
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/HSC 編

解説・手引・文法・操作書

3000-3-D77

マニュアルの購入方法

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

HITACHI

対象製品

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

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

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

P-F1M64-32316 uCosminexus TP1/NET/HSC 07-00

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

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

輸出時の注意

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

なお、ご不明な場合は、弊社担当営業にお問い合わせください。

商標類

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

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

発行

2006 年 11 月 (第 1 版) 3000-3-D77

著作権

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

はじめに

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

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

対象読者

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

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

マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 概要

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

第 2 章 HSC1 手順の機能

TP1/NET/HSC の HSC1 手順を使用した場合の、接続の確立や論理端末の端末タイプなどの AP 間通信の仕組み、メッセージの送受信、および HSC1 手順の機能について説明しています。

第 3 章 HSC2 手順の機能

TP1/NET/HSC の HSC2 手順 (非同期モードおよび同期モード) を使用した場合の、接続の確立や論理端末の端末タイプなどの AP 間通信の仕組み、メッセージの送受信、および HSC2 手順の機能について説明しています。

第 4 章 メッセージ送受信インタフェース

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

第 5 章 ユーザOWNコーディング、MCF イベントインタフェース

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

はじめに

第 6 章 システム定義

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

第 7 章 運用コマンド

TP1/NET/HSC で使用する運用コマンドについて説明しています。

第 8 章 組み込み方法

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

第 9 章 HSC1 手順の障害対策

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

第 10 章 HSC2 手順（非同期モード）の障害対策

TP1/NET/HSC の HSC2 手順（非同期モード）運用中に発生する障害と、TP1/NET/HSC の対応処理、およびメッセージの処理について説明しています。

第 11 章 HSC2 手順（同期モード）の障害対策

TP1/NET/HSC の HSC2 手順（同期モード）運用中に発生する障害と、TP1/NET/HSC の対応処理、およびメッセージの処理について説明しています。

付録 A メッセージ送受信の処理の流れ

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

付録 B 障害発生時の処理の流れ

メッセージを送受信するときに発生した障害処理の流れについて説明しています。

付録 C 理由コード一覧

障害通知イベントが発生した場合の理由コードについて説明しています。

付録 D 関数使用の制限（HSC2 手順（同期モード））

HSC2 手順（同期モード）の場合、ある関数のテキスト形態とリターン値によって、その次に呼び出せる関数の制限について説明しています。

付録 E 論理端末名称決定 UOC のコーディング例

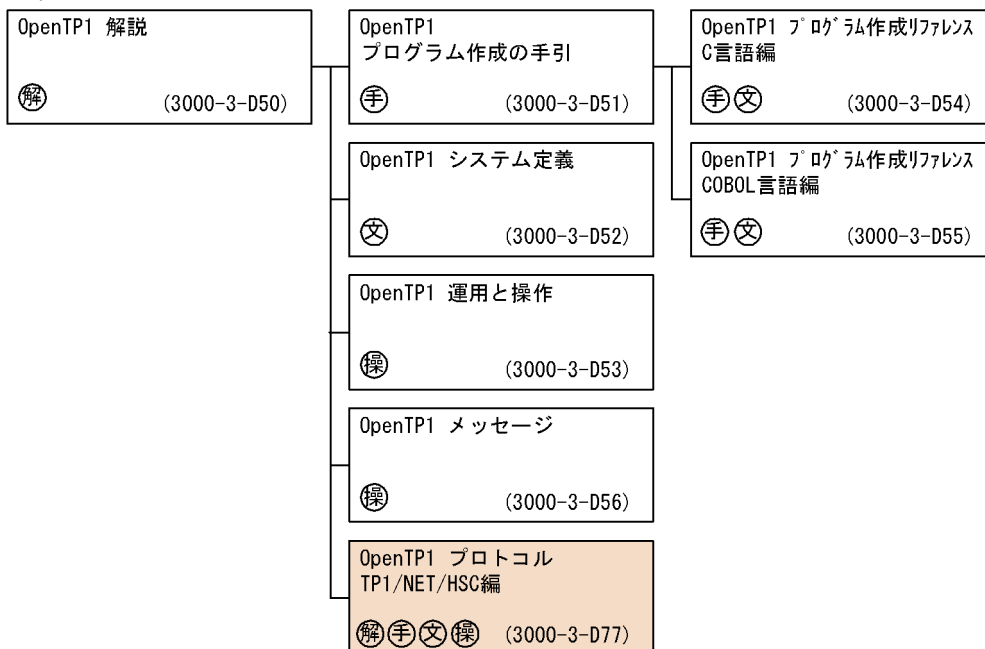
TP1/NET/HSC の論理端末名称決定 UOC のコーディング例について説明しています。

付録 F 用語解説

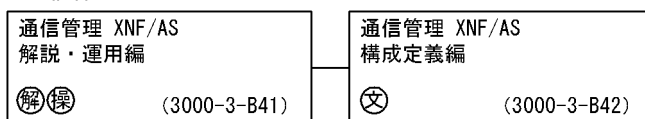
TP1/NET/HSC で使用する用語について説明しています。

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)

- ▼ : HSC1手順を使用する場合に必ず読む項目
- ▽ : HSC2手順を使用する場合に必ず読む項目

図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。

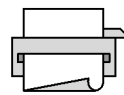
●ワークステーション、
端末



●論理端末



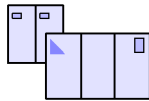
●端末プリンタ



●入出力の動作



●ホストコンピュータ



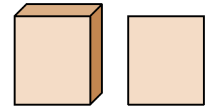
●プログラムの流れ



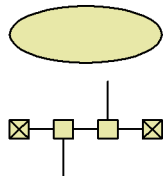
●ファイル



●プログラム



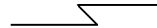
●ネットワーク



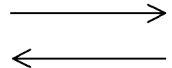
●データの流れ



●通信回線



●制御の流れ



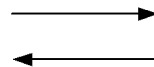
●障害



●論理回線



●その他の流れ



文法の記号

このマニュアルで使用する各種の記号を説明します。

(1) 文法記述記号

文法の記述形式について説明する記号です。

文法記述記号	意味
[]	この記号で囲まれている項目は省略できることを示します。 (例) [-s MCF 通信プロセス識別子] -s オプションとそのオペランドを指定するか、何も指定しないことを示します。
 (ストローク)	この記号で区切られた項目は選択できることを示します。 (例) -t reply request -t オプションに reply または request を指定できることを示します。 ただし、C 言語のインタフェースの説明でこの記号を使用した場合は、C 言語の文法規則に従います。

文法記述記号	意味
{ }	この記号で囲まれている複数の項目のうちから一つを選択できることを示します。 (例){DCMCFESI DCMCFEMI} DCMCFESI と DCMCFEMI のうち、どちらかを指定できることを示します。
(下線)	この記号で示す項目は、オペランド、オプションまたはコマンド引数を省略した場合の省略時解釈値を示します。 (例) -i auto <u>manual</u> -i オプションを省略した場合、manual を省略時解釈値とすることを示します。 ただし、データ操作言語の説明の場合、この下線記号で示す予約語は、必要語なので省略できないことを示します。 下線がない予約語は、補助語なので書いても書かなくてもかまいません。
...	この記号で示す直前の一つの項目を繰り返し指定できることを示します。 ただし、項目が括弧で囲まれている場合、括弧全体が一つの項目となります。
(白三角)	空白を示します。 (例) コネクション ID1 コネクション ID2 コネクション ID1 とコネクション ID2 の間に、空白を 1 個入力することを示します。

(2) 属性表示記号

ユーザ指定値の範囲などを説明する記号です。

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

(3) 構文要素記号

ユーザ指定値の内容を説明する記号です。

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

このマニュアルでの表記

(1) 製品名

このマニュアルで使用する製品名称の略称を次に示します。

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

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

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

略語一覧

このマニュアルで使用する英略語の一覧を次に示します。

英略語	英字での表記
AP	<u>A</u> pplication <u>P</u> rogram
DLE	<u>D</u> ata <u>L</u> ink <u>E</u> scape
EBCDIK	<u>E</u> xtended <u>B</u> inary <u>C</u> oded <u>D</u> ecimal <u>I</u> nterchange <u>K</u> ana <u>C</u> ode
ETB	<u>E</u> nd of <u>T</u> ransmission <u>B</u> lock
ETX	<u>E</u> nd of <u>T</u> ext
HSC	<u>H</u> itachi <u>S</u> tandard <u>S</u> ynchronous <u>C</u> ommunication
ISDN	<u>I</u> ntegrated <u>S</u> ervices <u>D</u> igital <u>N</u> etwork
JIS8	<u>J</u> apanese <u>I</u> ndustrial <u>S</u> tandard <u>C</u> ode 8
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
OS	<u>O</u> perating <u>S</u> ystem
SOH	<u>S</u> tart of <u>H</u> eading

英略語	英字での表記
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram
STX	<u>S</u> tart of <u>T</u> ext
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

常用漢字以外の漢字の使用について

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

個所（かしょ） 閉塞（へいそく）

KB（キロバイト）などの単位表記について

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

謝辞

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

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

COBOL は産業界の言語であり、特定の団体や組織の所有物ではない。

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

次に示す著作権表示付資料の著作者及び著作権者

FLOW-MATIC（Sperry Rand Corporation の商標）、
Programming for the Univac（R）I and II，Data Automation Systems，

Sperry Rand Corporation 著作権表示 1958 年，1959 年；
IBM Commercial Translator Form No.F 28-8013，IBM 著作権表示 1959 年；
FACT，DSI 27A5260-2760，Minneapolis-Honeywell，著作権表示 1960 年

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

目次

1	概要	1
1.1	AP 間通信の概要	2
1.2	AP 間通信の形態	3
1.2.1	HSC1 手順の通信形態	3
1.2.2	HSC2 手順（非同期モード）の通信形態	4
1.2.3	HSC2 手順（同期モード）の通信形態	4
1.2.4	適用範囲	7
1.3	ソフトウェアの構成	8
1.3.1	前提プログラム	8
1.3.2	ソフトウェア構成の例	8
2	HSC1 手順の機能	11
2.1	HSC1 手順の AP 間通信の仕組み	12
2.1.1	HSC1 手順の送受信の方法	12
2.1.2	コネクションと論理端末の関係	14
2.1.3	論理端末とアプリケーションの型の関係	15
2.1.4	論理端末の閉塞と閉塞解除	15
2.1.5	メッセージの分割と組み立て	17
2.2	HSC1 手順の AP 間通信メッセージの送受信	22
2.2.1	メッセージの送信	22
2.2.2	メッセージの受信	23
2.3	HSC1 手順の機能	24
3	HSC2 手順の機能	25
3.1	HSC2 手順の AP 間通信の仕組み	26
3.1.1	HSC2 手順（非同期モード）の送受信の方法	26
3.1.2	HSC2 手順（同期モード）の送受信の方法	30
3.1.3	コネクションと論理端末の関係	38
3.1.4	論理端末とアプリケーションの型の関係	41
3.1.5	論理端末の閉塞と閉塞解除	43
3.1.6	メッセージの分割と組み立て	44
3.2	HSC2 手順の AP 間通信メッセージの送受信	50

3.2.1	メッセージの送信	50
3.2.2	メッセージの受信	53
3.3	HSC2 手順の機能	57
3.4	回線リスト定義ユティリティ (非同期モード)	58
3.5	相手ターミナル ID リスト定義ユティリティ (同期モード)	59
3.6	EOT 交換	60
3.6.1	非同期モードの EOT 交換	60
3.6.2	同期モードの EOT 交換	63

4

メッセージ送受信インタフェース		67
メッセージ送受信インタフェースの一覧		69
dc_mcf_receive	- メッセージまたは接続情報通知の受信 (C 言語)	71
dc_mcf_recvsync	- 同期型のメッセージの受信 (C 言語)	75
dc_mcf_resend	- メッセージの再送 (C 言語)	79
dc_mcf_send	- 一方送信メッセージまたは回線接続要求の送信 (C 言語)	83
dc_mcf_sendsync	- 同期型のメッセージの送信 (C 言語)	88
CBLDCMCF('RECEIVE')	- メッセージまたは接続情報通知の受信 (COBOL 言語)	92
CBLDCMCF('RECVSYNC')	- 同期型のメッセージの受信 (COBOL 言語)	98
CBLDCMCF('RESEND')	- メッセージの再送 (COBOL 言語)	104
CBLDCMCF('SEND')	- 一方送信メッセージまたは回線接続要求の送信 (COBOL 言語)	110
CBLDCMCF('SENDSYNC')	- 同期型のメッセージの送信 (COBOL 言語)	116
RECEIVE	- メッセージの受信 (データ操作言語)	122
SEND	- メッセージの送信 (データ操作言語)	126
ユーザアプリケーションプログラム作成例		132

5

ユーザOWNコーディング, MCF イベントインタフェース		153
5.1	ユーザOWNコーディングインタフェース	154
5.1.1	論理端末名称の決定	154
5.1.2	論理端末名称決定 UOC インタフェース	155
5.1.3	入力メッセージの編集とアプリケーション名の決定	160
5.1.4	入力メッセージ編集 UOC インタフェース	163
5.1.5	出力メッセージの編集	168
5.1.6	出力メッセージ編集 UOC インタフェース	169
5.1.7	送信メッセージの通番編集	172

5.1.8	送信メッセージの通番編集 UOC インタフェース	173
5.1.9	UOC 作成上の注意事項	174
5.2	MCF イベントインタフェース	176
5.2.1	MCF イベントの種類	176
5.2.2	MCF イベント通知時のセグメント構成	177
5.2.3	MCF イベント情報の形式 (C 言語)	178
5.2.4	MCF イベント情報の形式 (COBOL 言語)	185

6

システム定義	195
TP1/NET/HSC の定義の概要	197
TP1/NET/HSC 固有のシステム定義の種類 - HSC1 手順	199
mcftalccn (コネクション定義の開始) - HSC1 手順	202
mcftalcle (論理端末定義) - HSC1 手順	207
mcftalced (コネクション定義の終了) - HSC1 手順	210
TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (非同期モード)	211
mcftalccn (コネクション定義の開始) - HSC2 手順 (非同期モード)	215
mcftalcle (論理端末定義) - HSC2 手順 (非同期モード)	223
mcftalced (コネクション定義の終了) - HSC2 手順 (非同期モード)	227
mcftlfn (回線リスト出力ファイル定義) - HSC2 手順 (非同期モード)	228
mcftllst (回線リスト定義) - HSC2 手順 (非同期モード)	229
mcftleof (回線リスト出力ファイル定義の終了) - HSC2 手順 (非同期モード)	230
TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (同期モード)	231
mcftalccn (コネクション定義の開始) - HSC2 手順 (同期モード)	236
mcftalcle (論理端末定義) - HSC2 手順 (同期モード)	243
mcftalced (コネクション定義の終了) - HSC2 手順 (同期モード)	245
mcftstf (出力ファイル定義) - HSC2 手順 (同期モード)	246
mcftstlst (発信用相手ターミナル ID リスト定義) - HSC2 手順 (同期モード)	247
mcfttrlst (着信用相手ターミナル ID リスト定義) - HSC2 手順 (同期モード)	248
mcfteof (出力ファイル定義の終了) - HSC2 手順 (同期モード)	249
システムサービス情報定義	250
システムサービス共通情報定義	251
MCF 定義オブジェクトの生成	253
回線リスト定義 - HSC2 手順 (非同期モード)	254
相手ターミナル ID リスト定義 - HSC2 手順 (同期モード)	255
自システムの通信管理プログラムと関連づける内容	257

定義例	260
7 運用コマンド	269
TP1/NET/HSC の運用コマンド	270
mcftactcn (コネクションの確立)	271
mcftactle (論理端末の閉塞解除)	273
mcftdctcn (コネクションの解放)	275
mcftdctle (論理端末の閉塞)	278
mcftlscn (コネクションの状態表示)	281
mcftlslc (論理端末の状態表示)	284
8 組み込み方法	287
8.1 TP1/NET/HSC の組み込みの流れ	288
8.2 MCF メイン関数の作成	289
8.3 定義オブジェクトファイルの生成	293
9 HSC1 手順の障害対策	297
9.1 障害の種類と対応処理	298
9.2 コネクション障害	305
9.2.1 メッセージ送信時のコネクション障害	305
9.2.2 メッセージ受信時のコネクション障害	306
9.3 ユーザアプリケーションプログラム異常終了	308
9.3.1 セグメント受信前の UAP 異常終了	308
9.3.2 セグメント受信後の UAP 異常終了	308
9.4 ユーザアプリケーションプログラム閉塞	310
9.5 入力キュー障害	311
9.6 出力キュー障害	312
10 HSC2 手順 (非同期モード) の障害対策	313
10.1 障害の種類と対応処理	314
10.2 コネクション障害	326
10.2.1 メッセージ送信時のコネクション障害	326
10.2.2 メッセージ受信時のコネクション障害	327

10.3 ユーザアプリケーションプログラム異常終了	329
10.3.1 セグメント受信前の UAP 異常終了	329
10.3.2 セグメント受信後の UAP 異常終了	329
10.4 ユーザアプリケーションプログラム閉塞	331
10.5 入力キュー障害	332
10.6 出力キュー障害	333

11 HSC2 手順（同期モード）の障害対策	335
11.1 障害の種類と対応処理	336
11.2 コネクション障害	348
11.3 ユーザアプリケーションプログラム異常終了	349
11.3.1 回線接続要求受信前の UAP 異常終了	349
11.3.2 回線接続要求受信後の UAP 異常終了	350
11.4 ユーザアプリケーションプログラム閉塞	351
11.5 HSC 手順障害	352

付録	353
付録 A メッセージ送受信の処理の流れ	354
付録 A.1 HSC1 手順のメッセージの送受信	354
付録 A.2 HSC2 手順（非同期モード）のメッセージの送受信	360
付録 A.3 HSC2 手順（同期モード）のメッセージの送受信	367
付録 B 障害発生時の処理の流れ	377
付録 B.1 HSC1 手順の障害	377
付録 B.2 HSC2 手順（非同期モード）の障害	382
付録 B.3 HSC2 手順（同期モード）の障害	388
付録 C 理由コード一覧	393
付録 C.1 ERREVT2 の理由コード一覧	393
付録 C.2 HSC1 手順の理由コード一覧	393
付録 C.3 HSC2 手順（非同期モード）の理由コード一覧	395
付録 C.4 HSC2 手順（同期モード）の理由コード一覧	396
付録 D 関数使用の制限（HSC2 手順（同期モード））	399
付録 E 論理端末名称決定 UOC のコーディング例	401
付録 F 用語解説	402

目次

図 1-1	TP1/NET/HSC を使用したネットワーク構成例	2
図 1-2	分岐送信形態の AP 間通信の例 (HSC1 手順の場合)	3
図 1-3	一方送信形態の AP 間通信の例 (HSC1 手順の場合)	3
図 1-4	HSC2 手順 (同期モード) を使用した通信形態	4
図 1-5	発信の場合の通信形態の組み合わせ	6
図 1-6	着信の場合の通信形態の組み合わせ	7
図 1-7	TP1/NET/HSC を組み込んだソフトウェア構成	9
図 2-1	HSC1 手順の送受信の方法	13
図 2-2	処理層とデータの関係 (HSC1 手順)	18
図 2-3	テキスト形態の種別 (HSC1 手順)	19
図 2-4	HSC1 手順のメッセージ送信の流れ	22
図 2-5	HSC1 手順のメッセージ受信の流れ	23
図 3-1	HSC2 手順 (非同期モード) のコネクションの確立	27
図 3-2	HSC2 手順 (非同期モード) のコネクションの解放	30
図 3-3	HSC2 手順 (同期モード) のコネクションの確立	32
図 3-4	発信の場合の回線の接続 (同期モード)	34
図 3-5	着信の場合の回線の接続 (同期モード)	35
図 3-6	HSC2 手順 (同期モード) のコネクションの解放	37
図 3-7	コネクションと論理端末の関係 (HSC2 手順 (非同期モード))	39
図 3-8	コネクションと論理端末の関係 (HSC2 手順 (同期モード))	41
図 3-9	処理層とデータの関係 (HSC2 手順)	45
図 3-10	テキスト形態の種別 (HSC2 手順)	47
図 3-11	HSC2 手順 (非同期モード) のメッセージ送信の流れ	51
図 3-12	HSC2 手順 (同期モード) のメッセージ送信の流れ	52
図 3-13	HSC2 手順 (非同期モード) のメッセージ受信の流れ	54
図 3-14	HSC2 手順 (同期モード) のメッセージ受信の流れ	55
図 3-15	非同期モードのメッセージ送信後の EOT 交換	61
図 3-16	非同期モードのメッセージ受信後の EOT 交換	62
図 3-17	同期モードのメッセージ送信後の EOT 交換	64
図 3-18	同期モードのメッセージ受信後の EOT 交換	65
図 4-1	UAP 作成例の処理の流れ	132
図 4-2	発信用 SPP, 発信用 MHP の例	137
図 4-3	着信用 MHP の例	137

図 5-1	UOC インタフェース用のパラメタとバッファの関係 (論理端末名称決定 UOC の場合)	159
図 5-2	アプリケーション名の決定の処理	162
図 5-3	UOC インタフェース用のパラメタとバッファの関係	167
図 5-4	MCF イベント通知時のセグメント構成	178
図 6-1	TP1/NET/HDLC のプロトコル固有定義コマンドの指定順序 (HSC1 手順)	201
図 6-2	TP1/NET/HDLC のプロトコル固有定義コマンドの指定順序 (HSC2 手順 (非同期モード))	214
図 6-3	TP1/NET/HDLC のプロトコル固有定義コマンドの指定順序 (HSC2 手順 (同期モード))	235
図 6-4	TP1/NET/HSC のシステム構成例 (HSC1 手順)	257
図 6-5	TP1/NET/HSC のシステム構成例 (HSC 2 手順 (非同期モード))	258
図 6-6	TP1/NET/HSC のシステム構成例 (HSC 2 手順 (同期モード))	259
図 8-1	MCF メイン関数のコーディング概要 (ANSI C, C++ の場合)	290
図 8-2	MCF メイン関数のコーディング概要 (K&R 版 C の場合)	290
図 8-3	MCF メイン関数のディレクトリへの組み込み方法	292
図 8-4	定義オブジェクトファイルの作成方法の概要	294
図 9-1	メッセージ送信前のコネクション障害 (HSC1 手順)	305
図 9-2	メッセージ消去前のコネクション障害 (HSC1 手順)	306
図 9-3	メッセージ書き込み前のコネクション障害 (HSC1 手順)	307
図 9-4	メッセージ書き込み後のコネクション障害 (HSC1 手順)	307
図 9-5	セグメント受信前の UAP 異常終了 (HSC1 手順)	308
図 9-6	セグメント受信後の UAP 異常終了 (HSC1 手順)	309
図 9-7	UAP 閉塞による障害 (HSC1 手順)	310
図 9-8	入力キュー障害 (HSC1 手順)	311
図 9-9	出力キュー障害 (HSC1 手順)	312
図 10-1	メッセージ送信前のコネクション障害 (HSC2 手順 (非同期モード))	326
図 10-2	メッセージ消去前のコネクション障害 (HSC2 手順 (非同期モード))	327
図 10-3	メッセージ書き込み前のコネクション障害 (HSC2 手順 (非同期モード))	328
図 10-4	メッセージ書き込み後のコネクション障害 (HSC2 手順 (非同期モード))	328
図 10-5	セグメント受信前の UAP 異常終了 (HSC2 手順 (非同期モード))	329
図 10-6	セグメント受信後の UAP 異常終了 (HSC2 手順 (非同期モード))	330
図 10-7	UAP 閉塞による障害 (HSC2 手順 (非同期モード))	331
図 10-8	入力キュー障害 (HSC2 手順 (非同期モード))	332
図 10-9	出力キュー障害 (HSC2 手順 (非同期モード))	333
図 11-1	メッセージ送受信時のコネクション障害 (HSC2 手順 (同期モード))	348

図 11-2	回線接続要求受信前の UAP 異常終了 (HSC2 手順 (同期モード))	349
図 11-3	回線接続要求受信後の UAP 異常終了 (HSC2 手順 (同期モード))	350
図 11-4	UAP 閉塞による障害 (HSC2 手順 (同期モード))	351
図 11-5	メッセージ送信時の HSC 手順障害 (HSC2 手順 (同期モード))	352
図 A-1	HSC1 手順のメッセージ送信の処理の流れ (単一セグメントの場合)	354
図 A-2	HSC1 手順のメッセージ送信の処理の流れ (複数セグメントの場合)	355
図 A-3	HSC1 手順のメッセージ送信の処理の流れ (列信の場合)	356
図 A-4	HSC1 手順のメッセージ受信の処理の流れ (単一セグメントの場合)	357
図 A-5	HSC1 手順のメッセージ受信の処理の流れ (複数セグメントの場合)	358
図 A-6	HSC1 手順のメッセージ受信の処理の流れ (列信の場合)	359
図 A-7	HSC2 手順 (非同期モード) のメッセージ送信の処理の流れ (単一セグメントの場合)	360
図 A-8	HSC2 手順 (非同期モード) のメッセージ送信の処理の流れ (複数セグメントの場合)	361
図 A-9	HSC2 手順 (非同期モード) のメッセージ送信の処理の流れ (列信の場合)	362
図 A-10	HSC2 手順 (非同期モード) のメッセージ受信の処理の流れ (単一セグメントの場合)	363
図 A-11	HSC2 手順 (非同期モード) のメッセージ受信の処理の流れ (複数セグメントの場合)	364
図 A-12	HSC2 手順 (非同期モード) のメッセージ受信の処理の流れ (列信の場合)	365
図 A-13	HSC2 手順 (非同期モード) の回線切断の処理の流れ (回線維持タイムアウトの場合)	366
図 A-14	送受信反転制御	367
図 A-15	HSC2 手順 (同期モード) のメッセージ送信の処理の流れ (単一セグメントで通常テキスト送信後, EOT を送信する場合)	368
図 A-16	HSC2 手順 (同期モード) のメッセージ送信の処理の流れ (単一セグメントで通常テキスト送信する場合)	369
図 A-17	HSC2 手順 (同期モード) のメッセージ送信の処理の流れ (複数セグメントの場合)	370
図 A-18	HSC2 手順 (同期モード) のメッセージ送信の処理の流れ (列信の場合)	371
図 A-19	HSC2 手順 (同期モード) のメッセージ受信の処理の流れ (単一セグメントの場合)	372
図 A-20	HSC2 手順 (同期モード) のメッセージ受信の処理の流れ (複数セグメントの場合)	373
図 A-21	HSC2 手順 (同期モード) のメッセージ受信の処理の流れ (列信の場合)	374
図 A-22	HSC2 手順 (同期モード) の回線切断の処理の流れ (回線切断要求 (DLE・EOT) 送信の場合)	375
図 A-23	HSC2 手順 (同期モード) の回線切断の処理の流れ (MHP の終了の場合)	375

図 A-24 HSC2 手順（同期モード）の回線切断の処理の流れ（回線切断の受信の場合）	376
図 B-1 コネクション切断	377
図 B-2 メッセージ送信時の障害	378
図 B-3 出力キュー読み込み障害	379
図 B-4 HSC1 手順の入力メッセージ編集 UOC 障害	380
図 B-5 HSC1 手順のコンテンション	381
図 B-6 HSC1 手順の逆中断（RVI）受信（mcftalccn コマンドの -u オプションで accept を指定した場合）	382
図 B-7 HSC2 手順（非同期モード）のメッセージ送信時の回復不可能障害	383
図 B-8 HSC2 手順（非同期モード）のメッセージ送信時の HSC 手順エラー	384
図 B-9 HSC2 手順（非同期モード）の入力メッセージ編集 UOC 障害	385
図 B-10 HSC2 手順（非同期モード）のコンテンション	386
図 B-11 HSC2 手順（非同期モード）の逆中断（RVI）受信	387
図 B-12 HSC2 手順（同期モード）のメッセージ送信時の回復不可能障害	388
図 B-13 HSC2 手順（同期モード）のメッセージ送信時の HSC 手順エラー	389
図 B-14 HSC2 手順（同期モード）の入力メッセージ編集 UOC 障害	390
図 B-15 HSC2 手順（同期モード）のコンテンション	391
図 B-16 HSC2 手順（同期モード）の逆中断（RVI）受信（mcftalccn コマンドの -u オプションで accept を指定した場合）	392

表目次

表 2-1	論理端末の端末タイプ，メッセージの種類，アプリケーションの型，UAP インタフェース，および通信形態の関係（HSC1 手順）	15
表 2-2	UAP および通信管理でのデータの扱い（HSC1 手順）	20
表 2-3	送信するテキスト形態の数値表現と内容（HSC1 手順）	21
表 2-4	受信するテキスト形態の数値表現と内容（HSC1 手順）	21
表 2-5	HSC1 手順の機能と TP1/NET/HSC の扱い	24
表 3-1	論理端末の端末タイプ，メッセージの種類，アプリケーションの型，UAP インタフェース，および通信形態の関係（HSC2 手順）	43
表 3-2	UAP および通信管理でのデータの扱い（HSC2 手順）	47
表 3-3	送信するテキスト形態の数値表現と内容（HSC2 手順）	49
表 3-4	受信するテキスト形態の数値表現と内容（HSC2 手順）	49
表 3-5	HSC2 手順の機能と TP1/NET/HSC の扱い	57
表 4-1	メッセージ送受信のライブラリ関数（C 言語）	69
表 4-2	メッセージ送受信の通信文（COBOL 言語）	70
表 4-3	メッセージ送受信の通信文（データ操作言語）	70
表 5-1	TP1/NET/HSC が通知する MCF イベントの種類	176
表 5-2	COBOL 言語の MCF イベント情報の内容（ERREVT1）	185
表 5-3	COBOL 言語の MCF イベント情報の内容（ERREVT2）	186
表 5-4	COBOL 言語の MCF イベント情報の内容（ERREVT3）	188
表 5-5	COBOL 言語の MCF イベント情報の内容（ERREVTA）	189
表 5-6	COBOL 言語の MCF イベント情報の内容（SCMPEVT）	190
表 5-7	COBOL 言語の MCF イベント情報の内容（SERREVT）	191
表 5-8	COBOL 言語の MCF イベント情報の内容（CERREVT）	193
表 5-9	COBOL 言語の MCF イベント情報の内容（COPNEVT，CCLSEVT）	194
表 6-1	MCF で使用する定義ファイル	197
表 6-2	TP1/NET/HSC 固有の定義の一覧（HSC1 手順）	199
表 6-3	TP1/NET/HSC 固有の定義の一覧（HSC2 手順（非同期モード））	211
表 6-4	TP1/NET/HSC 固有の定義の一覧（HSC2 手順（同期モード））	231
表 7-1	TP1/NET/HSC の運用コマンド	270
表 9-1	コネクション障害発生時の処理（HSC1 手順）	298
表 9-2	受信スケジュール関係の障害発生時の処理（HSC1 手順）	300
表 9-3	送信スケジュール関係の障害発生時の処理（HSC1 手順）	302
表 9-4	UAP 障害発生時の処理（HSC1 手順）	304

表 9-5	MCF の障害発生時の処理 (HSC1 手順)	304
表 10-1	通信障害発生時の処理 (HSC2 手順 (非同期モード))	314
表 10-2	HSC2 手順 (非同期モード) プロトコル障害発生時の処理	317
表 10-3	受信スケジュール関係の障害発生時の処理 (HSC2 手順 (非同期モード))	321
表 10-4	送信スケジュール関係の障害発生時の処理 (HSC2 手順 (非同期モード))	322
表 10-5	UAP 障害発生時の処理 (HSC2 手順 (非同期モード))	324
表 10-6	MCF の障害発生時の処理 (HSC2 手順 (非同期モード))	325
表 11-1	通信障害発生時の処理 (HSC2 手順 (同期モード))	336
表 11-2	HSC2 手順 (同期モード) プロトコル障害発生時の処理	339
表 11-3	受信スケジュール関係の障害発生時の処理 (HSC2 手順 (同期モード))	343
表 11-4	送信スケジュール関係の障害発生時の処理 (HSC2 手順 (同期モード))	344
表 11-5	UAP 障害発生時の処理 (HSC2 手順 (同期モード))	346
表 11-6	PRC 障害, sendsync 関数・recvsync 関数のリターン障害時の処理 (HSC2 手順 (同期モード))	347
表 11-7	MCF の障害発生時の処理 (HSC2 手順 (同期モード))	347
表 C-1	ERREVT2 の理由コード一覧	393
表 C-2	HSC1 手順のコネクション障害の理由コード一覧	394
表 C-3	HSC1 手順の論理端末障害の理由コード一覧	394
表 C-4	HSC2 手順 (非同期モード) のコネクション障害の理由コード一覧	395
表 C-5	HSC2 手順 (非同期モード) の論理端末障害の理由コード一覧	395
表 C-6	HSC2 手順 (同期モード) のコネクション障害の理由コード一覧	397
表 C-7	HSC2 手順 (同期モード) の論理端末障害の理由コード一覧	397
表 D-1	関数使用の制限	399

1

概要

TP1/NET/HSC は、OpenTP1 システムで HSC1 手順、または HSC2 手順を使用して通信をする場合の支援プログラムです。この章では、TP1/NET/HSC の概要について説明します。

1.1 AP 間通信の概要

1.2 AP 間通信の形態

1.3 ソフトウェアの構成

1.1 AP 間通信の概要

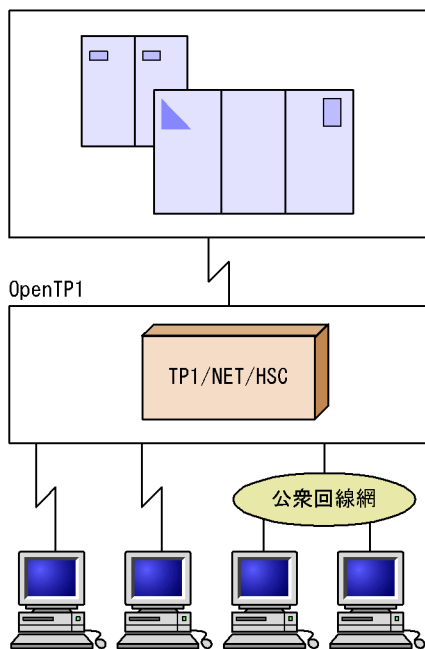
TP1/NET/HSC は、OpenTp1 システムと各種コンピュータシステムを HSC1 手順、または HSC2 手順で接続し、AP 間通信をするプログラムです。HSC1 手順は、相互起動方式の専用回線に適用できます。HSC2 手順は公衆回線に適用できます。また、HSC2 手順には、非同期モードと同期モードがあります。このマニュアルでは、手順またはモードによって機能に違いがある場合、手順ごとまたはモードごとに分けて説明しています。

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

TP1/NET/HSC 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、自システムからトランザクションを発生させて相手システムへ送ったりできます。

TP1/NET/HSC を使用したネットワーク構成例を次の図に示します。

図 1-1 TP1/NET/HSC を使用したネットワーク構成例



1.2 AP 間通信の形態

TP1/NET/HSC を使用した AP 間通信を，HSC1 手順と HSC2 手順に分けて説明します。

1.2.1 HSC1 手順の通信形態

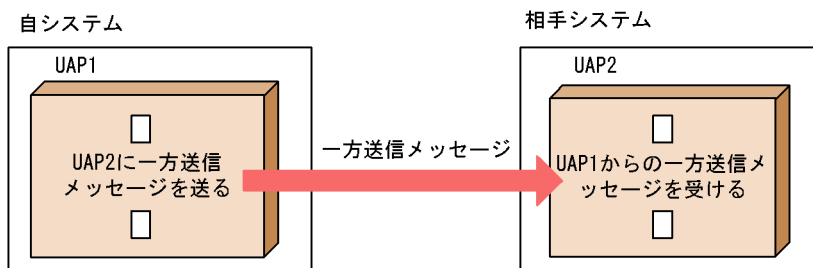
HSC1 手順の通信形態には，分岐送信形態と一方受信形態があります。

(1) 分岐送信形態

自システムから別のシステムに対して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に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（HSC1 手順の場合）

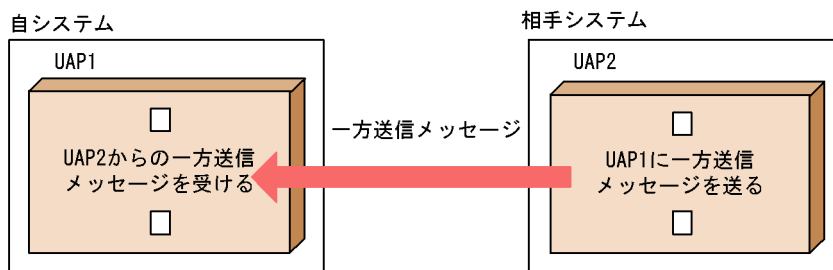


(2) 一方受信形態

相手システムからメッセージを受信する通信形態です。相手システムから受信するメッセージも，分岐送信形態と同様に一方送信メッセージといいます。

一方送信形態の AP 間通信の例を次の図に示します。

図 1-3 一方送信形態の AP 間通信の例（HSC1 手順の場合）



1. 概要

1.2.2 HSC2 手順（非同期モード）の通信形態

HSC2 手順（非同期モード）の通信形態には、分岐送信形態と一方受信形態があります。通信形態は、HSC1 手順の場合と同じです。通信形態については、「1.2.1 HSC1 手順の通信形態」を参照してください。

1.2.3 HSC2 手順（同期モード）の通信形態

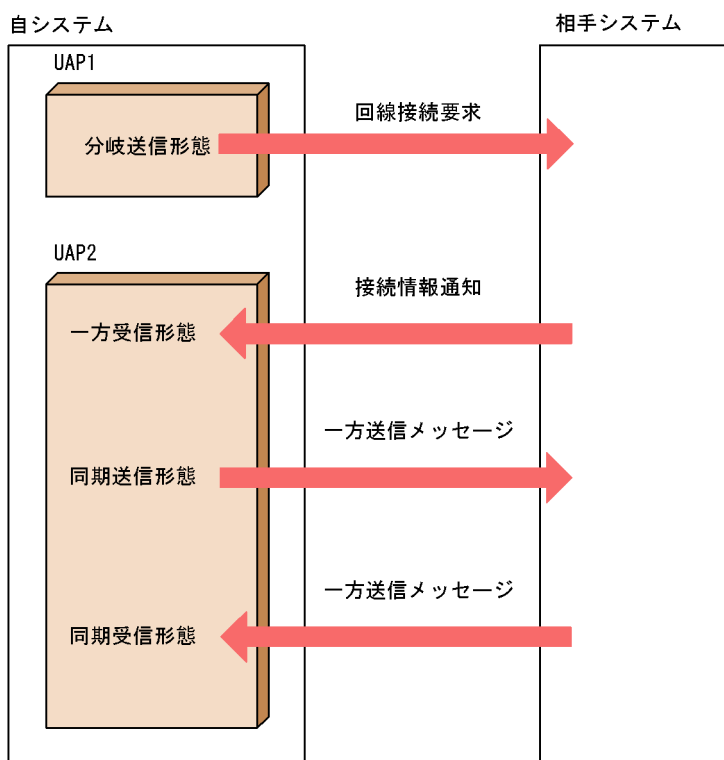
通信形態の種類と、AP 間通信をする場合の通信形態の組み合わせについて説明します。

（1）通信形態の種類

HSC2 手順（同期モード）の通信形態には、分岐送信形態、一方受信形態、同期送信形態、および同期受信形態があります。

HSC2 手順を使用した通信形態を、次の図に示します。

図 1-4 HSC2 手順（同期モード）を使用した通信形態



（a）分岐送信形態

回線接続要求をするときに、自システムの UAP が、相手ターミナル ID の送信要求をするために使用する形態です。

(b) 一方受信形態

相手システムとの回線が接続されたとき、相手ターミナル ID などの接続情報通知を受け取る形態です。

(c) 同期送信形態

自システムから相手システムにメッセージを送信する形態です。相手システムからの送達確認を受けるまで、自システムの UAP は待ち状態になります。

(d) 同期受信形態

相手システムからメッセージを受信する形態です。自システムの UAP は、相手システムからメッセージを受信するまで、待ち状態になります。

(2) 通信形態の組み合わせ

HSC2 手順（同期モード）の場合、通信形態を組み合わせで AP 間通信をします。発信と着信の場合で通信形態の組み合わせの順序は異なります。ここでは、通信形態の組み合わせの例を発信と着信の場合に分けて説明します。AP 間通信の詳細については、「3.

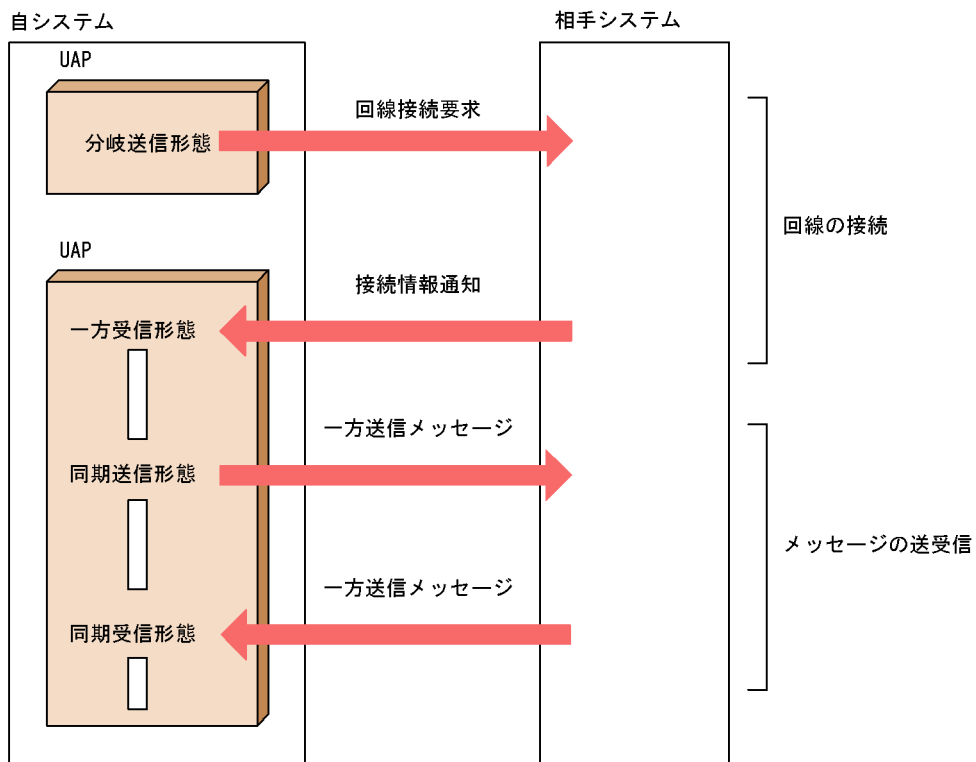
HSC2 手順の機能」を参照してください。

(a) 発信の場合

発信の場合の通信形態の組み合わせの例を、次の図に示します。

1. 概要

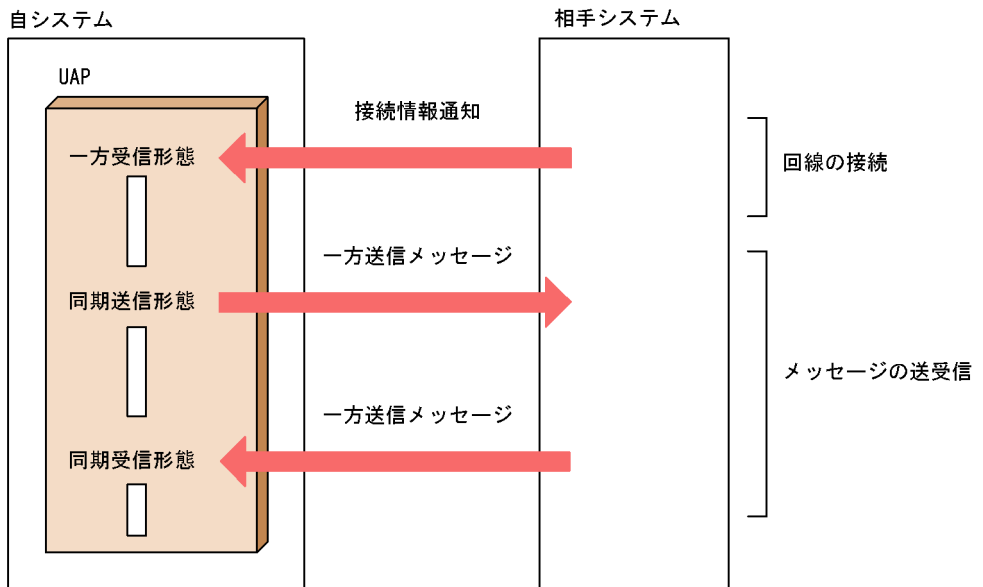
図 1-5 発信の場合の通信形態の組み合わせ



(b) 着信の場合

着信の場合の通信形態の組み合わせの例を、次の図に示します。

図 1-6 着信の場合の通信形態の組み合わせ



1.2.4 適用範囲

(1) HSC1 手順の場合

TP1/NET/HSC の HSC1 手順では、専用回線を使用します。

(2) HSC2 手順の場合

TP1/NET/HSC の HSC2 手順では、次に示す公衆回線を使用できます。

- 電話網 (ハイブリッドホン) (V.25bis)
- 公衆回線網 (X.21)
- ISDN (INS ネット 64-TA 経由) (X.21, V.25bis)
- ISDN (INS ネット 1500-TA 経由) (X.21)

1.3 ソフトウェアの構成

TP1/NET/HSC 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。

ここでは、TP1/NET/HSC を使用した MCF を実現するための前提プログラムと、TP1/NET/HSC 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/HSC 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-F1M14-511A XNF/AS/BASIC

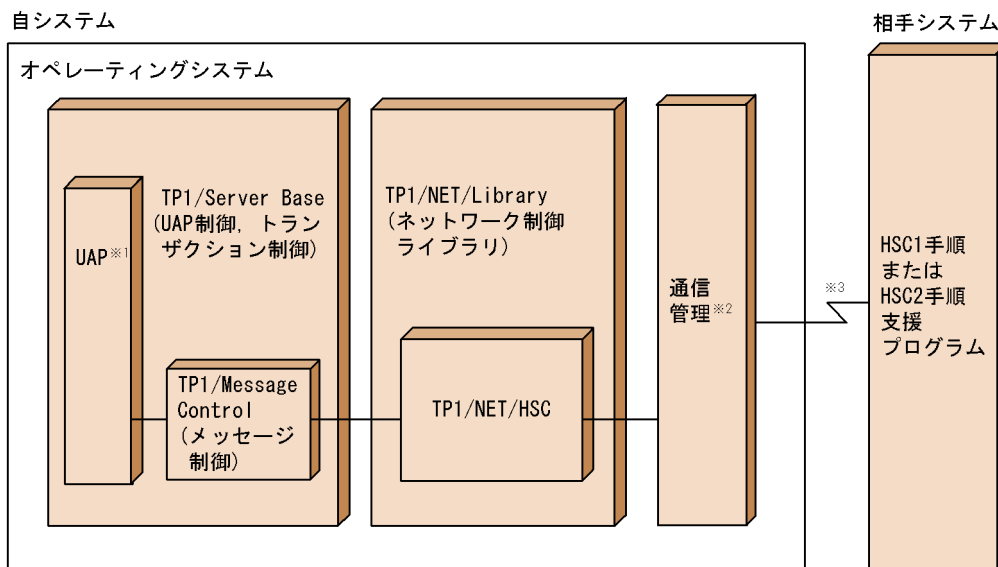
注

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

1.3.2 ソフトウェア構成の例

TP1/NET/HSC 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 1-7 TP1/NET/HSC を組み込んだソフトウェア構成



注 1

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

注 2

TP1/NET/HSC で使用できる通信管理は XNF/AS です。

注 3

HSC1 手順の場合は専用回線で、HSC2 手順の場合は公衆回線網です。

2

HSC1 手順の機能

TP1/NET/HSC は、使用する手順によって、AP 間通信の方法に違いがあります。この章では、HSC1 手順を使用した場合の AP 間通信の方法、メッセージ送受信の流れ、および HSC1 手順の機能サポートについて説明します。

2.1 HSC1 手順の AP 間通信の仕組み

2.2 HSC1 手順の AP 間通信メッセージの送受信

2.3 HSC1 手順の機能

2.1 HSC1 手順の AP 間通信の仕組み

TP1/NET/HSC の HSC1 手順を使用すると、HSC1 手順に対応する相手システムと AP 間通信できます。ここでは、HSC1 手順を使用した場合の AP 間通信の流れ、コネクションや論理端末を使用した AP 間通信の仕組みについて説明します。

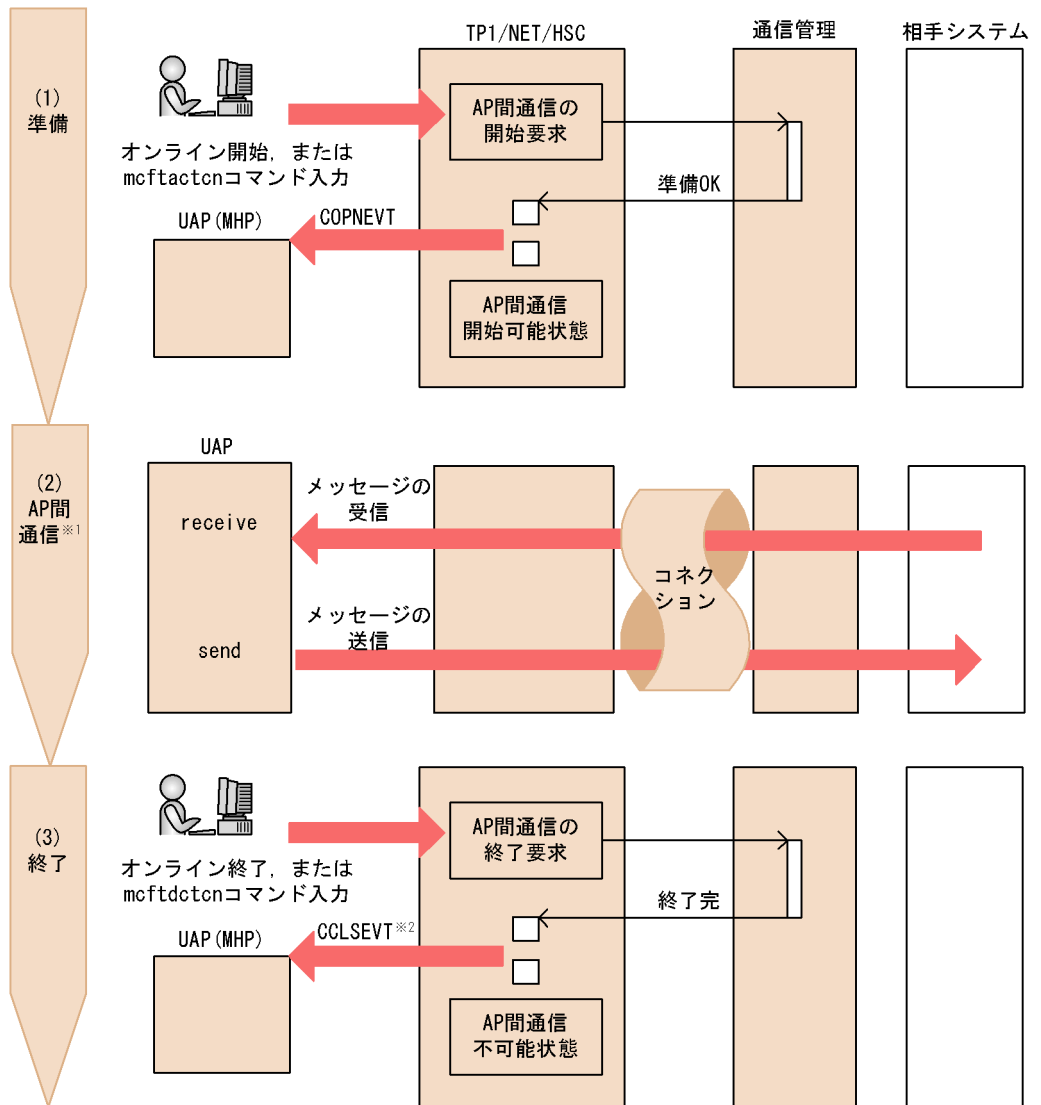
コネクションは通信管理との間の論理的な通信路です。論理端末はメッセージ送受信のための MCF 側の窓口です。

2.1.1 HSC1 手順の送受信の方法

TP1/NET/HSC での、相手システムとの AP 間通信の準備段階から終了までの流れを説明します。

HSC1 手順の送受信の方法を次の図に示します。

図 2-1 HSC1 手順の送受信の方法



注 1

AP間通信の詳細については、「2.2 HSC1 手順の AP 間通信メッセージの送受信」を参照してください。

注 2

オンライン終了時には通知しません。

(1) AP 間通信の準備

TP1/NET/HSC は、通信管理とのやり取りで AP 間通信の準備を整えます。

2. HSC1 手順の機能

OpenTP1 システムのオンライン開始、または運用コマンド (mcftactcn) の入力で、TP1/NET/HSC は、通信管理に対して AP 間通信の準備を要求します。TP1/NET/HSC は、通信管理の準備ができたことを通知されると、状態通知イベント (COPNEVT) を自システムの UAP に通知します。COPNEVT が通知されると、相手システムとの AP 間通信ができる状態になります。

(2) AP 間通信

TP1/NET/HSC は、COPNEVT 通知後から AP 間通信ができます。自システム、または相手システムのメッセージ送信要求を契機に、コネクションを介して AP 間通信をします。

TP1/NET/HSC は、コネクションが常に開いているとみなして相手システムと通信をします。AP 間通信ができる状態のときに、コネクション障害が発生した場合、TP1/NET/HSC は障害通知イベント (CERREVT) を自システムの UAP に通知します。

(3) AP 間通信の終了

OpenTP1 システムのオンライン終了、または運用コマンド (mcftdctcn) の入力で、TP1/NET/HSC は、通信管理に対して AP 間通信の終了を要求します。TP1/NET/HSC は、AP 間通信の終了を通知されると、状態通知イベント (CCLSEVT) を自システムの UAP に通知します。なお、オンライン終了の場合は、CCLSEVT は通知されません。

参考

コネクションの自動回復の抑止

HSC1 手順を使用する場合、コネクションの回復が可能な障害が発生したときに、自動回復を抑止するかどうかを指定できます。コネクションの回復が可能な障害には、タイムアウト、リトライオーバーなどがあります。回復可能障害の詳細については、「9.1(1) コネクション障害」を参照してください。

コネクションの自動回復を抑止するかどうかは、MCF 通信構成定義 (mcftalecn -x) で指定します。自動回復を抑止した場合にコネクションを再確立するには、運用コマンド (mcftactcn) を入力します。

2.1.2 コネクションと論理端末の関係

論理端末 (LE) は、UAP がメッセージの送受信をするための窓口です。TP1/NET/HSC を使用してメッセージの送受信をする場合、ユーザはコネクションと論理端末を意識する必要があります。

システム定義では、コネクション一つに対し、論理端末一つを指定します。また、UAP 作成では、メッセージ受信時には入力元論理端末として、メッセージ送信時には出力先論理端末として、論理端末を意識します。

2.1.3 論理端末とアプリケーションの型の関係

TP1/NET/HSC では、メッセージの送受信をするために、論理端末の端末タイプとアプリケーションの型を定義します。また、アプリケーションの型によって、UAP ができる機能が違います。ここでは、HSC1 手順の場合に TP1/NET/HSC で定義する内容について説明します。

論理端末の端末タイプ

論理端末は、メッセージの送受信をするためのユーザインタフェースで、利用形態による端末タイプがあります。TP1/NET/HSC では、1 コネクション上でのメッセージ送受信を実現するために、any 型（任意型論理端末）を端末タイプとします。

論理端末の端末タイプは、MCF 通信構成定義（mcftalcle -t）で指定します。

アプリケーションの型

アプリケーションは、ユーザが送受信データの中で指定したアプリケーション名をキーとして一つの UAP（MHP）プロセスで実行されます。アプリケーションには、処理形態によって型があります。アプリケーションの型は、MCF アプリケーション定義（mcfaalcap -n）で指定します。TP1/NET/HSC で扱うアプリケーションの型は、非応答型（noans：受信メッセージに対して応答しない MHP の型）だけです。MCF 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。

アプリケーション名

アプリケーション名は、アプリケーションを識別するための名称です。アプリケーション名には、入力メッセージ編集 UOC で決定する方法と、システム定義で決定する方法があります。アプリケーション名の決定については、「5.1.3 入力メッセージの編集とアプリケーション名の決定」を参照してください。また、システム定義で決定する場合は、MCF 通信構成定義（mcftalcle -v）で指定します。

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

表 2-1 論理端末の端末タイプ、メッセージの種類、アプリケーションの型、UAP インタフェース、および通信形態の関係（HSC1 手順）

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

2.1.4 論理端末の閉塞と閉塞解除

論理端末は、コネクションと UAP の間に位置し、メッセージの送受信をする論理的なイ

2. HSC1 手順の機能

インタフェース概念です。

論理端末の状態を次に示します。

閉塞状態

UAP が送信要求したメッセージを出力キューから取り出せない状態です。または、相手システムにメッセージを送信できない状態です。閉塞状態には、次の二つがあります。

- 暗黙閉塞
回線の未接続および障害によってシステムが暗黙的に閉塞しています。
- 明示閉塞
運用コマンド (mcftdctle) で明示的に閉塞しています。

閉塞解除状態

論理端末が持つ機能を使用できる状態です。閉塞解除状態には、次の二つがあります。

- 送信スケジュール可能
論理端末が持つすべての機能を使用できます。
- 送信スケジュール抑止
UAP が送信要求したメッセージのスケジュールを抑止します。TP1/NET/HSC が相手システムにメッセージを送信中のときは、この状態になっています。

(1) 論理端末の閉塞

論理端末の閉塞方法には、次に示す方法があります。

(a) 自動閉塞 (暗黙閉塞)

AP 間通信の終了、またはコネクション障害と同時に、自動的に論理端末を閉塞します。

(b) 手動閉塞 (明示閉塞)

ユーザが、運用コマンド (mcftdctle) を入力して論理端末を明示的に閉塞します。

メッセージの送信中に手動閉塞した場合、送信メッセージは未送信の扱いになります。ただし、閉塞のタイミングによって相手システムへメッセージが正常に送信される場合もあるため、メッセージ通番でメッセージが二重に送信されていないかをチェックしてください。

(c) 障害閉塞 (暗黙閉塞)

論理端末レベルの障害 (障害通知イベントの CERREVT) が発生したとき、障害が発生した論理端末を暗黙的に閉塞します。

(2) 論理端末の閉塞解除

論理端末の閉塞解除方法には、次に示す方法があります。

(a) 自動解除

AP 間通信の準備の完了と同時に、自動的に論理端末の閉塞解除をする方法です。

MCF 通信構成定義 (mcftalcen -l) で auto を指定しておくこと、AP 間通信の準備の完了と同時に論理端末を自動的に閉塞解除します。この時点から、相手システムへメッセージを送信できるようになります。ただし、運用コマンド (mcftactle) で論理端末を閉塞している場合、自動的に閉塞解除はしません。

(b) 手動解除

運用コマンド (mcftactle) を入力して論理端末の閉塞解除をする方法です。

MCF 通信構成定義 (mcftalcen -l) で manual を指定した場合、AP 間通信の準備が完了しても論理端末の閉塞解除はされません。この時点では、UAP が送信要求したメッセージ (send) は出力キューに保留されます。その場合、運用コマンド (mcftactle) を入力して論理端末の閉塞解除をします。

2.1.5 メッセージの分割と組み立て

論理メッセージは、付けられる情報やデータの内容が、処理層によって異なります。また、ユーザが作成した固有のデータは、データの内容によってテキスト形態が異なります。

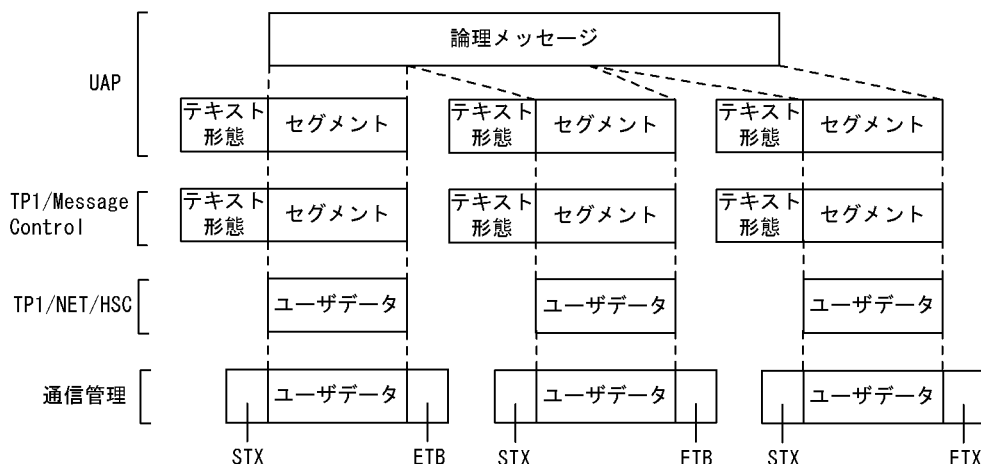
(1) 論理メッセージの扱い

TP1/NET/HSC の論理メッセージは、複数のセグメントに分割できます。メッセージ送受信の関数で処理するセグメントの先頭には、MCF で使用するヘッダ領域があります。このヘッダ領域の長さによって、バッファ形式 1 とバッファ形式 2 があります。通常はバッファ形式 1 を使用します。

OpenTP1 システムの UAP で作成した論理メッセージは、TP1/Message Control, TP1/NET/HSC, 通信管理の各処理層を介して相手システムに送信されます。相手システムからのメッセージの受信の場合は逆の順番になります。論理メッセージは、単独セグメントまたは複数セグメントのデータで構成されます。

処理層とデータの間を、次の図に示します。

図 2-2 処理層とデータの関係 (HSC1 手順)



- (凡例) STX : テキストの最初に付ける伝送制御符号
 ETX : テキストの伝送の終了を示す伝送制御符号
 ETB : データを1ブロックに区切る伝送制御符号

(2) テキスト形態と UAP でのデータの扱い

TP1/NET/HSC の HSC1 手順で扱うテキスト形態と、それぞれについての UAP でのデータの扱いを説明します。各テキスト形態は、16 進数字で表現します。テキスト形態の数値は、次に示すようにメッセージ送信時は送信領域に設定し、メッセージ受信時は受信領域に設定されます。送信領域、受信領域については、「4. メッセージ送受信インタフェース」を参照してください。

- メッセージ送信時
 send 関数を呼び出すときに、送信するメッセージセグメントの先頭に、ユーザがテキスト形態を設定します。
 通常テキストのデータは、EBCDIK コードで作成してください。JIS8 コードの場合、正しく送信されない場合があります。JIS8 コードのデータを送信する場合は、透過モードテキストとして送信してください。
- メッセージ受信時
 receive 関数を呼び出して受信するメッセージのセグメントの先頭に、MCF がテキスト形態を設定します。

また、TP1/NET/HSC を使用する UAP では、アプリケーション起動をされた場合も同様に、「(a) テキスト形態の種類」に示すテキスト形態が設定されることを前提とします。したがって、TP1/NET/HSC を使用する UAP をアプリケーション起動する場合、テキスト形態を含めた形式でメッセージを編集する必要があります。

(a) テキスト形態の種別

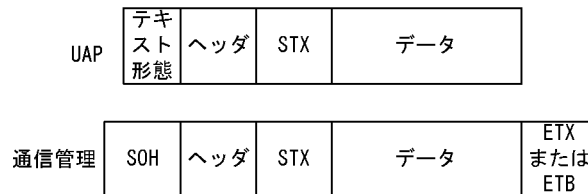
テキスト形態の種別を次の図に示します。

図 2-3 テキスト形態の種別 (HSC1 手順)

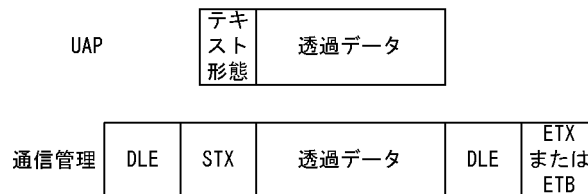
1. 通常テキスト



2. ヘッダ付きテキスト



3. 透過モードテキスト



4. ヘッダ付き透過モードテキスト



- (凡例) STX : テキストの最初に付ける伝送制御符号
 ETX : テキストの伝送の終了を示す伝送制御符号
 ETB : データを1ブロックに区切る伝送制御符号
 SOH : ヘッダの最初に付ける伝送制御符号
 DLE : 伝送制御記号を拡張するための符号

UAP および通信管理でのデータの扱いをテキスト形態の種別ごとに次の表に示します。

2. HSC1 手順の機能

表 2-2 UAP および通信管理でのデータの扱い (HSC1 手順)

項番	テキスト形態	種別	動作種別	データの扱い
1.	通常テキスト	UAP	送信	テキスト形態, ヘッダを作成します。
			受信	テキスト形態, ヘッダを受け取ります。
		通信管理	送信	STX, ETX (または ETB) を付加します。
			受信	STX, ETX (または ETB) を削除します。
2.	ヘッダ付きテキスト	UAP	送信	テキスト形態, ヘッダ, STX およびデータの作成します。
			受信	テキスト形態, ヘッダ, STX およびデータを受け取ります。
		通信管理	送信	SOH, ETX (または ETB) を付加します。
			受信	SOH, ETX (または ETB) を削除します。
3.	透過モードテキスト	UAP	送信	テキスト形態, 透過データの作成します。
			受信	テキスト形態, 透過データを受け取ります。
		通信管理	送信	DLE, STX, DLE および ETX (または ETB) を付加します。
			受信	DLE, STX, DLE および ETX (または ETB) を削除します。
4.	ヘッダ付き透過モードテキスト	UAP	送信	テキスト形態, ヘッダ, DLE, STX および透過データの作成します。
			受信	テキスト形態, ヘッダ, DLE, STX および透過データを受け取ります。
		通信管理	送信	SOH, DLE および ETX (または ETB) を付加します。
			受信	SOH, DLE および ETX (または ETB) を削除します。

(b) テキスト形態の数値表現と内容

送信するテキスト形態の数値表現と内容を表 2-3 に, 受信するテキスト形態の数値表現と内容を表 2-4 に示します。

表 2-3 送信するテキスト形態の数値表現と内容 (HSC1 手順)

項番	テキスト形態	数値表現 (16 進数字)	内容
1.	通常テキスト	00000002	通常データ
2.	ヘッダ付きテキスト	00000001	ヘッダ + STX + データ
3.	透過モードテキスト	00000004	透過モードで伝送されるデータ
4.	ヘッダ付透過モードテキスト	00000003	ヘッダ + DLE + STX + 透過データ

表 2-4 受信するテキスト形態の数値表現と内容 (HSC1 手順)

項番	テキスト形態	数値表現 (16 進数字)	内容
1.	通常テキスト	00000002	通常データ
2.	ヘッダ付きテキスト	00000001	ヘッダ + STX + データ
3.	透過モードテキスト	00000004	透過モードで伝送されるデータ
4.	ヘッダ付透過モードテキスト	00000003	ヘッダ + DLE + STX + 透過データ

2.2 HSC1 手順の AP 間通信メッセージの送受信

TP1/NET/HSC は、AP 間通信メッセージの送受信をします。TP1/NET/HSC で使用するメッセージは一方送信メッセージです。一方送信メッセージには、優先と一般があります。また、メッセージの送信と受信にはそれぞれ、単一セグメント、複数セグメント、および列信の場合があります。列信の場合は、優先の一方送信メッセージは使用しません。

ここでは、通信管理の準備完了を示す COPNEVT 通知後からの、メッセージの送信と受信の流れを説明します。

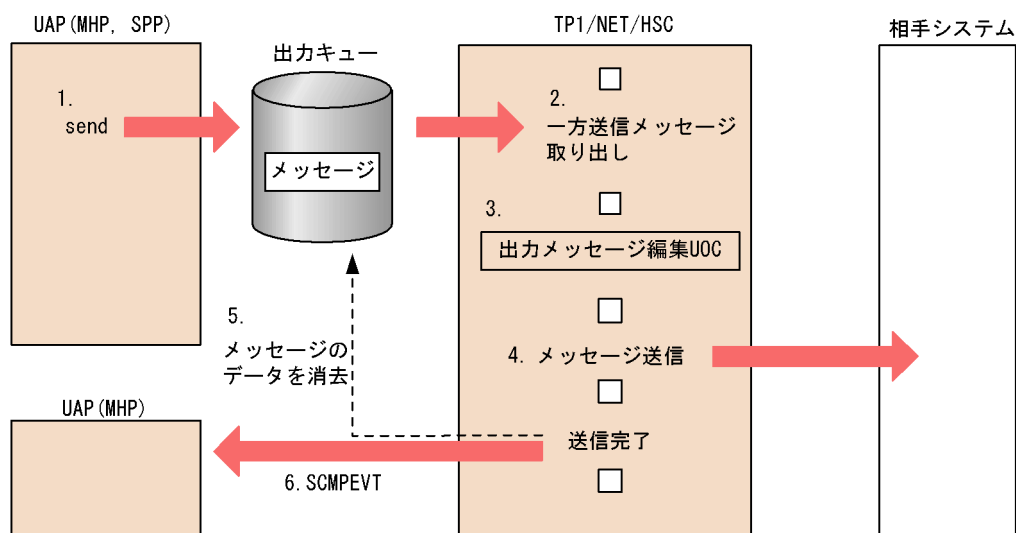
2.2.1 メッセージの送信

TP1/NET/HSC が、通信相手システムに一方送信メッセージを送信した場合の処理の流れを、単一セグメントの場合を例にして説明します。

また、列信でメッセージを送信する場合には、MCF 通信構成定義 (mcftalcle -w) の rowtrans オペランドで use を指定しておく必要があります。MCF 通信構成定義については、6 章の「TP1/NET/HSC 固有のシステム定義の種類 - HSC1 手順」を参照してください。

メッセージ送信の処理の流れを、次の図に示します。

図 2-4 HSC1 手順のメッセージ送信の流れ



1. MHP または SPP から any 型論理端末あてに、一方送信メッセージの送信要求をします。

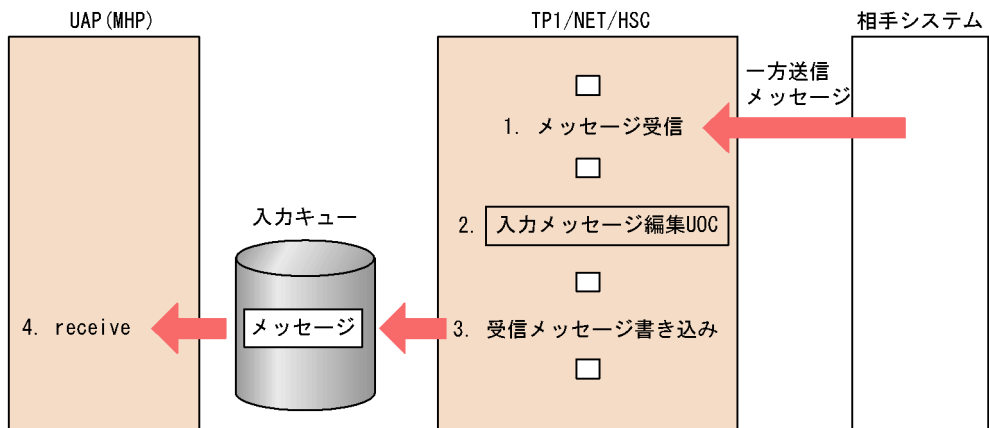
2. TP1/NET/HSC は、出力キューに書き込まれたメッセージを取り出します。
3. 出力メッセージ編集 UOC でメッセージを編集します。
4. 相手システムへメッセージを送信します。
5. 相手システムへの送信が完了した時点で、出力キューのメッセージを消去します。
6. 1. の send 関数で送信完了通知イベントを通知させることを指定した場合、送信完了通知イベント (SCMPEVT) を通知します。また、正常に送信できなかった場合、send 関数の指定によって送信障害通知イベント (SERREVT) を通知します。SERREVT については、「付録 B.1 HSC1 手順の障害」を参照してください。

2.2.2 メッセージの受信

TP1/NET/HSC が、通信相手システムから一方送信メッセージを受信した場合の処理の流れを、単一セグメントの場合を例にして説明します。

メッセージ受信の流れを、次の図に示します。

図 2-5 HSC1 手順のメッセージ受信の流れ



1. 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. MHP は receive 関数を呼び出してメッセージを受け取ります。

2.3 HSC1 手順の機能

TP1/NET/HSC は HSC1 手順のオプション機能のうち、次の表に示す範囲をサポートします。

表 2-5 HSC1 手順の機能と TP1/NET/HSC の扱い

HSC1 手順の機能	使用の可否	TP1/NET/HSC の扱い
ブロック転送		<ul style="list-style-type: none"> UAP からの send の 1 セグメントを 1 ブロックとして相手システムに送信します。 相手システムから受信したメッセージテキストは、ETX まで組み立ててから UAP に引き渡します。
列信		送信、受信の両方をサポートします。 (列信の送信は、一般の一方送信メッセージだけを対象とします)
逆中断 (RVI)		送信、受信の両方をサポートします。 (逆中断の送信は、相手システムからのメッセージを受信中に、ETX を受信したあと、優先の一方送信メッセージを送信する場合には行います。一般の一方送信メッセージの送信では、逆中断は行いません)
一時送信延期 (TTD)		送信、受信の両方をサポートします。
一時受信抑止 (WACK)		送信の場合は、MCF 通信構成定義 (mcfalccn -B) でサポートするかどうかを指定できます。
通常テキスト		送信、受信の両方をサポートします。
ヘッダ付きテキスト		
透過モードテキスト		
ヘッダ付き透過モードテキスト		
コンテンツ制御		MCF 通信構成定義 (mcfalccn -y) で優先局、非優先局を指定できます。
制御付き会話	×	サポートしません。

(凡例)

: TP1/NET/HSC の HSC1 手順で使用できます。

× : TP1/NET/HSC の HSC1 手順では使用できません。

3

HSC2 手順の機能

TP1/NET/HSC は、使用する手順によって、AP 間通信の方法に違いがあります。この章では、HSC2 手順を使用した場合の AP 間通信の方法、メッセージ送受信の流れ、および HSC2 手順の機能サポートについて説明します。

-
- 3.1 HSC2 手順の AP 間通信の仕組み
 - 3.2 HSC2 手順の AP 間通信メッセージの送受信
 - 3.3 HSC2 手順の機能
 - 3.4 回線リスト定義ユティリティ（非同期モード）
 - 3.5 相手ターミナル ID リスト定義ユティリティ（同期モード）
 - 3.6 EOT 交換
-

