメッセージを処理するための UAP を作成できます。この章では、UAP の作成方法、および作成例について説明します。

---

### メッセージ送受信インタフェースの一覧

---

dc\_mcf\_receive - 一方送信メッセージの受信 (C 言語)

---

dc\_mcf\_resend - メッセージの再送 (C 言語)

---

dc\_mcf\_send - 一方送信メッセージの送信 (C 言語)

---

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

---

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

---

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

---

RECEIVE - メッセージの受信 (データ操作言語)

---

SEND - 一方送信メッセージの送信 (データ操作言語)

---

ユーザアプリケーションプログラムの作成例

---

## メッセージ送受信インタフェースの一覧

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

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

### C 言語のメッセージ送受信

C 言語でメッセージを送受信する場合は、OpenTP1 システムで提供する関数を使用して UAP を作成します。

メッセージ送受信の関数を次の表に示します。

表 3-1 メッセージ送受信の関数 (C 言語)

関数名	機能
dc_mcf_receive	一方送信メッセージの受信
dc_mcf_resend	メッセージの再送
dc_mcf_send	一方送信メッセージの送信

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

### COBOL 言語のメッセージ送受信

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

メッセージ送受信の関数に対応するプログラムを次の表に示します。

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

プログラム名	データ名	機能
CBLDCMCF	'RECEIVE '	一方送信メッセージの受信
	'RESEND '	メッセージの再送
	'SEND '	一方送信メッセージの送信

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

### データ操作言語 (COBOL 言語) のメッセージ送受信

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

します。データ操作言語の形式の詳細については、マニュアル「OpenTP1 プログラム作成リファレンス COBOL 言語編」を参照してください。

TP1/NET/NCSB で固有の通信文を次の表に示します。

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 )

注

メッセージの再送 - CBLDCMCF(RESEND ) のデータ操作言語のインタフェースはありません。

## dc\_mcf\_receive - 一方送信メッセージの受信 (C 言語)

---

### 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

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

### 機能

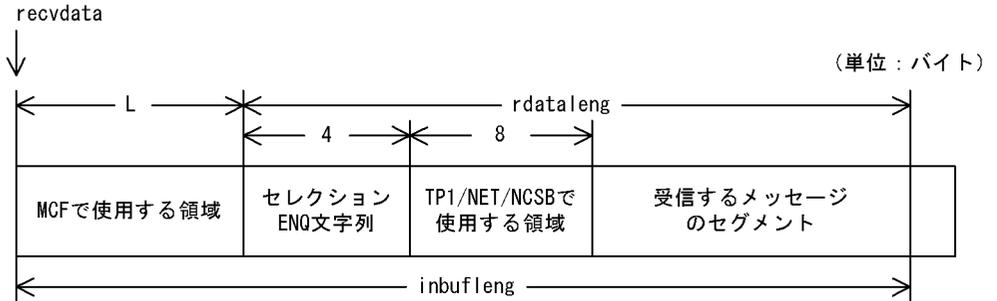
論理端末に届いたメッセージのうち、一つのセグメントを受信します。セグメントの数だけ dc\_mcf\_receive 関数を呼び出すと、一つの論理メッセージを受信できます。

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

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

先頭セグメントを受信する場合と、中間セグメントまたは最終セグメントを受信する場合の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。

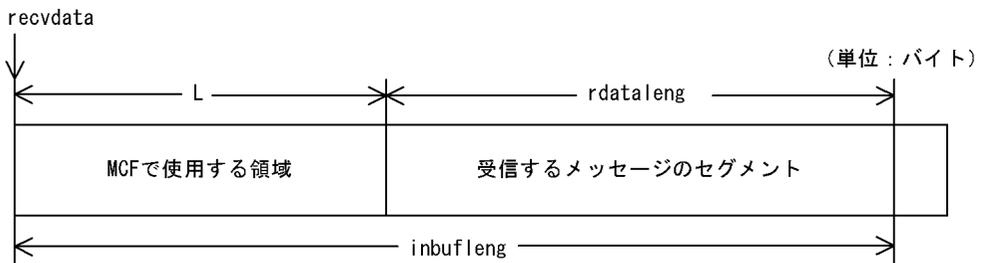
先頭セグメントの受信時



セレクション ENQ 文字列

通信管理から取得したセレクション文字列情報 (SA1, SA2, UA, ENQ の 4 バイト) が設定されます。

中間セグメントまたは最終セグメントの受信時



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、およびメッセージが単一セグメントの場合に設定します。

DCMCFSEG

中間セグメントまたは最終セグメントを受信する場合に設定します。

DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

### 3. メッセージ送受信インタフェース

dc\_mcf\_receive - 一方送信メッセージの受信 (C 言語)

DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

commform

DCNOFLAGS を設定します。

termnam

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

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

resv01

ヌル文字を設定します。

recvdata

セグメントを受信する領域を設定します。

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

処理終了後、recvdata には OpenTP1 から値が返ります。

inbufleng

セグメントを受信する領域の長さを設定します。

### OpenTP1 から値が返される引数

termnam

先頭セグメントを受信する場合だけ、入力元の論理端末名称が返されます。

中間セグメントまたは最終セグメントの受信時には、ここで返された論理端末名称を termnam に設定してください。

recvdata

受信したセグメントの内容が返されます。

rdataleng

受信したセグメントの長さが返されます。

time

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

## リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71000	-12000	先頭セグメントを受信する dc_mcf_receive 関数を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、action に DCMCFSEG を設定して dc_mcf_receive 関数を呼び出してください。
DCMCFRTN_71001	-12001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する dc_mcf_receive 関数を呼び出しています。直前に呼び出した dc_mcf_receive 関数でメッセージはすべて受信しました。このリターン値が返されたあとに、再び dc_mcf_receive 関数を呼び出した場合は、リターン値 DCMCFRTN_72000 が返されます。
DCMCFRTN_71002	-12002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。
DCMCFRTN_72000	-13000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none"> <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	-13001	termnam に設定した論理端末名称が間違っています。
DCMCFRTN_72013	-13013	inbufleng の指定値を超えるメッセージ (セグメント) を受信しました。inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRTN_72016	-13016	action に設定した値が間違っています。 resv01 に設定した値が間違っています。 引数に設定した値に間違いがあります。
DCMCFRTN_72024	-13024	commform に設定した値が間違っています。
DCMCFRTN_72025	-13025	action に設定したセグメント種別 (DCMCFRST または DCMCFSEG) の値が間違っています。
DCMCFRTN_72036	-13026	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。

### 3. メッセージ送受信インタフェース

dc\_mcf\_receive - 一方送信メッセージの受信 (C 言語)

(凡例)

- : 該当しません。

## dc\_mcf\_resend - メッセージの再送 (C 言語)

---

### 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

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

### 機能

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

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

対象としたメッセージが以前に送信されていない場合は、dc\_mcf\_resend 関数はリターン値 DCMCFRTN\_NOMSG を返します。また、メッセージキュー (ディスクキュー) 内に対象のメッセージがない場合も、リターン値 DCMCFRTN\_NOMSG を返します。このため、使用するメッセージキューの種別ではディスクキューを指定するとともに、メッセージキューファイルの容量および保持メッセージ数 (メッセージキューサービス定義の quegrp コマンドの -m オプションで指定) に余裕を持った値を指定してください。

## UAP で値を設定する引数

action

再送するセグメントに出力通番を付け直すかどうか、一般が優先かどうか、および最終出力通番のメッセージを再送するかどうかを、次の形式で設定します。

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

DCMCFSEQ

再送するメッセージに出力通番を付け直す場合に設定します。

DCMCFNSEQ

再送するメッセージに出力通番を付け直さない場合に設定します。

DCMCFNORM

一般の一方送信メッセージとして再送する場合に設定します。

DCMCFPRIO

優先の一方送信メッセージとして再送する場合に設定します。

DCMCFLAST

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

commform

一方送信を示す、DCMCFOUT を設定します。

rtermnam

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

resv01

ヌル文字を設定します。

oseqid

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

DCMCFRID\_NORM

一般の一方送信メッセージを対象とする場合に設定します。

DCMCFRID\_PRIO

優先の一方送信メッセージを対象とする場合に設定します。

省略した場合は、DCMCFRID\_NORM (一般の一方送信メッセージを対象) が設定されます。DCMCFRID\_PRIO を設定した場合は、otermnam に出力先の論理端末名称を設定できません。

orgseq

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

otermnam

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

resv02, resv03, resv04

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

## リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_0000	0	正常に終了しました。
DCMCFRTN_NO MSG	-11904	該当するメッセージがありません。
DCMCFRTN_BU F_SHORT	-11905	再送するメッセージのセグメントの長さが、UAP 共通定義 (mcfmuap -e) で指定した値を超えています。
DCMCFRTN_71002	-12002	メッセージキューへの出力処理中に障害が発生しました。
		メッセージキューが閉塞されています。
		メッセージキューが割り当てられていません。
		MCF が終了処理中のため、メッセージの再送を受け付けられません。
DCMCFRTN_71003	-12003	メッセージキューが満杯です。
DCMCFRTN_71004	-12004	メッセージキューから取り出したメッセージを格納するバッファを、メモリ上に確保できませんでした。
DCMCFRTN_71108	-12108	メッセージを再送しようとしたのですが、再送先の管理テーブルが確保できませんでした。
		プロセスのローカルメモリが不足しています。

### 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)

リターン値	リターン値 (数値)	意味
DCMCFRTN_720 00	-13000	< 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>
		< 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_720 01	-13001	rtermnam または otermnam に設定した論理端末名称が間違っています。
		dc_mcf_resend 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_720 16	-13016	action に設定したメッセージ種別 (DCMCFNORM または DCMCFPRIO) の値が間違っています。
		opcd に設定した値が間違っています。
		oseqid に設定した値が間違っています。
		resv01, resv02, resv03, または resv04 に設定した値が間違っています。
		引数に設定した値に間違いがあります。
DCMCFRTN_720 17	-13017	action に設定した出力通番の要否 (DCMCFSEQ または DCMCFNSEQ) の値が間違っています。
DCMCFRTN_720 24	-13024	commform に設定した値が間違っています。
上記以外	-	プログラムの破壊などによる, 予期しないエラーが発生しました。

(凡例)

- : 該当しません。

#### 注意事項

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

##### -e オプション

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

##### -l オプション

-l オプションでは, 通番に関して指定します。指定内容によっては, メッセージ

キューファイル内に同じ通番を持つメッセージが同時に存在する場合があります。同じ通番を持つメッセージが存在する場合、どのメッセージを再送するかは保証できません。

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

### 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

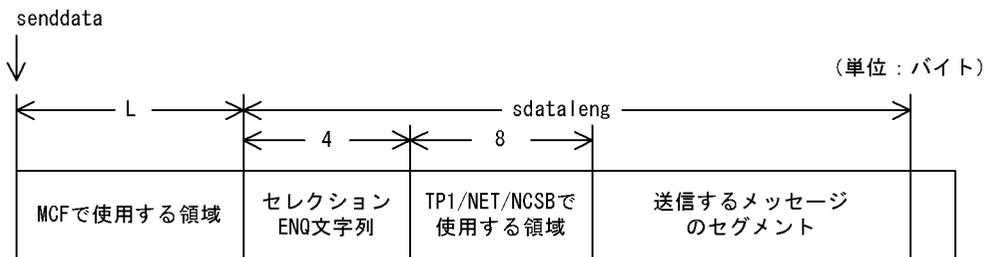
```
#include <dcmcf.h>
int dc_mcf_send(action,commform,termnam,resv01,senddata,sdataleng,
               resv02,opcd)
DCLONG      action;
DCLONG      commform;
char        *termnam;
char        *resv01;
char        *senddata;
DCLONG      sdataleng;
char        *resv02;
DCLONG      opcd;
```

### 機能

相手システムへ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。先頭セグメントを送信する場合は、セグメント内に通信回線の情報を付加します。

先頭セグメントを送信する場合と、中間セグメントまたは最終セグメントを送信する場合の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。

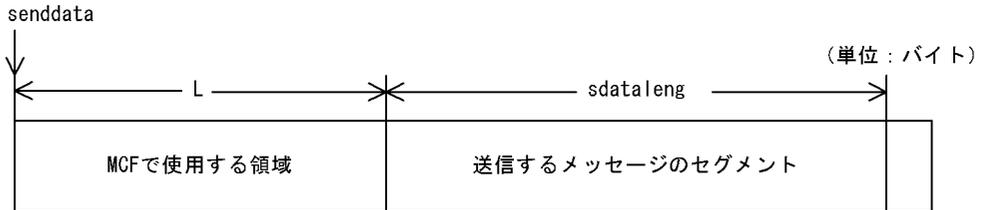
先頭セグメントの送信時



セレクション ENQ 文字列

通信管理に指示するセレクション文字列情報 (SA1, SA2, UA, ENQ の 4 バイト) を設定します。

中間セグメントまたは最終セグメントの送信時



## 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、一般が優先かどうか、出力通番を付けるかどうか、および使用するバッファ形式を、次の形式で設定します。

```
{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、およびメッセージが単一セグメントの場合に設定します。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。

### 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)

commform

一方送信を示す, DCMCFOUT を設定します。

termnam

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

resv01

ヌル文字を設定します。

senddata

送信するメッセージのセグメントの内容を設定した領域を設定します。メッセージの送信の終了を連絡する場合で、セグメントの内容がないときも、必ず設定してください。

sdataleng

送信するメッセージのセグメントの長さを設定します。先頭セグメントの場合は 1010 バイト以下、第 2 セグメント以降では 446 バイト以下の値を設定してください。

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場合は、0 を設定します。

resv02

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

### リターン値

リターン値	リターン値 (数値)	意味
DCMCFRT N_00000	0	正常に終了しました。
DCMCFRT N_71002	-12002	メッセージキューへの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。
DCMCFRT N_71003	-12003	メッセージキューが満杯です。

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	リターン値 (数値)	意味
DCMCFRT N_71004	-12004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRT N_71108	-12108	メッセージを送信しようとしたのですが、送信先の管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
DCMCFRT N_72000	-13000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、 dc_mcf_send 関数を呼び出しています。  < SPP の実行でリターンした場合 > トランザクションでない SPP の処理から、dc_mcf_send 関数を呼び 出しています。
DCMCFRT N_72001	-13001	termnam に設定した論理端末名称が間違っています。  termnam に設定した出力先の論理端末名称は、MCF で定義してい ません。  dc_mcf_send 関数を呼び出せない論理端末を設定しています。
DCMCFRT N_72005	-13005	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。
DCMCFRT N_72016	-13016	action に設定したメッセージ種別 (DCMCFNORM または DCMCFPRIO) の値が間違っています。  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。  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。
DCMCFRT N_72017	-13017	action に設定した出力通番の要否 (DCMCFSEQ または DCMCFNSEQ) の値が間違っています。
DCMCFRT N_72024	-13024	commform に設定した値が間違っています。
DCMCFRT N_72026	-13026	action に設定したセグメント種別 (DCMCFESI または DCMCFEMI) の値が間違っています。
DCMCFRT N_72041	-13041	単一セグメントの送信時に、sdataleag に 0 またはマイナス値を設定 しています。
上記以外	-	プログラムの破壊などによる、予期しないエラーが発生しました。

( 凡例 )

- : 該当しません。

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

---

### 形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
  02 データ名A PIC X(8) VALUE 'RECEIVE'.  
  02 データ名B PIC X(5).  
  02 FILLER PIC X(3).  
  02 データ名C PIC X(4).  
  02 データ名D PIC X(4) VALUE SPACE.  
  02 データ名E PIC 9(8).  
  02 データ名F PIC 9(8).  
  02 データ名G PIC 9(9) COMP.  
  02 データ名H PIC X(4) VALUE SPACE.  
  02 データ名I PIC X(4) VALUE SPACE.  
  02 データ名J PIC X(4) VALUE SPACE.  
  02 データ名K PIC X(4) VALUE SPACE.  
  02 データ名L PIC X(8) VALUE SPACE.  
  02 データ名M1 PIC X(4) VALUE SPACE.  
  02 データ名M2 PIC X(8) VALUE SPACE.  
  02 データ名M3 PIC X(4) VALUE SPACE.  
  02 データ名M4 PIC 9(9) COMP VALUE ZERO.  
  02 データ名M5 PIC 9(9) COMP VALUE ZERO.  
  02 データ名M6 PIC X(1) VALUE SPACE.  
  02 データ名M7 PIC X(1).  
  02 データ名N PIC X(14) VALUE LOW-VALUE.  
01 一意名2.  
  02 データ名O PIC X(4) VALUE SPACE.  
  02 データ名P PIC X(8).  
  02 データ名Q PIC X(8).  
  02 データ名R PIC X(8) VALUE SPACE.  
  02 データ名T PIC X(28) VALUE LOW-VALUE.  
01 一意名3.  
  02 データ名U PIC 9(x) COMP.  
  02 データ名V PIC X(x).  
  02 データ名W.  
    03 データ名W1 PIC X(4).  
    03 データ名W2 PIC X(8).  
    03 データ名W3 PIC X(n).  
  02 データ名X REDEFINES データ名W PIC X(n).
```

### 機能

論理端末に届いたメッセージのうち、一つのセグメントを受信します。セグメントの数

だけ CBLDCMCF('RECEIVE') を呼び出すと、一つの論理メッセージを受信できません。

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

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

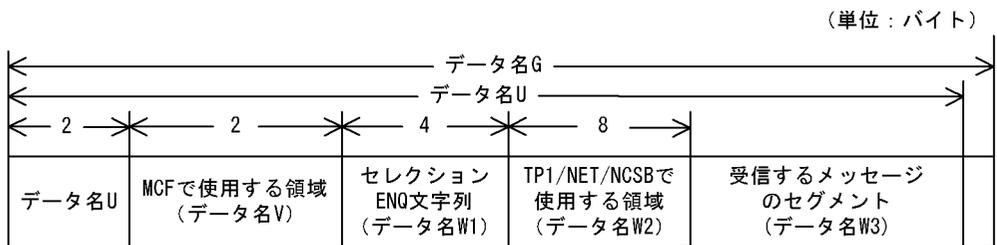
先頭セグメントを受信する場合と、中間セグメントまたは最終セグメントを受信する場合の領域 (一意名 3 で示す領域) の形式を次に示します。

先頭セグメントの受信時

バッファ形式 1 の場合

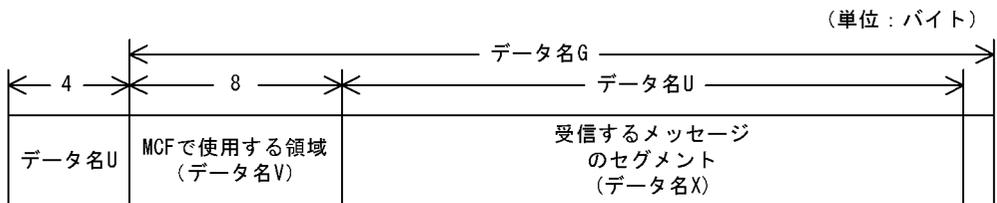


バッファ形式 2 の場合



中間セグメントまたは最終セグメントの受信時

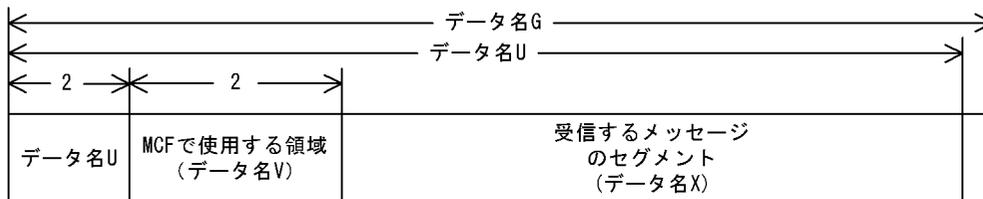
バッファ形式 1 の場合



バッファ形式 2 の場合

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

(単位: バイト)



UAP で値を設定するデータ領域

データ名 A

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

データ名 C

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

VALUE 'FRST'

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

VALUE 'SEG」

中間セグメントまたは最終セグメントを受信する場合に設定します。

データ名 D

空白を設定します。

データ名 G

セグメントを受信する領域の長さを設定します。

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

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白

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

データ名 N

MCF で使用する領域です。

データ名 O

空白を設定します。

データ名 P

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

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

データ名 Q

MCF で使用する領域です。

データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 V

【バッファ形式 1 の場合】PIC X(8)

【バッファ形式 2 の場合】PIC X(2)

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

データ名 B

ステータスコードが、5 けたの数字で返されます。

データ名 E

メッセージを受信した日付が YYYYMMDD (YYYY: 西暦年 MM: 月 DD: 日) の形

### 3. メッセージ送受信インタフェース

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

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 W1

先頭セグメントを受信する場合に、通信管理から取得したセレクション文字列情報 (SA1, SA2, UA, ENQ の 4 バイト) が返されます。

データ名 W2

先頭セグメントを受信する場合に、TP1/NET/NCSB で使用する領域です。

データ名 W3

先頭セグメントを受信する場合に、受信したセグメントが返されます。

データ名 X

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、受信したセグメントが返されます。

#### 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') - 一方送信メッセージの受信 (COBOL 言語)

ステータスコード	意味
71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する CBLDCMCF('RECEIVE') を呼び出しています。直前に呼び出した CBLDCMCF('RECEIVE') でメッセージはすべて受信しました。このステータスコードが返されたあとに、再び CBLDCMCF('RECEIVE') を呼び出した場合は、ステータスコード 72000 が返されます。
71002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。
71108	プロセスのローカルメモリが不足しています。
72000	<p>&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>&lt; バッファ形式 2 の場合 &gt;</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 M7 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>
72024	データ名 O に設定した値が間違っています。
72025	データ名 C に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72036	データ名 G の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

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

---

### 形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

### 機能

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

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

対象としたメッセージが以前に送信されていない場合は, CBLDCMCF('RESEND') はステータスコード 70904 を返します。また, メッセージキュー (ディスクキュー) 内に対象のメッセージがない場合も, ステータスコード 70904 を返します。このため, 使用するメッセージキューの種別ではディスクキューを指定するとともに, メッセージキューファイルの容量および保持メッセージ数 (メッセージキューサービス定義の quegrp コマンドの -m オプションで指定) に余裕を持った値を指定してください。

## UAP で値を設定するデータ領域

データ名 A

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

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H, データ名 I

空白を設定します。

データ名 J

一般として再送するか優先として再送するかを設定します。

VALUE 'NORM'

一般の一方送信メッセージとして再送する場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージとして再送する場合に設定します。

空白

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

データ名 K

再送するメッセージに出力通番を付け直すかどうかを設定します。

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

VALUE 'SEQ' '

再送するメッセージに出力通番を付け直す場合に設定します。

VALUE 'NSEQ'

再送するメッセージに出力通番を付け直さない場合に設定します。

空白

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

データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6, データ名 M7

空白を設定します。

データ名 N

MCF で使用する領域です。

データ名 O

一方送信を示す「VALUE 'OUT'」を設定します。

データ名 P

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

データ名 Q, データ名 R

空白を設定します。

データ名 S

MCF で使用する領域です。

データ名 T

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

データ名 U

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

定します。

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になります。

データ名 W

最終出力通番を持つメッセージを再送するかどうかを設定します。

VALUE 'LAST'

最終出力通番を持つメッセージを再送する場合に設定します。この値を設定した場合、データ名 V に設定した値は無効になります。

空白

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

データ名 X

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

データ名 B

ステータスコードが、5けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
70904	該当するメッセージがありません。

### 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
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。
	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 オプションおよび -l オプションの指定値に注意してください。

#### -e オプション

-e オプションでは、CBLDCMCF('RESEND') で使用する作業領域の大きさを

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

#### -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、メッセージキューファイル内に同じ通番を持つメッセージが同時に存在する場合があります。同じ通番を持つメッセージが存在する場合、どのメッセージを再送するかは保証できません。

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

---

### 形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
  02 データ名A PIC X(8) VALUE 'SEND ' .  
  02 データ名B PIC X(5) .  
  02 FILLER PIC X(3) .  
  02 データ名C PIC X(4) VALUE SPACE .  
  02 データ名D PIC X(4) VALUE SPACE .  
  02 データ名E PIC 9(8) .  
  02 データ名F PIC 9(8) .  
  02 データ名G PIC 9(9) COMP VALUE ZERO .  
  02 データ名H PIC X(4) .  
  02 データ名I PIC X(4) VALUE SPACE .  
  02 データ名J PIC X(4) .  
  02 データ名K PIC X(4) .  
  02 データ名L PIC X(8) VALUE SPACE .  
  02 データ名M1 PIC X(4) VALUE SPACE .  
  02 データ名M2 PIC X(8) VALUE SPACE .  
  02 データ名M3 PIC X(4) VALUE SPACE .  
  02 データ名M4 PIC 9(9) COMP VALUE ZERO .  
  02 データ名M5 PIC 9(9) COMP VALUE ZERO .  
  02 データ名M6 PIC X(1) VALUE SPACE .  
  02 データ名M7 PIC X(1) .  
  02 データ名N PIC X(14) VALUE LOW-VALUE .  
01 一意名2.  
  02 データ名O PIC X(4) VALUE 'OUT ' .  
  02 データ名P PIC X(8) .  
  02 データ名Q PIC X(8) VALUE SPACE .  
  02 データ名R PIC X(8) VALUE SPACE .  
  02 データ名T PIC X(28) VALUE LOW-VALUE .  
01 一意名3.  
  02 データ名U PIC 9(x) COMP .  
  02 データ名V PIC X(x) .  
  02 データ名W .  
    03 データ名W1 PIC X(4) .  
    03 データ名W2 PIC X(8) .  
    03 データ名W3 PIC X(n) .  
  02 データ名X REDEFINES データ名W PIC X(n) .
```

### 機能

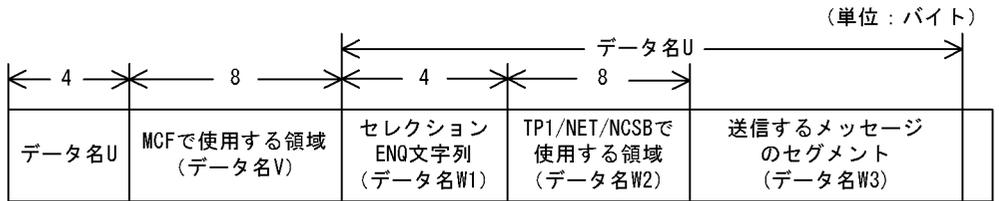
相手システムへ送る一方送信メッセージのうち、一つのスグメントを送信します。スグ

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。先頭セグメントを送信する場合は、セグメント内に通信回線の情報を付加します。

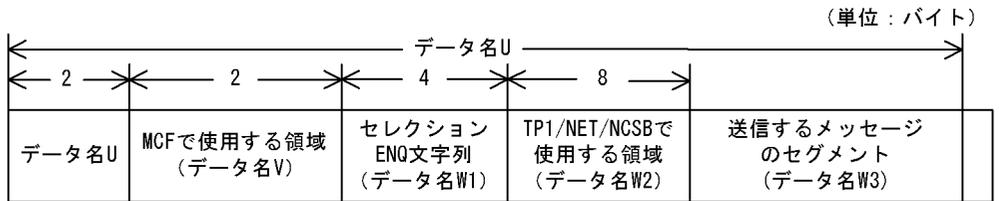
先頭セグメントを送信する場合と、中間セグメントまたは最終セグメントを送信する場合の領域（一意名 3 で示す領域）の形式を次に示します。

先頭セグメントの送信時

バッファ形式 1 の場合



バッファ形式 2 の場合

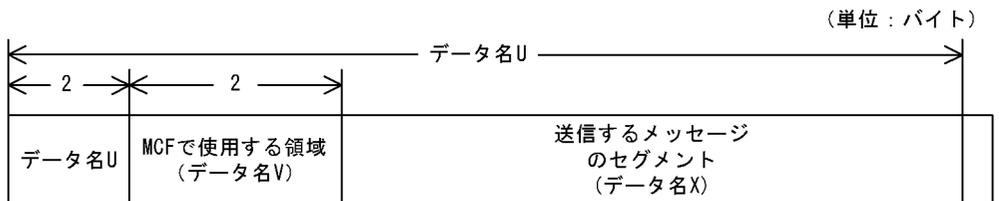


中間セグメントまたは最終セグメントの送信時

バッファ形式 1 の場合



バッファ形式 2 の場合



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)

### UAP で値を設定するデータ領域

データ名 A

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

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H

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。次のどちらかの値を設定します。

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、およびメッセージが単一セグメントの場合に設定します。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ます。

データ名 I

空白を設定します。

データ名 J

一般として送信するか優先として送信するかを設定します。

VALUE 'NORM'

一般の一方送信メッセージとして送信する場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージとして送信する場合に設定します。

空白

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

データ名 K

出力通番を付けるかどうかを設定します。

VALUE 'SEQ'

出力通番が必要な場合に設定します。

VALUE 'NSEQ'

出力通番が必要ない場合に設定します。

空白

省略されたものとして、「VALUE 'NSEQ」(出力通番は必要ない) が設定されます。

データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白

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

データ名 N

MCF で使用する領域です。

データ名 O

一方送信を示す「VALUE 'OUT」を設定します。

データ名 P

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

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 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。先頭セグメントの場合は 1010 バイト以下、中間セグメントまたは最終セグメントの場合は 446 バイト以下の値を設定してください。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。先頭セグメントの場合は 1014 バイト以下、中間セグメントまたは最終セグメントの場合は 450 バイト以下の値を設定してください。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 W1

先頭セグメントを送信する場合に、通信管理に指示するセレクション文字列情報 (SA1, SA2, UA, ENQ の 4 バイト) を設定します。

データ名 W2

先頭セグメントを送信する場合に、TP1/NET/NCSB で使用する領域です。

データ名 W3

先頭セグメントを送信する場合に、送信するメッセージのセグメントを設定します。

データ名 X

中間セグメントまたは最終セグメントを送信する場合に、メッセージのセグメントを設定します。

## OpenTP1 から値が返されるデータ領域

データ名 B

ステータスコードが、5 けたの数字で返されます。

### ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	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	先頭セグメントまたは中間セグメントを送信する CBLDCMCF('SEND') で、 長さが 0 バイトのセグメントを送信しています。
72016	データ名 J に設定した値が間違っています。
	データ名 M1 に設定した値が間違っています。
	データ名 M7 に設定した値が間違っています。
	データ名 N またはデータ名 T に設定した値が間違っています。
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
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。

### 3. メッセージ送受信インタフェース

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

ステータス コード	意味
72041	バッファ形式 1 の場合はデータ名 U に 0, またはマイナスの値を設定しています。バッファ形式 2 の場合はデータ名 U に 4 以下, またはマイナスの値を設定しています。
上記以外	プログラムの破壊などによる, 予期しないエラーが発生しました。

## RECEIVE - メッセージの受信 (データ操作言語)

---

### 形式

DATA DIVISION (通信記述項) の指定

CD 通信記述名

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

PROCEDURE DIVISION (通信文) の指定

RECEIVE 通信記述名

```
[FIRST] SEGMENT  
INTO 一意名1.
```

### 機能

次に示す CALL インタフェースの機能を実現します。

- 一方送信メッセージの受信 CBLDCMCF('RECEIVE ')

### 通信記述項に設定する項目

FOR 句

次の値を指定します。

INPUT

一方送信メッセージの受信

STATUS KEY 句

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

SYMBOLIC TERMINAL 句

入力元の論理端末名称を参照するデータ項目を指定します。

MESSAGE DATE 句

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

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 TIME 句

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

通信文に指定する項目

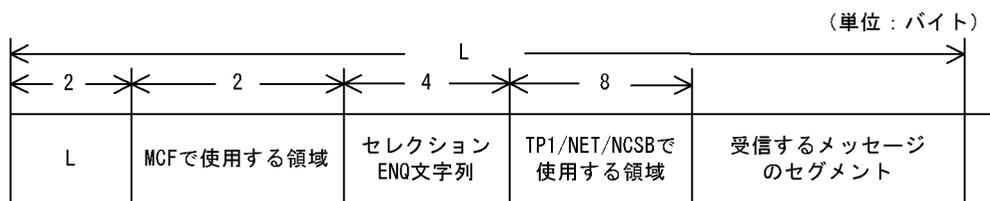
FIRST

先頭セグメントを受信する場合に指定します。

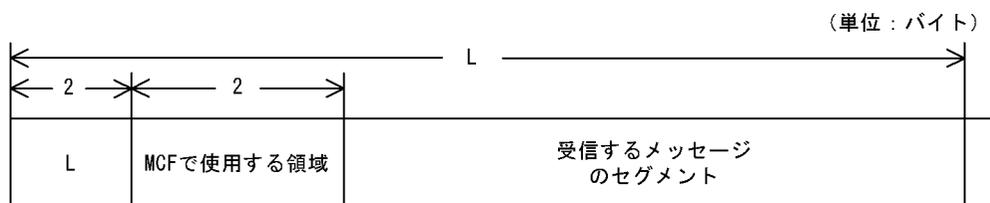
一意名 1

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

先頭セグメントの受信時



中間セグメントまたは最終セグメントの受信時



ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する RECEIVE 文を、2 回以上実行しています。中間セグメントまたは最終セグメントを受信する場合は、FIRST を指定しないで RECEIVE 文を実行してください。
71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する RECEIVE 文を実行しています。直前に実行した RECEIVE 文でメッセージはすべて受信しました。このステータスコードが返されたあとに、再び RECEIVE 文を実行した場合は、ステータスコード 72000 が返されます。
71002	メッセージキューからの入力処理中に障害が発生しました。
	メッセージキューが閉塞されています。

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
	メッセージキューが割り当てられていません。
	MCF が終了処理中のため、メッセージの受信を受け付けられません。
71108	メッセージ受信に必要な管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none"> <li>先頭セグメントを受信する RECEIVE 文を実行する前に、中間セグメントまたは最終セグメントを受信する RECEIVE 文を実行しています。先頭セグメントを受信する場合は、FIRST を指定して RECEIVE 文を実行してください。</li> <li>ステータスコード 71001 が返されたあとで、RECEIVE 文を実行しています。</li> </ul>
	< SPP の実行でリターンした場合 > SPP では、RECEIVE 文を実行できません。
72001	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。
	RECEIVE 文を実行できない論理端末を設定しています。
72013	一意名 1 の L の指定値を超えるセグメントを受信しました。一意名 1 の L の指定値を超えた部分は切り捨てられました。
	32763 バイトを超えるセグメントを受信しました。32763 バイトを超えた部分は切り捨てられました。
72016	WAITING TIME 句は設定できません。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72036	一意名 1 の L の長さが不足しています。5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

## SEND - 一方送信メッセージの送信 (データ操作言語)

---

形式 1 (セグメントの内容を設定して送信する場合)

DATA DIVISION (通信記述項) の指定

CD 通信記述名

FOR OUTPUT

{STATUS KEY IS データ名1}

{SYMBOLIC TERMINAL IS データ名2}

{SYNCHRONOUS MODE IS {ASYNC | データ名6}}

{SWITCHING MODE IS {NORMAL | PRIOR | データ名7}}

{DETAIL MODE IS データ名10}.

PROCEDURE DIVISION (通信文) の指定

SEND 通信記述名 FROM 一意名1

{WITH {ESI | EMI | 一意名2}}.

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

DATA DIVISION (通信記述項) の指定

CD 通信記述名

FOR OUTPUT

{STATUS KEY IS データ名1}

{SYMBOLIC TERMINAL IS データ名2}

{SWITCHING MODE IS {NORMAL | 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                   ')

### 通信記述項に設定する項目

FOR 句

次の項目を指定します。

OUTPUT

#### 一方送信メッセージの送信

##### STATUS KEY 句

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

##### SYMBOLIC TERMINAL 句

論理端末名称を設定したデータ項目を指定します。

##### SYNCHRONOUS MODE 句

次のどちらかの値を指定します。

##### ASYN

##### 一方送信メッセージの送信

##### データ名 6

次の値を設定したデータ項目

'0': 一方送信メッセージの送信

SYNCHRONOUS MODE 句は、省略できます。

##### SWITCHING MODE 句

一般か優先かを指定します。

##### NORMAL

一般の一方送信メッセージ

##### PRIOR

優先の一方送信メッセージ

##### データ名 7

次の値を設定したデータ項目

'0' または ' ': 一般の一方送信メッセージ

'1': 優先の一方送信メッセージ

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

##### DETAIL MODE 句

出力通番を付けるかどうかを指定します。

##### データ名 10

次の値を設定したデータ項目

'0' または ' ': 出力通番を付けます。

'1': 出力通番を付けません。

省略した場合は、出力通番を付けません。

3. メッセージ送受信インタフェース  
 SEND - 一方送信メッセージの送信 (データ操作言語)

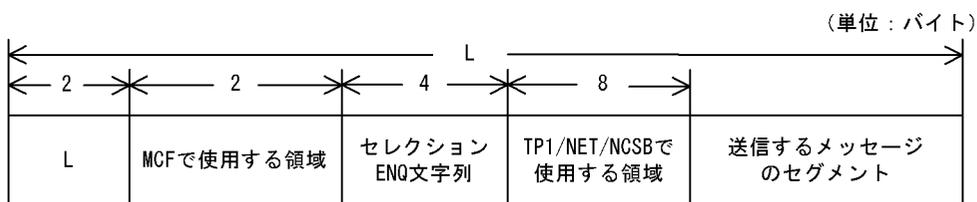
通信文に指定する項目

一意名 1

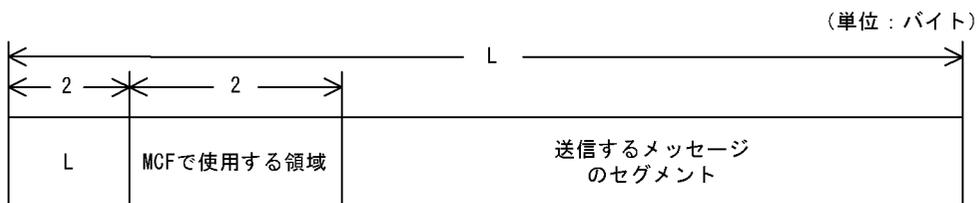
セグメントを送信するデータ項目を指定します。先頭セグメントの場合は 1014 バイト以下、中間セグメントまたは最終セグメントの場合は 450 バイト以下の値を設定してください。

一意名 1 の形式を次に示します。

先頭セグメントの送信時



中間セグメントまたは最終セグメントの送信時



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。

ESI

先頭セグメントまたは中間セグメントの送信

EMI

最終セグメントの送信

メッセージが単一セグメントの場合も EMI を指定します。

一意名 2

次の値を設定したデータ項目

'1': ESI (先頭セグメントまたは中間セグメントの送信)

'2': EMI (最終セグメントまたは単一セグメントの送信)

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

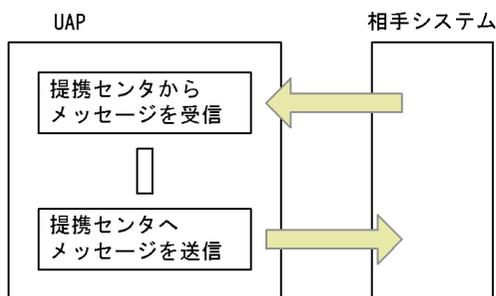
## ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	一意名 1 の L に 32000 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する RECEIVE 文を実行する前に、SEND 文を実行しています。
	< SPP の実行でリターンした場合 > トランザクションでない SPP の処理から、SEND 文を実行しています。
72001	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。
	SEND 文を実行できない論理端末を設定しています。
72005	先頭セグメントまたは中間セグメントを送信する SEND 文で、長さが 0 バイトのセグメントを送信しています。
72008	応答メッセージ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。
	SEND 文を実行して応答型のアプリケーションを起動させたあとに、応答メッセージを送信する SEND 文を実行しています。
72013	一意名 1 の L を超えるセグメントを受信しました。一意名 1 の L を超えた部分は切り捨てられました。
	32763 バイトを超えるセグメントを受信しました。32763 バイトを超えた部分は切り捨てられました。
72017	DETAIL MODE 句に設定した値が間違っています。
72018	SWITCHING MODE 句に設定した値が間違っています。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72026	WITH 句に設定した値が間違っています。
72036	一意名 3 の長さが不足しています。5 バイト以上の領域を確保してください。
72041	一意名 1 の L に 4 以下、またはマイナスの値を設定しています。
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の作成例

UAP の作成例の処理の流れを次の図に示し、その流れに沿ったコーディング例を C 言語、COBOL 言語、およびデータ操作言語別に示します。

図 3-1 UAP の作成例の処理の流れ



### C 言語

C 言語を使用した UAP のコーディング例を次に示します。

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

```
#include <dcmcf.h>

typedef struct senddata_s {
    char    mcf_area[8];
    char    enq_area[4];
    char    ncsb_area[8];
    char    seg_area[998];
} senddata_def;

ncsb_uap1()
{
    /* 引数定義 */
    char    termnam[9];
    static char    resv01[9] = "¥0";
    senddata_def    sdataarea;
    senddata_def    rdataarea;
    DCLONG    sdataleng;
    DCLONG    rdataleng;
    DCLONG    time;
    static char    resv02[9] = "¥0";
    int    rtn_code;

    memset(resv01, 0, 9);
    memset(resv02, 0, 9);
    memset(sdataarea.ncsb_area, 0, 8);
    memset(rdataarea.ncsb_area, 0, 8);
    memset(rdataarea.seg_area, 0, 998);
```

```

memset(sdataarea.seg_area, 0, 998);
strcpy(termnam, "INTLE01");

/* メッセージ受信開始 receive関数の呼び出し */

rtn_code = dc_mcf_receive( DCMCFRST, DCNOFLAGS, termnam,
                          resv01, &rdataarea, &rdataleng,
                          1018, &time );

/**** データの処理 ****/

/* メッセージ送信開始 send関数の呼び出し */

memcpy(sdataarea.enq_area, rdataarea.enq_area, 4);
sdataleng = rdatdaleng;

rtn_code = dc_mcf_send( DCMCFEMI, DCMCFOUT, termnam, resv01,
                       &sdataarea, sdataleng, resv02,
                       DCNOFLAGS );
if( rtn_code != DCMCFRTN_00000 )
{
    goto ERROR;
}
ERROR: ;
    return;
}

```

## COBOL 言語

COBOL 言語を使用した UAP のコーディング例を次に示します。

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

```

IDENTIFICATION DIVISION.
PROGRAM-ID. EXUAP1.

ENVIRONMENT DIVISION.

DATA DIVISION.
WORKING-STORAGE SECTION.

01 RCV.
   02 MSG-REC PIC X(8) VALUE 'RECEIVE'.
   02 STATUS-CODE1 PIC X(5).
   02 FILLER PIC X(3).
   02 SEG-CODE PIC X(4) VALUE 'FIRST'.
   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 1018.
   02 MCFUSE01 PIC X(4) VALUE SPACE.
   02 MCFUSE02 PIC X(4) VALUE SPACE.
   02 MCFUSE03 PIC X(4) VALUE SPACE.
   02 MCFUSE04 PIC X(4) VALUE SPACE.
   02 MCFUSE05 PIC X(8) VALUE SPACE.

```

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の作成例

```

02 MCFUSE06          PIC X(4)    VALUE  SPACE.
02 MCFUSE07          PIC X(8)    VALUE  SPACE.
02 MCFUSE08          PIC X(4)    VALUE  SPACE.
02 MCFUSE09          PIC 9(9)    COMP  VALUE  ZERO.
02 MCFUSE10          PIC 9(9)    COMP  VALUE  ZERO.
02 MCFUSE11          PIC X(1)    VALUE  SPACE.
02 MCFUSE12          PIC X(1)    VALUE  SPACE.
02 MCFUSE13          PIC X(14)   VALUE  LOW-VALUE.
01 CD1.
02 SEG-CODE1         PIC X(4)    VALUE  SPACE.
02 TERM-CODE         PIC X(8)    .
02 MCFUSE14          PIC X(8)    VALUE  SPACE.
02 MCFUSE15          PIC X(8)    VALUE  SPACE.
02 MCFUSE16          PIC X(28)   VALUE  LOW-VALUE.
01 DATA1.
02 MSGSEG-LENG1     PIC 9(9)    COMP.
02 MCFUSE17         PIC X(8)    .
02 MCFUSE18.
03 SELENQD1         PIC X(4)    .
03 MCFUSE20         PIC X(8)    .
03 REC-MSGSEG1     PIC X(998)  .
01 SND.
02 MSG-SEND1        PIC X(8)    VALUE  'SEND  '.
02 STATUS-CODE3     PIC X(5)    .
02 FILLER           PIC X(3)    .
02 MCFUSE30         PIC X(4)    VALUE  SPACE.
02 MCFUSE31         PIC X(4)    VALUE  SPACE.
02 MCFUSE32         PIC 9(8)    .
02 MCFUSE33         PIC 9(8)    .
02 MCFUSE34         PIC 9(9)    COMP  VALUE  ZERO.
02 SEND-SEG1        PIC X(4)    VALUE  'EMI  '.
02 SEND-SYNC1       PIC X(4)    VALUE  SPACE.
02 SEND-NORM1       PIC X(4)    VALUE  'NORM'.
02 SEND-SEQ1        PIC X(4)    VALUE  'NSEQ'.
02 MCFUSE35         PIC X(8)    VALUE  SPACE.
02 SEND-CODE1       PIC X(4)    VALUE  SPACE.
02 MCFUSE36         PIC X(8)    VALUE  SPACE.
02 MCFUSE37         PIC X(4)    VALUE  SPACE.
02 MCFUSE38         PIC 9(9)    COMP  VALUE  ZERO.
02 MCFUSE39         PIC 9(9)    COMP  VALUE  ZERO.
02 MCFUSE40         PIC X(1)    VALUE  SPACE.
02 MCFUSE41         PIC X(1)    VALUE  SPACE.
02 MCFUSE42         PIC X(14)   VALUE  LOW-VALUE.
01 CD2.
02 SENDSEG-CODE2    PIC X(4)    VALUE  'OUT  '.
02 TERM-CODE         PIC X(8)    VALUE  'TERMNAM1'.
02 MCFUSE50         PIC X(8)    VALUE  SPACE.
02 MCFUSE51         PIC X(8)    VALUE  SPACE.
02 MCFUSE52         PIC X(28)   VALUE  LOW-VALUE.
01 DATA2.
02 MSGSEG-LENG2     PIC 9(9)    COMP VALUE 1010.
02 MCFUSE60         PIC X(8)    .
02 MCFUSE61.
03 SELENQD2         PIC X(4)    .
03 MCFUSE62         PIC X(8)    .
03 REC-MSGSEG2     PIC X(998)  .

```

```
PROCEDURE DIVISION.  
CALL 'CBLDCMCF' USING RCV CD1 DATA1.  
MOVE SELENQD1 TO SELENQD2.  
**** データの処理 ****  
CALL 'CBLDCMCF' USING SND CD2 DATA2.  
EXIT PROGRAM.
```

## データ操作言語

データ操作言語を使用した UAP のコーディング例を次に示します。

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

```
**** MHPサービスプログラム ****  
*  
IDENTIFICATION DIVISION.  
  