3.1 HSC2 手順の AP 間通信の仕組み

TP1/NET/HSC の HSC2 手順を使用すると、HSC2 手順に対応する相手システムと AP 間通信ができます。ここでは、HSC2 手順の非同期モードおよび同期モードを使用した場合の AP 間通信の流れ、コネクションや論理端末を使用した AP 間通信の仕組みについて説明します。

3.1.1 HSC2 手順（非同期モード）の送受信の方法

HSC2 手順（非同期モード）を使用したメッセージの送受信の手順は、次のとおりです。

1. コネクションの確立
2. 回線の接続
3. AP 間通信
4. 回線の切断
5. コネクションの解放
6. コネクション切断後の再確立

（1）コネクションの確立

TP1/NET/HSC は、コネクションを確立してから、相手システムとの回線を接続して、メッセージの送受信をします。コネクションは、通信管理との論理的な通信路です。

ユーザは、通信相手システムと ID 交換をするかどうかを、MCF 通信構成定義 (mcftalccn -j) の idchange オペランドで指定できます。ID 交換をすることを指定した場合、回線接続時にターミナル ID を交換して、正しい相手システムであるかどうかを確認できます。ID 交換をしないことを指定した場合、ターミナル ID による相手システムの確認をしないでコネクションを確立し、回線の接続をします。

コネクションの確立には、自動確立と手動確立があります。コネクションが確立すると、状態通知イベント (COPNEVT) を通知します。

自動確立

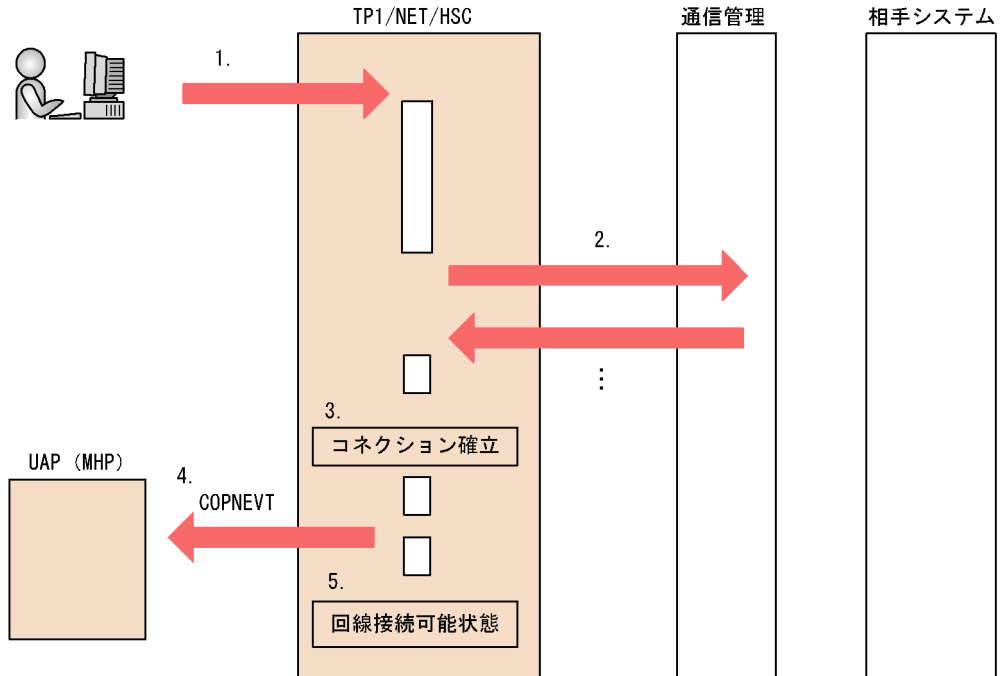
MCF 通信構成定義 (mcftalccn -i) で auto を指定した場合、OpenTP1 開始時および再開開始時に自動的に確立します。

手動確立

MCF 通信構成定義 (mcftalccn -i) で manual を指定した場合、運用コマンド (mcfactcn) で確立します。

HSC2 手順の場合のコネクションの確立について、次の図に示します。

図 3-1 HSC2 手順（非同期モード）のコネクションの確立



1. OpenTP1 システムの開始および再開始，または運用コマンド（mcftacten）を入力します。
2. TP1/NET/HSC は，通信管理に対して，該当するコネクションに定義されているすべての回線の接続準備をします。
3. 1 回線の回線接続の準備が終了した時点で，コネクション確立とみなされます。
4. TP1/NET/HSC は，コネクションが確立されると状態通知イベント（COPNEVT）を通知します。MCF アプリケーション定義に COPNEVT が定義されていない場合は，MHP は起動しません。
5. コネクションが確立し，相手システムとの回線を接続できる状態です。

（2）回線の接続

コネクション確立後，回線の接続をします。

回線が接続されると，自システムの論理端末，回線，相手システムの三者の結び付きは，回線が切断されるまで固定しています。したがって，回線の接続から切断までの間，メッセージは固定された論理端末と回線を経由して送受信されます。

回線の接続には，発信と着信の場合があります。

（a）発信の場合の回線の接続

ID 交換する場合

1. コネクション確立後に send 関数でメッセージの送信を要求すると，TP1/NET/

3. HSC2手順の機能

HSC は、MCF 通信構成定義 (mcftalcle -d) の dialno オペランドで指定した相手システムのダイヤル番号を決定します。

2. MCF 通信構成定義 (mcftalccn -n) の tid オペランドで指定した相手システムのターミナル ID と、1. のダイヤル番号でダイヤリングをされた相手システムからのターミナル ID を交換します。ID が一致した場合、回線を接続します。ID が一致しないときは、障害通知イベント (CERREVT) が通知されます。

ID 交換しない場合

1. コネクション確立後に send 関数でメッセージの送信を要求すると、TP1/NET/HSC は、MCF 通信構成定義 (mcftalcle -d) の dialno オペランドで指定した相手システムのダイヤル番号を決定します。
2. 1. のダイヤル番号でダイヤリングをし、相手システムが受諾すると、回線が接続されます。

(b) 着信の場合の回線の接続

ID 交換する場合

1. 相手システムからの回線接続を着信した場合、ID を交換します。ID が一致していれば相手システムからのメッセージの送信を待ちます。UAP に対して回線接続したことの通知はしません。
2. ID が許可されないときは回線を切断し、障害通知イベント (CERREVT) は通知されません。
3. 回線接続後、TP1/NET/HSC は、MCF 通信構成定義 (mcftalcle -n) の tid オペランドから、受信したターミナル ID に対応する論理端末を検索し、通信に使用する論理端末名称を決定します。

ID 交換しない場合

1. 相手システムからの回線接続を着信した場合、TP1/NET/HSC は、相手システムからのメッセージを受信します。最終セグメントを受信した時点で、論理端末名称決定 UOC を呼び出します。
2. 論理端末名称決定 UOC でメッセージ送受信に使用する論理端末を決定します。論理端末名称決定 UOC が登録されていない場合、該当するコネクションに定義されている論理端末のうち、先頭の論理端末を使用します。ただし、先頭の論理端末がほかの回線と結び付いている場合は、回線を切断します。

(3) AP 間通信

回線が接続すると、TP1/NET/HSC は、相手システムと AP 間通信をします。AP 間通信メッセージの送受信の流れについては、「3.2 HSC2 手順の AP 間通信メッセージの送受信」を参照してください。

(4) 回線の切断

相手システムと接続中の回線を切断する方法は次のとおりです。

(a) 相手システムからの切断

相手システムから回線切断要求を受信すると、回線が切断されます。

(b) 自システムからの切断

EOT 交換または時間監視の結果、一定時間にメッセージの送受信が発生しない場合、TP1/NET/HSC は、回線を切断します。EOT 交換については、「3.6.1 非同期モードの EOT 交換」を参照してください。

(5) コネクションの解放

コネクションの解放には、自動解放と手動解放があります。コネクションが解放されると、状態通知イベント (CCLSEVT) を通知します。

自動解放

オンライン終了時、自動的にコネクションを解放します。

手動解放 (正常解放)

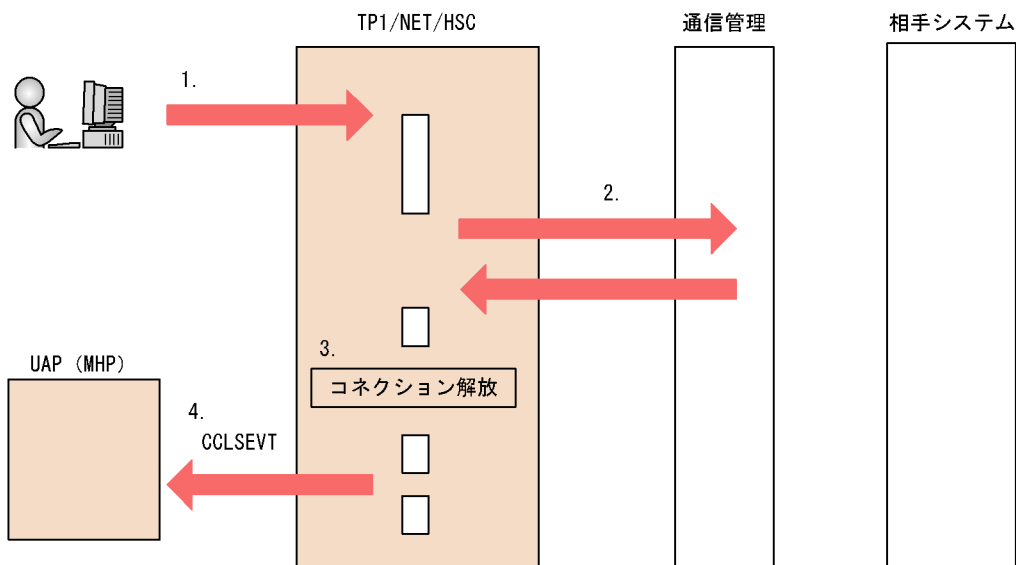
運用コマンド (meftdeten) に -f オプションを指定しないで入力して、コネクションを正常に解放します。コマンド入力時に送受信中のメッセージがある場合、送受信が完了した時点でコネクションを解放します。

手動解放 (強制解放)

運用コマンド (meftdeten) に -f オプションを指定して入力して、コネクションを強制的に解放します。コマンド入力時に送受信中のメッセージがある場合、送信メッセージは出力キューに戻し、受信メッセージは破棄してコネクションを解放します。

コネクションの自動解放および手動解放 (正常解放) の流れを次の図に示します。

図 3-2 HSC2 手順（非同期モード）の接続の解放



1. OpenTP1 システムの終了，または運用コマンド（mcftdctn）を入力します。
2. TP1/NET/HSC は，該当するコネクション配下のすべての回線を切断します。
3. すべての回線が切断され，コネクション解放の準備が終了すると，コネクションが解放されます。
4. TP1/NET/HSC は，コネクションが解放されると状態通知イベント（CCLSEVT）を通知します。MCF アプリケーション定義に CCLSEVT が定義されていない場合は，MHP は起動しません。

（6）コネクション切断後の再確立

すべての回線が使用できなくなると，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，障害通知イベント（CERREVT）を通知します。

コネクションを切断したあと，ユーザはログメッセージでコネクションの再確立が必要かどうかを判断します。必要であれば，運用コマンド（mcftactcn）を入力してコネクションの再確立をします。

3.1.2 HSC2 手順（同期モード）の送受信の方法

HSC2 手順（同期モード）を使用したメッセージの送受信の手順は，次のとおりです。

1. コネクションの確立
2. 回線の接続
3. AP 間通信
4. 回線の切断
5. コネクションの解放

6. コネクション切断後の再確立

(1) コネクションの確立

TP1/NET/HSC は、コネクションを確立してから、相手システムとの回線を接続して、メッセージの送受信をします。コネクションは、通信管理との論理的な通信路です。

ユーザは、通信相手システムと ID 交換をするかどうかを、MCF 通信構成定義 (mcftalccn -j) の idchange オペランドで指定できます。ID 交換をすることを指定した場合、コネクション確立時に相手ターミナル ID リストを読み込み、回線接続時にターミナル ID を交換して、正しい相手システムであるかどうかを確認できます。ID 交換をしないことを指定した場合、相手システムの確認をしないでコネクションを確立し、回線の接続をします。

ID 交換をすることを指定すると、通信相手システムとコネクションを確立するときに相手ターミナル ID リストの読み込みをします。相手ターミナル ID リストには、発信用と着信用があり、あらかじめ定義で指定しておく必要があります。

発信用相手ターミナル ID リストファイル

相手システムのターミナル ID と電話番号の対応を定義したファイルです。このファイルは、発信用相手ターミナル ID リストの定義 (mcftstf, mcftstlst) で指定します。定義の詳細は、6 章の「TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (同期モード)」を参照してください。

このファイルは定義で指定したあと、相手ターミナル ID 定義ユティリティで作成し、\$DCCONFPATH のディレクトリ下に作成します。

TP1/NET/HSC は、UAP から回線接続要求があった場合、このファイルの内容を参照し、相手システムの電話番号を決定してダイヤリングをします。

着信用相手ターミナル ID リストファイル

着信した相手システムのターミナル ID と、その相手システムと通信する UAP が使用する論理端末名称との対応を定義したファイルです。このファイルは、着信用相手ターミナル ID リストの定義 (mcftstf, mcftstlst) で指定します。定義の詳細は、6 章の「TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (同期モード)」を参照してください。

このファイルは定義で指定したあと、相手ターミナル ID 定義ユティリティで作成し、\$DCCONFPATH のディレクトリ下に作成します。

TP1/NET/HSC は、相手システムからの回線接続要求を着信したとき、読み込まれたファイルの内容を参照し、着信したコネクションとそのコネクションと結び付く論理端末名称を決定します。

コネクションの確立には、自動確立と手動確立があります。コネクションが確立すると、状態通知イベント (COPNEVT) を通知します。

自動確立

MCF 通信構成定義 (mcftalccn -i) で auto を指定した場合、OpenTP1 開始時および

3. HSC2 手順の機能

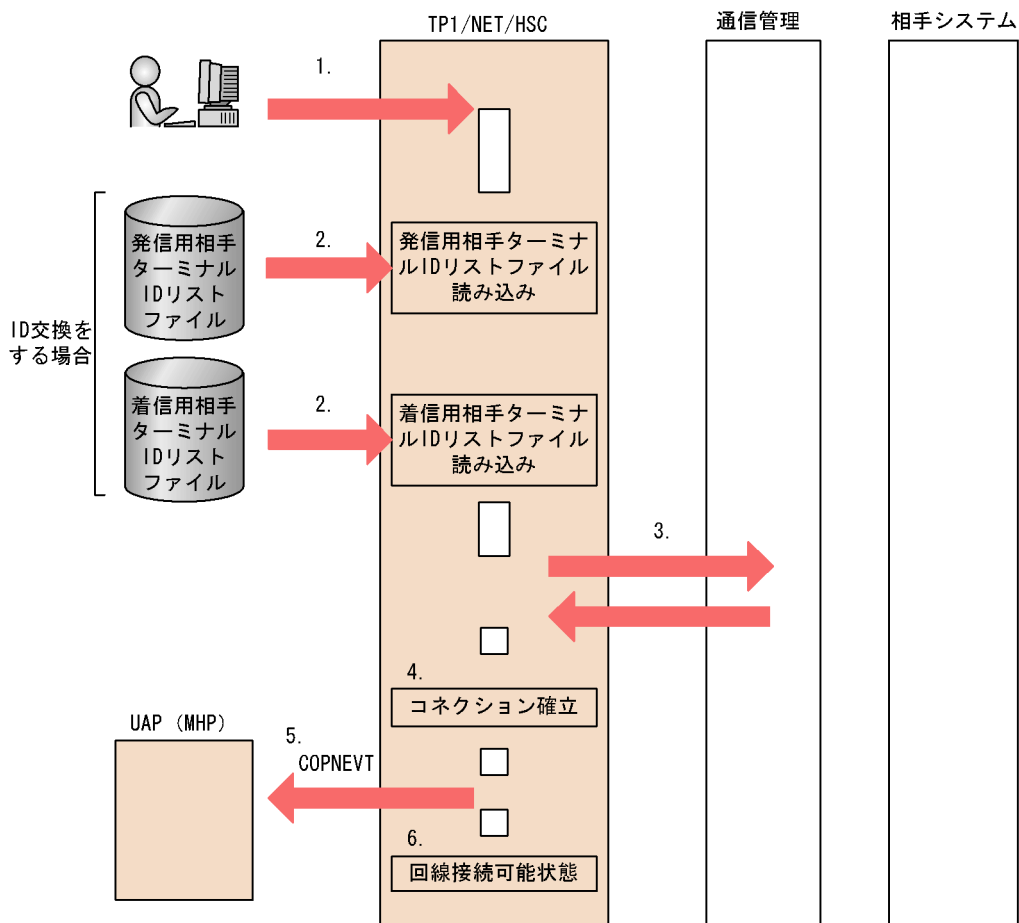
再開時に自動的に確立します。

手動確立

MCF 通信構成定義 (mcftalccn -i) で manual を指定した場合、運用コマンド (mcftactcn) で確立します。

HSC2 手順の場合の接続の確立について、次の図に示します。

図 3-3 HSC2 手順 (同期モード) の接続の確立



1. OpenTP1 システムの開始および再開時、または運用コマンド (mcftactcn) を入力します。
2. ID 交換をする場合、TP1/NET/HSC は、定義で指定された発信用相手ターミナル ID リストファイル、および着信用相手ターミナル ID リストファイルを読み込みます。
3. TP1/NET/HSC は、通信管理とコネクション確立の準備をします。
4. コネクションが確立されます。
5. TP1/NET/HSC は、コネクションが確立されると状態通知イベント (COPNEVT) を通知します。MCF アプリケーション定義に COPNEVT が定義されていない場合は、

MHP は起動しません。

6. コネクションが確立し、相手システムとの回線を接続できる状態です。

(2) 回線の接続

TP1/NET/HSC は、相手システムと回線を接続します。回線を接続するときに、相手システムの確認のためにターミナル ID の交換をします。ターミナル ID の交換をしない場合は、MCF 通信構成定義 (mcftalccn -j) の idchange オペランドで nouse を指定してください。

交換の結果、接続できるかどうかを決定します。接続できない場合は、直ちに回線を切断します。接続できる場合は、メッセージの送受信をする MHP を決定します。以後、回線が切断されるまで、この MHP は変更できません。また、回線接続中には、送受信中の相手ターミナル ID を別の ID と切り替えられません。

回線の接続には、発信と着信の場合があります。

(a) 発信の場合の回線の接続

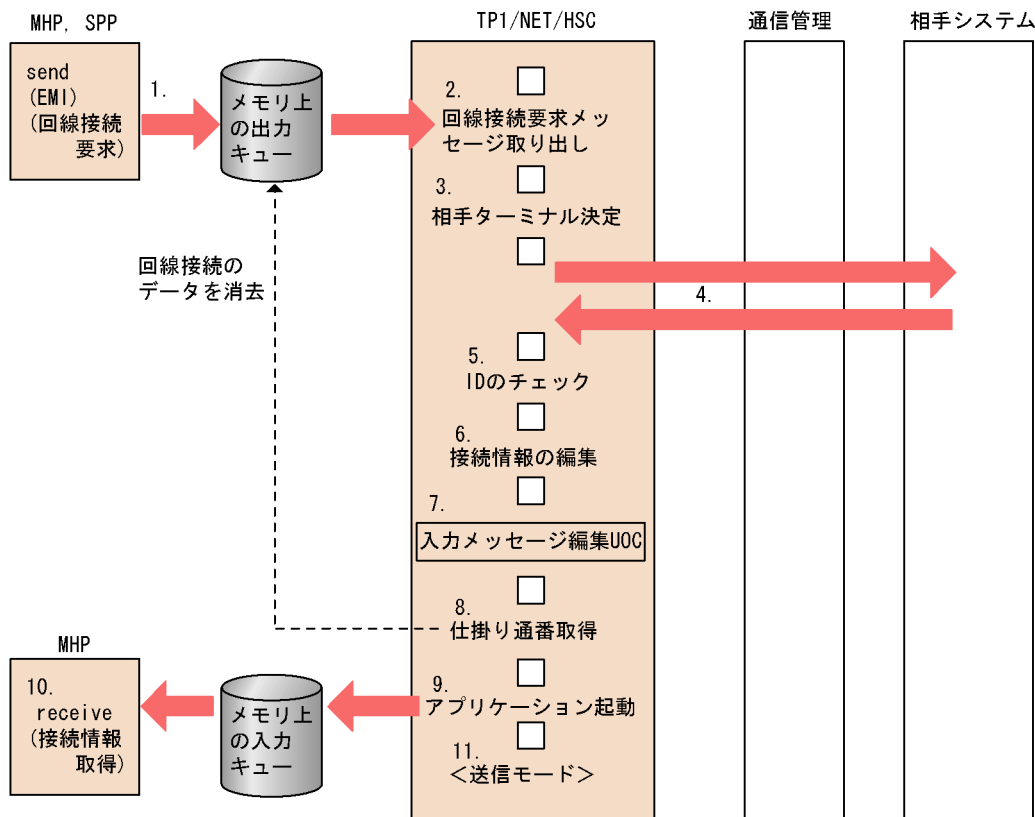
送信する自システムのターミナル ID は、MCF 通信構成定義 (mcftalccn -n) の tid オペランドで指定したコードであり、EBCDIK コードに変換します。

受信したメッセージに設定されている相手ターミナル ID は、JIS8 コードに変換して回線接続指示で指定された ID コードと比較します。

ID が不一致の場合、状態通知イベント (CERREVT) を通知し、論理端末を閉塞します。この場合の閉塞は、運用コマンド (mcftactle) を入力するか、相手システムから該当する論理端末にメッセージの送信があったときに解除されます。

発信の場合の回線の接続について、次の図に示します。

図 3-4 発信の場合の回線の接続（同期モード）



1. send 関数で相手システムへ回線接続を要求します。
2. メモリ上の出力キューから、回線接続要求メッセージを取り出します。
3. ID 交換をする場合、TP1/NET/HSC は、コネクション確立時に読み込まれた相手ターミナル ID リストファイルを検索し、発信に使用するダイヤル番号を決定します。ID 交換をしない場合、TP1/NET/HSC は、回線接続指示テキストの相手システムダイヤル番号を使用して、通信管理に対して回線の接続要求します。
4. 相手システムと回線接続の準備をします。
5. 回線の相手システムから回線接続の完了を受け取ります。ID 交換をする場合は、接続要求時の相手ターミナル ID と比較します。
6. 相手システムから受け取った接続情報を編集します。
7. 入力メッセージ編集 UOC を呼び出します。UOC が登録されていない場合、呼び出さないうちの処理へ続きます。
8. TP1/NET/HSC は、メモリ上の出力キューの回線接続のデータを消去し、仕掛け通番を取得します。
9. TP1/NET/HSC は、入力メッセージ編集 UOC で決定したアプリケーション、または MCF 通信構成定義 (mcftalcle -v) で指定したアプリケーションを起動します。
10. UAP は receive 関数を呼び出して、回線の接続情報を受け取ります。

11. 発信によって回線を接続したため、TP1/NET/HSC は、送信モードになります。

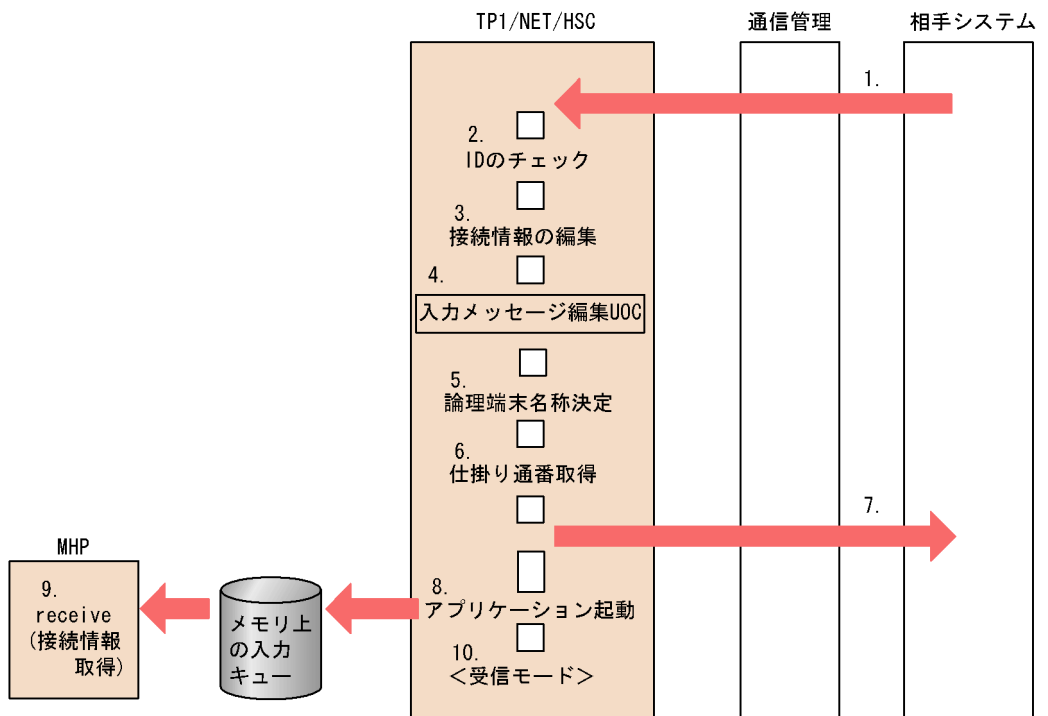
(b) 着信の場合の回線の接続

受信したメッセージに設定されている相手ターミナル ID は、JIS8 コードに変換し、コネクション確立時に読み込まれた着信用相手ターミナル ID リストに登録されている ID リストと比較します。

受信した ID が、着信用相手ターミナル ID リストにない場合、回線を切断します。ID チェックの結果、ID が登録されている場合、MCF 通信構成定義 (mcfalccn -n) の tid オペランドで指定した自システムのターミナル ID を EBCDIK コードに変換して、相手システムに送信します。このとき、ID の長さは定義で指定されている長さをそのまま使用します。

着信の場合の回線の接続について、次の図に示します。

図 3-5 着信の場合の回線の接続 (同期モード)



1. 相手システムから回線の接続情報を受け取ります。
2. ID 交換をする場合、TP1/NET/HSC は、受け取った ID が着信用相手ターミナル ID リストに登録されているかどうかをチェックします。
3. TP1/NET/HSC は、相手システムから受け取った接続情報を編集します。
4. 入力メッセージ編集 UOC を呼び出します。UOC が登録されていない場合、呼び出さずに次の処理へ続きます。

3. HSC2 手順の機能

5. ID 交換をする場合、着信用相手ターミナル ID リストファイルの検索結果から、相手システムとの通信に使用する論理端末名称を決定します。
ID 交換をしない場合、該当するコネクションに定義されている論理端末のうち、先頭の論理端末を使用します。
6. 仕掛け通番を取得します。
7. TP1/NET/HSC は、回線接続の受諾応答を送信します（ID 交換をする場合は、自システムの ID を送信します）。
8. TP1/NET/HSC は、入力メッセージ編集 UOC で決定したアプリケーション、または MCF 通信構成定義（mcfalcle -v）で指定したアプリケーションを起動します。
9. UAP は receive 関数を呼び出して、回線の接続情報を受け取ります。
10. 着信によって回線を接続したため、TP1/NET/HSC は、受信モードになります。

（3）AP 間通信

回線が接続すると、TP1/NET/HSC は、回線の接続時に起動された MHP を使用して相手システムと AP 間通信をします。

AP 間通信メッセージの送受信の流れについては、「3.2 HSC2 手順の AP 間通信メッセージの送受信」を参照してください。

（4）回線の切断

相手システムと接続中の回線を切断する方法は次のとおりです。

回線接続の流れについては、「付録 A.3 HSC2 手順（同期モード）のメッセージの送受信」を参照してください。

（a）メッセージ送信時の切断

メッセージ送信時に、sendsync 関数に回線切断指示のテキスト形態を設定して送信すると、回線が切断されます。

（b）MHP 終了での切断

MHP を return で終了させると回線が切断されます。

（c）相手システムからの切断

相手システムから回線切断要求を受信すると、回線が切断されます。

（5）コネクションの解放

回線切断後、コネクションを解放します。

コネクションの解放には、自動解放と手動解放があります。コネクションが解放されると、状態通知イベント（CCLSEVT）を通知します。

自動解放

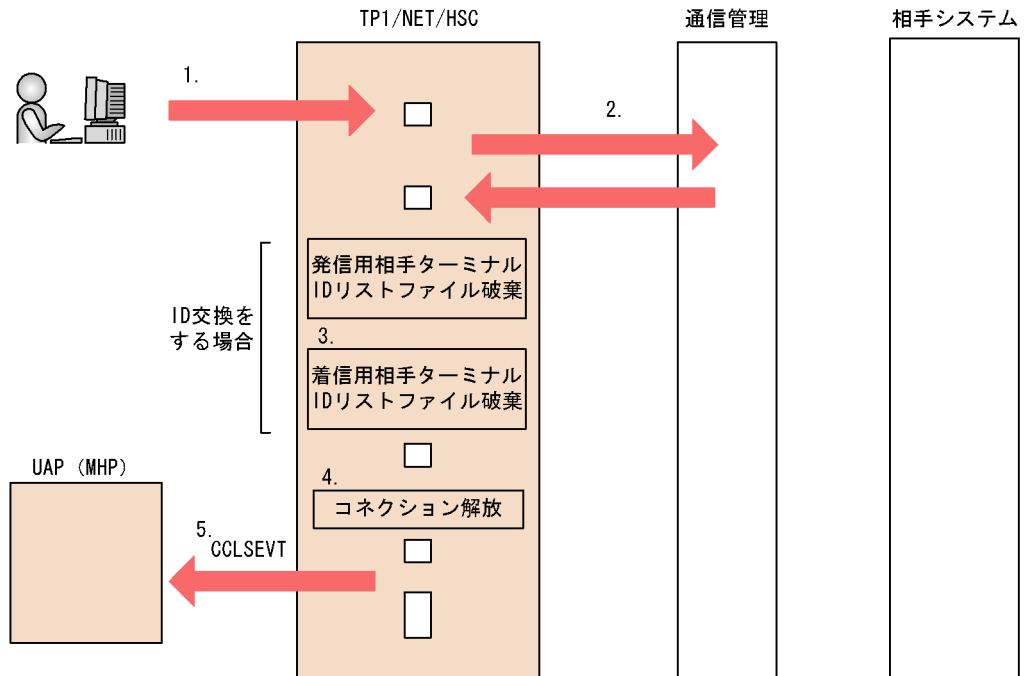
オンライン終了時、自動的にコネクションを解放します。

手動解放

運用コマンド (`mcfstdcten -f`) を入力して、コネクションを解放します。

コネクションの解放について次の図に示します。

図 3-6 HSC2 手順 (同期モード) のコネクションの解放



1. OpenTP1 システムの終了、または運用コマンド (`mcfstdcten -f`) を入力します。
2. TP1/NET/HSC は、コネクション解放の準備をします。
3. ID 交換をする場合、TP1/NET/HSC は、発信用相手ターミナル ID リストファイル、および着信用相手ターミナル ID リストファイルを破棄します。
4. コネクションが解放されます。
5. TP1/NET/HSC は、コネクションが解放されると状態通知イベント (CCLSEVT) を通知します。MCF アプリケーション定義に CCLSEVT が定義されていない場合は、MHP は起動しません。

(6) コネクション切断後の再確立

TP1/NET/HSC の HSC2 手順を使用した場合、通信管理から回復不能なコネクションの障害を検出すると、コネクションを切断します。コネクションが切断状態になると、障害通知イベント (CERREVT) を通知します。

コネクションを切断したあと、ユーザはログメッセージでコネクションの再確立が必要かどうかを判断します。必要であれば、運用コマンド (`mcfstacten`) を入力してコネクションの再確立をします。

3.1.3 コネクションと論理端末の関係

(1) 非同期モードの場合

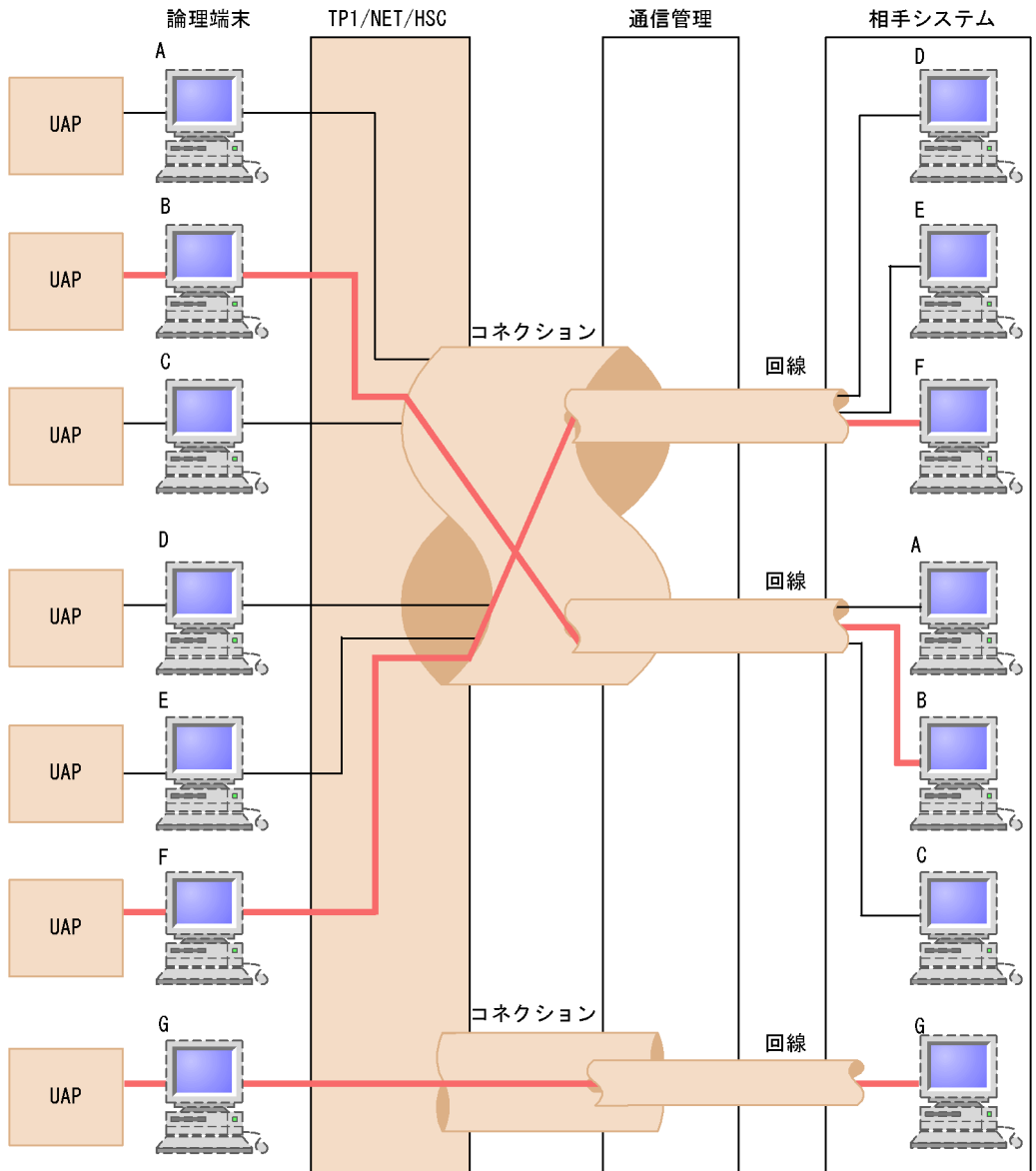
TP1/NET/HSC を使用してメッセージの送受信をする場合、ユーザはコネクションと論理端末を対応付けて定義する必要があります。

コネクションは、メッセージの送受信をするための論理的な通信路です。通信管理との接点であり、公衆回線網に加入している複数の回線を 1 グループとしてまとめるための通信点です。コネクションは、MCF 通信構成定義 (mcftalccn) で指定します。HSC2 手順では、システムに複数のコネクションを指定できます。

論理端末 (LE) は、UAP がメッセージの送受信をするための窓口です。論理端末は、MCF 通信構成定義 (mcftalcle) で指定します。HSC2 手順では、一つのコネクションに対し、論理端末を複数指定できます。論理端末は相手システムの論理端末と 1 対 1 に対応します。自システムの論理端末と相手システムの論理端末は、送受信データが発生した時点で接続されます。接続に使用する回線は、コネクションでグループ化されている中から動的に選択されます。

コネクションと論理端末の関係を次の図に示します。

図 3-7 コネクションと論理端末の関係 (HSC2 手順 (非同期モード))



(凡例) ——— : 構成の結び付きを示します。
 ——— (red) : 回線接続中の構成の結び付きの例を示します。

(2) 同期モードの場合

TP1/NET/HSC を使用してメッセージの送受信をする場合、ユーザはコネクションと論理端末を対応付けて定義する必要があります。

コネクションは、メッセージの送受信するための通信管理との論理的な通信路です。

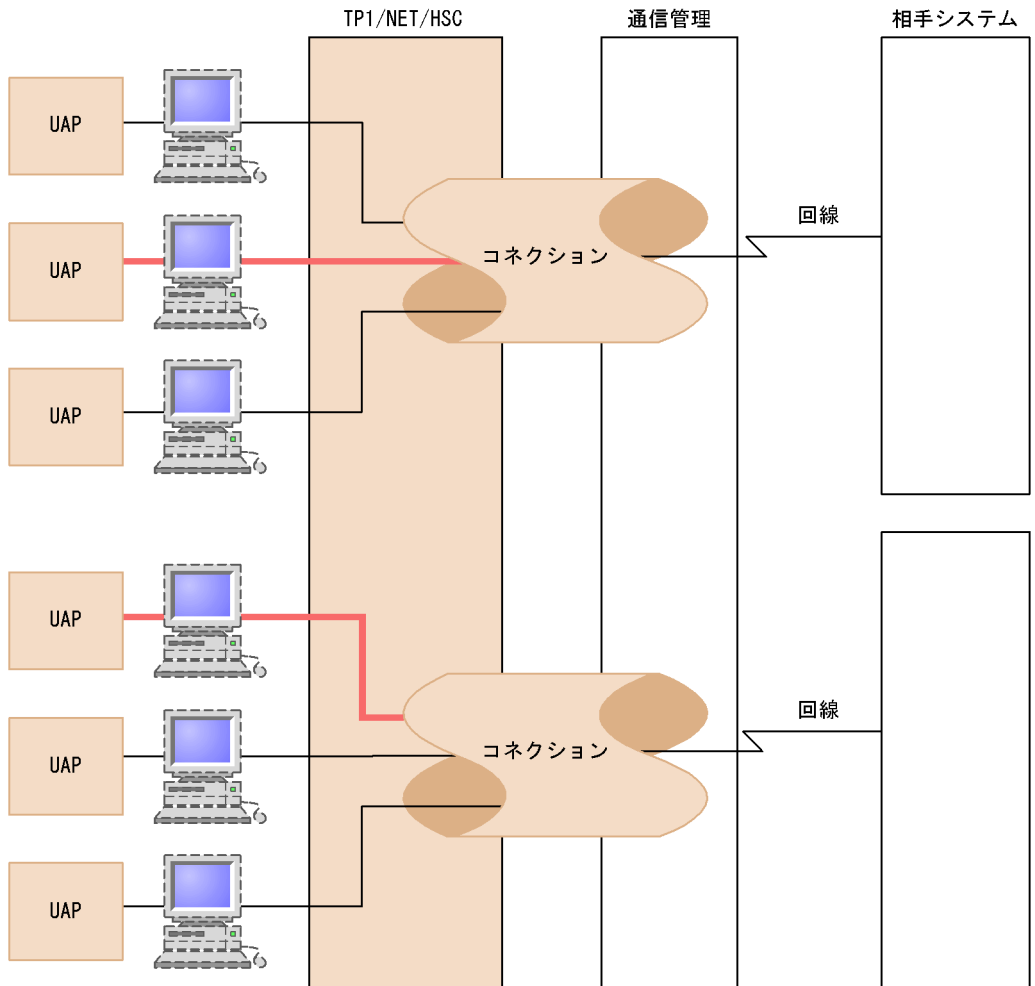
3. HSC2 手順の機能

公衆回線網の回線と 1 対 1 に対応します。コネクションは、MCF 通信構成定義 (mcftalccn) で指定します。HSC2 手順では、システムに複数のコネクションを指定できます。

論理端末 (LE) は、UAP がメッセージの送受信をするための窓口です。論理端末は、MCF 通信構成定義 (mcftalcle) で指定します。HSC2 手順では、一つのコネクションに対し、論理端末を複数指定できます。ただし、相手システムと回線がつながったとき、コネクションと論理端末は 1 対 1 で結び付き、回線が切断されるまで、ほかの論理端末とは結び付きません。

コネクションと論理端末の関係を次の図に示します。

図 3-8 コネクションと論理端末の関係 (HSC2 手順 (同期モード))



(凡例) — : 構成の結び付きを示します。

— : 回線接続中の構成の結び付きの例を示します。回線接続中は、コネクション、論理端末、UAPは1対1対1の関係です。

3.1.4 論理端末とアプリケーションの型の関係

TP1/NET/HSC では、メッセージの送受信をするために、論理端末タイプとアプリケーションの型を定義します。また、アプリケーションの型によって、UAP ができる機能が違います。ここでは、TP1/NET/HSC の HSC2 手順を使用する場合、TP1/NET/HSC で定義する内容について説明します。

論理端末の端末タイプ

論理端末は、メッセージの送受信をするためのユーザインタフェースで、利用形態に

3. HSC2 手順の機能

よる端末タイプがあります。TP1/NET/HSC では、相手システムとのメッセージ送受信をサポートするために、any 型（任意型論理端末）を端末タイプとします。論理端末の端末タイプは、MCF 通信構成定義（mcftalcle -t）で指定します。

論理端末の属性

HSC2 手順の場合、メッセージ送受信の形態が非同期モードと同期モードに分かれるため、モードに合わせて論理端末の属性を定義する必要があります。論理端末の属性は、MCF 通信構成定義（mcftalcle -b）の attr オペランドで指定します。非同期モードの場合、このオペランドの指定を省略するか、async を指定してください。同期モードの場合は、sync を指定してください。

async：論理端末は UAP と結び付くことはなく、非同期に送受信をします。

sync：論理端末は UAP と 1 対 1 で結び付き、同期に送受信をします。

同期モードの場合、論理端末と UAP は、相手システムとの回線が接続したときに結び付けられ、回線が切断したときに結び付きが終了します。また同期モードでは、論理端末と UAP の結び付きの単位でメッセージの入力通番（仕掛り通番）を管理します。仕掛り中の論理端末に対して、別の UAP からの回線接続要求があった場合は、出力キューに保留され、論理端末の仕掛り終了後に実行されます。

アプリケーションの型

アプリケーションは、ユーザが送受信データの中で指定したアプリケーション名をキーとして一つの UAP（MHP）プロセスで実行されます。アプリケーションには、処理形態によって型があります。アプリケーションの型は、MCF アプリケーション定義（mcfaalcap -n）で指定します。TP1/NET/HSC で扱うアプリケーションの型は、非応答型（noans：受信メッセージに対して応答しない MHP の型）だけです。アプリケーションの定義については、マニュアル「OpenTP1 システム定義」を参照してください。

アプリケーション名

アプリケーション名は、相手システムとの接続情報を基に、MHP を起動する名称です。

アプリケーション名は、相手システムとの回線接続時に入力メッセージ編集 UOC で決定する方法と、システム定義で決定する方法があります。アプリケーション名の決定については、「5.1.3 入力メッセージの編集とアプリケーション名の決定」を参照してください。また、システム定義で決定する場合は、MCF 通信構成定義（mcftalcle -v）で指定します。

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

表 3-1 論理端末の端末タイプ、メッセージの種類、アプリケーションの型、UAP インタフェース、および通信形態の関係 (HSC2 手順)

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

3.1.5 論理端末の閉塞と閉塞解除

論理端末は、コネクションと UAP の間に位置し、メッセージの送受信をする論理的なインタフェース概念です。UAP では、メッセージ受信時には入力元論理端末として、メッセージ送信時には出力先論理端末として論理端末を考慮して値を設定します。非同期モードの場合、相手システムと 1 対 1 で対応します。同期モードの場合、回線接続時、UAP は仕掛り中の論理端末と結び付き、その後、仕掛りが終了するまで同期送受信の通信接点として、その論理端末と結び付きます。

論理端末の状態を次に示します。

閉塞状態

UAP が送信要求または回線接続要求したメッセージを出力キューから取り出せない状態です。または、相手システムにメッセージを送信できない状態です。閉塞状態には、次の二つがあります。

- 暗黙閉塞
回線の未接続および障害によってシステムが暗黙的に閉塞しています。このとき着信すると、閉塞を解除して着信を受け入れます。
- 明示閉塞
運用コマンド (mcftdctle) で明示的に閉塞しています。このとき着信すると、相手システムに接続拒否を返します。

閉塞解除状態

論理端末が持つ機能を使用できる状態です。閉塞解除状態には、次の二つがあります。

- 非同期送信スケジュール可能
論理端末が持つすべての機能を使用できます。
- 非同期送信スケジュール抑止
非同期モードの場合、UAP が送信要求したメッセージのスケジュールを抑止します。コネクション内の回線に空きがない場合、または相手システムからのメッセージ受信中は、この状態になっています。
同期モードの場合、UAP の回線接続指示要求のメッセージのスケジュールを抑止します。UAP が相手システムとメッセージ送受信中はこの状態です。

(1) 論理端末の閉塞

論理端末の閉塞方法には、次に示す方法があります。

(a) 自動閉塞 (暗黙閉塞)

コネクションの解放または切断と同時に、自動的に論理端末を閉塞します。

(b) 手動閉塞 (明示閉塞)

ユーザが、運用コマンド (mcftdctle) を入力して論理端末を明示的に閉塞します。

(c) 障害閉塞 (暗黙閉塞)

論理端末レベルの障害 (障害通知イベントの CERREVT) が発生したとき、障害が発生した論理端末を暗黙的に閉塞します。

(2) 論理端末の閉塞解除

論理端末の閉塞解除方法には、次に示す方法があります。

(a) 自動解除

コネクションの確立と同時に、自動的に論理端末の閉塞解除をする方法です。

MCF 通信構成定義 (mcftalccn -l) で auto を指定しておく、コネクションの確立と同時に論理端末を自動的に閉塞解除します。この時点から、相手システムとの回線を接続できるようになります。ただし、運用コマンド (mcftdctle) で論理端末を閉塞している場合、自動的に閉塞解除はしません。

(b) 手動解除

運用コマンド (mcftactle) を入力して論理端末の閉塞解除をする方法です。

MCF 通信構成定義 (mcftalccn -l) で manual を指定した場合、コネクションが確立しても論理端末の閉塞解除はされません。この時点では、同期モードの回線接続要求のメッセージと非同期モードの送信メッセージ (send) は出力キューに保留され、同期モードで送信するメッセージ (sendsync) は論理端末で拒否されます。その場合、運用コマンド (mcftactle) を入力して論理端末の閉塞解除をします。

3.1.6 メッセージの分割と組み立て

論理メッセージは、付けられる情報やデータの内容が、処理層によって異なります。また、ユーザが作成した固有のデータは、データの内容によってテキスト形態が異なります。

(1) 論理メッセージの扱い

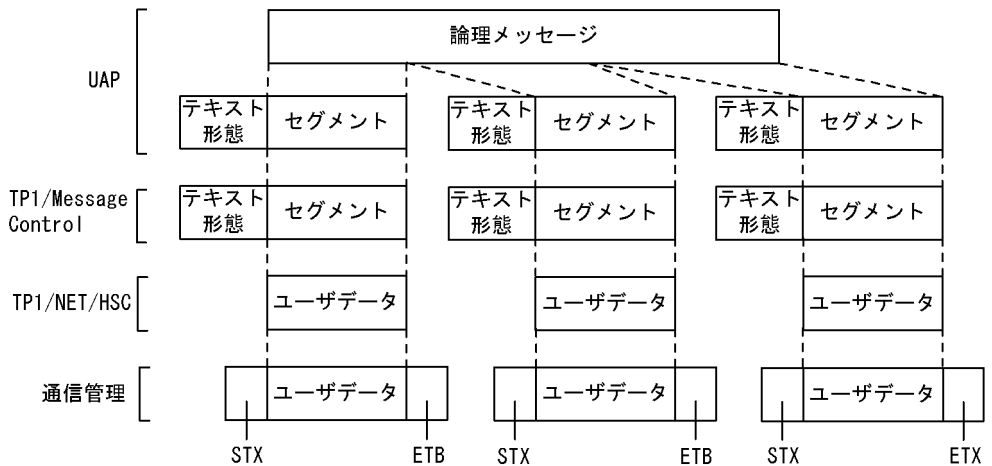
TP1/NET/HSC の論理メッセージは、複数のセグメントに分割できます。メッセージ送受信の関数で処理するセグメントの先頭には、MCF で使用するヘッダ領域があります。

このヘッダ領域の長さによって、バッファ形式 1 とバッファ形式 2 があります。通常はバッファ形式 1 を使用します。

OpenTP1 システムの UAP で作成した論理メッセージは、TP1/Message Control, TP1/NET/HSC, 通信管理の各処理層を介して相手システムに送信されます。相手システムからのメッセージの受信の場合は逆の順番になります。論理メッセージは、単独セグメントまたは複数セグメントのデータで構成されます。

処理層とデータの関係を次の図に示します。

図 3-9 処理層とデータの関係 (HSC2 手順)



- (凡例) STX : テキストの最初に付ける伝送制御符号)
 ETX : テキストの伝送の終了を示す伝送制御符号)
 ETB : データを1ブロックに区切る伝送制御符号)

(a) 送信メッセージの場合

非同期モード (論理端末の属性が `async`) の場合

MHP または SPP から `send` 関数で送信要求された各セグメントを 1 ブロックに対応付け、通信管理を介して相手システムに送信します。

同期モード (論理端末の属性が `sync`) の場合

`sendsync` で送信要求した 1 セグメントを 1 単位とし、通信管理を介して相手システムにメッセージを送信します。MHP が最終セグメント以外のセグメントを送信した場合 (`sendsync` で `DCMCFESI` を設定), 直ちに `sendsync` 関数をリターンし、最終セグメント (`sendsync` で `DCMCFEMI` を設定) を送信するまで、これまでのセグメントを保留します。TP1/NET/HSC では、各セグメントを 1 ブロックに対応付けて相手システムに送信し、最終セグメントの送信終了で終了させます。

セグメントの送信と MHP の動作を回線状態に同期させたい場合、1 セグメントで 1

論理メッセージとする形式でデータを編集し、列信で送信要求をする必要があります。

(b) 受信メッセージの場合

非同期モード（論理端末の属性が `async`）の場合

通信管理から受け取ったメッセージがセグメント分割されている場合、最終セグメント受信まで各セグメントを保留し、まとめて MHP に渡します。このとき、MHP は受信済みのセグメント数分の `receive` を呼び出して各セグメントを受け取ります。

同期モード（論理端末の属性が `sync`）の場合

MHP はセグメント分割された各メッセージを回線状態に同期して受け取ることはできません。したがって、回線上のテキスト受信と MHP の起動を同期させたい場合は、1 セグメントで 1 論理メッセージにする必要があります。セグメント分割されたメッセージを受信した場合、TP1/NET/HSC は最終セグメント受信まで各セグメントを保留し、最終テキスト受信後に終了させます。このとき、MHP はセグメント数分の `recvsync` を呼び出して各セグメントを受け取ります。

(2) テキスト形態と UAP でのデータの扱い

TP1/NET/HSC の HSC2 手順で扱うテキスト形態と、それぞれについての UAP でのデータの扱いを説明します。各テキスト形態は、16 進数字で表現します。テキスト形態の数値は、次に示すようにメッセージ送信時は送信領域に設定し、メッセージ受信時は受信領域に設定されます。送信領域、受信領域については、「4. メッセージ送受信インタフェース」を参照してください。

- メッセージ送信時
send 関数または `sendsync` 関数を呼び出すときに、送信するメッセージセグメントの先頭に、ユーザがテキスト形態を設定します。
通常テキストのデータは、EBCDIK コードで作成します。
- メッセージ受信時
`receive` 関数または `recvsync` 関数を呼び出して受信するメッセージセグメントの先頭に、TP1/NET/HSC がテキスト形態を設定します。

また、TP1/NET/HSC を使用する UAP では、アプリケーション起動をされた場合も同様に、「(a) テキスト形態の種別」に示すテキスト形態が設定されることを前提とします。したがって、TP1/NET/HSC を使用する UAP をアプリケーション起動する場合、テキスト形態を含めた形式でメッセージを編集する必要があります。

(a) テキスト形態の種別

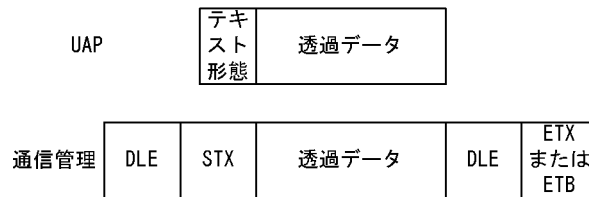
テキスト形態の種別を次の図に示します。

図 3-10 テキスト形態の種別 (HSC2 手順)

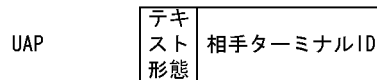
1. 通常テキスト



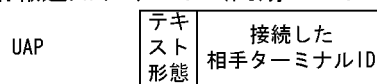
2. 透過モードテキスト



3. 回線接続要求テキスト (同期モードの場合に使用します)



4. 接続情報通知テキスト (同期モードの場合に使用します)



- (凡例) STX : テキストの最初に付ける伝送制御符号
 ETX : テキストの伝送の終了を示す伝送制御符号
 ETB : データを1ブロックに区切る伝送制御符号
 DLE : 伝送制御記号を拡張するための符号

UAP および通信管理でのデータの扱いをテキスト形態の種別ごとに次の表に示します。

表 3-2 UAP および通信管理でのデータの扱い (HSC2 手順)

項番	テキスト形態	種別	動作 種別	データの扱い
1.	通常テキスト	UAP	送信	テキスト形態とヘッダを作成します。
		TP1/NET/ HSC	送信	テキスト形態を削除して通信管理へ送信要求をします。
			受信	テキスト形態を付加します。
		通信管理	送信	STX, ETX (または ETB) を付加します。

3. HSC2 手順の機能

項番	テキスト形態	種別	動作種別	データの扱い
			受信	STX, ETX (または ETB) を削除します。
2.	透過モードテキスト	UAP	送信	テキスト形態と透過データを作成します。
		TP1/NET/HSC	送信	テキスト形態を削除して通信管理へ送信要求をします。
			受信	テキスト形態を付加します。
		通信管理	送信	DLE, STX, DLE および ETX (または ETB) を付加します。
			受信	DLE, STX, DLE および ETX (または ETB) を削除します。
3.	回線接続要求テキスト	UAP	発信	ID 交換をする場合、回線を接続するターミナル ID を設定し、send で発信要求をします。TP1/NET/HSC は相手ターミナル ID に対応したダイヤル番号 で通信管理に発信要求をします。ID 交換をしない場合、相手システムのダイヤル番号 を設定し、send で発信要求をします。相手システムのダイヤル番号 は、1 から 100 けたの符号なし整数 (アスタリスク (*), プラス (+), ビリオド (.) を含む) で指定します。
4.	接続情報通知テキスト	UAP	回線接続	ID 交換をする場合、TP1/NET/HSC は、テキスト形態で回線接続の発信、着信の区分を通知します。接続した相手ターミナル ID は、ID 交換時に相手システムから受信した ID を JIS8 コードに変換して設定し、通知します。したがって、ユーザには 1 から 15 けたの英数字で通知されます。ID 交換をしない場合、テキスト形態だけ通知されます。

注

ダイヤル番号の形式は次のとおりです。

V25.bis の場合：mmmmmmm

V21 (通常発信) の場合：mmmmmmm+

V21 (短縮ダイヤル, ペア型閉域) の場合：.nn+

- mmmmmmm：1 ~ 100 けたの符号なし整数です。
- nn：通信管理で回線網に登録する番号です。

(b) テキスト形態の数値表現と内容

送信するテキスト形態の数値表現と内容を表 3-3 に、受信するテキスト形態の数値表現

と内容を表 3-4 に示します。

表 3-3 送信するテキスト形態の数値表現と内容 (HSC2 手順)

項番	テキスト形態	数値表現 (16 進数字)	内容	注意事項
1.	通常テキスト	00000002	通常テキスト送信	-
		00010002	通常テキスト送信後, EOT を送信	ブロック分割をする場合, 最終テキストで指定。
2.	透過モードテキスト	00000004	透過モードテキスト送信	-
		00010004	透過モードテキスト送信後, EOT を送信	ブロック分割をする場合, 最終テキストで指定。
3.	EOT 送信要求テキスト	00010000	EOT 送信要求	単独テキストで指定。受信モードの場合, 指定できません。
4.	回線切断要求テキスト	00020000	回線切断要求 (DLE・EOT)	単独テキストで指定。
5.	回線接続要求テキスト	00030000	回線接続要求	単独テキストで指定。

(凡例)

- : ありません。

注

非同期モードで使用できるテキスト形態です。

表 3-4 受信するテキスト形態の数値表現と内容 (HSC2 手順)

項番	テキスト形態	数値表現 (16 進数字)	内容
1.	通常テキスト	00000002	通常テキスト受信
2.	透過モードテキスト	00000004	透過モードテキスト受信
3.	EOT 受信テキスト	00010000	EOT 受信
4.	回線切断要求テキスト	00020000	回線の切断
5.	接続情報通知テキスト	00040000	発信による回線接続
		00050000	着信による回線接続

注

非同期モードで使用できるテキスト形態です。

3.2 HSC2 手順の AP 間通信メッセージの送受信

TP1/NET/HSC は、AP 間通信メッセージの送受信をします。TP1/NET/HSC で使用するメッセージは一方送信メッセージです。一方送信メッセージには、優先と一般があります。また、メッセージの送信と受信にはそれぞれ、単一セグメント、複数セグメント、および列信の場合があります。列信の場合は、優先の一方送信メッセージは使用しません。

ここでは、コネクションの確立、回線の接続が完了したあとの AP 間通信のメッセージの流れを説明します。

HSC2 手順の場合、回線が接続している間、次のどれかの状態になっています。ただし、HSC2 手順（非同期モード）は回線維持モードだけです。

- 送信モード
発信によって回線を接続した場合にこのモードになります。送信モードの場合、`sendsync` 関数でメッセージを送信する必要があります。
- 受信モード
着信によって回線を接続した場合にこのモードになります。受信モードの場合、`recvsync` 関数でメッセージを受信する必要があります。
- 回線維持モード
回線が接続されていて、送信も受信もできる状態です。また、回線が切断されている状態を、回線切断モードといいます。

3.2.1 メッセージの送信

HSC2 手順の TP1/NET/HSC が、相手システムにメッセージを送信した場合の処理の流れを説明します。

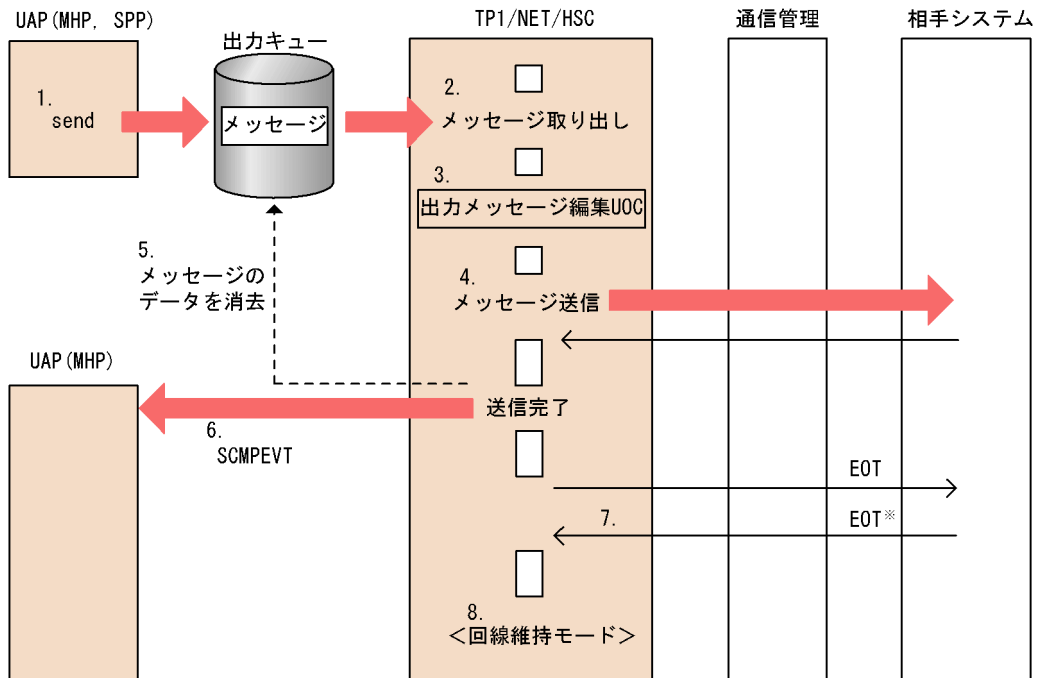
単一セグメントの場合のメッセージ送信の処理の流れを、非同期モードと同期モードの場合に分けて説明します。

(1) 非同期モードのメッセージの送信

非同期モードの場合、列信でメッセージを送信するには、MCF 通信構成定義 (`mcftalcle-w`) の `rowtrans` オペランドで `use` を指定しておく必要があります。MCF 通信構成定義については、6 章の「TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順（非同期モード）」を参照してください。

非同期モードの場合のメッセージ送信の処理の流れを、次の図に示します。

図 3-11 HSC2 手順 (非同期モード) のメッセージ送信の流れ



注

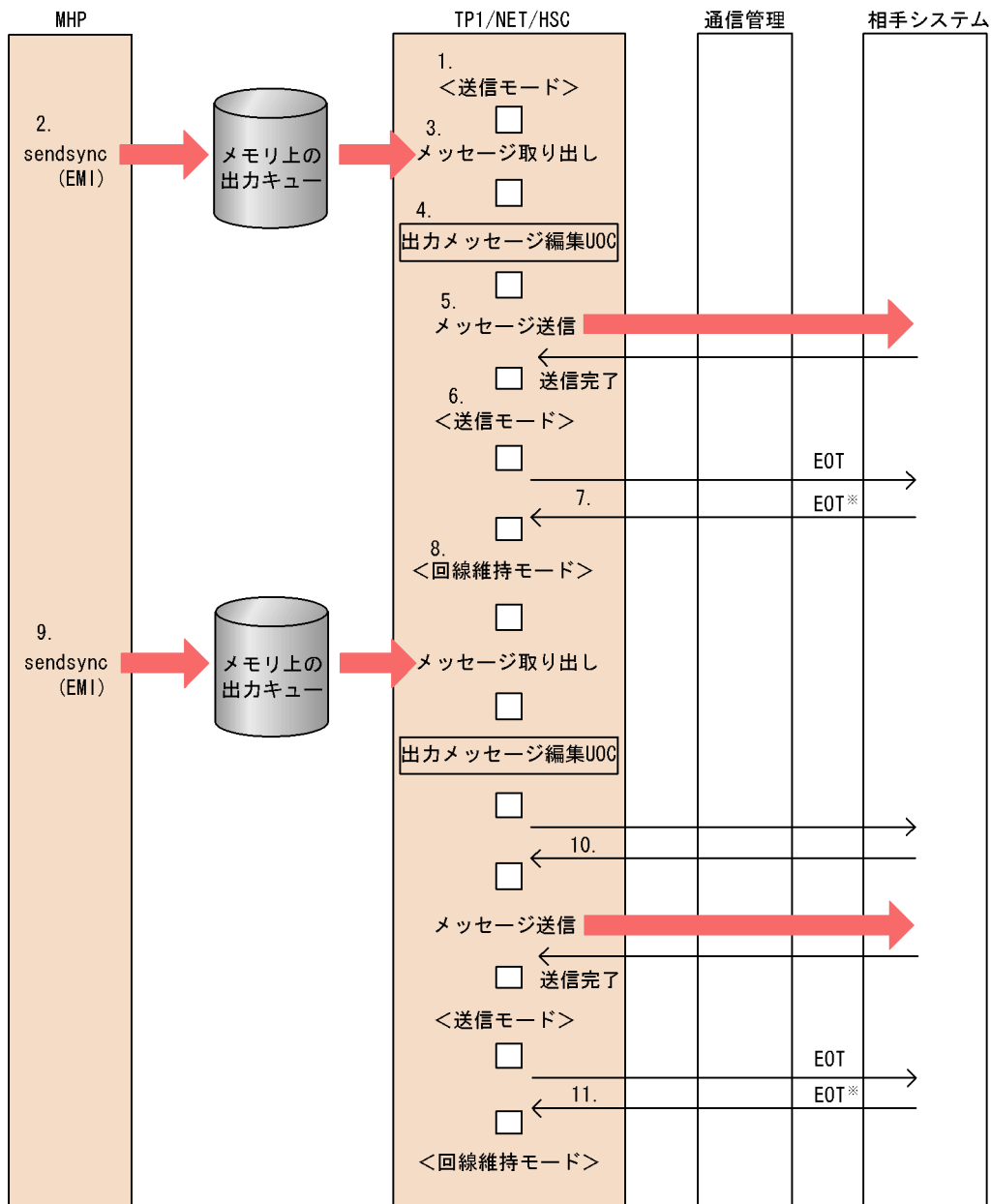
MCF 通信構成定義で EOT 交換をしないことを指定した場合、この EOT は送信されません。EOT 交換については、「3.6.1 非同期モードの EOT 交換」を参照してください。

- MHP または SPP から、メッセージ送信の send 関数を呼び出します。send 関数については、「4. メッセージ送受信インタフェース」を参照してください。
- TP1/NET/HSC 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。
- 出力メッセージ編集 UOC でメッセージを編集します。UOC が登録されていない場合、UOC を呼び出さないで次の処理へ続きます。
- 相手システムへメッセージを送信します。
- 相手システムへの送信が完了した時点で、出力キューのメッセージを消去します。
1. の send 関数で送信完了通知イベントを通知させることを指定した場合、送信完了通知イベント (SCMPEVT) を通知します。また、正常に送信できなかった場合、send 関数の指定によって送信障害通知イベント (SERREVT) を通知します。SERREVT については、「付録 B.2 HSC2 手順 (非同期モード) の障害」を参照してください。
- TP1/NET/HSC は、EOT を送信します。
- TP1/NET/HSC は、回線維持モードになります。

(2) 同期モードのメッセージの送信

同期モードの場合のメッセージ送信の処理の流れを、次の図に示します。

図 3-12 HSC2 手順 (同期モード) のメッセージ送信の流れ



注

MCF 通信構成定義で EOT 交換をしないことを指定した場合、この EOT は送信さ

れません。EOT 交換については、「3.6.2 同期モードの EOT 交換」を参照してください。

1. TP1/NET/HSC は、送信モードです。
2. MHP はメッセージ送信の `sendsync` 関数を呼び出します。`sendsync` 関数については、「4. メッセージ送受信インタフェース」を参照してください。
3. TP1/NET/HSC 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。UOC が登録されていない場合、UOC を呼び出さないで次の処理へ続きます。
5. 相手システムへメッセージを送信します。
6. TP1/NET/HSC は、2. の `sendsync` が EOT なしの場合、送信完了を受け取ると送信モードになります。
7. TP1/NET/HSC は、2. の `sendsync` が EOT ありの場合、回線維持モードになるようにします。
8. TP1/NET/HSC は、回線維持モードになります。
9. MHP はメッセージ送信の `sendsync` 関数を呼び出します。
10. 回線維持モードのときにメッセージ送信の要求をすると、メッセージ送信ができるようにします。
11. TP1/NET/HSC は、9. の `sendsync` が EOT ありの場合、回線維持モードになるようにします。9. の `sendsync` が EOT なしの場合、送信モードになります。

3.2.2 メッセージの受信

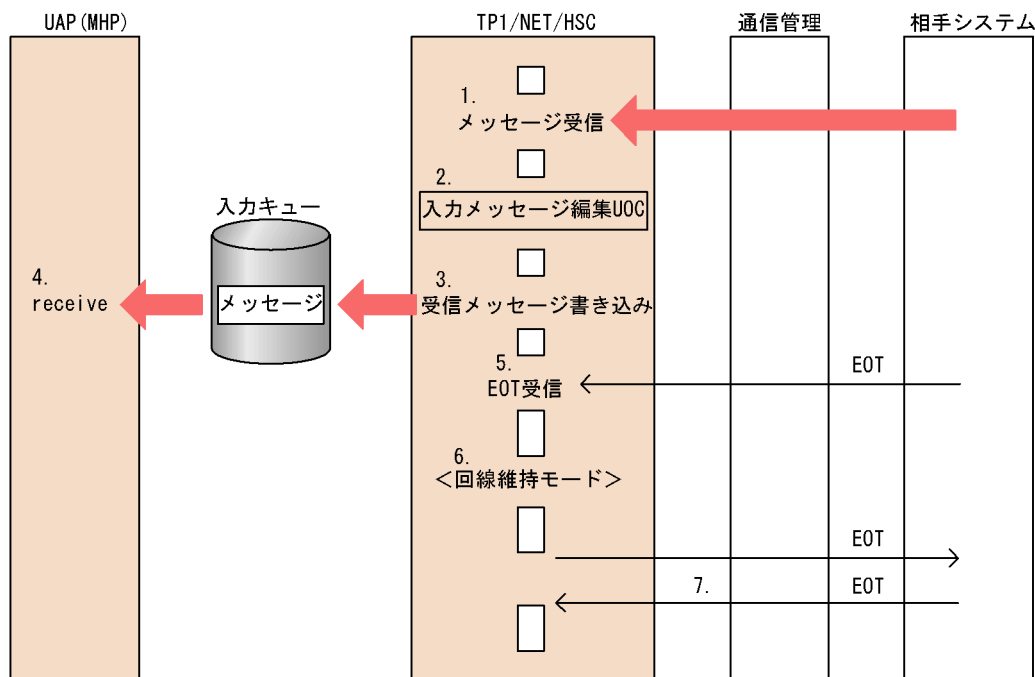
HSC2 手順の TP1/NET/HSC が、相手システムからのメッセージを受信した場合の処理の流れを説明します。

単一セグメントの場合のメッセージ受信の処理の流れを、非同期モードと同期モードの場合に分けて説明します。

(1) 非同期モードのメッセージの受信

非同期モードの場合のメッセージ受信の処理の流れを、次の図に示します。

図 3-13 HSC2 手順（非同期モード）のメッセージ受信の流れ

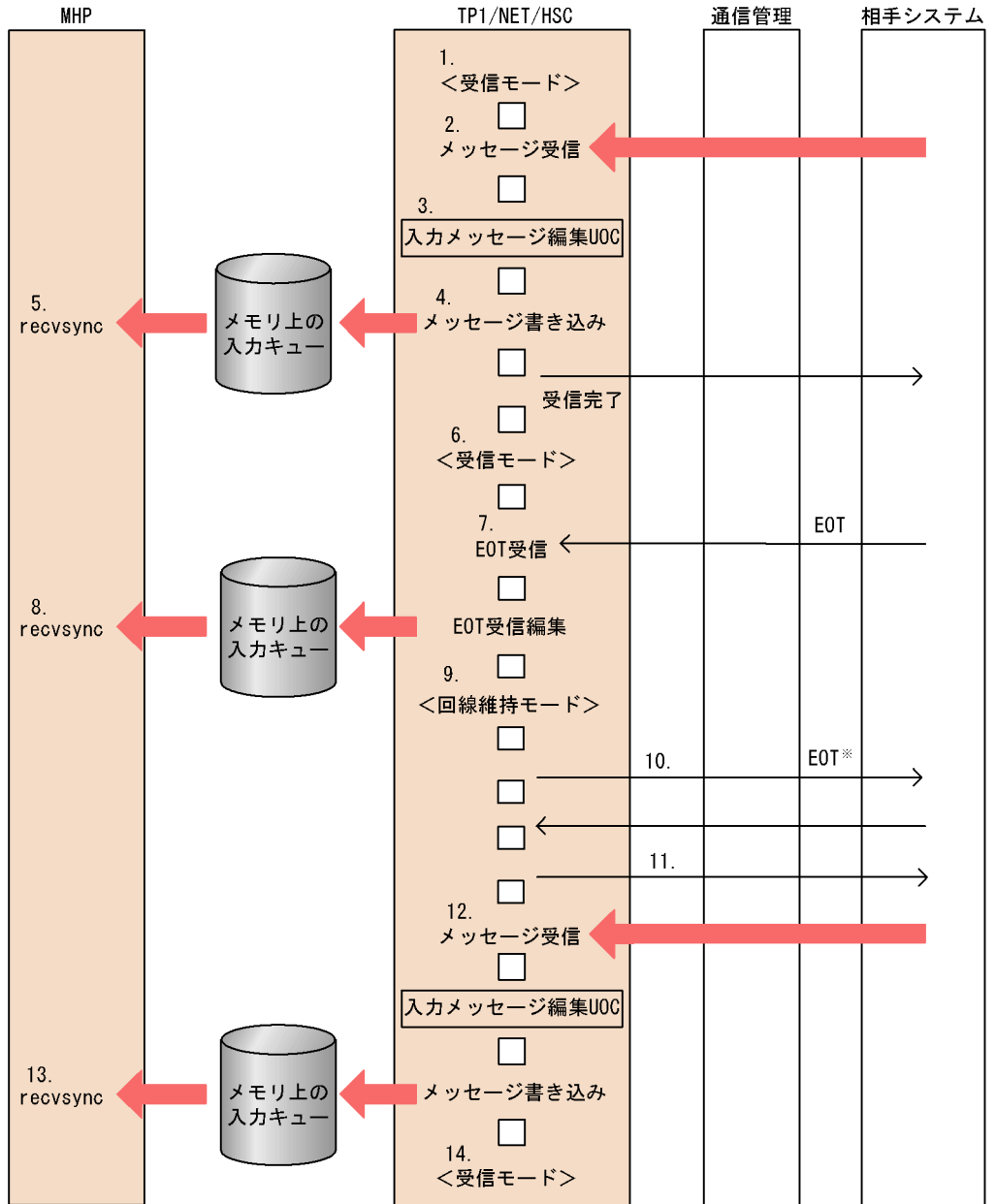


1. TP1/NET/HSC は、相手システムからメッセージを受信します。
2. 入力メッセージ編集 UOC でメッセージを編集します。UOC が登録されていない場合、UOC を呼び出さずに次の処理へ続きます。
3. 受信したメッセージを入力キューに書き込みます。
4. MHP は、メッセージ受信の receive 関数を呼び出してメッセージを受信します。receive 関数については、「4. メッセージ送受信インタフェース」を参照してください。
5. TP1/NET/HSC は、相手システムへから EOT を受信します。
6. TP1/NET/HSC は、回線維持モードになります。
7. MCF 通信構成定義で EOT 交換をすることを指定した場合、EOT 交換をして回線の維持をします。EOT 交換については、「3.6.1 非同期モードの EOT 交換」を参照してください。

(2) 同期モードのメッセージの受信

同期モードの場合のメッセージ受信の処理の流れを、次の図に示します。

図 3-14 HSC2 手順（同期モード）のメッセージ受信の流れ



注

MCF 通信構成定義で EOT 交換をしないことを指定した場合、この EOT は送信しません。EOT 交換については、「3.6.2 同期モードの EOT 交換」を参照してください。

1. TP1/NET/HSC は、受信モードです。

3. HSC2 手順の機能

2. TP1/NET/HSC は、相手システムからメッセージを受信します。
3. 入力メッセージ編集 UOC でメッセージを編集します。UOC が登録されていない場合、UOC を呼び出さないで次の処理へ続きます。
4. 受信したメッセージを入力キューに書き込みます。
5. MHP は、メッセージ受信の recvsync 関数を呼び出してメッセージを受信します。recvsync 関数については、「4. メッセージ送受信インタフェース」を参照してください。
6. TP1/NET/HSC は、相手システムに受信完了を送信し、受信モードになります。
7. TP1/NET/HSC は、相手システムから EOT を受信します。
8. MHP は、recvsync 関数を呼び出して、編集された EOT を受信します。
9. TP1/NET/HSC は、回線維持モードになります。
10. MCF 通信構成定義で EOT 交換をすることを指定した場合、EOT 交換をします。EOT 交換については、「3.6.2 同期モードの EOT 交換」を参照してください。
11. 相手システムからのメッセージ送信要求で、メッセージの受信ができるようにします。
12. TP1/NET/HSC は、受信モードになり、メッセージを受信します。
13. MHP は、recvsync 関数を呼び出して、メッセージを受信します。
14. TP1/NET/HSC は、受信モードになります。

3.3 HSC2 手順の機能

TP1/NET/HSC は、HSC2 手順のオプション機能のうち、次の表に示す範囲をサポートします。

表 3-5 HSC2 手順の機能と TP1/NET/HSC の扱い

HSC2 手順の機能	使用の可否	TP1/NET/HSC の扱い
ブロック転送		<ul style="list-style-type: none"> UAP からの send または sendsync (ESI) で送信要求された 1 セグメントを 1 ブロックとして相手システムに送信します。同期モードの場合、メッセージ送信は、sendsync (EMI) の送信完了で、相手システムとの同期がとられます。 相手システムから受信したメッセージテキストは、最終セグメントの受信まで TP1/NET/HSC で保留し、最終セグメントの受信で UAP を起動します。ユーザは、各ブロックを 1 セグメントとして受け取れます。
列信		<p>非同期モードの場合、MCF 通信構成定義 (mcftalcle -w) で指定する値によって TP1/NET/HSC で制御します。</p> <p>同期モードの場合、同期送受信によって UAP で制御します。</p>
逆中断 (RVI)		送信、受信の両方をサポートします。受信の場合は、MCF 通信構成定義 (mcftalccn -u) で受理するかどうかを指定できます。
一時送信延期 (TTD)		送信、受信の両方をサポートします。
一時受信抑止 (WACK)		送信の場合は、MCF 通信構成定義 (mcftalccn -B) でサポートするかどうかを指定できます。
通常テキスト		送信、受信の両方をサポートします。
ヘッダ付きテキスト	×	サポートしません。
透過モードテキスト		送信、受信の両方をサポートします。
ヘッダ付き透過モードテキスト	×	サポートしません。
コンテンツンション制御		MCF 通信構成定義 (mcftalccn -y) で優先局、非優先局を指定できます。
制御付き会話	×	サポートしません。
EOT 交換		EOT 交換をして回線接続を維持します。
HSC2 呼制御		発信と着信をサポートします。ID 交換をする場合、一度確定した ID の切り替えは回線の再接続が必要です。

(凡例)

： TP1/NET/HSC の HSC2 手順で使用できます。

×： TP1/NET/HSC の HSC2 手順では使用できません。

3.4 回線リスト定義ユティリティ（非同期モード）

TP1/NET/HSC は、HSC2 手順（非同期モード）の場合、回線リスト定義をあらかじめ指定しておく必要があります。

回線リストは、コネクションが管理する回線を指定しているリストです。回線リスト定義ユティリティは、回線リストファイルを作成するユティリティです。

回線リストファイルの作成順序を次に示します。

1. システム定義時に回線リスト定義に必要な項目を指定します。
回線リスト定義については、6 章の「TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順（非同期モード）」を参照してください。
2. 1. で指定した内容を、回線リスト定義ユティリティのコマンドを使用してファイルを作成します。ファイルは、\$DCCONFPATH のディレクトリ下に作成してください。
回線リスト定義ユティリティのコマンドについては、6 章の「回線リスト定義 - HSC2 手順（非同期モード）」を参照してください。

3.5 相手ターミナル ID リスト定義ユティリティ（同期モード）

TP1/NET/HSC は、HSC2 手順（同期モード）の場合、コネクションの確立時に、相手ターミナル ID リストファイルを取り込み、回線接続時に、相手システムとの ID を交換してメッセージの送受信を始めます。相手ターミナル ID リストファイルには、発信用と着信用があります。相手ターミナル ID リスト定義ユティリティは、相手ターミナル ID リストファイルを作成するユティリティです。

相手ターミナル ID リストファイルの作成順序を次に示します。

1. システム定義時に、発信用または着信用相手ターミナル ID リストの定義に必要な項目を指定します。
相手ターミナル ID リストの定義については、6 章の「TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順（同期モード）」を参照してください。
2. 1. で指定した内容を、相手ターミナル ID リスト定義ユティリティのコマンドを使用してファイルを作成します。ファイルは、\$DCCONFPATH のディレクトリ下に作成してください。
相手ターミナル ID リスト定義ユティリティのコマンドについては、6 章の「相手ターミナル ID リスト定義 - HSC2 手順（同期モード）の場合」を参照してください。

3.6 EOT 交換

HSC2 手順では、EOT 交換をして相手システムとの回線の接続状態を維持します。また、EOT 交換をしないで、時間監視で回線を維持する方法もあります。

3.6.1 非同期モードの EOT 交換

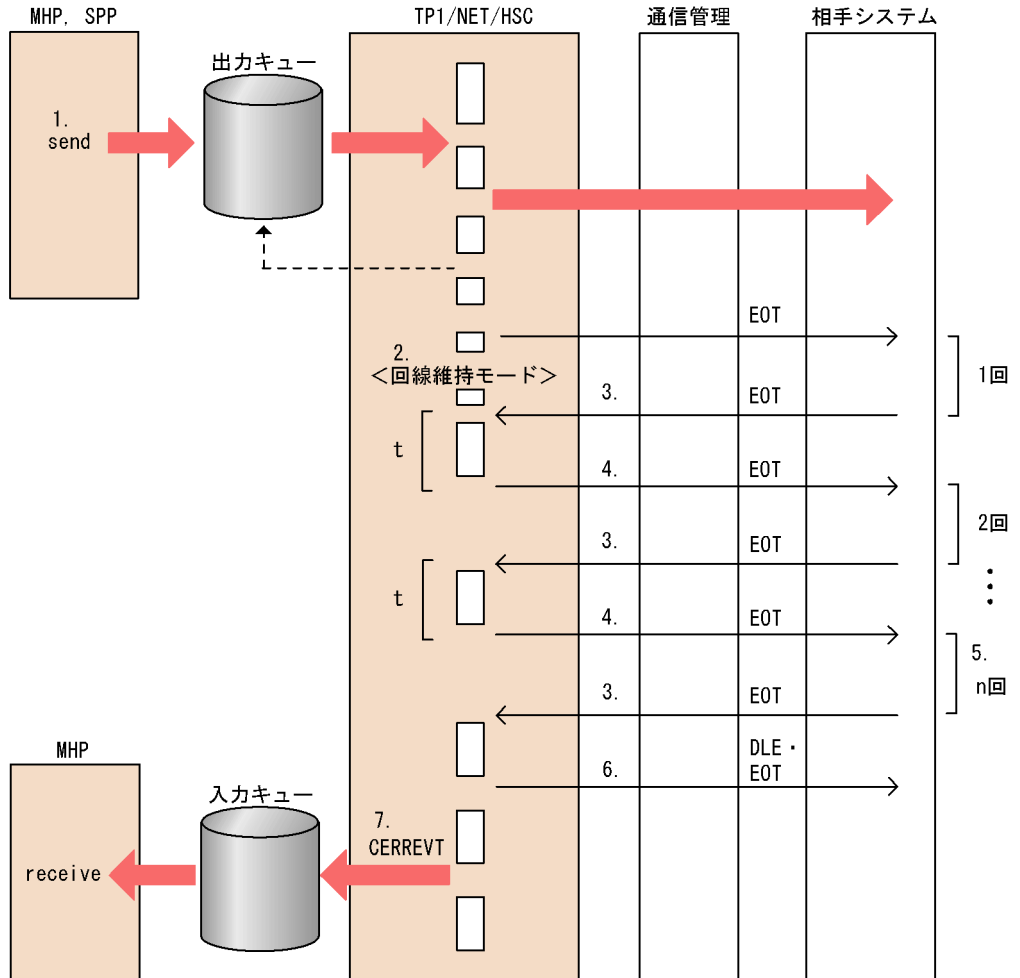
非同期モードの場合に回線を維持するための EOT 交換について説明します。

(1) EOT 交換

HSC2 手順（非同期モード）では、回線維持モードになったときに EOT 交換をして、相手システムとの回線接続状態を維持します。EOT 交換は、相手システムと EOT の送受信をすることで、回線接続状態を確認することです。MCF 通信構成定義（`mcftalcen -f`）の `eotctrl` オペランドで `use` を指定するか、またはこのオペランドを省略すると EOT 交換をします。EOT 交換中に送受信メッセージが発生すると、EOT 交換を中断し、送受信処理をします。また、EOT 交換の結果で回線が切断されると、MCF 通信構成定義（`mcftalcen -v`）の `discinf` オペランドを `use` に指定することで、回線維持終了通知として障害通知イベント（`CERREVT`）を通知します。MCF 通信構成定義については、6 章の「TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順（非同期モード）」を参照してください。

非同期モードの場合の EOT 交換の処理の流れを、メッセージ送信後と受信後に分けて、図 3-15 および図 3-16 に示します。

図 3-15 非同期モードのメッセージ送信後の EOT 交換



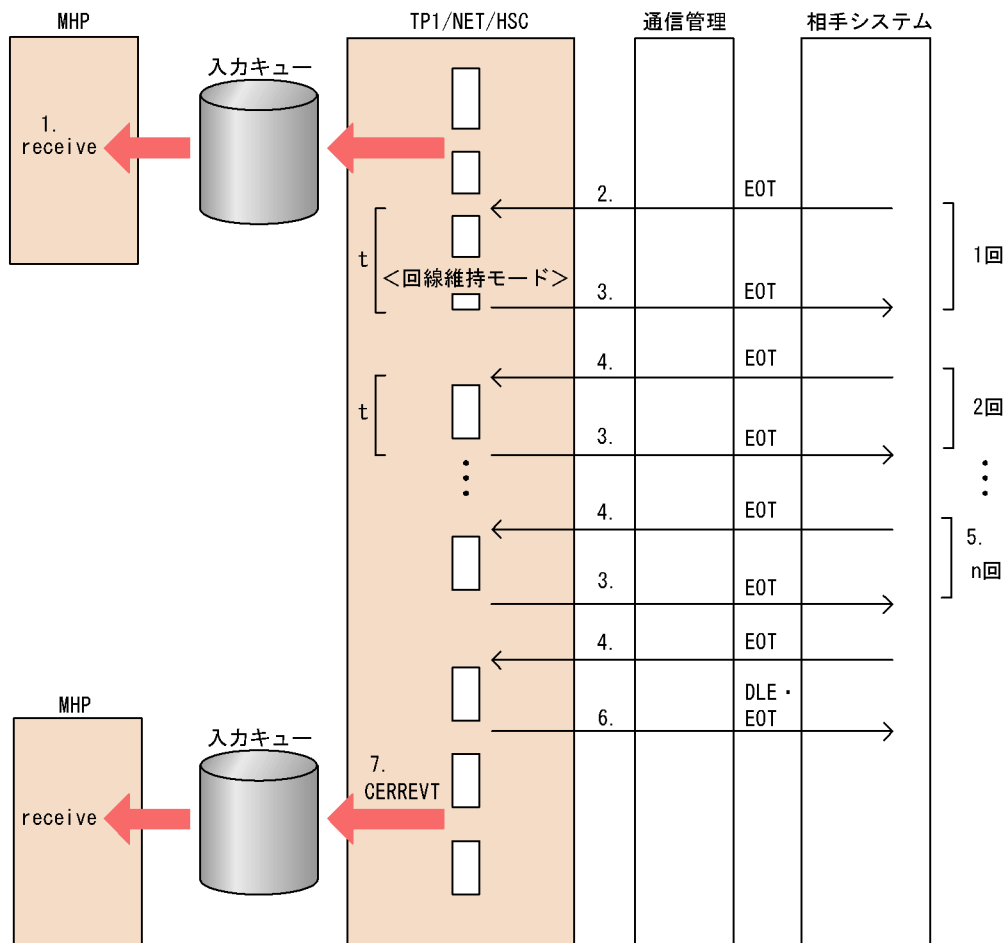
(凡例)

t: EOT交換時間 (MCF通信構成定義 mcftalccn -f eottime=で指定)

1. 自システムの MHP または SPP から、send 関数を呼び出し、相手システムにメッセージを送信します。
2. TP1/NET/HSC は、相手システムに EOT を送信し、回線維持モードになります。
3. 相手システムから回線の維持をする EOT を受信します。
4. TP1/NET/HSC から回線の維持をする EOT を相手システムに送ります。
5. MCF 通信構成定義 (mcftalccn -f) の eotexl または eotexr オペランドで指定した回数の EOT 交換をします。
6. 指定した回数の EOT 交換が終了すると、回線を切断します。
7. MCF 通信構成定義 (mcftalccn -v) の discinf オペランドで use を指定すると、回線維持状態が終了し、回線が切断されたことを示す障害通知イベント (CERREVT) を

通知します。

図 3-16 非同期モードのメッセージ受信後の EOT 交換



(凡例)

t: EOT交換時間 (MCF通信構成定義 mcftalccn -f eottime=で指定)

1. MHP は、receive 関数を呼び出して、相手システムからのメッセージを受信します。
2. 相手システムから、送信権放棄の EOT を受信します。TP1/NET/HSC は、回線維持モードになります。
3. TP1/NET/HSC から回線の維持をする EOT を相手システムに送ります。
4. 相手システムから回線の維持をする EOT を受信します。
5. MCF 通信構成定義 (mcftalccn -f) の eotexl または eotexr オペランドで指定した回数の EOT 交換をします。
6. 指定した回数の EOT 交換が終了すると、回線を切断します。
7. MCF 通信構成定義 (mcftalccn -v) の discinf オペランドで use を指定すると、回線

維持状態が終了し、回線が切断されたことを示す障害通知イベント (CERREVT) を通知します。

(2) EOT 交換の抑止

HSC2 手順 (非同期モード) は、MCF 通信構成定義 (mcftalccn -f) の eotctrl オペランドで nouse を指定すると、EOT 交換をしないで、時間監視をして回線を維持します。このとき、MCF 通信構成定義 (mcftalccn -f) の holdtime オペランドで指定した時間だけ、相手システムとの回線接続状態を維持します。指定した時間にメッセージの送受信がない場合は、接続中の回線を切断します。このとき、MCF 通信構成定義 (mcftalccn -v) の discinf オペランドで use を指定すると、回線維持終了通知として障害通知イベント (CERREVT) を通知します。また、メッセージ送受信があった場合は、時間監視を中止し、通常のメッセージの送受信をします。MCF 通信構成定義については、6 章の「TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (非同期モード)」を参照してください。

3.6.2 同期モードの EOT 交換

非同期モードの場合に回線を維持するための EOT 交換について説明します。

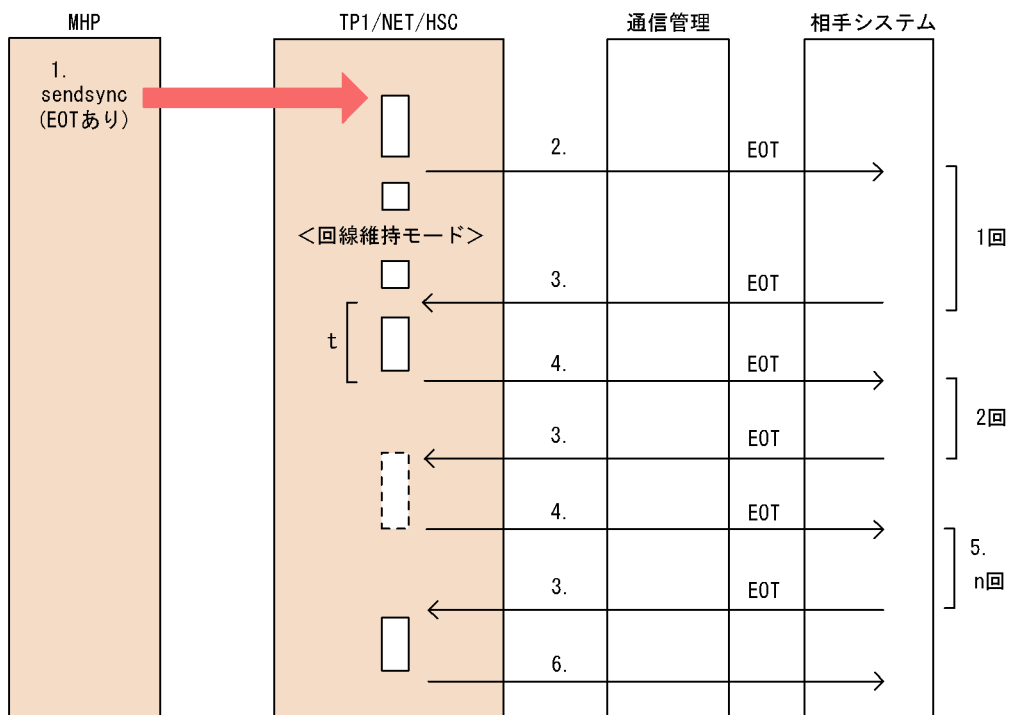
(1) EOT 交換

HSC2 手順 (同期モード) では、回線維持モードになったときに EOT 交換をして、相手システムとの回線接続状態を維持します。EOT 交換は、相手システムと EOT の送受信をすることで、回線接続状態を確認することです。EOT 交換中に送受信メッセージが発生すると、EOT 交換を中断し、送信モードまたは受信モードになります。MCF 通信構成定義 (mcftalccn -f) の eotctrl オペランドで use を指定するか、またはこのオペランドを省略すると EOT 交換をします。MCF 通信構成定義については、6 章の「TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (同期モード)」を参照してください。

同期モードの場合の EOT 交換の処理の流れを、メッセージ送信後と受信後に分けて、図 3-17 および図 3-18 に示します。

3. HSC2 手順の機能

図 3-17 同期モードのメッセージ送信後の EOT 交換

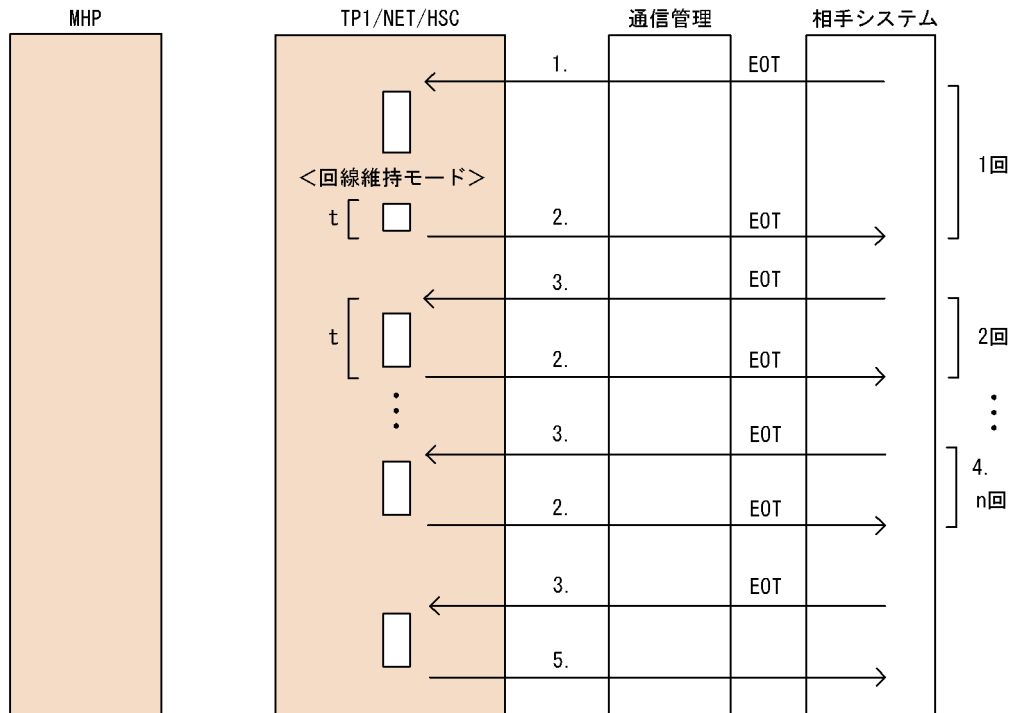


(凡例)

t: EOT交換間隔時間 (MCF通信構成定義 mcftalccn -f eottime=で指定)

1. 自システムの MHP から、関数の終了を示す EOT ありの sendsync 関数を呼び出します。
2. TP1/NET/HSC は、相手システムに EOT を送信し、回線維持モードになります。
3. 相手システムから回線の維持をする EOT を受信します。
4. TP1/NET/HSC から回線の維持をする EOT を相手システムに送ります。
5. MCF 通信構成定義 (mcftalccn -f) の eotexl または eotextr オペランドで指定した回数の EOT 交換をします。
6. 指定した回数の EOT 交換が終了すると、回線を切断します。

図 3-18 同期モードのメッセージ受信後の EOT 交換



(凡例)

t: EOT交換間隔時間 (MCF通信構成定義 `mcftalccn -f eottime=`で指定)

1. 相手システムから、送信権放棄の EOT を受信します。TP1/NET/HSC は、回線維持モードになります。
2. TP1/NET/HSC から回線の維持をする EOT を相手システムに送ります。
3. 相手システムから回線の維持をする EOT を受信します。
4. MCF 通信構成定義 (`mcftalccn -f`) の `eotexl` または `eotexr` オペランドで指定した回数の EOT 交換をします。
5. 指定した回数の EOT 交換が終了すると、回線を切断します。

(2) EOT 交換の抑止

HSC2 手順 (同期モード) は、MCF 通信構成定義 (`mcftalccn -f`) の `eotctrl` オペランドで `nouse` を指定すると、EOT 交換をしないで、時間監視をして回線を維持します。このとき、MCF 通信構成定義 (`mcftalccn -f`) の `holdtime` オペランドで指定した時間だけ、相手システムとの回線接続状態を維持します。指定した時間にメッセージの送受信がない場合は、接続中の回線を正常に切断します。また、メッセージ送受信があった場合は、時間監視を中止し、通常 of メッセージの送受信をします。MCF 通信構成定義については、6 章の「TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (同期モード)」を参照してください。

4

メッセージ送受信インタフェース

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

メッセージ送受信インタフェースの一覧

dc_mcf_receive - メッセージまたは接続情報通知の受信 (C 言語)

dc_mcf_recvsync - 同期型のメッセージの受信 (C 言語)

dc_mcf_resend - メッセージの再送 (C 言語)

dc_mcf_send - 一方送信メッセージまたは回線接続要求の送信 (C 言語)

dc_mcf_sendsync - 同期型のメッセージの送信 (C 言語)

CBLDCMCF('RECEIVE ') - メッセージまたは接続情報通知の受信 (COBOL 言語)

CBLDCMCF('RECVSYNC') - 同期型のメッセージの受信 (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または回線接続要求の送信 (COBOL 言語)

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

RECEIVE - メッセージの受信 (データ操作言語)

4. メッセージ送受信インタフェース

SEND - メッセージの送信（データ操作言語）

ユーザアプリケーションプログラム作成例

メッセージ送受信インタフェースの一覧

TP1/NET/HSC 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。

TP1/NET/HSC では、手順によって使用できる関数の種類と、使用する関数の意味が違います。詳細については、該当する関数の説明を参照してください。

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

C 言語のメッセージ送受信

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

メッセージ送受信の関数を次の表に示します。

表 4-1 メッセージ送受信のライブラリ関数 (C 言語)

関数名	使用できる手順	機能
dc_mcf_receive	HSC1, HSC2 (非同期モード)	メッセージの受信
	HSC2 (同期モード)	接続情報通知の受信
dc_mcf_recvsync	HSC2 (同期モード)	同期型のメッセージの受信
dc_mcf_resend	HSC1, HSC2 (非同期モード)	メッセージの再送
dc_mcf_send	HSC1, HSC2 (非同期モード)	一方送信メッセージの送信
	HSC2 (同期モード)	回線接続要求の送信
dc_mcf_sendsync	HSC2 (同期モード)	同期型のメッセージの送信

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

COBOL 言語のメッセージ送受信

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

メッセージ送受信の関数に対応するプログラムを次の表に示します。

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

表 4-2 メッセージ送受信の通信文 (COBOL 言語)

プログラム名	データ名	使用できる手順	機能
CBLDCMCF	'RECEIVE '	HSC1, HSC2 (非同期モード)	メッセージの受信
		HSC2 (同期モード)	接続情報通知の受信
	'RECVSYNC'	HSC2 (同期モード)	同期型のメッセージの受信
	'RESEND '	HSC1, HSC2 (非同期モード)	メッセージの再送
	'SEND '	HSC1, HSC2 (非同期モード)	一方送信メッセージの送信
		HSC2 (同期モード)	回線接続要求の送信
'SENDSYNC'	HSC2 (同期モード)	同期型のメッセージの送信	

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

データ操作言語 (COBOL 言語) のメッセージ送受信

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

TP1/NET/HSC で固有の通信文について、次の表に示します。

表 4-3 メッセージ送受信の通信文 (データ操作言語)

通信文	使用できる手順	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	HSC1, HSC2 (非同期モード)	メッセージの受信	CBLDCMCF('RECEIVE')
		HSC2 (同期モード)	接続情報通知の受信	
		HSC2 (同期モード)	同期型のメッセージの受信	CBLDCMCF('RECVSYNC')
	SEND	HSC1, HSC2 (非同期モード)	一方送信メッセージの送信	CBLDCMCF('SEND')
		HSC2 (同期モード)	回線接続要求の送信	
		HSC2 (同期モード)	同期型のメッセージの送信	CBLDCMCF('SENDSYNC')

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

dc_mcf_receive - メッセージまたは接続情報通知の受信 (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

HSC1 手順または HSC2 手順 (非同期モード) の場合、相手システムから送信されたメッセージを受信します。

HSC2 手順 (同期モード) の場合、相手システムからの接続情報通知を受信します。

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。なお、接続情報通知は単一セグメントです。

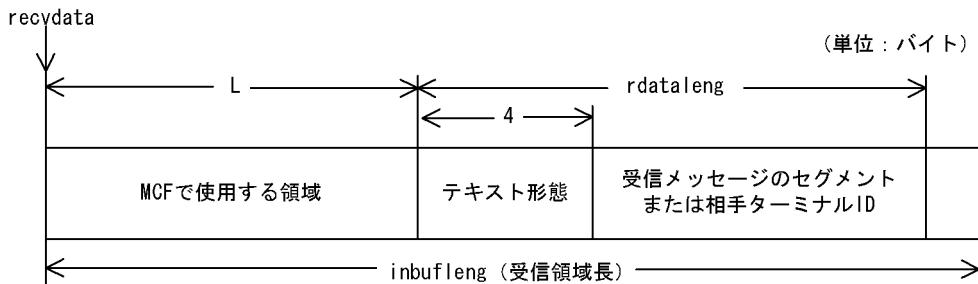
dc_mcf_receive 関数で受信できるメッセージの種類を次に示します。

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

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

4. メッセージ送受信インタフェース

dc_mcf_receive - メッセージまたは接続情報通知の受信 (C 言語)



UAP で値を設定する引数

action

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

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

DCMCFFRST

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

接続情報通知を受信する場合も、DCMCFFRST を設定します。

DCMCFSEG

中間セグメントまたは最終セグメントを受信する場合に設定します。

DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

commform

DCNOFLAGS を設定します。

termnam

先頭セグメントを受信する場合は、入力元の論理端末名称が返される領域を設定します。dc_mcf_receive 関数が終了すると、入力元の論理端末名称が返されます。論理端末名称の長さは最大 8 バイトです。論理端末名称の後ろにはヌル文字が付けられます。

中間セグメントまたは最終セグメントを受信する場合は、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、termnam には OpenTP1 から値が返されます。

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。受信できるセグメントの長さの最大値は、通信管理の受信する最大セグメント長にテキスト形態の 4 バイトを加えた値です。

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

inbufleng

セグメントを受信する領域の長さを設定します。

OpenTP1 から値が返される引数

termnam

先頭セグメントを受信する場合だけ、入力元の論理端末名称が返されます。

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

recvdata

次に示す内容が返されます。

【HSC1 手順または HSC2 手順 (非同期モード) の場合】

テキスト形態と受信したセグメントの内容

【HSC2 手順 (同期モード) で ID を交換する場合】

テキスト形態と相手ターミナル ID

【HSC2 手順 (同期モード) で ID を交換しない場合】

テキスト形態

rdatale

recvdata で返される内容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。

4. メッセージ送受信インタフェース

dc_mcf_receive - メッセージまたは接続情報通知の受信 (C 言語)

リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71000	先頭セグメントを受信する dc_mcf_receive 関数を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、action に DCMCFSEG を設定して dc_mcf_receive 関数を呼び出してください。
DCMCFRTN_71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する dc_mcf_receive 関数を呼び出しています。直前に呼び出した dc_mcf_receive 関数でメッセージはすべて受信しました。このリターン値が返されたあとに、再び dc_mcf_receive 関数を呼び出した場合は、リターン値 DCMCFRTN_72000 が返されます。
DCMCFRTN_71002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。
DCMCFRTN_72000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none"> 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、中間セグメントまたは最終セグメントを受信する dc_mcf_receive 関数を呼び出しています。先頭セグメントを受信する場合は、action に DCMCFRST を設定して dc_mcf_receive 関数を呼び出してください。 リターン値 DCMCFRTN_71001 が返されたあとに、再び dc_mcf_receive 関数を呼び出しています。 < SPP の実行でリターンした場合 > SPP では dc_mcf_receive 関数を呼び出せません。
DCMCFRTN_72001	termnam に設定した論理端末名称が間違っています。
DCMCFRTN_72013	inbufleng の指定値を超えるセグメントを受信しました。inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRTN_72016	action に設定した値が間違っています。 resv01 に設定した値が間違っています。 引数に設定した値に間違いがあります。
DCMCFRTN_72024	commform に設定した値が間違っています。
DCMCFRTN_72025	action に設定したセグメント種別 (DCMCFRST または DCMCFSEG) の値が間違っています。
DCMCFRTN_72036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

dc_mcf_recvsync - 同期型のメッセージの受信 (C 言語)

形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

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

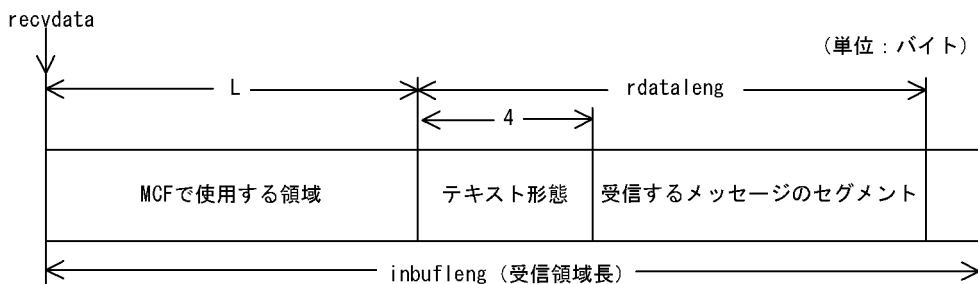
機能

TP1/NET/HSC では、HSC2 手順 (同期モード) のメッセージの受信に使用します。

相手システムから届いた同期型の受信メッセージのうち、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_recvsync 関数を呼び出すと、一つの論理メッセージを受信できます。

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

4. メッセージ送受信インタフェース
 dc_mcf_recvsync - 同期型のメッセージの受信 (C 言語)



UAP で値を設定する引数

action

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

{DCMCFFRST|DCMCFSEG} { | {DCMCFBUF1|DCMCFBUF2} }

DCMCFFRST

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

DCMCFSEG

中間セグメントまたは最終セグメントを受信する場合に設定します。

DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

commform

DCNOFLAGS を設定します。

termnam

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

resv01

ヌル文字を設定します。

recvdata

セグメントを受信する領域を設定します。受信できるセグメントの長さの最大値は、通信管理の受信する最大セグメント長にテキスト形態の 4 バイトを加えた値です。

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

inbufleng

セグメントを受信する領域の長さを設定します。

watchtime

この引数の設定は無効です。0 を設定しておいてください。また、MCF マネージャ定義の UAP 共通定義で同期受信監視時間を指定しても、時間監視をしません。

OpenTP1 から値が返される引数

recvdata

テキスト形態と受信したセグメントの内容が返されます。

rdataleng

テキスト形態と受信したセグメントの長さが返されます。

time

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

リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71001	メッセージの最終セグメントを受信したあとで、その次のセグメントを受信する dc_mcf_recvsync 関数を呼び出しています。直前に呼び出した dc_mcf_recvsync 関数でそのメッセージはすべて受信しました。このリターン値が返されたあとに、再び次のセグメントを受信する dc_mcf_recvsync 関数を呼び出した場合は、リターン値 DCMCFRTN_72000 が返されます。
DCMCFRTN_71002	MCF が終了処理中のため、メッセージの受信を受け付けられません。
DCMCFRTN_71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRTN_71108	メッセージ受信に必要な管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_recvsync 関数を呼び出しています。

4. メッセージ送受信インタフェース

dc_mcf_recvsync - 同期型のメッセージの受信 (C 言語)

リターン値	意味
	リターン値 DCMCFRTN_71001 が返されたあとに、再び dc_mcf_recvsync 関数を呼び出しています。
DCMCFRTN_72001	termnam に設定した論理端末名称が間違っています。 dc_mcf_recvsync 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72013	inbufleng の指定値を超えるセグメントを受信しました。inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRTN_72016	action に設定した値が間違っています。 resv01 に設定した値が間違っています。 引数に設定した値に間違いがあります。
DCMCFRTN_72024	commform に設定した値が間違っています。
DCMCFRTN_72025	action に設定したセグメント種別 (DCMCFRST または DCMCFSEG) の値が間違っています。
DCMCFRTN_72036	inbufleng の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
DCMCFRTN_73001	入力元の論理端末で障害が発生しました。
DCMCFRTN_73002	dc_mcf_recvsync 関数の処理中に、入力元の論理端末と MCF との間のコネクションまたは経路が切り離されました。 dc_mcf_recvsync 関数が呼び出されるまでに、入力元の論理端末と MCF との間のコネクションまたは経路が切り離されました。
DCMCFRTN_73008	入力元の論理端末は、次のコマンドで停止しています。 • mftdctle • mftdcten
DCMCFRTN_73015	メッセージ入力元の論理端末は、ほかの UAP で仕掛り中です。 メッセージ入力元の論理端末は、該当する UAP と仕掛り中ではありません。
DCMCFRTN_73018	watchtime に設定した値が間違っています。
DCMCFRTN_73020	設定した論理端末は停止しています。
DCMCFRTN_77900	同期受信要求の保留中に、回線維持モードが終了しました。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

dc_mcf_resend - メッセージの再送 (C 言語)

形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

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

機能

HSC1 手順または HSC2 手順 (非同期モード) の場合に使用できます。

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

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

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

UAP で値を設定する引数

action

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

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

DCMCFSEQ

再送するメッセージに出力通番を付け直す場合に設定します。

DCMCFNSEQ

再送するメッセージに出力通番を付け直さない場合に設定します。

DCMCFNORM

一般の一方送信メッセージとして再送する場合に設定します。

DCMCFPRIO

優先の一方送信メッセージとして再送する場合に設定します。

DCMCFLAST

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

commform

一方送信を示す、DCMCFOUT を設定します。

rtermnam

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

resv01

ヌル文字を設定します。

oseqid

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

DCMCFRID_NORM

一般の一方送信メッセージを対象とする場合に設定します。

DCMCFRID_PRIO

優先の一方送信メッセージを対象とする場合に設定します。

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

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

orgseq

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

otermnam

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

resv02, resv03, resv04

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

リターン値

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

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

注意事項

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

-e オプション

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

-l オプション

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

dc_mcf_send - 一方送信メッセージまたは回線接続要求の送信 (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

HSC1 手順または HSC2 手順 (非同期モード) の場合、相手システムへメッセージを送信します。

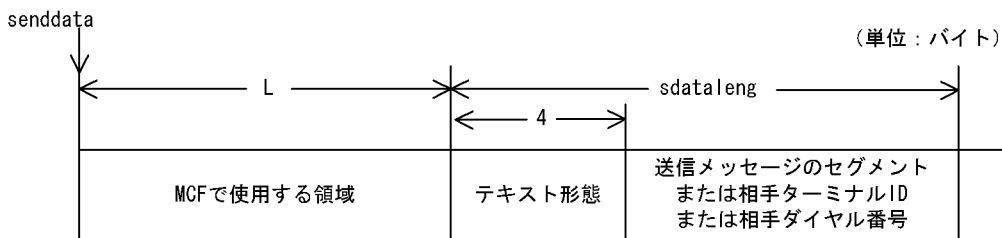
HSC2 手順 (同期モード) の場合、相手システムへ回線接続要求を送信します。

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。なお、回線接続要求は単一セグメントです。

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

4. 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または回線接続要求の送信 (C 言語)



UAP で値を設定する引数

action

メッセージの最終セグメントを送信するかどうか、優先か一般か、出力通番を付けるかどうか、使用するバッファ形式、および送信完了通知イベント・送信障害通知イベントを通知させるかどうかを、次の形式で設定します。

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

DCMCFESI

先頭セグメントまたは中間セグメントを送信する場合に設定します。

DCMCFEMI

最終セグメントを送信する場合に設定します。

メッセージが単一セグメントの場合も、DCMCFEMI を設定します。

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

回線接続要求を送信する場合にも、DCMCFEMI を設定します。

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。

DCMCFSEVT

送信完了通知イベントおよび送信障害通知イベントを通知させるときに設定します。ただし、これらのイベントを処理する MHP を MCF アプリケーション定義で指定していない場合は無効です。

メッセージが複数のセグメントで構成されている場合、先頭セグメントの送信時の指定が有効になります。

commform

一方送信を示す、DCMCFOUT を設定します。

termnam

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

resv01

ヌル文字を設定します。

senddata

次に示す内容を設定した領域を設定します。

【HSC1 手順または HSC2 手順 (非同期モード) の場合】

テキスト形態と送信するセグメントの内容

【HSC2 手順 (同期モード) で ID を交換する場合】

テキスト形態と相手ターミナル ID

【HSC2 手順 (同期モード) で ID を交換しない場合】

テキスト形態と相手ダイヤル番号

送信できるセグメントの長さの最大値は、通信管理が送信する最大セグメント長にテキスト形態の 4 バイトを加えた値です。

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

senddata で設定する内容の長さを設定します。

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

resv02

ヌル文字を設定します。

4. 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または回線接続要求の送信 (C 言語)

opcd

DCNOFLAGS を設定します。

リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	単一セグメント送信時、sdatale ^{ng} に 32000 バイトを超える値を設定していません。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
DCMCFRTN_71003	メッセージキューが満杯です。
DCMCFRTN_71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRTN_71108	メッセージを送信しようとしたますが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_send 関数を呼び出しています。
	< SPP の実行でリターンした場合 > トランザクションでない SPP の処理から、dc_mcf_send 関数を呼び出しています。
DCMCFRTN_72001	termnam に設定した論理端末名称が間違っています。
	dc_mcf_send 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72005	先頭セグメントまたは中間セグメントを送信する dc_mcf_send 関数で、長さが 0 バイトのセグメントを送信しています。
DCMCFRTN_72016	action に設定したメッセージ種別 (DCMCFNORM または DCMCFPRIO) の値が間違っています。
	action に設定した値が間違っています。
	opcd に設定した値が間違っています。
	resv01 または resv02 に設定した値が間違っています。
	引数に設定した値に間違いがあります。
DCMCFRTN_72017	action に設定した出力通番の要否 (DCMCFSEQ または DCMCFNSEQ) の値が間違っています。
DCMCFRTN_72024	commform に設定した値が間違っています。
DCMCFRTN_72026	action に設定したセグメント種別 (DCMCFESI または DCMCFEMI) の値が間違っています。

4. 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または回線接続要求の送信 (C 言語)

リターン値	意味
DCMCFRTN_72041	sdataleng に 0 以下の値を設定しています。
上記以外	プログラムの破壊などによる，予期しないエラーが発生しました。

dc_mcf_sendsync - 同期型のメッセージの送信 (C 言語)

形式

ANSI C, C++ の形式

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

K&R 版 C の形式

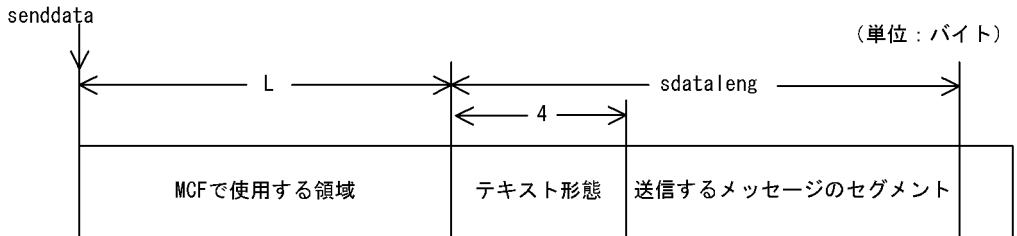
```
#include <dcmcf.h>
int dc_mcf_sendsync (action, commform, termnam, resv01,
                    senddata, sdataleng, resv02, opcd,
                    watchtime)
DCLONG          action;
DCLONG          commform;
char            *termnam;
char            *resv01;
char            *senddata;
DCLONG 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

機能

TP1/NET/HSC では、HSC2 手順 (同期モード) のメッセージの送信に使用します。

相手システムへ送る同期型のメッセージのうち、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_sendsync 関数を呼び出すと、一つの論理メッセージを送信できます。

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



UAP で値を設定する引数

action

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

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

DCMCFESI

先頭セグメントまたは中間セグメントを送信する場合に設定します。

DCMCFEMI

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

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

DCMCFBUF1

バッファ形式 1 を使用する場合に設定します。

DCMCFBUF2

バッファ形式 2 を使用する場合に設定します。

commform

DCNOFLAGS を設定します。

termnam

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

resv01

ヌル文字を設定します。

senddata

テキスト形態と送信するセグメントの内容を設定した領域を設定します。送信できるセ

4. メッセージ送受信インタフェース

dc_mcf_sendsync - 同期型のメッセージの送信 (C 言語)

グメントの長さの最大値は、通信管理が送信する最大セグメント長にテキスト形態の 4 バイトを加えた値です。

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。

sdata Leng

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

resv02

ヌル文字を設定します。

opcd

DCNOFLAGS を設定します。

watchtime

この引数の設定は無効です。0 を設定しておいてください。また、MCF マネージャ定義の UAP 共通定義で同期送信監視時間を指定しても、時間監視をしません。

リターン値

リターン値	意味
DCMCFRTN_00000	正常に終了しました。
DCMCFRTN_71002	sdata Leng に 32000 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
DCMCFRTN_71003	メッセージキューが満杯です。
DCMCFRTN_71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRTN_71108	メッセージを送信しようとしたが、送信先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_sendsync 関数を呼び出しています。
DCMCFRTN_72001	termnam に設定した論理端末名称が間違っています。
	dc_mcf_sendsync 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72005	先頭セグメントまたは中間セグメント送信時、sdata Leng に 0 以下の値を設定しています。
DCMCFRTN_72016	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。

4. メッセージ送受信インタフェース
dc_mcf_sendsync - 同期型のメッセージの送信 (C 言語)

リターン値	意味
	引数に設定した値に間違いがあります。
DCMCFRTN_72024	commform に設定した値が間違っています。
DCMCFRTN_72026	action に設定したセグメント種別 (DCMCFEMI または DCMCFESI) の値が間違っています。
DCMCFRTN_72041	単一セグメント送信時、sdataleag に 0 以下の値を設定しています。
DCMCFRTN_73001	出力先の論理端末で障害が発生しました。
DCMCFRTN_73002	dc_mcf_sendsync 関数の処理中に、出力先の論理端末と MCF との間のコネクションまたは経路が切り離されました。 dc_mcf_sendsync 関数が呼び出されるまでに、入力元の論理端末と MCF との間のコネクションまたは経路が切り離されました。
DCMCFRTN_73003	同期送信要求 (メッセージ) 時、メッセージ出力先論理端末から、メッセージ受信要求が発生しました。
DCMCFRTN_73008	出力先の論理端末は、次のコマンドで停止しています。 • mcftdctle • mcftdeten
DCMCFRTN_73010	入力または出力メッセージ編集 UOC で障害が発生しました。 メッセージ読み込み時に障害が発生しました。
DCMCFRTN_73015	メッセージ出力先の論理端末は、ほかの UAP で仕掛り中です。 メッセージ出力先の論理端末は、該当する UAP と仕掛り中ではありません。
DCMCFRTN_73018	watchtime に設定した値が間違っています。
DCMCFRTN_73020	出力先の論理端末は停止しています。
DCMCFRTN_77901	同期送信要求 (メッセージ) 時、メッセージ出力先論理端末から、EOT を受信しました。
DCMCFRTN_77902	同期送信要求 (EOT) 時、メッセージ出力先論理端末から、EOT を受信しました。
DCMCFRTN_77903	同期送信要求 (メッセージ) 時、メッセージ出力先論理端末からの RVI を受理しました。メッセージの送信は正常に終了しました。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

CBLDCMCF('RECEIVE') - メッセージまたは 接続情報通知の受信 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

機能

HSC1 手順または HSC2 手順 (非同期モード) の場合、相手システムから送信されたメッセージを受信します。

HSC2 手順 (同期モード) の場合、相手システムからの接続情報通知を受信します。

論理端末に届いたメッセージのうち、一つのセグメントを受信します。セグメントの数だけ CBLDCMCF('RECEIVE') を呼び出すと、一つの論理メッセージを受信できます。なお、接続情報通知は単一セグメントです。

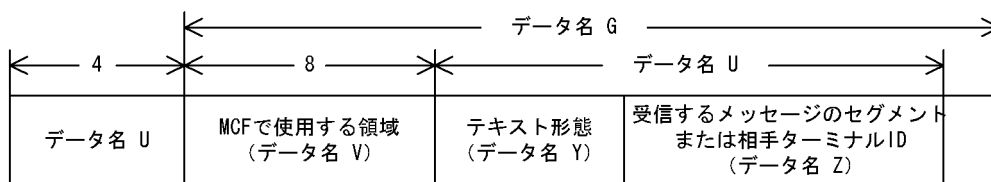
CBLDCMCF('RECEIVE') で受信できるメッセージの種類を次に示します。

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

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

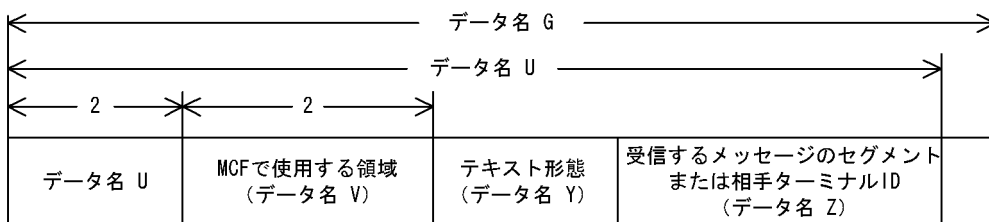
●バッファ形式 1 の場合

(単位: バイト)



●バッファ形式 2 の場合

(単位: バイト)



UAP で値を設定するデータ領域

データ名 A

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

データ名 C

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

VALUE 'FRST'

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

接続情報通知を受信する場合も、「VALUE 'FRST」'を設定します。

VALUE 'SEG」'

中間セグメントまたは最終セグメントを受信する場合に設定します。

4. メッセージ送受信インタフェース

CBLDCMCF('RECEIVE') - メッセージまたは接続情報通知の受信 (COBOL 言語)

データ名 D

空白を設定します。

データ名 G

セグメントを受信する領域の長さを設定します。

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

空白を設定します。

データ名 M4, データ名 M5

0 を設定します。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

VALUE '1'

バッファ形式 1 を使用する場合に設定します。

VALUE '2'

バッファ形式 2 を使用する場合に設定します。

空白

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

データ名 N

MCF で使用する領域です。

データ名 O

空白を設定します。

データ名 P

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

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

データ名 Q

MCF で使用する領域です。

データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 V

【バッファ形式 1 の場合】 PIC X(8)

【バッファ形式 2 の場合】 PIC X(2)

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

データ名 B

ステータスコードが、5 けたの数字で返されます。

データ名 E

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

データ名 F

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

データ名 P

先頭セグメントを受信する場合、入力元の論理端末名称が返されます。

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

データ名 U

【バッファ形式 1 の場合】 PIC 9(9)

テキスト形態とデータ Z で返される内容の長さが返されます。

【バッファ形式 2 の場合】 PIC 9(4)

テキスト形態とデータ Z で返される内容の長さ + 4 が返されます。

データ名 Y

受信したメッセージのテキスト形態が返されます。

4. メッセージ送受信インタフェース

CBLDCMCF('RECEIVE ') - メッセージまたは接続情報通知の受信 (COBOL 言語)

データ名 Z

次に示す内容が返されます。

【HSC1 手順または HSC2 手順 (非同期モード) の場合】

受信したセグメントの内容

【HSC2 手順 (同期モード) で ID を交換する場合】

相手ターミナル ID

【HSC2 手順 (同期モード) で ID を交換しない場合】

返される内容はありません。

受信できるセグメントの長さの最大値は、通信管理の受信する最大セグメント長にテキスト形態の 4 バイトを加えた値です。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する CBLDCMCF('RECEIVE ') を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、データ名 C に「VALUE 'SEG'」を設定して CBLDCMCF('RECEIVE ') を呼び出してください。
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	< MHP の実行でリターンした場合 >
	<ul style="list-style-type: none"> 先頭セグメントを受信する CBLDCMCF('RECEIVE ') を呼び出す前に、中間セグメントまたは最終セグメントを受信する CBLDCMCF('RECEIVE ') を呼び出しています。先頭セグメントを受信する場合は、データ名 C に「VALUE 'FRST'」を設定して CBLDCMCF('RECEIVE ') を呼び出してください。 ステータスコード 71001 が返されたあとに、再び CBLDCMCF('RECEIVE ') を呼び出しています。
72001	< SPP の実行でリターンした場合 >
	SPP では CBLDCMCF('RECEIVE ') を呼び出せません。
72001	データ名 P に設定した論理端末名称が間違っています。
72013	データ名 G の指定値を超えるセグメントを受信しました。データ名 G の指定値を超えた部分は切り捨てられました。
	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。

4. メッセージ送受信インタフェース
 CBLDCMCF('RECEIVE') - メッセージまたは接続情報通知の受信 (COBOL 言語)

ステータス コード	意味
72016	データ名 D に設定した値が間違っています。
	データ名 N またはデータ名 T に設定した値が間違っています。
	データ名 M7 に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72025	データ名 C に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72036	データ名 G の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

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

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
  02 データ名A PIC X(8) VALUE 'RECVSYNC'.  
  02 データ名B PIC X(5).  
  02 FILLER PIC X(3).  
  02 データ名C PIC X(4).  
  02 データ名D PIC X(4) VALUE SPACE.  
  02 データ名E PIC 9(8).  
  02 データ名F PIC 9(8).  
  02 データ名G PIC 9(9) COMP.  
  02 データ名H PIC X(4) VALUE SPACE.  
  02 データ名I PIC X(4) VALUE SPACE.  
  02 データ名J PIC X(4) VALUE SPACE.  
  02 データ名K PIC X(4) VALUE SPACE.  
  02 データ名L PIC X(8) VALUE SPACE.  
  02 データ名M1 PIC X(4) VALUE SPACE.  
  02 データ名M2 PIC X(8) VALUE SPACE.  
  02 データ名M3 PIC X(4) VALUE SPACE.  
  02 データ名M4 PIC 9(9) COMP VALUE ZERO.  
  02 データ名M5 PIC 9(9) COMP 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X PIC X(1).  
  02 データ名Y PIC X(4).  
  02 データ名Z PIC X(n).
```

機能

TP1/NET/HSC では、HSC2 手順 (同期モード) のメッセージの受信に使用します。

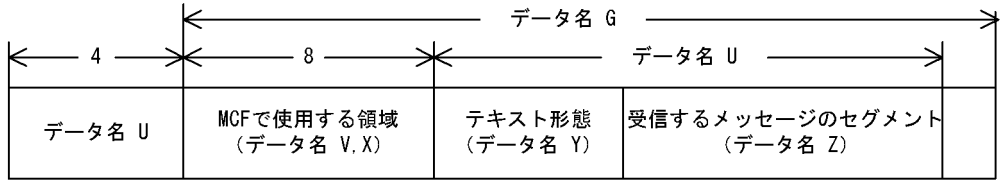
相手システムから届いた同期型の受信メッセージのうち、一つのセグメントを受信しま

す。セグメントの数だけ CBLDCMCF('RECVSYNC') を呼び出すと、一つの論理メッセージを受信できます。

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

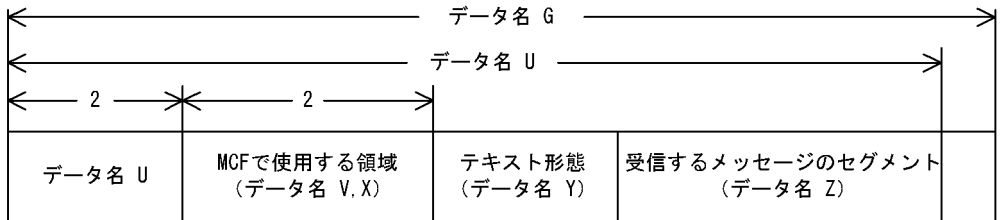
●バッファ形式 1 の場合

(単位: バイト)



●バッファ形式 2 の場合

(単位: バイト)



UAP で値を設定するデータ領域

データ名 A

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

データ名 C

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

VALUE 'FRST'

先頭セグメントを受信する場合に設定します。

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

次のメッセージの先頭セグメントを受信する場合も、「VALUE 'FRST」を設定します。

VALUE 'SEG '

中間セグメントまたは最終セグメントを受信する場合に設定します。

データ名 D

空白を設定します。

4. メッセージ送受信インタフェース

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

データ名 G

セグメントを受信する領域の長さを設定します。

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

空白を設定します。

データ名 M4

0 を設定します。

データ名 M5

このデータ領域は無効です。0 を設定しておいてください。また、MCF マネージャ定義の UAP 共通定義で同期受信監視時間を指定しても、時間監視をしません。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

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

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

データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 V

【バッファ形式 1 の場合】 PIC X(7)

【バッファ形式 2 の場合】 PIC X(1)

MCF で使用する領域です。

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。

データ名 E

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

データ名 F

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

データ名 U

【バッファ形式 1 の場合】 PIC 9(9)

テキスト形態と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)

テキスト形態と受信したセグメントの長さ + 4 が返されます。

データ名 Y

受信したメッセージのテキスト形態が返されます。

データ名 Z

受信したセグメントの内容が返されます。受信できるセグメントの長さの最大値は、通信管理の受信する最大セグメント長にテキスト形態の 4 バイトを加えた値です。

4. メッセージ送受信インタフェース

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

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71001	メッセージの最終セグメントを受信したあとで、その次のセグメントを受信する CBLDCMCF('RECVSYNC') を呼び出しています。直前に呼び出した CBLDCMCF('RECVSYNC') でそのメッセージはすべて受信しました。このステータスコードが返されたあとに、再び次のセグメントを受信する CBLDCMCF('RECVSYNC') を呼び出した場合は、ステータスコード 72000 が返されます。
71002	MCF が終了処理中のため、メッセージの受信を受け付けられません。
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	メッセージ受信に必要な管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
72000	先頭セグメントを受信する CBLDCMCF('RECEIVE ') を呼び出す前に、CBLDCMCF('RECVSYNC') を呼び出しています。 ステータスコード 71001 が返されたあとに、再び CBLDCMCF('RECVSYNC') を呼び出しています。
72001	データ名 P に設定した論理端末名称が間違っています。 CBLDCMCF('RECVSYNC') を呼び出せない論理端末を設定しています。
72013	データ名 G の指定値を超えるセグメントを受信しました。データ名 G の指定値を超えた部分は切り捨てられました。 バッファ形式 2 の場合で、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	データ名 N またはデータ名 T に設定した値が間違っています。 データ名 Q に設定した値が間違っています。 データ名 M7 に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72025	データ名 C に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72036	データ名 G の指定値が不足しています。バッファ形式 1 の場合は 9 バイト以上、バッファ形式 2 の場合は 5 バイト以上の領域を確保してください。
73001	入力元の論理端末で障害が発生しました。
73002	CBLDCMCF('RECVSYNC') の処理中に、入力元の論理端末と MCF との間のコネクションまたは経路が切り離されました。 CBLDCMCF('RECVSYNC') が呼び出されるまでに、入力元の論理端末と MCF との間のコネクションまたは経路が切り離されました。
73008	入力元の論理端末は、次のコマンドで停止しています。 • mcftdetle • mcftdeten

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

ステータスコード	意味
73015	入力元の論理端末は、ほかの UAP で仕掛り中です。
	入力元の論理端末は、該当する UAP と仕掛り中ではありません。
73018	データ名 M5 に設定した値が間違っています。
73020	設定した論理端末は停止しています。
77900	同期受信要求の保留中に、回線維持モードが終了しました。
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

HSC1 手順または HSC2 手順 (非同期モード) の場合に使用できます。

以前に送信したメッセージを、再び送信します。再送するメッセージは、以前に送信し

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

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

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

UAP で値を設定するデータ領域

データ名 A

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

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H, データ名 I

空白を設定します。

データ名 J

一般として再送するか優先として再送するかを設定します。

VALUE 'NORM'

一般の一方送信メッセージとして再送する場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージとして再送する場合に設定します。

空白

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

4. 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)

データ名 K

再送するメッセージに出力通番を付け直すかどうかを設定します。

VALUE 'SEQ'

再送するメッセージに出力通番を付け直す場合に設定します。

VALUE 'NSEQ'

再送するメッセージに出力通番を付け直さない場合に設定します。

空白

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

データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

データ名 M4, データ名 M5

0を設定します。

データ名 M6, データ名 M7

空白を設定します。

データ名 N

MCF で使用する領域です。

データ名 O

一方送信を示す「VALUE 'OUT」を設定します。

データ名 P

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

データ名 Q, データ名 R

空白を設定します。

データ名 S

MCF で使用する領域です。

データ名 T

再送するメッセージを検索するキーとして、以前に送信したメッセージの出力先の論理端末名称を設定します。論理端末名称の長さは最大 8 バイトです。8 バイトに満たない

名称を設定する場合は、後ろを空白で埋めてください。

データ名 U

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

VALUE 'NORM'

一般の一方送信メッセージを対象とする場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージを対象とする場合に設定します。

この値を設定した場合は、データ名 T に出力先の論理端末名称を設定できません。

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。

データ名 V

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

データ名 W

最終出力通番を持つメッセージを再送するかどうかを設定します。

VALUE 'LAST'

最終出力通番を持つメッセージを再送する場合に設定します。

この値を設定した場合は、データ名 V に設定した値は無効となります。

空白

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

データ名 X

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

データ名 B

ステータスコードが、5 けたの数字で返されます。

4. 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

ステータスコード	意味
00000	正常に終了しました。
70904	該当するメッセージがありません。
70905	再送するメッセージのセグメントの長さが、UAP 共通定義 (mcfmuap -e) で指定した値を超えています。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	MCF が終了処理中のため、メッセージの再送を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージキューから取り出したメッセージを格納するバッファ (作業領域) を、メモリ上に確保できませんでした。
71108	メッセージを再送しようとしたのですが、再送先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none"> 先頭セグメントを受信する CBLDCMCF('RECEIVE ') を呼び出す前に、CBLDCMCF('RESEND ') を呼び出しています。 非トランザクション属性の MHP から、CBLDCMCF('RESEND ') を呼び出しています。
	< SPP の実行でリターンした場合 > トランザクションでない SPP の処理から、CBLDCMCF('RESEND ') を呼び出しています。
72001	データ名 P またはデータ名 T に設定した論理端末名称が間違っています。
	CBLDCMCF('RESEND ') を呼び出せない論理端末を設定しています。
72016	データ名 J に設定した値が間違っています。
	データ名 M4 に設定した値が間違っています。
	データ名 U に設定した値が間違っています。
	データ名 N, データ名 S, またはデータ名 X に設定した値が間違っています。
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

注意事項

メッセージの再送時には、MCF マネージャ定義の UAP 共通定義 (mcfmuap) の -e オプ

ションと -l オプションの指定値に注意してください。

-e オプション

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

-l オプション

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

CBLDCMCF('SEND ') - 一方送信メッセージ または回線接続要求の送信 (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Y PIC X(4).  
02 データ名Z PIC X(n).
```

機能

HSC1 手順または HSC2 手順 (非同期モード) の場合、相手システムへメッセージを送信します。

HSC2 手順 (同期モード) の場合、相手システムへ回線接続要求を送信します。

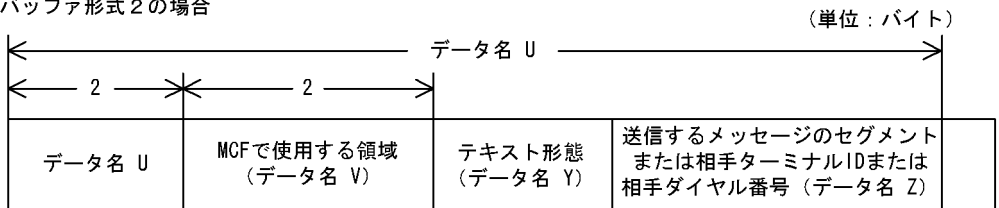
相手システムへ送る一方送信メッセージのうち、一つのセグメントを送信します。セグメントの数だけ CBLDCMCF('SEND ') を呼び出すと、一つの論理メッセージを送信できます。なお、回線接続要求は単一セグメントです。

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

●バッファ形式 1 の場合



●バッファ形式 2 の場合



UAP で値を設定するデータ領域

データ名 A

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

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H

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

VALUE 'ESI '

先頭セグメントまたは中間セグメントを送信する場合に設定します。

VALUE 'EMI '

最終セグメントを送信する場合や、メッセージが単一セグメントの場合に、

4. 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または回線接続要求の送信 (COBOL 言語)

「VALUE 'EMI」を設定します。

メッセージの送信の終了を連絡するために、最後は必ずこの値を設定してください。
回線接続要求を送信する場合にも、「VALUE 'EMI」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。論理端末名称の長さは最大 8 バイトです。8 バイトに満たない名称を設定する場合は、後ろを空白で埋めてください。

データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 U

【バッファ形式 1 の場合】 PIC 9(9)

テキスト形態とデータ名 Z で設定する内容のセグメントの長さを設定します。メッセージの送信の終了を連絡する場合で、セグメントの内容がないときは、0 を設定してください。

【バッファ形式 2 の場合】 PIC 9(4)

テキスト形態とデータ名 Z で設定する内容のセグメントの長さ + 4 を設定します。メッセージの送信の終了を連絡する場合で、セグメントの内容がないときは、4 を設定してください。

データ名 V

【バッファ形式 1 の場合】 PIC X(8)

【バッファ形式 2 の場合】 PIC X(2)

- 領域の 2^4 ビットに 1 を設定
送信完了通知イベントまたは送信障害通知イベントを通知させるときに設定します。ただし、これらのイベントを処理する MHP を MCF アプリケーション定義で

4. 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または回線接続要求の送信 (COBOL 言語)

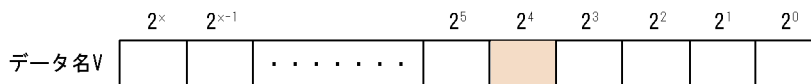
指定していない場合は無効です。


メッセージが複数のセグメントで構成される場合、先頭セグメント送信時の指定が有効になります。

- 領域の 2^4 ビットに 0 を設定

送信完了通知イベントまたは送信障害通知イベントを通知させません。

データ名 V とビットの設定位置の関係を次に示します。



(凡例)  : 送信完了通知イベントおよび送信障害通知イベントの要否を設定するビット

注

X の値は次のとおりです。

【バッファ形式 1 の場合】 X=63

【バッファ形式 2 の場合】 X=15

データ名 Y

送信メッセージのテキスト形態を設定します。

データ名 Z

次に示す内容を設定します。

【HSC1 手順または HSC2 手順 (非同期モード) の場合】

送信するセグメントの内容

【HSC2 手順 (同期モード) で ID を交換する場合】

相手ターミナル ID

【HSC2 手順 (同期モード) で ID を交換しない場合】

相手ダイヤル番号

送信できるセグメントの長さの最大値は、通信管理が送信する最大セグメント長にテキスト形態の 4 バイトを加えた値です。

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

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

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
02 データ名A PIC X(8) VALUE 'SENDSYNC'.  
02 データ名B PIC X(5).  
02 FILLER PIC X(3).  
02 データ名C PIC X(4) VALUE SPACE.  
02 データ名D PIC X(4) VALUE SPACE.  
02 データ名E PIC 9(8).  
02 データ名F PIC 9(8).  
02 データ名G PIC 9(9) COMP VALUE ZERO.  
02 データ名H PIC X(4).  
02 データ名I PIC X(4) VALUE SPACE.  
02 データ名J PIC X(4).  
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 SPACE.  
02 データ名P PIC X(8).  
02 データ名Q PIC X(8) VALUE SPACE.  
02 データ名R PIC X(8) VALUE SPACE.  
02 データ名T PIC X(28) VALUE LOW-VALUE.  
01 一意名3.  
02 データ名U PIC 9(x) COMP.  
02 データ名V PIC X(x).  
02 データ名Y PIC X(4).  
02 データ名Z PIC X(n).
```

機能

TP1/NET/HSC では、HSC2 手順 (同期モード) のメッセージの送信に使用します。

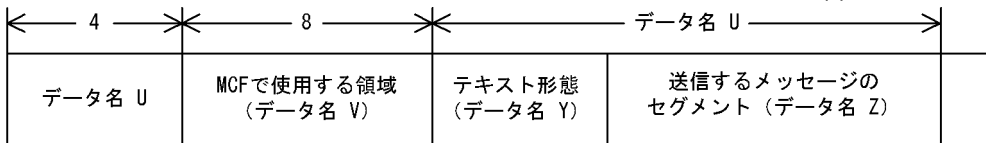
相手システムへ送る同期型のメッセージのうち、一つのセグメントを送信します。セグメントの数だけ CBLDCMCF('SENDSYNC') を呼び出すと、一つの論理メッセージを送

信できます。

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

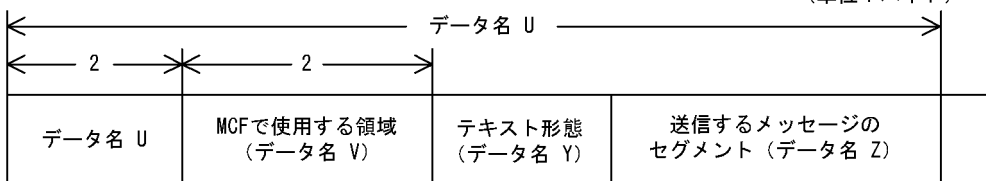
●バッファ形式 1 の場合

(単位: バイト)



●バッファ形式 2 の場合

(単位: バイト)



UAP で値を設定するデータ領域

データ名 A

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

データ名 C, データ名 D

空白を設定します。

データ名 E, データ名 F

MCF で使用する領域です。

データ名 G

0 を設定します。

データ名 H

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

VALUE 'ESI '

先頭セグメントまたは中間セグメントを送信する場合に設定します。

VALUE 'EMI '

最終セグメントを送信する場合や, メッセージが単一セグメントの場合に,

「VALUE 'EMI '」を設定します。

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

4. メッセージ送受信インタフェース

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

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

空白を設定します。

データ名 M4

0 を設定します。

データ名 M5

このデータ領域は無効です。0 を設定しておいてください。また, MCF マネージャ定義の UAP 共通定義で同期送信監視時間を指定しても, 時間監視をしません。

データ名 M6

空白を設定します。

データ名 M7

使用するバッファ形式を設定します。

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

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

データ名 Q, データ名 R

空白を設定します。

データ名 T

MCF で使用する領域です。

データ名 U

【バッファ形式 1 の場合】 PIC 9(9)

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

【バッファ形式 2 の場合】 PIC9(4)

テキスト形態と送信するメッセージのセグメントの長さ + 4 を設定します。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】 PICX(8)

【バッファ形式 2 の場合】 PICX(2)

MCF で使用する領域です。

データ名 Y

送信するメッセージのテキスト形態を設定します。

データ名 Z

送信するセグメントの内容を設定します。送信できるセグメントの長さの最大値は、通信管理が送信する最大セグメント長にテキスト形態の 4 バイトを加えた値です。

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。
71002	データ名 U に 32000 バイトを超える値を設定しています。 MCF が終了処理中のため、同期型の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。

4. メッセージ送受信インタフェース

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

ステータスコード	意味
72000	先頭セグメントを受信する CBLDCMCF('RECEIVE ') を呼び出す前に、CBLDCMCF('SENDSYNC') を呼び出しています。
72001	データ名 P に設定した論理端末名称が間違っています。 CBLDCMCF('SENDSYNC') を呼び出せない論理端末を設定しています。
72005	バッファ形式 1 の先頭セグメントまたは中間セグメント送信時、データ名 U に 0 以下の値を設定しています。または、バッファ形式 2 の先頭セグメントまたは中間セグメント送信時、データ名 U に 4 以下の値を設定しています。
72016	データ名 N またはデータ名 T に設定した値が間違っています。 データ名 M1 またはデータ名 M7 に設定した値が間違っています。
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72026	データ名 H に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72041	バッファ形式 1 の単一セグメント送信時、データ名 U に 0 以下の値を設定しています。または、バッファ形式 2 の単一セグメント送信時、データ名 U に 4 以下の値を設定しています。
73001	出力先の論理端末で障害が発生しました。
73002	CBLDCMCF('SENDSYNC') の処理中に、入力元の論理端末と MCF との間のコネクションまたは経路が切り離されました。 CBLDCMCF('SENDSYNC') が呼び出されるまでに、出力先の論理端末と MCF との間のコネクションまたは経路が切り離されました。
73003	同期送信要求 (メッセージ) 時、メッセージ出力先論理端末から、メッセージ受信要求が発生しました。
73008	出力先の論理端末は、次のコマンドで停止しています。 • mcftdctle • mcftdeten
73010	入力または出力メッセージ編集 UOC で障害が発生しました。 メッセージ読み込み時に障害が発生しました。
73015	出力先の論理端末は、ほかの UAP で仕掛り中です。 出力先の論理端末は、該当する UAP と仕掛り中ではありません。
73018	データ名 M5 に設定した値が間違っています。
73020	出力先の論理端末は停止しています。
77901	同期送信要求 (メッセージ) 時、メッセージ出力先論理端末から、EOT を受信しました。
77902	同期送信要求 (EOT) 時、メッセージ出力先論理端末から、EOT を受信しました。

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

ステータスコード	意味
77903	同期送信要求 (メッセージ) 時, メッセージ出力先論理端末からの RVI を受理しました。メッセージの送信は正常に終了しました。
上記以外	プログラムの破壊などによる, 予期しないエラーが発生しました。

RECEIVE - メッセージの受信 (データ操作言語)

形式

DATA DIVISION (通信記述項) の指定

CD 通信記述名

```
FOR {INPUT | I-O}  
{STATUS KEY IS データ名1}  
{SYMBOLIC TERMINAL IS データ名2}  
{MESSAGE DATE IS データ名3}  
{MESSAGE TIME IS データ名4}  
{SYNCHRONOUS MODE IS {SYNC | データ名6}}  
{WAITING TIME IS データ名11}.
```

PROCEDURE DIVISION (通信文) の指定

```
RECEIVE 通信記述名  
  {FIRST} SEGMENT  
  INTO 一意名1.
```

機能

次に示す CALL インタフェースの機能を実現します。

【HSC1 手順または HSC2 手順 (非同期モード) の場合】

一方送信メッセージの受信 CBLDCMCF('RECEIVE ')

【HSC2 手順 (同期モード) の場合】

接続情報通知の受信 CBLDCMCF('RECEIVE ')

同期型のメッセージの受信 CBLDCMCF('RECVSYNC')

通信記述項に設定する項目

FOR 句

次のどちらかの値を指定します。

INPUT

メッセージの受信, または接続情報通知の受信

I-O

メッセージの受信, 接続情報通知の受信, または同期型のメッセージの受信

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。

MESSAGE TIME 句

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

SYNCHRONOUS MODE 句

同期型のメッセージを受信する場合に指定します。次のどちらかの値を指定してください。

SYNC

同期型のメッセージの受信

データ名 6

次の値を設定したデータ項目

'1'...同期型のメッセージの受信

指定を省略した場合は、非同期型のメッセージの受信、または接続情報通知の受信が設定されます。

WAITING TIME 句

この句の指定は無効です。0を指定しておいてください。また、MCF マネジャ定義のUAP 共通定義で同期受信監視時間を指定しても、時間監視をしません。

通信文に指定する項目

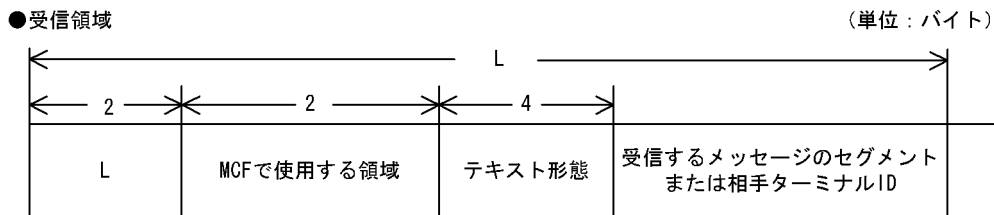
FIRST

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。相手システムから送信されるセグメントの最大長は 31966 バイトです。一意名 1 の形式を次に示します。