PROGRAM-ID. SVRA.  
  
ENVIRONMENT DIVISION.  
CONFIGURATION SECTION.  
*  
**** ワーク変数 ****  
*  
DATA DIVISION.  
WORKING-STORAGE SECTION.  
*  
**** 一方受信領域 ****  
*  
01 RECV-AREA.  
02 RE-DATALENG PIC 9(4) COMP VALUE 1014.  
02 RE-RSV1 PIC X(2).  
02 RE-MCFUSE1.  
03 RE-SELENQ PIC X(4).  
03 RE-MCFUSE2 PIC X(8).  
03 RE-DATA PIC X(998).  
  
*  
**** 一方送信領域 ****  
*  
01 SEND-IO-AREA.  
02 IO-DATALENG PIC 9(4) COMP.  
02 IO-RSV1 PIC X(2).  
02 IO-MCFUSE1.  
03 IO-SELENQ PIC X(4).  
03 IO-MCFUSE2 PIC X(8).  
03 IO-DATA PIC X(998).  
  
*  
**** 通信記述項 ****
```

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

```
*
  COMMUNICATION SECTION.
*
**** 一方受信 ****
*
      CD  RECV-INF
          FOR  INPUT
          STATUS KEY IS          RE-STATUS
          SYMBOLIC TERMINAL IS  RE-TERMNAM
          MESSAGE DATE IS       RE-DATE
          MESSAGE TIME IS       RE-TIME.
*
**** 一方送信 ****
*
      CD  SEND-IO
          FOR  OUTPUT
          STATUS KEY IS          SE-STATUS-IO
          SYMBOLIC TERMINAL IS  SE-TERMNAM-IO
*
**** 通信文 ****
*
  PROCEDURE DIVISION.
*
**** 一方受信 ****
*
      RECEIVE  RECV-INF
              FIRST  SEGMENT
              INTO  RECV-AREA.
*
*
**** データの処理 ****
*
      MOVE 'TERMNAM1' TO SE-TERMNAM-IO.
      MOVE RE-SELENQ TO IO-SELENQ.
      SEND SEND-IO
          FROM SEND-IO-AREA
          WITH EMI.
*
  EXIT PROGRAM.
```

# 4

## ユーザOWNコーディング， MCF イベントインタフェース

この章では，TP1/NET/NCSB 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/NCSB で使用できる UOC を次に示します。

- 入力メッセージ編集 UOC
- 出力メッセージ編集 UOC
- 送信メッセージの迂回判定 UOC
- 送信メッセージの通番編集 UOC

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

### 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。また，受信した論理メッセージを基に，ユーザ任意のアプリケーション名を決定できます。

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

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

#### （1）入力メッセージの編集

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

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

#### （2）アプリケーション名の決定

論理メッセージの内容を基に，ユーザ任意のアプリケーション名を決定できます。

UOC でアプリケーション名を決定しなかった場合，MCF 通信構成定義（`mcftalcle -v`）

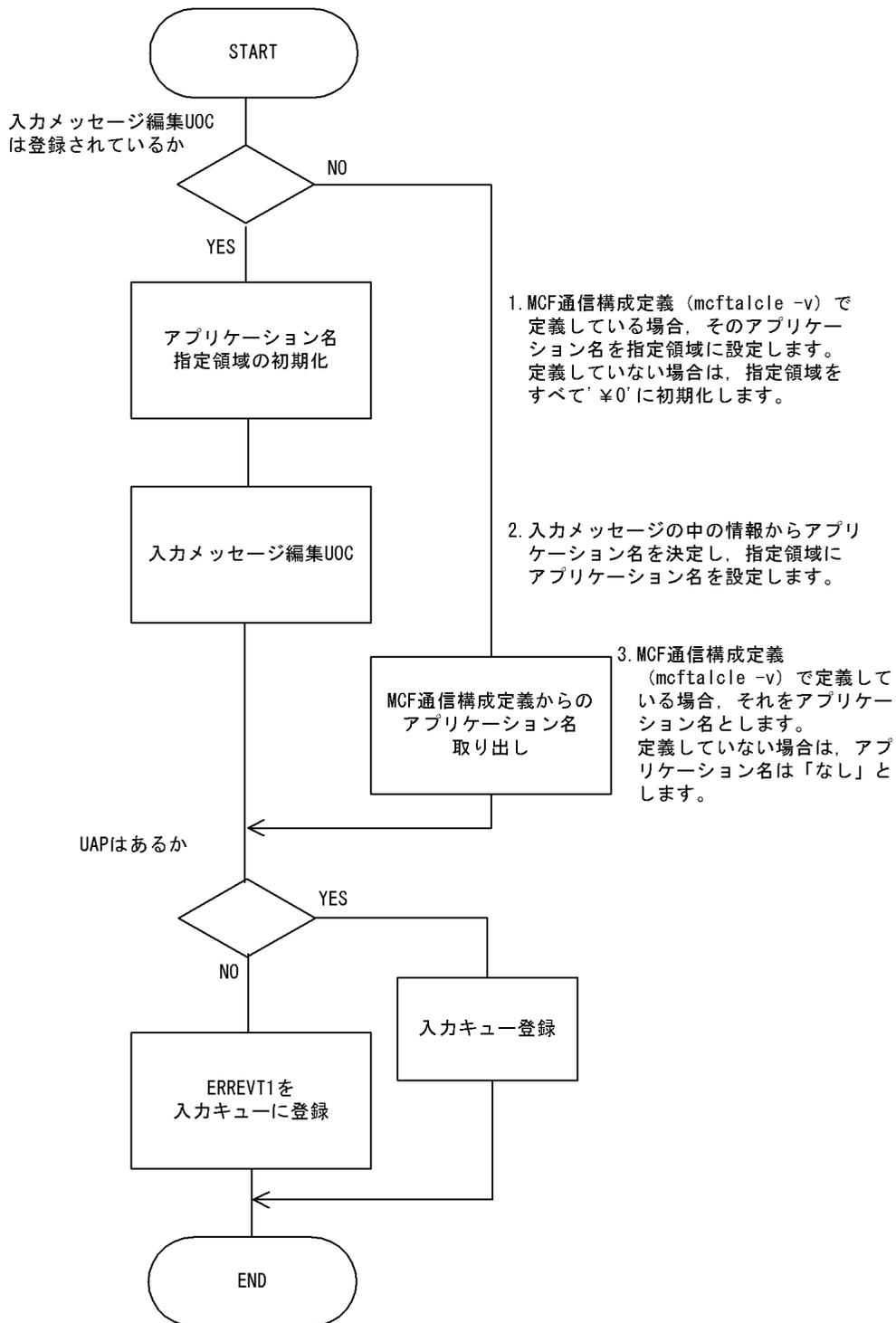
で指定したアプリケーション名を採用します。UOC か MCF 通信構成定義で、必ずアプリケーション名を決定してください。

アプリケーション名は、'¥0' で終わる 1 ~ 8 バイトの英数字です。決定したアプリケーション名は、アプリケーション名格納領域に設定してください。先頭から 9 バイト目までに '¥0' がないときは、アプリケーション名を不正とし、不正アプリケーション名検出通知イベント (ERREVT1) を UAP に通知します。

アプリケーション名の決定の処理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 4-1 アプリケーション名の決定の処理



### (3) UOC エラーリターン処理

UOC から DCMCF\_UOC\_MSG\_NG でリターンした場合, MCF はメッセージログを出力し, 障害通知イベント (CERREVT) を通知します。

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

### (4) UOC パラメタ不正の場合の処理

UOC で設定した値に不正があった場合, MCF はメッセージログを出力し, 障害通知イベント (CERREVT) を通知します。

### (5) OpenTP1 への組み込み方法

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

## 4.1.2 入力メッセージ編集 UOC インタフェース

入力メッセージ編集 UOC は, 次に示す形式で呼び出します。

### (1) 形式

ANSI C, C++ の形式

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

DCLONG uoc_func(dcmcf_uoc_min_n *parm)
```

K&R 版 C の形式

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

dcmcf_uoc_min_n *parm ;
```

#### 4. ユーザOWNコーディング, MCF イベントインタフェース

### (2) 説明

uoc\_func (入力メッセージ編集 UOC) を呼び出すとき, MCF は次に示す所定のパラメータを parm に設定します。

### (3) パラメタの内容

#### (a) dcmcf\_uoc\_min\_n の内容

```
typedef struct {
    DCLONG pro_kind;           ...プロトコル種別
    char   le_name[9];        ...論理端末名称
    char   reserve1[7];       ...予備
    DCLONG rcv_prim;          ...受信サービスプリミティブ
    dcmcf_uocbuff_list_n *buflist_adr; ...受信バッファリストアドレス
    dcmcf_uocbuff_list_n *ebuflist_adr; ...編集バッファリストアドレス
    char   aplname[9];        ...アプリケーション名
    char   reserve2[7];       ...予備
    char   *pro_indv_ifa;      ...MCFが使用
    DCLONG rtn_detail;        ...詳細リターンコード
    char   reserve3[8];       ...予備
} dcmcf_uoc_min_n;
```

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

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

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

```
typedef struct {
    char *buf_adr;           ...バッファアドレス
    DCLONG buf_size;        ...バッファ最大長
    DCLONG 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
    DCLONG buff_addr;       ...MCF内部情報2
    char   reserve2[4];     ...予備
} dcmcf_uocbufinf_n;
```

### (4) MCF が値を設定する項目

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

pro\_kind

プロトコル種別として, 次の値が設定されます。

DCMCF\_UOC\_PRO\_NCSB

## NCSB プロトコル

le\_name

メッセージを入力した論理端末の名称が設定されます。

rcv\_prim

受信サービスプリミティブとして, 次の値が設定されます。

DCMCF\_UOC\_RCV\_BRD

一方送信メッセージの受信

buflist\_adr

受信用バッファリストのアドレスが設定されます。

ebuflist\_adr

編集用バッファリストのアドレスが設定されます。

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

aplname

MCF 通信構成定義 (mcftalccle -v) で指定したアプリケーション名が設定されます。このアプリケーション名を定義していない場合,ヌルが設定されます。

pro\_indv\_ifa

MCF で使用するパラメタです。

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

buf\_num

バッファ情報の数が設定されます。

buf\_array

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

### (c) dcmcf\_uocbufinf\_n (バッファ情報)

buf\_adr

バッファのアドレスが設定されます。

buf\_size

バッファの最大長が設定されます。

seg\_size

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

buff\_id, buff\_addr

MCF で使用するパラメタです。

#### 4. ユーザOWNコーディング, MCF イベントインタフェース

##### (5) ユーザが値を設定する項目

###### (a) dcmcf\_uoc\_min\_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 に渡されません。MCF は, 詳細リターンコードをメッセージログファイルに出力します。

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

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

used\_buf\_num

使用したバッファ情報の数を設定します。

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

seg\_size

バッファの使用長を設定します。

##### (6) リターン値

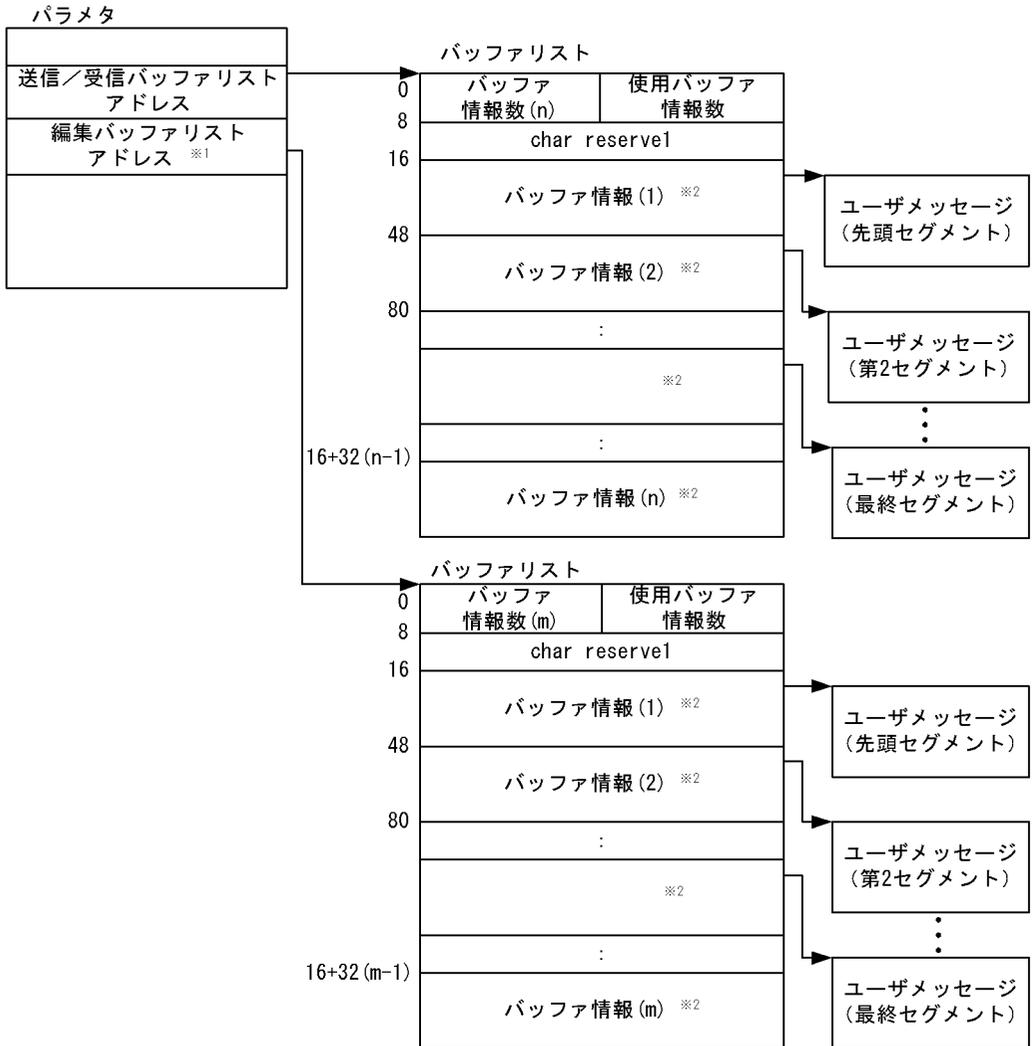
uoc\_func() は次のコードでリターンしてください。

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

##### (7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係を次の図に示します。

図 4-2 UOC インタフェース用のパラメタとバッファの関係



注 1  
送信メッセージの迂回判定 UOC の場合, または MCF 通信構成定義 (mcfaltcn -e) を指定しない場合はヌル文字となり, バッファリストとバッファは確保されません。

注 2  
バッファ情報は 32 バイトで次の形式をしています。

#### 4. ユーザOWNコーディング，MCF イベントインタフェース



### 4.1.3 出力メッセージの編集

出力メッセージ編集 UOC は，送信する論理メッセージを編集する UOC です。出力メッセージ編集 UOC は，UAP が発行した送信メッセージを相手システムに実際に送信する前に処理するように位置させます。出力キューからセグメントを読み出すと起動します。

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

#### (1) 出力メッセージの編集

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

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

#### (2) UOC エラーリターン処理

UOC から DCMCF\_UOC\_MSG\_NG でリターンした場合，MCF はメッセージログを出力し，障害通知イベント (CERREVT) を通知します。該当するメッセージは破棄されます。

#### (3) UOC パラメタ不正の場合の処理

UOC で設定した値に不正があった場合，MCF はメッセージログを出力し，障害通知イベント (CERREVT) を通知します。該当するメッセージは破棄されます。

## (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 <dcmcf.h>
#include <dcmcsb.h>
#include <dcmcfuoc.h>

DCLONG uoc_func(dcmcf_uoc_mout_n *parm)
```

K&R 版 C の形式

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

dcmcf_uoc_mout_n *parm ;
```

#### (2) 説明

uoc\_func (出力メッセージ編集 UOC) を呼び出すとき, MCF は次に示す所定のパラメータを parm に設定します。

#### (3) パラメータの内容

(a) dcmcf\_uoc\_mout\_n の内容

typedef struct {	
DCLONG pro_kind;	...プロトコル種別
char le_name[9];	...論理端末名称
char reserve1[7];	...予備
dcmcf_uocbuff_list_n *buflist_adr;	...送信バッファリストアドレス
dcmcf_uocbuff_list_n *ebuflist_adr;	...編集バッファリストアドレス
DCLONG output_no;	...メッセージ出力通番
char msg_type;	...メッセージ種別
char outputno_flag;	...メッセージ出力通番有効フラグ
char resend_flag;	...再送フラグ
char reserve2[1];	...予備
char *pro_indv_ifa;	...プロトコル個別インタフェース領域アドレス
DCLONG rtn_detail;	...詳細リターンコード

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

```
char    reserve3[20];          ... 予備
} dcmcf_uoc_mout_n;
```

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

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

#### (4) MCF が値を設定する項目

(a) dcmcf\_uoc\_mout\_n

pro\_kind

プロトコル種別として，次の値が設定されます。

DCMCF\_UOC\_PRO\_NCSB

NCSB プロトコル

le\_name

メッセージを入力した論理端末の名称が設定されます。

buflist\_adr

送信用バッファリストのアドレスが設定されます。

ebuflist\_adr

編集用バッファリストのアドレスが設定されます。

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

output\_no

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

msg\_type

メッセージ種別として，次の値が設定されます。ただし，outputno\_flag が DCMCF\_UOC\_OUTPUTNO\_OK のときだけ有効です。

'n'

一般の一方送信メッセージ

'p'

優先の一方送信メッセージ

outputno\_flag

メッセージ出力通番の有効フラグとして，次のどちらかの値が設定されます。

DCMCF\_UOC\_OUTPUTNO\_OK

メッセージ出力通番を有効にします。

DCMCF\_UOC\_OUTPUTNO\_NG

メッセージ出力通番を無効にします。

resend\_flag

再送フラグとして，次のどちらかの値が設定されます。

'r'

再送メッセージです。

'n'

再送メッセージではありません。

pro\_indv\_ifa

MCF で使用するパラメタです。

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

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

## (5) ユーザが値を設定する項目

(a) dcmcf\_uoc\_mout\_n

rtn\_detail

詳細リターンコードを設定します。

このコードは，UOC が DCMCF\_UOC\_MSG\_NG をリターンしたときに，MCF に渡されます。MCF は，詳細リターンコードをメッセージログファイルに出力します。

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

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

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

## (6) リターン値

uoc\_func() は次のコードでリターンしてください。

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

### (7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係は，入力メッセージ編集 UOC 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 送信メッセージの迂回判定

コネクション障害によって使用中のコネクションが使用できなくなった場合，送信メッセージの迂回判定 UOC で，同じコネクショングループ内のほかのコネクションを使用（迂回）するかどうかを判断できます。メッセージを迂回することを決定すると，メッセージの再送を試みます。

送信メッセージの迂回判定 UOC は，次に示す二つの条件を満たす場合に起動します。

- メッセージ送信時に通信管理から障害が通知された場合
- コネクショングループ内のコネクション数が二つ以上あり，またコネクショングループ内の正常な状態のコネクション数が一つ以上ある場合

送信メッセージの迂回判定 UOC が起動されない場合は，該当する送信メッセージを送信処理以前の状態に戻します。このとき，論理端末が閉塞されていなければ，送信メッセージの送信を続行します。また，論理端末が閉塞されていれば，送信メッセージは出力キュー上に残されます。

ユーザは，MCF メイン関数で UOC 関数のアドレスを設定します。

### (1) 送信メッセージの迂回判定

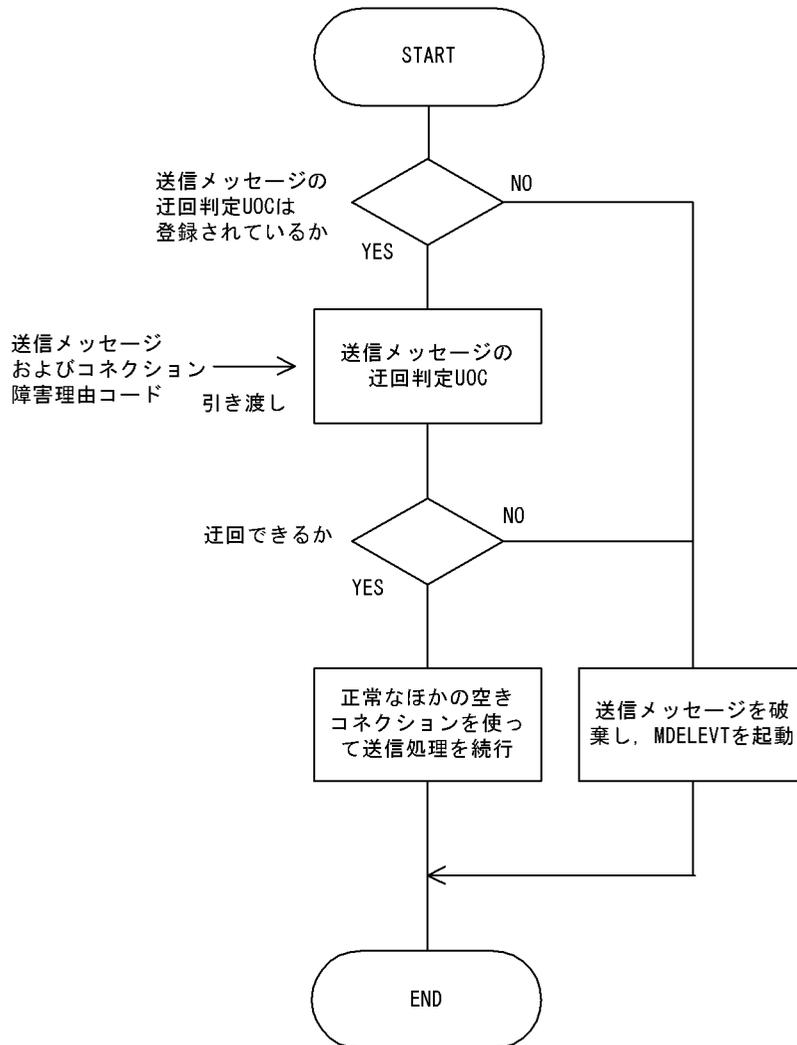
送信するメッセージが格納されている送信バッファが送信メッセージの迂回判定 UOC に渡されます。UOC では，送信バッファの情報，および通信管理から通知されるリターンコードを使用して，送信メッセージの迂回判定をします。

判定後のメッセージを迂回するかどうかは，リターン値で示します。

送信メッセージを迂回しない場合は，迂回不可メッセージ廃棄通知イベント（MDELEVT）を通知して，送信メッセージを廃棄します。

送信メッセージの迂回判定の処理を次の図に示します。

図 4-3 送信メッセージの迂回判定の処理



## (2) UOC エラーリターン処理

UOC から DCMCF\_UOC\_MSG\_NG でリターンした場合，MCF はメッセージログを出力し，迂回不可メッセージ廃棄通知イベント（MDELEVT）を通知します。該当するメッセージは廃棄されます。

## (3) UOC パラメタ不正の場合の処理

UOC で設定した値に不正があった場合，MCF はメッセージログを出力し，迂回不可メッセージ廃棄通知イベント（MDELEVT）を通知します。該当するメッセージは廃棄されます。

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

##### (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。

### 4.1.6 送信メッセージの迂回判定 UOC インタフェース

送信メッセージの迂回判定 UOC の呼び出しの形式を次に示します。

#### (1) 形式

ANSI C, C++ の場合

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

DCLONG uoc_func(dcmcf_uoc_mout_n *parm)
```

K&R 版 C の場合

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

dcmcf_uoc_mout_n *parm ;
```

#### (2) 説明

uoc\_func (送信メッセージの迂回判定 UOC) が呼び出されるとき，MCF によって次の所定のパラメタが parm に設定されます。

#### (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 reserve2[2];	...予備
char *pro_indv_ifa;	...プロトコル個別インタフェース 領域アドレス
DCLONG rtn_detail;	...詳細リターンコード
char reserve3[20];	...予備

```
} dcmcf_uoc_mout_n;
```

## (b) dcmcsb\_uoc\_prot2 の内容

```
typedef struct {
    DCLONG xnf_errcode;           ...コネクション障害の理由コード
    DCLONG reroute_code;        ...迂回するかどうかの判定
    DCLONG reserve2[14];        ...予備
} dcmcsb_uoc_prot2;
```

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

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

## (4) MCF が値を設定する項目

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

pro\_kind

プロトコル種別として、次の値が設定されます。

DCMCF\_UOC\_PRO\_NCSB

NCSB プロトコル

le\_name

メッセージを送信した論理端末の名称が設定されます。

buflist\_adr

送信用バッファリストのアドレスが設定されます。

ebuflist\_adr

編集用バッファリストのアドレスが設定されます。

送信メッセージの迂回判定 UOC を呼び出した場合、ebuflist\_adr にはヌルが設定されます。

output\_no

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

msg\_type

メッセージ種別として、次の値が設定されます。ただし、outputno\_flag が DCMCF\_UOC\_OUTPUTNO\_OK のときだけ有効です。

'n'

一般の一方送信メッセージ

#### 4. ユーザOWNコーディング, MCF イベントインタフェース

'p'

優先の一方送信メッセージ

outputno\_flag

メッセージ出力通番の有効フラグとして, 次のどちらかの値が設定されます。

DCMCF\_UOC\_OUTPUTNO\_OK

メッセージ出力通番を有効にします。

DCMCF\_UOC\_OUTPUTNO\_NG

メッセージ出力通番を無効にします。

pro\_indv\_ifa

プロトコル個別インタフェース領域のアドレスが設定されます。この場合, dcmcsb\_uoc\_prot2 を示すポインタ変数です。

##### (b) dcmcsb\_uoc\_prot2

xfn\_errcode

コネクション障害の理由コードが設定されます。理由コードの内容については, マニュアル「通信管理 XNF/AS 解説・運用編」を参照してください。

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

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

#### (5) ユーザが値を設定する項目

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

rtn\_detail

詳細リターンコードを設定します。

このコードは, UOC が DCMCF\_UOC\_MSG\_NG をリターンしたときに, MCF に渡されます。MCF は, 詳細リターンコードをメッセージログファイルに出力します。

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

##### (b) dcmcsb\_uoc\_prot2

reroute\_code

迂回するかどうかを次のどちらかの値で設定します。ただし, UOC が DCMCF\_UOC\_MSG\_OK\_SND をリターンしたときだけ有効です。

DCMCSB\_REROUTE

迂回します。

DCMCSB\_NOREROUTE

迂回しません。

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

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

## (6) リターン値

`uoc_func()` は次のコードでリターンしてください。

リターン値	意味
<code>DCMCF_UOC_MSG_OK_SND</code>	正常終了
<code>DCMCF_UOC_MSG_NG</code>	UOC エラー

## (7) パラメタとバッファの関係

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

## 4.1.7 送信メッセージの通番編集

### (1) 送信メッセージの通番編集

送信メッセージの通番編集 UOC は, 受け取った通番を基に, ユーザ独自の処理をするための UOC です。例えば, 通番を使用して送信メッセージを編集できます。

送信メッセージの通番編集を起動する場合, メッセージ送信の関数で, 出力通番が必要であることを設定してください。UOC は, UAP が先頭セグメントを送信する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, 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, 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++ の場合

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

#### 4. ユーザOWNコーディング, MCF イベントインタフェース

```
DCLONG dataleng,  
char *senddata))
```

##### K&R 版 C の場合

```
#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

DCMCF\_SEND\_UOC を指定します。

uoc\_addr

flags に対応する UOC のアドレスを指定します。

##### (c) リターン値

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

##### (d) メイン関数への登録例

###### • MHP の場合

```
main()  
{  
extern DCLONG send_uoc();  
  
dc_rpc_open();  
dc_mcf_open();  
dc_mcf_register(DCMCF_SEND_UOC, send_uoc);  
dc_mcf_mainloop();  
dc_mcf_close();  
dc_rpc_close();  
}
```

###### • SPP の場合

```
main()  
{  
extern DCLONG send_uoc();  
  
dc_rpc_open();  
dc_mcf_open();  
dc_mcf_register(DCMCF_SEND_UOC, send_uoc);  
dc_rpc_mainloop();  
}
```

```
dc_mcf_close();
dc_rpc_close();
}
```

## 4.1.8 送信メッセージの通番編集 UOC インタフェース

送信メッセージの通番編集 UOC は、次に示す形式で、send\_uoc 関数として作成します。

### (1) 形式

ANSI C, C++ の場合

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

K&R 版 C の場合

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

### (2) MCF から値が渡される引数

flags

送信メッセージの通番編集 UOC がいつ呼ばれたかが設定されます。

DCMCF\_SEND\_DML

メッセージを送信する関数、または命令文が呼び出されたときを意味します。

DCMCF\_RESEND\_DML

メッセージを再送する関数または命令文が呼び出されたときを意味します。

termname

送信先の論理端末名称が渡されます。

sendno

送信メッセージの通番が渡されます。

sendid

送信するメッセージ種別が渡されます。次のどちらかが渡されます。

DCMCF\_SEND\_PRIO

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

優先の一方送信メッセージ

DCMCF\_SEND\_NORM

一般の一方送信メッセージ

dataleng

送信メッセージ長が渡されます。

senddata

先頭セグメントの内容が渡されます。

### (3) リターン値

リターン値	意味
DC_OK	正常に終了しました。

## 4.1.9 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に設計してください。また、UOC の中で関数の再帰呼び出しはしないでください。

### (2) UOC で使用できる関数

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

メモリを操作する関数

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

時間取得関数

### (3) UOC の異常処理

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

### (4) UOC の実行タイミング

MCF が起動する UOC の実行タイミングは、OpenTP1 システム、および UAP の開始、終了シーケンスと同期しない場合があります。UAP より先に UOC が実行されたり、

UAP がすべて終了してから UOC が実行されたりしてもよいように作成してください。

#### (5) ユーザ情報の操作方法

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

## 4.2 MCF イベントインタフェース

OpenTP1 でメッセージ送受信すると, 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, 入力メッセージの編集 UOC は呼び出しません。

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

### 4.2.1 MCF イベントの種類

TP1/NET/NCSB が通知する MCF イベントを次の表に示します。

表 4-1 MCF イベント一覧

MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP の処理の例
不正アプリケーション名検出通知イベント	ERREVT1	メッセージのアプリケーション名が MCF アプリケーション定義にありません。	該当するアプリケーション名がなかったことを通知します。
メッセージ廃棄通知イベント	ERREVT2	次の理由で受信メッセージを廃棄しました。 <ul style="list-style-type: none"> <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, 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>アプリケーション起動時(即時起動の場合)に障害が発生しました。</li> </ul>	メッセージを廃棄したことを報告します。
UAP 異常終了通知イベント	ERREVT3	MHP のセグメント受信関数に, セグメントを渡したあとに MHP の異常終了が発生しました。	UAP 異常終了時の対処障害メッセージを送信します。
タイマ起動メッセージ廃棄通知イベント	ERREVT4	アプリケーション起動時(タイマ起動の場合)に障害が発生しました。	メッセージを廃棄したことを報告します。

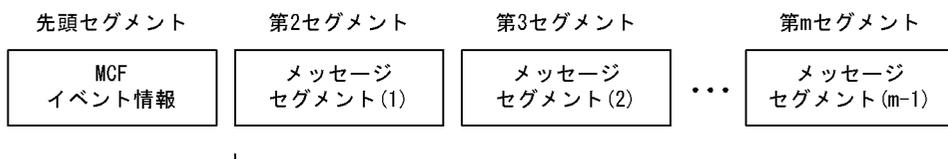
MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP の処理の例
未処理送信メッセージ廃棄通知イベント	ERREVTn	次の理由で未処理送信メッセージを破棄しました。 <ul style="list-style-type: none"> <li>• MCF の正常終了処理時に，未処理送信メッセージの滞留時間監視の時間切れ（タイムアウト）が発生しました。</li> <li>• 運用コマンド（mcftdlqle）の入力によって，出力キューが削除されました。</li> </ul>	未処理送信メッセージを廃棄したことを報告します。
障害通知イベント	CERREVT	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。	メッセージの送受信ができることを報告します。
	CCLSEVT	コネクションが正常に解放されました。	メッセージの送受信ができないことを報告します。
迂回不可メッセージ廃棄通知イベント	MDELEVT	コネクション障害発生によってメッセージの送信，または迂回ができなくなりました。	送信メッセージを廃棄したことを報告します。

## 4.2.2 MCF イベント通知時のセグメント構成

MCF イベントを MHP に通知する場合，先頭セグメントに MCF イベント情報を設定します。エラーイベント（ERREVTn），および迂回不可メッセージ廃棄通知イベント（MDELEVT）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 4-4 MCF イベント通知時のセグメント構成



エラーイベント（ERREVTn）  
 および迂回不可メッセージ廃棄通知イベント（MDELEVT）  
 通知時に渡されます。

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

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

COBOL 言語を使用したエラーイベント (ERREVTn) および迂回不可メッセージ破棄通知イベント (MDELEVT) の場合は，バッファ形式 1 とバッファ形式 2 とで先頭の内容が異なります。このため，それ以降の項目の位置にずれがあります。「4.2.4 MCF イベント情報の形式 (COBOL 言語)」のエラーイベント (ERREVTn) および迂回不可メッセージ破棄通知イベント (MDELEVT) の表では，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。

MCF イベントの共通ヘッダ (dc\_mcf\_evtheadr)，およびエラーイベント (ERREVTn) で使用する構造体は，<demcf.h> で定義してあります。また，CERREVT 通知インタフェース (dcmcsb\_carrevt)，COPNEVT，CCLSEVT 通知インタフェース (dcmcsb\_statevt)，および MDELEVT 通知インタフェース (dcmcsb\_mdelevt) は，<dcmcsb.h> で定義されています。<dcmcsb.h> の前に <demcf.h> を取り込んでおいてください。

MCF イベント情報の先頭部分の形式を次に示します。

#### (1) MCF イベントの共通ヘッダ

##### (a) 形式

```
struct dc_mcf_evtheadr {
    char    mcfevt_name[9] ;           ...MCFイベント情報コード
    char    le_name[16] ;             ...論理端末名称
    char    cn_name[9] ;              ...コネクション名またはコネクション
                                     グループ名
    unsigned char    format_kind;     ...MCFが使用
    char    reserve01;                ...予備
    DCLONG time ;                     ...メッセージ入力時刻
};
```

##### (b) MCF イベントとして設定される項目

le\_name

メッセージを入力した論理端末名称が設定されます。

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

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

ERREVT1 の場合は, メッセージを出力する論理端末名称が設定されます。  
CERREVT, COPNEVT, および CCLSEVT の場合は無効です。

cn\_name

コネクション名またはコネクショングループ名が設定されます。  
ERREVT2 または ERREVT3 で, 次に示す場合には, '\*' が設定されます。

- SPP がアプリケーション起動をした MHP で, 障害が発生した場合
- 上記の障害発生時に MCF イベントとして起動された MHP 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  evthead; ...MCFイベント情報共通ヘッダ
    char reserve01[12];          ...予備
    char reserve02[10];          ...予備
    char reserve03[2];           ...予備
    char ap_name[10];            ...アプリケーション名
    char reserve04[2];           ...予備
};
```

### (b) MCF イベントとして設定される項目

ap\_name

次に示すどちらかが設定されます。

- 形式不正のアプリケーション名
- 定義されていないアプリケーション名

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

MHP 以外から送信された場合は, ヌル文字が設定されます。

## (3) ERREVT2

### (a) 形式

```
struct dc_mcf_evt2_type {
    struct dc_mcf_evtheader  evthead; ...MCFイベント情報共通ヘッダ
    char reserve01[12];          ...予備
    char reserve02[10];          ...予備
    char reserve03[2];           ...予備
    char ap_name[10];            ...アプリケーション名
    short reason_code;           ...理由コード
};
```

#### 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

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

#### (4) ERREVT3

##### (a) 形式

```
struct dc_mcf_evt3_type {
    struct dc_mcf_evtheader evtheader ; ...MCFイベント情報共通ヘッダ
    char reserve01[12] ; ...予備
    char map_name[10] ; ...MCFが使用
    char reserve03[2] ; ...予備
    char ap_name[10] ; ...アプリケーション名
    char reserve04[2] ; ...予備
    char service_name[32] ; ...サービス名
    char serv_grp_name[32] ; ...サービスグループ名
    char bid[36] ; ...トランザクションブランチID領域
};
```

##### (b) MCF イベントとして設定される項目

ap\_name

異常が発生した MHP のアプリケーション名が設定されます。

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

MHP 以外から送信された場合は，ヌル文字が設定されます。

service\_name

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

serv\_grp\_name

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

bid

トランザクションのブランチ ID が設定されます。

#### (5) ERREVT4

##### (a) 形式

```
struct dc_mcf_evta_type {
    struct dc_mcf_evtheader evtheader ; ...MCFイベント情報共通ヘッダ
    char reserve01[12] ; ...予備
};
```

```

char    map_name[10] ;           ...MCFが使用
char    reserve03[2] ;         ...予備
char    ap_name[10] ;          ...アプリケーション名
char    reserve04[2] ;         ...予備
char    reserve05[32] ;        ...予備
char    reserve06[32] ;        ...予備
DCLONG  user_leng;             ...他プロトコルの場合に有効
char    user_data[16] ;        ...他プロトコルの場合に有効
char    reserve07[16] ;        ...予備
};

```

## (b) MCF イベントとして設定される項目

ap\_name

正常終了したメッセージのアプリケーション名が設定されます。

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

MHP 以外から送信された場合は，ヌル文字が設定されます。

## (6) CERREVT

## (a) 形式

```

typedef struct {
    struct dc_mcf_evtheader  header ;           ...MCFイベント情報共通ヘッダ
    DCLONG  err_fact ;                         ...障害要因コード
    DCLONG  err_reason1 ;                      ...理由コード1
    DCLONG  err_reason2 ;                      ...理由コード2
    char    reserve[4] ;                       ...予備
    char    group_name[16] ;                   ...コネクショングループ名
    char    reserve1[26] ;                     ...予備
} dcmcsb_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

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。

group\_name

コネクショングループ名が設定されます。

## (7) COPNEVT, CCLSEVT

## (a) 形式

```

typedef struct {
    struct dc_mcf_evtheader  header ;           ...MCFイベント情報共通ヘッダ
    DCLONG  noti_fact ;                         ...通知要因コード

```

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

```

char  reserve[12] ;           ... 予備
char  group_name[16] ;       ... コネクショングループ名
char  reserve1[26] ;         ... 予備
} dcmcsb_statevt ;

```

(b) MCF イベントとして設定される項目

group\_name  
コネクショングループ名が設定されます。

### (8) MDELEVT

(a) 形式

```

typedef struct {
  struct dc_mcf_evtheader header ;   ...MCFイベント情報共通ヘッダ
  DCLONG del_reason ;                ...メッセージ廃棄理由コード
  DCLONG xnf_errcode ;               ...コネクション障害理由コード
  char  reserve[8] ;                 ... 予備
  char  group_name[16] ;              ...コネクショングループ名
  char  reserve1[26] ;                ... 予備
} dcmcsb_mdelevt ;

```

(b) MCF イベントとして設定される項目

group\_name  
コネクショングループ名が設定されます。

del\_reason  
メッセージ廃棄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。

xnf\_errcode  
コネクション障害理由コードが設定されます。マニュアル「通信管理 XNF/AS 解説・運用編」を参照してください。

## 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 の場合の MCF イベントの内容を表 4-2 ~ 表 4-8 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	-	-

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			
エラーイベント コード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT1 を示す '1' が設定されま す。
入力元論理端末 名称	12	8	8	英数字	メッセージを入力した論理端末名称で す。
予備	20	16	20	-	-
アプリケーション 名	40	36	8	英数字	次に示すどれかが設定されます。 • 形式不正となったアプリケーション 名 • 定義されていないアプリケーション 名
予備	48	44	8	-	-
予備	56	52	8	-	-
予備	64	60	8	-	-
コネクショング ループ名	72	68	8	英数字	コネクショングループ名です。
予備	80	76	16	-	-
メッセージが入 力された日付	96	92	8	外部 10 進数字	端末入力メッセージを入力した日付で す。 YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入 力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻で す。 HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
予備	112	108	16	-	-

(凡例)

- : 該当しません。または，使用されません。

4. ユーザOWNコーディング，MCF イベントインタフェース

表 4-3 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。 1. 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 2. 上記の障害発生時に MCF イベント として起動された MHP によって、 さらにアプリケーション起動され た MHP で、障害が発生した場合
予備	20	16	20	-	-
アプリケーション名	40	36	8	英数字	エラーになった UAP のアプリケー ション名が設定されます。
予備	48	44	8	-	-
予備	56	52	8	-	-
予備	64	60	8	-	-
コネクショングループ名	72	68	8	英数字	コネクショングループ名です。 次に示す場合には、'*' が設定され ます。 1. 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 2. 上記の障害発生時に MCF イベント として起動された MHP によって、 さらにアプリケーション起動をさ れた MHP で、障害が発生した場 合
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

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

(凡例)

- : 該当しません。または、使用されません。

注

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

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベントコード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT3 を示す '3' が設定されます。
入力元論理端末名称	12	8	8	英数字	メッセージを入力した論理端末名称です。 次に示す場合は、'*' が設定されます。 1. 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 2. 上記の障害発生時に MCF イベントとして起動された MHP が、さらにアプリケーション起動をした MHP で、障害が発生した場合
予備	20	16	20	-	-
予備	40	36	8	-	-
予備	48	44	8	-	MCF が使用します。

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
アプリケーション名	56	52	8	英数字	異常が発生したメッセージのアプリケーション名です。
予備	64	60	8	-	-
コネクショングループ名	72	68	8	英数字	コネクショングループ名です。 次に示す場合は，'*' が設定されます。 1. 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 2. 上記の障害発生時に MCF イベントとして起動された MHP が，さらにアプリケーション起動をした MHP で，障害が発生した場合
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進数字	端末入力メッセージを入力した日付です。 YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻です。 HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
予備	112	108	16	-	-
サービス名	128	124	31	英数字	異常が発生した MHP のアプリケーション名に対応するサービス名です。
予備	159	155	1	-	-
サービスグループ名	160	156	31	英数字	異常が発生した MHP 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 (ERREVT A)

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベントコード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT A を示す 'A' が設定されます。
出力先論理端末名称	12	8	8	英数字	メッセージを出力する論理端末名称です。
予備	20	16	20	-	-
予備	40	36	8	-	-
マップ名	48	44	8	-	MCF が使用します。
アプリケーション名	56	52	8	英数字	正常終了したメッセージのアプリケーション名 (MHP から送信されたメッセージの場合に設定されます。MHP 以外から送信された場合は空白が設定されます)
予備	64	60	8	-	-
コネクショングループ名	72	68	8	英数字	コネクショングループ名です。
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進数字	端末入力メッセージを入力した日付です。 YYYYMMDD の形式です。 YYYY : 西暦の年 MM : 月 DD : 日
メッセージが入力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻です。 HHMMSS00 の形式です。 HH : 時 MM : 分 SS : 秒 00 は固定です。
予備	112	108	16	-	-
予備	128	124	31	-	-
予備	159	155	1	-	-
予備	160	156	31	-	-

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備	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。
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，またはコネクショングループ名が設定されます。
メッセージが入力された日付	32	8	外部 10 進数字	CERREVT 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	40	8	外部 10 進数字	CERREVT 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。
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
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	4	-	-
コネクショングループ名	64	16	英数字	コネクショングループ名が設定されます。
予備	80	28	-	-

(凡例)

- : 該当しません。または，使用されません。

注

理由コード1および理由コード2については、「付録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	論理端末名称が設定されます。
予備	16	8	-	-
入力元コネクション名	24	8	英数字	コネクション名,またはコネクショングループ名が設定されます。 コネクション名,またはコネクショングループ名が決定しない場合,ヌルが設定されます。
メッセージが入力された日付	32	8	外部 10 進数字	COPNEVT, CCLSEVT 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	40	8	外部 10 進数字	COPNEVT, CCLSEVT 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。
通知要因コード	48	4	2 進数字	通知要因が設定されます。 (00000030) <sub>16</sub> : コネクション確立, または解放 (00000031) <sub>16</sub> : 論理端末の閉塞解除, または閉塞
予備	52	12	-	-
コネクショングループ名	64	16	英数字	コネクショングループ名が設定されます。
予備	80	28	-	-

(凡例)