4. 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）



テキスト形態の次に続く領域には、次に示す内容が返されます。

【HSC1 手順または HSC2 手順（非同期モード）の場合】

受信したセグメントの内容

【HSC2 手順（同期モード）で ID を交換する場合】

相手ターミナル ID

【HSC2 手順（同期モード）で ID を交換しない場合】

返される内容はあります。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する RECEIVE 文を、2 回以上実行しています。中間セグメントまたは最終セグメントを受信する場合は、FIRST を指定しないで RECEIVE 文を実行してください。
71001	メッセージの最終セグメントを受信したあとで、その次のセグメントを受信する RECEIVE 文を実行しています。直前に実行した RECEIVE 文でそのメッセージはすべて受信しました。 このステータスコードが返されたあとに、再び次のメッセージを受信する RECEIVE 文を実行した場合は、ステータスコード 72000 が返されます。
71002	メッセージキューからの入力処理中に障害が発生しました。 メッセージキューが閉塞されています。 メッセージキューが割り当てられていません。 MCF が終了処理中のため、メッセージの受信を受け付けられません。
71108	メッセージの受信に必要な管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none"> 非同期型のメッセージの先頭セグメントを受信する RECEIVE 文を実行する前に、そのほかのセグメントを受信する RECEIVE 文を実行しています。 ステータスコード 71000 が返されたあとで、再び非同期型のメッセージを受信する RECEIVE 文を実行しています。 < SPP の実行でリターンした場合 > SPP では非同期型のメッセージを受信する RECEIVE 文を実行できません。

4. 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
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 句に設定した値が間違っています。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72036	一意名 1 の L の指定値が不足しています。5 バイト以上の領域を確保してください。
73001	入力元の論理端末で障害が発生しました。
73002	同期型のメッセージを受信しているときに、入力元の論理端末と MCF との間のコネクションまたは経路が切り離されました。 同期型のメッセージを受信するまでに、入力元の論理端末と MCF との間のコネクションまたは経路が切り離されました。
73008	入力元の論理端末は、次のコマンドで停止しています。 • mcftdctle • mcftdcten
73010	入力または出力メッセージ編集 UOC で障害が発生しました。 メッセージ読み込み時に障害が発生しました。
73015	入力元の論理端末は、ほかの UAP で仕掛り中です。 入力元の論理端末は、該当する UAP と仕掛り中ではありません。
73018	同期型のメッセージ受信を発行したときに、データ名 11 に設定した監視時間が間違っています。
73020	設定した論理端末は停止しています。
77900	同期受信要求の保留中に、回線維持モードが終了しました。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

SEND - メッセージの送信 (データ操作言語)

形式 1 (セグメントの内容を設定して送信する場合, または回線接続要求を送信する場合)

DATA DIVISION (通信記述項) の指定

CD 通信記述名

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

PROCEDURE DIVISION (通信文) の指定

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

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

DATA DIVISION (通信記述項) の指定

CD 通信記述名

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

PROCEDURE DIVISION (通信文) の指定

```
SEND 通信記述名 WITH EMI.
```

機能

次に示す CALL インタフェースの機能を実現します。

【HSC1 手順または HSC2 手順 (非同期モード) の場合】

一方送信メッセージの送信 CBLDCMCF('SEND ')

【HSC2 手順 (同期モード) の場合】

回線接続要求の送信 CBLDCMCF('SEND ')

同期型のメッセージの送信 CBLDCMCF('SENDSYNC')

通信記述項に設定する項目

FOR 句

次のどちらかの値を指定します。

OUTPUT

一方送信メッセージ，または回線接続要求の送信の送信

I-O

同期型のメッセージの送信

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

メッセージ送信が同期型か非同期型かを指定します。

SYNC

同期型のメッセージ送信

同期型のメッセージの送信の場合に設定します。

ASYNCR

非同期型のメッセージ送信

一方送信メッセージ，または回線接続要求の送信の場合に設定します。

データ名 6

次の値を設定したデータ項目

'0'...非同期型のメッセージ送信

'1'...同期型のメッセージ送信

指定を省略した場合は，非同期型のメッセージ送信 (ASYNCR) が設定されます。

SWITCHING MODE 句

一方送信メッセージの場合に，一般か優先かを指定します。

NORMAL

一般の一方送信メッセージ

PRIOR

優先の一方送信メッセージ

データ名 7

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

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

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

DETAIL MODE 句

一方送信メッセージの場合に、出力通番を付けるかどうかを指定します。

データ名 10

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

省略した場合は、出力通番を付けません。

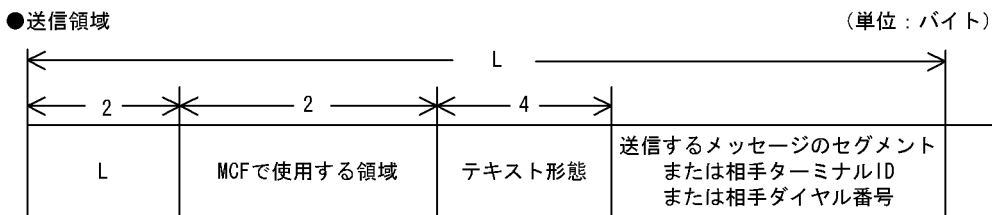
WAITING TIME 句

この句の指定は無効です。0を指定しておいてください。また、MCF マネージャ定義の UAP 共通定義で同期送信監視時間を指定しても、時間監視をしません。

通信文に指定する項目

一意名 1

セグメントを送信するデータ項目を指定します。送信できるセグメントの最大長は、通信管理が送信する最大セグメント長にテキスト形態の 4 バイトを加えた値です。一意名 1 の形式を次に示します。



テキスト形態の次に続く領域には、次に示す内容を設定します。

【HSC1 手順または HSC2 手順 (非同期モード) の場合】

送信するセグメントの内容

【HSC2 手順 (同期モード) で ID を交換する場合】

相手ターミナル ID

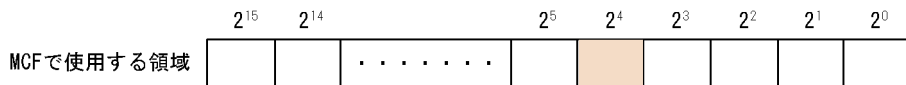
【HSC2 手順 (同期モード) で ID を交換しない場合】


相手ダイヤル番号

また、MCF で使用する領域の 2^4 ビット目に、送信完了通知イベントまたは送信障害通知イベントを通知させるかどうかを指定してください。

- 領域の 2^4 ビットに 1 を設定
送信完了通知イベントまたは送信障害通知イベントを通知させるときに設定します。ただし、これらのイベントを処理する MHP を MCF アプリケーション定義で指定していない場合は無効です。
メッセージが複数のセグメントで構成される場合、先頭セグメント送信時の指定が有効になります。
- 領域の 2^4 ビットに 0 を設定
送信完了通知イベントまたは送信障害通知イベントを通知させません。

一意名 1 の MCF で使用する領域とビットの設定位置の関係を次に示します。



(凡例)  : 送信完了通知イベントおよび送信障害通知イベントの要否を設定するビット

WITH 句

メッセージの最終セグメントを送信するかどうかを指定します。非同期型のメッセージ送信（一方送信メッセージ）または回線接続要求の送信の場合に指定します。

ESI

先頭セグメントまたは中間セグメントの送信

EMI

最終セグメントまたは単一セグメントの送信

一意名 2

次の値を設定したデータ項目

'1'...ESI（先頭セグメントまたは中間セグメントの送信）

'2'...EMI（最終セグメント，単一セグメント，または回線接続要求の送信）

省略した場合は、EMI（最終セグメント，単一セグメント，または回線接続要求の送信）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。

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

ステータスコード	意味
	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 > <ul style="list-style-type: none"> 先頭セグメントを受信する RECEIVE 文を実行する前に、SEND 文を実行しています。 非応答型の MHP から応答メッセージを送信しています。
	< SPP の実行でリターンした場合 > <ul style="list-style-type: none"> トランザクションでない SPP の処理から、SEND 文を実行しています。 SPP では、応答メッセージを送信する SEND 文を実行できません。
72001	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送信時、一意名 1 の L に 4 以下の値を設定しています。
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。
72011	MAP NAME 句は設定できません。
72013	一意名 3 の L の指定値を超えるセグメントを受信しました。一意名 3 の L の指定値を超えた部分は切り捨てられました。
72017	DETAIL MODE 句に設定した値が間違っています。
72018	SWITCHING MODE 句に設定した値が間違っています。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72026	WITH 句に設定した値が間違っています。
72036	一意名 3 の L の指定値が不足しています。5 バイト以上の領域を確保してください。
72037	BEFORE 句に設定した値が間違っています。
72041	単一セグメント送信時、一意名 1 の L に 4 以下の値を設定しています。
73001	出力先の論理端末で障害が発生しました。
73002	MCF 通信サービスで障害が発生しました。

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

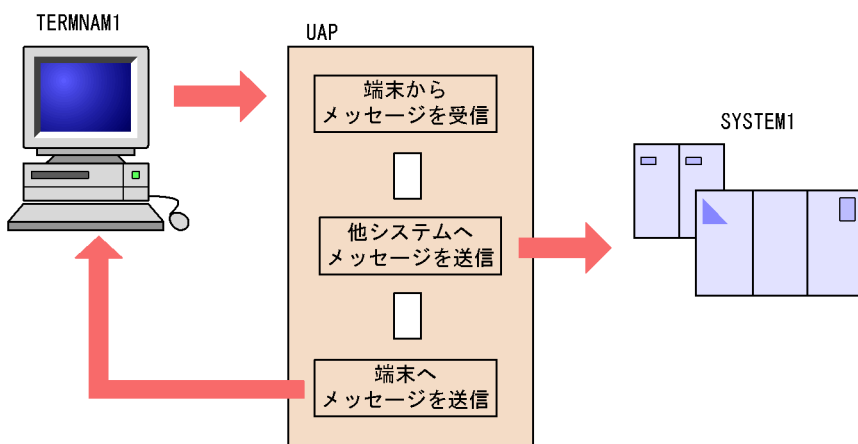
ステータスコード	意味
73003	同期送信要求（メッセージ）時，メッセージ出力先論理端末から，メッセージ受信要求が発生しました。
73008	論理端末が閉塞中，または MCF が終了処理中に，UAP から SEND 文が呼び出されました。
73010	入力，または出力メッセージ編集 UOC で障害が発生しました。
	メッセージ編集エラーが発生しました。
	メッセージの読み込み時に障害が発生しました。
73015	メッセージ出力先の論理端末は，ほかの UAP で仕掛り中です。
	メッセージ出力先の論理端末は，該当する UAP と仕掛り中ではありません。
73018	同期型のメッセージ送信を発行したときに，データ名 11 に設定した監視時間が間違っています。
73020	出力先の論理端末は停止しています。
77901	同期送信要求（メッセージ）時，メッセージ出力先論理端末から，EOT を受信しました。
77902	同期送信要求（EOT）時，メッセージ出力先論理端末から，EOT を受信しました。
77903	同期送信要求（メッセージ）時，メッセージ出力先論理端末からの RVI を受理しました。メッセージの送信は正常に終了しました。
上記以外	プログラムの破壊などによる，予期しないエラーが発生しました。

ユーザアプリケーションプログラム作成例

HSC1 手順または HSC2 手順（非同期モード）の例

UAP の作成例として、HSC1 手順または HSC2 手順（非同期モード）の場合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（K&R 版）、COBOL 言語、およびデータ操作言語で示します。

図 4-1 UAP 作成例の処理の流れ



C 言語（K&R 版）

C 言語（K&R 版）を使用した UAP のコーディング例を示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/c/ap1.c のファイルで提供しています。

```
#include <dcmcf.h>
#include <dcmhsc.h>
void ex_uap1()
{
    char    termnam[9] ;
    char    recvdata[2048];
    DCLONG rdataleng ;
    DCLONG time ;
    static char resv01[9] = "¥0" ;           /* 予備領域の初期化 */
    static char resv02[9] = "¥0" ;         /* 予備領域の初期化 */
    char    *workadd = recvdata ;

    struct {
        DCLONG    mcf[2];
        DCLONG    text_code;
        char      data[2048];
    } senddata;

    dc_mcf_receive(DCMCFRST,DCNOFLAGS,termnam,resv01,
                  workadd,&rdataleng,2048,&time) ; /* メッセージの受信 */
```



```

/***** データの処理 *****/

    senddata.text_code = DCMHSC_DLETEXT ;

    dc_mcf_send(DCMCFEMI,DCMCFOUT,"SYSTEM1",
               resv01,&senddata,504,resv02,DCNOFLAGS) ;
               /*メッセージの送信(分岐送信) */

    dc_mcf_send(DCMCFEMI,DCMCFOUT,"TERMNAM1",
               resv01,&senddata,504,resv02,DCNOFLAGS) ;
               /*メッセージの送信(一方送信) */
}

```

COBOL 言語

COBOL 言語を使用した UAP のコーディング例を示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/cobol/ap1.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 'FRST'.
    02  RTN-CODE      PIC X(4) VALUE SPACE.
    02  DAY-ID        PIC 9(8).
    02  TIME-ID       PIC 9(8).
    02  SEG-LENG      PIC 9(9) COMP VALUE 2048.
    02  MCFUSE01      PIC X(4) VALUE SPACE.
    02  MCFUSE02      PIC X(4) VALUE SPACE.
    02  MCFUSE03      PIC X(4) VALUE SPACE.
    02  MCFUSE04      PIC X(4) VALUE SPACE.
    02  MCFUSE05      PIC X(8) VALUE SPACE.
    02  MCFUSE06      PIC X(4) VALUE SPACE.
    02  MCFUSE07      PIC X(8) VALUE SPACE.
    02  MCFUSE08      PIC X(4) VALUE SPACE.
    02  MCFUSE09      PIC 9(9) COMP VALUE ZERO.
    02  MCFUSE10      PIC 9(9) COMP VALUE ZERO.
    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.

```

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

```

02 MCFUSE17      PIC X(4).
02 MCFUSE18      PIC X(2).
02 MCFUSE19      PIC X(1).
02 MCFUSE20      PIC X(1).
02 TEXT-TYPE1    PIC X(4).
02 REC-MSGSEG1   PIC X(2044).
01 SND1.
02 MSG-SEND      PIC X(8) VALUE 'SEND  '.
02 STATUS-CODE2  PIC X(5).
02 FILLER        PIC X(3).
02 MCFUSE21      PIC X(4) VALUE SPACE.
02 MCFUSE22      PIC X(4) VALUE SPACE.
02 MCFUSE23      PIC 9(8).
02 MCFUSE24      PIC 9(8).
02 MCFUSE25      PIC 9(9) COMP VALUE ZERO.
02 SEND-SEG      PIC X(4) VALUE 'EMI  '.
02 SEND-SYNC     PIC X(4) VALUE SPACE.
02 SEND-NORM     PIC X(4) VALUE 'NORM'.
02 SEND-NO       PIC X(4) VALUE 'NSEQ'.
02 MCFUSE26      PIC X(8) VALUE SPACE.
02 SEND-CODE     PIC X(4) VALUE SPACE.
02 MCFUSE27      PIC X(8) VALUE SPACE.
02 MCFUSE28      PIC X(4) VALUE SPACE.
02 MCFUSE29      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE30      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE31      PIC X(1) VALUE SPACE.
02 MCFUSE32      PIC X(1) VALUE SPACE.
02 MCFUSE33      PIC X(14) VALUE LOW-VALUE.
01 CD2.
02 SENDSEG-CODE2 PIC X(4) VALUE 'OUT  '.
02 TERM-CODE2    PIC X(8) VALUE 'SYSTEM1'.
02 MCFUSE34      PIC X(8) VALUE SPACE.
02 MCFUSE35      PIC X(8) VALUE SPACE.
02 MCFUSE36      PIC X(28) VALUE LOW-VALUE.
01 DATA2.
02 MSGSEG-LENG2  PIC 9(9) COMP VALUE 512.
02 MCFUSE37      PIC X(8).
02 TEXT-TYPE2    PIC X(4).
02 REC-MSGSEG2   PIC X(508).
01 SND2.
02 MSG-SEND1     PIC X(8) VALUE 'SEND  '.
02 STATUS-CODE3  PIC X(5).
02 FILLER        PIC X(3).
02 MCFUSE41      PIC X(4) VALUE SPACE.
02 MCFUSE42      PIC X(4) VALUE SPACE.
02 MCFUSE43      PIC 9(8).
02 MCFUSE44      PIC 9(8).
02 MCFUSE45      PIC 9(9) COMP VALUE ZERO.
02 SEND-SEG1     PIC X(4) VALUE 'EMI  '.
02 SEND-SYNC1    PIC X(4) VALUE SPACE.
02 SEND-NORM1    PIC X(4) VALUE 'NORM'.
02 SEND-NO1     PIC X(4) VALUE 'NSEQ'.
02 MCFUSE46      PIC X(8) VALUE SPACE.
02 SEND-CODE1    PIC X(4) VALUE SPACE.
02 MCFUSE47      PIC X(8) VALUE SPACE.
02 MCFUSE48      PIC X(4) VALUE SPACE.
02 MCFUSE49      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE50      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE51      PIC X(1) VALUE SPACE.
02 MCFUSE52      PIC X(1) VALUE SPACE.
02 MCFUSE53      PIC X(14) VALUE LOW-VALUE.
01 CD3.
02 SENDSEG-CODE3 PIC X(4) VALUE 'OUT  '.
02 TERM-CODE3    PIC X(8) VALUE 'TERMNAM1'.
02 MCFUSE54      PIC X(8) VALUE SPACE.

```

```

02 MCFUSE55          PIC X(8) VALUE SPACE.
02 MCFUSE56          PIC X(28) VALUE LOW-VALUE.
      :
      :
PROCEDURE DIVISION.
CALL 'CBLDCMCF' USING RCV CD1 DATA1.  ** メッセージの受信 **

*           処理 1           *

CALL 'CBLDCMCF' USING SND1 CD2 DATA2.  ** メッセージの送信 (分岐送信)
**

*           処理 2           *

CALL 'CBLDCMCF' USING SND2 CD3 DATA2.  ** メッセージの送信 (一方送信)
**

*           処理 3           *

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示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/dml/ap1.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. EX-UAP1.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*
*           ワーク変数           *
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
*           メッセージ受信領域           *
*
01 RECV-AREA.
   02 RE-DATALENG          PIC 9(4) COMP VALUE 1028.
   02 RE-RSV1             PIC X(2).
   02 RE-DATA             PIC X(1024).
*
*           一方送信メッセージ送信領域           *
*
01 SEND-IO-AREA.
   02 IO-DATALENG          PIC 9(4) COMP VALUE 20.
   02 IO-RSV1             PIC X(2).
   02 IO-TCODE            PIC X(4).
   02 IO-DATA             PIC X(16) VALUE 'SVRA SEND DATA1'.
*
*           通信記述項           *

```

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

```
*
COMMUNICATION SECTION.
*
*       メッセージの受信の通信記述項           *
*
CD RECV-INF
FOR INPUT
STATUS KEY IS           RE-STATUS
SYMBOLIC TERMINAL IS   RE-TERMNAM
MESSAGE DATE IS        RE-DATE
MESSAGE TIME IS         RE-TIME.
*
*       メッセージ送信の通信記述項           *
*
CD SEND-IO
FOR OUTPUT
STATUS KEY IS           SE-STATUS-IO
SYMBOLIC TERMINAL IS   SE-TERMNAM-IO.

PROCEDURE DIVISION.

*
*       メッセージの受信           *
*
RECEIVE RECV-INF
FIRST SEGMENT
INTO RECV-AREA.
*
*       一方送信メッセージの送信           *
*
MOVE 'SYSTEM1' TO SE-TERMNAM-IO.
MOVE X'00000004' TO IO-TCODE.
SEND SEND-IO
FROM SEND-IO-AREA
WITH EMI.
*
*       一方送信メッセージの送信           *
*
MOVE 'TERMNAM1' TO SE-TERMNAM-IO.
SEND SEND-IO
FROM SEND-IO-AREA
WITH EMI.

EXIT PROGRAM.
```

HSC2 手順 (同期モード)

HSC2 手順 (同期モード) の場合の UAP のコーディング例を C 言語 (K&R 版), COBOL 言語, およびデータ操作言語で示します。

UAP のコーディング例で示す発信用 SPP の例と発信用 MHP の例を図 4-2 に, 着信用 MHP の例を図 4-3 に示します。

図 4-2 発信用 SPP, 発信用 MHP の例

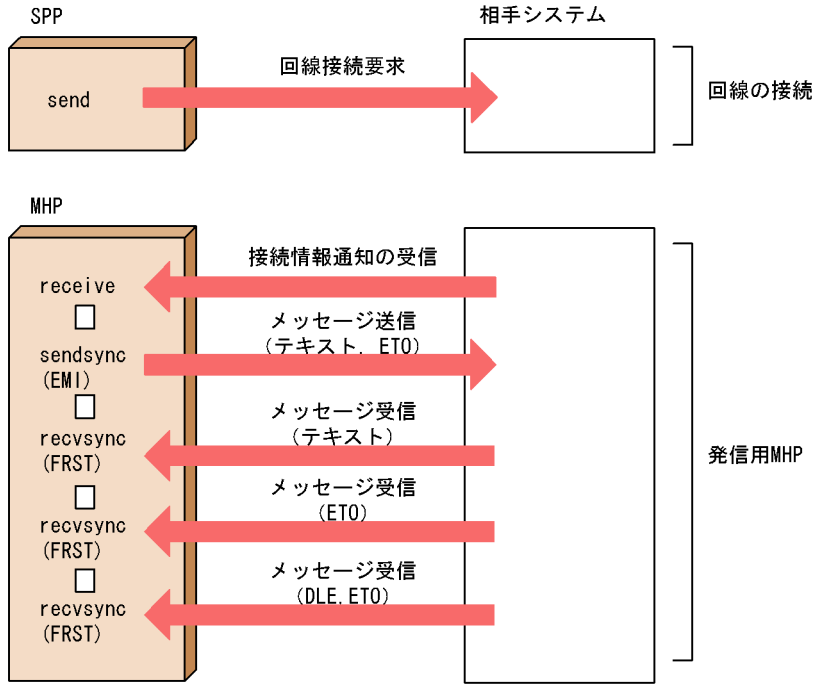
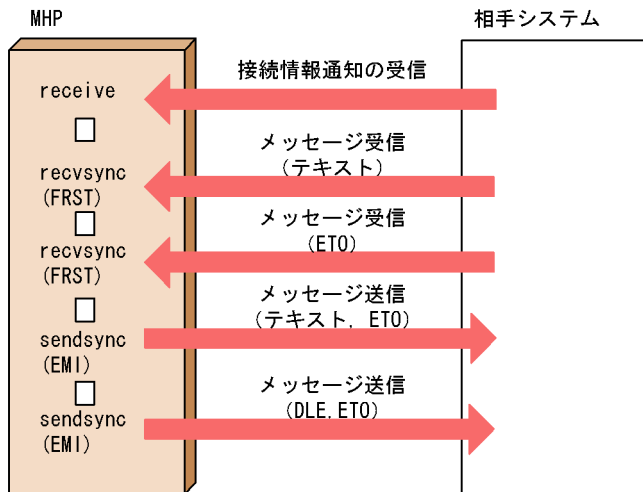


図 4-3 着信用 MHP の例



C 言語 (K&R 版)

C 言語 (K&R 版) を使用した UAP のコーディング例を示します。

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

発信用 SPP のコーディング例

図 4-2 に示した発信用 SPP の例のコーディングを示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/c/sv1.c のファイルで提供しています。

```
/*
 * TP1/NET/HSC HSC2手順 発信用SPP
 */
# include <dcpcf.h>
# include <dcmhsc.h>
# include <dcmhsc2.h>
void hsc2spp(in,in_len,out,out_len)
    char *in ;
    DCULONG *in_len ;
    char *out ;
    DCULONG *out_len ;

{
    static char resv01[9] = "¥0" ;
    static char resv02[9] = "¥0" ;
    dcmhs2_connect_inf data;
    int rtn_code;
    DCLONG size;
/*
 * 回線接続要求
 */
    data.text_code = DCMHSC_CONNECT_REQ; /* テキスト形態設定 */
    strcpy(data.termid,"tid1"); /* ターミナルID設定 */
    size = strlen(data.termid) + sizeof(data.text_code);
    /* データ長設定 */
    rtn_code = dc_mcf_send(DCMCFEMI,DCMCFOUT,"hsc2le1",
        resv01,&data,size,resv02,
        DCNOFLAGS) ;
    if(rtn_code != DCMCFRTN_00000)
        abort();
}
```

発信用 MHP のコーディング例

図 4-2 に示した発信用 MHP の例のコーディングを示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/c/ap2.c のファイルで提供しています。

```
/*
 * TP1/NET/HSC HSC2手順 発信用MHP
 */
# include <dcpcf.h>
# include <dcmhsc.h>
# include <dcmhsc2.h>
typedef struct {
    DCLONG mcf[2];
    DCLONG text_code;
    char data[2048];
}data_t ;
```

```

/*
 * MHP サービス関数
 */
void hsc2mhp()
{
  char      leid[16];
  data_t    sdata;
  data_t    rdata;
  char      resv01[9];
  char      resv02[9];
  int       rtn_code;
  DCLONG    rdataleng;
  DCLONG    *time;
/*
 *  接続情報通知の受信
 */
  memset(resv01, '¥0', sizeof(resv01));
  memset(resv02, '¥0', sizeof(resv02));
  rtn_code = dc_mcf_receive(DCMCFRST, DCNOFLAGS, leid, resv01, &rdata,
                           &rdataleng, sizeof(data_t), &time);
  if( rtn_code != DCMCFRTN_00000 )
    abort();
/*
 *  メッセージの送信 (テキスト, EOT付き)
 */
  sdata, text_code = DCMHSC_DELETETEXT_EOT;
  memset(resv01, '¥0', sizeof(resv01));
  memset(resv02, '¥0', sizeof(resv02));

  /***** メッセージの編集 *****/

  rtn_code = dc_mcf_sendsync(DCMCFEMI, DCNOFLAGS, leid, resv01, &sdata,
                             200, resv02, DCNOFLAGS, 0);
  if( rtn_code != DCMCFRTN_00000 )
    abort();
/*
 *  メッセージの受信 (テキスト)
 */
  memset(resv01, '¥0', sizeof(resv01));
  memset(resv02, '¥0', sizeof(resv02));
  rtn_code = dc_mcf_recvsync(DCMCFRST, DCNOFLAGS, leid, resv01, &rdata,
                             &rdataleng, sizeof(data_t), &time, 0);
  if( rtn_code != DCMCFRTN_00000 )
    abort();

  /***** メッセージの解析 *****/

/*
 *  メッセージの受信 (EOT)
 */
  memset(resv01, '¥0', sizeof(resv01));
  memset(resv02, '¥0', sizeof(resv02));
  rtn_code = dc_mcf_recvsync(DCMCFRST, DCNOFLAGS, leid, resv01, &rdata,
                             &rdataleng, sizeof(data_t), &time, 0);
  if( rtn_code != DCMCFRTN_00000 )
    abort();
/*
 *  メッセージの受信 (DLE・EOT)
 */
  memset(resv01, '¥0', sizeof(resv01));
  memset(resv02, '¥0', sizeof(resv02));
  rtn_code = dc_mcf_recvsync(DCMCFRST, DCNOFLAGS, leid, resv01, &rdata,
                             &rdataleng, sizeof(data_t), &time, 0);
  if( rtn_code != DCMCFRTN_00000 )

```

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

```
    abort();  
}
```

着信用 MHP のコーディング例

図 4-3 に示した着信用 MHP の例のコーディングを示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/c/ap3.c のファイルで提供しています。

```
/*  
 * TP1/NET/HSC HSC2手順 着信用MHP  
 */  
# include <dcmcf.h>  
# include <dcmhsc.h>  
# include <dcmhsc2.h>  
typedef struct {  
    DCLONG      mcf[2];  
    DCLONG      text_code;  
    char        data[2048];  
}data_t ;  
/*  
 * MHP サービス関数  
 */  
void rcvmhp()  
{  
char      leid[16];  
data_t    sdata;  
data_t    rdata;  
char      resv01[9];  
char      resv02[9];  
int       rtn_code;  
DCLONG    rdataleng;  
DCLONG    size;  
DCLONG    req,seg;  
DCLONG    inbufleng;  
DCLONG    *time;  
/*  
 * 接続情報通知の受信  
 */  
memset(resv01,'¥0',sizeof(resv01));  
memset(resv02,'¥0',sizeof(resv02));  
rtn_code = dc_mcf_receive(DCMCFRST,DCNOFLAGS,leid,resv01,&rdata,  
                          &rdataleng,sizeof(data_t),&time);  
if( rtn_code != DCMCFRTN_00000 )  
    abort();  
/*  
 * メッセージの受信 (テキスト)  
 */  
memset(resv01,'¥0',sizeof(resv01));  
memset(resv02,'¥0',sizeof(resv02));  
rtn_code = dc_mcf_recvsync(DCMCFRST,DCNOFLAGS,leid,resv01,&rdata,  
                          &rdataleng,sizeof(data_t),&time,0);  
if( rtn_code != DCMCFRTN_00000 )  
    abort();  
  
/***** メッセージの解析 *****/  
  
/*  
 * メッセージの受信 (EOT)
```



```

*/
memset(resv01,'¥0',sizeof(resv01));
memset(resv02,'¥0',sizeof(resv02));
rtn_code = dc_mcf_recvsync(DCMCFRST,DCNOFLAGS,leid,resv01,&rdata,
                          &rdata Leng,sizeof(data_t),&time,0);
if( rtn_code != DCMCFRTN_00000 )
  abort();
/*
*   メッセージの送信 (テキスト, EOT付き)
*/
sdata,text_code = DCMHSC_DLETEXT_EOT;
memset(resv01,'¥0',sizeof(resv01));
memset(resv02,'¥0',sizeof(resv02));

/*****   メッセージの編集   *****/

rtn_code = dc_mcf_sendsync(DCMCFEMI,DCNOFLAGS,leid,resv01,&sdata,
                          200,resv02,DCNOFLAGS,0);
if( rtn_code != DCMCFRTN_00000 )
  abort();
/*
*   メッセージの送信 (DLE・EOT)
*/
sdata,text_code = DCMHSC_DLE_EOT;
memset(resv01,'¥0',sizeof(resv01));
memset(resv02,'¥0',sizeof(resv02));
rtn_code = dc_mcf_sendsync(DCMCFEMI,DCNOFLAGS,leid,resv01,&sdata,
                          4,resv02,DCNOFLAGS,0);
if( rtn_code != DCMCFRTN_00000 )
  abort();
}

```

COBOL 言語

COBOL 言語を使用した UAP のコーディング例を示します。

発信用 SPP のコーディング例

図 4-2 に示した発信用 SPP の例のコーディングを示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/cobol/sv1.cbl のファイルで提供しています。

```

*
*   TP1/NET/HSC   HSC2手順   発信用SPP
*
IDENTIFICATION DIVISION.
PROGRAM-ID.   SNDSPP.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
01   SND.
    02   MSG-SEND           PIC X(8) VALUE 'SEND   '.
    02   STATUS-CODE       PIC X(5).
    02   FILLER            PIC X(3).
    02   MCFUSE01         PIC X(4) VALUE SPACE.
    02   MCFUSE02         PIC X(4) VALUE SPACE.
    02   MCFUSE03         PIC 9(8).

```

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

```

02 MCFUSE04      PIC 9(8).
02 MCFUSE05      PIC 9(9) COMP VALUE ZERO.
02 SEND-SEG      PIC X(4) VALUE 'EMI '.
02 MCFUSE06      PIC X(4) VALUE SPACE.
02 MCFUSE07      PIC X(4) VALUE SPACE.
02 MCFUSE08      PIC X(4) VALUE SPACE.
02 MCFUSE09      PIC X(8) VALUE SPACE.
02 MCFUSE10      PIC X(4) VALUE SPACE.
02 MCFUSE11      PIC X(8) VALUE SPACE.
02 MCFUSE12      PIC X(4) VALUE SPACE.
02 MCFUSE13      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE14      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE15      PIC X(1) VALUE SPACE.
02 MCFUSE16      PIC X(1) VALUE SPACE.
02 MCFUSE17      PIC X(14) VALUE LOW-VALUE.
01 CD1.
02 SENDSEG-CODE  PIC X(4) VALUE 'OUT '.
02 LEID          PIC X(8) VALUE 'hsc2le1 '.
02 MCFUSE18      PIC X(8) VALUE SPACE.
02 MCFUSE19      PIC X(8) VALUE SPACE.
02 MCFUSE20      PIC X(28) VALUE LOW-VALUE.
01 DATA1.
02 MSGSEG-LENG   PIC 9(9) COMP VALUE 11.
02 MCFUSE21      PIC X(8).
02 TEXT-CODE     PIC X(4) VALUE X'00030000'.
02 TERMINAL-ID   PIC X(16) VALUE 'Termid1'.
PROCEDURE DIVISION.
CALL 'CBLDCMCF' USING SND CD1 DATA1.  ** 回線接続要求 **

```

* 処理 1 *

*
EXIT PROGRAM.

発信用 MHP のコーディング例

図 4-2 に示した発信用 MHP の例のコーディングを示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/cobol/ap2.cbl のファイルで提供しています。

```

*
* TP1/NET/HSC  HSC2手順  発信用MHP
*
IDENTIFICATION DIVISION.
PROGRAM-ID.  SNDMHP.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
01 RCV1.
02 MSG-REC      PIC X(8) VALUE 'RECEIVE '.
02 STATUS-CODE1 PIC X(5).
02 FILLER       PIC X(3).
02 SEG-CODE     PIC X(4) VALUE 'FRST'.
02 RTN-CODE     PIC X(4) VALUE SPACE.
02 DAY-ID       PIC 9(8).
02 TIME-ID      PIC 9(8).

```

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

```

02 SEG-LENG          PIC 9(9) COMP VALUE 2048.
02 MCFUSE01         PIC X(4) VALUE SPACE.
02 MCFUSE02         PIC X(4) VALUE SPACE.
02 MCFUSE03         PIC X(4) VALUE SPACE.
02 MCFUSE04         PIC X(4) VALUE SPACE.
02 MCFUSE05         PIC X(8) VALUE SPACE.
02 MCFUSE06         PIC X(4) VALUE SPACE.
02 MCFUSE07         PIC X(8) VALUE SPACE.
02 MCFUSE08         PIC X(4) VALUE SPACE.
02 MCFUSE09         PIC 9(9) COMP VALUE ZERO.
02 MCFUSE10         PIC 9(9) COMP VALUE ZERO.
02 MCFUSE11         PIC X(1) VALUE SPACE.
02 MCFUSE12         PIC X(1) VALUE SPACE.
02 MCFUSE13         PIC X(14) VALUE LOW-VALUE.
01 CD1.
02 SEG-CODE1        PIC X(4) VALUE SPACE.
02 TERM-CODE        PIC X(8).
02 MCFUSE14         PIC X(8) VALUE SPACE.
02 MCFUSE15         PIC X(8) VALUE SPACE.
02 MCFUSE16         PIC X(28) VALUE LOW-VALUE.
01 DATA1.
02 MSGSEG-LENG1    PIC 9(9) COMP.
02 MCFUSE17         PIC X(4).
02 MCFUSE18         PIC X(2).
02 MCFUSE19         PIC X(1).
02 MCFUSE20         PIC X(1).
02 TEXT-CODE1      PIC X(4).
02 TERMINAL-ID     PIC X(16).
*
01 SND1.
02 MSG-SEND         PIC X(8) VALUE 'SENDSYNC'.
02 STATUS-CODE     PIC X(5).
02 FILLER           PIC X(3).
02 MCFUSE01         PIC X(4) VALUE SPACE.
02 MCFUSE02         PIC X(4) VALUE SPACE.
02 MCFUSE03         PIC 9(8).
02 MCFUSE04         PIC 9(8).
02 MCFUSE05         PIC 9(9) COMP VALUE ZERO.
02 SEND-SEG        PIC X(4) VALUE 'EMI '.
02 MCFUSE06         PIC X(4) VALUE SPACE.
02 MCFUSE07         PIC X(4) VALUE SPACE.
02 MCFUSE08         PIC X(4) VALUE SPACE.
02 MCFUSE09         PIC X(8) VALUE SPACE.
02 MCFUSE10         PIC X(4) VALUE SPACE.
02 MCFUSE11         PIC X(8) VALUE SPACE.
02 MCFUSE12         PIC X(4) VALUE SPACE.
02 MCFUSE13         PIC 9(9) COMP VALUE ZERO.
02 MCFUSE14         PIC 9(9) COMP VALUE ZERO.
02 MCFUSE15         PIC X(1) VALUE SPACE.
02 MCFUSE16         PIC X(1) VALUE SPACE.
02 MCFUSE17         PIC X(14) VALUE LOW-VALUE.
01 CD2.
02 SENDSEG-CODE    PIC X(4) VALUE 'OUT '.
02 TERM-ID         PIC X(8) VALUE 'hsc2le1 '.
02 MCFUSE18         PIC X(8) VALUE SPACE.
02 MCFUSE19         PIC X(8) VALUE SPACE.
02 MCFUSE20         PIC X(28) VALUE LOW-VALUE.
01 DATA2.
02 MSGSEG-LENG     PIC 9(9) COMP VALUE 128.
02 MCFUSE21         PIC X(8).
02 TEXT-CODE2      PIC X(4).
02 REC-MSGSEG      PIC X(124).
*
01 RCV2.
02 MSG-REC         PIC X(8) VALUE 'RCVSYNC'.

```

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

```

02 STATUS-CODE1 PIC X(5).
02 FILLER PIC X(3).
02 SEG-CODE PIC X(4) VALUE 'FRST'.
02 RTN-CODE PIC X(4) VALUE SPACE.
02 DAY-ID PIC 9(8).
02 TIME-ID PIC 9(8).
02 SEG-LENG PIC 9(9) COMP VALUE 2048.
02 MCFUSE01 PIC X(4) VALUE SPACE.
02 MCFUSE02 PIC X(4) VALUE SPACE.
02 MCFUSE03 PIC X(4) VALUE SPACE.
02 MCFUSE04 PIC X(4) VALUE SPACE.
02 MCFUSE05 PIC X(8) VALUE SPACE.
02 MCFUSE06 PIC X(4) VALUE SPACE.
02 MCFUSE07 PIC X(8) VALUE SPACE.
02 MCFUSE08 PIC X(4) VALUE SPACE.
02 MCFUSE09 PIC 9(9) COMP VALUE ZERO.
02 MCFUSE10 PIC 9(9) COMP VALUE ZERO.
02 MCFUSE11 PIC X(1) VALUE SPACE.
02 MCFUSE12 PIC X(1) VALUE SPACE.
02 MCFUSE13 PIC X(14) VALUE LOW-VALUE.
01 DATA3.
02 MSGSEG-LENG1 PIC 9(9) COMP.
02 MCFUSE17 PIC X(4).
02 MCFUSE18 PIC X(2).
02 MCFUSE19 PIC X(1).
02 MCFUSE20 PIC X(1).
02 TEXT-CODE3 PIC X(4).
02 DATA0 PIC X(2044).
*
01 RCV3.
02 MSG-REC PIC X(8) VALUE 'RCVSYNC'.
02 STATUS-CODE1 PIC X(5).
02 FILLER PIC X(3).
02 SEG-CODE PIC X(4) VALUE 'FRST'.
02 RTN-CODE PIC X(4) VALUE SPACE.
02 DAY-ID PIC 9(8).
02 TIME-ID PIC 9(8).
02 SEG-LENG PIC 9(9) COMP VALUE 2048.
02 MCFUSE01 PIC X(4) VALUE SPACE.
02 MCFUSE02 PIC X(4) VALUE SPACE.
02 MCFUSE03 PIC X(4) VALUE SPACE.
02 MCFUSE04 PIC X(4) VALUE SPACE.
02 MCFUSE05 PIC X(8) VALUE SPACE.
02 MCFUSE06 PIC X(4) VALUE SPACE.
02 MCFUSE07 PIC X(8) VALUE SPACE.
02 MCFUSE08 PIC X(4) VALUE SPACE.
02 MCFUSE09 PIC 9(9) COMP VALUE ZERO.
02 MCFUSE10 PIC 9(9) COMP VALUE ZERO.
02 MCFUSE11 PIC X(1) VALUE SPACE.
02 MCFUSE12 PIC X(1) VALUE SPACE.
02 MCFUSE13 PIC X(14) VALUE LOW-VALUE.
*
01 RCV4.
02 MSG-REC PIC X(8) VALUE 'RCVSYNC'.
02 STATUS-CODE1 PIC X(5).
02 FILLER PIC X(3).
02 SEG-CODE PIC X(4) VALUE 'FRST'.
02 RTN-CODE PIC X(4) VALUE SPACE.
02 DAY-ID PIC 9(8).
02 TIME-ID PIC 9(8).
02 SEG-LENG PIC 9(9) COMP VALUE 2048.
02 MCFUSE01 PIC X(4) VALUE SPACE.
02 MCFUSE02 PIC X(4) VALUE SPACE.
02 MCFUSE03 PIC X(4) VALUE SPACE.
02 MCFUSE04 PIC X(4) VALUE SPACE.

```

```

02 MCFUSE05      PIC X(8) VALUE SPACE.
02 MCFUSE06      PIC X(4) VALUE SPACE.
02 MCFUSE07      PIC X(8) VALUE SPACE.
02 MCFUSE08      PIC X(4) VALUE SPACE.
02 MCFUSE09      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE10      PIC 9(9) COMP VALUE ZERO.
02 MCFUSE11      PIC X(1) VALUE SPACE.
02 MCFUSE12      PIC X(1) VALUE SPACE.
02 MCFUSE13      PIC X(14) VALUE LOW-VALUE.
*
PROCEDURE DIVISION.
CALL 'CBLDCMCF' USING RCV1 CD1 DATA1.  ** 接続情報通知の受信 **
*           処理 1           *
MOVE X'00000002' TO TEXT-CODE2.
CALL 'CBLDCMCF' USING SND1 CD2 DATA2.  ** メッセージの送信 (テキスト ,
EOT付き) **
*           処理 2           *
CALL 'CBLDCMCF' USING RCV2 CD1 DATA3.  ** メッセージの受信 (テキスト)
**
*           処理 3           *
CALL 'CBLDCMCF' USING RCV3 CD1 DATA3.  ** メッセージの受信 (EOT) **
*           処理 4           *
CALL 'CBLDCMCF' USING RCV4 CD1 DATA3.  ** メッセージの受信 (DLE・EOT)
**
*           処理 5           *
EXIT PROGRAM.

```

着信用 MHP のコーディング例

図 4-3 に示した着信用 MHP の例のコーディングを示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/cobol/ap3.cbl のファイルで提供しています。

```

*
* TP1/NET/HSC  HSC2手順  着信用MHP
*
IDENTIFICATION DIVISION.
PROGRAM-ID. RCV1MHP.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
01 RCV1.
02 MSG-REC      PIC X(8) VALUE 'RECEIVE '.
02 STATUS-CODE1 PIC X(5).
02 FILLER       PIC X(3).

```

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

```

02 SEG-CODE          PIC X(4) VALUE 'FRST'.
02 RTN-CODE          PIC X(4) VALUE SPACE.
02 DAY-ID            PIC 9(8).
02 TIME-ID           PIC 9(8).
02 SEG-LENG          PIC 9(9) COMP VALUE 2048.
02 MCFUSE01          PIC X(4) VALUE SPACE.
02 MCFUSE02          PIC X(4) VALUE SPACE.
02 MCFUSE03          PIC X(4) VALUE SPACE.
02 MCFUSE04          PIC X(4) VALUE SPACE.
02 MCFUSE05          PIC X(8) VALUE SPACE.
02 MCFUSE06          PIC X(4) VALUE SPACE.
02 MCFUSE07          PIC X(8) VALUE SPACE.
02 MCFUSE08          PIC X(4) VALUE SPACE.
02 MCFUSE09          PIC 9(9) COMP VALUE ZERO.
02 MCFUSE10          PIC 9(9) COMP VALUE ZERO.
02 MCFUSE11          PIC X(1) VALUE SPACE.
02 MCFUSE12          PIC X(1) VALUE SPACE.
02 MCFUSE13          PIC X(14) VALUE LOW-VALUE.
01 CD1.
02 SEG-CODE1         PIC X(4) VALUE SPACE.
02 TERM-CODE         PIC X(8).
02 MCFUSE14          PIC X(8) VALUE SPACE.
02 MCFUSE15          PIC X(8) VALUE SPACE.
02 MCFUSE16          PIC X(28) VALUE LOW-VALUE.
01 DATA1.
02 MSGSEG-LENG1     PIC 9(9) COMP.
02 MCFUSE17          PIC X(4).
02 MCFUSE18          PIC X(2).
02 MCFUSE19          PIC X(1).
02 MCFUSE20          PIC X(1).
02 TEXT-CODE1       PIC X(4).
02 TERMINAL-ID      PIC X(16).
*
01 SND1.
02 MSG-SEND          PIC X(8) VALUE 'SENDSYNC'.
02 STATUS-CODE       PIC X(5).
02 FILLER            PIC X(3).
02 MCFUSE01          PIC X(4) VALUE SPACE.
02 MCFUSE02          PIC X(4) VALUE SPACE.
02 MCFUSE03          PIC 9(8).
02 MCFUSE04          PIC 9(8).
02 MCFUSE05          PIC 9(9) COMP VALUE ZERO.
02 SEND-SEG          PIC X(4) VALUE 'EMI '.
02 MCFUSE06          PIC X(4) VALUE SPACE.
02 MCFUSE07          PIC X(4) VALUE SPACE.
02 MCFUSE08          PIC X(4) VALUE SPACE.
02 MCFUSE09          PIC X(8) VALUE SPACE.
02 MCFUSE10          PIC X(4) VALUE SPACE.
02 MCFUSE11          PIC X(8) VALUE SPACE.
02 MCFUSE12          PIC X(4) VALUE SPACE.
02 MCFUSE13          PIC 9(9) COMP VALUE ZERO.
02 MCFUSE14          PIC 9(9) COMP VALUE ZERO.
02 MCFUSE15          PIC X(1) VALUE SPACE.
02 MCFUSE16          PIC X(1) VALUE SPACE.
02 MCFUSE17          PIC X(14) VALUE LOW-VALUE.
01 CD2.
02 SENDSEG-CODE     PIC X(4) VALUE 'OUT '.
02 TERM-ID          PIC X(8) VALUE 'hsc21e1 '.
02 MCFUSE18          PIC X(8) VALUE SPACE.
02 MCFUSE19          PIC X(8) VALUE SPACE.
02 MCFUSE20          PIC X(28) VALUE LOW-VALUE.
01 DATA2.
02 MSGSEG-LENG      PIC 9(9) COMP VALUE 128.
02 MCFUSE21          PIC X(8).
02 TEXT-CODE2       PIC X(4).

```

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

```

02 REC-MSGSEG PIC X(124).
*
01 RCV2.
02 MSG-REC PIC X(8) VALUE 'RECVSYNC' .
02 STATUS-CODE1 PIC X(5).
02 FILLER PIC X(3).
02 SEG-CODE PIC X(4) VALUE 'FRST' .
02 RTN-CODE PIC X(4) VALUE SPACE.
02 DAY-ID PIC 9(8).
02 TIME-ID PIC 9(8).
02 SEG-LENG PIC 9(9) COMP VALUE 2048.
02 MCFUSE01 PIC X(4) VALUE SPACE.
02 MCFUSE02 PIC X(4) VALUE SPACE.
02 MCFUSE03 PIC X(4) VALUE SPACE.
02 MCFUSE04 PIC X(4) VALUE SPACE.
02 MCFUSE05 PIC X(8) VALUE SPACE.
02 MCFUSE06 PIC X(4) VALUE SPACE.
02 MCFUSE07 PIC X(8) VALUE SPACE.
02 MCFUSE08 PIC X(4) VALUE SPACE.
02 MCFUSE09 PIC 9(9) COMP VALUE ZERO.
02 MCFUSE10 PIC 9(9) COMP VALUE ZERO.
02 MCFUSE11 PIC X(1) VALUE SPACE.
02 MCFUSE12 PIC X(1) VALUE SPACE.
02 MCFUSE13 PIC X(14) VALUE LOW-VALUE.
01 CD3.
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 DATA3.
02 MSGSEG-LENG1 PIC 9(9) COMP.
02 MCFUSE17 PIC X(4).
02 MCFUSE18 PIC X(2).
02 MCFUSE19 PIC X(1).
02 MCFUSE20 PIC X(1).
02 TEXT-CODE3 PIC X(4).
02 DATA0 PIC X(2044).
*
01 RCV3.
02 MSG-REC PIC X(8) VALUE 'RECVSYNC' .
02 STATUS-CODE1 PIC X(5).
02 FILLER PIC X(3).
02 SEG-CODE PIC X(4) VALUE 'FRST' .
02 RTN-CODE PIC X(4) VALUE SPACE.
02 DAY-ID PIC 9(8).
02 TIME-ID PIC 9(8).
02 SEG-LENG PIC 9(9) COMP VALUE 2048.
02 MCFUSE01 PIC X(4) VALUE SPACE.
02 MCFUSE02 PIC X(4) VALUE SPACE.
02 MCFUSE03 PIC X(4) VALUE SPACE.
02 MCFUSE04 PIC X(4) VALUE SPACE.
02 MCFUSE05 PIC X(8) VALUE SPACE.
02 MCFUSE06 PIC X(4) VALUE SPACE.
02 MCFUSE07 PIC X(8) VALUE SPACE.
02 MCFUSE08 PIC X(4) VALUE SPACE.
02 MCFUSE09 PIC 9(9) COMP VALUE ZERO.
02 MCFUSE10 PIC 9(9) COMP VALUE ZERO.
02 MCFUSE11 PIC X(1) VALUE SPACE.
02 MCFUSE12 PIC X(1) VALUE SPACE.
02 MCFUSE13 PIC X(14) VALUE LOW-VALUE.
*
01 SND2.
02 MSG-SEND PIC X(8) VALUE 'SENDSYNC' .
02 STATUS-CODE PIC X(5).

```

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

```
02 FILLER          PIC X(3).
02 MCFUSE01       PIC X(4) VALUE SPACE.
02 MCFUSE02       PIC X(4) VALUE SPACE.
02 MCFUSE03       PIC 9(8).
02 MCFUSE04       PIC 9(8).
02 MCFUSE05       PIC 9(9) COMP VALUE ZERO.
02 SEND-SEG       PIC X(4) VALUE 'EMI '.
02 MCFUSE06       PIC X(4) VALUE SPACE.
02 MCFUSE07       PIC X(4) VALUE SPACE.
02 MCFUSE08       PIC X(4) VALUE SPACE.
02 MCFUSE09       PIC X(8) VALUE SPACE.
02 MCFUSE10       PIC X(4) VALUE SPACE.
02 MCFUSE11       PIC X(8) VALUE SPACE.
02 MCFUSE12       PIC X(4) VALUE SPACE.
02 MCFUSE13       PIC 9(9) COMP VALUE ZERO.
02 MCFUSE14       PIC 9(9) COMP VALUE ZERO.
02 MCFUSE15       PIC X(1) VALUE SPACE.
02 MCFUSE16       PIC X(1) VALUE SPACE.
02 MCFUSE17       PIC X(14) VALUE LOW-VALUE.
*
PROCEDURE DIVISION.
CALL 'CBLDCMCF' USING RCV1 CD1 DATA1.  ** 接続情報通知の受信 **
*      処理 1      *
CALL 'CBLDCMCF' USING RCV2 CD3 DATA3.  ** メッセージの受信 (テキスト)
**
*      処理 2      *
CALL 'CBLDCMCF' USING RCV3 CD3 DATA3.  ** メッセージの受信 (EOT) **
*      処理 3      *
MOVE X'00010002' TO TEXT-CODE2.
CALL 'CBLDCMCF' USING SND1 CD2 DATA2.  ** メッセージの送信 (テキスト,
EOT付き) **
*      処理 4      *
MOVE X'00020000' TO TEXT-CODE3.
CALL 'CBLDCMCF' USING SND2 CD3 DATA3.  ** メッセージの受信 (DLE・EOT)
**
*      処理 5      *
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示します。

発信用 SPP のコーディング例

図 4-2 に示した発信用 SPP の例のコーディングを示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/dml/sv1.cbl のファイルで提供しています。


```

*
*   TP1/NET/HSC   HSC2手順   発信用SPP
*
IDENTIFICATION DIVISION.
PROGRAM-ID.   SNDSPP.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
01 SEND-AREA.
   02 SE-DATALENG          PIC 9(4) COMP VALUE 15.
   02 SE-RSV1              PIC 9(2).
   02 SE-TEXTCODE          PIC X(4) VALUE X'00030000'.
   02 SE-TERMIID          PIC X(16) VALUE 'TERMIID1'.
*
*           通信記述項           *
*
COMMUNICATION SECTION.
*
*           通信記述項 (送信)           *
*
CD SEND-INF
   FOR OUTPUT
   STATUS KEY IS           SE-STATUS
   SYMBOLIC TERMINAL IS SE-TERMIID.

PROCEDURE DIVISION.
*
*           回線接続要求           *
*
SEND      SEND-INF
          FROM SEND-AREA.
EXIT PROGRAM.

```

発信用 MHP のコーディング例

図 4-2 に示した発信用 MHP の例のコーディングを示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/dml/ap2.cbl のファイルで提供しています。

```

*
*   TP1/NET/HSC   HSC2手順   発信用のMHP
*
IDENTIFICATION DIVISION.
PROGRAM-ID.   SNDMHP.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
01 RECV-AREA.
   02 RE-DATALENG          PIC 9(4) COMP VALUE 508.
   02 RE-RSV1              PIC 9(2).
   02 RE-TEXTCODE          PIC X(4).
   02 RE-DATA              PIC X(500).
*

```

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

```

01 SEND-AREA.
   02 SE-DATALENG          PIC 9(4) COMP VALUE 100.
   02 SE-RSV1             PIC 9(2).
   02 SE-TEXTCODE        PIC X(4).
   02 SE-DATA            PIC X(100).
*
*       通信記述項           *
*
COMMUNICATION SECTION.
*
*       通信記述項 ( 接続情報通知の受信 )           *
*
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.
*
*       通信記述項 ( メッセージの受信 )           *
*
CD RECVSYNC-INF
  FOR I-O
  STATUS KEY IS          RE-STATUS
  SYMBOLIC TERMINAL IS RE-TERMNAM-S
  SYNCHRONOUS MODE IS SYNC
  MESSAGE DATE IS       RE-DATE
  MESSAGE TIME IS       RE-TIME.
*
*       通信記述項 ( メッセージの送信 )           *
*
CD SEND-INF
  FOR I-O
  STATUS KEY IS          SE-STATUS
  SYMBOLIC TERMINAL IS SE-TERMNAM
  SYNCHRONOUS MODE IS SYNC.
PROCEDURE DIVISION.
*
*       接続情報通知の受信           *
*
RECEIVE  RECV-INF
         FIRST SEGMENT
         INTO RECV-AREA.
*
*       メッセージの送信 ( テキスト , EOT付き )           *
*
MOVE RE-TERMNAM TO SE-TERMNAM.
MOVE X'00010002' TO SE-TEXTCODE.
SEND    SEND-INF
        FROM SEND-AREA.
*
*       メッセージの受信 ( テキスト )           *
*
MOVE RE-TERMNAM TO RE-TERMNAM-S.
RECEIVE RECVSYNC-INF
         FIRST SEGMENT
         INTO RECV-AREA.
*
*       メッセージの受信 ( EOT )           *
*
RECEIVE RECVSYNC-INF
         FIRST SEGMENT
         INTO RECV-AREA.
*

```

```
*          メッセージの受信 (DLE・EOT)          *
*
RECEIVE  RECVSYNC-INF
          FIRST SEGMENT
          INTO RECV-AREA.
EXIT PROGRAM.
```

着信用 MHP のコーディング例

図 4-3 に示した着信用 MHP の例のコーディングを示します。

また、このコーディング例を、/BeTRAN/examples/mcf/HSC/aplib/dml/ap3.cbl のファイルで提供しています。

```
*
* TP1/NET/HSC  HSC2手順  着信用のMHP
*
IDENTIFICATION DIVISION.
PROGRAM-ID.  RCMHP.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
01 RECV-AREA.
   02 RE-DATALENG          PIC 9(4) COMP VALUE 1028.
   02 RE-RSV1              PIC 9(2).
   02 RE-TEXTCODE         PIC X(4).
   02 RE-DATA              PIC X(1024).
*
01 SEND-AREA.
   02 SE-DATALENG          PIC 9(4) COMP VALUE 100.
   02 SE-RSV1              PIC 9(2).
   02 SE-TEXTCODE         PIC X(4).
   02 SE-DATA              PIC X(200).
*
*          通信記述項          *
*
COMMUNICATION SECTION.
*
*          通信記述項 (接続情報通知の受信)          *
*
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.
*
*          通信記述項 (メッセージの受信)          *
*
CD RECVSYNC-INF
FOR I-O
STATUS KEY IS          RE-STATUS
SYMBOLIC TERMINAL IS RE-TERMNAM-S
SYNCHRONOUS MODE IS SYNC
MESSAGE DATE IS        RE-DATE
MESSAGE TIME IS        RE-TIME.
*
```

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

```
*          通信記述項 (メッセージの送信)          *
*
CD SEND-INF
  FOR I=0
    STATUS KEY IS          SE-STATUS
    SYNCHRONOUS MODE IS SYNC
    SYMBOLIC TERMINAL IS SE-TERMNAM.
  PROCEDURE DIVISION.
*
*          接続情報通知の受信          *
*
RECEIVE  RECV-INF
         FIRST SEGMENT
         INTO RECV-AREA.
*
*          メッセージの受信 (テキスト)          *
*
MOVE RE-TERMNAM TO RE-TERMNAM-S.
RECEIVE  RECVSYNC-INF
         FIRST SEGMENT
         INTO RECV-AREA.
*
*          メッセージの受信 (EOT)          *
*
RECEIVE  RECVSYNC-INF
         FIRST SEGMENT
         INTO RECV-AREA.
*
*          メッセージの送信 (テキスト, EOT付き)          *
*
MOVE RE-TERMNAM TO SE-TERMNAM.
MOVE X'00010002' TO SE-TEXTCODE.
SEND     SEND-INF
         FROM SEND-AREA.
*
*          メッセージの送信 (DLE・EOT)          *
*
MOVE X'00020000' TO SE-TEXTCODE.
MOVE 8 TO SE-DATALENG.
SEND     SEND-INF
         FROM SEND-AREA.
EXIT PROGRAM.
```

5

ユーザOWNコーディング， MCF イベントインタフェース

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

5.1 ユーザOWNコーディングインタフェース

5.2 MCF イベントインタフェース

5.1 ユーザOWNコーディングインタフェース

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

TP1/NET/HSC で使用できる UOC を次に示します。

- 論理端末名称の決定 UOC（HSC2 手順（非同期モード）だけ）
- 入力メッセージ編集 UOC
- 出力メッセージ編集 UOC
- 送信メッセージ通番編集 UOC（HSC1 手順または HSC2 手順（非同期モード）だけ）

注

入力メッセージ編集 UOC では，受信した論理メッセージの編集のほかに，起動するアプリケーション名も決定できます。

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。また，UOC は C 言語で作成します。

5.1.1 論理端末名称の決定

HSC2 手順（非同期モード）の場合だけ使用できます。

論理端末名称決定 UOC は，ID を交換しない場合，着信による回線接続後，相手システムとのメッセージ送受信に使用する論理端末を決定します。

論理端末名称決定 UOC は，相手システムとの接続が完了してから最初の論理メッセージを受信した時点，かつ入力メッセージの編集 UOC で呼び出す前に起動されます。ただし，MCF イベント発生時と UAP からのアプリケーションプログラム起動時には，UOC は起動しません。

（1）論理端末名称の形式

論理端末名称決定 UOC で論理端末名称を決定する場合の形式は，UOC がリターン情報として返す論理端末名称格納領域の先頭から，'¥0' の手前までの 1～8 バイトの識別子です。先頭から 9 バイト目までに '¥0' がいないときは，入力論理端末名称形式不正とします。また，識別子と '¥0' の間に空白がある場合は，空白を '¥0' に置き換えます。

（2）UOC の障害処理

TP1/NET/HSC は，UOC からのリターンが次に示す場合，相手システムに対して DLE・EOT を送信して回線を切断します。

- DCMCF_UOC_MSG_NG でリターンしてきた場合
- 不正な論理端末名称でリターンしてきた場合
- 該当するコネクションに存在しない論理端末名称をリターンしてきた場合
- ほかの回線で使用中の論理端末名称をリターンしてきた場合

(3) OpenTP1 への組み込み方法

MCF メイン関数で呼び出すスタート関数 (dc_mcf_svstart) に，作成した UOC の関数アドレスを指定します。論理端末名称決定 UOC の関数アドレスは任意に決められます。UOC のオブジェクトファイルは，MCF メイン関数をコンパイル・リンケージすれば，TP1/NET/HSC の実行形式ファイルに結合されて実行できる状態になります。MCF メイン関数の詳細については，「8.2 MCF メイン関数の作成」を参照してください。

(4) 論理端末名称決定 UOC の処理の例

論理端末名称決定 UOC では，相手システムからのメッセージを受信すると，自システム内の未使用の論理端末を検索し，使用する論理端末を決定します。このような処理をする UOC のコーディング例を，「付録 E 論理端末名称決定 UOC のコーディング例」に示します。

5.1.2 論理端末名称決定 UOC インタフェース

論理端末名称決定 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 <dcpcf.h>
#include <dcpcf_uoc.h>
#include <dcmpcm.h>
#include <dcmhsc2.h>
DCLONG    uoc_func(dcpcf_uoc_min_n *parm)
```

K&R 版 C の形式

```
#include <dcpcf.h>
#include <dcpcf_uoc.h>
#include <dcmpcm.h>
#include <dcmhsc2.h>
DCLONG    uoc_func(parm)

dcpcf_uoc_min_n *parm ;
```

(2) 説明

uoc_func (論理端末名称決定 UOC) を呼び出すとき，MCF は次に示す所定のパラメタを parm に設定します。

(3) パラメタの内容

(a) dcmcf_uoc_min_n の内容

```

typedef struct {
    DCLONG pro_kind;           ...プロトコル種別
    char   le_name[9];        ...論理端末名称
    char   reserve1[7];       ...予備
    DCLONG rcv_prim;          ...受信サ - ビスプリミティブ
    dcmcf_uocbuff_list_n *buflist_adr;
                               ...受信バッファリストアドレス
    dcmcf_uocbuff_list_n *ebuflist_adr;
                               ...未使用
    char   aplname[9];        ...未使用
    char   reserve2[7];       ...予備
    char   *pro_indv_ifa;     ...dcmhs2_uoc_protを示すポインタ変数
    DCLONG rtn_detail;        ...詳細リターンコード
    char   reserve3[8];       ...予備
} dcmcf_uoc_min_n;

```

なお, dcmcf_uoc_min_n は, <dcmcfuoc.h> で定義されています。

(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;

```

なお, dcmcf_uocbuff_list_n は, <dcmcfuoc.h> で定義されています。

(c) dcmcf_uocbufinf_n (バッファ情報) の内容

```

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

```

なお, dcmcf_uocbufinf_n は, <dcmcfuoc.h> で定義されています。

(d) dcmhsc2_uoc_prot の内容

```

typedef struct {
    DCLONG le_num;           ...定義されている論理端末数
    DCLONG reserve[3];       ...予備
    dcmhs2_le_inf le_list[DCMHS2_UOC_LE_MAX];

```



```

} dcmhsc2_uoc_prot;
typedef struct {
    char    le_name[16];    ...論理端末名称
    DCLONG le_inf;        ...論理端末の状態
} dcmhs2_le_inf

```

(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_HSC2

HSC2 手順

le_name

ヌル文字が設定されます。

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

未使用の項目です。ヌル文字が設定されます。

aplname

未使用の項目です。ヌル文字が設定されます。

pro_indv_ifa

dcmhs2_uoc_prot を示すポインタ変数です。

(b) dcmcf_uocbuff_list_n (バッファリスト)

buf_num

バッファ情報の数が設定されます。

buf_array

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

(c) dcmcf_uocbufinf_n (バッファ情報)

buf_adr

バッファのアドレスが設定されます。

buf_size

5. ユーザOWNコーディング，MCF イベントインタフェース

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（セグメント長）が設定されます。

buff_id, buff_addr

MCF で使用するパラメタです。

(d) dcmhsc2_uoc_prot

le_num

定義されている論理端末の数が設定されます。

le_list

論理端末名称のリストが設定されます。

le_inf

論理端末の状態が設定されます。

DCMHS2_LE_USE

論理端末はほかの回線で使用されています。

DCMHS2_LE_NOUSE

論理端末は使用されていません。

(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 の範囲で指定してください。

(b) dcmhsc2_uoc_prot

le_name

論理端末名称を設定します。

(6) リターン値

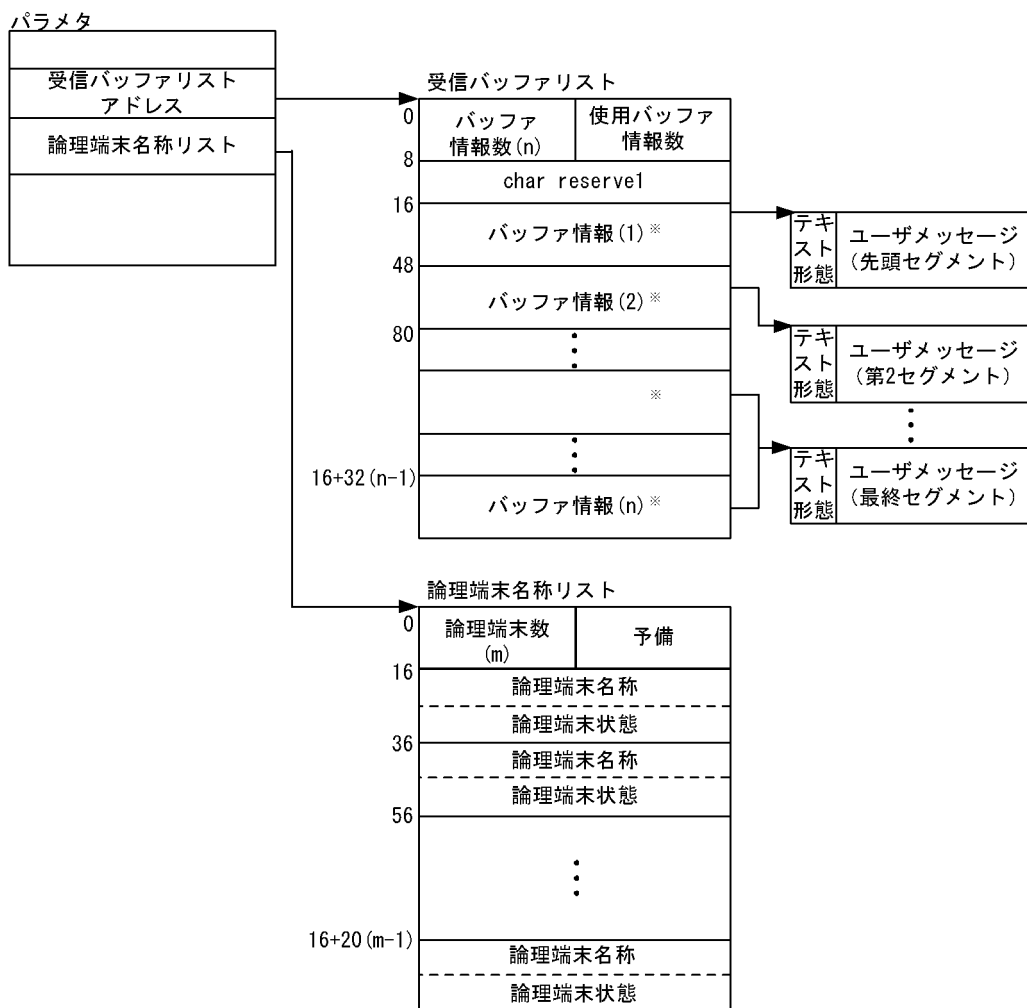
uoc_func() は次のコードでリターンしてください。

リターン値	意味
DCMCF_UOC_MSG_OK	正常リターン
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 5-1 UOC インタフェース用のパラメタとバッファの関係 (論理端末名称決定 UOC の場合)



注

バッファ情報は 32 バイトで, 次の形式をしています。

5. ユーザOWNコーディング, MCF イベントインタフェース



5.1.3 入力メッセージの編集とアプリケーション名の決定

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

UOC は、HSC1 手順または HSC2 手順（非同期モード）の場合、UAP を起動するメッセージの最終セグメントを受信すると起動します。HSC2 手順（同期モード）の場合は、相手システムとの接続が完了したとき、または最終セグメントを受信したときに起動します。ただし、MCF イベント発生時と UAP からのアプリケーションプログラム起動時には、UOC は起動しません。また、アプリケーション名は、相手システムとの回線が接続されたときだけに決定します。

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（mcftalccn -e）で、メッセージ編集用バッファグループ番号を定義します。

また、通常テキスト受信時、通常テキストのデータは EBCDIK コードのため、必要であれば、JIS8 コードに変換するように UOC を作成してください。

(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リスト形式で引き渡します。UOC では、これらのバッファを使用して、入力メッセージの編集ができます。

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

(2) アプリケーション名の決定

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

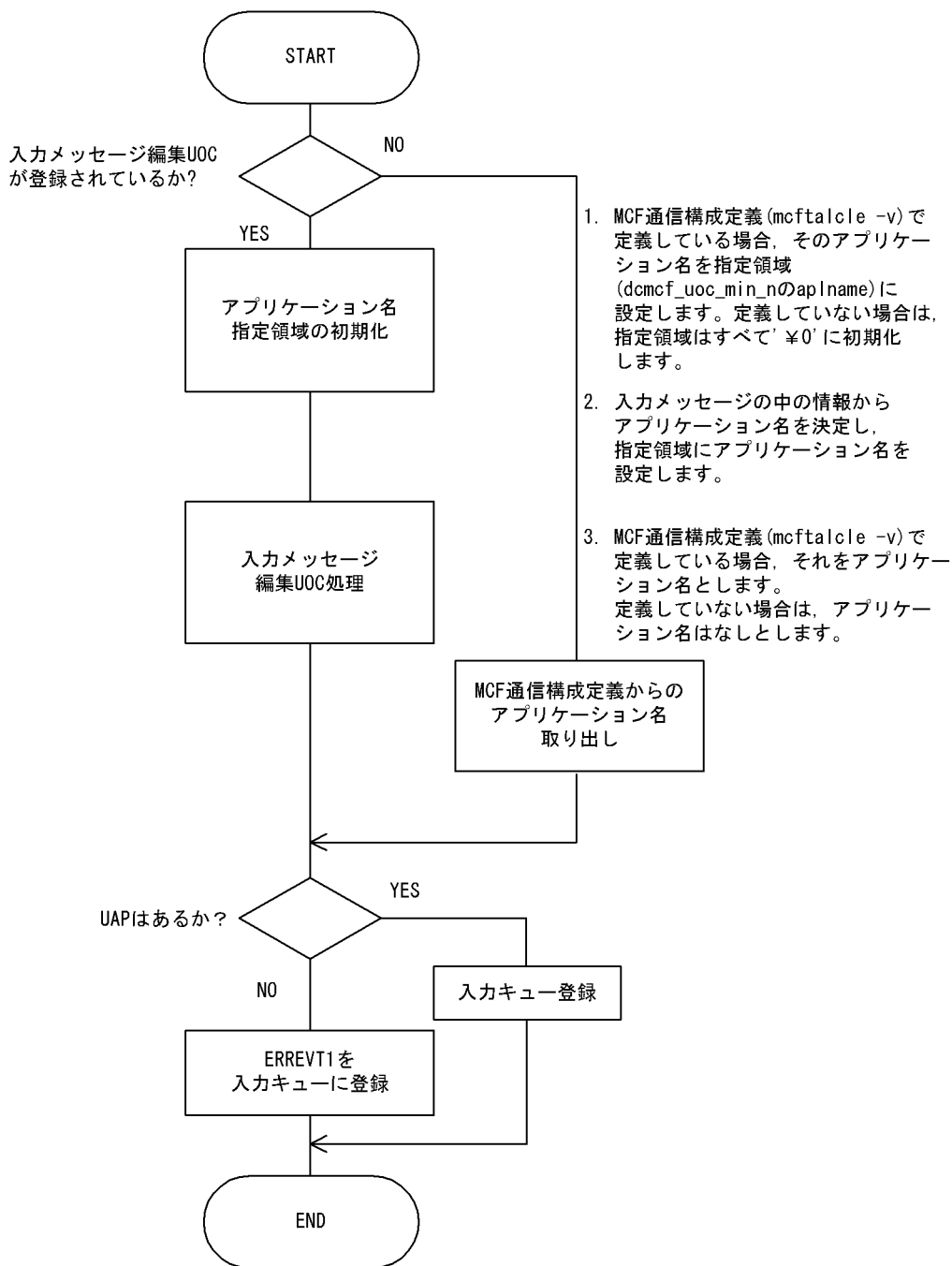
UOC でアプリケーション名を決定する場合, アプリケーション名の形式は, アプリケーション名格納領域の先頭から, '¥0' の手前までの 1 ~ 8 バイトの英数字です。先頭から 9 バイト目までに '¥0' がいないときは, アプリケーション名を不正とし, 不正アプリケーション名検出通知イベント (ERREVT1) を起動します。

UOC が登録されていない, または登録されていてもアプリケーション名を決定しない場合, MCF 通信構成定義で指定したアプリケーション名を採用します。MCF 通信構成定義でも指定されていないときは, 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) を起動します。

HSC2 手順 (同期モード) の場合, 相手システムとの回線接続時に, 起動するアプリケーション名を決定します。回線接続中のメッセージ受信中には, この UOC でアプリケーション名の変更はできません。