- : 該当しません。または, 使用されません。

#### 4. ユーザOWNコーディング，MCF イベントインタフェース

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

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	-	-
イベントコード	4	0	8	英数字	「MDELEVT」が設定されます。
入力元論理端末名称	12	8	8	英数字	論理端末名称が設定されます。
予備	20	16	8	-	-
入力元コネクション名	28	24	8	英数字	コネクション名，またはコネクショングループ名が設定されます。 コネクション名，またはコネクショングループ名が決定しない場合，ヌルが設定されます。
メッセージが入力された日付	36	32	8	外部 10 進数字	MDELEVT 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	44	40	8	外部 10 進数字	MDELEVT 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。
メッセージ廃棄理由コード <sup>1</sup>	52	48	4	2 進数字	メッセージ廃棄の理由コードが設定されます。
コネクション障害理由コード <sup>2</sup>	56	52	4	2 進数字	コネクション障害の理由コードが設定されます。
予備	60	56	8	-	-
コネクショングループ名	68	64	16	英数字	コネクショングループ名が設定されます。
予備	84	80	28	-	-

(凡例)

- : 該当しません。または，使用されません。

注 1

メッセージ廃棄理由コードの内容については，「付録 C 理由コード一覧」を参照してください。

注 2

コネクション障害理由コードの内容については，マニュアル「通信管理 XNF/AS 解説・運用編」

#### 4. ユーザOWNコーディング, MCF イベントインタフェース

を参照してください。



# 5

## システム定義

この章では、NCSB プロトコルを使用するために必要な、OpenTP1 システム定義の中での TP1/NET/NCSB 固有の定義について説明します。また、自システム内にある通信管理プログラムと関連づける内容、およびシステム定義例について説明します。

---

TP1/NET/NCSB の定義の概要

---

TP1/NET/NCSB 固有のシステム定義の種類

---

mcftalccn (コネクション定義)

---

mcftalcle (論理端末定義)

---

mcftgrpcn (コネクショングループ定義の開始)

---

mcftgrped (コネクショングループ定義の終了)

---

システムサービス情報定義

---

システムサービス共通情報定義

---

MCF 定義オブジェクトの生成

---

自システムの通信管理プログラムと関連づける内容

---

定義例

---

## TP1/NET/NCSB の定義の概要

TP1/NET/NCSB のシステム定義は、OpenTP1 のネットワークコミュニケーション定義の中で定義します。

### OpenTP1 のネットワークコミュニケーション定義の中での定義

OpenTP1 のネットワークコミュニケーション定義のうち、TP1/NET/NCSB 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/NCSB 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使用して作成します。それらの中には、プロトコルで共通のものと、プロトコルに固有のものがあります。表 5-1 の定義の中で、TP1/NET/NCSB に固有の定義があるものを次に示します。

- MCF 通信構成定義

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

### TP1/NET/NCSB 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/NCSB 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/NCSB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

NCSB 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/NCSB のシステム定義内容を自システムの通信管理プログラムと関連づける必要があります。

この章では、自システムの通信管理プログラムと関連づける内容を示します。

## 5. システム定義

TP1/NET/NCSB 固有のシステム定義の種類

# TP1/NET/NCSB 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/NCSB に固有の定義の一覧を次の表に示します。

表 5-2 TP1/NET/NCSB に固有の定義の一覧

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》		
M C F 通 信 構 成 定 義	共通定義	プロトコル共通のコマンドだけで定義できます。共通のコマンドについては、マニュアル「OpenTP1 システム定義」を参照してください。					
	プロ ト コ ル 固 有 定 義	mcftgrpen (コネクショングループ定義の開始) 指定数: 1 ~ 512	-g	-	コネクショングループ名	1 ~ 8 文字の識別子	
		mcftalcle (論理端末定義) 指定数: mcftgrpen と 同数	-l	-	論理端末名称	1 ~ 8 文字の識別子	
			-t	-	論理端末タイプ	any	
			-v	-	アプリケーション名	1 ~ 8 文字の識別子	
			-m	mmsgcnt		メモリ出力メッセージ最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
				dmsgcnt		ディスク出力メッセージ最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
			-k	quekind		出力キューの媒体の種類	《memory》   disk
				quegrpid		キューグループ ID	1 ~ 8 文字の識別子
			-o	aj		送信完了時の情報を取得するかどうか	《yes》   no
mcftalccn (コネクション定義)	-c	-	コネクション ID	1 ~ 8 文字の識別子			

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 ~ 512	-p	-	プロトコルの種別	ncsb
		-g	sndbuf	送信用バッファグループ番号	符号なし整数 ((1 ~ 512))
			rcvbuf	受信用バッファグループ番号	符号なし整数 ((1 ~ 512))
		-e	msgbuf	編集用バッファグループ番号	符号なし整数 ((1 ~ 512))
			count	編集バッファ数	符号なし整数 ((1 ~ 131070))
		-i	-	コネクションの確立方法	auto   《manual》
		-k	-	NCSB 回線名称	1 ~ 8 文字の識別子
		-y	-	コンテンション発生時の優先指定	《winner》   loser
		-u	-	データリンクプロトコル種別	NCS_B   CAFIS   ARU   ACS   KOSIN   SINKIN
		-b	bretry	コネクションの確立再試行をするかどうか	《yes》   no
			bretrycnt	コネクションの確立再試行回数	符号なし整数 ((0 ~ 65535)) 《0》 (単位：回)
			bretryint	コネクションの確立再試行間隔	符号なし整数 ((0 ~ 2550)) 《60》 (単位：秒)
		mcftgrped (コネクショングループ定義の終了) 指定数： mcftgrpcn と同数	-	-	コネクション定義の終了

( 凡例 )

- : 該当しません。

### 定義の指定順序

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

## 5. システム定義

### TP1/NET/NCSB 固有のシステム定義の種類

(	mcftgrpcn	(コネクショングループ定義の開始)	)	一組として繰り返し可
	mcftalcle	(論理端末定義)		
	mcftalccn*	(コネクション定義)		
	⋮			
	mcftgrped	(コネクショングループ定義の終了)		

#### 注

コネクション定義は、一つの繰り返し単位の中に複数指定できます。

## mcftalccn (コネクション定義)

---

### 形式

```
mcftalccn -c コネクションID
          -p ncsb
          -g "sndbuf = メッセージ送信用バッファグループ番号
             rcvbuf = メッセージ受信用バッファグループ番号"
          [-e " {msgbuf = メッセージ編集用バッファグループ番号}
             {count = メッセージ編集用バッファ数} " ]
          [-i auto | manual ]
          -k 回線名称
          [-y winner | loser ]
          -u NCS_B | CAFIS | ARU | ACS | KOSIN | SINKIN
          [-b " {bretry = yes | no}
             {bretrycnt = コネクション確立障害時の確立再試行回数}
             {bretryint = コネクション確立障害時の確立再試行間隔} " ]
```

### 機能

コネクションに関する環境を定義します。

### オプション

-c コネクション ID     ~   1 ~ 8 文字の識別子

OpenTP1 システム内で、一意となるコネクション ID を指定します。

-p ncsb

プロトコルの種別を指定します。

ncsb

NCSB プロトコル

-g

(オペランド)

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

メッセージ送信用バッファのグループ番号を指定します。

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

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

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))

メッセージ受信バッファのグループ番号を指定します。

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

## 5. システム定義

### mcftalccn (コネクション定義)

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

-e

(オペランド)

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

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

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

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

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

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 オペランド、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ます。

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

msgbuf オペランドを省略した場合は、このオペランドの指定は無効です。

-i auto | manual ~ 《manual》

MCF 開始時および再開時に、自動的にコネクションを確立するかどうかを指定します。

auto

コネクションを自動的に確立します。

manual

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

-k 回線名称 ~ 1 ~ 8 文字の識別子

使用する回線の名称を指定します。

通信管理の定義の、baseline 文の name オペランドに指定した回線名称を指定してください。

-y winner | loser ~ 《winner》

コンテンツ発生時の優先 / 非優先を指定します。

winner

優先局

loser

非優先局

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

-u NCS\_B | CAFIS | ARU | ACS | KOSIN | SINKIN

使用するデータリンクプロトコルの種別 (NCS-B 手順のプロトコルクラス) を指定します。

NCS\_B

NCS-B インタフェース接続仕様書で規定されたプロトコル

CAFIS

クレジット情報データ通信システム接続仕様書で規定されたプロトコル

ARU

ARU システム接続仕様書で規定されたプロトコル

ACS

地銀 CD 全国ネットサービスデータ通信システム接続仕様書で規定されたプロトコル

KOSIN

個人信用情報センタ接続仕様書で規定されたプロトコル

SINKIN

信用金庫オンライン接続仕様書で規定されたプロトコル

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

-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、コネクションの確立を再試行するかどうかを指定します。

yes

コネクションの確立を再試行します。

no

コネクションの確立を再試行しません。

## 5. システム定義

mcftalccn (コネクション定義)

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

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

コネクション確立時に障害が発生した場合の、コネクションの確立を再試行する回数を指定します。

このオペランドを省略した場合、または 0 を指定した場合、無限に再試行されます。

bretry オペランドで no を指定した場合、このオペランドの指定は無効です。

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

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

コネクション確立時に障害が発生した場合の、コネクションの確立を再試行する時間間隔を指定します。

0 を指定した場合、障害後直ちに再試行されます。

bretry オペランドで no を指定した場合、このオペランドの指定は無効です。

同じコネクショングループに属するコネクションでは、この指定値を一致させてください。

### 注意事項

同じコネクショングループに属するコネクションでは、次のオプションおよびオペランドの指定値がすべて一致する必要があります。

- -g オプションの sndbuf オペランド, および rcvbuf オペランド
- -e オプションの msgbuf オペランド
- -i オプション
- -y オプション
- -u オプション
- -b オプションの bretry オペランド, bretrycnt オペランド, および bretryint オペランド

-g オプションおよび -e オプションで指定するバッファグループ番号は、バッファグループ定義の mcftbuf コマンドに対応しています。mcftbuf コマンドでは、1 コネクショングループ単位に次の表に示す資源が必要です。バッファグループ定義については、マニュアル「OpenTP1 システム定義」を参照してください。

バッファ種別	mcftbuf コマンド	
	length オペランドの指定値	count オペランドの指定値
送信用バッファ (sndbuf)	送信最大メッセージ長 + MCF 使用領域長 (92 バイト)	最大セグメント分割数 × コネクション数

バッファ種別	mcfbuf コマンド	
	length オペランドの指定値	count オペランドの指定値
受信用バッファ ( revbuf )	受信最大メッセージ長 + MCF 使用領域長 ( 92 バイト )	( 最大セグメント分割数 + 1 ) × 同時受信メッセージ数 × コネクション数
編集用バッファ ( msgbuf )	メッセージ編集 UOC で編集後の最大 メッセージ長 + MCF 使用領域長 ( 92 バイト )	コネクション数 × 最大セグメント分割数 × 2

注

UOC でメッセージを編集しない場合は、指定の必要はありません。

## mcftalcle (論理端末定義)

---

### 形式

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

OpenTP1 システム内で、一意となる論理端末名称を指定します。

-t any

論理端末の端末タイプを指定します。

any

任意型論理端末

-v アプリケーション名     ~   1 ~ 8 文字の識別子

該当する論理端末で受信したメッセージの起動先アプリケーション名を指定します。アプリケーション定義 (mcfalcap の -n オプションの name オペランド) で指定した名称を指定してください。アプリケーション定義については、マニュアル「OpenTP1 システム定義」を参照してください。

このオプションは、入力メッセージ編集 UOC でアプリケーション名を決定する場合は不要です。

-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) はエラーとなります。  
0 を指定した場合、メモリキューで待ち合わせをする出力メッセージの数は無制限になります。

dmsgcnt = ディスク出力メッセージ最大格納数 ~ 符号なし整数 ((0 ~ 65535))  
《0》

ディスクキューで待ち合わせをする出力メッセージの最大格納数を指定します。出力メッセージの待ち合わせ数が指定した最大数になると、それ以後 UAP からの送信要求 (dc\_mcf\_send 関数または SEND 文) はエラーとなります。

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になります。

-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) のどれかを指定します。

この quegrpid オペランドは、quekind オペランドで disk を指定した場合だけ指定してください。

-o

(オペランド)

aj = yes | no ~ 《yes》

送信完了時の情報を取得するかどうかを指定します。

yes

取得します。

no

取得しません。

## mcftgrpcn (コネクショングループ定義の開始)

---

### 形式

mcftgrpcn -g コネクショングループ名

### 機能

コネクショングループ定義の開始を示します。

### オプション

-g コネクショングループ名    ~    1 ~ 8 文字の識別子

コネクショングループ名を指定します。

このコネクショングループ名は、ほかの mcftgrpcn コマンドの -g オプションで指定するコネクショングループ名と重複して指定できません。また、mcftalcle コマンドで指定する論理端末名称、mcftalccn コマンドで指定するコネクション ID とも重複して指定できません。

## mcftgrped (コネクショングループ定義の終了)

---

形式

mcftgrped

機能

コネクショングループの定義の終了を示します。

オプション

ありません。

## システムサービス情報定義

---

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

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

システムサービス情報定義は、テキストエディタを使用して作成します。

システムサービス情報定義の完全パス名を次に示します。

```
$DCDIR/lib/sysconf/定義ファイル名
```

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

### 形式

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

MCF 通信サービスを起動するための実行形式プログラム名を指定します。

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

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

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

システムサービス共通情報定義の完全パス名を次に示します。

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

### 形式

#### set 形式

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

### 機能

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

### 説明

#### set 形式のオペランド

max\_socket\_descriptors= ソケット用ファイル記述子の最大数    ~    符号なし整数  
(64 ~ 2048)

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

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

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

注 1

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

注 2

## 5. システム定義

### システムサービス共通情報定義

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

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

max\_open\_fds=MCF 通信プロセスでアクセスするファイルの最大数 ~ 符号なし整数 ((100 ~ 2048))

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

ファイル記述子の最大数を求める計算式を次に示します。

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

注 1

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

注 2

MCF 通信プロセスが扱う定義ファイルなどの数の最大値です。

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

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。

$$(\text{「このオペランドの指定値」} + \text{同定義内の「max_socket_descriptorsオペランドの指定値」}) \leq 2048$$

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。

$$2048 - (\text{同定義内の「max_socket_descriptorsオペランドの指定値」})$$

### 注意事項

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

## 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

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

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

TP1/NET/NCSB 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。

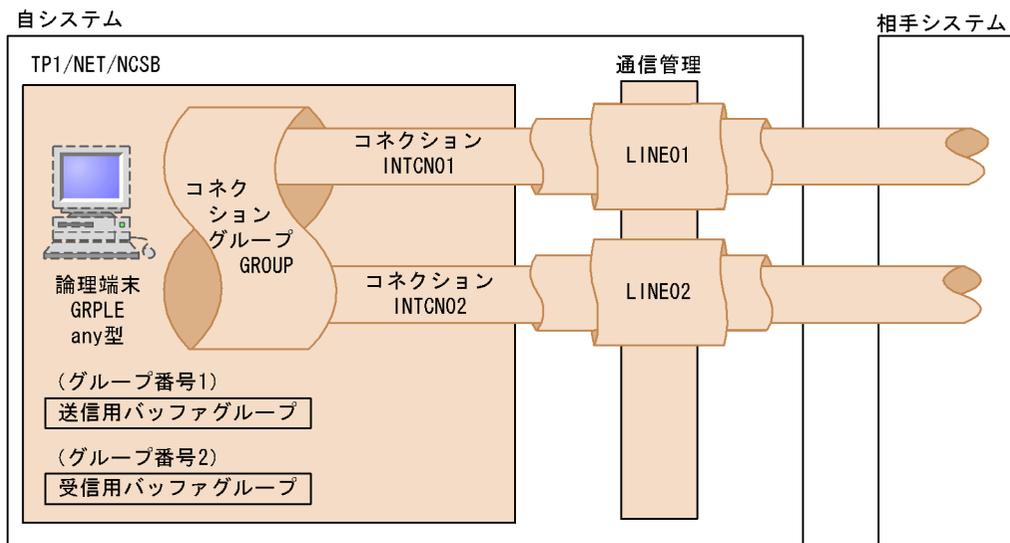
## 自システムの通信管理プログラムと関連づける内容

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

TP1/NET/NCSB で使用する通信管理は XNF/AS です。TP1/NET/NCSB の MCF 通信構成定義 (mcftalcn -k) で指定する回線名称を XNF/AS の構成定義のベーシック回線定義文 (baseline 文の name オペランド) の指定値と合わせます。

通信管理に XNF/AS を使用した TP1/NET/NCSB のシステム構成例を次の図に示します。

図 5-1 TP1/NET/NCSB のシステム構成例



## 定義例

---

ここでは、図 5-1 の「TP1/NET/NCSB のシステム構成例」で示したシステム構成のコーディング例を示します。

XNF/AS の構成定義のコーディング例、ならびに TP1/NET/NCSB の定義のコーディング例を示します。

また、TP1/NET/NCSB 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。

- 共通定義の例  
/BeTRAN/examples/mcf/NCSB/conf/com\_c
- プロトコル固有定義の例  
/BeTRAN/examples/mcf/NCSB/conf/com\_d

### XNF/AS 構成定義のコーディング例

```
/**** NCSB モデル定義 NCSB *****/
/**** 回線2本 *****/

configuration
    version 01
    max_Line_adapter 1
    max_NCSBline 2
    ;

Line_adapter
    name LADP01
    location_code 27-08
    adapter_type BASIC
    initial_status active
    auto_start yes
    ;

group
    name NCSB1
    type NCSB
    ;

    basicline
        name LINE01
        number 02
        speed 2.4K
        modem_clock synch_ST2
        ;

    basicline
        name LINE02
        number 03
        speed 2.4K
        modem_clock synch_ST2
        ;
```

5. システム定義  
定義例

TP1/NET/NCSB の定義のコーディング例 (共通定義)

```
#
#   MCF通信構成定義 (共通定義)
#
#
#--- MCF環境定義
#
#       mcftenv      -s  01                ¥
#                   -a  NCapli
#
#
#--- MCF共通定義
#
#       mcftcomn
#
#
#--- 最大処理多重度定義
#
#       mcfttred
#
#
#--- トレース環境定義
#
#       mcfttrc     -t  "size   =204800  ¥
#                       disk    =yes     ¥
#                       bufcnt  =100     ¥
#                       trrcnt  =6"
#
#
#--- 送受信バッファ定義
#
#--- メッセージ送受信バッファ定義 (送信用)
#
#       mcftbuf     -g  "groupno=1      ¥
#                       length =1024    ¥
#                       count   =10"
#
#
#--- メッセージ送受信バッファ定義 (受信用)
#
#       mcftbuf     -g  "groupno=2      ¥
#                       length =1024    ¥
#                       count   =10"
#
#
#--- 編集バッファ定義
#
#       mcftbuf     -g  "groupno=4      ¥
#                       length =1024    ¥
#                       count   =10"
#
#
#----- 終わり
```

TP1/NET/NCSB の定義のコーディング例 (プロトコル固有定義)

```
#   MCF通信構成定義 (プロトコル固有定義)
```

```

#
#
#
#--- コネクショングループ定義の開始
#
    mcftgrpcn    -g    GROUP
#
#--- 論理端末定義
#
    mcftalcle    -l    GRPLE                                ¥
                  -t    any                                ¥
                  -m    "mmsgcnt=0                        ¥
                        dmsgcnt=0"                        ¥
                  -k    "quekind=disk                    ¥
                        quegrp01"                         ¥
                  -o    "aj = yes"                       ¥
                  -v    aptpr501
#
#--- コネクション定義
#
    mcftalccn    -c    INTCN01                                ¥
                  -p    ncsb                                ¥
                  -g    "sndbuf    =1                    ¥
                        rcvbuf    =2"                    ¥
                  -e    "msgbuf    =4                    ¥
                        count    =5"                    ¥
                  -i    auto                                ¥
                  -k    LINE01                              ¥
                  -b    "bretry = yes                    ¥
                        bretrycnt = 5                    ¥
                        bretryint = 10"                  ¥
                  -y    winner                              ¥
                  -u    NCS_B
    mcftalccn    -c    INTCN02                                ¥
                  -p    ncsb                                ¥
                  -g    "sndbuf    =1                    ¥
                        rcvbuf    =2"                    ¥
                  -e    "msgbuf    =4                    ¥
                        count    =5"                    ¥
                  -i    auto                                ¥
                  -k    LINE02                              ¥
                  -b    "bretry = yes                    ¥
                        bretrycnt = 5                    ¥
                        bretryint = 10"                  ¥
                  -y    winner                              ¥
                  -u    NCS_B
#
#--- コネクショングループ定義の終了
#
    mcftgrpdcn
#
#--- 終わり
#

```



# 6

## 運用コマンド

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

---

TP1/NET/NCSB 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 )

---

## TP1/NET/NCSB の運用コマンド

TP1/NET/NCSB は、オンライン中に運用コマンドを入力できます。

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

表 6-1 TP1/NET/NCSB で使用する運用コマンド

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

(凡例)

○ : 組み込み、または実行ができます。

× : 組み込み、または実行ができません。

## mcfactcn (コネクションの確立)

---

### 形式

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

### 機能

コネクションを確立します。

### オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID ~ 1 ~ 8 文字の識別子

確立するコネクションのコネクション 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。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定の場合も、引用符 (") で囲みます。

\*: すべてのコネクションを確立します。

先行文字列 \*: 先行文字列で始まるすべてのコネクションを確立します。

< 複数指定の例 > cnn1, cnn2, cnn3 を指定する場合

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

< 一括指定の例 > cnn で始まるすべてのコネクションを指定する場合

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

-g コネクショングループ名 ~ 1 ~ 8 文字の識別子

確立するコネクショングループの名称を指定します。コネクショングループ内のすべてのコネクションを確立します。

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

## 6. 運用コマンド

### mcftacten (コネクションの確立)

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

また、コネクショングループ名は、\* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクショングループ名を混在して指定できません。一括指定の場合も、引用符 (") で囲みます。

\*: すべてのコネクショングループを確立します。

先行文字列 \*: 先行文字列で始まるすべてのコネクショングループを確立します。

< 複数指定の例 > gnn1, gnn2, gnn3 を指定する場合

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

< 一括指定の例 > gnn で始まるすべてのコネクションを指定する場合

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

### 出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftacten 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	mcftacten 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	mcftacten コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftacten 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
KFCA15600-I	コネクションを確立しました。	メッセージログファイル
KFCA15603-E	コネクション確立に失敗しました。	メッセージログファイル
KFCA15610-I	回線をオープンしました。	メッセージログファイル
KFCA15611-E	回線障害が発生しました。	メッセージログファイル
KFCA15630-E	MCF 運用コマンド処理中に異常が発生しました。	メッセージログファイル

出力メッセージ ID	内容	出力先
KFCA15632-E	コネクションが確立済みのため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15633-E	コネクション確立処理中のため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15634-E	コネクション解放処理中のため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15697-E	内部処理実行中に異常が発生しました。処理を続行します。	メッセージログファイル
KFCA15699-E	内部処理実行中に異常が発生しました。	メッセージログファイル
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] -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。

```
-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。

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。同一論理端末名称は重複して指定できません。

また、論理端末名称は、\* を使って一括指定ができます。一括指定は一つだけ指定できません。一括指定と一括指定以外の論理端末名称を混在して指定できません。一括指定の場合も、引用符 (") で囲みます。

\*: すべての論理端末の閉塞を解除します。

先行文字列 \*: 先行文字列で始まるすべての論理端末の閉塞を解除します。

<複数指定の例> len1, len2, len3 を指定する場合

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

<一括指定の例> len で始まるすべての論理端末を指定する場合

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

### 出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcfactle 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  
mcfactle (論理端末の閉塞解除)

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	mcfactle 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	mcfactle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcfactle 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
KFCA10391-E	mcfactle コマンドはサポートされていません。	標準エラー出力
KFCA10395-E	指定したコネクションには未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10503-I	ヘルプメッセージ	標準出力
KFCA15630-E	MCF 運用コマンド処理中に異常が発生しました。	メッセージログファイル
KFCA15631-E	コネクションが未確立のため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15636-E	論理端末が閉塞解除中のため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15650-I	論理端末を閉塞解除しました。	メッセージログファイル
KFCA15699-I	内部処理実行中に異常が発生しました。	メッセージログファイル
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 | -g コネクショングループ名} [-f]
```

### 機能

コネクションを解放します。

### オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定の場合も、引用符 (") で囲みます。

\*: すべてのコネクションを解放します。

先行文字列 \*: 先行文字列で始まるすべてのコネクションを解放します。

< 複数指定の例 >    cnn1, cnn2, cnn3 を指定する場合

```
-c "cnn1    cnn2    cnn3"
```

< 一括指定の例 >    cnn で始まるすべてのコネクションを指定する場合

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

-g コネクショングループ名    ~    1 ~ 8 文字の識別子

解放するコネクショングループの名称を指定します。コネクショングループ内のすべてのコネクションを解放します。

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

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

また、コネクショングループ名は、\* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクショングループ名を混在して指定できません。一括指定の場合も、引用符 (") で囲みます。

\* : すべてのコネクショングループを解放します。

先行文字列 \* : 先行文字列で始まるすべてのコネクショングループを解放します。

< 複数指定の例 > gnn1 , gnn2 , gnn3 を指定する場合  
-g "gnn1 gnn2 gnn3"

< 一括指定の例 > gnn で始まるすべてのコネクションを指定する場合  
-g "gnn\*"
  
-f

該当するコネクションを強制的に解放します。

このオプションを指定すると、該当するコネクションが仕掛り中の場合、仕掛り中の処理をしないで無条件にコネクショングループを解放します。

このオプションの指定を省略すると、該当するコネクションが仕掛り中の場合、mcfstdctcn 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	mcfstdctcn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcfstdctcn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcfstdctcn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcfstdctcn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力

## 6. 運用コマンド

mcftdctn (コネクションの解放)

出力メッセージ ID	内容	出力先
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
KFCA15601-I	コネクションを解放しました。	メッセージログファイル
KFCA15630-E	MCF 運用コマンド処理中に異常が発生しました。	メッセージログファイル
KFCA15631-E	コネクションが未確立のため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15633-E	コネクション確立処理中のため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15634-E	コネクション解放処理中のため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15641-E	コネクションが確立状態で、かつメッセージ送信処理中のため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15697-E	内部処理実行中に異常が発生しました。処理を続行します。	メッセージログファイル
KFCA15699-E	内部処理実行中に異常が発生しました。	メッセージログファイル
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] -l 論理端末名称
```

### 機能

論理端末を閉塞します。

### 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、mcftdctl コマンドを実行します。

```
-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。

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。同一論理端末名称は、重複して指定できません。

また、論理端末名称は、\* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外の論理端末名称を混在して指定できません。一括指定の場合も、引用符 (") で囲みます。

\*: すべての論理端末を閉塞します。

先行文字列 \*: 先行文字列で始まるすべての論理端末を閉塞します。

< 複数指定の例 > len1, len2, len3 を指定する場合

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

< 一括指定の例 > len で始まるすべての論理端末を指定する場合

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

### 出力メッセージ

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

6. 運用コマンド  
mcftdctl (論理端末の閉塞)

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	mcftdctl コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftdctl コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftdctl コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定した接続は登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定された接続 ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10395-E	指定した接続には未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10504-I	ヘルプメッセージ	標準出力
KFCA15630-E	MCF 運用コマンド処理中に異常が発生しました。	メッセージログファイル
KFCA15635-E	論理端末が閉塞中のため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15637-E	論理端末に送信処理中のメッセージがあるため運用コマンドは受け付けられません。	メッセージログファイル
KFCA15651-I	論理端末を閉塞しました。	メッセージログファイル
KFCA15699-I	内部処理実行中に異常が発生しました。	メッセージログファイル
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

### 注意事項

メッセージの受信仕掛り中にこのコマンドを実行した場合、論理端末が閉塞状態でも、メッセージを受信します。論理端末閉塞によるメッセージ受信処理への影響はありません。

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

## mcftlscn ( コネクションの状態表示 )

---

### 形式

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

### 機能

コネクションの状態を標準出力に表示します。

### オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID    ~    1 ~ 8 文字の識別子

状態を表示するコネクションのコネクション ID を指定します。

コネクション ID は、一度に 8 個まで指定できます。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。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定の場合も、引用符 (") で囲みます。

\*: すべてのコネクションの状態を表示します。

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

< 複数指定の例 >    cnn1, cnn2, cnn3 を指定する場合

```
-c "cnn1    cnn2    cnn3"
```

< 一括指定の例 >    cnn で始まるすべてのコネクションを指定する場合

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

-g コネクショングループ名    ~    1 ~ 8 文字の識別子

状態を表示するコネクショングループの名称を指定します。コネクショングループ内のすべてのコネクションの状態を表示します。

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

## 6. 運用コマンド

mcftlscn (コネクションの状態表示)

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

また、コネクショングループ名は、\* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクショングループ名を混在して指定できません。一括指定の場合も、引用符 (") で囲みます。

\* : すべてのコネクショングループの状態を表示します。

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

<複数指定の例> gnn1, gnn2, gnn3 を指定する場合

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

<一括指定の例> gnn で始まるすべてのコネクションを指定する場合

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

```
-d
```

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

このオプションの指定を省略すると、コネクションの状態だけを表示します。

### 出力形式

```
mmm cccccccc ppp sssss gggggggg  
  llllllll ttt ssvv 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 cccccccc ppp sssss gggggggg」の行だけ出力されます。

- mmm : MCF 識別子
- cccccccc : コネクション ID
- ppp : プロトコル種別  
CSB...NCSB プロトコル
- sssss : コネクション状態  
ACT...確立  
ACT/B...確立処理中  
DCT...解放  
DCT/B...解放処理中
- gggggggg : コネクショングループ名
- llllllll : 論理端末名称
- ttt : 論理端末の端末タイプ  
ANY...any 型
- ss : 保守情報 1

- vv : 保守情報 2
- xxxx : 保守情報 3

### 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
KFCA10361-I	標準情報	標準出力
KFCA10362-I	詳細情報	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
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]
          -l 論理端末名称 [-q] [-t]
```

### 機能

論理端末の状態を標準出力に表示します。

### オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

`-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。

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。同一論理端末名称は、重複して指定できません。

また、論理端末名称は、\* を使って一括指定ができます。一括指定は一つだけ指定できません。一括指定と一括指定以外の論理端末名称を混在して指定できません。一括指定の場合も、引用符 (") で囲みます。

\* : すべての論理端末の状態を表示します。

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

< 複数指定の例 > len1, len2, len3 を指定する場合

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

< 一括指定の例 > len で始まるすべての論理端末を指定する場合

```
-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 llllllll sss {ggg} {tttt}
    PRIO xxxxxxxxxxxx yyyyyyyyyy zzzzzzzzzz
    NORM      :           :           :
    iii ooo
```

- mmm : MCF 識別子
- llllllll : 論理端末名称
- sss : 論理端末状態  
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 オプション指定時だけ表示)  
NOS...非セキュア状態  
SEC...セキュア状態
- tttt : 論理端末のテストモード状態 (TP1/Message Control/Tester 使用時だけ表示します)  
TEST...テストモード  
空白...非テストモード
- 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 : 入力キューの入力の保留状態 (-q オプション指定時だけ表示)  
NOH...保留解除  
HLD...保留
- ooo : 出力キューの出力の保留状態 (-q オプション指定時だけ表示)  
NOH...保留解除  
HLD...保留

## 出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftlsle コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力

## 6. 運用コマンド

### mcftlsle (論理端末の状態表示)

出力メッセージ ID	内容	出力先
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftlsle コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10360-I	状態表示を開始します。	標準出力
KFCA10364-I	表示情報	標準出力
KFCA10365-I	表示情報	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10371-I	mcftlsle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftlsle コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定したコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10395-E	指定したコネクションには未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10505-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

# 7

## 組み込み方法

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

---

7.1 TP1/NET/NCSB 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/NCSB の組み込みの流れ

---

TP1/NET/NCSB 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/NCSB を起動するためには、ユーザが MCF メイン関数をコーディングし、コンパイルおよびリンケージしておく必要があります。詳細は、「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/NCSB を実行するために、MCF サービス名をシステムサービス構成定義で定義しておく必要があります。

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

詳細は、マニュアル「OpenTP1 システム定義」を参照してください。

### (3) システムサービス情報定義ファイルの作成

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

### (4) 定義オブジェクトファイルの作成

OpenTP1 のネットワークコミュニケーション定義の各ソースファイルから定義オブジェクトファイルを生成します。詳細は、「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/NCSB は、OpenTP1 プロセスサービスによって起動されます。

TP1/NET/NCSB を起動するには、ユーザが MCF メイン関数をコーディングし、コンパイル、およびリンケージを行って TP1/NET/NCSB 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、mcfplncsb 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。

MCF メイン関数のコーディング概要を図 7-1 および図 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は、次のファイルで提供されています。

- ANSI C, C++ の場合  
/BeTRAN/examples/mcf/NCSB/cmlib/ansi/com.c
- K&R 版 C の場合  
/BeTRAN/examples/mcf/NCSB/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 <dcmsbm.h>                                /*TP1/NET/NCSB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 */
extern DCLONG msgsend01(dcmcf_uoc_mout_n *); /*出力メッセージ編集UOC関数extern宣言 */
extern DCLONG msgrou01(dcmcf_uoc_mout_n *); /*迂回判定UOC関数extern宣言 */      2.

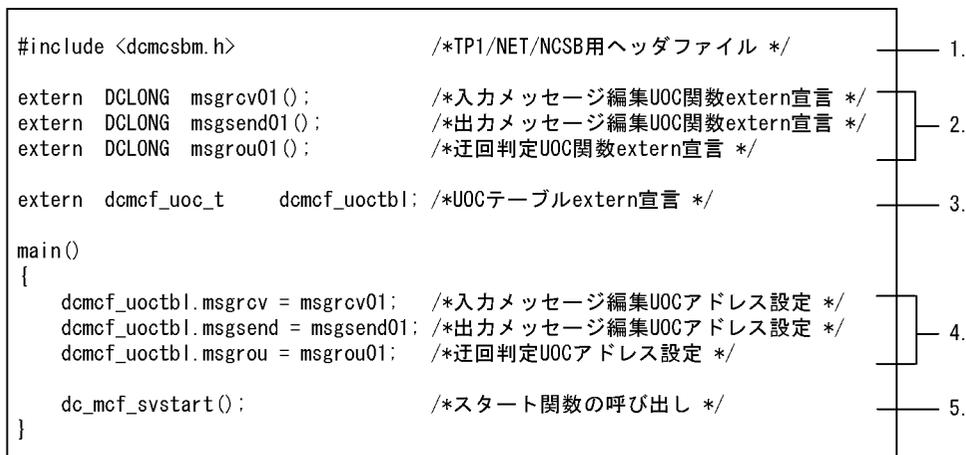
extern dcmcf_uoc_t dcmcf_uoctbl;                  /*UOCテーブルextern宣言 */          3.

int main()
{
    dcmcf_uoctbl.msgrcv = (dcmcf_uocfunc)msgrcv01;
                                                /*入力メッセージ編集UOCアドレス設定 */
    dcmcf_uoctbl.msgsend = (dcmcf_uocfunc)msgsend01;
                                                /*出力メッセージ編集UOCアドレス設定 */
    dcmcf_uoctbl.msgrou = (dcmcf_uocfunc)msgrou01;
                                                /*迂回判定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)



1. TP1/NET/NCSB 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。

使用する UOC に対するコーディングだけが必要です。

3. UOC テーブルを extern 宣言します。入力メッセージ編集 UOC または出力メッセージ編集 UOC を使用する場合、このとおりにコーディングしてください。

4. 各 UOC 関数のアドレスを、次に示すシステム提供変数に設定します。

```

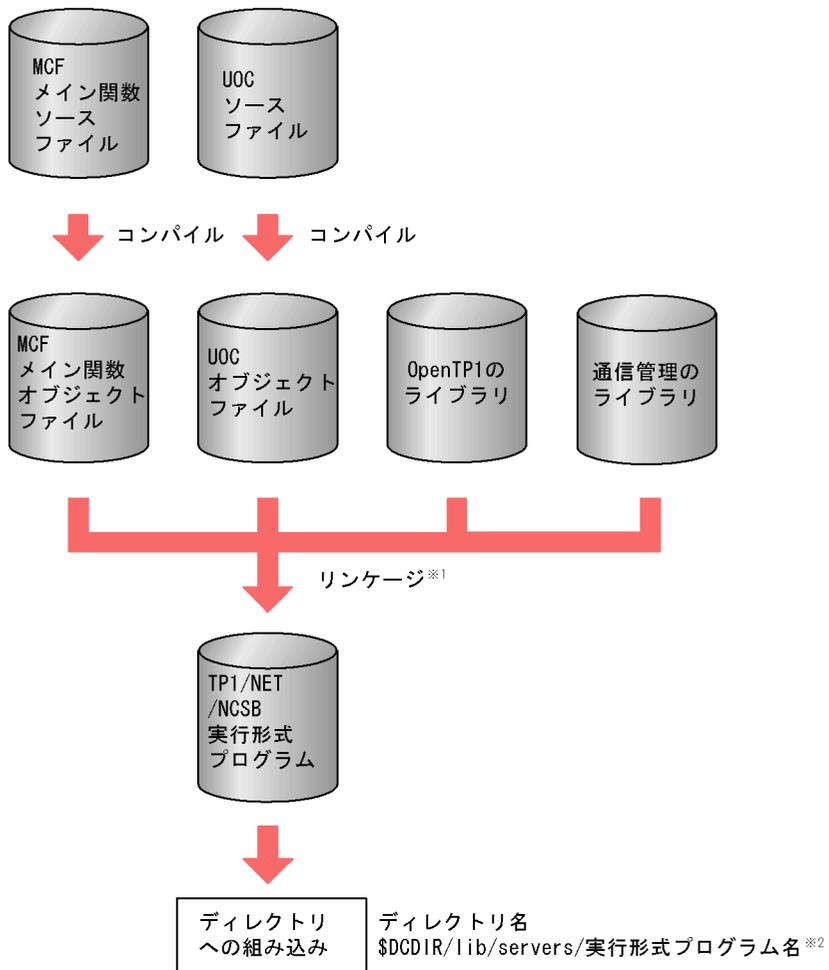
dcmcf_uoctbl.msgrcv /*入力メッセージ編集UOCアドレス*/
dcmcf_uoctbl.msgsend /*出力メッセージ編集UOCアドレス*/
dcmcf_uoctbl.msgrou /*迂回判定UOCアドレス設定*/

```

使用する UOC に対するコーディングだけが必要です。

5. スタート関数を呼び出します。必ずコーディングしてください。

図 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  
mcfplncsb コマンドでリンケージします。

注 2  
TP1/NET/NCSB 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

---

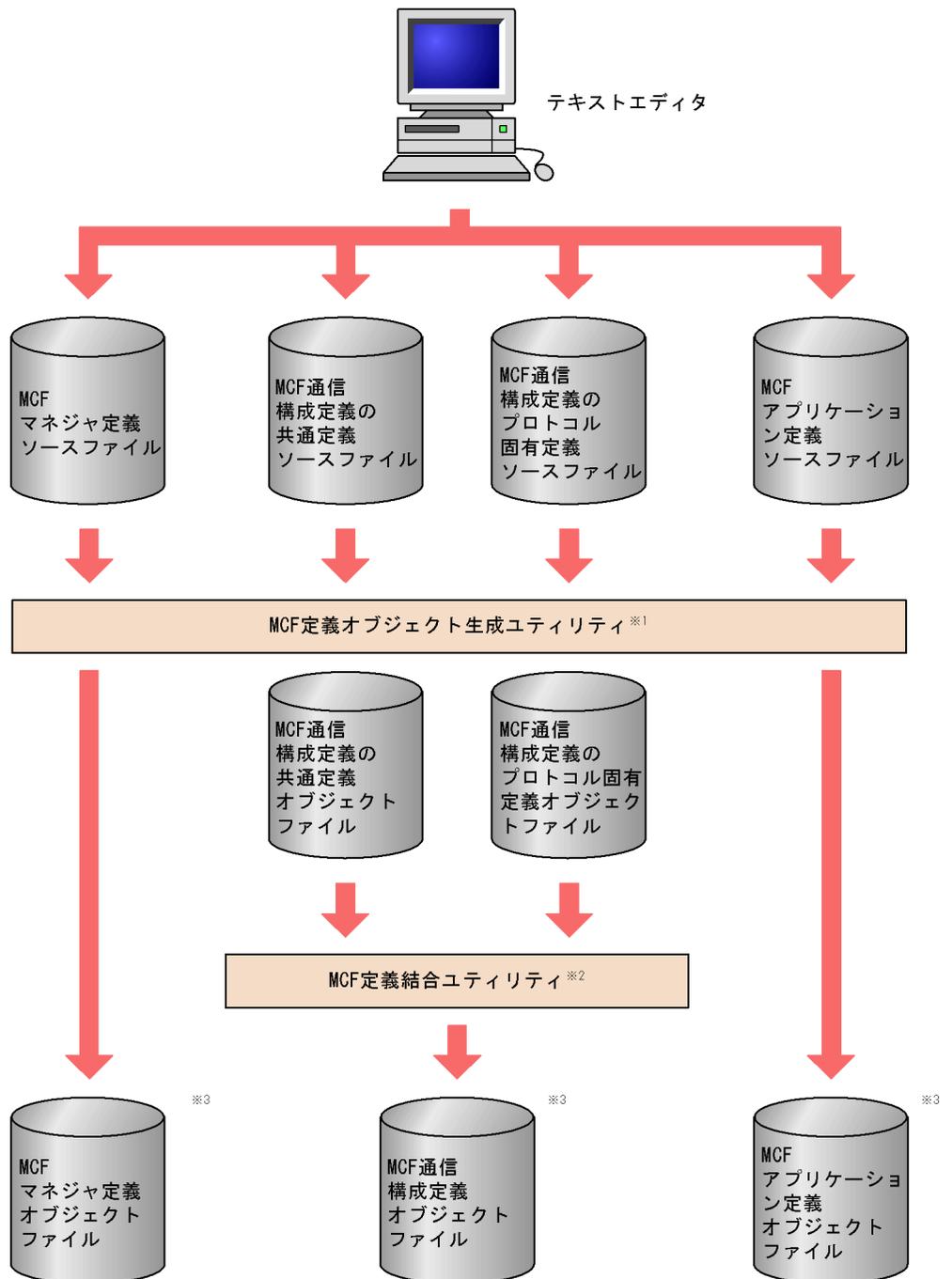
定義オブジェクトファイルを次の手順で作成します。

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ないでください。変更した場合、再開始時に正常に動作しないおそれがあるためご注意ください。