アプリケーション名の決定の処理を次の図に示します。

図 5-2 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, TP1/NET/HSC は相手システ

ムには通知しません。MCF はメッセージログを出力し、障害通知イベント (CERREVT) を起動します。

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

(4) UOC パラメタ不正の場合の処理

UOC で設定したパラメタに不正があった場合、MCF はメッセージログを出力し、障害通知イベント (CERREVT) を起動します。

(5) OpenTP1 への組み込み方法

スタート関数 (dc_mcf_svstart) を発行する MCF メイン関数に、作成した UOC の関数アドレスを設定します。入力メッセージ編集 UOC の関数アドレスは任意に決められます。UOC 関数をコンパイルして生成した UOC オブジェクトファイルを、UOC 関数を登録した MCF メイン関数と結合して、TP1/NET/HSC の実行形式プログラムを生成します。MCF メイン関数の詳細については、「8.2 MCF メイン関数の作成」を参照してください。

5.1.4 入力メッセージ編集 UOC インタフェース

入力メッセージ編集 UOC は、次に示す形式で呼び出します。

(1) 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

(2) 説明

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

(3) パラメタの内容

(a) dcmcf_uoc_min_n の内容

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

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

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

(c) dcmcf_uocbufinf_n (バッファ情報) の内容

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

(4) MCF が値を設定する項目

(a) dcmcf_uoc_min_n

pro_kind

プロトコル種別として, 次のどちらかの値が設定されます。

DCMCF_UOC_PRO_HSC1

HSC1 手順

DCMCF_UOC_PRO_HSC2

HSC2 手順

le_name

メッセージを入力した論理端末名称が設定されます。

rcv_prim

受信サービスプリミティブとして, 次の値が設定されます。

DCMCF_UOC_RCV_BRD

一方送信メッセージの受信

DCMCF_UOC_RCV_REP_RE

同期型的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 (mcftalccn -e) を省略した場合, ebuflist_adr にはヌル文字が設定されます。

aplname

MCF 通信構成定義 (mcftalcle -v) で指定したアプリケーション名が設定されます。

pro_indv_ifa

MCF で使用するパラメタです。

(b) dcmcf_uocbuff_list_n (バッファリスト)

buf_num

バッファ情報の数が設定されます。

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 (セグメント長) が設定されます。

buff_id, buff_addr

MCF で使用するパラメタです。

5. ユーザ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

aplname

UOC で決定したアプリケーション名を設定します。

rtn_detail

詳細リターンコードを設定します。

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

(b) dcmcf_uocbuff_list_n (バッファリスト)

used_buf_num

使用したバッファ情報の数を設定します。

(c) dcmcf_uocbufinf_n (バッファ情報)

seg_size

バッファの使用長を設定します。

(6) リターン値

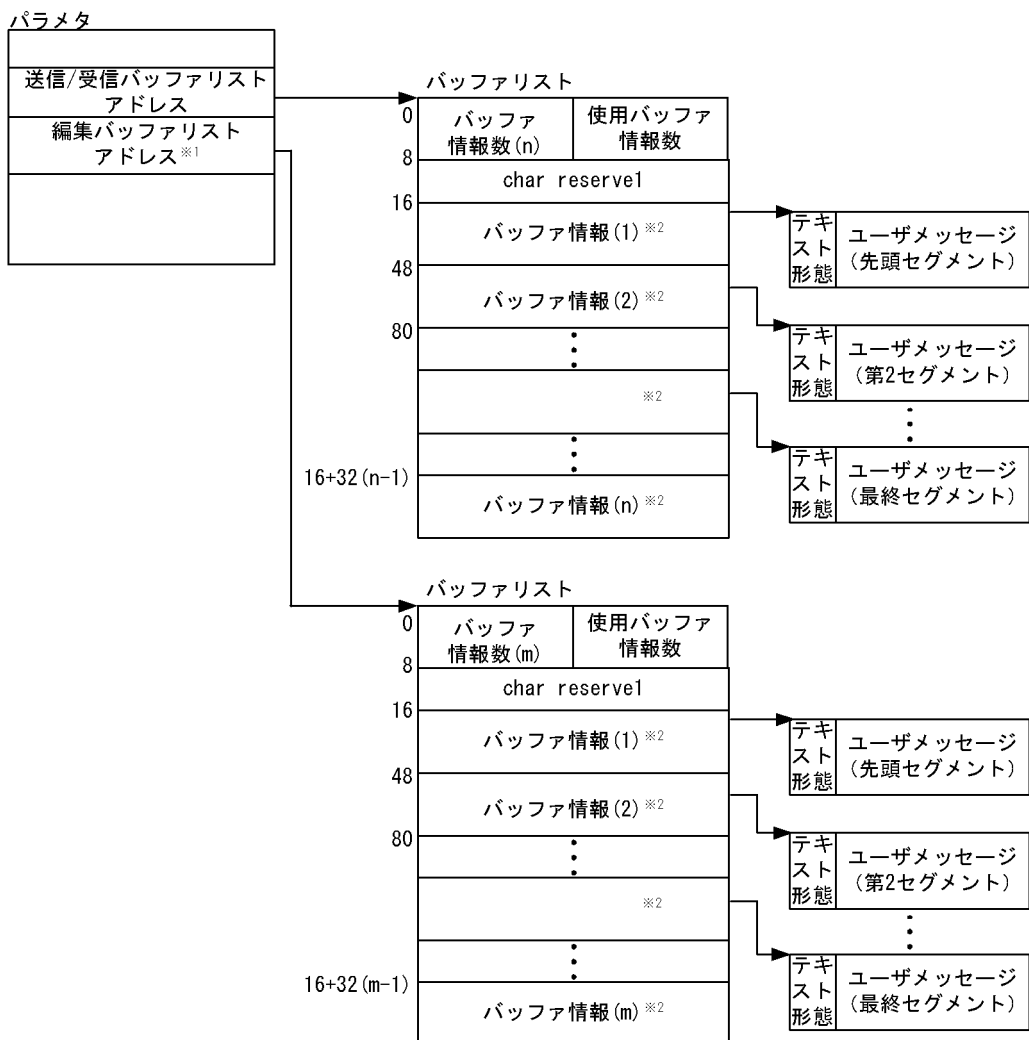
uoc_func() は次のコードでリターンしてください。

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

(7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係を次の図に示します。

図 5-3 UOC インタフェース用のパラメタとバッファの関係



注 1

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

注 2

バッファ情報は次の形式をしています。

5. ユーザOWNコーディング，MCF イベントインタフェース



5.1.5 出力メッセージの編集

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

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

また，通常テキスト送信時，通常テキストのデータは EBCDIK コードで作成する，JIS8 コードに変換するように UOC を作成してください。

(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リスト形式で引き渡します。UOC では，これらのバッファを使用して，出力メッセージの編集処理ができます。

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

(2) UOC エラーリターン処理

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

(3) UOC パラメタ不正の場合の処理

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

(4) OpenTP1 への組み込み方法

入力メッセージ編集 UOC の組み込み方法と同じです。「5.1.3 (5) OpenTP1 への組み込み方法」を参照してください。

5.1.6 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 <dcpcf.h>
#include <dcpcf_uoc.h>
DCLONG    uoc_func(dcpcf_uoc_mout_n *parm)
```

K&R 版 C の形式

```
#include <dcpcf.h>
#include <dcpcf_uoc.h>
DCLONG    uoc_func(parm)

dcpcf_uoc_mout_n *parm ;
```

(2) 説明

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

(3) パラメタの内容

(a) dcpcf_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
    dcpcf_uocbuff_list_n *buflist_adr;
                                ...送信バッファリストアドレス
    dcpcf_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
```

5. ユーザOWNコーディング, MCF イベントインタフェース

```
char    reserve2[1];        ... 予備
char    *pro_indv_ifa;      ...MCF使用領域
DCLONG  rtn_detail;        ...詳細リターンコード
char    reserve3[20];      ... 予備
} dcmcf_uoc_mout_n;
```

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

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

(4) MCF が値を設定する項目

(a) dcmcf_uoc_mout_n

pro_kind

プロトコル種別として, 次のどちらかの値が設定されます。

DCMCF_UOC_PRO_HSC1

HSC1 手順

DCMCF_UOC_PRO_HSC2

HSC2 手順

le_name

メッセージを入力した論理端末名称が設定されます。

buflist_adr

送信用バッファリストのアドレスが設定されます。

ebuflist_adr

編集用バッファリストのアドレスが設定されます。

メッセージ編集用バッファが未定義の場合, つまり, MCF 通信構成定義 (mcftalccn コマンド -e) を省略した場合, ebuflist_adr にはヌル文字が設定されます。

output_no

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

msg_type

メッセージ種別として, 次の値が設定されます。

'n'

一般一方送信メッセージ

'p'

優先一方送信メッセージ (HSC1 手順の場合だけ設定されます)

outputno_flag

メッセージ出力通番の有効フラグとして, 次のどちらかの値が設定されます。

DCMCF_UOC_OUTPUTNO_OK

メッセージ出力通番を有効にします。

DCMCF_UOC_OUTPUTNO_NG

メッセージ出力通番を無効にします。

resend_flag

再送メッセージフラグとして，次のどちらかの値が設定されます。HSC1 手順の場合だけ設定されます。

'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)

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

(5) ユーザが値を設定する項目

(a) dcmcf_uoc_mout_n

rtn_detail

詳細リターンコードを設定します。

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

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

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

(6) リターン値

uoc_func() は次のコードでリターンしてください。

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

(7) パラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係は，入力メッセージ編集の場合と同じです。「5.1.4(7) パラメタとバッファの関係」を参照してください。

5.1.7 送信メッセージの通番編集

(1) 通番編集

HSC1 手順または HSC2 手順 (非同期モード) の場合に使用できる UOC です。

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

送信メッセージの通番編集 UOC を起動する場合, メッセージ送信の関数で, 出力通番を付けるように設定してください。UOC は, UAP が send 関数または resend 関数を発行したときに, MCF によって起動されます。

(2) OpenTP1 への組み込み方法

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

(a) 形式

ANSI C, C++ の形式

```
#include<dcpcf.h>

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

K&R 版 C の形式

```
#include<dcpcf.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。

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

(d) メイン関数への登録例

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

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

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

5.1.8 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, 次を示す形式で, send_uoc 関数として作成します。なお, UOC の関数名称はユーザの任意です。

(1) 形式

ANSI C, C++ の形式

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

K&R 版 C の形式

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

(2) MCF が値を設定する項目

flags

送信メッセージの通番編集 UOC がいつ呼ばれたかが設定されます。

5. ユーザOWNコーディング，MCF イベントインタフェース

DCMCF_SEND_DML

メッセージを送信する関数または命令文が呼び出されたときを意味します。

termname

送信先の論理端末名称が設定されます。

sendno

送信メッセージの通番が設定されます。

sendid

送信するメッセージ種別が設定されます。

DCMCF_SEND_PRIO

優先の一方送信メッセージ

DCMCF_SEND_NORM

一般の一方送信メッセージ

dataleng

送信メッセージ長が設定されます。

senddata

セグメントの内容が設定されます。

(3) リターン値

リターン値	意味
DC_OK	正常に終了しました。

5.1.9 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作成してください。また、UOC の中で関数の再帰呼び出しはしないでください。

(2) UOC で使用できる関数

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

メモリ操作をする関数

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

時間取得関数

(3) UOC の異常処理

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

(4) UOC の実行タイミング

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

5.2 MCF イベントインタフェース

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

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

5.2.1 MCF イベントの種類

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

表 5-1 TP1/NET/HSC が通知する MCF イベントの種類

MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP の処理の例
不正アプリケーション名検出通知イベント	ERREVT1	メッセージのアプリケーション名が MCF アプリケーション定義にありません。	該当するアプリケーション名がなかったことを報告します。
メッセージ廃棄通知イベント	ERREVT2	次の理由で受信メッセージを破棄しました。 <ul style="list-style-type: none"> 入力キューに障害が発生しました。 MHP のサービス，サービスグループ，またはアプリケーションが閉塞しました。 MHP のセグメント受信関数に，セグメントを渡す前に MHP の異常終了が発生しました。 アプリケーション名に相当する MHP のサービスがありません。 アプリケーションの即時起動時に障害が発生しました。 MHP のアプリケーション，サービスグループがセキュア状態です。 	メッセージを廃棄したことを報告します。
UAP 異常終了通知イベント	ERREVT3	MHP のセグメント受信関数に，セグメントを渡したあとに MHP の異常終了 が発生しました。	UAP 異常終了時の対処障害メッセージを送信します。
タイマ起動メッセージ廃棄通知イベント	ERREVT4	アプリケーションのタイマ起動時に障害が発生しました。	メッセージを廃棄したことを報告します。

MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP の処理の例
未処理送信メッセージ廃棄通知イベント	ERREVT	次の理由で未処理送信メッセージを破棄しました。 <ul style="list-style-type: none"> MCF の正常終了処理時に、未処理送信メッセージの滞留時間監視の時間切れ (タイムアウト) が発生しました。 運用コマンド (mcftdlqle) の入力によって、出力キューが削除されました。 	未処理送信メッセージを廃棄したことを報告します。
送信完了通知イベント	SCMPEVT	メッセージの送信を正常に完了しました。	HSC1 手順または HSC2 手順 (非同期モード) の場合、送信の完了を確認して任意の処理ができます。
送信障害通知イベント	SERREVT	メッセージの送信に障害が発生しました。	HSC1 手順または HSC2 手順 (非同期モード) の場合、送信障害が発生したことを報告して、任意の処理ができます。
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	メッセージの送受信ができる状態になりました。	HSC1 手順の場合、メッセージ送受信の準備ができたことを報告します。 HSC2 手順の場合、コネクションが確立したことを報告します。
	CCLSEVT	メッセージの送受信ができない状態になりました。	HSC1 手順の場合、メッセージ送受信ができないことを報告します。 HSC2 手順の場合、コネクションが解放されたことを報告します。

注

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

5.2.2 MCF イベント通知時のセグメント構成

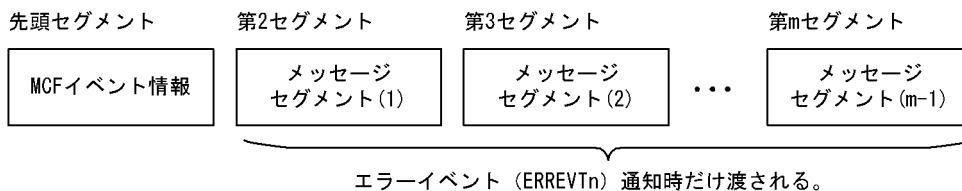
MCF イベントを MHP に通知する場合、先頭セグメントに MCF イベント情報を設定し

5. ユーザOWNコーディング，MCF イベントインタフェース

まず、エラーイベント (ERREVTn) の場合は、第 2 セグメント以降に処理できなかったメッセージセグメントを最終セグメントまで設定します。

MCF イベント通知時のセグメント構成を次の図に示します。

図 5-4 MCF イベント通知時のセグメント構成



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

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

このため、それ以降の項目の位置にずれがあります。「5.2.4 MCF イベント情報の形式 (COBOL 言語)」のエラーイベントの表では ERREVT1, ERREVT2, ERREVT3, および ERREVT4 のバッファ形式別に位置 (バイト) を分けて説明しています。

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

5.2.3 MCF イベント情報の形式 (C 言語)

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

エラーイベント (ERREVTn), SCMPEVT, および SERREVT で使用する構造体は、<dcmcf.h> で定義してあります。

(1) MCF イベントの共通ヘッダ

(a) 形式

```
struct dc_mcf_evtheader {
    char    mcfevt_name[9];    ...MCF イベント情報コード
    char    le_name[16];      ...入力元論理端末名称
                                (ERREVT1, ERREVT2, ERREVT3の場合)
                                出力先論理端末名称
                                (ERREVT4, SCMPEVT, SERREVTの場合)
    char    cn_name[9];       ...コネクション名
    unsigned char format_kind; ...MCF使用領域
    char    reserve01;        ...予備
```

```

    DCLONG time;                ...メッセージ入力時刻
};

```

(b) MCF イベントとして設定される項目

le_name

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

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

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

ERREVT4, SCMP EVT, または SERREVT では, メッセージを出力する論理端末名称が設定されます。

CERREVT, COPNEVT, または CCLSEVT では無効です。

cn_name

コネクション名が設定されます。

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

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

time

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

(2) ERREVT1

(a) 形式

```

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

```

(b) MCF イベントとして設定される項目

ap_name

次を示すどちらかが設定されます。

5. ユーザOWNコーディング，MCF イベントインタフェース

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

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

(3) ERREVT2

(a) 形式

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

(b) MCF イベントとして設定される項目

ap_name

エラーになった UAP のアプリケーション名が設定されます。

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

MHP 以外から送信された場合は，ヌル文字が設定されます。

reason_code

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

(4) ERREVT3

(a) 形式

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


(b) MCF イベントとして設定される項目

ap_name

異常が発生した MHP のアプリケーション名が設定されます。

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

MHP 以外から送信された場合は, ヌル文字が設定されます。

service_name

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

serv_grp_name

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

bid

トランザクションのブランチ ID が設定されます。

(5) ERREVTA

(a) 形式

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

(b) MCF イベントとして設定される項目

ap_name

正常終了したメッセージのアプリケーション名が設定されます。

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

MHP 以外から送信された場合は, ヌル文字が設定されます。

(6) SCMP EVT

(a) 形式

```
struct dc_mcf_scmpevt_type {
    struct dc_mcf_evtheader  evtheader ;
                                ...MCFイベント共通ヘッダ
    DCLONG  output_no ;      ...出力通番
    char  map_name[9] ;      ...MCF使用領域
};
```

5. ユーザOWNコーディング, MCF イベントインタフェース

```
char    msg_type ;           ...メッセージ種別
char    reason_code ;       ...起動理由
char    reserve01[9] ;      ...予備
char    pro_indv_inf[16] ;   ...プロトコル個別情報
char    reserve02[16];      ...予備
} ;
```

(b) MCF イベントとして設定される項目

output_no

出力通番が設定されます。通番がない場合は, '-1' が設定されます。

msg_type

メッセージ種別として, 次の値が設定されます。

n

一般一方送信メッセージ

p

優先一方送信メッセージ

reason_code

起動理由として, 次の値が設定されます。

メッセージが正常に送信されました。

pro_indv_inf

ヌル文字が設定されます。

(7) SERREVT

(a) 形式

```
struct dc_mcf_serrevert_type {
    struct dc_mcf_evtheader evtheader ;           ...MCFイベント共通ヘッダ
    DCLONG output_no ;                             ...出力通番
    char    map_name[9] ;                          ...MCF使用領域
    char    msg_type ;                             ...メッセージ種別
    char    send_error_code ;                      ...送信エラーコード
    unsigned char trouble_code;                   ...障害コード
    unsigned char detail_info;                   ...詳細情報
    unsigned char error_code;                    ...エラーコード
    unsigned char error_subcode;                 ...エラーサブコード
    char reserve01[5] ;                            ...予備
    char pro_indv_inf[16] ;                       ...プロトコル個別情報
    char reserve02[16];                          ...予備
} ;
```

(b) MCF イベントとして設定される項目

output_no

出力通番が設定されます。通番がない場合は，'-1' が設定されます。

msg_type

メッセージ種別として，次の値が設定されます。

n

一般一方送信メッセージ

p

優先一方送信メッセージ

send_error_code

送信エラーコードとして，'E' が設定されます。

trouble_code

障害コードとして，次の値が設定されます。

DCMCF_SERR_SEND

送信エラー

DCMCF_SERR_LOGICAL

論理エラー

detail_info

詳細情報として，次の値が設定されます。

DCMCF_SERR_ERROR

エラーによる送信障害

DCMCF_SERR_DCTCN

mcftdctcn コマンドによる送信障害

DCMCF_SERR_DCTLE

mcftdetle コマンドによる送信障害

error_code

ヌル文字が設定されます。

error_subcode

ヌル文字が設定されます。

pro_indv_inf

ヌル文字が設定されます。

(8) CERREVT

(a) 形式

typedef struct {

5. ユーザOWNコーディング, MCF イベントインタフェース

```
struct dc_mcf_evtheader header ;
                                ...MCFイベント共通ヘッダ
DCLONG err_fact ;                ...障害要因コード(4バイト)
DCLONG err_reason1 ;            ...理由コード1(4バイト)
DCLONG err_reason2 ;            ...理由コード2(4バイト)
DCLONG recover_code ;           ...回復可否識別
DCLONG retry_code ;             ...自動再確立有無識別
} dcmpcm_cerrevt ;
typedef struct {
    dcmpcm_cerrevt comheader     ...プロトコル共通ヘッダ
    char reserve1[36] ;          ...予備
} dcmhsc_cerrevt ;
```

なお, dcmpcm_cerrevt は, <dcmpcm.h> で定義されています。<dcmhsc1.h> または <dcmhsc2.h> の前に <dcmpcm.h> を取り込んでおいてください。また, dc_mcf_evtheader は, <dcmcf.h> で定義されています。<dcmcf.h>, <dcmpcm.h>, <dcmhsc1.h> または <dcmhsc2.h> の順に取り込んでください。

(b) MCF イベントとして設定される項目

err_fact

CERREVT の障害要因コードとして, 次の値が設定されます。

(00000030)₁₆

コネクション障害発生

(00000031)₁₆

論理端末障害発生

err_reason1, err_reason2

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

recover_code

障害が回復できるかどうか, 次の値で設定されます。

(00000000)₁₆

回復可能

(00000010)₁₆

回復不可能

retry_code

障害発生後, 自動再確立するかどうか, 次の値で設定されます。

(00000000)₁₆

自動再確立をします。

(00000010)₁₆

自動再確立をしません。

(9) COPNEVT , CCLSEVT

(a) 形式

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

なお，dcmPCM_statevt は，<dcmhsc1.h> または <dcmhsc2.h> で定義されています。
<dcmhsc1.h> または <dcmhsc2.h> の前に <dcmcf.h> を取り込んでおいてください。

(b) MCF イベントとして設定される項目

ありません。

5.2.4 MCF イベント情報の形式 (COBOL 言語)

COBOL 言語の場合はセグメントの並びとして渡されます。

COBOL 言語の UAP の場合，MCF イベントの内容を表 5-2 ~ 表 5-9 に示します。

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベント コード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT1 を示す '1' が設定されます。
入力元論理端末 名称	12	8	8	英数字	メッセージを入力した論理端末名称です。
予備	20	16	20	-	-
アプリケーション 名	40	36	8	英数字	次に示すどちらかが設定されます。 • 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	-	-

5. 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	64	60	8	-	-
コネクション名	72	68	8	英数字	コネクション名です。
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進数字	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
予備	112	108	16	-	-

(凡例)

- : 該当しません。または，使用されません。

表 5-3 COBOL 言語の MCF イベント情報の内容 (ERREVT2)

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

5. 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	20	16	20	-	-
アプリケーション名	40	36	8	英数字	エラーになった UAP のアプリケーション名です。
予備	48	44	8	-	-
予備	56	52	8	-	-
予備	64	60	8	-	-
コネクション名	72	68	8	英数字	コネクション名です。次に示す場合には，'*' が設定されます。 <ul style="list-style-type: none"> • SPP がアプリケーション起動をした MHP で障害が発生した場合 • 上記の障害発生時に MCF イベントとして起動された MHP によって，さらにアプリケーション起動された MHP で，障害が発生した場合
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進数字	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
理由コード	112	108	4	外部 10 進数字	理由コードが設定されます。
予備	116	112	12	-	-

(凡例)

-：該当しません。または，使用されません。

注

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

5. ユーザOWNコーディング，MCF イベントインタフェース

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベントコード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVT3 を示す '3' が設定されます。
入力元論理端末名称	12	8	8	英数字	メッセージを入力した論理端末名称です。次に示す場合は， '*' が設定されます。 <ul style="list-style-type: none"> • SPP がアプリケーション起動をした MHP で，障害が発生した場合 • 上記の障害発生時に MCF イベントとして起動された MHP が，さらにアプリケーション起動をした MHP で，障害が発生した場合
予備	20	16	20	-	-
予備	40	36	8	-	-
マップ名	48	44	8	-	MCF が使用します。
アプリケーション名	56	52	8	英数字	異常が発生したメッセージのアプリケーション名です。
予備	64	60	8	-	-
コネクション名	72	68	8	英数字	コネクション名です。次に示す場合は， '*' が設定されます。 <ul style="list-style-type: none"> • SPP がアプリケーション起動をした MHP で，障害が発生した場合 • 上記の障害発生時に MCF イベントとして起動された MHP が，さらにアプリケーション起動をした MHP で，障害が発生した場合
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進数字	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
メッセージが入力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
予備	112	108	16	-	-
サービス名	128	124	31	英数字	異常が発生した UAP のアプリケーション名に対応するサービス名です。
予備	159	155	1	-	-
サービスグループ名	160	156	31	英数字	異常が発生した UAP のサービスグループ名です。
予備	191	187	1	-	-
トランザクション ID (BID)	192	188	36	2 進数字	異常が発生したトランザクションの BID です。
予備	228	224	28	-	-

(凡例)

- : 該当しません。または、使用されません。

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
予備 (形式 1 のときだけ)	0	-	2	-	-
予備 (形式 1 のときだけ)	2	-	2	-	-
エラーイベントコード	4	0	3	英数字	'ERR' が設定されます。
	7	3	3	-	-
	10	6	2	英数字	ERREVTA を示す 'A ' が設定されます。
出力先論理端末名称	12	8	8	英数字	メッセージを出力する論理端末名称です。
予備	20	16	20	-	-
予備	40	36	8	-	-
マップ名	48	44	8	-	MCF が使用します。

5. 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	正常終了したメッセージのアプリケーション名です。 MHP から送信されたメッセージの場合設定されます。MHP 以外から送信された場合は空白が設定されます。
予備	64	60	8	-	-
コネクション名	72	68	8	英数字	コネクション名です。
予備	80	76	16	-	-
メッセージが入力された日付	96	92	8	外部 10 進数字	端末入力メッセージを入力した日付です。YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	104	100	8	外部 10 進数字	端末入力メッセージを入力した時刻です。HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
予備	112	108	16	-	-
予備	128	124	31	-	-
予備	159	155	1	-	-
予備	160	156	31	-	-
予備	191	187	1	-	-
予備	192	188	36	-	-
予備	228	224	28	-	-

(凡例)

- : 該当しません。または，使用されません。

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
長さ (形式 1 のときだけ)	0	-	2	-	MCF が使用します。
予備 (形式 1 のときだけ)	2	-	2	-	-
イベントコード	4	0	8	英数字	「SCMPEVT」が設定されます。

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
出力先論理端末 名称	12	8	8	英数字	メッセージの送信が完了した論理端末 名称です。
予備	20	16	8	-	-
コネクション名	28	24	8	英数字	コネクション名です。
メッセージが入 力された日付	36	32	8	外部 10 進数字	SCMPEVT が通知された日付です。 YYYYMMDD の形式です。 YYYY : 西暦の年 MM : 月 DD : 日
メッセージが入 力された時刻	44	40	8	外部 10 進数字	SCMPEVT が通知された時刻です。 HHMMSS00 の形式です。 HH : 時 MM : 分 SS : 秒 00 は固定です。
出力通番	52	48	4	2 進数 字	出力通番です。出力通番がない場合は、 (FFFFFFFF) ₁₆ が設定されます。
出力マップ名	56	52	8	英数字	MCF が使用します。
メッセージ種別	64	60	1	英数字	メッセージ種別が設定されます。 n : 一般一方送信メッセージ p : 優先一方送信メッセージ
理由コード	65	61	1	英数字	理由コードが設定されます。 : 送信が正常に完了しました。
予備	66	62	9	-	-
プロトコル個別 情報	75	71	16	英数字	MCF が使用します。
予備	91	87	19	-	-

(凡例)

- : 該当しません。または、使用されません。

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

項目	位置 (バイト)		長さ (バイト)	属性	内容
	形式 1	形式 2			
長さ (形式 1 の ときだけ)	0	-	2	-	MCF が使用します。
予備 (形式 1 の ときだけ)	2	-	2	-	-
イベントコード	4	0	8	英数字	「SERREVT」が設定されます。

5. 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			
出力先論理端末名称	12	8	8	英数字	送信障害となった論理端末名称です。
予備	20	16	8	-	-
コネクション名	28	24	8	英数字	コネクション名です。
メッセージが入力された日付	36	32	8	外部 10 進数字	SERREVT が通知された日付です。 YYYYMMDD の形式です。 YYYY：西暦の年 MM：月 DD：日
メッセージが入力された時刻	44	40	8	外部 10 進数字	SERREVT が通知された時刻です。 HHMMSS00 の形式です。 HH：時 MM：分 SS：秒 00 は固定です。
出力通番	52	48	4	2 進数字	出力通番です。出力通番がない場合は、 (FFFFFFF) ₁₆ が設定されます。
出力マップ名	56	52	8	英数字	MCF が使用します。
メッセージ種別	64	60	1	英数字	メッセージ種別が設定されます。 n：一般一方送信メッセージ p：優先一方送信メッセージ
送信エラーコード	65	61	1	英数字	'E' が設定されます。
障害コード	66	62	1	2 進数字	障害コードが設定されます。
詳細情報	67	63	1	2 進数字	詳細情報が設定されます。 (01) ₁₆ ：エラーによる送信障害 (02) ₁₆ ：mcftdeten コマンドによる送信障害 (03) ₁₆ ：mcftdetle コマンドによる送信障害
エラーコード	68	64	1	2 進数字	MCF が使用します。
サブエラーコード	69	65	1	2 進数字	MCF が使用します。
予備	70	66	5	-	-
プロトコル個別情報	75	71	16	英数字	MCF が使用します。
予備	91	87	19	-	-

(凡例)

- : 該当しません。または，使用されません。

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

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	イベントコード「CERREVT」が設定されます。
入力元論理端末名称	8	8	英数字	障害の発生した論理端末名称が設定されます。コネクション障害時は無効です。
予備	16	8	-	-
入力元コネクション名	24	8	英数字	コネクション名が設定されます。
メッセージ入力日付	32	8	外部 10 進数字	CERREVT を入力した日付です。
メッセージ入力時刻	40	8	外部 10 進数字	CERREVT を入力した時刻です。
障害要因コード	48	4	2 進数字	障害要因が設定されます。 (00000030) ₁₆ : コネクション障害 (00000031) ₁₆ : 論理端末障害
理由コード 1	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。
回復可否識別	60	4	2 進数字	発生した障害が，自動回復できるかが設定されます。 (00000000) ₁₆ : 回復可能 (00000010) ₁₆ : 回復不可能
自動再確立有無識別	64	4	2 進数字	障害発生後，自動再確立するかが設定されます。 (00000000) ₁₆ : 自動再確立をする (00000010) ₁₆ : 自動再確立をしない
予備	68	36	-	-

(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 は，「付録 C 理由コード一覧」を参照してください。

5. ユーザOWNコーディング，MCF イベントインタフェース

表 5-9 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	無効です。
予備	16	8	-	-
入力元コネクション名	24	8	英数字	コネクション名が設定されます。
メッセージ入力日付	32	8	外部 10 進数字	COPNEVT, CCLSEVT を入力した日付 です。
メッセージ入力時刻	40	8	外部 10 進数字	COPNEVT, CCLSEVT を入力した時刻 です。
予備	48	56	-	-

(凡例)

- : 該当しません。または, 使用されません。

6

システム定義

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

TP1/NET/HSC の定義の概要

TP1/NET/HSC 固有のシステム定義の種類 - HSC1 手順

mcftalccn (コネクション定義の開始) - HSC1 手順

mcftalcle (論理端末定義) - HSC1 手順

mcftalced (コネクション定義の終了) - HSC1 手順

TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (非同期モード)

mcftalccn (コネクション定義の開始) - HSC2 手順 (非同期モード)

mcftalcle (論理端末定義) - HSC2 手順 (非同期モード)

mcftalced (コネクション定義の終了) - HSC2 手順 (非同期モード)

mcftlnf (回線リスト出力ファイル定義) - HSC2 手順 (非同期モード)

mcftllst (回線リスト定義) - HSC2 手順 (非同期モード)

mcftleof (回線リスト出力ファイル定義の終了) - HSC2 手順 (非同期モード)

TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (同期モード)

mcftalccn (コネクション定義の開始) - HSC2 手順 (同期モード)

mcftalcle (論理端末定義) - HSC2 手順 (同期モード)

mcftalced (コネクション定義の終了) - HSC2 手順 (同期モード)

6. システム定義

mcftstf (出力ファイル定義) - HSC2 手順 (同期モード)

mcftstlst (発信用相手ターミナル ID リスト定義) - HSC2 手順 (同期モード)

mcftlst (着信用相手ターミナル ID リスト定義) - HSC2 手順 (同期モード)

mcfteof (出力ファイル定義の終了) - HSC2 手順 (同期モード)

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

回線リスト定義 - HSC2 手順 (非同期モード)

相手ターミナル ID リスト定義 - HSC2 手順 (同期モード)

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/HSC の定義の概要

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

OpenTP1 のネットワークコミュニケーション定義の中での定義

OpenTP1 のネットワークコミュニケーション定義のうち、TP1/NET/HSC 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/HSC を起動するには、定義ファイルに環境情報を設定する必要があります。MCF で使用する定義ファイルを次の表に示します。

表 6-1 MCF で使用する定義ファイル

定義の種類	定義のソースファイル	定義の内容
MCF マネジャ定義	MCF マネジャ定義ソースファイル	MCF 全体の実行環境
MCF 通信構成定義	共通定義ソースファイル	プロトコルごとの実行環境
	プロトコル固有定義ソースファイル	
	発信用相手ターミナル ID リストファイル	通信相手のターミナル ID (HSC2 手順 (同期モード) の場合)
	着信用相手ターミナル ID リストファイル	
	回線リストファイル	通信に使用する回線 (HSC2 手順 (非同期モード) の場合)
MCF アプリケーション定義	MCF アプリケーション定義ソースファイル	アプリケーションの属性

定義のソースファイルは、定義コマンド、オプション、およびオペランドを指定して作成します。それらの中には、プロトコルで共通のものと、プロトコルに固有のものがあります。表 6-1 の定義の中で、TP1/NET/HSC 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/HSC 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、HSC1 手順、HSC2 手順 (非同期モード) および HSC2 手順 (同期モード) の場合に分けて説明します。プロトコル共通の定義については、マニュアル「OpenTP1 システム定義」を参照してください。ただし、mcftbuf (バッファグループ定義) の length, count オペランドの指定値については、次に示す個所に記載してあります。

【HSC1 手順の場合】

「mftalcen (コネクション定義の開始) - HSC1 手順」の注意事項

【HSC2 手順 (非同期モード) の場合】

「mftalcen (コネクション定義の開始) - HSC2 手順 (非同期モード)」の注意事項

【HSC2 手順 (同期モード) の場合】

「mftalcen (コネクション定義の開始) - HSC2 手順 (同期モード)」の注意事項

TP1/NET/HSC 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/HSC を OpenTP1 システムに組み込むときに必要なファイルです。

- システムサービス情報定義ファイル
- システムサービス共通情報定義ファイル
- MCF 定義オブジェクトファイル

この章では、システムサービス情報定義ファイルとシステムサービス共通情報定義ファイルの記述内容、および MCF 定義オブジェクトファイルを生成するユーティリティの起動コマンドについて説明します。TP1/NET/HSC を組み込む方法については、「8. 組み込み方法」を参照してください。

通信定義の内容の関連づけ

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

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

TP1/NET/HSC 固有のシステム定義の種類 - HSC1 手順

MCF 通信構成定義のうち、HSC1 手順を使用する場合の TP1/NET/HSC に固有の定義の一覧を次の表に示します。

表 6-2 TP1/NET/HSC 固有の定義の一覧 (HSC1 手順)

定義名	コマンド	オプション・オペランド	定義内容	指定値 ((値範囲)) 《省略時解釈値》	
MCF 通信構成定義	共通定義		プロトコル共通のコマンドだけで指定できます。共通のコマンドについては、マニュアル「OpenTP1 システム定義」を参照してください。		
	プロトコル固有定義	mcfaltcn (コネクション定義の開始) 指定数: 1 ~ 512	-c -	コネクション ID	1 ~ 8 文字の識別子
		-p -	プロトコルの種別	hsc1	
		-g sndbuf	メッセージ送信用バッファグループ番号	符号なし整数 ((1 ~ 512))	
			rcvbuf	メッセージ受信用バッファグループ番号	符号なし整数 ((1 ~ 512))
		-e msgbuf	メッセージ編集用バッファグループ番号	符号なし整数 ((1 ~ 512))	
			count	メッセージ編集用バッファ数	符号なし整数 ((1 ~ 131070))
		-i -	システム開始時、および再開始時に、自動的に AP 間通信開始の準備をす るかどうか	auto 《manual》	
		-m mode	使用する通信管理	xnfas	
		-k -	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、非優先の指定	《winner》 loser	

6. システム定義

TP1/NET/HSC 固有のシステム定義の種類 - HSC1 手順

定義名	コマンド	オプション・オペランド		定義内容	指定値 ((値範囲)) 《省略時解釈値》
		-l	-	論理端末の閉塞解除をするかどうか	auto 《manual》
		-u	-	逆中断受信の場合の処理	《accept》 ignore
		-x	-	コネクション回復可能障害時、自動回復動作を抑制するかどうか	《auto》 manual
		-B	ttd	TTD を送信するかどうか	《use》 nouse
			wack	WACK を送信するかどうか	《use》 nouse
	mcftalcle (論理端末定義) 指定数 : mcftalcen と 同数	-l	-	論理端末名称	1 ~ 8 文字の識別子
		-t	-	この論理端末のタイプ	any
		-m	mmsgcnt	メモリキュー内の出力メッセージの最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
			dmsgcnt	ディスクキュー内の出力メッセージの最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
		-k	quekind	出力キューの媒体の種類	《memory》 disk
			quegrpID	キューグループ ID	1 ~ 8 文字の識別子
		-o	aj	送信完了時の情報を取得するかどうか	《yes》 no
		-v	-	アプリケーション名	1 ~ 8 文字の識別子
		-w	rowtrans	送信メッセージを列信にするかどうか	use 《nouse》
mcftalced (コネクション定義の終了) 指定数 : mcftalcen と 同数	-	-	コネクション定義の終了	-	

(凡例)

- : 該当する内容がないことを表します。

注

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

定義の指定順序

HSC1 手順を使用する場合の TP1/NET/HSC のプロトコル固有定義コマンドの指定順序を、次の図に示します。

図 6-1 TP1/NET/HDLC のプロトコル固有定義コマンドの指定順序 (HSC1 手順)

$\left\{ \left\{ \begin{array}{l} \text{mcftalccn} \\ \text{mcftalcle} \\ \text{mcftalced} \end{array} \right\} \right\}$	繰り返し可	(コネクション定義の開始)
		(論理端末定義)
		(コネクション定義の終了)

注 mcftalccnとmcftalcleの指定は、1対1になるようにしてください。

mcftalccn (コネクション定義の開始) - HSC1 手順

形式

```
mcftalccn -c コネクションID
          -p hsc1
          -g "sndbuf=メッセージ送信用バッファグループ番号
             rcvbuf=メッセージ受信用バッファグループ番号"
          [-e " [msgbuf=メッセージ編集用バッファグループ番号]
             [count=メッセージ編集用バッファ数] " ]
          [-i auto | manual ]
          -m " mode=xnfas"
          -k 回線名称
          [-y winner | loser ]
          [-l auto | manual ]
          [-u accept | ignore ]
          [-x auto | manual ]
          [-B " [ttd=use | nouse ]
             [wack=use | nouse] " ]
```

機能

コネクションに関する環境を定義します。

オプション

-c コネクション ID ~ 1 ~ 8 文字の識別子

コネクション ID を指定します。

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

-p hsc1

プロトコルの種別を指定します。

hsc1

HSC1 手順

-g

(オペランド)

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

メッセージ送信用バッファのグループ番号を指定します。

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

rcvbuf= メッセージ受信用バッファグループ番号 ~ 符号なし整数 ((1 ~ 512))
メッセージ受信用バッファのグループ番号を指定します。
mcftbuf コマンドの -g オプションの groupno オペランドに指定されているバッファ
グループ番号を指定してください。

-e

(オペランド)

msgbuf= メッセージ編集用バッファグループ番号 ~ 符号なし整数 ((1 ~ 512))
入力および出力メッセージ編集 UOC の場合、メッセージ編集用として使用する
バッファグループ番号を指定します。
このオペランドを省略した場合は、メッセージ編集用バッファは確保されません。
メッセージ編集用バッファグループ番号は、mcftbuf コマンドの -g オプションの
groupno オペランドで指定するバッファグループ番号を指定してください。

count= メッセージ編集用バッファ数 ~ 符号なし整数 ((1 ~ 131070))
入力および出力メッセージ編集 UOC の場合、メッセージ編集用として使用する
バッファの数を指定します。
msgbuf オペランドで指定するメッセージ編集用バッファグループ番号に対応する
mcftbuf コマンドの -g オプションの count、および extend オペランドで指定する
バッファ数の中から、メッセージ編集用に使用するバッファ数を指定してください。
また、このオペランドの指定は mcftbuf コマンドの -g オプションの count、および
extend オペランドで指定されたバッファ数の合計値を超える指定はできません。
-e オプションの msgbuf オペランドを省略した場合は、このオペランドの指定は無
効です。

-i auto | manual ~ 《manual》

OpenTP1 システム開始時および再開時に、自動的に AP 間通信開始の準備をするかど
うかを指定します。

auto

OpenTP1 システム開始時および再開時に、自動的に AP 間通信開始の準備をし
ます。

manual

MCF 起動後、運用コマンド (mcftactcn) を入力して、AP 間通信開始の準備をし
ます。

-m

(オペランド)

mode=xnfas

使用する通信管理を指定します。

6. システム定義

mcftalccn (コネクション定義の開始) - HSC1 手順

xfas

XNF/AS を使用します。

このオペランドは必ず指定してください。

-k 回線名称 ~ 1 ~ 8 文字の識別子

HSC1 手順の回線名称を指定します。

通信管理の構成定義 (baseline 文の name オペランド) で指定した回線名称と同じ回線名称を指定してください。

XNF/AS の構成定義については、マニュアル「通信管理 XNF/AS 構成定義編」を参照してください。

-y winner | loser ~ 《winner》

コンテンション発生時の優先、非優先を指定します。

winner

優先局

loser

非優先局

-l auto | manual ~ 《manual》

AP 間通信の開始準備と同時に、論理端末の閉塞解除をするかどうかを指定します。

auto

論理端末を自動的に閉塞解除します。

manual

運用コマンド (mcftactle) を入力して、論理端末を閉塞解除します。

-u accept | ignore ~ 《accept》

相手システムから逆中断 (RVI) を受信した場合の対処方法を指定します。

accept

逆中断を受理してテキストの送信を中断します。

ignore

逆中断を無視してテキストの送信を継続します。

-x auto | manual ~ 《auto》

コネクションの回復が可能な障害が発生したときに、自動回復を抑止するかどうかを指定します。

auto

コネクション自動回復を行い、コネクションを自動的に再確立します。

manual

コネクションの自動回復を抑止します。

この場合、コネクションを再確立するには、運用コマンド (mcftactcn) を入力します。

-B

(オペランド)

ttd=use | nouse ~ 《use》

メッセージ送信時、TP1/NET/HSC の内部処理で一時的に送信ができない場合、相手システムに対して TTD (一時送信延期) を送信するかどうかを指定します。

use

TTD を送信します。

nouse

TTD を送信しません。

wack=use | nouse ~ 《use》

メッセージ受信時、TP1/NET/HSC の内部処理で一時的に受信ができない場合、相手システムに対して WACK (一時受信抑止) を送信するかどうかを指定します。

use

WACK を送信します。

nouse

WACK を送信しません。

注意事項

HSC1 手順を使用する場合、-g オプションおよび -e オプションで指定するバッファグループ番号に対応するバッファグループ定義の mcftbuf コマンドでは、1 コネクション単位に次の表に示す資源が必要です。バッファグループ定義については、マニュアル「OpenTP1 システム定義」を参照してください。

バッファ種別	length オペランド	count オペランド
sndbuf	送信最大セグメント長 ¹	最大セグメント分割数以上
revbuf	受信最大セグメント長 ¹	(最大セグメント分割数 + 1) 以上
msgbuf ²	入力および出力メッセージ編集 UOC で編集したあとの最大セグメント長	(最大セグメント分割数 × 2) 以上

注 1

送信最大セグメント長、および受信最大セグメント長は、8162 バイトとします。た

6. システム定義

mcftalccn (コネクション定義の開始) - HSC1 手順

だし、透過モードの場合は、8160 バイトにしてください。この場合のセグメント長は、テキスト形態の 4 バイトを含みます。

注 2

入力および出力メッセージ編集 UOC を使用しない場合は不要です。

なお、この値より小さい値を指定した場合、オンライン実行中にバッファ不足が発生する可能性があるため、必ず上記の値またはそれ以上の値を指定してください。また、複数のコネクションでバッファグループを共有する場合、バッファ数はコネクションごとの和を、バッファ長はコネクション間で最大値を指定してください。

mcftalcle (論理端末定義) - HSC1 手順

形式

```
mcftalcle-l 論理端末名称
-t any
[-m " {mmsgcnt=メモリ出力メッセージ最大格納数}
      {dmsgcnt=ディスク出力メッセージ最大格納数} " ]
[-k " {quekind=memory | disk}
      {quegrpid=キューグループID} " ]
[-o " {aj=yes | no} " ]
[-v アプリケーション名]
[-w " {rowtrans=use | nouse} " ]
```

機能

論理端末に関する環境を定義します。

オプション

-l 論理端末名称 ~ 1 ~ 8 文字の識別子

論理端末名称を指定します。

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

-t any

この論理端末の端末タイプを指定します。

any

任意型論理端末

-m

(オペランド)

mmsgcnt= メモリ出力メッセージ最大格納数 ~ 符号なし整数 ((0 ~ 65535)) 《0》
メモリキューで待ち合わせをする出力メッセージの最大格納数を指定します。
出力メッセージの待ち合わせ数が指定した最大数になると、それ以後 UAP からの送信要求 (dc_mcf_send 関数または SEND 文) はエラーリターンとなります。
0 を指定した場合、メモリキューで待ち合わせをする出力メッセージの数は無制限となります。ただし、実際に待ち合わせをできる出力メッセージ数は動的共用メモリの容量に依存します。

dmsgcnt= ディスク出力メッセージ最大格納数 ~ 符号なし整数 ((0 ~ 65535)) 《0》
ディスクキューで待ち合わせをする出力メッセージの最大格納数を指定します。
出力メッセージの待ち合わせ数が指定した最大数になると、それ以後 UAP からの送

6. システム定義

mcftalcle (論理端末定義) - HSC1 手順

信要求 (dc_mcf_send 関数または SEND 文) はエラーリターンとなります。

0 を指定した場合、ディスクキューで待ち合わせをする出力メッセージの数は無制限となります。ただし、実際に待ち合わせをできる出力メッセージ数はメッセージキューファイルの容量に依存します。

-k

(オペランド)

quekind=memory | disk ~ 《memory》

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

memory

メモリキューだけに割り当てます。

disk

ディスクキューおよびメモリキューに割り当てます。

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

quegrpид= キューグループ ID ~ 1 ~ 8 文字の識別子

ディスクキューで待ち合わせをする出力メッセージに使用するキューグループ ID を指定します。MCF マネージャ定義の、mcfmqgid コマンドで指定するキューグループ ID (キューグループ種別は otq) のどれかを指定してください。

このオペランドは、quekind オペランドで disk を指定した場合だけ指定します。

-o

(オペランド)

aj=yes | no ~ 《yes》

送信完了時の情報を取得するかどうかを指定します。

yes

メッセージ送信完了ジャーナルを取得します。

no

メッセージ送信完了ジャーナルを取得しません。

-v アプリケーション名

該当する論理端末で受信したメッセージを処理するアプリケーション名を指定します。

MCF アプリケーション定義 (mcfaalcap -n) の name オペランドで指定した名称を指定してください。MCF 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。

-w

(オペランド)

rowtrans=use | nouse ~ 《nouse》

送信メッセージを列信として扱うかどうかを指定します。

use

送信メッセージを列信として扱います。

nouse

送信メッセージを通常の送信として扱います。

mcftalced (コネクション定義の終了) - HSC1 手順

形式

mcftalced

機能

一つのコネクション定義の終了を示します。

オプション

ありません。

TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (非同期モード)

MCF 通信構成定義のうち、HSC2 手順 (非同期モード) を使用する場合は TP1/NET/HSC に固有の定義の一覧を次の表に示します。

表 6-3 TP1/NET/HSC 固有の定義の一覧 (HSC2 手順 (非同期モード))

定義名	コマンド	オプション・オペランド	定義内容	指定値 ((値範囲)) 《省略時解釈値》		
MCF 通信構成定義	共通定義		プロトコル共通のコマンドだけで指定できます。共通のコマンドについては、マニュアル「OpenTP1 システム定義」を参照してください。			
	プロトコル固有定義	mftalecn (コネクション定義の開始) 指定数: 1 ~ 512	-c	-	コネクション ID	1 ~ 8 文字の識別子
			-p	-	プロトコルの種別	hsc2
			-g	sndbuf	メッセージ送信用バッファグループ番号	符号なし整数 ((1 ~ 512))
				rcvbuf	メッセージ受信用バッファグループ番号	符号なし整数 ((1 ~ 512))
			-e	msgbuf	メッセージ編集用バッファグループ番号	符号なし整数 ((1 ~ 512))
				count	メッセージ編集用バッファ数	符号なし整数 ((1 ~ 131070))
			-i	-	システム開始時、および再開時にコネクションを自動的に確立するかどうか	auto 《manual》
			-k	-	回線定義リストファイル名称	1 ~ 8 文字の識別子
		-y	-	コンテンツ発生時の優先、非優先の指定	《winner》 loser	

6. システム定義

TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (非同期モード)

定義名	コマンド	オプション・オペランド	定義内容	指定値 ((値範囲)) 《省略時解釈値》	
		-l	-	論理端末の閉塞解除をするかどうか	auto 《manual》
		-u	-	逆中断受信の場合の処理	《accept》 ignore
		-n	tid	自システムのターミナル ID	1~15 文字の英数字
		-w	calltone	呼び出しトーン時間	符号なし整数 ((0 ~ 204)) 《0》(単位 : 秒)
			anstone	アンサトーン時間	符号なし整数 ((0 ~ 204)) 《0》(単位 : 秒)
		-f	eotctrl	EOT 交換をするかどうか	《use》 nouse
			holdtime	回線接続の維持時間	符号なし整数 ((0 ~ 4096)) 《1》(単位 : 秒)
			eotexl	自システム発信時の EOT 交換回数	符号なし整数 ((1 ~ 255)) 《1》
			eotexr	相手システム発信時の EOT 交換回数	符号なし整数 ((1 ~ 255)) 《255》
			eottime	EOT 交換間隔時間	符号なし整数 ((0 ~ 10)) 《1》(単位 : 秒)
		-x	-	発信による回線接続時に回線接続を自動的に再試行するかどうか	《auto》 manual
		-d	retrycnt	発信リトライ回数	符号なし整数 ((1 ~ 255)) 《6》
			retryint	発信リトライ間隔時間	符号なし整数 ((1 ~ 180)) 《60》(単位 : 秒)
		-r	reverse	送受信反転制御を使用するかどうか	use 《nouse》
		-j	idchange	ID を交換するかどうか	《use》 nouse
		-v	discinf	回線維持終了時の CERREVT を通知するかどうか	use 《nouse》
		-z	linecont	複数回線間のコンテンション発生時の優先, 非優先の指定	《winner》 loser
		-B	ttd	TTD の送信	《use》 nouse
			wack	WACK の送信	《use》 nouse

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》
	mcftalcle (論理端末定義) 指定数: 1 ~ 2048	-l	-	論理端末名称	1 ~ 8 文字の識別子
		-t	-	この論理端末のタイプ	any
		-m	mmsgcnt	メモリキュー内の出力メッセージの最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
			dmsgcnt	ディスクキュー内の出力メッセージの最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
		-k	quekind	出力キューの媒体の種類	《memory》 disk
			quegrpid	キューグループ ID	1 ~ 8 文字の識別子
		-o	aj	送信完了時の情報を取得するかどうか	《yes》 no
		-v	-	アプリケーション名	1 ~ 8 文字の識別子
		-b	attr	論理端末の属性	async
		-w	rowtrans	送信メッセージを列信にするかどうか	use 《nouse》
		-n	tid	相手システムのターミナル ID	1 ~ 15 文字の英数字
		-d	dialno	相手システムのダイヤル番号	1 ~ 100 けたの符号なし整数 (アスタリスク (*), プラス (+), ピリオド (.) を含む)
			mcftalced (コネクション定義の終了) 指定数: mcftalecn と同数	-	-
回線リスト定義	mcftlnf (回線リスト出力ファイル定義) 指定数: 1 ~ 512	-o	-	回線リスト出力オブジェクトファイル名称	1 ~ 8 文字の識別子

6. システム定義

TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (非同期モード)

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》
	mcftllst (回線リスト定義) 指定数 : 1 ~ 400	-k	-	回線名称	1 ~ 8 文字の識別子
	mcftleof (回線リスト出力ファイル定義の終了) 指定数 : mcftlnf と同様	-	-	回線リスト出力ファイル定義の終了	-

(凡例)

- : 該当する内容がないことを表します。

注

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

定義の指定順序

HSC2 手順 (非同期モード) を使用する場合の TP1/NET/HSC のプロトコル固有定義コマンドの指定順序を、次の図に示します。

図 6-2 TP1/NET/HDLC のプロトコル固有定義コマンドの指定順序 (HSC2 手順 (非同期モード))

- ・ HSC2 手順 (非同期モード) の TP1/NET/HSC プロトコル固有定義の指定順序

$\left\{ \left\{ \begin{array}{l} \text{mcftalccn} \\ \{\{\text{mcftalcle}\}\} \\ \text{mcftalced} \end{array} \right\} \right\}$	繰り返し可	(コネクション定義の開始)
		(論理端末定義)
		(コネクション定義の終了)

注 mcftalccn と mcftalcle の指定は、1対nです。

- ・ 回線リストの定義の指定順序

$\left\{ \left\{ \begin{array}{l} \text{mcftlnf} \\ \{\{\text{mcftllst}\}\} \\ \text{mcftleof} \end{array} \right\} \right\}$	繰り返し可	(回線リスト出力ファイル定義)
		(回線リスト定義)
		(回線リスト出力ファイル定義の終了)

注 mcftlnf と mcftleof は 1対1で、最大512個指定できます。
また、一つの mcftlnf に対して、mcftllst は、最大400個指定できます。

mcftalccn (コネクション定義の開始) - HSC2 手順 (非同期モード)

形式

```
mcftalccn  -c コネクションID
           -p hsc2
           -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 回線リストファイル名称
           [-y winner | loser ]
           [-l auto | manual ]
           [-u accept | ignore ]
           [-n "tid=自システムのターミナルID" ]
           [-w " {calltone=呼び出しトーン時間}
              {anstone=アンサトーン時間} " ]
           [-f " {eotctrl=use | nouse}
              {holdtime=回線接続維持時間}
              {eotexl=自システム発信時のEOT交換回数}
              {eotexr=相手システム発信時のEOT交換回数}
              {eottime=EOT交換間隔時間} " ]
           [-x auto | manual ]
           [-d " {retrycnt=発信再試行回数}
              {retryint=発信再試行間隔時間} " ]
           [-r "reverse=use | nouse" ]
           [-j "idchange=use | nouse" ]
           [-v "discinf=use | nouse" ]
           [-z "linecont=winner | loser" ]
           [-B " {ttd=use | nouse}
              {wack=use | nouse} " ]
```

機能

コネクションに関する環境を定義します。

オプション

-c コネクション ID ~ 1 ~ 8 文字の識別子

コネクション ID を指定します。

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

-p hsc2

プロトコルの種別を指定します。

6. システム定義

mcftalccn (コネクション定義の開始) - HSC2 手順 (非同期モード)

hsc2

HSC2 手順

-g

(オペランド)

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

メッセージ送信用バッファのグループ番号を指定します。

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

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

メッセージ受信用バッファのグループ番号を指定します。

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

-e

(オペランド)

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

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

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

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

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

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

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

-i auto | manual ~ 《manual》

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

auto

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

manual

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

-k 回線リストファイル名称 ~ 1 ~ 8 文字の識別子

HSC2 手順 (非同期モード) で使用する回線に対応する回線リストファイル名称を指定します。

通信管理の構成定義 (basicline の name オペランド) で指定した回線名称と同じ回線名称を指定してください。

XNF/AS の構成定義については、マニュアル「通信管理 XNF/AS 構成定義編」を参照してください。

-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

-l auto | manual ~ 《manual》

コネクションの確立と同時に、論理端末の閉塞解除をするかどうかを指定します。

auto

論理端末を自動的に閉塞解除します。

manual

運用コマンド (mcfactle) を入力して、論理端末を閉塞解除します。

-u accept | ignore ~ 《accept》

相手システムから逆中断 (RVI) を受信した場合の対処方法を指定します。

accept

逆中断を受理してテキストの送信を中断します。

ignore

逆中断を無視してテキストの送信を継続します。

-n

(オペランド)

tid= 自システムのターミナル ID ~ 1 ~ 15 文字の英数字

自システムのターミナル ID を指定します。この ID は、回線接続時に相手システムに通知されます。

6. システム定義

mcftalccn (コネクション定義の開始) - HSC2 手順 (非同期モード)

このオプションは、mcftalccn コマンド -j オプションの idchange オペランドで nouse を指定して、ID 交換をしない場合には無効です。

-w

(オペランド)

calltone= 呼び出しトーン時間 ~ 符号なし整数 ((0 ~ 204)) 《0》(単位: 秒)

電話網接続での発信時に送出する呼び出しトーン時間を指定します。

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、呼び出しトーンは送出しません。

このオペランドは、回線交換網接続時および着信時は不要です。また、ハードウェアの仕様によって、指定した値が 0.8 の倍数に切り上げられる場合があります。

anstone= アンサトーン時間 ~ 符号なし整数 ((0 ~ 204)) 《0》(単位: 秒)

電話網接続での着信時に送出するアンサトーン時間を指定します。

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、アンサトーンは送出しません。

このオペランドは、回線交換網接続時および着信時は不要です。また、ハードウェアの仕様によって、指定した値が 0.8 の倍数に切り上げられる場合があります。

-f

(オペランド)

eotctrl=use | nouse ~ 《use》

EOT 交換をするかどうかを指定します。

use

EOT 交換をします。

nouse

EOT 交換をしません。

holdtime= 回線接続維持時間 ~ 符号なし整数 ((0 ~ 4096)) 《1》(単位: 秒)

送受信のメッセージがなくなってから、回線を切断するまでの回線接続維持時間を秒単位で指定します。通信管理の構成定義で指定した時間より、小さい値を指定してください。0 を指定した場合、時間監視はしません。eotctrl オペランドで nouse を指定したときだけ指定してください。

eotexl= 自システム発信時の EOT 交換回数 ~ 符号なし整数 ((1 ~ 255)) 《1》

自システムから発信し、回線を接続した場合の EOT 交換をする回数を指定します。このオペランドで 255 を指定した場合、無限回 EOT 交換をします。

eotctrl オペランドで nouse を指定した場合、このオペランドは無効です。

eotexr= 相手システム発信時の EOT 交換回数 ~ 符号なし整数 ((1 ~ 255)) 《255》

相手システムから発信し、回線を接続した場合の EOT 交換を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 255 を指定した場合、無限回 EOT 交換をします。

eotctrl オペランドで nouse を指定した場合、このオペランドは無効です。

eottime=EOT 交換間隔時間 ~ 符号なし整数 ((0 ~ 10)) 《1》(単位: 秒)

1 回の EOT 交換が終了してから、次の EOT 交換が始まるまでの間の時間を指定します。

このオペランドで 0 を指定した場合、間隔時間を空けずに、即時に EOT 交換をします。

eotctrl オペランドで nouse を指定した場合、このオペランドは無効です。

-x auto | manual ~ 《auto》

発信による回線接続時、相手システムが使用中などの要因で接続できなかった場合、自動的に回線接続の再試行をするかどうかを指定します。

auto

自動的に回線を接続します。

manual

自動的に回線を接続しません。

-d

(オペランド)

retrycnt= 発信再試行回数 ~ 符号なし整数 ((1 ~ 255)) 《6》(単位: 秒)

発信による回線接続を再試行する回数を指定します。

255 を指定した場合、無限回再試行します。

-x オプションで manual を指定した場合、このオペランドは無効です。

retryint= 発信再試行間隔時間 ~ 符号なし整数 ((1 ~ 180)) 《60》(単位: 秒)

回線接続の再試行間隔時間を指定します。

-x オプションで manual を指定した場合、このオペランドは無効です。

発信再試行間隔時間が経過しても、回線に空きがない場合は、発信の再試行処理が遅れることがあります。

-r

(オペランド)

reverse=use | nouse ~ 《nouse》

送受信反転制御を使用するかどうかを指定します。

TP1/NET/HSC の HSC2 手順 (非同期モード) では、内部で送受信のコンテンションが発生した場合、受信が優先されます。受信完了後直ちに送受信処理をすることを送受信反転制御といいます。送受信反転制御を使用する場合には、use を指定し

6. システム定義

mcftalccn (コネクション定義の開始) - HSC2 手順 (非同期モード)

てください。

use

送受信反転制御を使用します。

nouse

送受信反転制御を使用しません。

-j

(オペランド)

idchange=use | nouse ~ 《use》

回線接続時、ID 交換をするかどうかを指定します。

use

ID 交換をします。

nouse

ID 交換をしません。

-v

(オペランド)

discinf=use | nouse ~ 《nouse》

回線接続終了時に CERREVT を通知するかどうかを指定します。

use

回線接続終了時に CERREVT を通知します。

nouse

回線接続終了時に CERREVT を通知しません。

-z

(オペランド)

linecont=winner | loser ~ 《winner》

複数の回線間で発信または着信のコンテンションが発生した場合の優先とする側を指定します。

このオペランドは、非同期モードで一つのコネクションに複数の回線を指定した場合に有効です。

winner

自システムからの発信を優先して回線を接続します。

loser

相手システムからの発信を優先して回線を接続します。

-B

(オペランド)

ttd=use | nouse ~ 《use》

メッセージ送信時、TP1/NET/HSC の内部処理で一時的に送信ができない場合、相手システムに対して TTD (一時送信延期) を送信するかどうかを指定します。

use

TTD を送信します。

nouse

TTD を送信しません。

wack=use | nouse ~ 《use》

メッセージ受信時、TP1/NET/HSC の内部処理で一時的に受信ができない場合、相手システムに対して WACK (一時受信抑止) を送信するかどうかを指定します。

use

WACK を送信します。

nouse

WACK を送信しません。

注意事項

HSC2 手順 (非同期モード) を使用する場合、g オプションおよび -e オプションで指定するバッファグループ番号に対応するバッファグループ定義の mcfbuf コマンドでは、1 コネクション単位に次の表に示す資源が必要です。バッファグループ定義については、マニュアル「OpenTP1 システム定義」を参照してください。

バッファ種別	length オペランド	count オペランド
sndbuf	送信最大セグメント長 ¹	(回線数 × 最大セグメント分割数) 以上
rcvbuf	受信最大セグメント長 ¹	(回線数 × 最大セグメント分割数 + 1) 以上
msgbuf ²	入力および出力メッセージ編集 UOC で編集したあとの最大セグメント長	(最大セグメント分割数 × 2) 以上

注 1

送信最大セグメント長、および受信最大セグメント長は、8162 バイトとします。ただし、透過モードの場合は、8160 バイトにしてください。この場合のセグメント長は、テキスト形態の 4 バイトを含みます。

注 2

6. システム定義

mcftalccn (コネクション定義の開始) - HSC2 手順 (非同期モード)

入力および出力メッセージ編集 UOC を使用しない場合は不要です。

なお、この値より小さい値を指定した場合、オンライン実行中にバッファ不足が発生する可能性があるため、必ず上記の値またはそれ以上の値を指定してください。また、複数のコネクションでバッファグループを共有する場合、バッファ数はコネクションごとの和を、バッファ長はコネクション間で最大値を指定してください。

mcftalcle (論理端末定義) - HSC2 手順 (非同期モード)

形式

```
mcftalcle -l 論理端末名称
          -t any
          [-m " {mmsgcnt=メモリ出力メッセージ最大格納数}
            {dmsgcnt=ディスク出力メッセージ最大格納数} " ]
          [-k " {quekind=memory | disk}
            {quegrpид=キューグループID} " ]
          [-o " {aj=yes | no} " ]
          [-v アプリケーション名]
          [-b "attr=async" ]
          [-w " {rowtrans=use | nouse} " ]
          [-n "tid=相手システムのターミナルID" ]
          [-d "dialno=相手システムのダイヤル番号" ]
```

機能

論理端末に関する環境を定義します。

オプション

-l 論理端末名称 ~ 1 ~ 8 文字の識別子

論理端末名称を指定します。

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

-t any

この論理端末の端末タイプを指定します。

any

任意型論理端末

-m

(オペランド)

mmsgcnt= メモリ出力メッセージ最大格納数 ~ 符号なし整数 ((0 ~ 65535)) 《0》
メモリキューで待ち合わせをする出力メッセージの最大格納数を指定します。
出力メッセージの待ち合わせ数が指定した最大数になると、それ以後 UAP からの送信要求 (dc_mcf_send 関数または SEND 文) はエラーリターンとなります。
0 を指定した場合、メモリキューで待ち合わせをする出力メッセージの数は無制限となります。ただし、実際に待ち合わせをできる出力メッセージ数は動的共用メモリの容量に依存します。

6. システム定義

mcftalcle (論理端末定義) - HSC2 手順 (非同期モード)

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

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

-k

(オペランド)

quekind=memory | disk ~ 《memory》

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

memory

メモリキューだけに割り当てます。

disk

ディスクキューおよびメモリキューに割り当てます。

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

quegrpид= キューグループ ID ~ 1 ~ 8 文字の識別子

ディスクキューで待ち合わせをする出力メッセージに使用するキューグループ ID を指定します。MCF マネージャ定義の、mcfmqgid コマンドで指定するキューグループ ID (キューグループ種別は otq) のどれかを指定してください。

このオペランドは、quekind オペランドで disk を指定した場合だけ指定します。

-o

(オペランド)

aj=yes | no ~ 《yes》

送信完了時の情報を取得するかどうかを指定します。

yes

メッセージ送信完了ジャーナルを取得します。

no

メッセージ送信完了ジャーナルを取得しません。

-v アプリケーション名

該当する論理端末で受信したメッセージを処理するアプリケーション名を指定します。

MCF アプリケーション定義 (mcfaalcap -n) の name オペランドで指定した名称を指定してください。MCF 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。

-b

(オペランド)

attr=async ~ 《async》

論理端末の属性を指定します。HSC2 手順 (非同期モード) の場合, この `async` を指定するか, このオプションを省略してください。

`async`

論理端末は UAP と結び付くことはなく, 非同期に送受信をします。

-w

(オペランド)

rowtrans=use | nouse ~ 《nouse》

送信メッセージを列信として扱うかどうかを指定します。

`use`

送信メッセージを列信として扱います。

`nouse`

送信メッセージを通常の送信として扱います。

-n

(オペランド)

tid= 相手システムのターミナル ID ~ 1 ~ 15 文字の英数字

相手システムのターミナル ID を指定します。

一つの接続では, 重複したターミナル ID は指定できません。

このオプションは, `mcfcalc` コマンド `-j` オプションの `idchange` オペランドで `nouse` を指定して, ID 交換をしない場合には無効です。

-d

(オペランド)

dialno= 相手システムのダイヤル番号 ~ 1 ~ 100 けたの符号なし整数 (アスタリスク (*), プラス (+), ピリオド (.) を含む)

相手システムのダイヤル番号を指定します。

このオペランドは, HSC2 手順 (非同期モード) で, 自システムからの発信による接続がある場合有効です。

ダイヤル番号の形式は次のとおりです。

【V25.bis の場合】

mmmmmmm

【V21 (通常発信) の場合】

6. システム定義

mcftalcle (論理端末定義) - HSC2 手順 (非同期モード)

mmmmmmm+

【V21 (短縮ダイヤル, ペア型閉域) の場合】

.nn+

(凡例)

mmmmmmm : 1 ~ 100 けたの符号なし整数

nn : 通信管理で回線網に登録する番号

mcftalced (コネクション定義の終了) - HSC2 手順 (非同期モード)

形式

mcftalced

機能

一つのコネクション定義の終了を示します。

オプション

ありません。

mcftlnf (回線リスト出力ファイル定義) - HSC2 手順 (非同期モード)

形式

mcftlnf -o 回線リスト出力オブジェクトファイル名称

機能

HSC2 手順 (非同期モード) の場合, 回線リストを出力するオブジェクトファイルの名称を指定します。

オプション

-o 回線リスト出力オブジェクトファイル名称 ~ 1 ~ 8 文字の識別子

回線リストを出力するオブジェクトファイルの名称を指定します。

mcftllst (回線リスト定義) - HSC2 手順 (非同期モード)

形式

```
mcftllst -k "回線名称"
```

機能

HSC2 手順 (非同期モード) の場合、回線の名称を指定します。

オプション

-k "回線名称" ~ 1 ~ 8 文字の識別子

回線名称を指定します。

通信管理の構成定義 (baseline 文の name オペランド) で指定した回線名称を指定します。

6. システム定義

mcftleaf (回線リスト出力ファイル定義の終了) - HSC2 手順 (非同期モード)

mcftleaf (回線リスト出力ファイル定義の終了) - HSC2 手順 (非同期モード)

形式

mcftleaf

機能

回線リスト出力ファイル定義の終了を示します。

オプション

ありません。

TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (同期モード)

MCF 通信構成定義のうち、HSC2 手順 (同期モード) を使用する場合は TP1/NET/HSC に固有の定義の一覧を次の表に示します。

表 6-4 TP1/NET/HSC 固有の定義の一覧 (HSC2 手順 (同期モード))

定義名	コマンド	オプション・オペランド	定義内容	指定値 ((値範囲)) 《省略時解釈値》	
MCF 通信構成定義	共通定義		プロトコル共通のコマンドだけで指定できます。共通のコマンドについては、マニュアル「OpenTP1 システム定義」を参照してください。		
	プロトコル固有定義	mcfaltcnn (コネクション定義の開始) 指定数: 1 ~ 512	-c	-	コネクション ID 1 ~ 8 文字の識別子
			-p	-	プロトコルの種別 hsc2
			-g	sndbuf	メッセージ送信用バッファグループ番号 符号なし整数 ((1 ~ 512))
				rcvbuf	メッセージ受信用バッファグループ番号 符号なし整数 ((1 ~ 512))
			-e	msgbuf	メッセージ編集用バッファグループ番号 符号なし整数 ((1 ~ 512))
				count	メッセージ編集用バッファ数 符号なし整数 ((1 ~ 131070))
			-i	-	システム開始時、および再開時にコネクションを自動的に確立するかどうか auto 《manual》
			-k	-	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、非優先の指定 《winner》 loser
			-l	-	論理端末の閉塞解除をどうか auto 《manual》

6. システム定義

TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (同期モード)

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》	
		-u	-	逆中断受信の場合の処理	《accept》 ignore
		-n	tid	自システムのターミナル ID	1~15 文字の英数字
		-w	calltone	呼び出しトーン時間	符号なし整数 ((0 ~ 204)) 《0》(単位: 秒)
			anstone	アンサトーン時間	符号なし整数 ((0 ~ 204)) 《0》(単位: 秒)
		-o	slist	発信用相手ターミナル ID リストファイル名称	1 ~ 8 文字の識別子
			rlist	着信用相手ターミナル ID リストファイル名称	1 ~ 8 文字の識別子
		-f	eotctrl	EOT 交換をするかどうか	《use》 nouse
			holdtime	回線接続の維持時間	符号なし整数 ((0 ~ 4096)) 《1》(単位: 秒)
			eotexl	自システム発信時の EOT 交換回数	符号なし整数 ((1 ~ 255)) 《1》
			eotexr	相手システム発信時の EOT 交換回数	符号なし整数 ((1 ~ 255)) 《255》(単位: 秒)
			eottime	EOT 交換間隔時間	符号なし整数 ((0 ~ 10)) 《1》(単位: 秒)
		-j	idchange	ID を交換するかどうか	《use》 nouse
		-B	ttd	TTD の送信	《use》 nouse
			wack	WACK の送信	《use》 nouse
	mcfalcle (論理端末定義) 指定数: 1 ~ 2048	-l	-	論理端末名称	1 ~ 8 文字の識別子
		-t	-	この論理端末のタイプ	any
		-m	mmsgent	メモリキュー内の出力メッセージの最大格納数	符号なし整数 ((0 ~ 65535)) 《0》
		-k	quekind	出力キューの媒体の種類	《memory》

定義名	コマンド	オプション・オペランド		定義内容	指定値 ((値範囲)) 《省略時解釈値》
		-o	aj	送信完了時の情報を取得するか	《yes》 no
		-v	-	アプリケーション名	1 ~ 8 文字の識別子
		-b	attr	論理端末の属性	sync
	mcfstalced (コネクション定義の終了) 指定数: mcfstalcen と同数	-	-	コネクション定義の終了	-
相手ターミナル ID リストの定義	mcfststf (出力ファイル定義) 指定数: 1 ~ 512 (発信用), 1 ~ 512 (着信用)	-o	-	出力オブジェクトファイル名	1 ~ 8 文字の識別子
	mcfststlst (発信用相手ターミナル ID リスト定義) 指定数: 1 ~ 4096	-t	tid	相手システムのターミナル ID	1 ~ 15 文字の英数字
		-d	dialno	相手システムのターミナルのダイヤル番号	1 ~ 100 けたの符号なし整数 (アスタリスク (*), プラス (+), ピリオド (.) を含む)
mcftrtlst (着信用相手ターミナル ID リスト定義) 指定数: 1 ~ 4096	-t	tid	相手システムのターミナル ID	1 ~ 15 文字の英数字	

6. システム定義

TP1/NET/HSC 固有のシステム定義の種類 - HSC2 手順 (同期モード)

定義名	コマンド	オプション・オペランド		定義内容	指定値 ((値範囲)) 《省略時解釈値》
		-l	leid	送受信をする論理端末名称	1 ~ 8 文字の識別子
	mcfteof (出力ファイル定義の終了) 指定数: mcftstf と同数	-	-	出力ファイル定義の終了	-

(凡例)

- : 該当する内容がないことを表します。

注

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

定義の指定順序

HSC2 手順 (同期モード) を使用する場合は TP1/NET/HSC のプロトコル固有定義コマンドの指定順序を、次の図に示します。

図 6-3 TP1/NET/HDLC のプロトコル固有定義コマンドの指定順序 (HSC2 手順 (同期モード))

・ HSC2 手順 (同期モード) の TP1/NET/HSC プロトコル固有定義の指定順序

$\left\{ \left(\begin{array}{l} \text{mcftalccn} \\ \{\{\text{mcftalcle}\}\} \\ \text{mcftalced} \end{array} \right) \right\}$	繰り返し可	(コネクション定義の開始) (論理端末定義) (コネクション定義の終了)
---	-------	--

注 mcftalccn と mcftalcle の指定は、1対nです。

・ 相手ターミナルIDリストの定義の指定順序 (ID交換をする場合)

発信相手ターミナルIDリスト

$\left\{ \left(\begin{array}{l} \text{mcftstf} \\ \{\{\text{mcftstlst}\}\} \\ \text{mcftleof} \end{array} \right) \right\}$	繰り返し可	(出力ファイル定義) (発信相手ターミナルIDリスト定義) (出力ファイル定義の終了)
--	-------	---

着信用相手ターミナルIDリスト

$\left\{ \left(\begin{array}{l} \text{mcftstf} \\ \{\{\text{mcftstlst}\}\} \\ \text{mcftleof} \end{array} \right) \right\}$	繰り返し可	(出力ファイル定義) (着信用相手ターミナルIDリスト定義) (出力ファイル定義の終了)
--	-------	--

注 mcftstf と mcftleof は 1対1で、発信用、着信用それぞれで最大512個指定できます。
また、一つの mcftstf に対して、mcftstlst または mcfttrtlst は、最大4096個指定できます。

mcftalccn (コネクション定義の開始) - HSC2 手順 (同期モード)

形式

```
mcftalccn -c コネクションID
          -p hsc2
          -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 ]
          [-l auto | manual ]
          [-u accept | ignore ]
          [-n "tid=自システムのターミナルID" ]
          [-w " [calltone=呼び出しトーン時間]
             [anstone=アンサトーン時間] " ]
          [-o "slist=発信用相手ターミナルIDリストファイル名称
             rlist=着信用相手ターミナルIDリストファイル名称" ]
          [-f " [eotctrl=use | nouse ]
             [holdtime=回線接続維持時間]
             [eotexl=自システム発信時のEOT交換回数]
             [eotexr=相手システム発信時のEOT交換回数]
             [eottime=EOT交換間隔時間] " ]
          [-j "idchange=use | nouse_" ]
          [-B " [ttd=use | nouse ]
             [wack=use | nouse ] " ]
```

機能

コネクションに関する環境を定義します。

オプション

-c コネクション ID ~ 1 ~ 8 文字の識別子

コネクション ID を指定します。

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

-p hsc2

プロトコルの種別を指定します。

hsc2

HSC2 手順

-g

(オペランド)

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

メッセージ送信用バッファのグループ番号を指定します。

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

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

メッセージ受信用バッファのグループ番号を指定します。

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

-e

(オペランド)

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

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

このオペランドを省略した場合は、メッセージ編集用バッファは確保されません。

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

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

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

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

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

-i auto | manual ~ 《manual》

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

auto

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

manual

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

6. システム定義

mcftalccn (コネクション定義の開始) - HSC2 手順 (同期モード)

-k 回線名称 ~ 1 ~ 8 文字の識別子

HSC2 手順 (同期モード) の回線名称を指定します。

通信管理の構成定義 (baseline の name オペランド) で指定した回線名称と同じ回線名称を指定してください。

XNF/AS の構成定義については、マニュアル「通信管理 XNF/AS 構成定義編」を参照してください。

-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

-l auto | manual ~ 《manual》

コネクションの確立と同時に、論理端末の閉塞解除をするかどうかを指定します。

auto

論理端末を自動的に閉塞解除します。

manual

運用コマンド (mcftactle) を入力して、論理端末を閉塞解除します。

-u accept | ignore ~ 《accept》

相手システムから逆中断 (RVI) を受信した場合の対処方法を指定します。

accept

逆中断を受理してテキストの送信を中断します。

ignore

逆中断を無視してテキストの送信を継続します。

-n

(オペランド)

tid= 自システムのターミナル ID ~ 1 ~ 15 文字の英数字

自システムのターミナル ID を指定します。この ID は、回線接続の発信時および着信時に相手システムに通知されます。

このオプションは、mcftalccn コマンド -j オプションの idchange オペランドで nouse を指定して、ID 交換をしない場合には無効です。

-w

(オペランド)

calltone= 呼び出しトーン時間 ~ 符号なし整数 ((0 ~ 204))《0》(単位: 秒)

電話網接続での発信時に送出する呼び出しトーン時間を指定します。

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、呼び出しトーンは送出しません。

このオペランドは、回線交換網接続時および着信時は不要です。また、ハードウェアの仕様によって、指定した値が 0.8 の倍数に切り上げられる場合があります。

anstone= アンサトーン時間 ~ 符号なし整数 ((0 ~ 204))《0》(単位: 秒)

電話網接続での着信時に送出するアンサトーン時間を指定します。

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、アンサトーンは送出しません。

このオペランドは、回線交換網接続時および着信時は不要です。また、ハードウェアの仕様によって、指定した値が 0.8 の倍数に切り上げられる場合があります。

-o

(オペランド)

slist= 発信用相手ターミナル ID リストファイル名称 ~ 1 ~ 8 文字の識別子

該当するコネクションから発信できる相手システムのターミナル ID の一覧をリストとして登録してあるファイル名称を指定します。

このオペランドは、ID 交換する場合に指定します。-j オプションの idchange オペランドで nouse を指定した場合、このオペランドは無効です。

このファイルは、発信用相手ターミナル ID リスト定義ユティリティ (mcfhs2snd) で作成します。発信用相手ターミナル ID リスト定義ユティリティについては、この章の「相手ターミナル ID リスト定義 - HSC2 手順 (同期モード)」を参照してください。

rlist= 着信用相手ターミナル ID リストファイル名称 ~ 1 ~ 8 文字の識別子

該当するコネクションで接続を許可する相手システムのターミナル ID の一覧をリストとして登録してあるファイル名称を指定します。

このオペランドは、ID 交換する場合に指定します。-j オプションの idchange オペランドで nouse を指定した場合、このオペランドは無効です。

このファイルは、着信用相手ターミナル ID リスト定義ユティリティ (mcfhs2rcv) で作成します。着信用相手ターミナル ID リスト定義ユティリティについては、この章の「相手ターミナル ID リスト定義 - HSC2 手順 (同期モード)」を参照してください。

-f

(オペランド)

6. システム定義

mcftalccn (コネクション定義の開始) - HSC2 手順 (同期モード)

eotctrl=use | nouse ~ 《use》

EOT 交換をするかどうかを指定します。

use

EOT 交換をします。

nouse

EOT 交換をしません。

holdtime= 回線接続維持時間 ~ 符号なし整数 ((0 ~ 4096)) 《1》(単位: 秒)

送受信のメッセージがなくなってから、回線を切断するまでの回線接続維持時間を秒単位で指定します。通信管理の構成定義で指定した時間より、小さい値を指定してください。0 を指定した場合、時間監視はしません。eotctrl オペランドで nouse を指定したときだけ指定してください。

eotexl= 自システム発信時の EOT 交換回数 ~ 符号なし整数 ((1 ~ 255)) 《1》

自システムから発信し、回線を接続した場合の EOT 交換をする回数を指定します。このオペランドで 255 を指定した場合、無限回 EOT 交換をします。eotctrl オペランドで nouse を指定した場合、このオペランドは無効です。

eotexr= 相手システム発信時の EOT 交換回数 ~ 符号なし整数 ((1 ~ 255)) 《255》

相手システムから発信し、回線を接続した場合の EOT 交換を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 255 を指定した場合、無限回 EOT 交換をします。eotctrl オペランドで nouse を指定した場合、このオペランドは無効です。

eottime=EOT 交換間隔時間 ~ 符号なし整数 ((0 ~ 10)) 《1》(単位: 秒)

1 回の EOT 交換が終了してから、次の EOT 交換が始まるまでの間の時間を指定します。このオペランドで 0 を指定した場合、間隔時間を空けないで、即時に EOT 交換をします。eotctrl オペランドで nouse を指定した場合、このオペランドは無効です。

j

(オペランド)

idchange=use | nouse ~ 《use》

回線接続時、ID 交換をするかどうかを指定します。

use

ID 交換をします。

nouse

ID 交換をしません。

-B

(オペランド)

ttd=use | nouse ~ 《use》

メッセージ送信時、TP1/NET/HSC の内部処理で一時的に送信ができない場合、相手システムに対して TTD (一時送信延期) を送信するかどうかを指定します。

use

TTD を送信します。

nouse

TTD を送信しません。

wack=use | nouse ~ 《use》

メッセージ受信時、TP1/NET/HSC の内部処理で一時的に受信ができない場合、相手システムに対して WACK (一時受信抑止) を送信するかどうかを指定します。

use

WACK を送信します。

nouse

WACK を送信しません。

注意事項

HSC2 手順 (同期モード) を使用する場合、g オプションおよび -e オプションで指定するバッファグループ番号に対応するバッファグループ定義の mcftbuf コマンドでは、1 コネクション単位に次の表に示す資源が必要です。バッファグループ定義については、マニュアル「OpenTP1 システム定義」を参照してください。

バッファ種別	length オペランド	count オペランド
sndbuf	送信最大セグメント長 ¹	最大セグメント分割数以上
rcvbuf	受信最大セグメント長 ¹	(最大セグメント分割数 + 1) 以上
msgbuf ²	入力および出力メッセージ編集 UOC で編集したあとの最大セグメント長	(最大セグメント分割数 × 2) 以上

注 1

送信最大セグメント長、および受信最大セグメント長は、8162 バイトとします。ただし、透過モードの場合は、8160 バイトにしてください。この場合のセグメント長は、テキスト形態の 4 バイトを含みます。

注 2

入力および出力メッセージ編集 UOC を使用しない場合は不要です。

6. システム定義

mcftalccn (コネクション定義の開始) - HSC2 手順 (同期モード)

なお、この値より小さい値を指定した場合、オンライン実行中にバッファ不足が発生する可能性があるため、必ず上記の値またはそれ以上の値を指定してください。また、複数のコネクションでバッファグループを共有する場合、バッファ数はコネクションごとの和を、バッファ長はコネクション間で最大値を指定してください。

mcftalcle (論理端末定義) - HSC2 手順 (同期モード)

形式

```
mcftalcle -l 論理端末名称  
          -t any  
          [-m " {mmsgcnt=メモリ出力メッセージ最大格納数} " ]  
          [-k " {quekind=memory} " ]  
          [-o " {aj=yes | no} " ]  
          [-v アプリケーション名]  
          [-b "attr=sync" ]
```

機能

論理端末に関する環境を定義します。

オプション

-l 論理端末名称 ~ 1 ~ 8 文字の識別子

論理端末名称を指定します。

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

-t any

この論理端末の端末タイプを指定します。

any

任意型論理端末

-m

(オペランド)

mmsgcnt= メモリ出力メッセージ最大格納数 ~ 符号なし整数 ((0 ~ 65535)) 《0》
メモリキューで待ち合わせをする出力メッセージの最大格納数を指定します。
出力メッセージの待ち合わせ数が指定した最大数になると、それ以後 UAP からの送信要求 (dc_mcf_send 関数または SEND 文) はエラーリターンとなります。
0 を指定した場合、メモリキューで待ち合わせをする出力メッセージの数は無制限となります。ただし、実際に待ち合わせをできる出力メッセージ数は動的共用メモリの容量に依存します。

-k

(オペランド)

6. システム定義

mcftalcle (論理端末定義) - HSC2 手順 (同期モード)

quekind=memory ~ 《memory》

出力メッセージの割り当て先 (メモリキュー) を指定します。

memory

メモリキューに割り当てます。

-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

メッセージ送信完了ジャーナルを取得します。

no

メッセージ送信完了ジャーナルを取得しません。

-v アプリケーション名

該当する論理端末で受信したメッセージを処理するアプリケーション名を指定します。

MCF アプリケーション定義 (mcfaalcap -n) の name オペランドで指定した名称を指定してください。MCF 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。

-b

(オペランド)

attr=sync

論理端末の属性を指定します。HSC2 手順 (同期モード) の場合、必ず sync を指定してください。

sync

論理端末は UAP と 1 対 1 で結び付き、同期に送受信をします。

mcftalced (コネクション定義の終了) - HSC2 手順 (同期モード)

形式

mcftalced

機能

一つのコネクション定義の終了を示します。

オプション

ありません。

mcftstf (出力ファイル定義) - HSC2 手順 (同期モード)

形式

`mcftstf -o` [パス名] 出力ファイル名

機能

発信用相手ターミナル ID リストファイル, および着信用相手ターミナル ID リストファイルの名称を定義します。

オプション

`-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

出力オブジェクトファイル名を指定します。

このファイル名は, ほかの `mcftstf` コマンドの `-o` オプションで指定するファイル名と重複して指定できません。重複して指定した場合, この `mcftstf` コマンドから `mcfteof` コマンドが指定されるまでの定義は, すべて無視されます。また, 指定した出力オブジェクトファイルは生成されません。

mcftstlst (発信用相手ターミナル ID リスト定義) - HSC2 手順 (同期モード)

形式

```
mcftstlst -t "tid=相手システムのターミナルID"  
          -d "dialno=相手システムのターミナルのダイヤル番号"
```

機能

発信用相手ターミナル ID リストの内容を定義します。

オプション

-t

(オペランド)

tid= 相手システムのターミナル ID ~ 1 ~ 15 文字の英数字
相手システムのターミナル ID を指定します。

同じ出力オブジェクトファイル内のほかの mcftstlst コマンドの -t オプションで指定する相手システムのターミナル ID と重複して指定できません。

-d

(オペランド)

dialno= 相手システムのターミナルのダイヤル番号 ~ 1 ~ 100 けたの符号なし整数 (アスタリスク (*), プラス (+), ピリオド (.) を含む)

相手システムのターミナル ID に対応した相手システムのダイヤル番号を指定します。

同じ出力オブジェクトファイル内のほかの mcftstlst コマンドの -d オプションで指定する相手システムのターミナルのダイヤル番号と重複して指定できません。

ダイヤル番号の形式は次のとおりです。

【V25.bis の場合】

mmmmmmm

【V21 (通常発信) の場合】

mmmmmmm+

【V21 (短縮ダイヤル , ペア型閉域) の場合】

.nn+

(凡例)

mmmmmmm : 1 ~ 100 けたの符号なし整数

nn : 通信管理で回線網に登録する番号

mcfttrlst (着信用相手ターミナル ID リスト定義) - HSC2 手順 (同期モード)

形式

```
mcfttrlst -t "tid=相手システムのターミナルID"  
          -l "leid=送受信をする論理端末名称"
```

機能

着信用相手ターミナル ID リストの内容を定義します。

オプション

-t

(オペランド)

tid= 相手システムのターミナル ID ~ 1 ~ 15 文字の英数字

相手システムのターミナル ID を指定します。

同じ出力オブジェクトファイル内のほかの mcfttrlst コマンドの -t オプションで指定する相手システムのターミナル ID と重複して指定できません。

-l

(オペランド)

leid= 送受信をする論理端末名称 ~ 1 ~ 8 文字の識別子

相手システムのターミナル ID に指定した相手システムのから着信した場合の送受信をする論理端末名称を指定します。

このオプションで指定する論理端末名称は、メッセージを着信するコネクションに対応して定義された論理端末の名称を指定してください。

このオプションを省略した場合は、メッセージを着信したコネクションに対応して定義された論理端末のうち、先頭に定義された論理端末を使用してメッセージの送受信をします。

mcfteof (出力ファイル定義の終了) - HSC2 手順 (同期モード)

形式

mcfteof

機能

発信用相手ターミナル ID リスト定義, および着信用相手ターミナル ID リスト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/HSC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/HSC の実行形式プログラム名" ~ 1 ~ 8 文字の識別子
```

MCF 通信サービスを起動するための実行形式プログラム名を指定します。

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

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

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

システムサービス共通情報定義の完全パス名を次に示します。

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

形式

set 形式

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

機能

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

set 形式のオペランド

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

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

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

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

注 1

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

注 2

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

6. システム定義

システムサービス共通情報定義

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

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

各 MCF 通信プロセスでアクセスするファイル数の中の最大値を指定します。

ファイル記述子の最大数を求める計算式を次に示します。

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

注 1

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

注 2

MCF 通信プロセスが扱う定義ファイルなどの数の最大値です。

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

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

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

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

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

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

注意事項

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

MCF 定義オブジェクトの生成

MCF 定義オブジェクト生成ユーティリティでは、MCF の定義ファイルの構文のチェックと定義オブジェクトファイルへの変換をします。

HSC2 手順（同期モード）で ID を交換する場合、相手ターミナル ID リスト定義ユーティリティを使用してファイルを作成する必要があります。

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

HSC1 手順の場合

```
mcfhsc1 -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
```

HSC2 手順の場合

```
mcfhsc2 -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/HSC のプロトコル固有定義ファイルの構文をチェックし、定義オブジェクトファイルを作成します。

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

TP1/NET/HSC 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。

回線リスト定義 - HSC2 手順 (非同期モード)

回線リスト定義ユーティリティでは、回線リスト定義オブジェクトファイルを作成します。このユーティリティは、HSC2 手順 (非同期モード) の場合にだけ使用できます。

形式

mcfhhs2lin [パス名] 入力ファイル名

機能

回線リスト定義で指定した内容を、このユーティリティを使用して、回線リストファイルを作成します。このユーティリティで作成されるファイルは、出力ファイル定義 (mcftlnf -o) で指定した出力オブジェクトファイル名が付きます。

出力ファイル定義で指定する出力オブジェクトファイル名は、MCF 通信構成定義 (mcftalcen -k) で指定したファイル名と同じにしてください。

1 回のユーティリティの起動で作成できる出力オブジェクトファイル名の数は、最大 512 個です。

オプション

[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。

入力ファイル名は複数指定できます。

相手ターミナルID リスト定義 - HSC2 手順 (同期モード)

相手ターミナルID リスト定義ユーティリティでは、発信用および着信用相手ターミナルID リストの定義オブジェクトファイルを作成します。このユーティリティは、HSC2 手順 (同期モード) の場合にだけ使用できます。

形式

発信用相手ターミナルID リスト定義ユーティリティの場合

`mcfhs2snd` [パス名] 入力ファイル名

着信用相手ターミナルID リスト定義ユーティリティの場合

`mcfhs2rcv` [パス名] 入力ファイル名

機能

発信用および着信用相手ターミナルID リスト定義で指定した内容を、このユーティリティを使用して、発信用および着信用相手ターミナルID リストファイルを作成します。このユーティリティで作成されるファイルは、出力ファイル定義 (`mcftstf -o`) で指定した出力オブジェクトファイル名が付きます。

出力ファイル定義で指定する出力オブジェクトファイル名は、次に示すファイル名称と同じにしてください。

【発信用の場合】

MCF 通信構成定義 (`mcftalccn -o`) の `slist` オペランドで指定した発信用相手ターミナルID リストファイル名称

【着信用の場合】

MCF 通信構成定義 (`mcftalccn -o`) の `rlist` オペランドで指定した着信用相手ターミナルID リストファイル名称

作成されたファイルは、ID 交換をすることを指定した場合、相手システムとの接続確立時に読み込まれます。

1 回のユーティリティの起動で作成できる出力オブジェクトファイルの数は、最大 512 個です。

オプション

[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。

6. システム定義

相手ターミナルIDリスト定義 - HSC2 手順（同期モード）

入力ファイル名は複数指定できます。

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

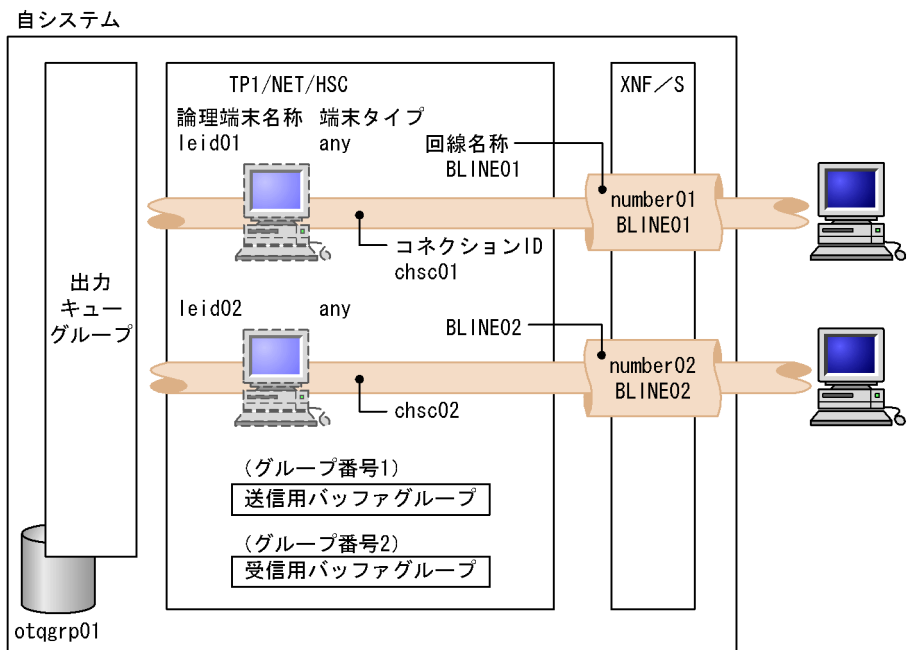
自システムの通信管理との定義の関係 (HSC1 手順)

TP1/NET/HSC の HSC1 手順を使用する場合の、自システムの通信管理 (XNF/AS) との定義の関係を説明します。

TP1/NET/HSC の定義では、XNF/AS 構成定義と同じ回線名称を指定します。指定の方法は、TP1/NET/HSC の MCF 通信構成定義 (mcfcalcen -k) で指定する回線名称と、XNF/AS のベーシック回線定義文 (baseline name) で指定する回線名称を同じにします。XNF/AS 構成定義については、マニュアル「通信管理 XNF/AS 構成定義編」を参照してください。

通信管理 XNF/AS を使用した HSC1 手順の TP1/NET/HSC のシステム構成例を次の図に示します。

図 6-4 TP1/NET/HSC のシステム構成例 (HSC1 手順)



自システムの通信管理との定義の関係 (HSC2 手順 (非同期モード))

TP1/NET/HSC の HSC2 手順 (非同期モード) を使用する場合の、自システムの通信管

6. システム定義

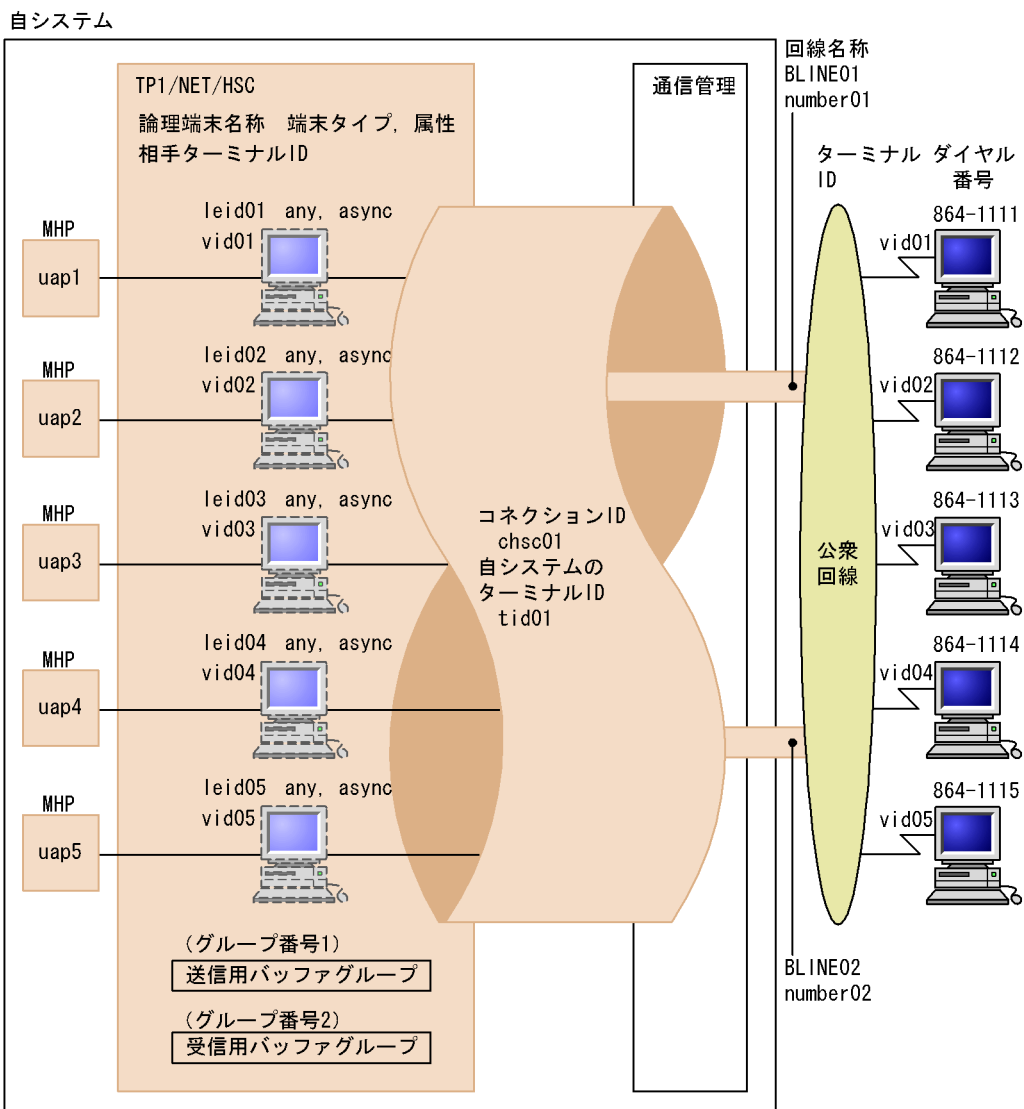
自システムの通信管理プログラムと関連づける内容

理 (XNF/AS) との定義の関係を説明します。

TP1/NET/HSC の定義では、XNF/AS 構成定義と同じ回線名称を指定します。指定の方法は、TP1/NET/HSC の MCF 通信構成定義 (mcftalecn -k) で指定する回線名称と、XNF/AS のベーシック回線定義文 (baseline name) で指定する回線名称を同じにします。XNF/AS 構成定義については、マニュアル「通信管理 XNF/AS 構成定義編」を参照してください。

通信管理 XNF/AS を使用した HSC 2 手順 (非同期モード) の TP1/NET/HSC のシステム構成例を次の図に示します。

図 6-5 TP1/NET/HSC のシステム構成例 (HSC 2 手順 (非同期モード))



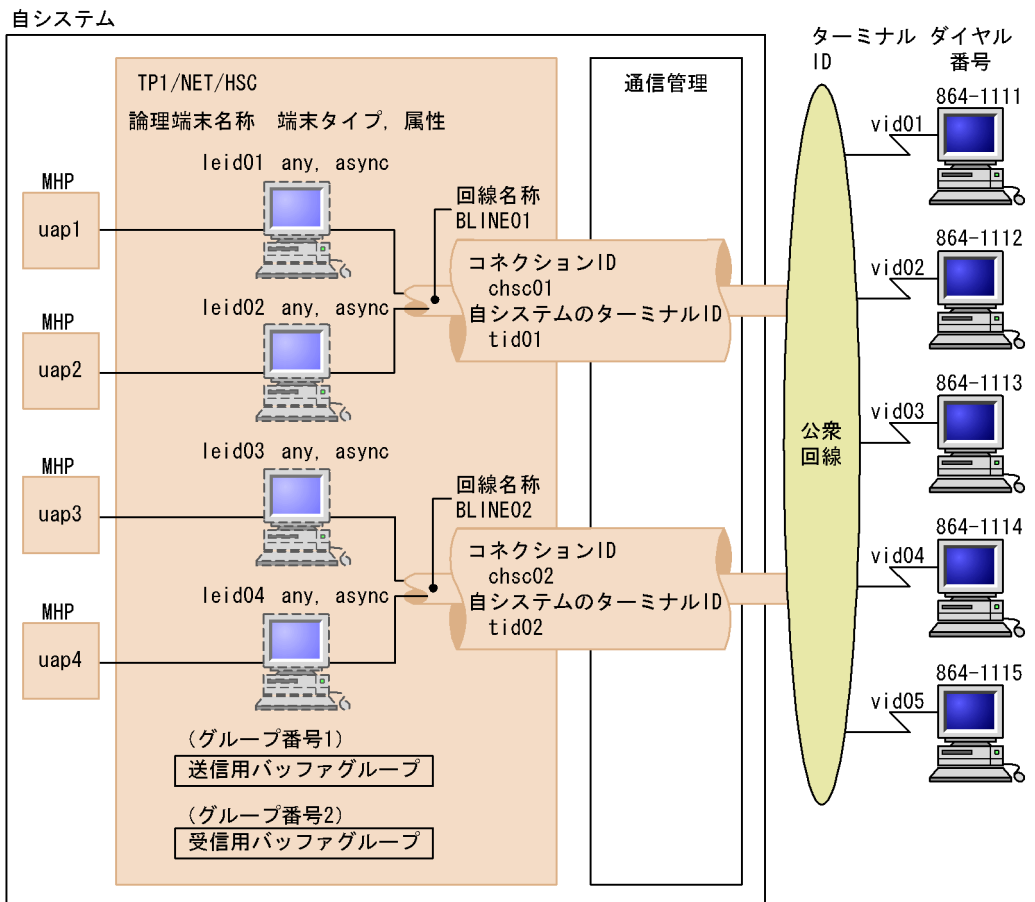
自システムの通信管理との定義の関係 (HSC2 手順 (同期モード))

TP1/NET/HSC の HSC2 手順 (同期モード) を使用する場合は、自システムの通信管理 (XNF/AS) との定義の関係を説明します。

TP1/NET/HSC の定義では、XNF/AS 構成定義と同じ回線名称を指定します。指定の方法は、TP1/NET/HSC の MCF 通信構成定義 (mcftalcen -k) で指定する回線名称と、XNF/AS のベーシック回線定義文 (baseline name) で指定する回線名称を同じにします。XNF/AS 構成定義については、マニュアル「通信管理 XNF/AS 構成定義編」を参照してください。

通信管理 XNF/AS を使用した HSC 2 手順 (同期モード) の TP1/NET/HSC のシステム構成例を次の図に示します。

図 6-6 TP1/NET/HSC のシステム構成例 (HSC 2 手順 (同期モード))



定義例

ここでは、この章の「自システムの通信管理プログラムと関連づける内容」で示したシステム構成例に沿ったコーディング例を示します。

XNF/AS の通信構成定義例

XNF/AS の通信構成定義例を示します。XNF/AS の通信構成定義例は HSC1 手順と HSC2 手順で異なります。

XNF/AS 通信構成定義例 (HSC1 手順)

```
configuration
  version 01
  max_Line_adapter 1
  max_HScline 2
  ;

Line_adapter
  name LADP01
  location_code 20-60
  adapter_type BASIC
  initial_status active
  auto_start yes
  ;

  group
    name HSC1
    type HSC1
    ;

    basiccline
      name BLINE01
      number 00
      speed 9.6K
      ;

    basiccline
      name BLINE02
      number 01
      speed 9.6K
      ;
```

注

回線アダプタの実装位置によって異なります。

XNF/AS 通信構成定義例 (HSC2 手順)

```
configuration
  version 01
  max_Line_adapter 1
  max_HScline 2
  ;

Line_adapter
  name LADP01
```



```

location_code 20-60
adapter_type BASIC
initial_status active
auto_start yes
;

group
  name HSC2
  type HSC2
;

basicline
  name BLINE01
  number 00
  speed 2.4K
  switch_type V25bis
;

basicline
  name BLINE02
  number 01
  speed 2.4K
  switch_type V25bis
;

```

注

回線アダプタの実装位置によって異なります。

HSC1 手順の定義例

HSC1 手順の TP1/NET/HSC のシステム定義例（MCF マネージャ定義の例，MCF 通信構成定義共通の例，および MCF 通信構成定義固有の例）を示します。

また，HSC1 手順の TP1/NET/HSC のシステム定義例は次のファイルで提供しています。

- MCF マネージャ定義の例：/BeTRAN/examples/mcf/HSC/conf/mngr
- MCF 通信構成定義共通の例：/BeTRAN/examples/mcf/HSC/conf/com_c1
- MCF 通信構成定義固有の例：/BeTRAN/examples/mcf/HSC/conf/com_d1

MCF マネージャ定義の例

```

###                                MCF マネージャ定義                                ###
#
### MCF マネージャ環境定義
mcfmenv      -m "name=mcfmgr"
#
### MCF マネージャ共通定義
mcfmcomn     -n 20                                ¥
              -p 300                              ¥
              -j 4096                              ¥
#
### 通信サービス定義
mcfmcname    -s "mcfsvname=_muHSC1                ¥
              syssvname=mcfu01"
mcfmcname    -s "mcfsvname=_muHSC2                ¥
              syssvname=mcfu02"
#

```

6. システム定義
定義例

```

### UAP共通定義
mcfmuap    -d 65535
           -t "sndtim=3600
              sndrcvtim=3600"
           -j 32768
           -l "initseq=1
              maxseq=9999
              minseq=1"
           -e "seqsize=8162"

#
### 入出力キュー定義
mcfmqgid   -q "quekind=otq
              quegrpId=otqgrp01"
mcfmqgid   -q "quekind=itq
              quegrpId=itqgrp01"

#
### 拡張予約定義
mcfmexp    -g 100
           -l 30
           -i dg
           -o dg

#
### 状態引き継ぎ定義
mcfmsts    -g 20
           -v 100

```

MCF 通信構成定義共通の例

```

###          MCF通信構成定義  共通定義          ###

#
### MCF環境定義
mcfteiv    -s 01
           -a apliHSC1

#
### MCF通信構成共通定義
mcfteomn   -j 32768

#
### 最大処理多重度定義
mcfteired  -m 10

#
### タイマ定義
mcfteim    -t "btim=5
              mtim=180
              rmtim=180"

#
### トレース環境定義
mcfteirc   -t "size=20480
              disk=yes
              bufcnt=50
              trccnt=3"
           -m del

#
### 状態引き継ぎ定義
mcftests   -a 10
           -l 10

#
### バッファグループ定義
mcftebuf   -g "groupno=1
              length=8162
              count=256"

```

```

                                extend=256"
#
mcftbuf      -g "groupno=2
                                length=8162
                                count=256
                                extend=256"
#
mcftbuf      -g "groupno=3
                                length=8162
                                count=256
                                extend=256"

```

MCF 通信構成定義固有の例

```

###          MCF通信構成定義   HSC1手順のprotocols固有定義          ###
#
### コネクション定義の開始
mcftalccn   -c chsc01
              -p hsc1
              -k BLINE01
              -m "mode=xnfas"
              -g "sndbuf=1
                  rcvbuf=2"
### 論理端末定義
mcftalcle   -l leid01
              -t any
              -m "dmsgcnt=20"
              -k "quekind=disk
                  quegrp=otqgrp01"
### コネクション定義の終了
mcftalced
#
### コネクション定義の開始
mcftalccn   -c chsc02
              -p hsc1
              -k BLINE02
              -m "mode=xnfas"
              -g "sndbuf=1
                  rcvbuf=2"
### 論理端末定義
mcftalcle   -l leid02
              -t any
              -m "dmsgcnt=20"
              -k "quekind=disk
                  quegrp=otqgrp01"
### コネクション定義の終了
mcftalced
              :
              :

```

HSC2 手順（非同期モード）の定義例

HSC2 手順（非同期モード）の場合，TP1/NET/HSC の固有定義以外に，回線リストの定義があります。

回線リスト定義を指定した場合，回線リスト定義ユーティリティを使用すると，名称が linefile という回線リスト定義ファイルが作成されます。

6. システム定義
定義例

HSC2 手順（非同期モード）の TP1/NET/HSC のシステム定義例（MCF 通信構成定義共通の例，MCF 通信構成定義固有の例）および回線リスト定義例を示します。

また，HSC2 手順（非同期モード）の TP1/NET/HSC のシステム定義例，および回線リスト定義例は次のファイルで提供しています。

- MCF 通信構成定義共通の例：/BeTRAN/examples/mcf/HSC/conf/com_c3
- MCF 通信構成定義固有の例：/BeTRAN/examples/mcf/HSC/conf/com_d3
- 回線リスト定義の例：/BeTRAN/examples/mcf/HSC/conf/line

MCF 通信構成定義共通の例

```
###          MCF通信構成定義  共通定義          ###
#
### MCF環境定義
mcftenv      -s  22                               ¥
              -a  mcfALPI
#
### MCF通信構成共通定義
mcftcomm
#
### トレース環境定義
mcfttrc      -t  "size=40960                       ¥
              disk=yes                             ¥
              bufcnt=10"
#
### バッファグループ定義（送信用バッファグループ）
mcftbuf      -g  "groupno=1                         ¥
              length=2048                          ¥
              count=256                             ¥
              extend=256"
#
### バッファグループ定義（受信用バッファグループ）
mcftbuf      -g  "groupno=2                         ¥
              length=2048                          ¥
              count=256                             ¥
              extend=256"
```

MCF 通信構成定義固有の例

```
# MCF通信構成定義 HSC2手順（非同期モード）の protocol固有定義 #
#
### コネクション定義の開始
mcftalccn    -c  chsc01                             ¥
              -p  hsc2                               ¥
              -g  "sndbuf=1                          ¥
                  rcvbuf=2"                         ¥
              -u  ignore                             ¥
              -n  "tid=tid01"                       ¥
              -f  "eotctrl=nouse                    ¥
                  holdtime=10"                     ¥
              -k  linefile
#
### 論理端末定義
mcftalcle    -l  leid01                             ¥
              -t  any                               ¥
```

```

-v uap ¥
-t "tid=vid01" ¥
-d "dialno=8641111" ¥
-b "attr=async"
#
mcftalcle -l leid02 ¥
-t any ¥
-v uap ¥
-t "tid=vid02" ¥
-d "dialno=8641112" ¥
-b "attr=async"
#
mcftalcle -l leid03 ¥
-t any ¥
-v uap ¥
-t "tid=vid03" ¥
-d "dialno=8641113" ¥
-b "attr=async"
#
mcftalcle -l leid04 ¥
-t any ¥
-v uap ¥
-t "tid=vid04" ¥
-d "dialno=8641114" ¥
-b "attr=async"
#
mcftalcle -l leid05 ¥
-t any ¥
-v uap ¥
-t "tid=vid05" ¥
-d "dialno=8641115" ¥
-b "attr=async"
#
### コネクション定義の終了
mcftalced
#

```

回線リスト定義の例

```

###                      回線リスト定義                      ###
#
### 回線リストの定義
mcftlnf      -o linefile
  mcftllst   -k "BLINE01"
  mcftllst   -k "BLINE02"
mcfttleof

```

HSC2 手順（同期モード）の定義例

HSC2 手順（同期モード）の場合、TP1/NET/HSC の固有定義以外に、相手ターミナル ID リスト（発信用相手ターミナル ID リスト定義、着信用相手ターミナル ID リスト定義）の定義があります。

発信用相手ターミナル ID リスト定義を指定した場合、相手ターミナル ID リスト定義ユーティリティを使用すると、名称が cn01slst, cn02slst という発信用相手ターミナル ID リストファイルが作成されます。着信用相手ターミナル ID リスト定義の場合は、名称が cn01rlst, cn02rlst という着信用相手ターミナル ID リストファイルが作成されます。

6. システム定義 定義例

HSC2 手順（同期モード）の TP1/NET/HSC のシステム定義例（MCF 通信構成定義共通の例，MCF 通信構成定義固有の例）および相手ターミナル ID リスト定義例（発信用相手ターミナル ID リスト定義の例，着信用相手ターミナル ID リスト定義の例）を示します。

また，HSC2 手順（非同期モード）の TP1/NET/HSC のシステム定義例，および相手ターミナル ID リスト定義例は次のファイルで提供しています。

- MCF 通信構成定義共通の例：/BeTRAN/examples/mcf/HSC/conf/com_c2
- MCF 通信構成定義固有の例：/BeTRAN/examples/mcf/HSC/conf/com_d2
- 発信用相手ターミナル ID リスト定義の例：/BeTRAN/examples/mcf/HSC/conf/s_trm
- 着信用相手ターミナル ID リスト定義の例：/BeTRAN/examples/mcf/HSC/conf/r_trm

MCF 通信構成定義共通の例

```
###                MCF通信構成定義  共通定義                ###
#
### MCF環境定義
mcftevn            -s  22                                ¥
                  -a  mcfALPI
#
### MCF通信構成共通定義
mftcomn
#
### トレース環境定義
mfttrc            -t  "size=40960                        ¥
                  disk=yes                               ¥
                  bufcnt=10"
#
### バッファグループ定義（送信用バッファグループ）
mftbuf            -g  "groupno=1                         ¥
                  length=2048                           ¥
                  count=256                             ¥
                  extend=256"
#
### バッファグループ定義（受信用バッファグループ）
mftbuf            -g  "groupno=2                         ¥
                  length=2048                           ¥
                  count=256                             ¥
                  extend=256"
```

MCF 通信構成定義固有の例

```
#  MCF通信構成定義 HSC2手順（同期モード）の protocols固有定義  ##
#
### コネクション定義の開始
mftalccn          -c  chsc01                            ¥
                  -p  hsc2                             ¥
                  -k  BLINE01                          ¥
                  -g  "sndbuf=1                         ¥
                  rcvbuf=2"                             ¥
                  -u  ignore                            ¥
                  -n  "tid=tid01"                       ¥
                  -o  "slist=cn01slst"                  ¥
```

```

                                rlist=cn01rlst"           ¥
                                -f "eotctrl=nouse         ¥
                                holdtime=10"
### 論理端末定義
mcftalcle -l leid01           ¥
           -t any             ¥
           -v uap1           ¥
           -b "attr=sync"
#
mcftalcle -l leid02           ¥
           -t any             ¥
           -v uap2           ¥
           -b "attr=sync"
### コネクション定義の終了
mcftalced
#
### コネクション定義の開始
mcftalccn -c chsc02           ¥
           -p hsc2           ¥
           -k BLINE02        ¥
           -g "sndbuf=1      ¥
           rcvbuf=2"         ¥
           -u ignore         ¥
           -n "tid=tid02"    ¥
           -o "slist=cn02slst ¥
           rlist=cn02rlst"   ¥
           -f "eotctrl=use   ¥
           eotexl=1          ¥
           eotexr=255        ¥
           eotttime=1"
### 論理端末定義
mcftalcle -l leid03           ¥
           -t any             ¥
           -v uap3           ¥
           -b "attr=sync"
#
mcftalcle -l leid04           ¥
           -t any             ¥
           -v uap4           ¥
           -b "attr=sync"
### コネクション定義の終了
mcftalced
:
:

```

発信用相手ターミナルIDリスト定義

```

###          発信用相手ターミナルIDリスト定義          ###
#
### 出力ファイル定義
mcftstf -o cn01slst
#
### 発信用相手ターミナルIDリスト定義
mcftstlst -t "tid=vid01"           ¥
           -d "dialno=8641111"    ¥
mcftstlst -t "tid=vid02"           ¥
           -d "dialno=8641112"    ¥
mcftstlst -t "tid=vid03"           ¥
           -d "dialno=8641113"    ¥
#
### 出力ファイル定義の終了

```

6. システム定義
定義例

```
mcfteof
#
### 出力ファイル定義
mcfststf      -o cn02slst
#
### 発信相手ターミナルIDリスト定義
mcfststlst   -t "tid=vid04"           ¥
              -d "dialno=8641114"
mcfststlst   -t "tid=vid05"           ¥
              -d "dialno=8641115"
#
### 出力ファイル定義の終了
mcfteof
```

着信用相手ターミナル ID リスト定義

```
###          着信用相手ターミナルIDリスト定義          ###
#
### 出力ファイル定義
mcfststf      -o cn01rlst
#
### 着信用相手ターミナルIDリスト定義
mcftrtlst    -t "tid=vid01"           ¥
              -l "leid=leid01"
mcftrtlst    -t "tid=vid02"           ¥
              -l "leid=leid01"
mcftrtlst    -t "tid=vid03"           ¥
              -l "leid=leid02"
#
### 出力ファイル定義の終了
mcfteof
#
### 出力ファイル定義
mcfststf      -o cn02rlst
#
### 着信用相手ターミナルIDリスト定義
mcftrtlst    -t "tid=vid04"           ¥
              -l "leid=leid03"
mcftrtlst    -t "tid=vid05"           ¥
              -l "leid=leid04"
#
### 出力ファイル定義の終了
mcfteof
```


7

運用コマンド

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

TP1/NET/HSC 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/HSC の運用コマンド

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

表 7-1 TP1/NET/HSC の運用コマンド

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

(凡例)

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

mcfactcn (コネクションの確立)

形式

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

機能

コネクションを確立します。

ただし、HSC1 手順の場合は、通信管理に対して AP 間通信の開始の準備を要求します。

また、HSC2 手順 (非同期モード) の場合、回線障害で使用されていない回線の障害を取り除き、このコマンドを入力すると、回線の回復ができます。

オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID ~ 1 ~ 8 文字の識別子

確立するコネクションのコネクション ID を指定します。

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

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

* : すべてのコネクションを確立します。

先行文字列 * : 先行文字列で始まるすべてのコネクションを確立します。

複数指定の例 cnn1, cnn2, cnn3 を指定する場合

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

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

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

7. 運用コマンド

mcftactcn (コネクションの確立)

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftactcn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftactcn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftactcn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftactcn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10500-I	ヘルプメッセージ	標準出力
KFCA14012-W	コネクション確立処理中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA14013-W	コネクション確立処理済みのため、運用コマンドは受け付けられません。	標準エラー出力
KFCA14014-W	コネクション解放処理中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA14424-W	該当するコネクション配下の全回線が起動中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	指定したコネクショングループは登録されていません。	標準エラー出力

mctactle (論理端末の閉塞解除)

形式

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

機能

論理端末の閉塞を解除します。

オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID ~ 1 ~ 8 文字の識別子

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

コネクション ID は複数指定できません。また、一括指定もできません。

-l 論理端末名称 ~ 1 ~ 8 文字の識別子

閉塞解除する論理端末の名称を指定します。

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

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

同一論理端末名称は、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。一括指定するときは、引用符 (") で囲んで指定します。

* : すべての論理端末の閉塞を解除します。

先行文字列 * : 先行文字列で始まるすべての論理端末の閉塞を解除します。

複数指定の例 len1 , len2 , len3 を指定する場合

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

一括指定の例 len で始まるすべての論理端末を指定する場合

7. 運用コマンド

mcftactle (論理端末の閉塞解除)

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

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftactle コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftactle コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftactle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftactle コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定されたコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
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
KFCA14016-W	論理端末自動起動指定のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA14017-W	論理端末が閉塞解除済みのため、運用コマンドは受け付けられません。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

注意事項

HSC1 手順で論理端末の状態が次に示す場合、このコマンドを入力して論理端末の閉塞を解除できます。

- mcftactle コマンドを入力後の明示閉塞状態
- 論理端末障害による論理端末閉塞後の暗黙閉塞状態

mcftdctcn (コネクションの解放)

形式

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

注

-f オプションは、HSC2 手順 (同期モード) の場合、必ず指定してください。

機能

コネクションを解放します。

ただし、HSC1 手順の場合は、通信管理に対して、AP 間通信の終了の準備を要求しません。

オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID ~ 1 ~ 8 文字の識別子

解放するコネクションのコネクション ID を指定します。

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

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

* : すべてのコネクションを解放します。

先行文字列 * : 先行文字列で始まるすべてのコネクションを解放します。

複数指定の例 cnn1, cnn2, cnn3 を指定する場合

```
-c "cnn1    cnn2    cnn3"
```

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

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

7. 運用コマンド

mcftdctn (コネクションの解放)

-f

該当するコネクションを強制的に解放します。

HSC1 手順または HSC2 手順 (非同期モード) の場合, 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は正常解放します。

HSC2 手順 (同期モード) の場合, このオプションは必ず指定してください。

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftdctn 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	mcftdctn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftdctn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftdctn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10501-I	ヘルプメッセージ	標準出力
KFCA14014-W	コネクション解放処理中のため, 運用コマンドは受け付けられません。	標準エラー出力
KFCA14015-W	コネクションが未確立のため, 運用コマンドは受け付けられません。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	指定したコネクショングループは登録されていません。	標準エラー出力

注意事項

HSC1 手順または HSC2 手順 (非同期モード) の場合, TP1/NET/HSC は, 送信仕掛け中にコネクションが強制解放されると, 送信メッセージを送信前の状態に戻します。このとき, 場合によっては相手システムにメッセージが送信されている場合があります。したがって, コネクション再確立後にメッセージが重複して送信されていないかどうか, メッセージ通番などを使用して確認してください。受信仕掛け中にコネクションが強制

解放されると、受信メッセージは破棄されます。

mcftdctl (論理端末の閉塞)

形式

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

機能

論理端末を強制的に閉塞します。

HSC1 手順の場合、メッセージ送信中にこのコマンドを入力すると、メッセージ送信を中止し、送信中であったメッセージは送信前の状態に戻します。その後、論理端末の閉塞解除の `mcftactl` コマンドを入力すると、メッセージを再送します。

HSC2 手順の場合のコマンド入力時に送信中であったメッセージの扱いについては、「10.1 障害の種類と対応処理」を参照してください。なお、論理端末が閉塞されても、UAP からの送信要求はできます。

オプション

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

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

`-c` コネクション ID ~ 1 ~ 8 文字の識別子

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

コネクション ID は複数指定できません。また、一括指定もできません。

`-l` 論理端末名称 ~ 1 ~ 8 文字の識別子

閉塞する論理端末の名称を指定します。

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

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

同一論理端末名称は、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。一括指定するときは、引用符 (") で囲んで指定します。

* : すべての論理端末を閉塞します。

先行文字列 * : 先行文字列で始まるすべての論理端末を閉塞します。

複数指定の例 len1, len2, len3 を指定する場合

```
-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	mcfstdctle 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	mcfstdctle 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	mcfstdctle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcfstdctle 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
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
KFCA10504-I	ヘルプメッセージ	標準出力
KFCA14016-W	論理端末自動起動指定のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA14018-W	論理端末が閉塞済みのため、運用コマンドは受け付けられません。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

7. 運用コマンド mcftdctl (論理端末の閉塞)

注意事項

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

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

mcftlscn (コネクションの状態表示)

形式

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

機能

コネクションの状態を標準出力に出力します。

オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID ~ 1 ~ 8 文字の識別子

状態を表示するコネクションのコネクション ID を指定します。

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

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

* : すべてのコネクションの状態を表示します。

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

複数指定の例 cnn1 , cnn2 , cnn3 を指定する場合

```
-c "cnn1    cnn2    cnn3"
```

一括指定の例 cnn で始まるすべてのコネクションを指定する場合

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

```
-d
```

コネクションの状態と該当するコネクションに対応する論理端末の情報および回線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。

7. 運用コマンド

mcftlscn (コネクションの状態表示)

出力形式

```
mmm cccccccc ppp sssss ddd
  llllllllll ttt uuuu
  llllllllll ttt uuuu
  :
  kkkkkkkkkk aaa eeee
  kkkkkkkkkk aaa eeee
  :
```

注 1

-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 ddd」の行だけ出力されます。

注 2

「llllllllll ttt uuuu」はコネクションに対する論理端末数と同じ行数を出力します。

注 3

「kkkkkkkkkk aaa eeee」はコネクション配下の回線数と同じ行数を出力します。

- mmm : MCF 識別子
- cccccccc : コネクション ID
- ppp : プロトコル種別
HS1...HSC1 手順
HS2...HSC2 手順
- sssss : コネクションの状態
ACT...確立
ACT/B...確立処理中
DCT...解放
DCT/B...解放処理中
- dddd : 詳細ステータス (保守情報)
- llllllll : 論理端末名称
- ttt : 論理端末タイプ
ANY...any 型
- uuuu : HSC2 手順の場合に出力されます。
TP1/NET/HSC で管理している論理端末の状態
0000...暗黙閉塞 (障害による閉塞)
0001...閉塞解除
0002...閉塞解除 (非同期スケジュール抑止)
0003...明示閉塞 (コマンドによる閉塞)
- kkkk : 回線名称
- aaa : 回線状態
ACT...回線接続

DCT...回線未接続

- eee : 保守情報

出力メッセージ

出力メッセージ ID	内容	出力先
KFCA10350-I	mcftlscn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	標準エラー出力
KFCA10359-W	mcftlscn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10360-I	状態表示を開始します。	標準出力
KFCA10361-I	標準情報を表示します。	標準出力
KFCA10362-I	詳細情報を表示します。	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10371-I	mcftlscn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftlscn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10502-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16424-E	指定したコネクショングループは登録されていません。	標準エラー出力

mcftlsle (論理端末の状態表示)

形式

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

機能

論理端末の状態を標準出力に出力します。

オプション

-s MCF 通信プロセス識別子 ~ 16 進数字 ((01 ~ ef))

MCF 通信プロセス識別子を指定します。

MCF 通信プロセス識別子は複数指定できません。

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

-c コネクション ID ~ 1 ~ 8 文字の識別子

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

-l 論理端末名称 ~ 1 ~ 8 文字の識別子

状態を表示する論理端末の名称を指定します。

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

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

同一論理端末名称は、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。一括指定するときは、引用符 (") で囲んで指定します。

* : すべての論理端末の状態を表示します。

先行文字列 * : 先行文字列で始まるすべての論理端末の状態を表示します。

複数指定の例 len1, len2, len3 を指定する場合

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

一括指定の例 len で始まるすべての論理端末を指定する場合

-l "len*"

-q

指定した論理端末に対応する出力キューの保留状態を表示します。

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

-t

指定した論理端末がセキュア状態かどうかを表示します。

出力形式

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

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

7. 運用コマンド

mcftlsle (論理端末の状態表示)

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
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

8

組み込み方法

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

-
- 8.1 TP1/NET/HSC の組み込みの流れ
 - 8.2 MCF メイン関数の作成
 - 8.3 定義オブジェクトファイルの生成
-

8.1 TP1/NET/HSC の組み込みの流れ

TP1/NET/HSC 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/HSC を起動するためには、MCF メイン関数をコーディングし、コンパイル、およびリンケージしておく必要があります。詳細は、「8.2 MCF メイン関数の作成」を参照してください。

(2) MCF サービス名の登録

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

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

詳細は、マニュアル「OpenTP1 システム定義」を参照してください。

(3) システムサービス情報定義ファイルの作成

システムサービス情報定義ファイルをテキストエディタで作成します。作成するファイルのパス名は、「\$DCDIR/lib/sysconf/ システムサービス情報定義ファイル名」です。ファイルの定義形式については、6 章の「システムサービス情報定義」を参照してください。

(4) 定義オブジェクトファイルの生成

OpenTP1 のネットワークコミュニケーション定義の各ソースファイルから定義オブジェクトファイルを生成します。詳細は、「8.3 定義オブジェクトファイルの生成」を参照してください。

8.2 MCF メイン関数の作成

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

TP1/NET/HSC を起動するためには、ユーザが MCF メイン関数を作成し、コンパイル、およびリンケージを行って TP1/NET/HSC 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、HSC1 手順の場合は `mcfplhs1` コマンド、HSC2 手順の場合は `mcfplhs2` コマンドを使用します。

MCF メイン関数では、スタート関数 (`dc_mcf_svstart`) を呼び出します。UOC を使用する場合は、MCF メイン関数で UOC の関数アドレスを指定してください。

UOC は、MCF メイン関数と同じ言語 (ANSI C、C++ または K&R 版 C) で作成してください。

MCF メイン関数のコーディング概要を図 8-1 と図 8-2 に示します。また、ディレクトリへの組み込み方法を図 8-3 に示します。

なお、これらのコーディング例は、次のファイルで提供しています。

【ANSI C、C++ の場合】

HSC 手順 1 : `/BeTRAN/examples/mcf/HSC/cmlib/ansi/com1.c`

HSC 手順 2 : `/BeTRAN/examples/mcf/HSC/cmlib/ansi/com2.c`

【K&R 版 C の場合】

HSC 手順 1 : `/BeTRAN/examples/mcf/HSC/cmlib/c/com1.c`

HSC 手順 2 : `/BeTRAN/examples/mcf/HSC/cmlib/c/com2.c`

8. 組み込み方法

図 8-1 MCF メイン関数のコーディング概要 (ANSI C, C++ の場合)

```

#include <dcmhsc1m.h>※                               /*TP1/NET/HSC用ヘッダファイル */ 1.
extern DCLONG ledtmn01(dcmcf_uoc_min_n *);           /*論理端末名称決定UOC関数extern宣言 */
extern DCLONG msgrcv01(dcmcf_uoc_min_n *);           /*入力メッセージ編集UOC関数extern宣言*/ 2.
extern DCLONG msgsend01(dcmcf_uoc_mout_n *);         /*出力メッセージ編集UOC関数extern宣言*/

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

int main()
{
    dcmcf_uoctbl.ledtmn = (dcmcf_uocfunc)ledtmn01;     /*論理端末名称決定UOCアドレス設定 */
    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 */ 4.

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

```

注

HSC1 手順の場合のコーディングです。HSC2 手順の場合は、次のようにコーディングをしてください。

```
#include<dcmhsc2m.h>
```

図 8-2 MCF メイン関数のコーディング概要 (K&R 版 C の場合)

```

#include <dcmhsc1m.h>※                               /*TP1/NET/HSC用ヘッダファイル */ 1.
extern DCLONG ledtmn01();                             /*論理端末名称決定UOC関数extern宣言 */
extern DCLONG msgrcv01();                             /*入力メッセージ編集UOC関数extern宣言*/ 2.
extern DCLONG msgsend01();                             /*出力メッセージ編集UOC関数extern宣言*/

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

main()
{
    dcmcf_uoctbl.ledtmn = ledtmn01;                   /*論理端末名称決定UOCアドレス設定 */
    dcmcf_uoctbl.msgrcv = msgrcv01;                   /*入力メッセージ編集UOCアドレス設定 */
    dcmcf_uoctbl.msgsend = msgsend01;                 /*出力メッセージ編集UOCアドレス設定 */ 4.

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

```

注

HSC1 手順の場合のコーディングです。HSC2 手順の場合は、次のようにコーディングをしてください。

```
#include<dcmhsc2m.h>
```

1. TP1/NET/HSC で提供するヘッダファイルを取り込みます。
2. 使用する UOC 関数を `extern` 宣言します。UOC のリターン値は `DCLONG` 型にしてください。

UOC をまったく使用しない場合、このコーディングは必要ありません。

論理端末名称決定 UOC は、HSC2 手順（非同期モード）の場合だけ使用できます。

3. UOC テーブルを `extern` 宣言します。UOC を使用する場合、必ずこのとおりにコーディングしてください。

UOC をまったく使用しない場合、このコーディングは必要ありません。

4. 各 UOC 関数のアドレスを、次に示すシステム提供変数に設定します。使用する UOC だけコーディングしてください。

```
dcmcf_uoctbl.ledtmn /*論理端末名称決定UOCアドレス*/
dcmcf_uoctbl.msgrev /*入力メッセージ編集UOCアドレス*/
dcmcf_uoctbl.msgsend /*出力メッセージ編集UOCアドレス*/
```

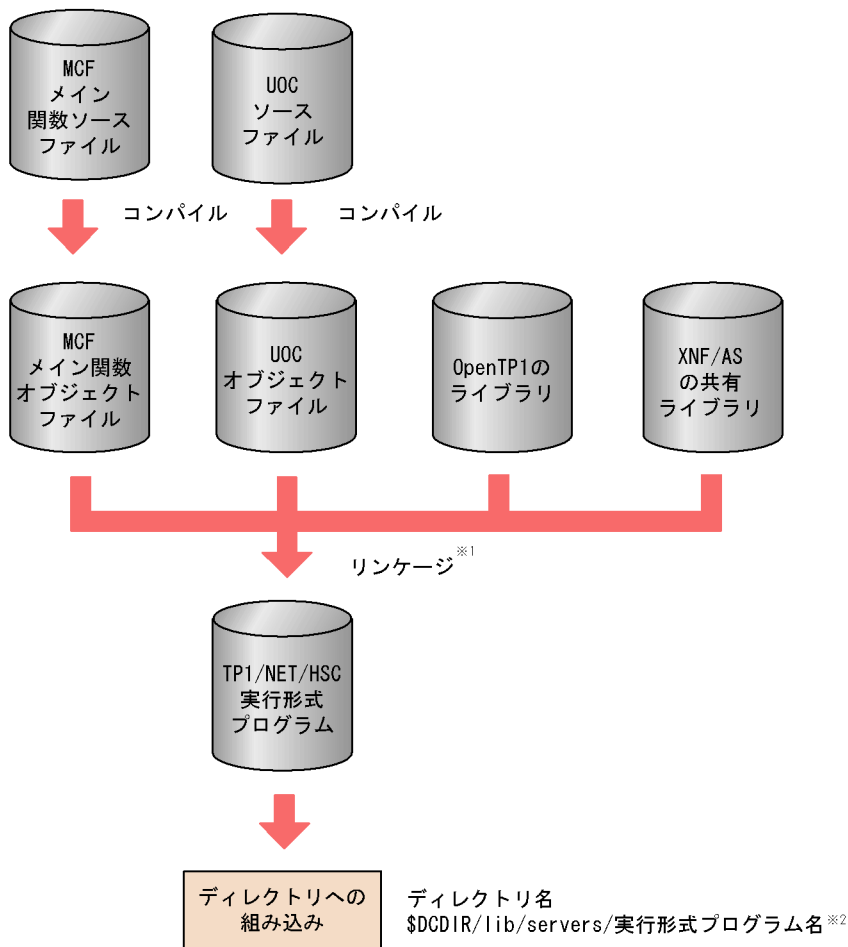
UOC をまったく使用しない場合、このコーディングは必要ありません。

論理端末名称決定 UOC は、HSC2 手順（非同期モード）の場合だけ使用できます。

5. スタート関数を呼び出します。MCF メイン関数には必ずコーディングしてください。

8. 組み込み方法

図 8-3 MCF メイン関数のディレクトリへの組み込み方法



注 1

HSC1 手順の場合は mcfplhs1 コマンド，HSC2 手順の場合は mcfplhs2 コマンドでリンケージします。

注 2

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

8.3 定義オブジェクトファイルの生成

定義オブジェクトファイルを次の手順で生成します。

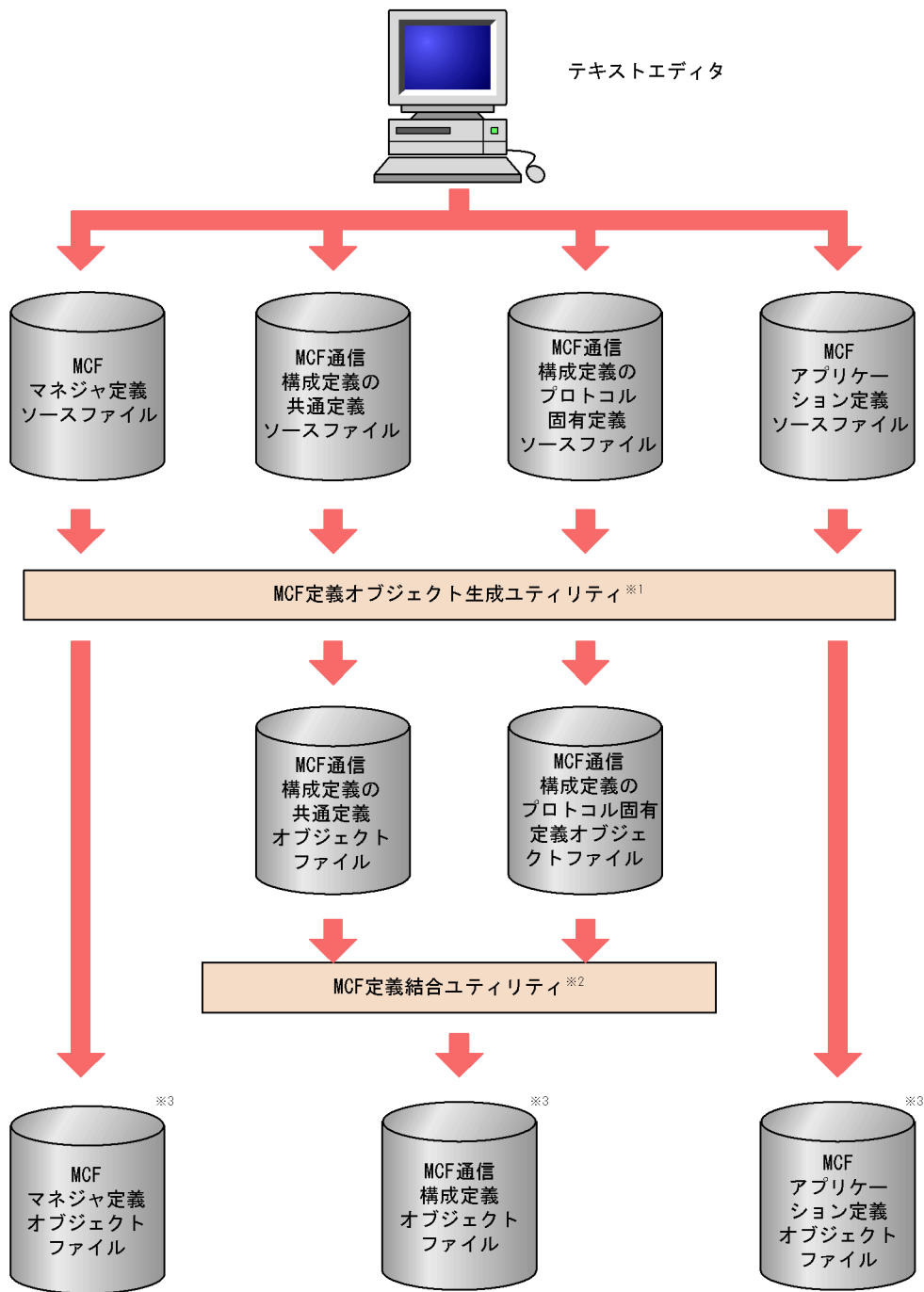
ただし、開始から再開始の間に定義オブジェクトファイルを変更しないでください。変更した場合、再開始時に正常に動作しない場合がありますのでご注意ください。

1. テキストエディタで、MCF の定義ファイルから、次に示す定義ソースファイルを作成します。
 - MCF マネージャ定義ソースファイル
 - MCF 通信構成定義の共通定義ソースファイル
 - MCF 通信構成定義の TP1/NET/HSC のプロトコル固有定義ソースファイル
 - 回線リスト定義ソースファイル（HSC2 手順（非同期モード）の場合）
 - 相手ターミナル ID リスト定義ソースファイル（HSC2 手順（同期モード）の場合）
 - 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/HSC のプロトコル固有定義オブジェクトファイル
 - 回線リスト定義オブジェクトファイル（HSC2 手順（非同期モード）の場合）
 - 相手ターミナル ID リスト定義オブジェクトファイル（HSC2 手順（同期モード）の場合）
 - MCF アプリケーション定義オブジェクトファイル
3. MCF 定義結合ユーティリティを使用して、MCF 通信構成定義の共通定義とプロトコル固有定義のオブジェクトファイルを結合し、次に示すオブジェクトファイルを作成します。
 - MCF 通信構成定義オブジェクトファイル

定義オブジェクトファイルの作成方法の概要を次の図に示します。

8. 組み込み方法

図 8-4 定義オブジェクトファイルの作成方法の概要



注 1
次に示すコマンドで生成します。

```
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。

- mcfmngr : MCF マネ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
- mcfhsc1 : MCF 通信構成定義のプロトコル (TP1/NET/HSC の HSC1 手順) 固有定義ソースファイル
- mcfhsc2 : MCF 通信構成定義のプロトコル (TP1/NET/HSC の HSC2 手順) 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 mcfhsc1 および mcfhsc2 コマンドについては 6 章の「MCF 定義オブジェクトの生成」を、その他のコマンドについてはマニュアル「OpenTP1 システム定義」を参照してください。