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

定義オブジェクトファイルの作成方法の概要を次の図に示します。

図 7-4 定義オブジェクトファイルの作成方法の概要



注 1

次に示すコマンドで生成します。

## 7. 組み込み方法

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

mcfXXXX は、ソースファイルごとに異なります。

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

### 注 2

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

```
mcfllink -i 共通定義オブジェクトファイル名  
          TP1/NET/NCSB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/NCSB の障害時の処理とユーザの対策について説明します。

---

### 8.1 障害の種類と対応処理

## 8.1 障害の種類と対応処理

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

TP1/NET/NCSB 運用中の障害と対応処理について、障害の種類ごとに示します。

### (1) コネクション障害

表 8-1 コネクション障害の種類と対応処理

障害内容	TP1/NET/NCSB の処理	ユーザの処理
コネクション確立時コネクション障害による確立失敗	<ol style="list-style-type: none"> <li>(1) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。</li> <li>(2) コネクション確立失敗を通知するメッセージログ (KFCA15603-E) を出力します。</li> <li>(3) CERREVT (コネクション障害) を起動します。</li> </ol>	障害の要因を取り除いたあと、mcftactcn コマンドを入力します。
通信管理マクロ異常終了	<ol style="list-style-type: none"> <li>(1) メッセージ送信中の障害の場合、送信メッセージの迂回判定 UOC を呼び出します。迂回できる場合、ほかのコネクションに迂回します。</li> <li>(2) 送信メッセージの迂回判定 UOC で迂回しないことを指示された場合、出力キュー上からメッセージを廃棄し、MDELEVT 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 (KFCA15602-E) を出力します。</li> <li>(4) CERREVT (コネクション障害) を起動します。</li> <li>(5) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。</li> <li>(6) CERREVT (論理端末閉塞) を起動します。</li> </ol>	障害の要因を取り除いたあと、mcftactcn コマンドを入力します。
受信バッファオーバーフロー	<ol style="list-style-type: none"> <li>(1) 受信メッセージを破棄します。</li> <li>(2) コネクションを解放します。</li> <li>(3) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。</li> <li>(4) CERREVT (コネクション障害) を起動します。</li> <li>(5) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。</li> <li>(6) CERREVT (論理端末閉塞) を起動します。</li> </ol>	構成定義を修正します。

障害内容	TP1/NET/NCSB の処理	ユーザの処理
受信バッファ不足	(1) 受信メッセージを破棄します。 (2) 資源不足を通知するメッセージログ (KFCA13213-E) を出力します。 (3) コネクションを解放します。 (4) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。 (5) CERREVT (コネクション障害) を起動します。 (6) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。 (7) CERREVT (論理端末閉塞) を起動します。	構成定義を修正します。
通信管理マクロエラーリターン (構成定義不正)	(1) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。 (2) CERREVT (コネクション障害) を起動します。	構成定義を修正します。

注

コネクショングループ内のすべてのコネクションの障害時に、論理端末は閉塞します。

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

表 8-2 受信スケジュール関係の障害の種類と対応処理

障害内容	TP1/NET/NCSB の処理	ユーザの処理
ユーザアプリケーション名称未定義	(1) ERREVT1 を起動します。 (2) 入力キュー障害を通知するメッセージログ (KFCA10604-E) を出力します。 (3) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。 (4) 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。 (5) コネクションを解放します。 (6) CERREVT (コネクション障害) を起動します。 (7) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。 (8) CERREVT (論理端末閉塞) を起動します。	構成定義を修正します。
システムアプリケーション名未定義 <ul style="list-style-type: none"> <li>• ERREVT x</li> <li>• C x x x EVT</li> <li>• MDELEVT</li> </ul>	メッセージおよびイベントを廃棄します。	ありません。

8. 障害対策

障害内容	TP1/NET/NCSB の処理	ユーザの処理
MHP サービス, サービスグループ閉塞	<ol style="list-style-type: none"> <li>(1) サービス閉塞を通知するメッセージログ (KFCA11051-E) を出力します。</li> <li>(2) ERREVT2 を起動します。</li> <li>(3) 入力キュー障害を通知するメッセージログ (KFCA10604-E) を出力します。</li> <li>(4) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。</li> <li>(5) 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。</li> <li>(6) コネクションを解放します。</li> <li>(7) CERREVT (コネクション障害) を起動します。</li> <li>(8) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。</li> <li>(9) CERREVT (論理端末閉塞) を起動します。</li> </ol>	サービス, サービスグループの閉塞解除後, mcfactcn コマンドを入力します。
入力キュー書き込み障害	<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) ERREVT2 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 (KFCA15602-E) を出力します。</li> <li>(4) 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。</li> <li>(5) コネクションを解放します。</li> <li>(6) CERREVT (コネクション障害) を起動します。</li> <li>(7) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。</li> <li>(8) CERREVT (論理端末閉塞) を起動します。</li> </ol>	障害の要因を取り除いたあと, mcfactcn コマンドを入力します。
入力メッセージ編集 UOC エラー	<ol style="list-style-type: none"> <li>(1) UOC エラーリターンを通知するメッセージログ (KFCA10611-E), または UOC パラメタ不正を通知するメッセージログ (KFCA10620-E) を出力します。</li> <li>(2) 受信メッセージを破棄します。</li> <li>(3) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。</li> <li>(4) 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。</li> <li>(5) コネクションを解放します。</li> <li>(6) CERREVT (コネクション障害) を起動します。</li> <li>(7) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。</li> <li>(8) CERREVT (論理端末閉塞) を起動します。</li> </ol>	入力メッセージ編集 UOC の処理を見直します。

障害内容	TP1/NET/NCSB の処理	ユーザの処理
ユーザアプリケーション名形式不正	(1) アプリケーション名取得失敗を通知するメッセージログ (KFCA10610-E) を出力します。 (2) ERREVT1 を起動します。 (3) 受信メッセージを破棄します。 (4) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。 (5) 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。 (6) コネクションを解放します。 (7) CERREVT (コネクション障害) を起動します。 (8) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。 (9) CERREVT (論理端末閉塞) を起動します。	障害の要因を取り除いたあと、mcfactctn コマンドを入力します。

注

コネクショングループ内のすべてのコネクションの障害時に、論理端末は閉塞します。

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

表 8-3 送信スケジュール関係の障害の種類と対応処理

障害内容	TP1/NET/NCSB の処理	ユーザの処理
出力キュー読み込み障害	(1) 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 (2) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。 (3) 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。 (4) コネクションを解放します。 (5) CERREVT (コネクション障害) を起動します。 (6) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。 (7) CERREVT (論理端末閉塞) を起動します。	障害の要因を取り除いたあと、mcfactctn コマンドを入力します。

8. 障害対策

障害内容	TP1/NET/NCSB の処理	ユーザの処理
出力メッセージ編集 UOC エラーリターン, または UOC インタ フェースエラー	(1) UOC エラーリターンを通知するメッセージログ (KFCA10611-E), または UOC パラメータ不正を通知するメッセージログ (KFCA10620-E または KFCA15661-E) を出力します。 (2) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。 (3) 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。 (4) コネクションを解放します。 (5) CERREVT (コネクション障害) を起動します。 (6) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。 (7) CERREVT (論理端末閉塞) を起動します。	出力メッセージ編集 UOC の処理を見直します。
メッセージ消去障害	(1) 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 (2) 処理を続行します。	ありません。
送信バッファ不足	(1) ローカルメモリ不足を通知するメッセージログ (KFCA10618-E) を出力します。 (2) 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 (3) 送信中断を通知するメッセージログ (KFCA10608-E) を出力します。 (4) コネクション障害を通知するメッセージログ (KFCA15602-E) を出力します。 (5) 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。 (6) コネクションを解放します。 (7) CERREVT (コネクション障害) を起動します。 (8) 論理端末閉塞を通知するメッセージログ (KFCA15651-I) を出力します。 (9) CERREVT (論理端末閉塞) を起動します。	構成定義を修正します。
送信メッセージの迂回判定 UOC エラーリターン, または UOC インタ フェースエラー	(1) UOC エラーリターンを通知するメッセージログ (KFCA15660-E), またはパラメータ不正を通知するメッセージログ (KFCA15661-E) を出力します。 (2) メッセージを迂回しないで, 出力キュー上から廃棄し, MDELEVT を起動します。	送信メッセージの迂回判定 UOC の処理を見直します。

注

コネクショングループ内のすべてのコネクションの障害時に, 論理端末は閉塞します。

## (4) UAP 障害

表 8-4 UAP 障害の種類と対応処理

障害内容	TP1/NET/NCSB の処理	ユーザの処理
セグメント受信前の UAP 障害	ERREVT2 を起動します。	UAP の処理を見直します。
セグメント受信後の UAP 障害	ERREVT3 を起動します。	UAP の処理を見直します。

## (5) MCF 障害

表 8-5 MCF 障害の種類と対応処理

障害内容	TP1/NET/NCSB の処理	ユーザの処理
内部論理矛盾 他 P.P とのインタフェースエラー	(1) 障害情報取得を通知するメッセージログ (KFCA10189-E) を出力します。 (2) 内部論理矛盾を通知するメッセージログ (KFCA15697-E, KFCA15698-E, または KFCA15699-E) を出力します。 (3) KFCA15697-E 出力の場合, 処理を続行します。 KFCA15698-E 出力の場合, コネクション切断後, 処理を続行します。 KFCA15699-E 出力の場合, 異常終了します。	ありません。



# 付録

---

付録 A メッセージ送受信の処理の流れ

---

付録 B 障害発生時の処理の流れ

---

付録 C 理由コード一覧

---

付録 D 旧製品からの移行に関する注意事項

---

付録 E バージョンアップ時の変更点

---

付録 F 用語解説

---

## 付録 A メッセージ送受信の処理の流れ

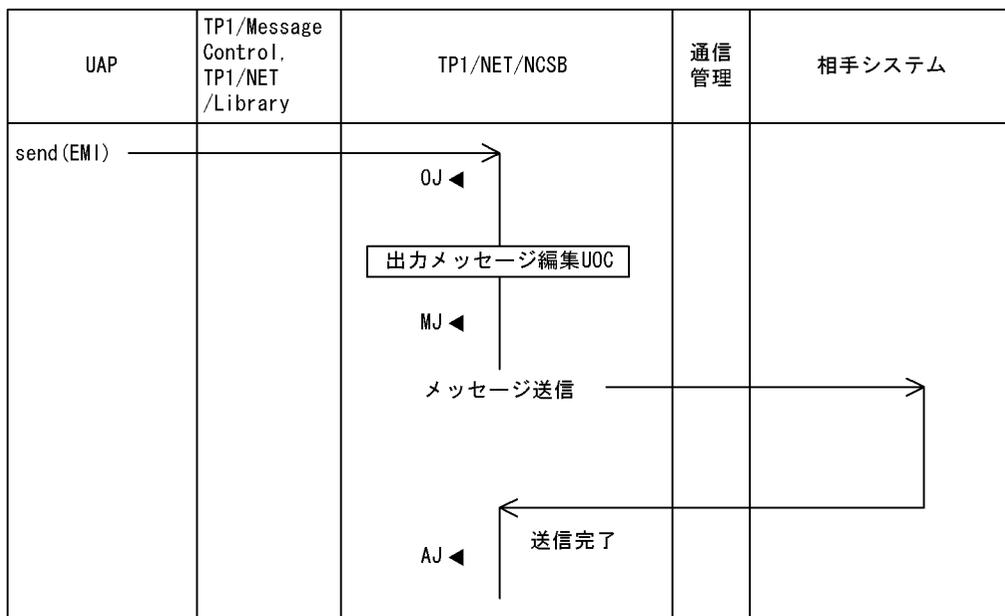
メッセージを送受信するときのデータの流れ，ジャーナルの取得タイミングなどを示します。

### 付録 A.1 メッセージの送信

一方送信メッセージを送信するときの処理の流れを，図 A-1 ~ 図 A-2 に示します。

#### (1) 単一セグメントの場合

図 A-1 メッセージ送信の処理の流れ (単一セグメントの場合)

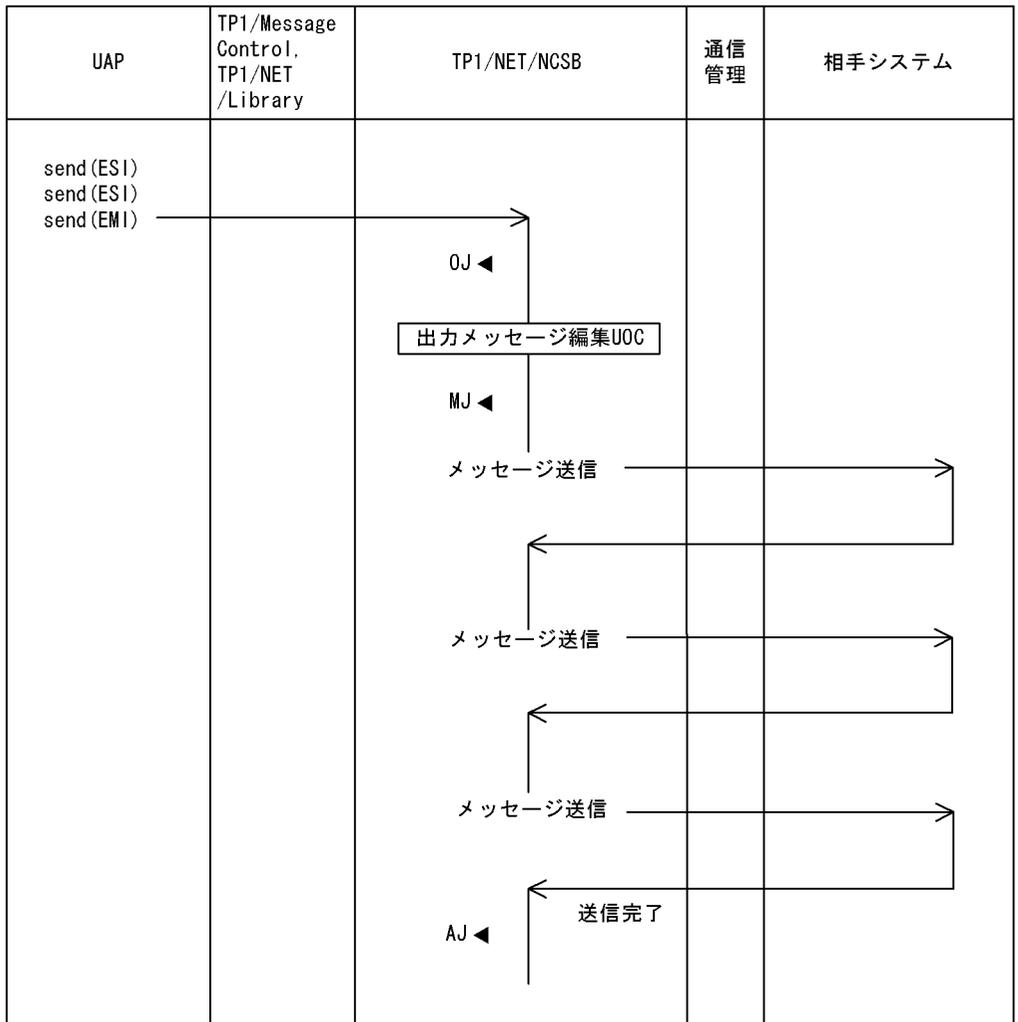


(凡例)

- 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
- AJ ◀ : メッセージ送信完了ジャーナル取得

(2) 複数セグメントの場合

図 A-2 メッセージ送信の処理の流れ (複数セグメントの場合)



(凡例)

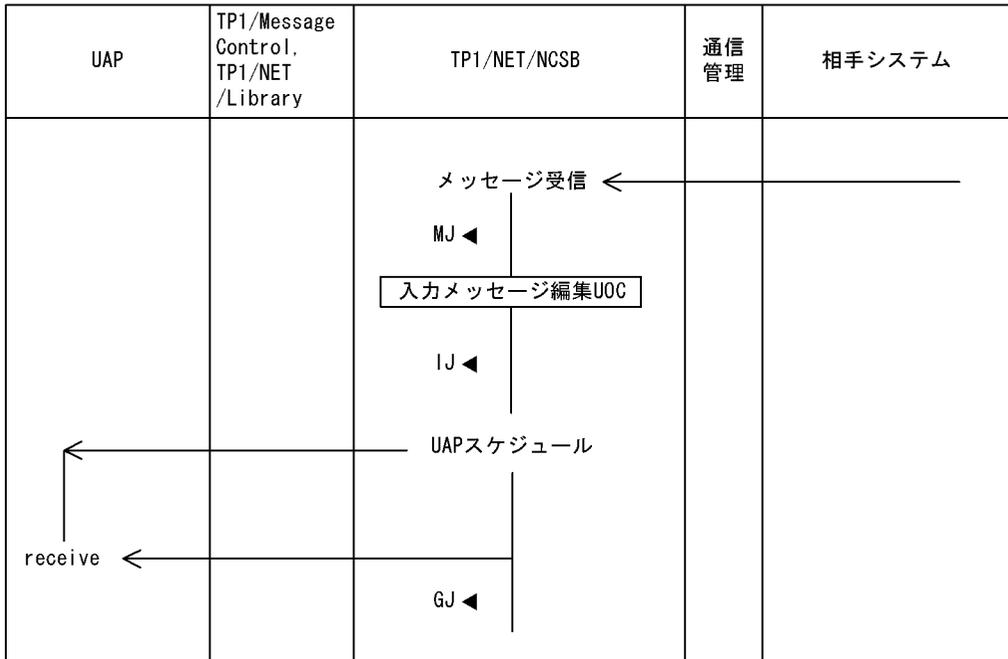
- OJ ◀: メッセージ出カジャーナル取得
- MJ ◀: メッセージジャーナル取得
- AJ ◀: メッセージ送信完了ジャーナル取得

付録 A.2 メッセージの受信

一方送信メッセージを受信するときの処理の流れを, 図 A-3 ~ 図 A-4 に示します。

(1) 単一セグメントの場合

図 A-3 メッセージ受信の処理の流れ (単一セグメントの場合)

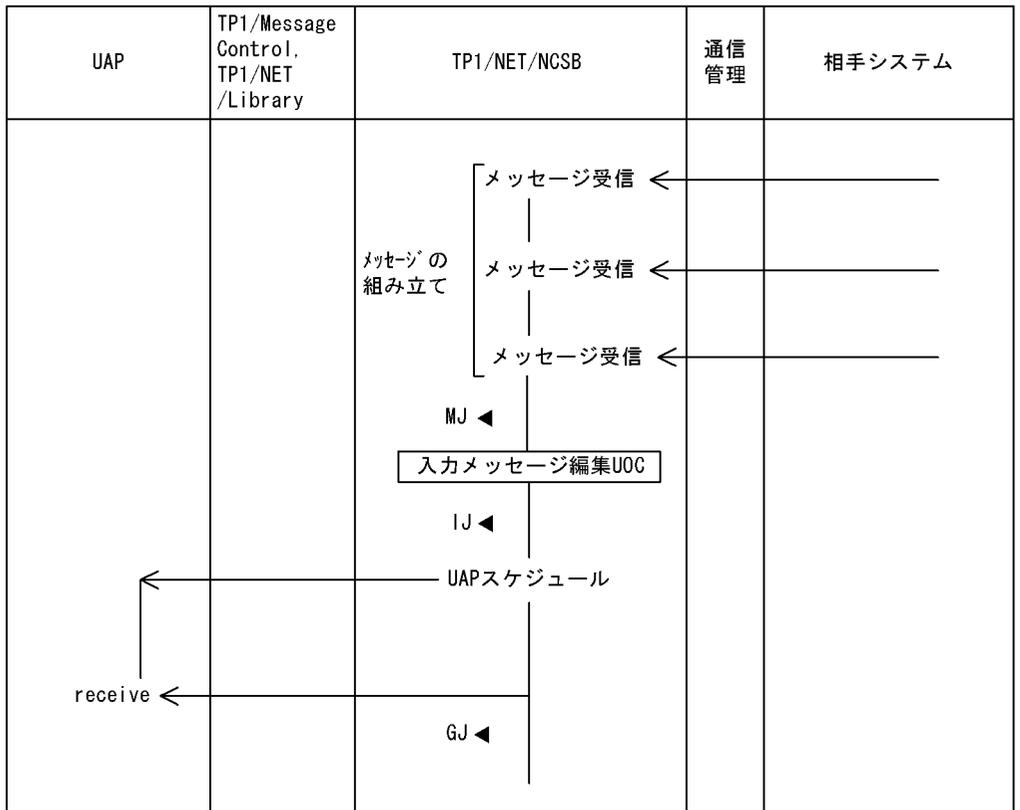


(凡例)

- 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

(2) 複数セグメントの場合

図 A-4 メッセージ受信の処理の流れ (複数セグメントの場合)



- (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 B 障害発生時の処理の流れ

障害発生時の処理の流れを図 B-1 ~ 図 B-5 に示します。

図 B-1 コネクション障害

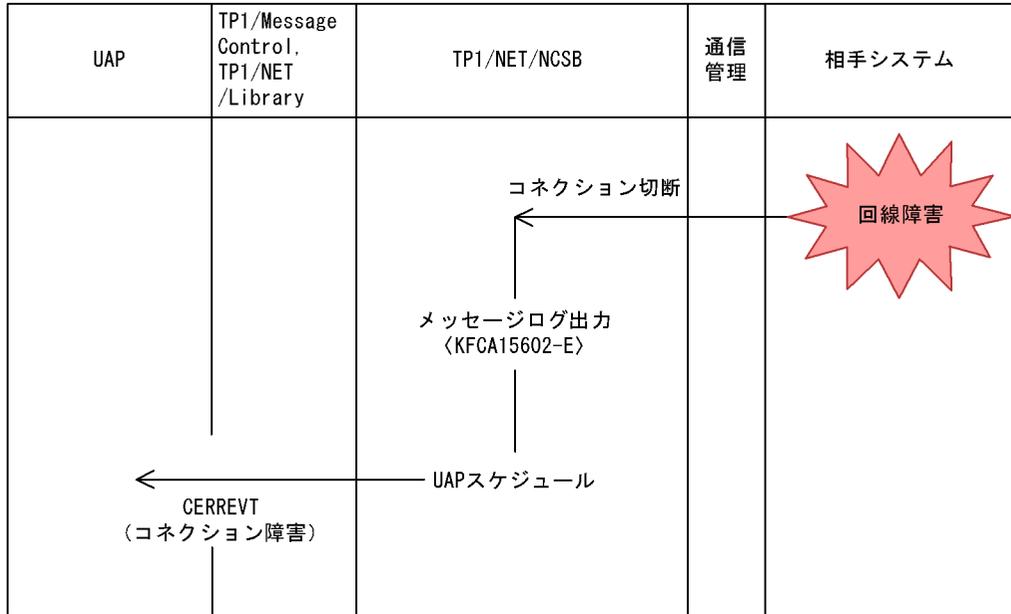




図 B-3 メッセージ送信中の回線障害（送信メッセージの迂回判定 UOC で迂回しない場合）

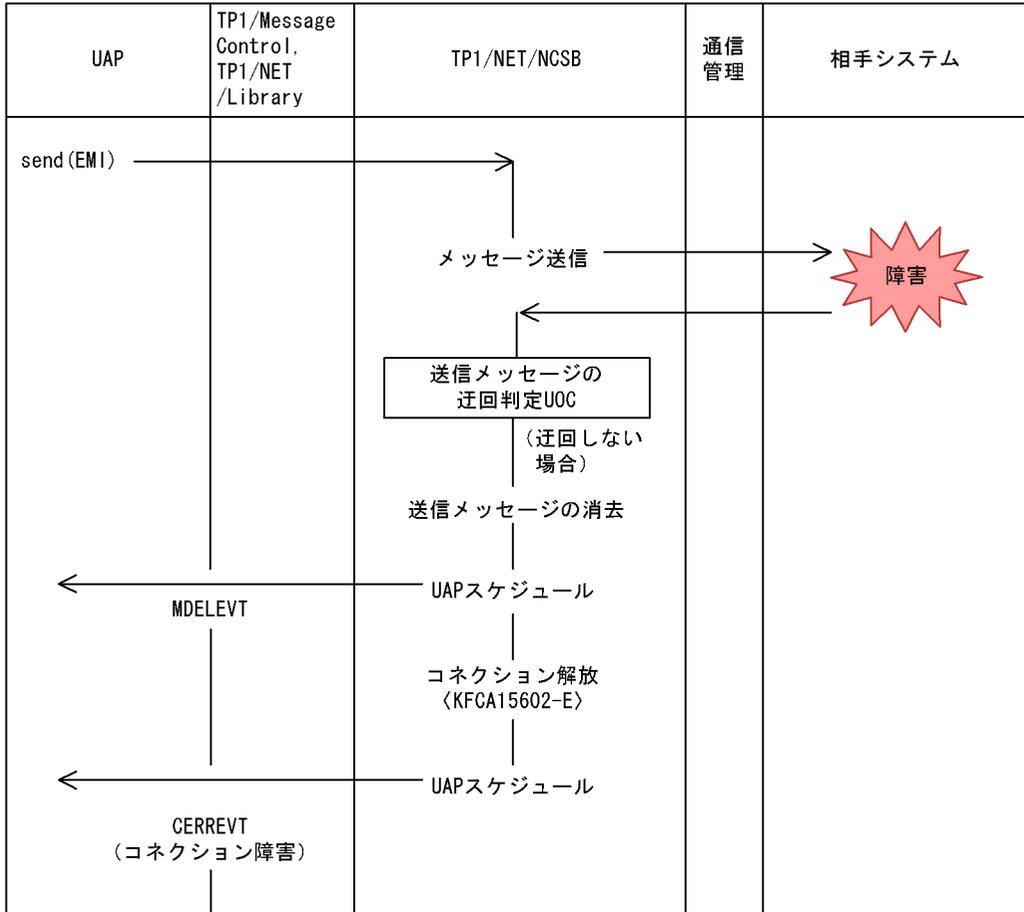


図 B-4 入力メッセージ編集 UOC エラーリターン

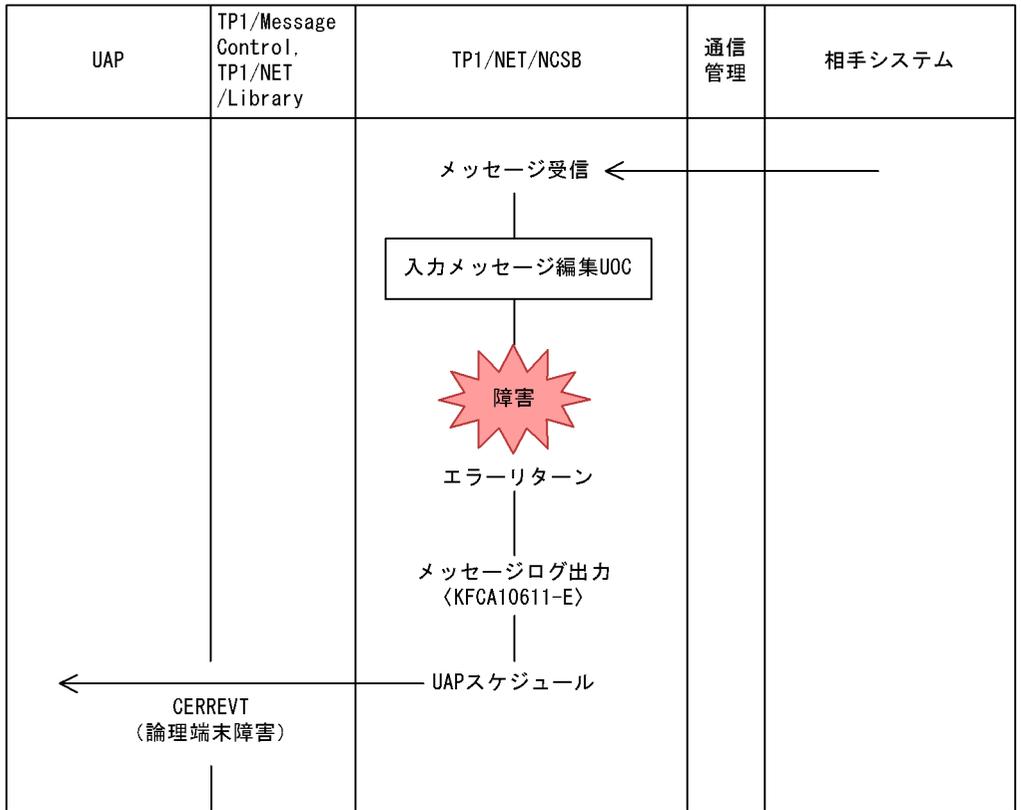
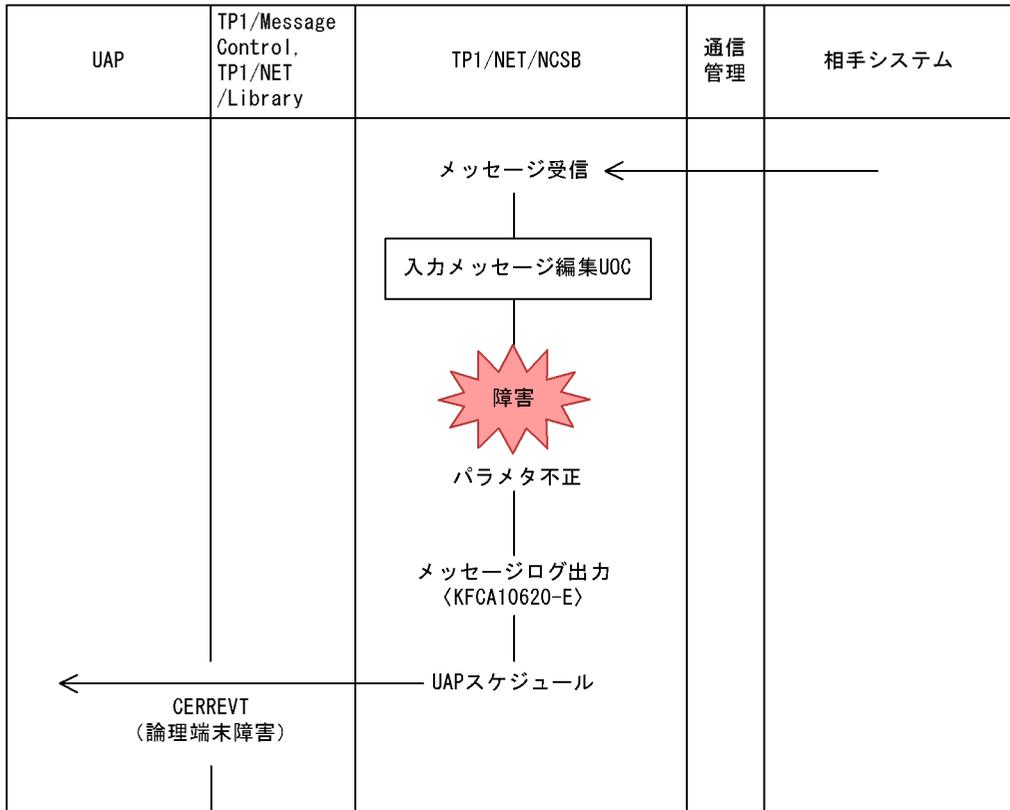


図 B-5 入力メッセージ編集 UOC パラメタ不正



## 付録 C 理由コード一覧

TP1/NET/NCSB が通知する ERREVT2, CERREVT および MDELEVT の理由コードを示します。

### 付録 C.1 ERREVT2 の理由コード

メッセージ廃棄通知イベント (ERREVT2) の理由コードおよび通知理由を次の表に示します。

表 C-1 ERREVT2 の理由コード

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

### 付録 C.2 CERREVT の理由コード

障害通知イベント (CERREVT) を通知したときの理由コードを示します。コンネクション障害の場合の理由コードを表 C-2 に, 論理端末障害の場合の理由コードを表 C-3 に示します。

表 C-2 コネクション障害の場合の理由コード

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	CERREVT の通知理由
DCMCSB_RSN1_MCF (00000001) MCF 障害	DCMCSB_RSN2_CNNC (00000000)	コネクション確立不可能