注 2

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

```
mcfmlink -i 共通定義オブジェクトファイル名
          TP1/NET/HSC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。

9

HSC1 手順の障害対策

この章では、TP1/NET/HSC の HSC1 手順の運用中に発生する障害と、TP1/NET/HSC の HSC1 手順の対応処理、およびメッセージの処理について説明します。

9.1 障害の種類と対応処理

9.2 コネクション障害

9.3 ユーザアプリケーションプログラム異常終了

9.4 ユーザアプリケーションプログラム閉塞

9.5 入力キュー障害

9.6 出力キュー障害

9.1 障害の種類と対応処理

HSC1 手順の場合の TP1/NET/HSC 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

コネクション障害時の TP1/NET/HSC の障害処理については、「9.2 コネクション障害」を参照してください。

表 9-1 コネクション障害発生時の処理 (HSC1 手順)

障害の内容	TP1/NET/HSC の処理	ユーザの処理
コネクション確立時障害 (回復不可能)	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 3. CERREVT (コネクションの確立不可能) を起動します。 	障害の要因を取り除いたあと、運用コマンド (mctactcn) を入力して、再びコネクションを確立します。
コネクション切断時障害 (回復不可能)	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 3. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (コネクションの切断, コネクション回復不可能) を起動します。 	障害の要因を取り除いたあと、運用コマンド (mctactcn) を入力して、再びコネクションを確立します。
受信バッファオーバーフロー	<ol style="list-style-type: none"> 1. 受信メッセージの破棄, および受信メッセージのオーバーフローを通知するメッセージログ (KFCA14300-W) を出力します。 2. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 3. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (受信バッファオーバーフロー) を起動します。 	バッファグループ定義を修正してください。

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ送信時障害（回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. コネクションの解放を通知するメッセージログ（KFCA14001-I）を出力します。 3. コネクションの閉塞を通知するメッセージログ（KFCA14002-E）を出力します。 4. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 5. CERREVT（コネクションの切断，コネクション回復不可能）を起動します。 6. メッセージは出力キューに残し，未送信扱いとします。 	<p>障害の要因を取り除いたあと，運用コマンド（mcftacten）を入力して，再びコネクションを確立します。</p>
メッセージ送信時回線上の HSC 手順エラー（回復可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. メッセージの送信中断を通知するメッセージログ（KFCA10608-W）を出力します。 3. 送信を中断したメッセージを取り出し，前の状態に戻します。 4. コネクションの解放を通知するメッセージログ（KFCA14001-I）を出力します。 5. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 6. CERREVT（回線上の HSC 手順エラー，回復可能およびリトライ実行中）を起動します。 7. コネクションの回復動作を通知するメッセージログ（KFCA14501-I）を出力します。 8. 回復障害時，COPNEVT を起動します。 9. コネクションの確立を通知するメッセージログ（KFCA14000-I）を出力します。 10. 論理端末の閉塞解除を通知するメッセージログ（KFCA14003-I）を出力します（論理端末自動起動の場合）。 11. 相手システムからのメッセージを受信したあとに，メッセージを再送します。 	<ul style="list-style-type: none"> • 障害が回復する前に，送信要求が発生した場合，運用コマンド（mcftactle）を入力してください（自システム起動によるコネクション再確立の場合）。 • 障害回復動作を中止する場合，運用コマンド（mcftdcten）を入力してください。

9. HSC1 手順の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ受信時障害（回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. 受信メッセージの破棄、およびコネクションの切断を通知するメッセージログ（KFCA14300-W）を出力します。 3. コネクションの解放を通知するメッセージログ（KFCA14001-I）を出力します。 4. コネクションの閉塞を通知するメッセージログ（KFCA14002-E）を出力します。 5. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 6. CERREVT（コネクションの切断、コネクション回復不可能）を起動します。 	<p>障害の要因を取り除いたあと、運用コマンド（mcftactcn）を入力して、再びコネクションを確立します。</p>
メッセージ受信時回線上の HSC 手順エラー（回復可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. 受信メッセージの破棄、および回線上の HSC 手順エラーを通知するメッセージログ（KFCA14300-W）を出力します。 3. コネクションの解放を通知するメッセージログ（KFCA14001-I）を出力します。 4. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 5. CERREVT（回線上の HSC 手順エラー、回復可能およびリトライ実行中）を起動します。 6. コネクションの回復動作を通知するメッセージログ（KFCA14501-I）を出力します。 7. 回復障害時、COPNEVT を起動します。 8. コネクションの確立を通知するメッセージログ（KFCA14000-I）を出力します。 9. 論理端末の閉塞解除を通知するメッセージログ（KFCA14003-I）を出力します（論理端末自動起動の場合）。 10. 相手システムからのメッセージを受信します。 	<ul style="list-style-type: none"> • 障害が回復する前に、送信要求が発生した場合、運用コマンド（mcftactle）を入力してください（自システム起動によるコネクション再確立の場合）。 • 障害回復動作を中止する場合、運用コマンド（mcftdctcn）を入力してください。

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

入力キュー障害時の障害処理については、「9.5 入力キュー障害」を参照してください。

表 9-2 受信スケジュール関係の障害発生時の処理（HSC1 手順）

障害の内容	TP1/NET/HSC の処理	ユーザの処理
アプリケーション名未定義	ERREVT1 を起動します。	アプリケーション定義を修正してください。
MCF イベントの名称未定義 • ERREVTx • SxxxEVT • CxxxEVT	メッセージ、および MCF イベントを破棄します。	ありません。

障害の内容	TP1/NET/HSC の処理	ユーザの処理
<ul style="list-style-type: none"> • MHP サービス、サービスグループ閉塞 • 入力キュー書き込み障害 	<ol style="list-style-type: none"> 1. ERREVT2 を起動します。 2. 入力キュー障害を通知するメッセージログ (KFCA10604-E) を出力します。 	ありません。
入力メッセージ編集 UOC エラーリターン	<ol style="list-style-type: none"> 1. UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 2. 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。 3. 論理端末を閉塞します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (UOC 障害) を起動します。 	ありません。
入力メッセージ編集 UOC パラメタ不正	<ol style="list-style-type: none"> 1. UOC パラメタ不正を通知するメッセージログ (KFCA10620-E) を出力します。 2. 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。 3. 論理端末を閉塞します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (UOC 障害) を起動します。 	UOC を見直してください。
アプリケーション名形式不正	<ol style="list-style-type: none"> 1. アプリケーション名の取得失敗を通知するメッセージログ (KFCA10610-E) を出力します。 2. ERREVT1 を起動します。 	UOC を見直してください。
メッセージ受信 (ETB) 後の EOT 受信	受信メッセージの破棄、および ETB 受信後の EOT 受信を通知するメッセージログ (KFCA14300-W) を出力します。	ありません。

9. HSC1 手順の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
受信バッファ不足	<ol style="list-style-type: none"> 1. 受信バッファ不足を通知するメッセージログ (KFCA10618-E) を出力します。 2. グループバッファ取得失敗を通知するメッセージログ (KFCA14005-E) を出力します。 3. 受信メッセージの破棄を通知するメッセージログ (KFCA14300-W) を出力します (ETB ブロック受信済みの場合)。 4. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 5. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 6. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 7. CERREVT (受信バッファ不足) を起動します。 	受信バッファの数を見直してください。
入力キュー書き込み後の障害	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 3. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (コネクション障害) を起動します。 	障害回復後、相手システムからメッセージが二重に送信される場合があります。メッセージ通番で二重送信をチェックしてください。

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

出力キュー障害時の障害処理については、「9.6 出力キュー障害」を参照してください。

表 9-3 送信スケジュール関係の障害発生時の処理 (HSC1 手順)

障害の内容	TP1/NET/HSC の処理	ユーザの処理
出力キュー読み込み障害	<ol style="list-style-type: none"> 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. CERREVT (出力キュー読み込み障害) を起動します。 3. 該当するメッセージを消去します。 4. 論理端末を閉塞します。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 	相手システムに出力キュー障害を通知 (send) してください。
出力メッセージ編集 UOC エラーリターン	<ol style="list-style-type: none"> 1. UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 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. CERREVT (UOC 障害) を起動します。 4. 該当するメッセージを消去します。 5. 論理端末を閉塞します。 6. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 	相手システムに出力キュー障害を通知 (send) してください。

障害の内容	TP1/NET/HSC の処理	ユーザの処理
出力メッセージ編集 UOC パラメタ不正	<ol style="list-style-type: none"> 1. UOC パラメタ不正を通知するメッセージログ (KFCA10620-E) を出力します。 2. 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. CERREVT (UOC 障害) を起動します。 4. 該当するメッセージを消去します。 5. 論理端末を閉塞します。 6. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 	相手システムに出力キュー障害を通知 (send) してください。また、UOC を見直してください。
メッセージ消し込み障害	<ol style="list-style-type: none"> 1. メッセージ送信完了障害を通知するメッセージログ (KFCA10617-E) を出力します。 2. 処理を続行します。 	ありません。
送信バッファオーバーフロー	<ol style="list-style-type: none"> 1. 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. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 3. 論理端末を閉塞します。 4. 該当するメッセージを消去します。 5. CERREVT (受信バッファオーバーフロー) を起動します。 	バッファグループ定義、または UAP を修正してください。
送信バッファ不足	<ol style="list-style-type: none"> 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. 該当するメッセージを取り出し前の状態に戻します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. 論理端末を閉塞します。 6. CERREVT (送信バッファ不足) を起動します。 	一定時間経過後、運用コマンド (mcftactle) を入力して論理端末の閉塞を解除し、メッセージを再送信してください。また、送信バッファの数を見直してください。
出力キュー消去前の障害	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 3. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (コネクション障害) を起動します。 	障害回復後、相手システムへメッセージが二重に送信される場合があります。メッセージ通番で二重送信をチェックしてください。

9. HSC1 手順の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
送信テキスト形態不正	<ol style="list-style-type: none"> 1. 送信テキスト形態不正を通知するメッセージログ (KFCA14502-E) を出力します。 2. 該当するメッセージを消去します。 3. CERREVT (論理端末レベル障害, 送信テキスト形態不正) を起動します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 	送信テキスト形態の設定を見直してください。
テキスト送信時の EOT 受信	<ol style="list-style-type: none"> 1. 該当するメッセージを取り出し前の状態に戻します。 2. 例外事項発生を通知するメッセージログ (KFCA14301-W) を出力します。 3. メッセージ送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 4. CERREVT (論理端末レベル障害, テキスト送信時の EOT 受信) を起動します。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 	一定時間経過後, 運用コマンド (mcftactle) を入力して論理端末の閉塞を解除し, メッセージを再送信してください。また, 該当するメッセージの内容が不正ならば mcftdlqle コマンドなどでメッセージを削除してください。

(4) UAP 障害

表 9-4 UAP 障害発生時の処理 (HSC1 手順)

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

(5) MCF の障害

表 9-5 MCF の障害発生時の処理 (HSC1 手順)

障害の内容	TP1/NET/HSC の処理	ユーザの処理
<ul style="list-style-type: none"> • 内部論理矛盾 • 他 PP とのインタフェースエラー 	<ol style="list-style-type: none"> 1. 障害情報の取得を通知するメッセージログ (KFCA10189-E) を出力します。 2. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 3. CERREVT (コネクションレベル) を起動します。(コネクション障害の場合) 	ありません。

9.2 コネクション障害

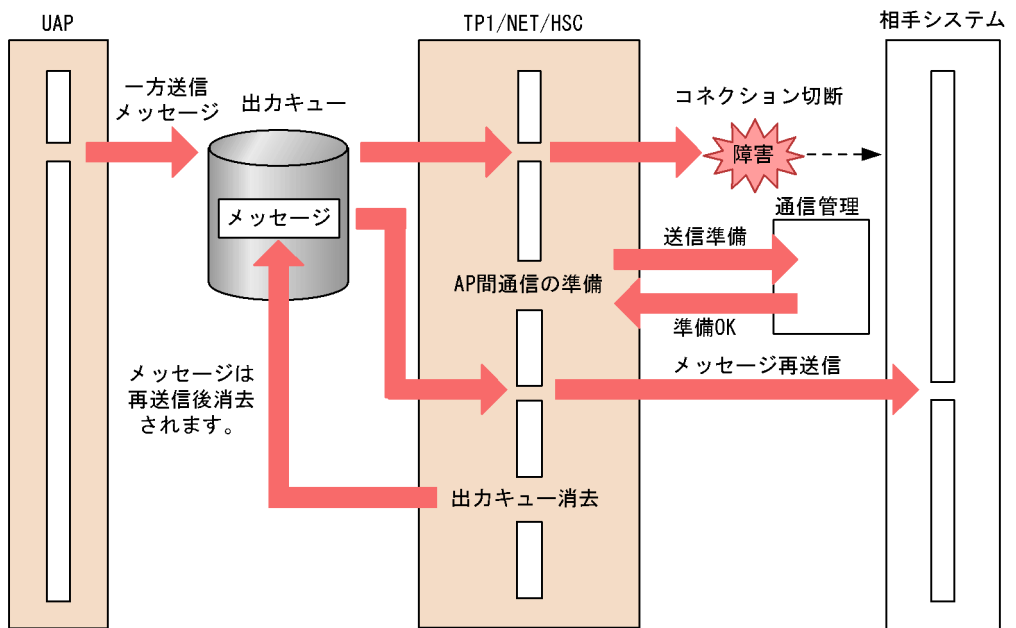
コネクションが切断された場合、TP1/NET/HSC は自動的に回復しません。通信管理が管理している通信機器に障害が発生した場合も同様です。障害の要因を取り除いたあと、運用コマンド (mcftactcn) を入力してコネクションの再確立してください。

9.2.1 メッセージ送信時のコネクション障害

(1) メッセージ送信前のコネクション障害

メッセージはロールバックされ、AP 間通信の開始の準備ができたあとに再送信されます。メッセージ送信前のコネクション障害の例を次の図に示します。

図 9-1 メッセージ送信前のコネクション障害 (HSC1 手順)

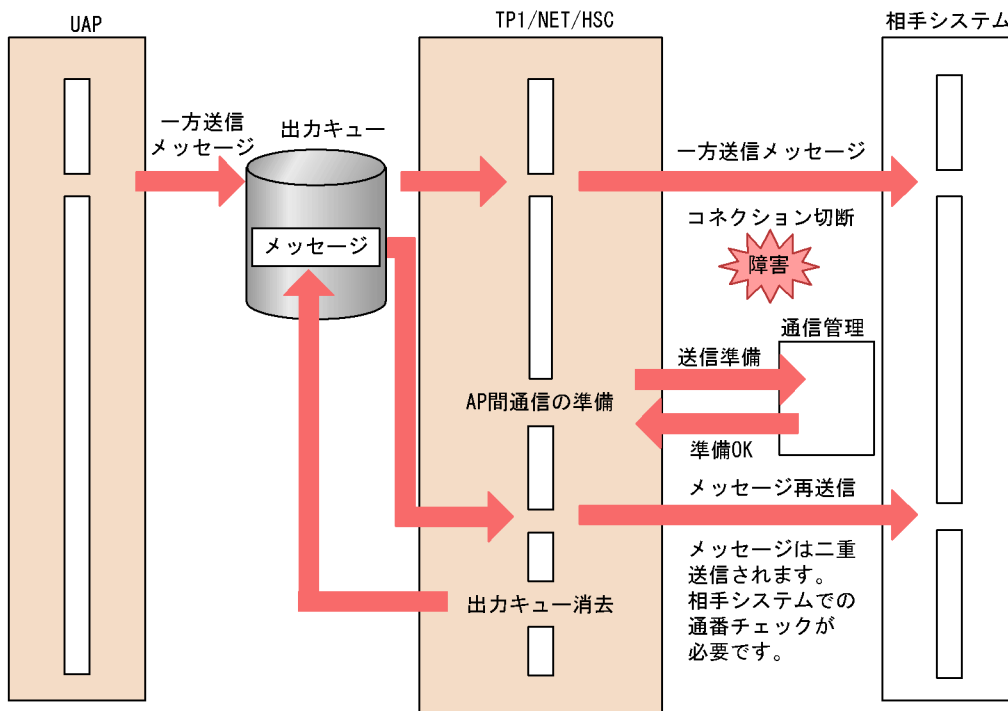


(2) 出力キュー消去前のコネクション障害

メッセージは二重に送信されます。このとき相手システムの UAP は、メッセージに通番を付けることで、二重送信をチェックします。

メッセージ消去前のコネクション障害の例を次の図に示します。

図 9-2 メッセージ消去前の接続障害 (HSC1 手順)



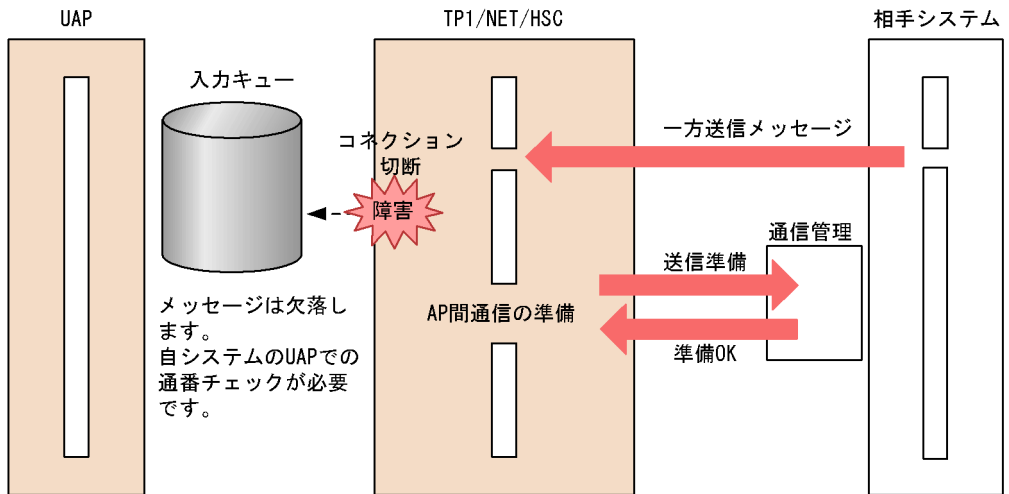
9.2.2 メッセージ受信時の接続障害

(1) 入力キュー書き込み前の接続障害

メッセージは欠落します。このとき自システムのUAPは、メッセージに通番を付けることで、欠落しているかどうかをチェックします。

メッセージ書き込み前の接続障害の例を次の図に示します。

図 9-3 メッセージ書き込み前の接続障害 (HSC1 手順)

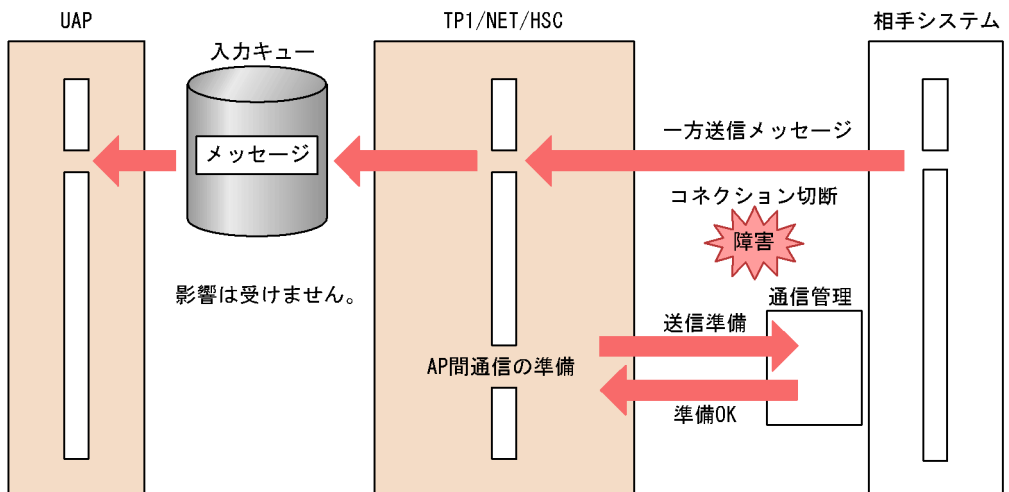


(2) 入力キュー書き込み後の接続障害

メッセージは影響を受けません。

メッセージ書き込み後の接続障害の例を次の図に示します。

図 9-4 メッセージ書き込み後の接続障害 (HSC1 手順)



9.3 ユーザアプリケーションプログラム異常終了

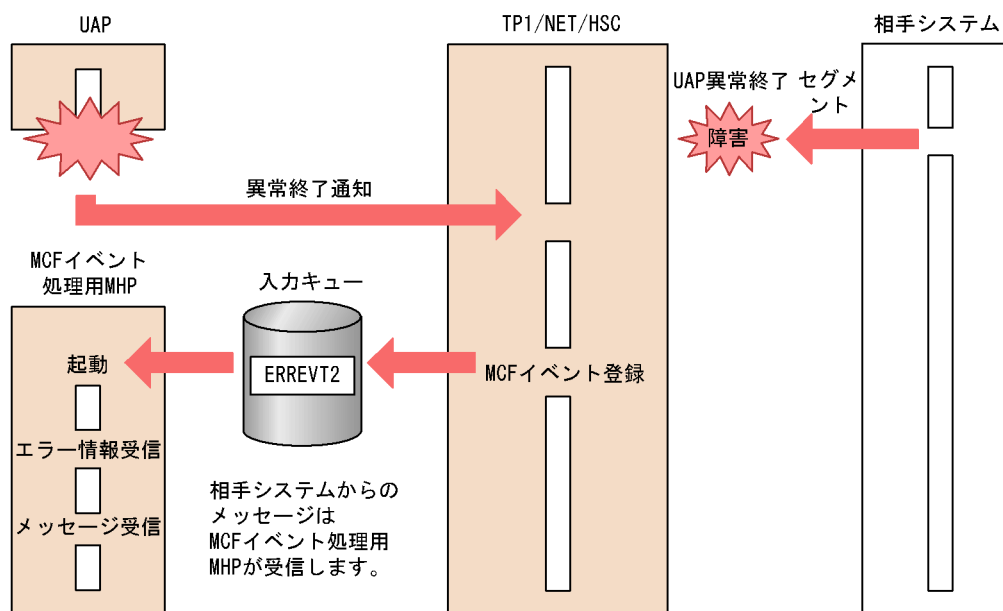
UAP が異常終了した場合、TP1/NET/HSC は入力キューに MCF イベントを登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

9.3.1 セグメント受信前の UAP 異常終了

TP1/NET/HSC は、入力キューに ERREVT2 (メッセージ破棄通知イベント) を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

セグメント受信前の UAP 異常終了の例を次の図に示します。

図 9-5 セグメント受信前の UAP 異常終了 (HSC1 手順)

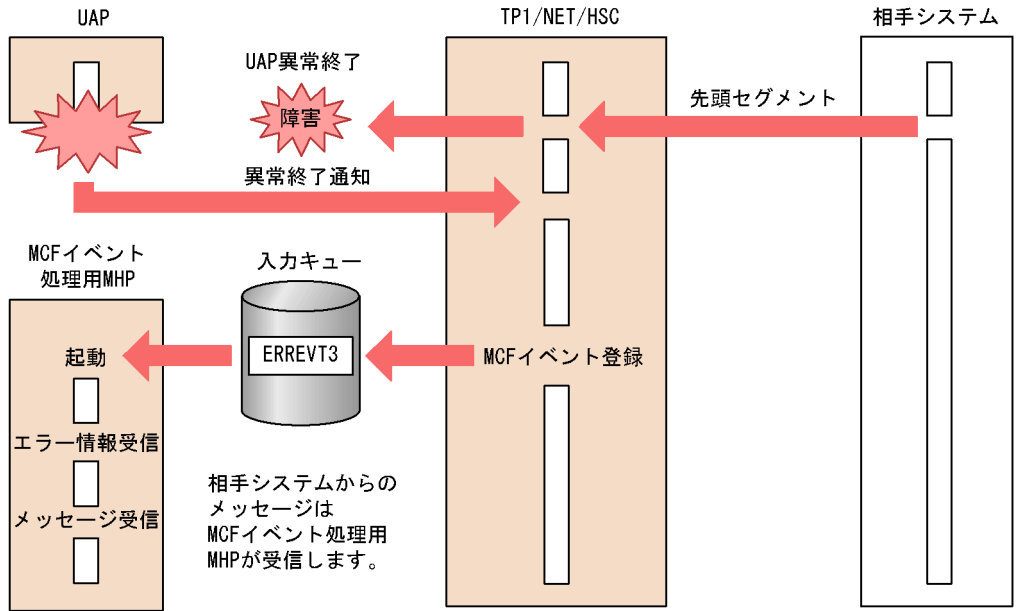


9.3.2 セグメント受信後の UAP 異常終了

TP1/NET/HSC は、入力キューに ERREVT3 (UAP 異常終了通知イベント) を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

セグメント受信後の UAP 異常終了の例を次の図に示します。

図 9-6 セグメント受信後の UAP 異常終了 (HSC1 手順)

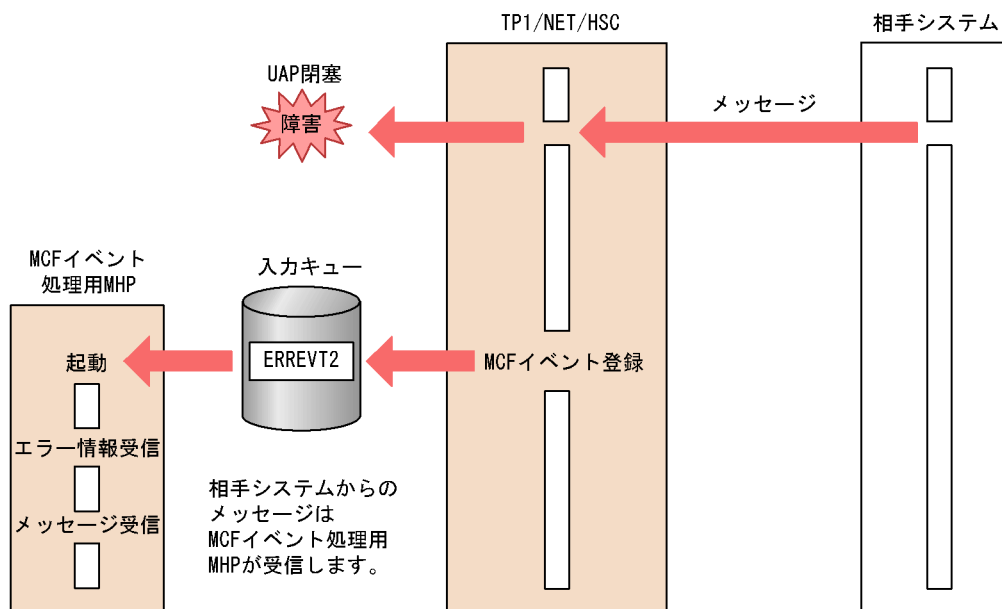


9.4 ユーザアプリケーションプログラム閉塞

UAP が閉塞していて使用できない場合，TP1/NET/HSC は入力キューに ERREVT2（メッセージ破棄通知イベント）を登録し，MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は，エラー情報と入力されたメッセージを受信します。

UAP 閉塞による障害の例を次の図に示します。

図 9-7 UAP 閉塞による障害（HSC1 手順）

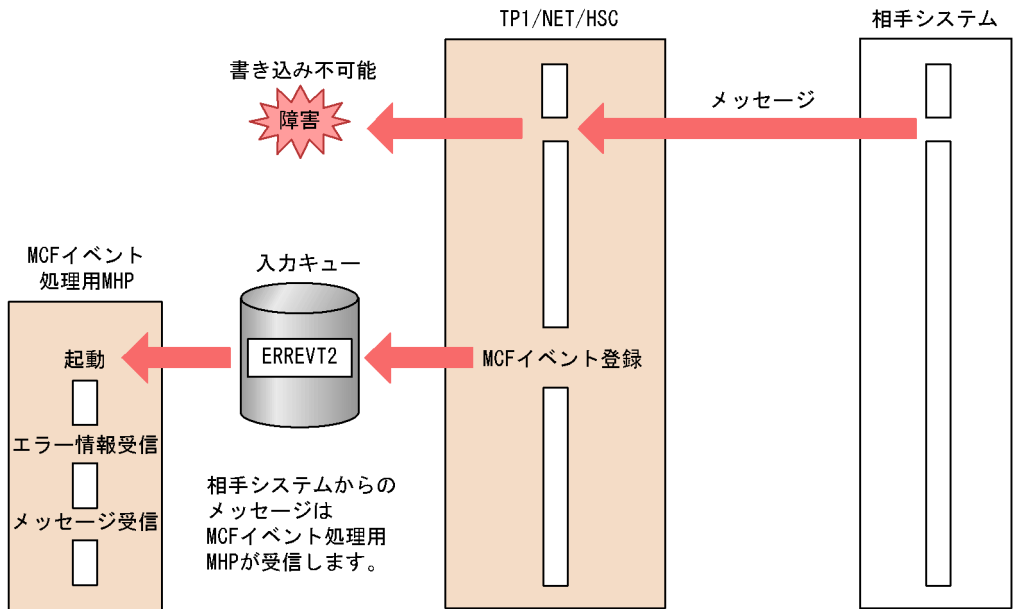


9.5 入力キュー障害

メッセージを入力キューに書き込むときに障害が発生した場合、TP1/NET/HSC は入力キューに ERREVT2 (メッセージ破棄通知イベント) を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

入力キュー障害の例を次の図に示します。

図 9-8 入力キュー障害 (HSC1 手順)

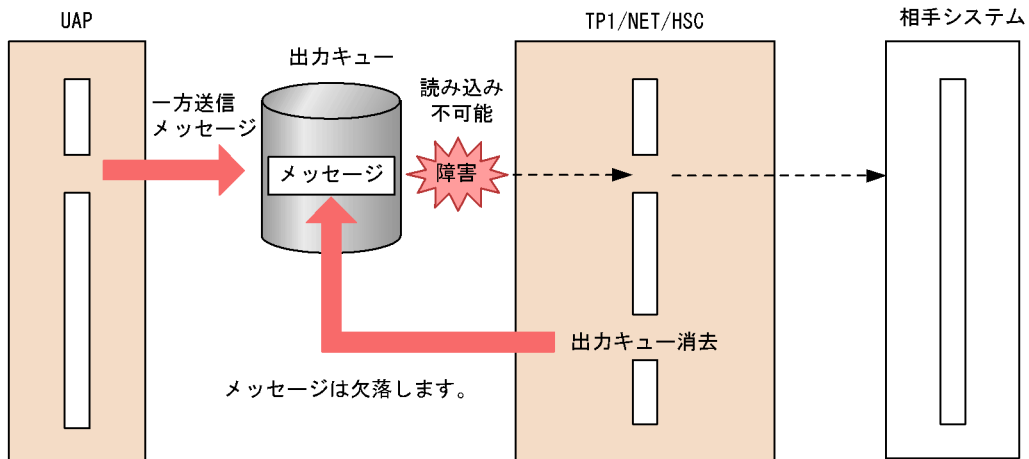


9.6 出力キュー障害

メッセージを出力キューから読み込むときに障害が発生した場合、メッセージは欠落します。

出力キュー障害の例を次の図に示します。

図 9-9 出力キュー障害 (HSC1 手順)



10 HSC2 手順（非同期モード） の障害対策

この章では、TP1/NET/HSC の HSC2 手順（非同期モード）の運用中に発生する障害と、HSC2 手順（非同期モード）の対応処理、およびメッセージの処理について説明します。

10.1 障害の種類と対応処理

10.2 コネクション障害

10.3 ユーザアプリケーションプログラム異常終了

10.4 ユーザアプリケーションプログラム閉塞

10.5 入力キュー障害

10.6 出力キュー障害

10.1 障害の種類と対応処理

HSC2 手順（非同期モード）の場合の TP1/NET/HSC 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）通信障害

コネクション障害時の TP1/NET/HSC の障害処理については、「10.2 コネクション障害」を参照してください。

表 10-1 通信障害発生時の処理（HSC2 手順（非同期モード））

障害の内容	TP1/NET/HSC の処理	ユーザの処理
コネクション確立時障害（回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 3. コネクションの閉塞を通知するメッセージログ（KFCA14002-E）を出力します。 4. CERREVT（コネクションの確立不可能）を起動します。 	障害の要因を取り除いたあと、運用コマンド（mefactcn）を入力して、再びコネクションを確立します。
回線接続前障害（回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. 回線をクローズします。 3. 回線クローズを通知するメッセージログ（KFCA14504-I）を出力します。 	障害の要因を取り除いたあと、運用コマンド（mefactcn）を入力して、再びコネクションを確立します。
コネクション切断障害（全回線回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 3. コネクションの閉塞を通知するメッセージログ（KFCA14002-E）を出力します。 4. CERREVT（コネクションの切断）を起動します。 	障害の要因を取り除いたあと、運用コマンド（mefactcn）を入力して、再びコネクションを確立します。
メッセージ送信時障害（回線回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. メッセージの送信中断を通知するメッセージログ（KFCA10608-W）を出力します。 3. 送信メッセージを出力キューから取り出す前の状態に戻します。 4. SERREVT を起動します（send 関数で設定した場合）。 5. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 6. CERREVT（メッセージ送信中の回線回復不可能障害）を起動します。 7. 回線の切断を通知するメッセージログ（KFCA14411-I）を出力します。 8. 回線をクローズします。 9. 回線のクローズを通知するメッセージログ（KFCA14504-I）を出力します。 	障害の要因を取り除いたあと、運用コマンド（mefactcl）を入力して、論理端末の閉塞を解除します。

障害の内容	TP1/NET/HSC の処理	ユーザの処理
回線接続維持中障害（回線回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 3. CERREVT（回線維持中の回線回復不可能障害）を起動します。 4. 回線の切断を通知するメッセージログ（KFCA14411-I）を出力します。 5. 回線をクローズします。 6. 回線のクローズを通知するメッセージログ（KFCA14504-I）を出力します。 	<p>障害の要因を取り除いたあと、運用コマンド（mcfactle）を入力して、論理端末の閉塞を解除します。</p>
メッセージ受信時障害（回線回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. 受信テキストの破棄を通知するメッセージログ（KFCA14401-W）を出力します。 3. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 4. CERREVT（メッセージ受信中の回線回復不可能障害）を起動します。 5. 回線の切断を通知するメッセージログ（KFCA14411-I）を出力します。 6. 回線をクローズします。 7. 回線のクローズを通知するメッセージログ（KFCA14504-I）を出力します。 	<p>障害の要因を取り除いたあと、運用コマンド（mcfacten）を入力して、再びコネクションを確立します。</p>
メッセージ送信時の HSC 手順エラー	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. メッセージの送信中断を通知するメッセージログ（KFCA10608-W）を出力します。 3. 送信メッセージを出力キューから取り出す前の状態に戻します。 4. SERREVT を起動します（send 関数で設定した場合）。 5. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 6. CERREVT（メッセージ送信中の HSC 手順エラー）を起動します。 7. DLE・EOT を送信します。 8. 回線の切断を通知するメッセージログ（KFCA14411-I）を出力します。 	<p>障害の要因を取り除いたあと、運用コマンド（mcfactle）を入力して、論理端末の閉塞を解除します。</p>
メッセージ受信時の HSC 手順エラー	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. 受信メッセージの破棄を通知するメッセージログ（KFCA14401-W）を出力します。 3. CERREVT（メッセージ受信時の HSC 手順エラー）を起動します。 4. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 5. DLE・EOT を送信します。 6. 回線の切断を通知するメッセージログ（KFCA14411-I）を出力します。 	<p>障害の要因を取り除いたあと、運用コマンド（mcfactle）を入力して、論理端末の閉塞を解除します。</p>

10. HSC2 手順 (非同期モード) の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
回線接続維持中の HSC 手順エラー	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 3. CERREVT (HSC 手順エラー) を起動します。 4. DLE・EOT を送信します。 5. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	<p>障害の要因を取り除いたあと、運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除します。</p>
回線接続処理中 HSC 手順エラー	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. DLE・EOT を送信します。 	<p>ありません。</p>
発信失敗	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. 発信処理を繰り返します。 発信の再試行をすることを指定している場合、リトライオーバーまたは発信リトライの定義がないときは、次の処理をします。 3. 発信の失敗を通知するメッセージログ (KFCA14432-W) を出力します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (発信失敗) を起動します。 再試行間隔時間を経過しても回線に空きがない場合、すぐに再試行できないときがあります。 	<p>障害の要因を取り除いたあと、運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除します。</p>
メッセージ受信時の受信バッファオーバーフロー	<ol style="list-style-type: none"> 1. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します。 2. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. 全回線をクローズします。 5. 回線のクローズを通知するメッセージログ (KFCA14504-I) を出力します。 6. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 7. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 8. CERREVT (受信バッファオーバーフロー) を起動します (メッセージ受信時)。 	<p>構成定義を見直してください。</p>

障害の内容	TP1/NET/HSC の処理	ユーザの処理
着信時に該当するターミナル ID が ない場合	<ol style="list-style-type: none"> 1. ターミナル ID 未定義を通知するメッセージログ (KFCA14421-W) を出力します。 2. DLE・EOT を送信します。 	構成定義を見直してください。
発信時の相手システム接続拒否	<ol style="list-style-type: none"> 1. 相手システム接続拒否を通知するメッセージログ (KFCA14409-W) を出力します。 2. メッセージの送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 3. 送信メッセージを出力キューから取り出す前の状態に戻します。 4. SERREVT を起動します (send 関数で設定した場合)。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. CERREVT (発信時の接続拒否) を起動します。 	障害の要因を取り除いたあと、運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除します。

(2) HSC2 手順（非同期モード）プロトコル障害

表 10-2 HSC2 手順（非同期モード）プロトコル障害発生時の処理

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ送信時の DLE・EOT 受信	<ol style="list-style-type: none"> 1. メッセージ送受信中の DLE・EOT 受信を通知するメッセージログ (KFCA14412-W) を出力します。 2. メッセージの送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 3. 送信メッセージを出力キューから取り出す前の状態に戻します。 4. SERREVT を起動します (send 関数で設定した場合)。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. CERREVT (メッセージ送受信時の DLE・EOT 受信) を起動します。 7. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	障害の要因を取り除いたあと、運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除し、発信要求からします。
メッセージ受信時 (ETB ブロック受信後) の DLE・EOT 受信	<ol style="list-style-type: none"> 1. メッセージ送受信中の DLE・EOT 受信を通知するメッセージログ (KFCA14412-W) を出力します。 2. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (メッセージ受信時の DLE・EOT 受信) を起動します。 5. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	障害の要因を取り除いたあと、運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除し、発信要求からします。

10. HSC2 手順（非同期モード）の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ送信中の EOT 受信	<ol style="list-style-type: none"> 1. メッセージ送受信中の EOT 受信を通知するメッセージログ (KFCA14422-W) を出力します。 2. メッセージの送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 3. 送信メッセージを出力キューから取り出す前の状態に戻します。 4. SERREVT を起動します (send 関数で設定した場合)。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. CERREVT (メッセージ送信中の EOT 受信) を起動します。 7. DLE・EOT を送信します。 8. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	<p>障害の要因を取り除いたあと、運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除し、発信要求からします。</p>
メッセージ受信中 (ETB ブロック受信後) の EOT 受信	<ol style="list-style-type: none"> 1. メッセージ送受信中の EOT 受信を通知するメッセージログ (KFCA14422-W) を出力します。 2. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (メッセージ受信中の EOT 受信) を起動します。 5. DLE・EOT を送信します。 6. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	<p>障害の要因を取り除いたあと、運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除します。</p>
ETX ブロック送受信後の DLE・EOT 受信	<ol style="list-style-type: none"> 1. DLE・EOT 受信を通知するメッセージログ (KFCA14432-W) を出力します。 2. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	<p>ありません。</p>
回線接続時の受信バッファ不足	<ol style="list-style-type: none"> 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. グループバッファ取得失敗を通知するメッセージログ (KFCA14005-E) を出力します。 3. 全回線をクローズします。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 6. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 7. CERREVT (受信バッファ不足) を起動します。 	<p>構成定義を見直してください。</p>

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ受信時の受信バッファ不足	<ol style="list-style-type: none"> 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. グループバッファ取得失敗を通知するメッセージログ (KFCA14005-E) を出力します。 3. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します。 4. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. 全回線をクローズします。 7. 回線のクローズ (全回線数分) を通知するメッセージログ (KFCA14504-I) を出力します。 8. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 9. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 10. CERREVT (受信バッファ不足) を起動します。 	構成定義を見直してください。
送信テキスト形態不正	<ol style="list-style-type: none"> 1. 送信テキスト形態不正を通知するメッセージログ (KFCA14400-W) を出力します。 2. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. 送信メッセージを出力キューから消去します。 4. SERREVT を起動します (send 関数で設定した場合)。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. CERREVT (送信テキスト形態不正) を起動します。 7. DLE・EOT を送信します。 8. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します (回線切断中の場合)。 	UAP を見直してください。
送信最大セグメント長オーバー	<ol style="list-style-type: none"> 1. 送信最大セグメント長オーバーを通知するメッセージログ (KFCA14418-W) を出力します。 2. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. 送信メッセージを出力キューから消去します。 4. SERREVT を起動します (send 関数で設定した場合)。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. CERREVT (送信最大セグメント長オーバー) を起動します。 7. DLE・EOT を送信します。 8. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します (回線切断中の場合)。 	UAP を見直してください。
自システムからの回線接続処理中の論理端末に着信	回線接続処理中の回線に DLE・EOT を送信します。	ありません。

10. HSC2 手順（非同期モード）の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
回線接続完了後の論理端末に着信	<ol style="list-style-type: none"> 1. 使用中の論理端末に着信を通知するメッセージログ（KFCA14421-W）を出力します。 2. 着信した回線に DLE・EOT を送信します。 	ありません。
論理端末明示閉塞中の着信	DLE・EOT を送信します。	障害の要因を取り除いたあと、運用コマンド（mcftactle）を入力して、論理端末の閉塞を解除します。
メッセージ送信時の運用コマンド（mcftdctle）による論理端末明示閉塞	<ol style="list-style-type: none"> 1. メッセージの送信中断を通知するメッセージログ（KFCA10608-W）を出力します。 2. 送信メッセージを出力キューから取り出す前の状態に戻します。 3. SERREVT を起動します（send 関数で設定した場合）。 4. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 5. CERREVT（メッセージ送信中の運用コマンド（mcftdctle）の入力）を起動します。 6. DLE・EOT を送信します。 7. 回線の切断を通知するメッセージログ（KFCA14411-I）を出力します。 	ありません。
メッセージ受信時または回線維持中の運用コマンド（mcftdctle）による論理端末明示閉塞	<ol style="list-style-type: none"> 1. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 2. 受信メッセージの破棄を通知するメッセージログ（KFCA14401-W）を出力します（メッセージ受信時）。 3. CERREVT（運用コマンド（mcftdctle）による論理端末明示閉塞）を起動します。 4. DLE・EOT を送信します。 5. 回線の切断を通知するメッセージログ（KFCA14411-I）を出力します。 	ありません。
相手ダイヤル番号未定義	<ol style="list-style-type: none"> 1. 相手ダイヤル番号未定義を通知するメッセージログ（KFCA14433-W）を出力します。 2. 送信メッセージの破棄を通知するメッセージログ（KFCA10607-W）を出力します。 3. SERREVT を起動します（send 関数で設定した場合）。 4. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 5. CERREVT（相手ダイヤル番号未定義）を起動します。 	構成定義を見直してください。

（3）受信スケジュール関係（入力キュー，入力メッセージ編集 UOC）の障害

入力キュー障害時の障害処理については、「10.5 入力キュー障害」を参照してください。

表 10-3 受信スケジュール関係の障害発生時の処理（HSC2 手順（非同期モード））

障害の内容	TP1/NET/HSC の処理	ユーザの処理
アプリケーション名未定義	<ol style="list-style-type: none"> ERREVT1 を起動します。 アプリケーション名未定義を通知するメッセージログ (KFCA10604-E) を出力します。 	アプリケーション定義を見直してください。
アプリケーション名形式不正	<ol style="list-style-type: none"> ERREVT1 を起動します。 アプリケーション名の形式不正を通知するメッセージログ (KFCA10610-E) を出力します。 	UOC を見直してください。
システムアプリケーション名未定義 <ul style="list-style-type: none"> ERREVTn SxxxEVT CxxxEVT 	メッセージ、および MCF イベントを破棄します。	ありません。
<ul style="list-style-type: none"> MHP サービス、サービスグループ閉塞 入力キュー書き込み障害 	<ol style="list-style-type: none"> ERREVT2 を起動します。 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。 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 DLE・EOT を送信します。 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	ありません。
メッセージ受信時の入力メッセージ編集 UOC エラーリターン	<ol style="list-style-type: none"> UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 CERREVT (UOC 障害) を起動します。 DLE・EOT を送信します。 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	ありません。
メッセージ受信時の入力メッセージ編集 UOC パラメータ不正	<ol style="list-style-type: none"> UOC パラメータ不正を通知するメッセージログ (KFCA10620-E) を出力します。 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 CERREVT (UOC 障害) を起動します。 DLE・EOT を送信します。 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UOC を見直してください。
論理端末名称決定 UOC エラーリターン	<ol style="list-style-type: none"> 論理端末名称決定 UOC エラーリターンを通知するメッセージログ (KFCA14504-I) を出力します。 DLE・EOT を送信します。 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	論理端末名称決定 UOC がエラーリターンした理由を調査してください。
論理端末名称決定 UOC でリターンした論理端末名称未定義	<ol style="list-style-type: none"> 論理端末名称未定義を通知するメッセージログ (KFCA14429-E) を出力します。 DLE・EOT を送信します。 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UOC を見直してください。

10. HSC2 手順（非同期モード）の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
論理端末名称決定 UOC でリターンした論理端末名称形式不正	<ol style="list-style-type: none"> 1. 論理端末名称形式不正を通知するメッセージログ (KFCA14431-E) を出力します。 2. DLE・EOT を送信します。 3. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UOC を見直してください。
論理端末名称決定 UOC で決定した論理端末名称が使用中	<ol style="list-style-type: none"> 1. 論理端末使用中を通知するメッセージログ (KFCA14431-E) を出力します。 2. DLE・EOT を送信します。 	ありません。

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

出力キュー障害時の障害処理については、「10.6 出力キュー障害」を参照してください。

表 10-4 送信スケジュール関係の障害発生時の処理（HSC2 手順（非同期モード））

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ送信時の出力キュー読み込み障害	<ol style="list-style-type: none"> 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. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. メッセージを出力キューから消去します。 4. SERREVT を起動します（send 関数で設定した場合）。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. CERREVT（出力キュー読み込み障害）を起動します。 7. DLE・EOT を送信します。 8. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	ありません。
出力キュー消去障害	<ol style="list-style-type: none"> 1. 出力キュー障害を通知するメッセージログ (KFCA10617-E) を出力します。 2. 処理を続行します。 	ありません。

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ送信時の送信バッファオーバーフロー	<ol style="list-style-type: none"> 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. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. メッセージを出力キューから消去します。 4. SERREVT を起動します (send 関数で設定した場合)。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. CERREVT (受信バッファオーバーフロー) を起動します。 7. DLE・EOT を送信します。 8. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	ありません。
メッセージ送信時の送信バッファ不足	<ol style="list-style-type: none"> 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. メッセージ送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 4. 送信メッセージを出力キューから取り出す前の状態に戻します。 5. SERREVT を起動します (send 関数で設定した場合)。 6. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 7. CERREVT (受信バッファ不足) を起動します。 8. DLE・EOT を送信します。 9. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	構成定義を見直してください。

10. HSC2 手順（非同期モード）の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ送信時の出力メッセージ編集 UOC パラメータ不正	<ol style="list-style-type: none"> 1. UOC パラメータ不正を通知するメッセージログ (KFCA10620-E) を出力します。 2. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. メッセージを出力キューから消去します。 4. SERREVT を起動します (send 関数で設定した場合)。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. CERREVT (論理端末障害) を起動します。 7. DLE・EOT を送信します。 8. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UOC を見直してください。
メッセージ送信時の出力メッセージ編集 UOC エラーリターン	<ol style="list-style-type: none"> 1. UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 2. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. メッセージを出力キューから消去します。 4. SERREVT を起動します (send 関数で設定した場合)。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. CERREVT (UOC 障害) を起動します。 7. DLE・EOT を送信します。 8. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UOC を見直してください。