	DCMCSB_RSN2_RBFOF (00000002)	受信バッファオーバーフロー
	DCMCSB_RSN2_RBFST (00000003)	受信バッファ不足
	DCMCSB_RSN2_SBFOF (00000004)	送信バッファオーバーフロー
	DCMCSB_RSN2_SBFST (00000005)	送信バッファ不足
	DCMCSB_RSN2_TIMER (00000006)	応答監視タイマ起動失敗
	DCMCSB_RSN2_SNDERR (00000007)	送信失敗
	DCMCSB_RSN2_MEMST (00000008)	メモリ不足
	DCMCSB_RSN2_EVTNG (00000009)	MCF イベント起動失敗
	DCMCSB_RSN2_OTGET (0000000a)	出力キュー障害
	DCMCSB_RSN2_DCTCN_F (0000000b)	mcfstdcten -f による強制終了
	DCMCSB_RSN2_APL (0000000c)	アプリケーション名不正
	DCMCSB_RSN1_XNF (00000002) コネクション障害	(00020000)
(00060000)		XNF 未起動
(00100000)		回線二重起動
XNF/NCSB の終了コード, エラー詳細コード		XNF/NCSB からの障害通知の詳細については, マニュアル「通信管理 XNF/AS 解説・運用編」を参照してください。
DCMCSB_RSN1_UOC (00000003) UOC 障害	UOC からの詳細リターンコード	UOC エラーリターン
	DCMCSB_RSN2_BCNT (00000001)	使用バッファ数不正
	DCMCSB_RSN2_SEG (00000002)	有効セグメント不正

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	CERREVT の通知理由
	DCMCSB_RSN2_BADR (00000003)	編集バッファアドレス不正
	DCMCSB_RSN2_TIVL (00000004)	タイマ値範囲不正
	DCMCSB_RSN2_TICD (00000005)	タイマセット指示種別不正
	DCMCSB_RSN2_RERO (00000006)	迂回可否指示種別不正

表 C-3 論理端末障害の場合の理由コード

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	CERREVT の通知理由
DCMCSB_RSN1_MCF (00000001) MCF 障害	DCMCSB_RSN2_CNER (00000001)	コネクション障害によって論理端末が閉塞しました。コネクショングループ使用の場合は、グループ内のすべてのコネクションに障害が発生しました。

## 付録 C.3 MDELEVT の理由コード

迂回不可メッセージ廃棄通知イベント (MDELEVT) の理由コードおよび通知理由を次の表に示します。

表 C-4 MDELEVT の理由コード

理由コード 1 (16 進数字)	MDELEVT の通知理由
DCMCSB_DRSN_CAN (00000001)	送信メッセージの迂回判定 UOC が迂回しないことを指示
DCMCSB_DRSN_ERR (00000002)	出力メッセージ編集 UOC エラーリターン, または送信メッセージの迂回判定 UOC エラーリターン
DCMCSB_DRSN_NON (00000003)	送信メッセージの迂回判定 UOC 未登録
DCMCSB_DRSN_BUFST (00000004)	バッファ不足

---

## 付録 D 旧製品からの移行に関する注意事項

旧製品からの移行に関する注意事項について説明します。

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

表 D-1 バージョン 6 以前で使用していたソースファイルの互換性

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

## 付録 E バージョンアップ時の変更点

各バージョンでの変更点を次に示す分類ごとに示します。

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

### 付録 E.1 07-00 での変更点

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

TP1/NET/NCSB 07-00 での動作の変更点を次に示します。

表 E-1 TP1/NET/NCSB 07-00 での動作の変更点

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

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

---

## 付録 F 用語解説

### (英字)

---

#### AJ (メッセージ送信完了ジャーナル)

TP1/NET/NCSB で取得する履歴情報の一つです。メッセージの送信完了情報である、送信通番と送信先論理端末名称とで構成されます。

AJ の取得タイミングは相手システムにメッセージを送信し、その送信完了を受信した直後です。

#### GJ (メッセージ受信ジャーナル)

TP1/NET/NCSB で取得する履歴情報の一つです。論理端末名称、入力メッセージなどの情報で構成されます。

GJ の取得タイミングは UAP に receive 関数でメッセージを引き渡す直前です。

#### IJ (メッセージ入力ジャーナル)

入力キューに入力された情報です。論理端末名称、メッセージ種別、入力メッセージなどの情報で構成されます。

IJ の取得タイミングは相手システムから受信したメッセージを入力キューに入力する直前です。

#### MJ (メッセージジャーナル)

TP1/NET/NCSB で取得する情報の一つです。端末名称、メッセージ種別、入力メッセージ編集前のデータ、出力メッセージ編集後のデータなどの情報で構成されます。

MJ の取得タイミングはメッセージ送信前、出力メッセージ編集 UOC 処理後、および入力メッセージ編集 UOC 処理前です。

#### OJ (メッセージ出力ジャーナル)

出力キューに入力された情報です。メッセージの送信通番、論理端末名称、メッセージ種別、出力メッセージ、セグメント種別などの情報で構成されます。

OJ の取得タイミングは UAP から send を受け付けたときです。

### (力行)

---

#### 回線の迂回

回線障害が発生したとき、同じコネクション内の別の正常な回線を使用して、再び通信を試みる機能です。

#### コネクション

通信相手システムとの論理的な通信路です。MCF の通信管理側の通信接点であり、MCF と通信管理はコネクション単位にメッセージを送受信します。

## (ラ行)

---

### ラウンドロビン方式

実行順序を決めるスケジューリング方式の一つです。待ち行列の最後尾の次が先頭になる円状にして、常に一定方向で実行させることによって対象を平等な優先順位で扱います。

### 論理端末 (LE)

MCF の UAP 側の通信接点であり、MCF と UAP は論理端末を通してメッセージを送受信します。



---

# 索引

## A

---

AJ (メッセージ送信完了ジャーナル) 180  
any 14  
AP 間通信 2  
AP 間通信の概要 2  
AP 間通信の形態 3  
AP 間通信の仕組み 8  
AP 間通信メッセージの送受信 16  
ARU システム 115

## C

---

CBLDCMCF('RECEIVE') 36  
CBLDCMCF('RESEND') 42  
CBLDCMCF('SEND') 48  
CCLSEVT 95, 103  
CERREVT 95, 102  
CERREVT の理由コード 175  
COBOL 言語のメッセージ送受信 20  
COPNEVT 95, 103  
C 言語のメッセージ送受信 20

## D

---

dc\_mcf\_receive 22  
dc\_mcf\_resend 27  
dc\_mcf\_send 32

## E

---

ERREVT1 93, 96  
ERREVT2 93, 98  
ERREVT2 の理由コード 175  
ERREVT3 94, 99  
ERREVT4 94, 101

## G

---

GJ (メッセージ受信ジャーナル) 180

## I

---

IJ (メッセージ入力ジャーナル) 180

## M

---

max\_open\_fds 124  
max\_socket\_descriptors 123  
mftactcn 133  
mftactcle 136  
mftalcen 113  
mftalcle 118  
mftdctcn 138  
mftdctcle 141  
mftgrpcn 120  
mftgrped 121  
mftlscn 143  
mftlsle 146  
MCF アプリケーション定義 108  
MCF イベント 90  
MCF イベント一覧 90  
MCF イベントインタフェース 90  
MCF イベント情報 91  
MCF イベント情報の形式 (COBOL 言語) 96  
MCF イベント情報の形式 (C 言語) 92  
MCF イベント処理用 MHP 90, 92  
MCF イベント通知時のセグメント構成 91  
MCF イベントの共通ヘッダ 92  
MCF イベントの種類 90  
MCF サービス名の登録 150  
MCF 障害 163  
MCF 通信構成定義 108  
MCF 通信プロセスでアクセスするファイルの最大数 124  
MCF 定義オブジェクトの生成 125  
MCF で使用する定義ファイル 108  
MCF マネージャ定義 108  
MCF メイン関数のコーディング 151  
MCF メイン関数の作成 151  
MDELEVT 96, 104

MDELEVT の理由コード 177  
 MJ (メッセージジャーナル) 180  
 module 122

## N

---

NCS-B インタフェース 115  
 noans 14

## O

---

OJ (メッセージ出力ジャーナル) 180

## R

---

RECEIVE 55

## S

---

SEND 58

## T

---

TP1/NET/NCSB に固有の定義の一覧 110  
 TP1/NET/NCSB の運用コマンド 132  
 TP1/NET/NCSB の組み込みの流れ 150  
 TP1/NET/NCSB の定義 108  
 TP1/NET/NCSB を組み込んだソフトウェア  
 構成の例 4  
 TP1/NET/NCSB を使用したネットワーク構  
 成の例 2

## U

---

UAP 異常終了通知イベント 90  
 UAP 障害 163  
 UOC 68  
 UOC 作成上の注意事項 88  
 UOC で使用できる関数 88  
 UOC の異常処理 88  
 UOC の構造 88  
 UOC の実行タイミング 88

## X

---

XNF/AS 構成定義 127

## あ

---

アプリケーション属性定義 14  
 アプリケーション名 118  
 アプリケーション名の決定 68

## い

---

一方受信形態 3  
 一方送信メッセージ 3  
 一方送信メッセージの受信 17  
 一方送信メッセージの受信 (COBOL 言語)  
 36  
 一方送信メッセージの受信 (C 言語) 22  
 一方送信メッセージの送信 16  
 一方送信メッセージの送信 (COBOL 言語)  
 48  
 一方送信メッセージの送信 (C 言語) 32  
 一方送信メッセージの送信 (データ操作言  
 語) 58

## う

---

迂回不可メッセージ廃棄通知イベント 91  
 運用コマンド 131

## か

---

回線の迂回 180  
 回線名称 114  
 概要 1

## き

---

機能 7  
 キューグループ ID 119  
 旧製品からの移行に関する注意事項 178

## く

---

組み込み方法 149  
 クレジット情報データ通信システム 115

## こ

---

個人信用情報センタ 115  
 コネクション 8, 180

コネクション, コネクショングループ, およ  
 び論理端末の関係 8  
 コネクション ID 113  
 コネクションおよびコネクショングループの  
 解放 10  
 コネクションおよびコネクショングループの  
 確立 9  
 コネクション確立の流れ 10  
 コネクショングループ 8  
 コネクショングループ定義の開始 120  
 コネクショングループ定義の終了 121  
 コネクショングループと論理端末 12  
 コネクショングループ名 120  
 コネクション障害 158, 170  
 コネクション定義 113  
 コネクションとコネクショングループ 8  
 コネクションの解放 138  
 コネクションの確立 133  
 コネクションの状態表示 143  
 コネクションの切断 12  
 コンテンション発生時の優先 / 非優先 114

## し

---

システム構成例 126  
 システムサービス共通情報定義 123  
 システムサービス情報定義 122  
 システムサービス情報定義ファイルの作成  
 150  
 システム定義 107  
 実行形式プログラム名 122  
 受信スケジュール関係の障害 159  
 出力キュー 16  
 出力メッセージの編集 76  
 出力メッセージ編集 UOC インタフェース  
 77  
 障害通知イベント 91  
 障害の種類と対応処理 158  
 障害発生時の処理の流れ 170  
 状態通知イベント 91  
 信用金庫オンライン 115

## せ

---

セグメント 15  
 前提プログラム 4

## そ

---

送信スケジュール関係の障害 161  
 送信メッセージの迂回判定 80  
 送信メッセージの迂回判定 UOC 80  
 送信メッセージの迂回判定 UOC インタ  
 フェース 82  
 送信メッセージの通番編集 85  
 送信メッセージの通番編集 UOC 85  
 送信メッセージの通番編集 UOC インタ  
 フェース 87  
 ソケット用ファイル記述子の最大数 123  
 ソフトウェアの構成 4

## た

---

タイマ起動メッセージ廃棄通知イベント 90

## ち

---

地銀 CD 全国ネットサービスデータ通信シス  
 テム 115

## つ

---

通信管理プログラムと関連づける内容 126

## て

---

定義オブジェクトファイルの生成 154  
 定義の指定順序 111  
 定義の種類 110  
 定義例 127  
 ディスク出力メッセージ最大格納数 119  
 データ操作言語 (COBOL 言語) のメッセー  
 ジ送受信 20

## に

---

入力キュー 17  
 入力メッセージの編集 68

入力メッセージの編集とアプリケーション名の決定 68  
 入力メッセージ編集 UOC 68  
 入力メッセージ編集 UOC インタフェース 71  
 入力メッセージ編集 UOC エラーリターン 173  
 入力メッセージ編集 UOC パラメタ不正 174  
 任意型論理端末 14

## は

---

バッファ形式 1 15  
 バッファ形式 2 15

## ひ

---

非応答型 14

## ふ

---

不正アプリケーション名検出通知イベント 90  
 分岐送信形態 3  
 分岐送信形態の AP 間通信の例 3

## へ

---

ベーシック回線定義文 126

## み

---

未処理送信メッセージ廃棄通知イベント 91

## め

---

メッセージ受信用バッファグループ番号 113  
 メッセージ送受信インタフェース 19  
 メッセージ送受信インタフェースの一覧 20  
 メッセージ送受信の関数 (C 言語) 20  
 メッセージ送受信の関数に対応するプログラム (COBOL 言語) 20  
 メッセージ送受信の処理の流れ 166  
 メッセージ送受信の通信文 (データ操作言語) 21  
 メッセージ送信時のコネクション選択 13

メッセージ送信中の回線障害 (送信メッセージの迂回判定 UOC で迂回しない場合) 172  
 メッセージ送信中の回線障害 (送信メッセージの迂回判定 UOC で迂回する場合) 171  
 メッセージ送信用バッファグループ番号 113  
 メッセージの再送 (COBOL 言語) 42  
 メッセージの再送 (C 言語) 27  
 メッセージの受信 (データ操作言語) 55  
 メッセージの分割と組み立て 15  
 メッセージ廃棄通知イベント 90  
 メッセージ編集用バッファグループ番号 114  
 メッセージ編集用バッファ数 114  
 メモリ出力メッセージ最大格納数 118

## ゆ

---

ユーザアプリケーションプログラムの作成例 62  
 ユーザオウンコーディング 68  
 ユーザオウンコーディング, MCF イベントインタフェース 67  
 ユーザオウンコーディングインタフェース 68  
 ユーザ情報の操作方法 89

## ら

---

ラウンドロビン方式 181

## り

---

理由コード一覧 175

## ろ

---

論理端末 181  
 論理端末定義 118  
 論理端末とアプリケーションの型との関係 14  
 論理端末の状態表示 146  
 論理端末の閉塞 12, 141  
 論理端末の閉塞解除 13, 136  
 論理端末名称 118  
 論理メッセージ 15

# ソフトウェアマニュアルのサービス ご案内

## 1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しています。

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

ホームページのメニューは次のとおりです。

マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル	日立ソフトウェアマニュアルと製品群別CD-ROMマニュアルの仕様について記載しています。
マニュアルのご購入	マニュアルご購入時のお申し込み方法を記載しています。
オンラインマニュアル	一部製品のマニュアルをインターネットで公開しています。
サポートサービス	ソフトウェアサポートサービスお客様向けページでのマニュアル公開サービスを記載しています。
ご意見・お問い合わせ	マニュアルに関するご意見、ご要望をお寄せください。

## 2. インターネットでのマニュアル公開

2種類のマニュアル公開サービスを実施しています。

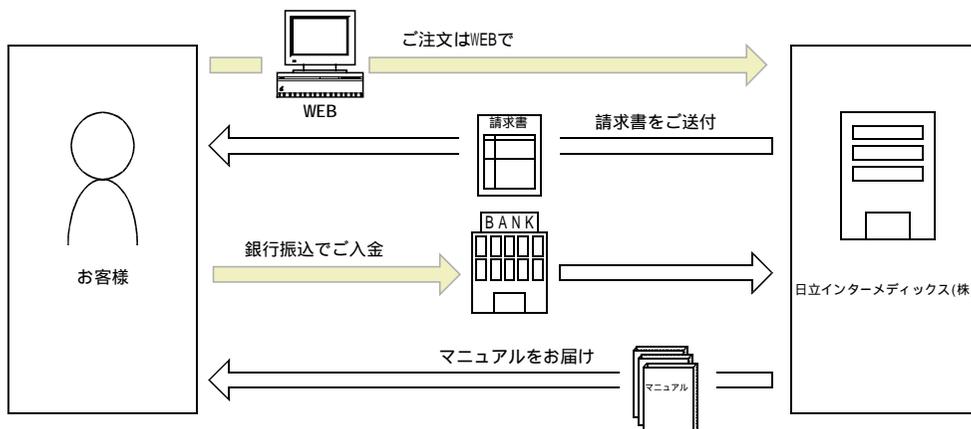
### (1) マニュアル情報ホームページ「オンラインマニュアル」での公開

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

### (2) ソフトウェアサポートサービスお客様向けページでのマニュアル公開

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

## 3. マニュアルのご注文



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

ご注文いただいたマニュアルについて請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。

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