(5) UAP 障害

表 10-5 UAP 障害発生時の処理 (HSC2 手順 (非同期モード))

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

(6) MCF の障害

表 10-6 MCF の障害発生時の処理（HSC2 手順（非同期モード））

障害の内容	TP1/NET/HSC の処理	ユーザの処理
<ul style="list-style-type: none"> • 内部論理矛盾 • 他 PP とのインタフェースエラー 	<ol style="list-style-type: none"> 1. 障害情報の取得を通知するメッセージログ (KFCA10189-E) を出力します。 2. 全回線をクローズします。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 5. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 6. CERREVT (内部論理矛盾) を起動します。 	ありません。

10.2 コネクション障害

回線に回復できない障害が発生すると、TP1/NET/HSC は、ほかの使用可能な回線を利用して処理を続行します。ほかに使用可能な回線がない場合、コネクション障害となります。

障害の発生した回線は、自動的に回復しません。障害の要因を取り除いたあと、運用コマンド（mcftactcn）を入力してコネクションの確立を再試行してください。

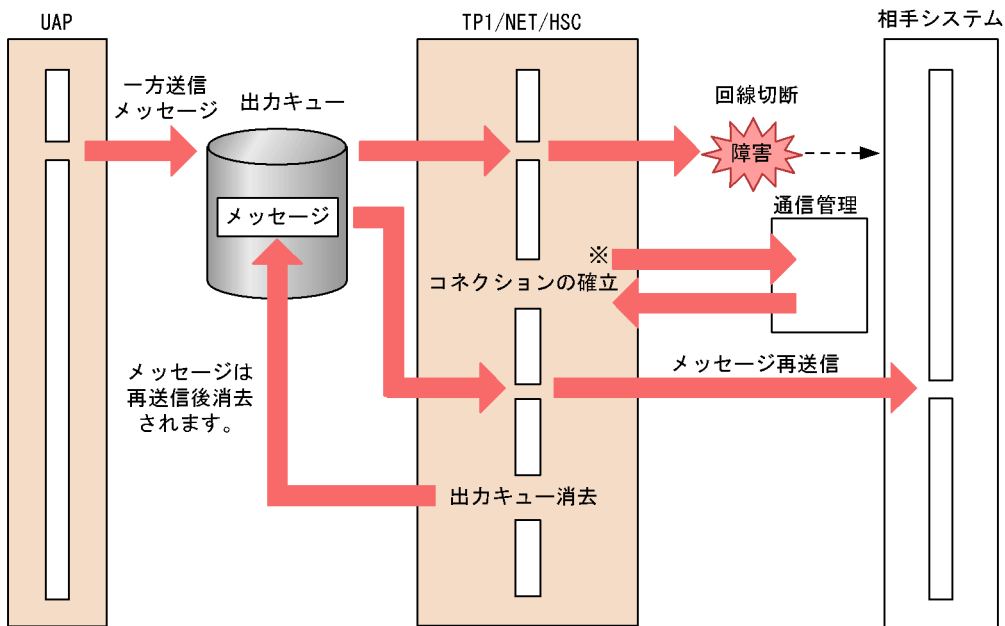
10.2.1 メッセージ送信時のコネクション障害

（1）メッセージ送信前のコネクション障害

メッセージはロールバックされ、別回線を使用してまたはコネクションの確立をして再送信されます。

メッセージ送信前のコネクション障害の例を次の図に示します。

図 10-1 メッセージ送信前のコネクション障害（HSC2 手順（非同期モード））



注

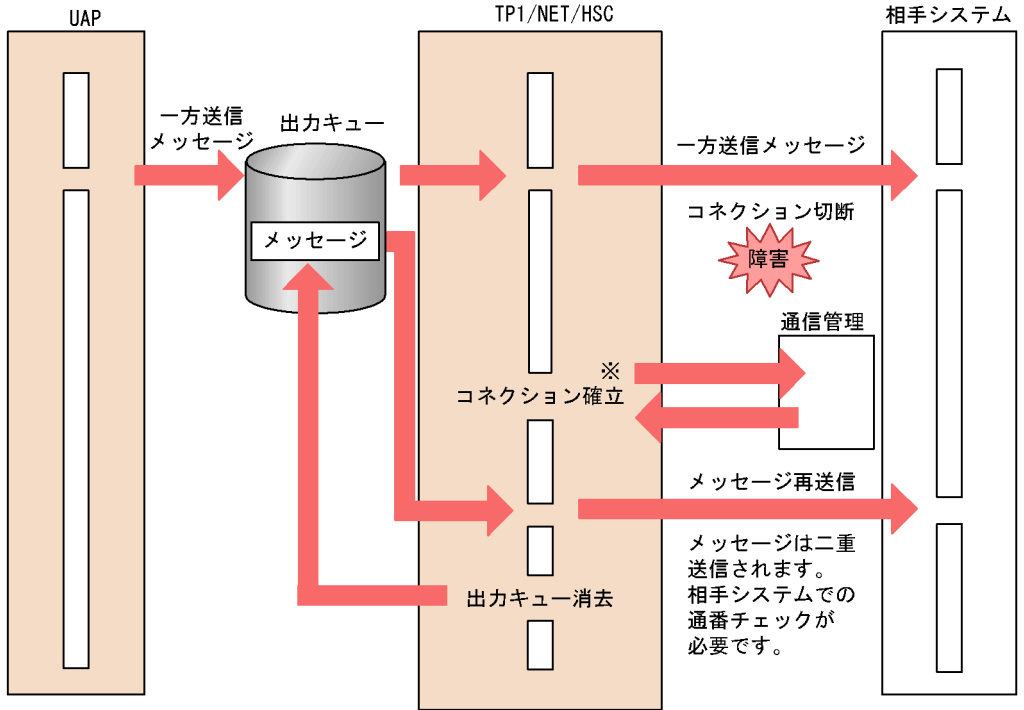
別の回線が使用できる場合、この処理は飛ばされます。

（2）出力キュー消去前のコネクション障害

メッセージは二重に送信されます。このとき相手システムの UAP は、メッセージに通番を付けることで、二重送信をチェックします。

メッセージ消去前のコネクション障害の例を次の図に示します。

図 10-2 メッセージ消去前のコネクション障害（HSC2 手順（非同期モード））



注

別の回線が使用できる場合、この処理は飛ばされます。

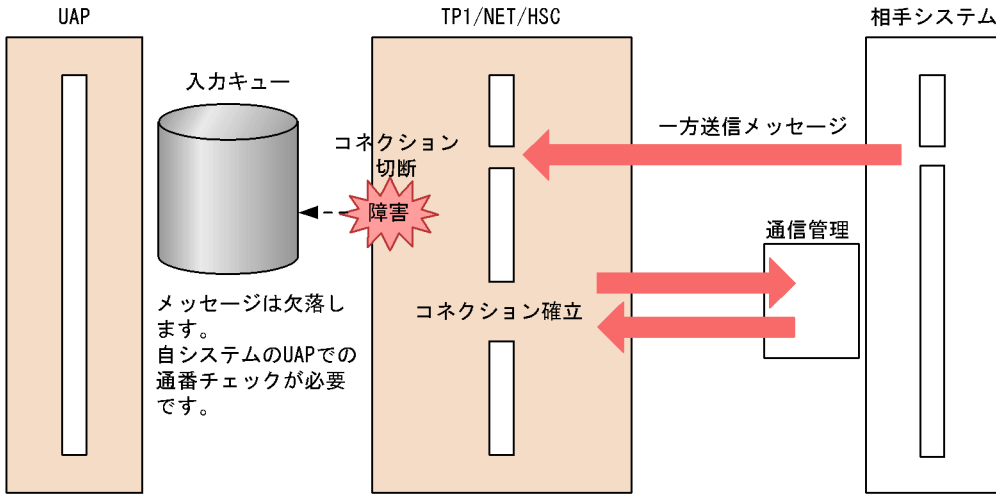
10.2.2 メッセージ受信時のコネクション障害

(1) 入力キュー書き込み前のコネクション障害

メッセージは欠落します。このとき自システムの UAP は、メッセージに通番を付けることで、欠落しているかどうかをチェックします。

メッセージ書き込み前のコネクション障害の例を次の図に示します。

図 10-3 メッセージ書き込み前のコネクション障害（HSC2 手順（非同期モード））

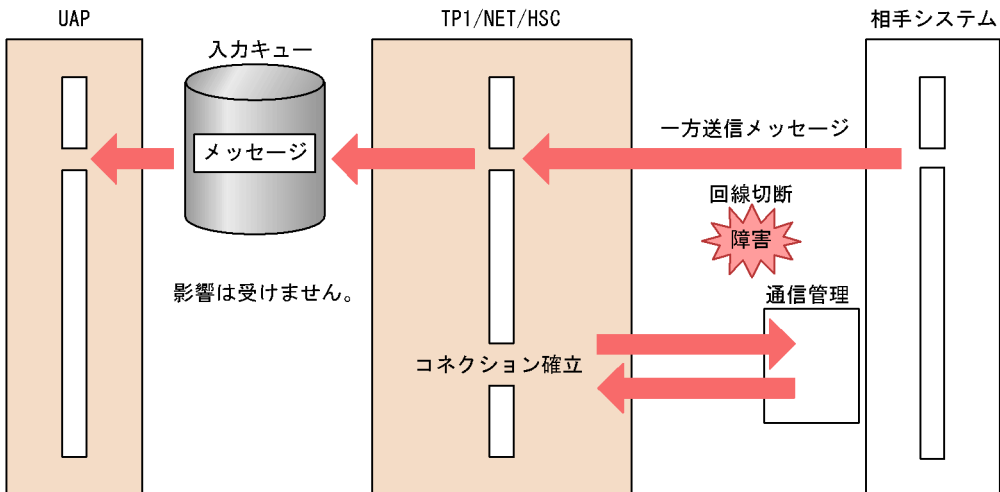


(2) 入力キュー書き込み後のコネクション障害

メッセージは影響を受けません。

メッセージ書き込み後のコネクション障害の例を次の図に示します。

図 10-4 メッセージ書き込み後のコネクション障害（HSC2 手順（非同期モード））



10.3 ユーザアプリケーションプログラム異常終了

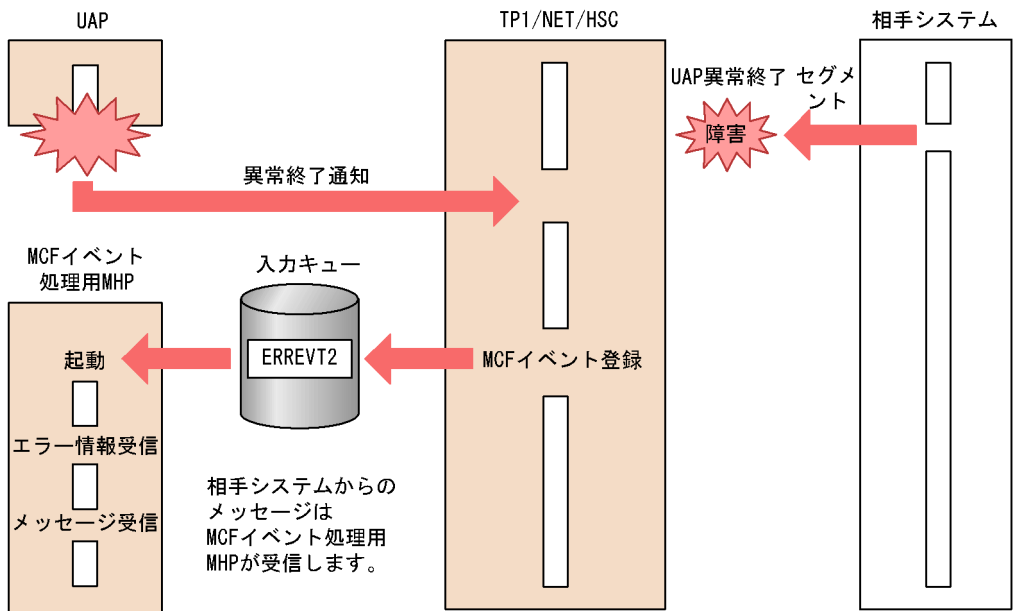
UAP が異常終了した場合、TP1/NET/HSC は入力キューに MCF イベントを登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

10.3.1 セグメント受信前の UAP 異常終了

TP1/NET/HSC は、入力キューに ERREVT2（メッセージ破棄通知イベント）を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

セグメント受信前の UAP 異常終了の例を次の図に示します。

図 10-5 セグメント受信前の UAP 異常終了（HSC2 手順（非同期モード））



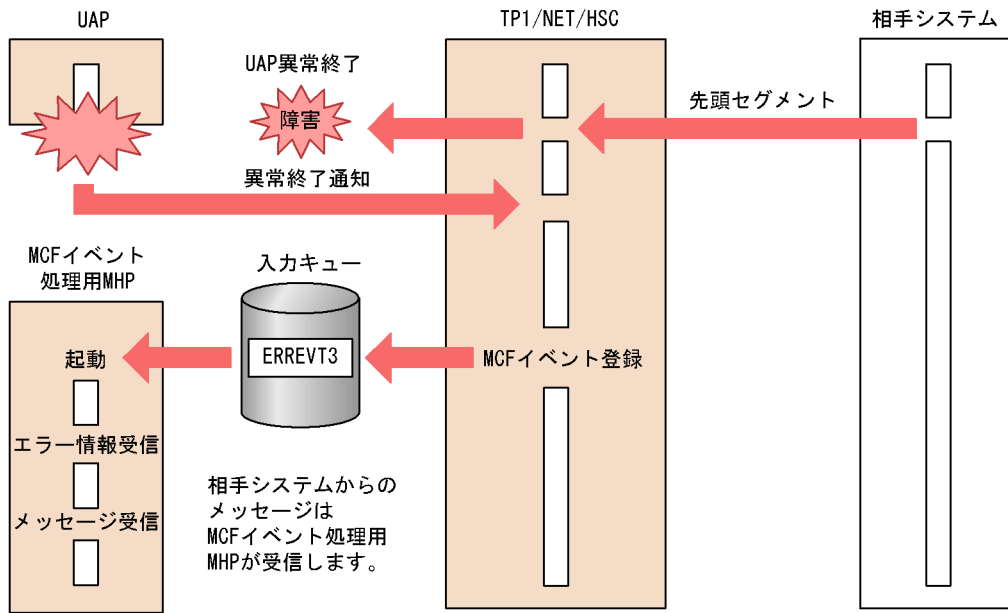
10.3.2 セグメント受信後の UAP 異常終了

TP1/NET/HSC は、入力キューに ERREVT3（UAP 異常終了通知イベント）を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

セグメント受信後の UAP 異常終了の例を次の図に示します。

10. HSC2 手順（非同期モード）の障害対策

図 10-6 セグメント受信後の UAP 異常終了（HSC2 手順（非同期モード））

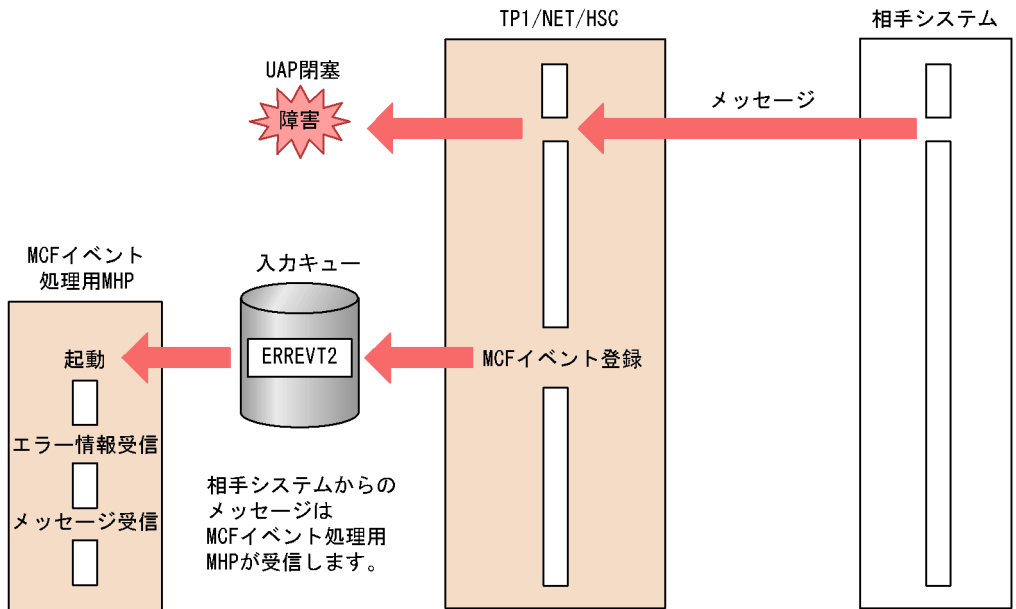


10.4 ユーザアプリケーションプログラム閉塞

UAP が閉塞して使用できない場合、TP1/NET/HSC は入力キューに ERREVT2（メッセージ破棄通知イベント）を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

UAP 閉塞による障害の例を次の図に示します。

図 10-7 UAP 閉塞による障害（HSC2 手順（非同期モード））

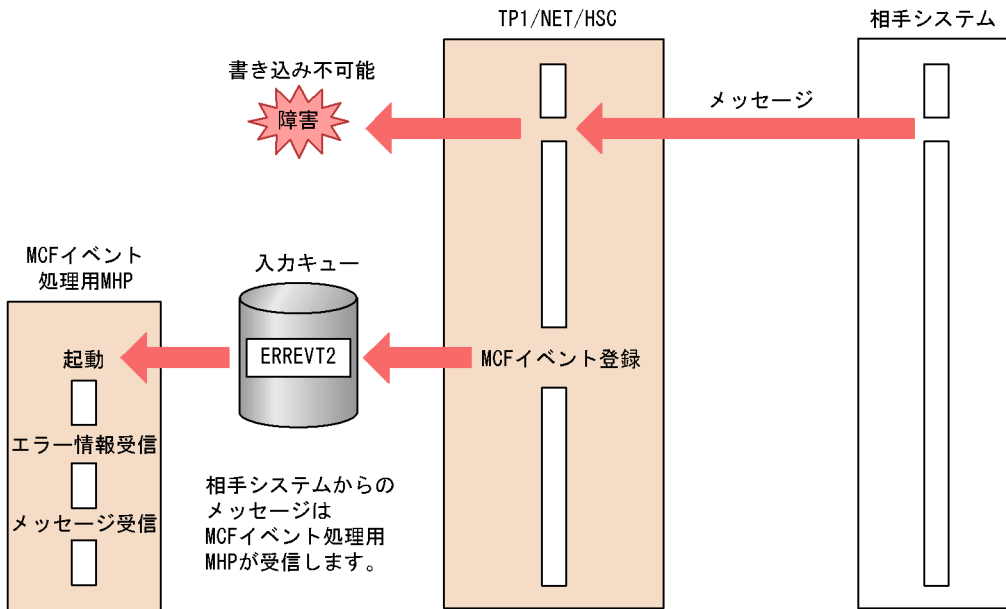


10.5 入力キュー障害

メッセージを入力キューに書き込むときに障害が発生した場合、TP1/NET/HSC は入力キューに ERREVT2（メッセージ破棄通知イベント）を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

入力キュー障害の例を次の図に示します。

図 10-8 入力キュー障害（HSC2 手順（非同期モード））

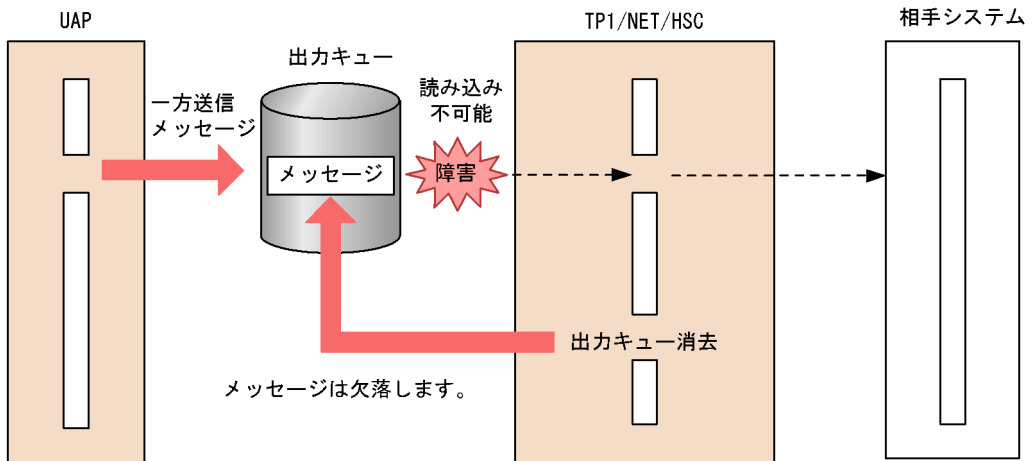


10.6 出力キュー障害

メッセージを出力キューから読み込むときに障害が発生した場合、メッセージは欠落します。

出力キュー障害の例を次の図に示します。

図 10-9 出力キュー障害（HSC2 手順（非同期モード））



11 HSC2 手順（同期モード）の 障害対策

この章では、TP1/NET/HSC の HSC2 手順（同期モード）の運用中に発生する障害と、HSC2 手順（同期モード）の対応処理、およびメッセージの処理について説明します。

11.1 障害の種類と対応処理

11.2 コネクション障害

11.3 ユーザアプリケーションプログラム異常終了

11.4 ユーザアプリケーションプログラム閉塞

11.5 HSC 手順障害

11.1 障害の種類と対応処理

HSC2 手順（同期モード）の場合の TP1/NET/HSC 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）通信障害

コネクション障害時の TP1/NET/HSC の障害処理については、「11.2 コネクション障害」を参照してください。

表 11-1 通信障害発生時の処理（HSC2 手順（同期モード））

障害の内容	TP1/NET/HSC の処理	ユーザの処理
コネクション確立時障害（回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. 発信、着信用相手ターミナル ID リストを破棄します。 3. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 4. コネクションの閉塞を通知するメッセージログ（KFCA14002-E）を出力します。 5. CERREVT（コネクションの確立不可能）を起動します。 	障害の要因を取り除いたあと、運用コマンド（mctactcn）を入力して、再びコネクションを確立します。
発信、着信要求時障害（回復不可能）	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ（KFCA14500-W）を出力します。 2. メッセージの送信中断を通知するメッセージログ（KFCA10608-W）を出力します。 3. 発信要求を出力キューから取り出す前の状態に戻します（発信要求時）。 4. 発信、着信用相手ターミナル ID リストを破棄します。 5. 論理端末の閉塞を通知するメッセージログ（KFCA14004-I）を出力します。 6. コネクションの解放を通知するメッセージログ（KFCA14001-I）を出力します。 7. コネクションの閉塞を通知するメッセージログ（KFCA14002-E）を出力します。 8. CERREVT（コネクションの切断）を起動します。 	障害の要因を取り除いたあと、運用コマンド（mctactcn）を入力して、再びコネクションを確立します。

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ送受信時障害 (回線回復不可能)	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. 受信テキストの破棄を通知するメッセージログ (KFCA14401-W) を出力します (メッセージ受信時)。 3. 同期送受信要求の異常完了 (コネクション障害)。 4. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 5. 発信, 着信用相手ターミナル ID リストを破棄します。 6. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 7. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 8. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 9. CERREVT (コネクションの切断) を起動します。 	<p>障害の要因を取り除いたあと, 運用コマンド (mftactcn) を入力して, 論理端末の閉塞を解除します。</p>
発信要求時の HSC 手順エラー	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. メッセージの送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 3. 発信要求を出力キューから取り出す前の状態に戻します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (HSC 手順エラー) を起動します。 	<p>障害の要因を取り除いたあと, 運用コマンド (mftactcl) を入力して, 論理端末の閉塞を解除し, 発信要求をします。または, 着信待ちとします。</p>
着信要求時の HSC 手順エラー	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. 着信時の障害を通知するメッセージログ (KFCA14417-W) を出力します。 	<p>障害の要因を取り除いたあと, 運用コマンド (mftactcl) を入力して, 論理端末の閉塞を解除し, 発信要求をします。または, 着信待ちとします。</p>
メッセージ送受信時 HSC 手順エラー	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します (メッセージ受信時)。 3. 同期送受信要求の異常完了 (論理端末障害)。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (HSC 手順エラー) を起動します。 6. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	<p>障害の要因を取り除いたあと, 運用コマンド (mftactcl) を入力して, 論理端末の閉塞を解除し, 発信要求をします。または, 着信待ちとします。</p>

11. HSC2 手順 (同期モード) の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
発信失敗	<ol style="list-style-type: none"> 1. コネクションの障害を通知するメッセージログ (KFCA14500-W) を出力します。 2. メッセージの送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 3. 発信要求を出力キューから取り出す前の状態に戻します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (発信失敗) を起動します。 	<p>一定時間経過後、運用コマンド (mcfactle) を入力して、論理端末の閉塞を解除し、発信要求をします。または、着信待ちとします。</p>
発信要求時の受信バッファオーバーフロー	<ol style="list-style-type: none"> 1. 受信バッファオーバーフロー (ID 交換時) を通知するメッセージログ (KFCA14407-W) を出力します。 2. メッセージの送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 3. 発信要求を出力キューから取り出す前の状態に戻します (発信要求時)。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (受信バッファオーバーフロー (ID 交換時)) を起動します。 	<p>ありません。</p>
着信要求時の受信バッファオーバーフロー	<ol style="list-style-type: none"> 1. 受信バッファオーバーフロー (ID 交換時) を通知するメッセージログ (KFCA14407-W) を出力します。 2. 着信時の障害を通知するメッセージログ (KFCA14417-W) を出力します。 	<p>ありません。</p>
メッセージ受信時の受信バッファオーバーフロー	<ol style="list-style-type: none"> 1. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します。 2. 同期受信要求の異常完了 (コネクション障害)。 3. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 4. 発信、着信用相手ターミナル ID リストを破棄します。 5. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 6. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 7. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 8. CERREVT (受信バッファオーバーフロー (メッセージ受信時)) を起動します。 	<p>構成定義を見直してください。</p>

(2) HSC2 手順 (同期モード) プロトコル障害

表 11-2 HSC2 手順 (同期モード) プロトコル障害発生時の処理

障害の内容	TP1/NET/HSC の処理	ユーザの処理
発信要求時に該当する相手ターミナル ID がない場合	<ol style="list-style-type: none"> 1. 相手ターミナル ID なし (発信要求時) を通知するメッセージログ (KFCA14402-W) を出力します。 2. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. 発信要求を出力キューから消去します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (相手ターミナル ID なし (発信要求時)) を起動します。 	発信用相手ターミナル ID リスト, または UAP を見直してください。
着信要求時に該当する相手ターミナル ID がない場合	相手ターミナル ID なし (着信要求時) を通知するメッセージログ (KFCA14404-W) を出力します。	着信用相手ターミナル ID リストを見直してください。
発信要求時の相手ターミナル ID チェック不正	<ol style="list-style-type: none"> 1. 相手ターミナル ID チェック不正を通知するメッセージログ (KFCA14403-W) を出力します。 2. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. 発信要求を出力キューから消去します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (相手ターミナル ID チェック不正) を起動します。 	発信用相手ターミナル ID リストを見直してください。
発信要求時の接続拒否	<ol style="list-style-type: none"> 1. 相手システムの接続拒否を通知するメッセージログ (KFCA14409-W) を出力します。 2. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. 発信要求を出力キューから消去します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (発信要求時の接続拒否) を起動します。 	障害の要因を取り除いたあと, 運用コマンド (mcfactle) を入力して, 論理端末の閉塞を解除し, 発信要求をします。または, 着信待ちとします。
発信要求時の ID・ENQ 受信	<ol style="list-style-type: none"> 1. 発信要求時の ID・ENQ 受信を通知するメッセージログ (KFCA14414-W) を出力します。 2. メッセージの送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 3. 発信要求を出力キューから取り出す前の状態に戻します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (発信要求時の ID・ENQ 受信) を起動します。 	障害の要因を取り除いたあと, 運用コマンド (mcfactle) を入力して, 論理端末の閉塞を解除し, 発信要求をします。または, 着信待ちとします。

11. HSC2 手順 (同期モード) の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ送信時の DLE・EOT 受信	<ol style="list-style-type: none"> 1. メッセージ送受信中の DLE・EOT 受信を通知するメッセージログ (KFCA14412-W) を出力します。 2. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します (メッセージ受信時)。 3. 同期送受信要求の異常完了 (論理端末障害)。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (メッセージ送受信時の DLE・EOT 受信) を起動します。 6. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	<p>障害の要因を取り除いたあと、運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除し、発信要求をします。または、着信待ちとします。</p>
発信, 着信時の受信バッファ不足	<ol style="list-style-type: none"> 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. グループバッファ取得失敗を通知するメッセージログ (KFCA14005-E) を出力します。 3. メッセージの送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 4. 発信要求を出力キューから取り出す前の状態に戻します (発信要求時)。 5. 発信, 着信用相手ターミナル ID リストを破棄します。 6. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 7. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 8. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 9. CERREVT (受信バッファ不足) を起動します。 	<p>構成定義を見直してください。</p>
メッセージ受信時の受信バッファ不足	<ol style="list-style-type: none"> 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. グループバッファ取得失敗を通知するメッセージログ (KFCA14005-E) を出力します。 3. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します。 4. 同期受信要求の異常完了 (メモリ不足)。 5. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 6. 発信, 着信用相手ターミナル ID リストを破棄します。 7. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 8. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 9. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 10. CERREVT (受信バッファ不足) を起動します。 	<p>構成定義を見直してください。</p>

障害の内容	TP1/NET/HSC の処理	ユーザの処理
着信要求時の論理 端末未定義	<ol style="list-style-type: none"> 1. 論理端末未定義を通知するメッセージログ (KFCA14416-E) を出力します。 2. 発信, 着信用相手ターミナル ID リストを破棄します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. コネクションの解放を通知するメッセージログ (KFCA14001-I) を出力します。 5. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 6. CERREVT (論理端末未定義) を起動します。 	UAP を見直してください。
発信要求時の送信 テキスト形態不正	<ol style="list-style-type: none"> 1. 送信テキスト形態不正を通知するメッセージログ (KFCA14400-W) を出力します。 2. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. 送信要求を出力キューから消去します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (送信テキスト形態不正) を起動します。 	UAP を見直してください。
メッセージ送信時 の送信テキスト形 態不正	<ol style="list-style-type: none"> 1. 送信テキスト形態不正を通知するメッセージログ (KFCA14400-W) を出力します。 2. 同期送信要求の異常完了 (論理端末障害)。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (送信テキスト形態不正) を起動します。 5. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UAP を見直してください。
送信最大セグメン ト長オーバー	<ol style="list-style-type: none"> 1. 送信最大セグメント長オーバーを通知するメッセージログ (KFCA14418-W) を出力します。 2. 同期送信要求の異常完了 (論理端末障害)。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (送信最大セグメント長オーバー) を起動します。 5. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UAP を見直してください。
発信要求時のダイ ヤル番号長不正	<ol style="list-style-type: none"> 1. ダイヤル番号長不正を通知するメッセージログ (KFCA14425-W) を出力します。 2. 送信メッセージの破棄を通知するメッセージログ (KFCA10607-W) を出力します。 3. 送信要求を出力キューから消去します。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (ダイヤル番号長不正) を起動します。 	UAP を見直してください。

11. HSC2 手順（同期モード）の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
分岐種別不正	<ol style="list-style-type: none"> 1. HSC2 制御未サポート機能要求を通知するメッセージログ (KFCA14415-W) を出力します。 2. 発信要求を出力キューから消去します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (分岐種別不正) を起動します。 	UAP を見直してください。
シーケンス不正	<ol style="list-style-type: none"> 1. シーケンス不正を通知するメッセージログ (KFCA14408-W) を出力します。 2. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します (メッセージ受信時)。 3. 同期送受信要求の異常完了 (論理端末名称)。 4. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 5. CERREVT (シーケンス不正) を起動します。 6. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UAP を見直してください。
発信要求時の運用コマンド (mcftdctle) による論理端末明示閉塞	<ol style="list-style-type: none"> 1. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 2. メッセージの送信中断を通知するメッセージログ (KFCA10608-W) を出力します。 3. 発信要求を出力キューから取り出す前の状態に戻します。 4. CERREVT (運用コマンド (mcftdctle) による論理端末明示閉塞) を起動します。 	ありません。
メッセージ送受信時の運用コマンド (mcftdctle) による論理端末明示閉塞	<ol style="list-style-type: none"> 1. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 2. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します (メッセージ受信時)。 3. 同期送受信要求の異常完了 (論理端末名称)。 4. CERREVT (運用コマンド (mcftdctle) による論理端末明示閉塞) を起動します。 5. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	ありません。
発信, 着信用相手ターミナル ID リスト読み込み障害	<ol style="list-style-type: none"> 1. 発信, 着信用相手ターミナル ID リスト読み込み障害を通知するメッセージログ (KFCA14406-W) を出力します。 2. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 3. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 4. CERREVT (発信, 着信用相手ターミナル ID リスト読み込み障害) を起動します。 	構成定義を見直してください。

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

表 11-3 受信スケジュール関係の障害発生時の処理 (HSC2 手順 (同期モード))

障害の内容	TP1/NET/HSC の処理	ユーザの処理
アプリケーション名未定義	<ol style="list-style-type: none"> 1. ERREVT1 を起動します。 2. アプリケーション名未定義を通知するメッセージログ (KFCA10604-E) を出力します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (アプリケーション起動不可) を起動します。 	アプリケーション定義を見直してください。
アプリケーション名形式不正	<ol style="list-style-type: none"> 1. ERREVT1 を起動します。 2. アプリケーション名の形式不正を通知するメッセージログ (KFCA10610-E) を出力します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (アプリケーション起動不可) を起動します。 	UOC を見直してください。
MCF イベントの名称未定義 <ul style="list-style-type: none"> • ERREVTx • SxxxEVT • CxxxEVT 	メッセージ, および MCF イベントを破棄します。	ありません。
<ul style="list-style-type: none"> • MHP サービス, サービスグループ閉塞 • 入力キュー書き込み障害 	<ol style="list-style-type: none"> 1. ERREVT2 を起動します。 2. 入力キュー障害を通知するメッセージログ (KFCA10604-E) を出力します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (アプリケーション起動不可) を起動します。 	ありません。
回線接続時の入力メッセージ編集 UOC エラーリターン	<ol style="list-style-type: none"> 1. UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 2. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 3. CERREVT (UOC 障害) を起動します。 	ありません。
メッセージ受信時の入力メッセージ編集 UOC エラーリターン	<ol style="list-style-type: none"> 1. UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 2. 同期受信要求の異常完了 (UOC 障害)。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (UOC 障害) を起動します。 5. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	ありません。

11. HSC2 手順（同期モード）の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
回線接続時の入力メッセージ編集 UOC パラメタ不正	<ol style="list-style-type: none"> 1. UOC パラメタ不正を通知するメッセージログ (KFCA10620-E) を出力します。 2. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 3. CERREVT (UOC 障害) を起動します。 	UOC を見直してください。
メッセージ受信時の入力メッセージ編集 UOC パラメタ不正	<ol style="list-style-type: none"> 1. UOC パラメタ不正を通知するメッセージログ (KFCA10620-E) を出力します。 2. 同期受信要求の異常完了 (UOC 障害)。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (UOC 障害) を起動します。 5. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UOC を見直してください。

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

表 11-4 送信スケジュール関係の障害発生時の処理（HSC2 手順（同期モード））

障害の内容	TP1/NET/HSC の処理	ユーザの処理
発信要求時の出力キュー読み込み障害	<ol style="list-style-type: none"> 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. 発信要求を出力キューから消去します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (出力キュー読み込み障害) を起動します。 	ありません。
メッセージ送信時の出力キュー読み込み障害	<ol style="list-style-type: none"> 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. 同期送信要求の異常完了 (メッセージ読み込み障害)。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (出力キュー読み込み障害) を起動します。 5. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	ありません。
出力キュー消し込み障害	<ol style="list-style-type: none"> 1. 出力キュー障害を通知するメッセージログ (KFCA10617-E) を出力します。 2. 処理を続行します。 	ありません。
発信要求時の送信バッファオーバーフロー	<ol style="list-style-type: none"> 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. 発信要求を出力キューから消去します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (送信バッファオーバーフロー) を起動します。 	バッファグループ定義または UAP を見直してください。

障害の内容	TP1/NET/HSC の処理	ユーザの処理
メッセージ送信時の送信バッファオーバーフロー	<ol style="list-style-type: none"> 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 同期送信要求の異常完了 (メッセージ読み込み障害) 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 CERREVT (送信バッファオーバーフロー) を起動します。 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	<p>バッファグループ定義または UAP を見直してください。</p>
発信要求時の送信バッファ不足	<ol style="list-style-type: none"> ローカルメモリ不足を通知するメッセージログ (KFCA10618-E) を出力します。 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 発信要求を出力キューから取り出す前の状態に戻します。 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 CERREVT (受信バッファ不足) を起動します。 	<ul style="list-style-type: none"> 一定時間経過後、運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除し、発信要求をします。または、着信待ちとします。 構成定義を見直してください。
メッセージ送信時の送信バッファ不足	<ol style="list-style-type: none"> ローカルメモリ不足を通知するメッセージログ (KFCA10618-E) を出力します。 出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 同期送信要求の異常完了 (メモリ不足) 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 CERREVT (送信バッファ不足) を起動します。 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	<ul style="list-style-type: none"> 一定時間経過後、運用コマンド (mcftactle) を入力して、論理端末の閉塞を解除し、発信要求をします。または、着信待ちとします。 構成定義を見直してください。
発信要求時の出力メッセージ編集 UOC パラメタ不正	<ol style="list-style-type: none"> UOC パラメタ不正を通知するメッセージログ (KFCA10620-E) を出力します。 発信要求を出力キューから消去します。 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 CERREVT (UOC 障害) を起動します。 	<p>UOC を見直してください。</p>
メッセージ送信時の出力メッセージ編集 UOC パラメタ不正	<ol style="list-style-type: none"> UOC パラメタ不正を通知するメッセージログ (KFCA10620-E) を出力します。 同期送信要求の異常完了 (UOC 障害) 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 CERREVT (UOC 障害) を起動します。 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	<p>UOC を見直してください。</p>

11. HSC2 手順（同期モード）の障害対策

障害の内容	TP1/NET/HSC の処理	ユーザの処理
発信要求時の出力メッセージ編集 UOC エラーリターン	<ol style="list-style-type: none"> 1. UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 2. 発信要求を出力キューから消去します。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (UOC 障害) を起動します。 	ありません。
メッセージ送信時の出力メッセージ編集 UOC エラーリターン	<ol style="list-style-type: none"> 1. UOC エラーリターンを通知するメッセージログ (KFCA10611-E) を出力します。 2. 同期送信要求の異常完了 (UOC 障害)。 3. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 4. CERREVT (UOC 障害) を起動します。 5. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	ありません。

(5) UAP 障害

表 11-5 UAP 障害発生時の処理 (HSC2 手順 (同期モード))

障害の内容	TP1/NET/HSC の処理	ユーザの処理
回線接続通知前の UAP 異常終了	ERREVT2 を起動します。	UAP を見直してください。
回線接続通知後の UAP 異常終了	<ol style="list-style-type: none"> 1. ERREVT3 を起動します。 2. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UAP を見直してください。
メッセージ送受信時の UAP 異常終了	<ol style="list-style-type: none"> 1. ERREVT3 を起動します。 2. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 3. メッセージ送受信時の UAP プロセス終了を通知するメッセージログ (KFCA14413-W) を出力します。 4. 受信メッセージの破棄を通知するメッセージログ (KFCA14401-W) を出力します (メッセージ受信時)。 5. CERREVT (メッセージ送受信時の UAP プロセス終了) を起動します。 6. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	UAP を見直したあと、運用コマンド (meftdctle) を入力してください。

(6) PRC 障害, sendsync 関数・recvsync 関数のリターン障害

表 11-6 PRC 障害, sendsync 関数・recvsync 関数のリターン障害時の処理 (HSC2 手順 (同期モード))

障害の内容	TP1/NET/HSC の処理	ユーザの処理
同期要求完了障害	<ol style="list-style-type: none"> 1. 同期応答障害を通知するメッセージログ (KFCA10606-E) を出力します。 2. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 3. CERREVT (同期要求完了エラーリターン) を起動します。 4. 回線の切断を通知するメッセージログ (KFCA14411-I) を出力します。 	ありません。

(7) MCF の障害

表 11-7 MCF の障害発生時の処理 (HSC2 手順 (同期モード))

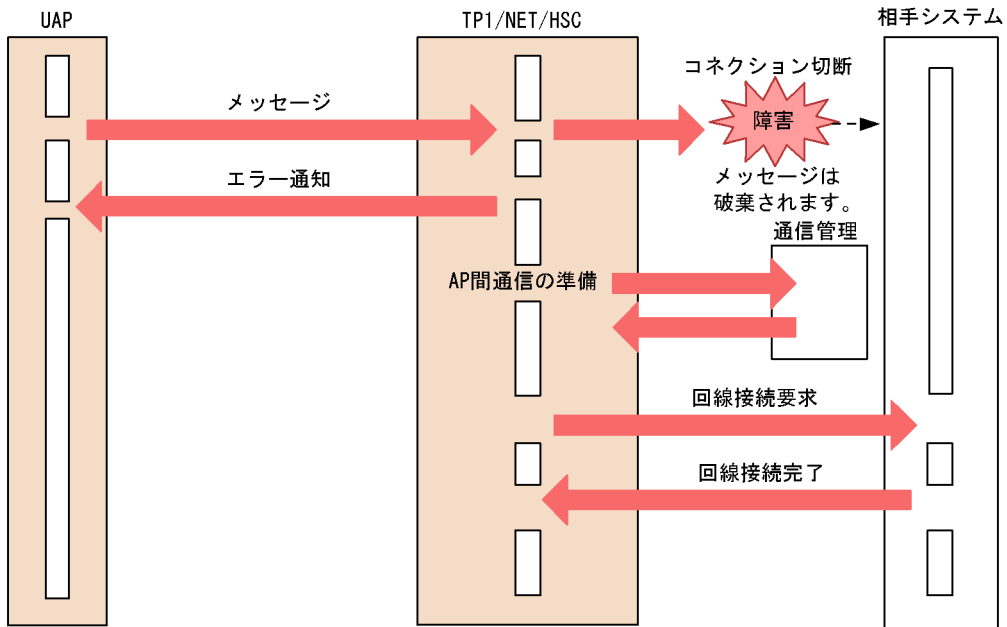
障害の内容	TP1/NET/HSC の処理	ユーザの処理
<ul style="list-style-type: none"> • 内部論理矛盾 • 他 PP とのインタフェースエラー 	<ol style="list-style-type: none"> 1. 障害情報の取得を通知するメッセージログ (KFCA10189-E) を出力します。 2. 論理端末の閉塞を通知するメッセージログ (KFCA14004-I) を出力します。 3. コネクションの解放を通知するメッセージログ (KFCA14001-E) を出力します。 4. コネクションの閉塞を通知するメッセージログ (KFCA14002-E) を出力します。 5. CERREVT (内部論理矛盾) を起動します。 	ありません。

11.2 コネクション障害

コネクションが切断された場合、TP1/NET/HSC は自動的に回復しません。通信管理が管理している通信機器に障害が発生した場合も同様です。障害の要因を取り除いたあと、運用コマンド (mcftactcn) を入力してコネクションを再確立してください。そのあと、発信または着信時の方法で回線を接続します。

メッセージ送受信時のコネクション障害の例を次の図に示します。

図 11-1 メッセージ送受信時のコネクション障害 (HSC2 手順 (同期モード))



11.3 ユーザアプリケーションプログラム異常終了

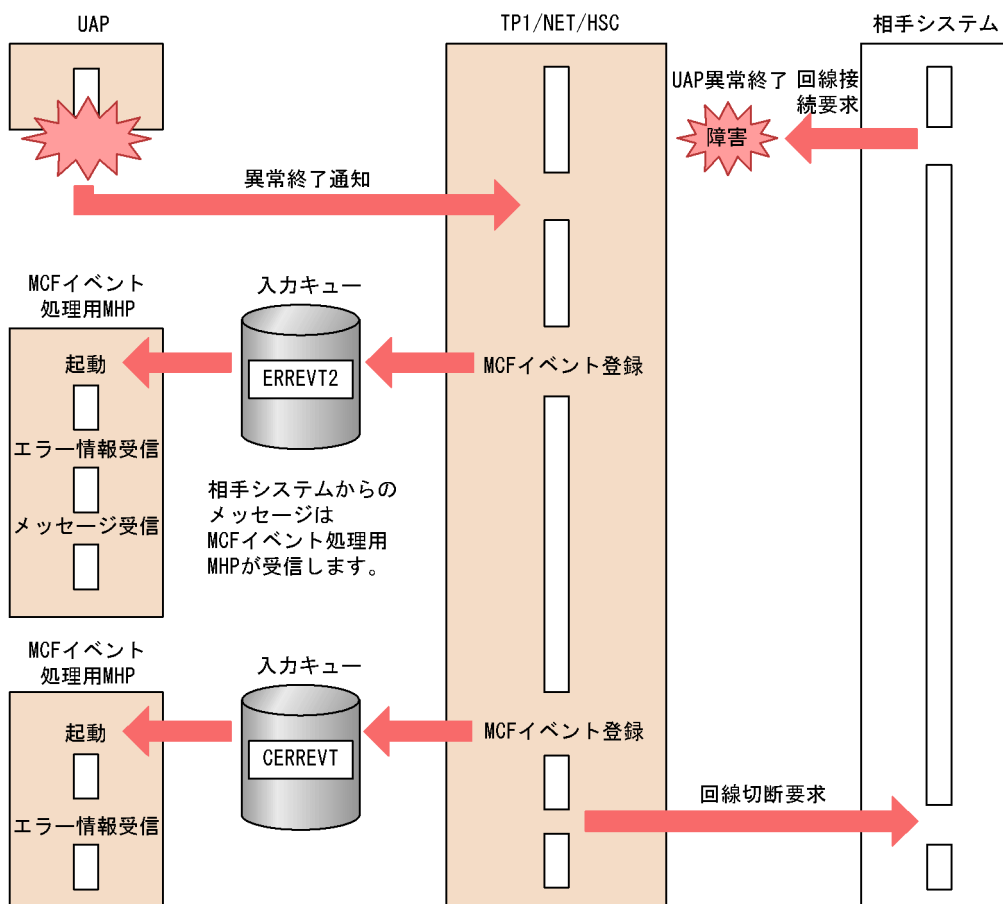
UAP が異常終了した場合、TP1/NET/HSC は入力キューに MCF イベントを登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

11.3.1 回線接続要求受信前の UAP 異常終了

TP1/NET/HSC は、入力キューに ERREVT2（メッセージ破棄通知イベント）を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

回線接続要求受信前の UAP 異常終了の例を次の図に示します。

図 11-2 回線接続要求受信前の UAP 異常終了（HSC2 手順（同期モード））

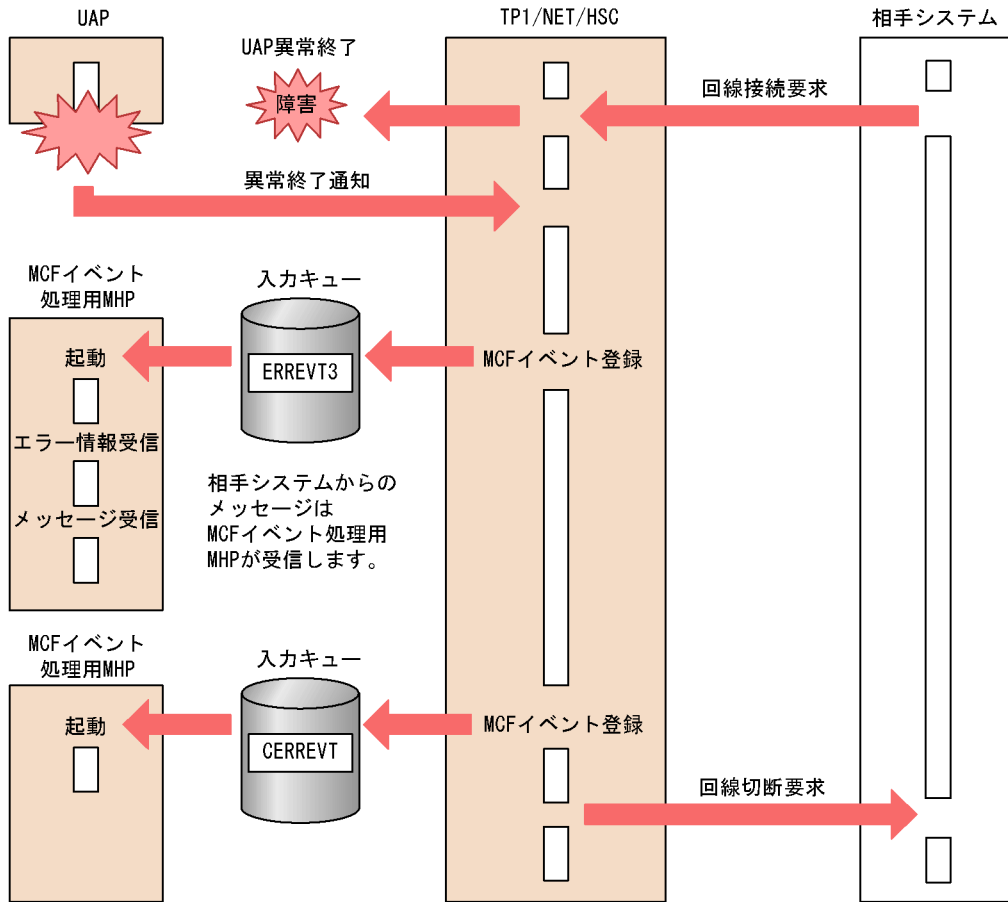


11.3.2 回線接続要求受信後の UAP 異常終了

TP1/NET/HSC は、入力キューに ERREVT3（UAP 異常終了通知イベント）を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

回線接続要求受信後の UAP 異常終了の例を次の図に示します。

図 11-3 回線接続要求受信後の UAP 異常終了（HSC2 手順（同期モード））

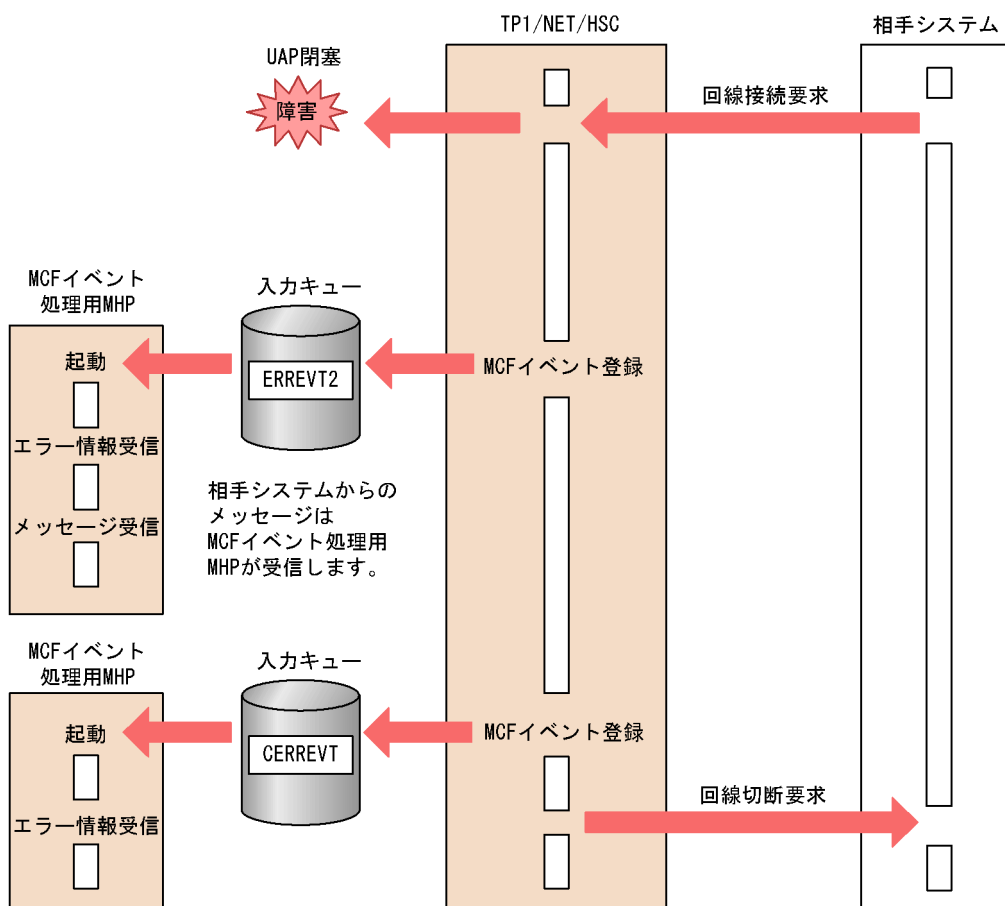


11.4 ユーザアプリケーションプログラム閉塞

UAP が閉塞して使用できない場合、TP1/NET/HSC は入力キューに ERREVT2 (メッセージ破棄通知イベント) を登録し、MCF イベント処理用 MHP を起動します。起動された MCF イベント処理用 MHP は、エラー情報と入力されたメッセージを受信します。

UAP 閉塞による障害の例を次の図に示します。

図 11-4 UAP 閉塞による障害 (HSC2 手順 (同期モード))

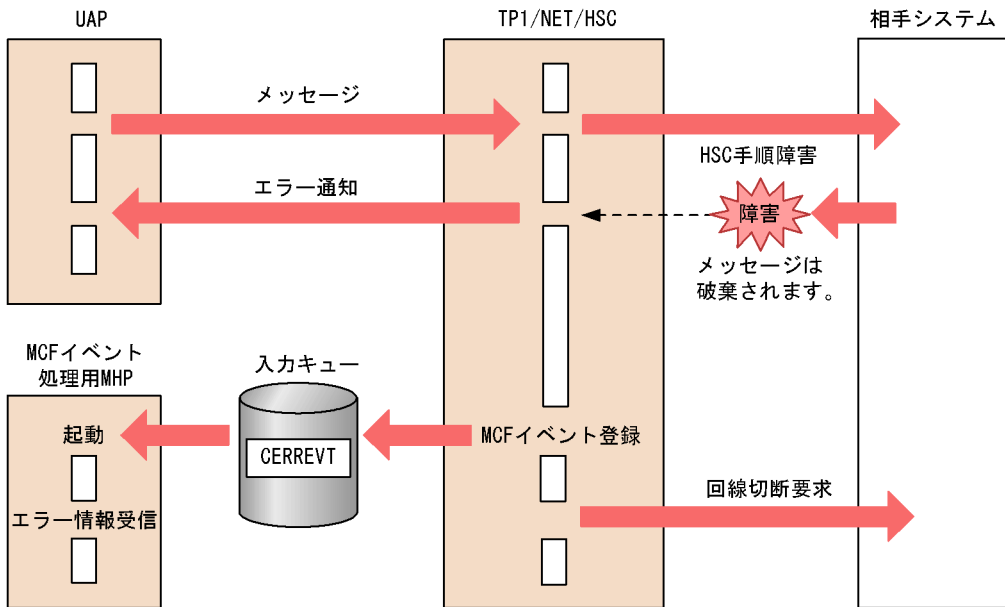


11.5 HSC 手順障害

HSC 手順障害が発生した場合、TP1/NET/HSC は UAP にエラーを通知したあと、入力キューに CERREVT (障害通知イベント) を登録し、MCF イベント処理用 MHP を起動します。メッセージは、コネクション障害と同様に破棄されます。

メッセージ送信時の HSC 手順障害の例を次の図に示します。

図 11-5 メッセージ送信時の HSC 手順障害 (HSC2 手順 (同期モード))



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 関数使用の制限 (HSC2 手順 (同期モード))

付録 E 論理端末名称決定 UOC のコーディング例

付録 F 用語解説

付録 A メッセージ送受信の処理の流れ

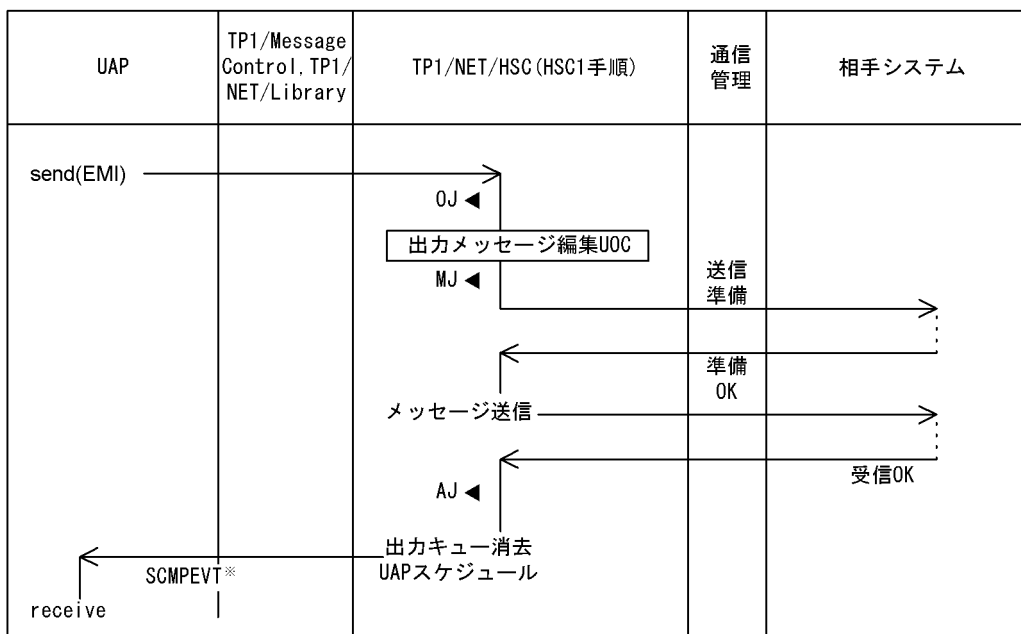
メッセージを送受信するときのデータの流れ、ジャーナルの取得タイミングなどを、HSC1 手順、HSC2 手順（非同期モード）および HSC2 手順（同期モード）に分けて説明します。

付録 A.1 HSC1 手順のメッセージの送受信

HSC1 手順のメッセージを送受信するときの処理の流れを、図 A-1 ~ 図 A-6 に示します。

(1) 単一セグメントの送信

図 A-1 HSC1 手順の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 ◀: メッセージ出力ジャーナル取得

MJ ◀: メッセージジャーナル取得

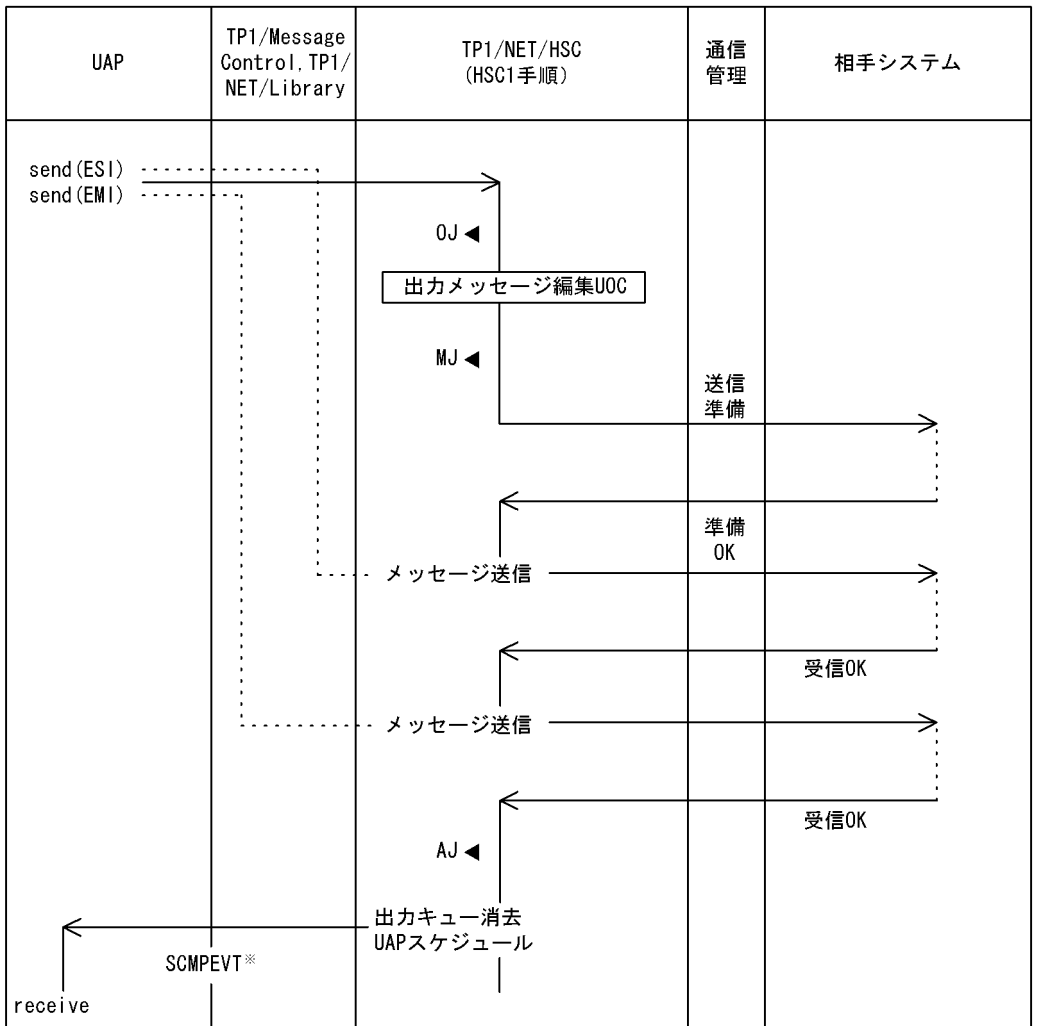
AJ ◀: メッセージ送信完了ジャーナル取得

注

send 関数で送信完了通知イベントを通知させることを指定した場合

(2) 複数セグメントの送信

図 A-2 HSC1 手順の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)

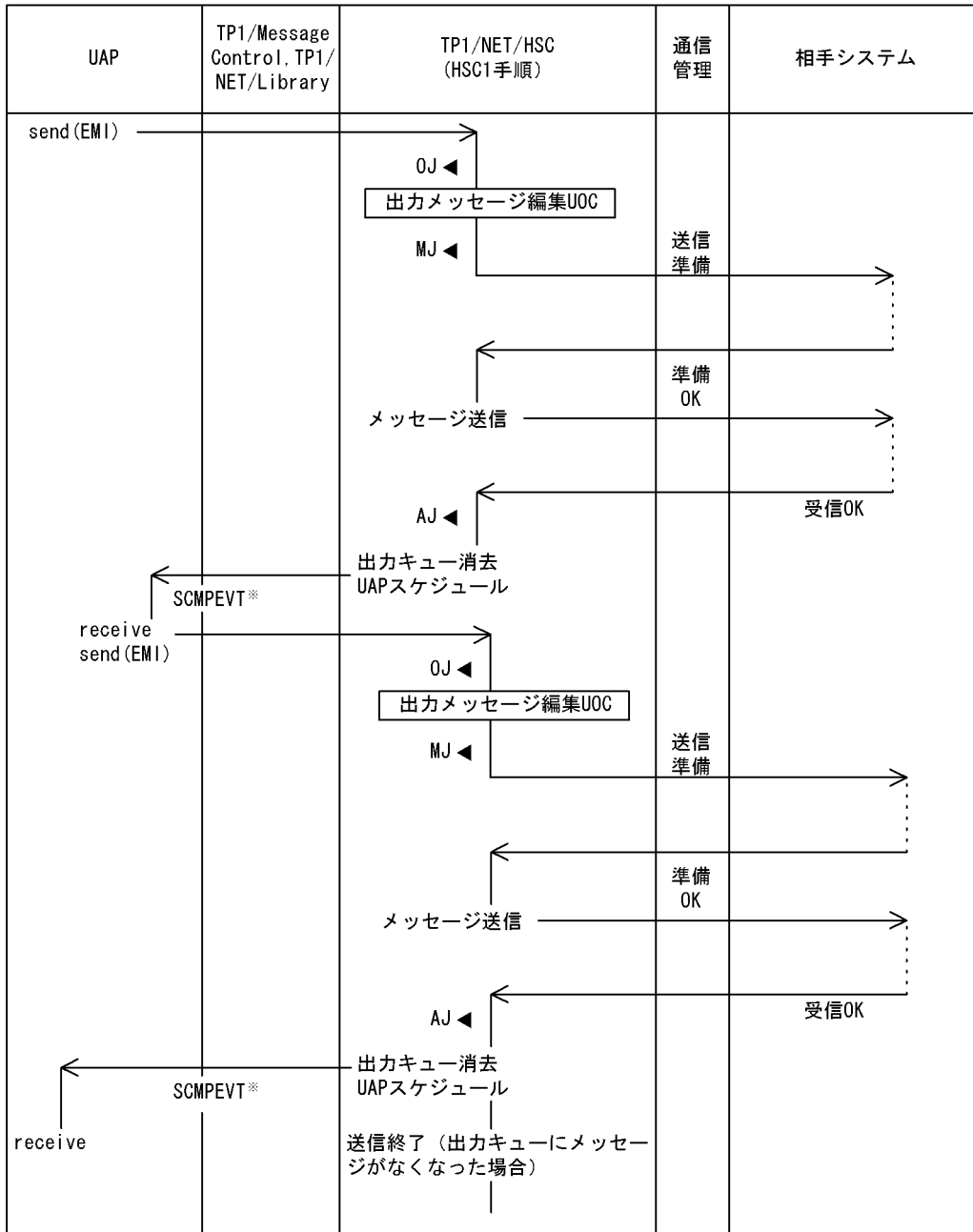
- 0J ◀: メッセージ出力ジャーナル取得
- MJ ◀: メッセージジャーナル取得
- AJ ◀: メッセージ送信完了ジャーナル取得

注

send 関数で送信完了通知イベントを通知させることを指定した場合

(3) 列信の場合の送信

図 A-3 HSC1 手順のメッセージ送信の処理の流れ (列信の場合)



(凡例)

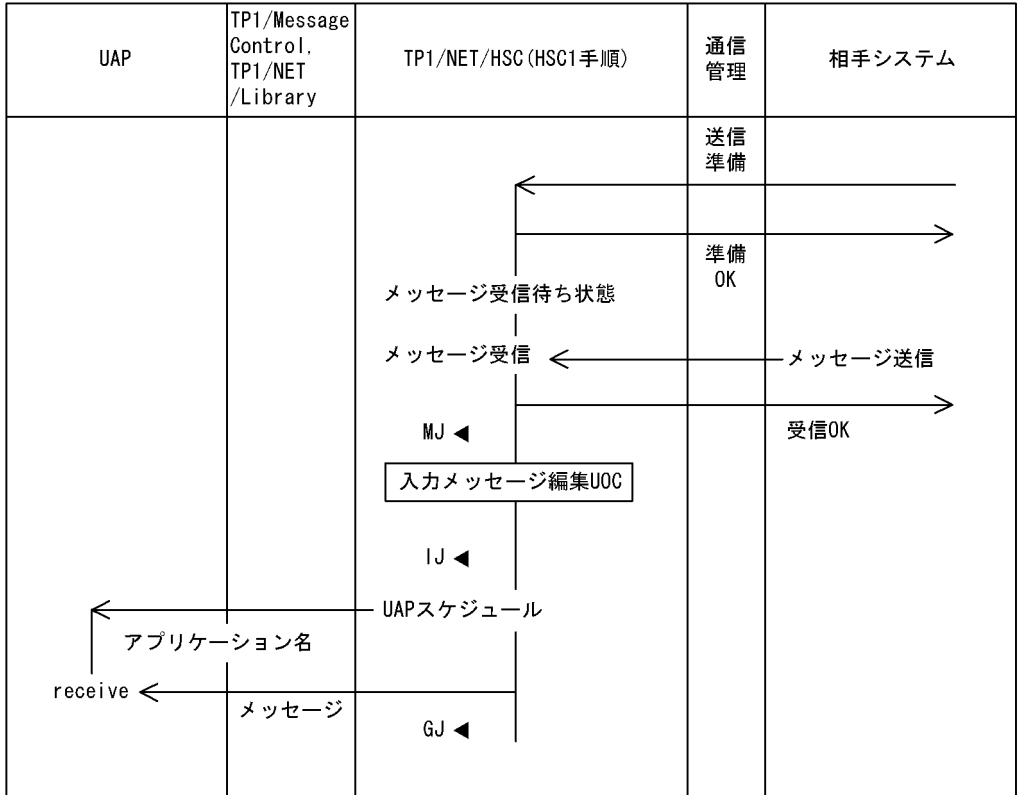
- OJ ◀: メッセージ出力ジャーナル取得
- MJ ◀: メッセージジャーナル取得
- AJ ◀: メッセージ送信完了ジャーナル取得

注

send 関数で送信完了通知イベントを通知させることを指定した場合

(4) 単一セグメントの受信

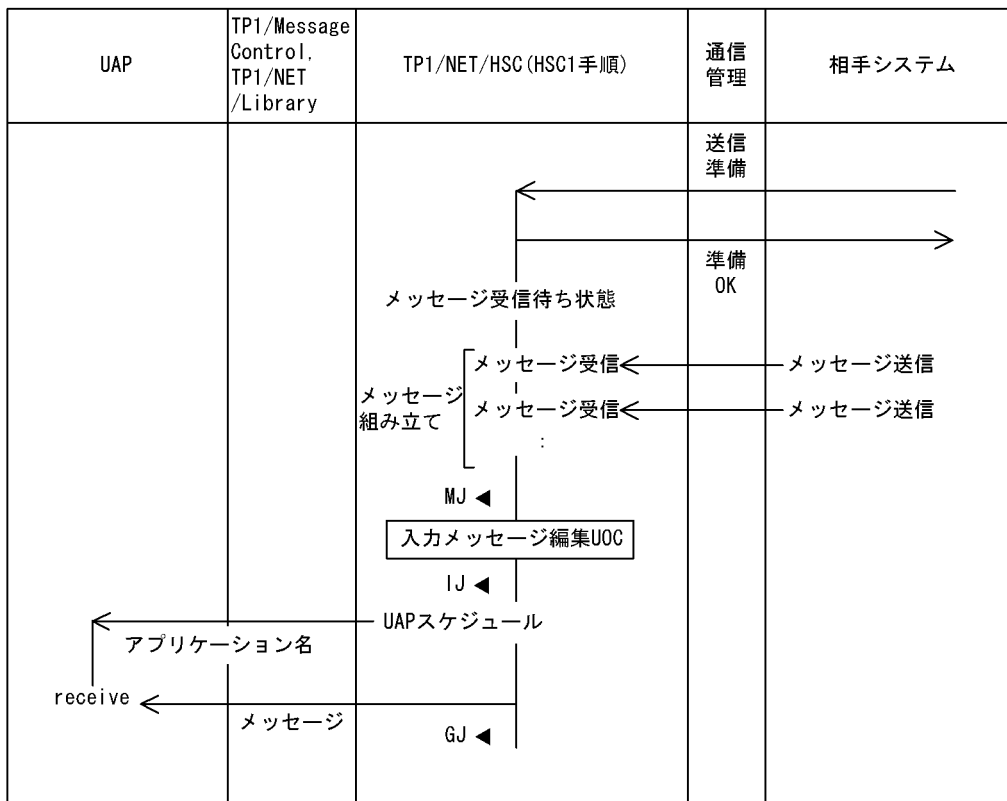
図 A-4 HSC1 手順のメッセージ受信の処理の流れ (単一セグメントの場合)



- (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

(5) 複数セグメントの受信

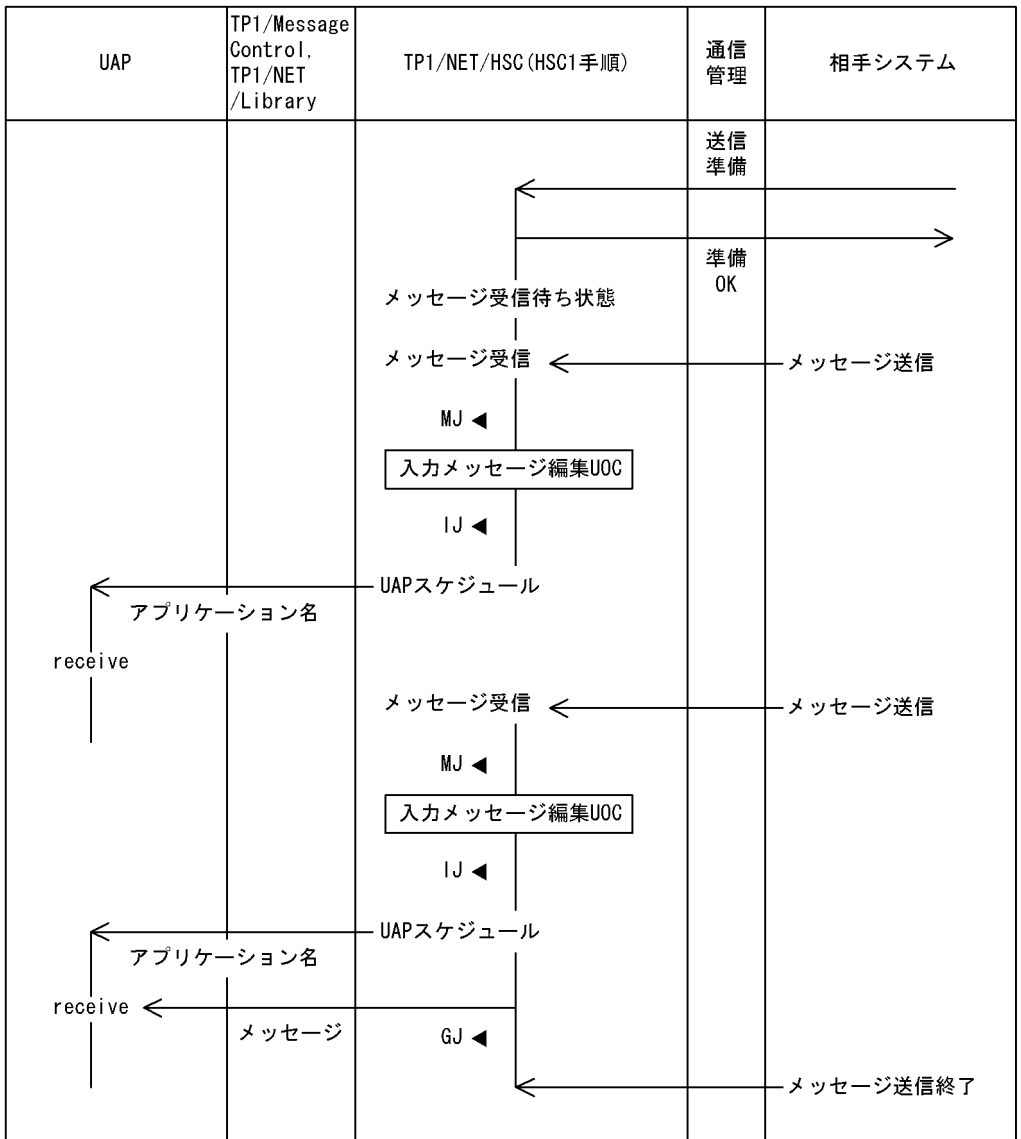
図 A-5 HSC1 手順のメッセージ受信の処理の流れ (複数セグメントの場合)



- (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

(6) 列信の場合の受信

図 A-6 HSC1 手順のメッセージ受信の処理の流れ (列信の場合)



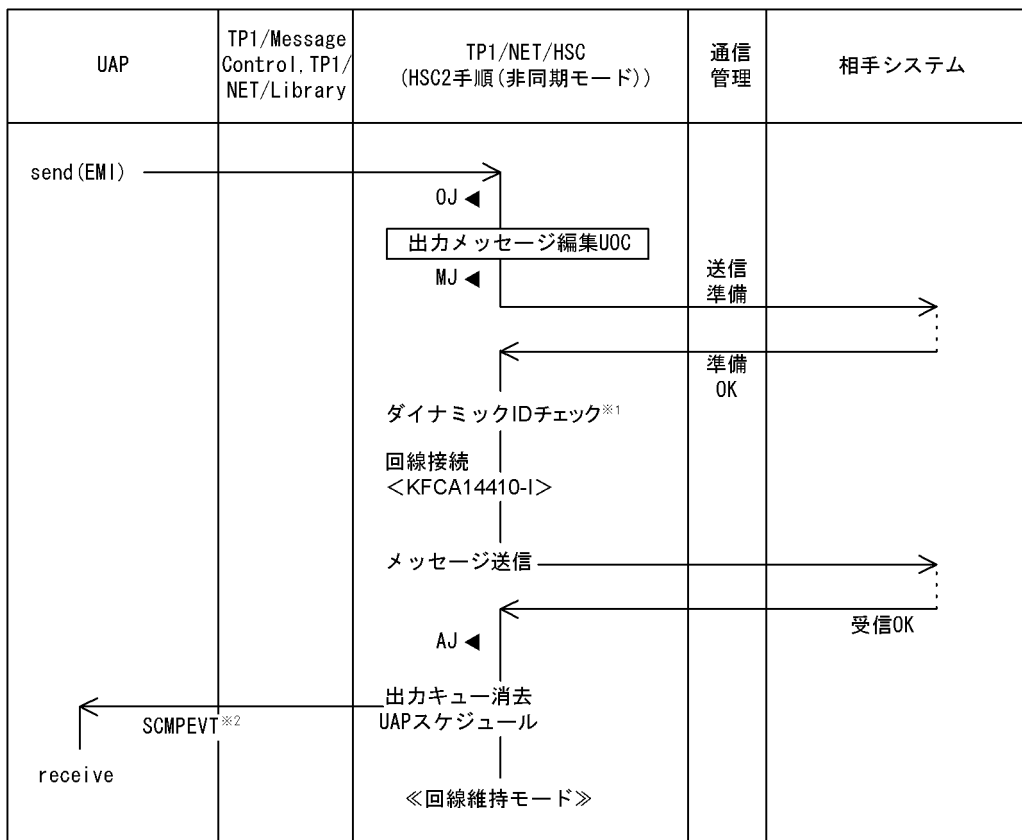
- (凡例) MJ ◀: メッセージジャーナル取得
 IJ ◀: メッセージ入力ジャーナル取得
 GJ ◀: メッセージ受信ジャーナル取得

付録 A.2 HSC2 手順 (非同期モード) のメッセージの送受信

HSC2 手順 (非同期モード) のメッセージを送受信するときの処理の流れを、図 A-7 ~ 図 A-14 に示します。

(1) 単一セグメントの送信

図 A-7 HSC2 手順 (非同期モード) の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

注 1

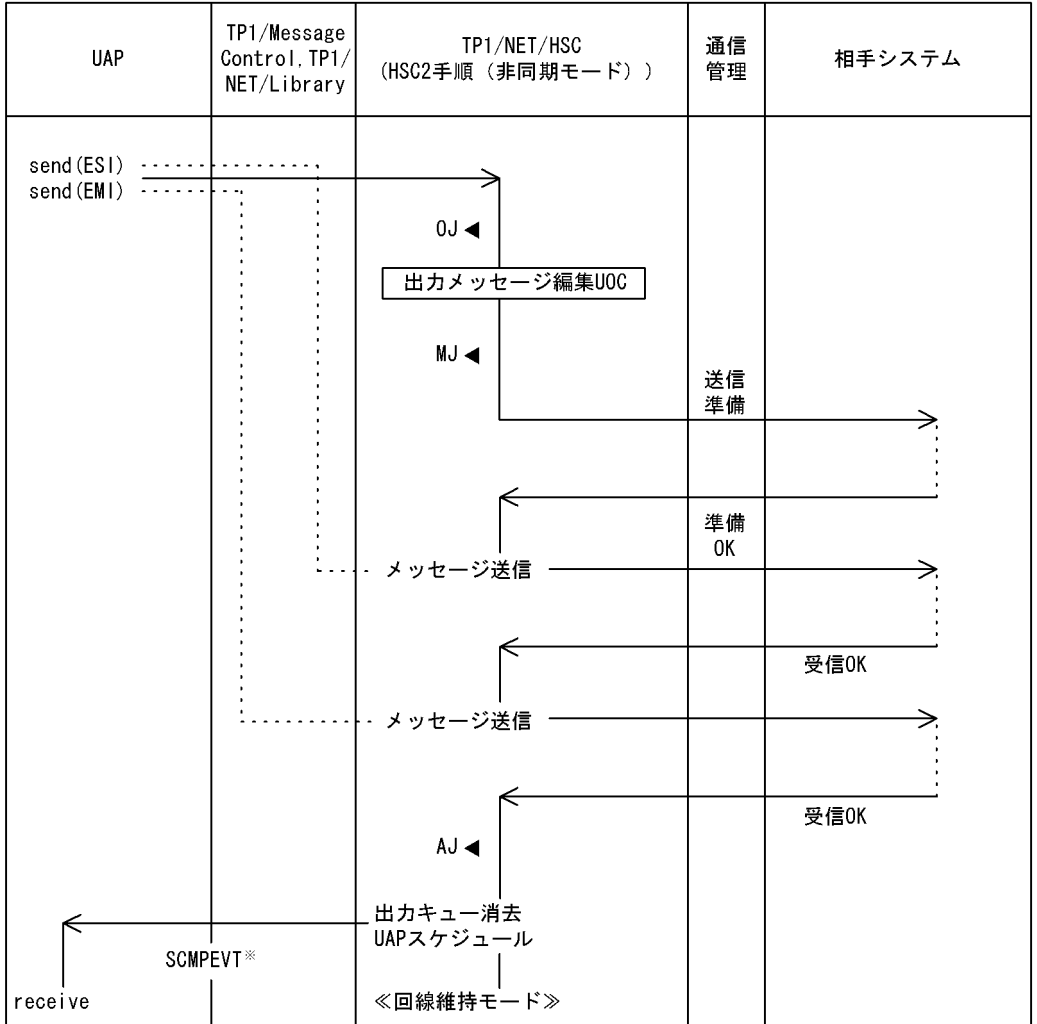
MCF 通信構成定義で ID 交換あり (mcfaltccn コマンドの -j オプション idchange オペランドで use) を指定した場合

注 2

send 関数で送信完了通知イベントを通知させることを指定した場合

(2) 複数セグメントの送信

図 A-8 HSC2 手順 (非同期モード) のメッセージ送信の処理の流れ (複数セグメントの場合)



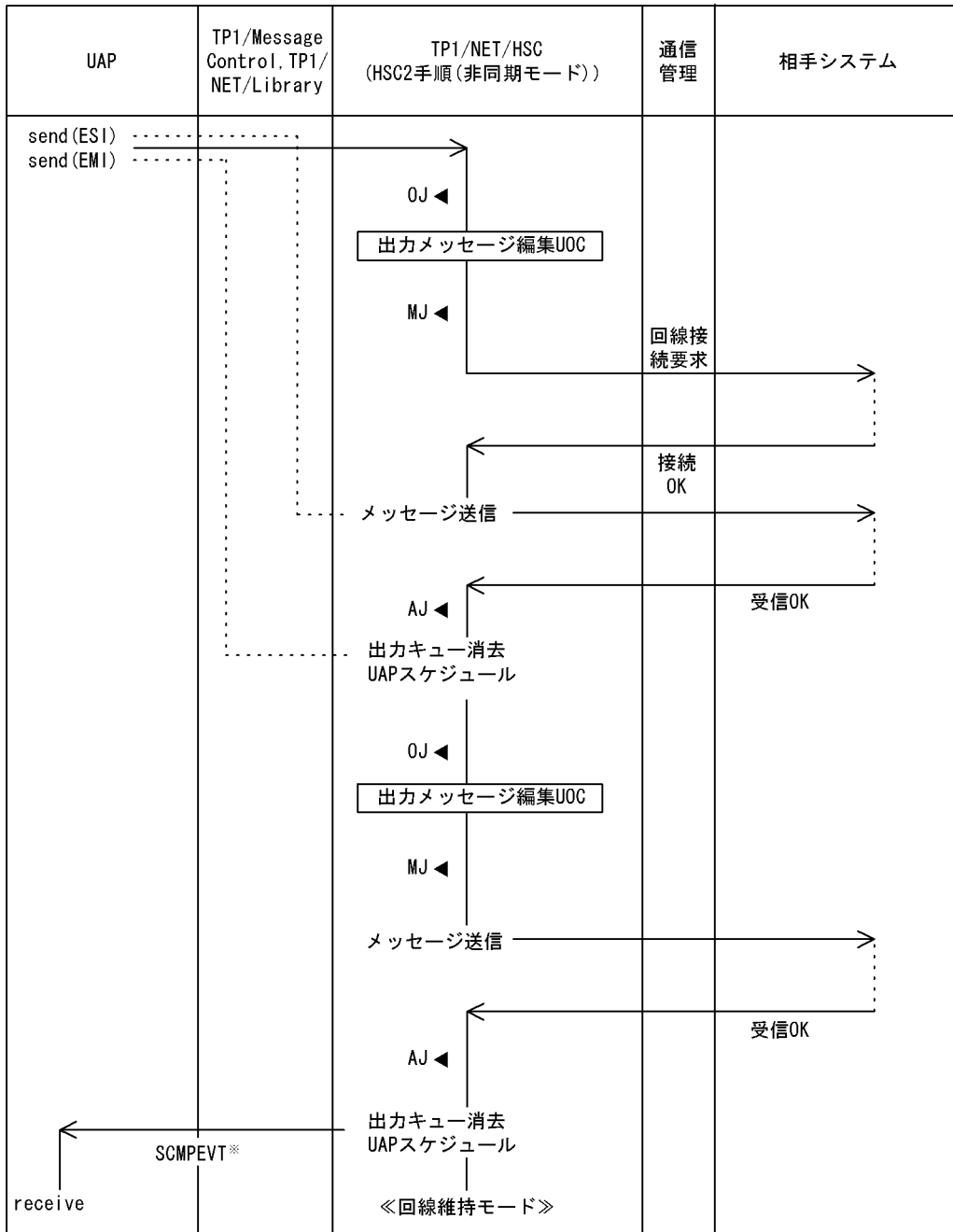
- (凡例)
- OJ ◀: メッセージ出力ジャーナル取得
 - MJ ◀: メッセージジャーナル取得
 - AJ ◀: メッセージ送信完了ジャーナル取得

注

send 関数で送信完了通知イベントを通知させることを指定した場合

(3) 列信の場合の送信

図 A-9 HSC2 手順 (非同期モード) のメッセージ送信の処理の流れ (列信の場合)



(凡例)

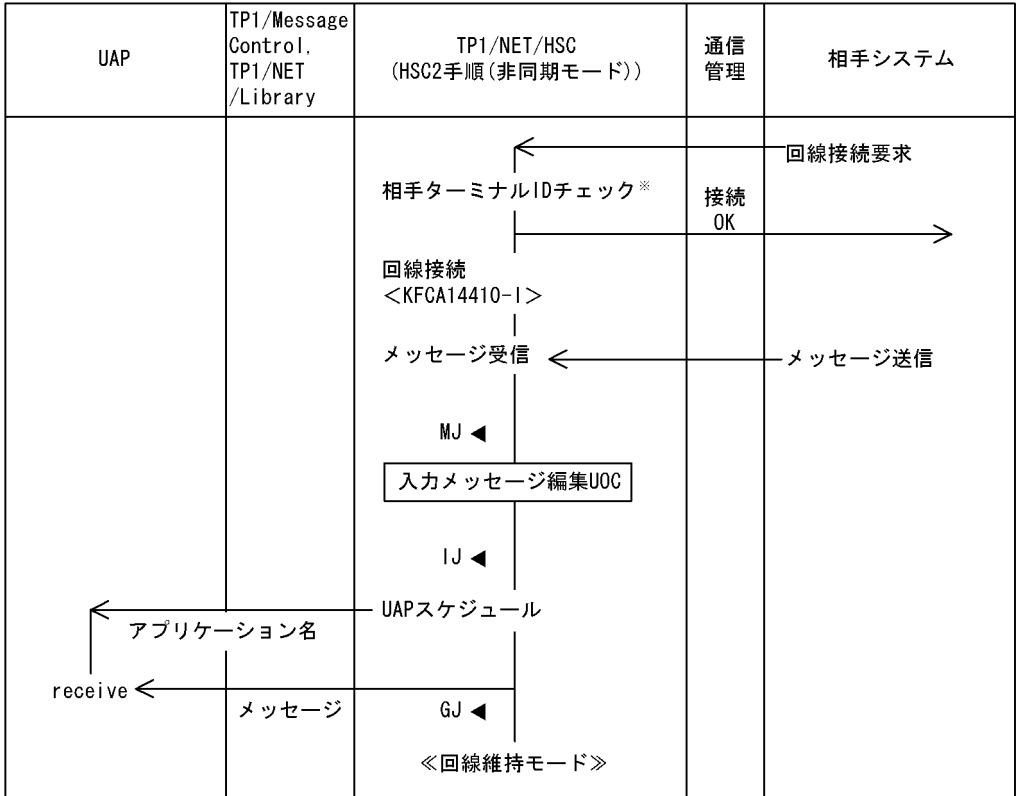
- OJ ◀: メッセージ出力ジャーナル取得
- MJ ◀: メッセージジャーナル取得
- AJ ◀: メッセージ送信完了ジャーナル取得

注

send 関数で送信完了通知イベントを通知させることを指定した場合

(4) 単一セグメントの受信

図 A-10 HSC2 手順 (非同期モード) のメッセージ受信の処理の流れ (単一セグメントの場合)



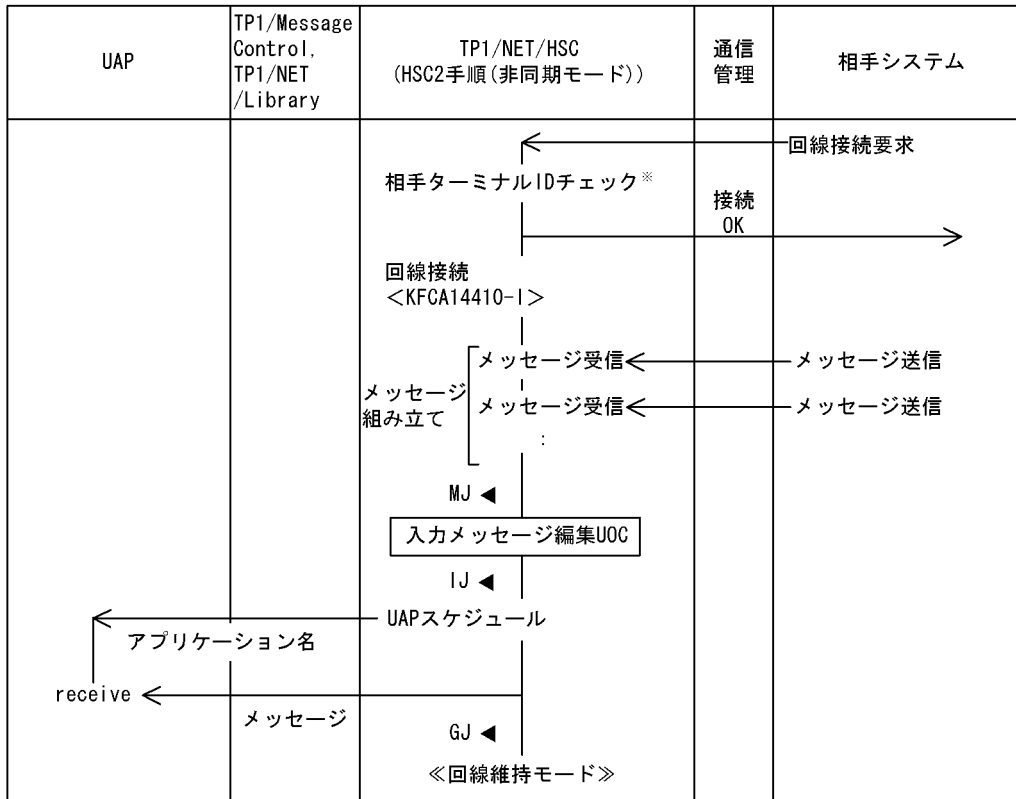
- (凡例) MJ ◀: メッセージジャーナル取得
 IJ ◀: メッセージ入力ジャーナル取得
 GJ ◀: メッセージ受信ジャーナル取得

注

MCF 通信構成定義で ID 交換あり (mcftalcen コマンドの j オプション idchange オペランドで use) を指定した場合

(5) 複数セグメントの受信

図 A-11 HSC2 手順 (非同期モード) のメッセージ受信の処理の流れ (複数セグメントの場合)



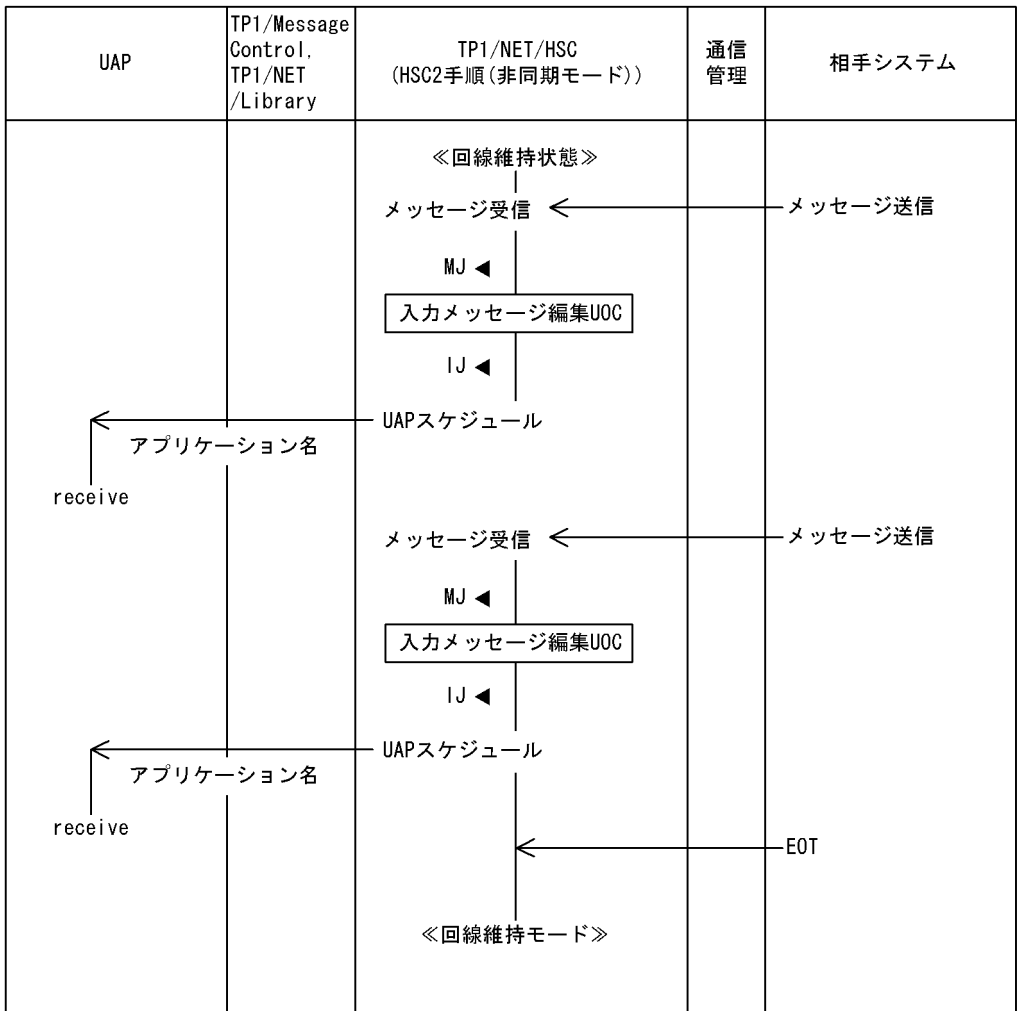
- (凡例) MJ ◀: メッセージジャーナル取得
 IJ ◀: メッセージ入力ジャーナル取得
 GJ ◀: メッセージ受信ジャーナル取得

注

MCF 通信構成定義で ID 交換あり (mcfaltccn コマンドの -j オプション idchange オプションで use) を指定した場合

(6) 列信の場合の受信

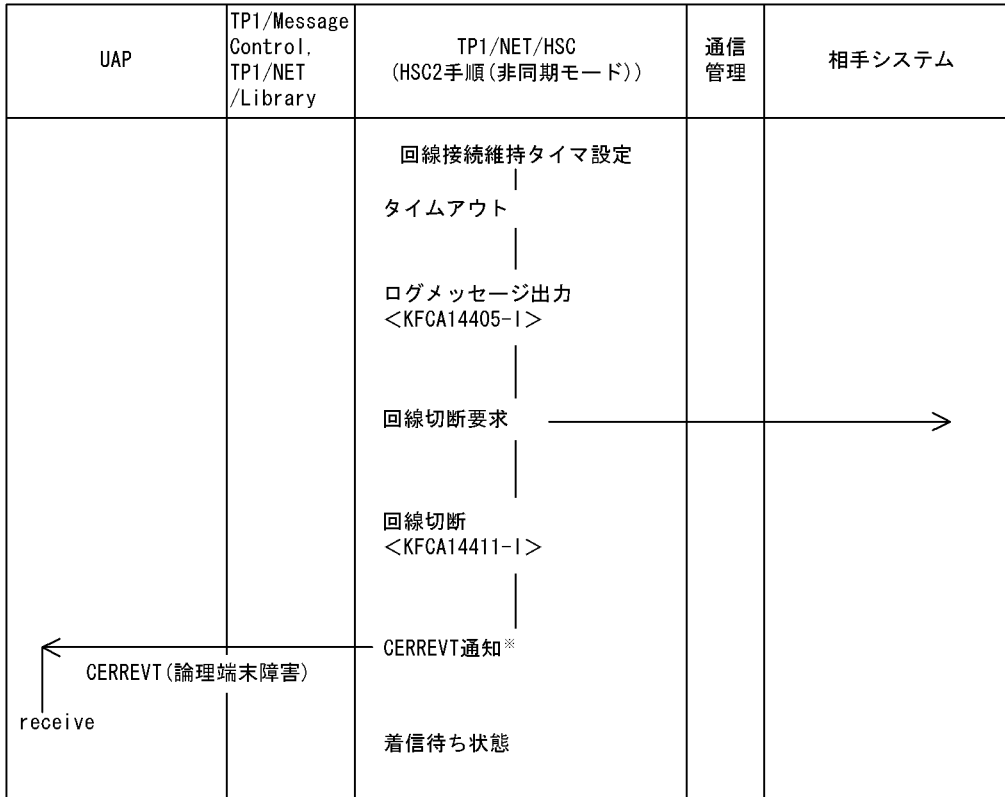
図 A-12 HSC2 手順 (非同期モード) のメッセージ受信の処理の流れ (列信の場合)



(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

(7) 回線の切断

図 A-13 HSC2 手順 (非同期モード) の回線切断の処理の流れ (回線維持タイムアウトの場合)

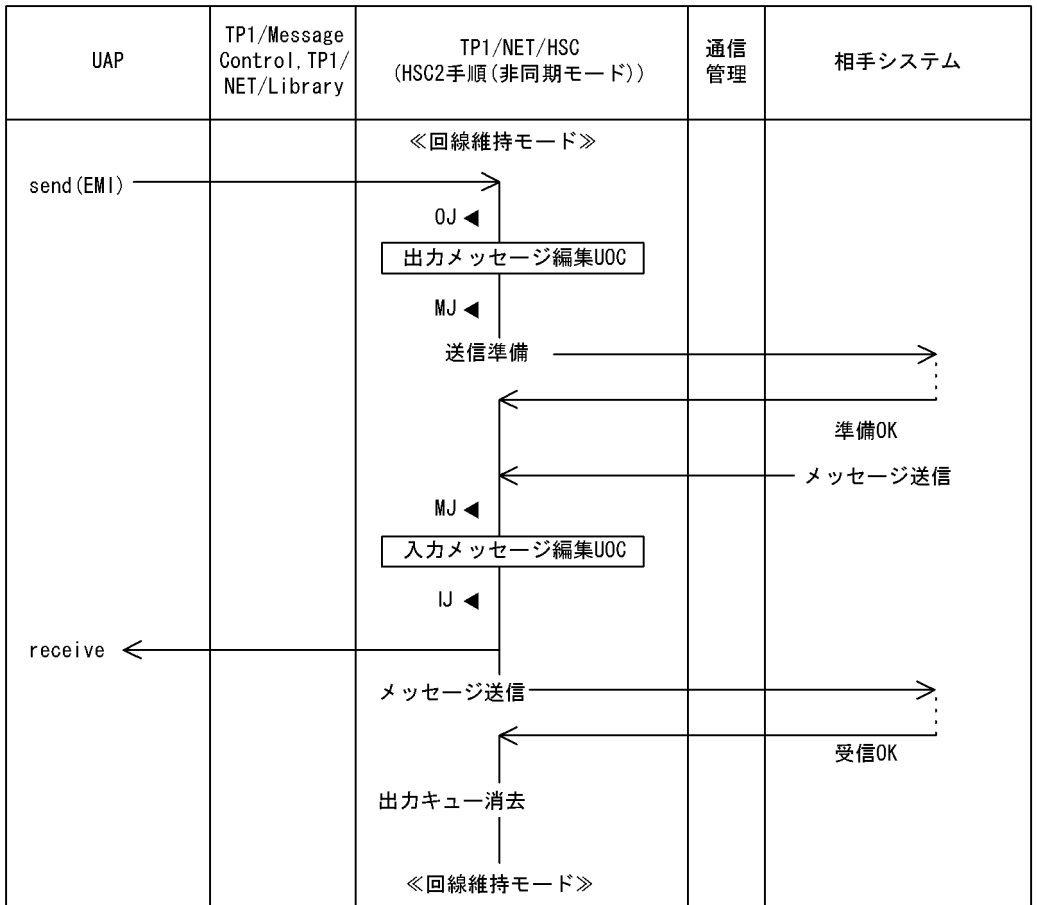


注

MCF 通信構成定義の mcftalcn コマンドの -v オプション discinf オペランドで use を指定した場合

(8) 送受信反転制御

図 A-14 送受信反転制御



(凡例)

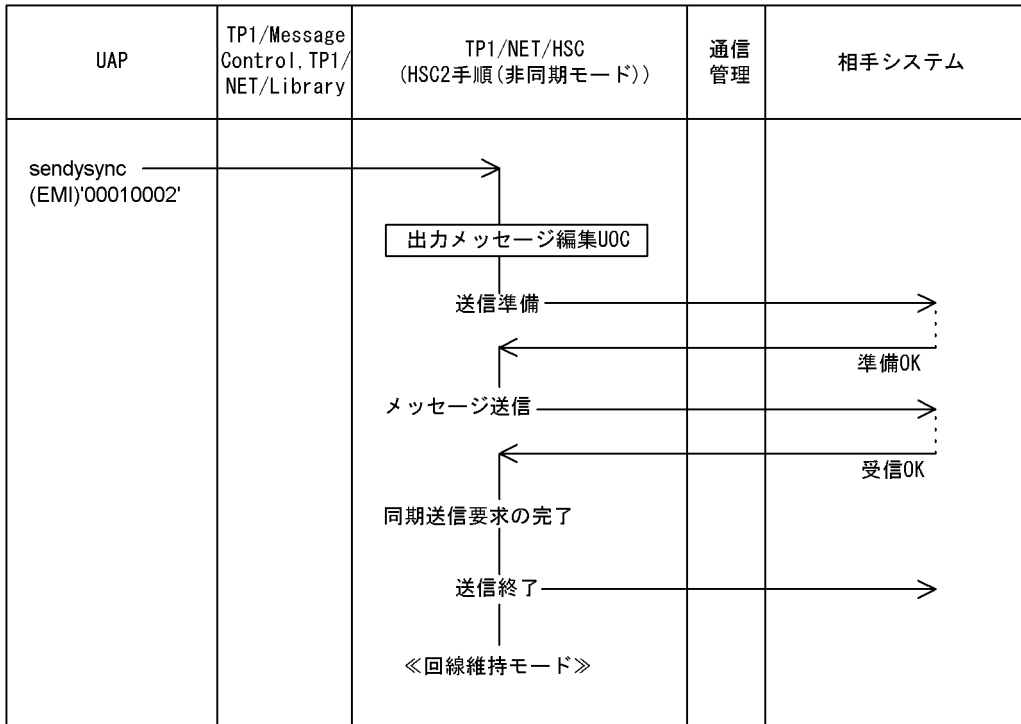
- 0J ◀: メッセージ出力ジャーナル取得
- MJ ◀: メッセージジャーナル取得
- IJ ◀: メッセージ送信完了ジャーナル取得

付録 A.3 HSC2 手順 (同期モード) のメッセージの送受信

HSC2 手順 (同期モード) のメッセージを送受信するときの処理の流れを、図 A-15 ~ 図 A-24 に示します。

(1) 単一セグメントの送信

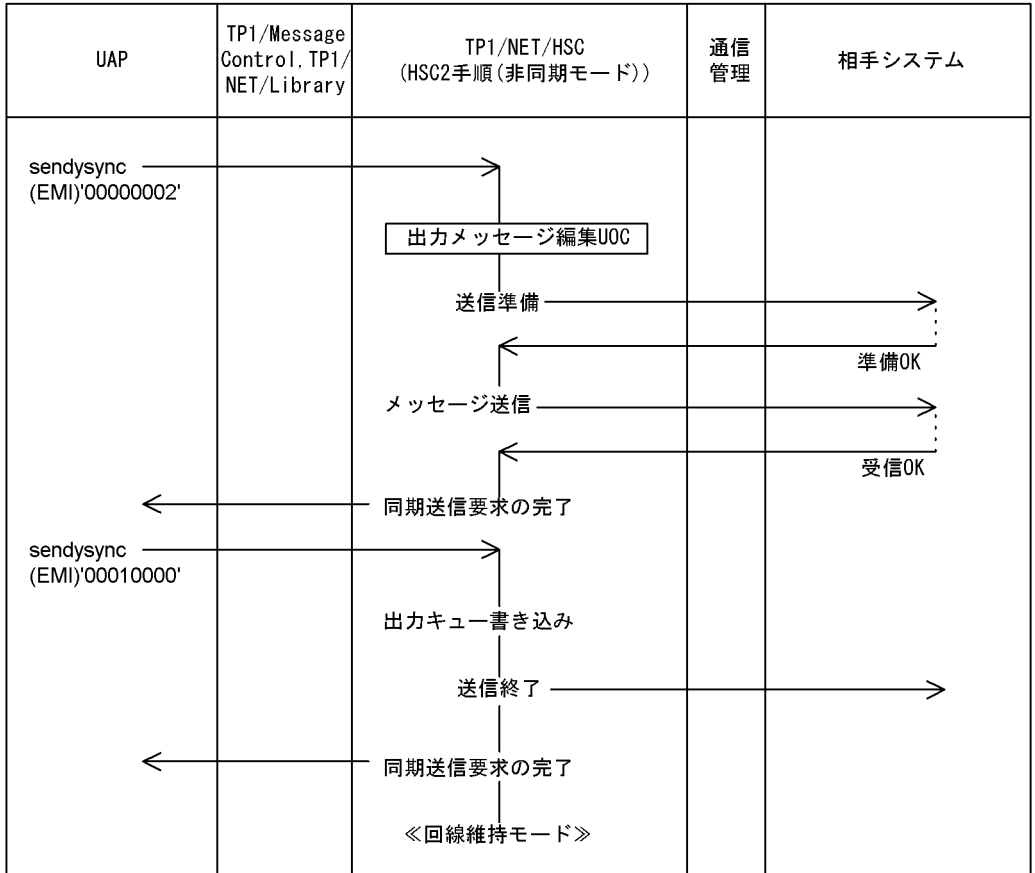
図 A-15 HSC2 手順 (同期モード) のメッセージ送信の処理の流れ (単一セグメントで通常テキスト送信後, EOT を送信する場合)



(凡例)

'00010002': 通常テキスト送信後, EOTを送信することを表すテキスト形態の数値表現

図 A-16 HSC2 手順 (同期モード) のメッセージ送信の処理の流れ (単一セグメントで通常テキスト送信する場合)



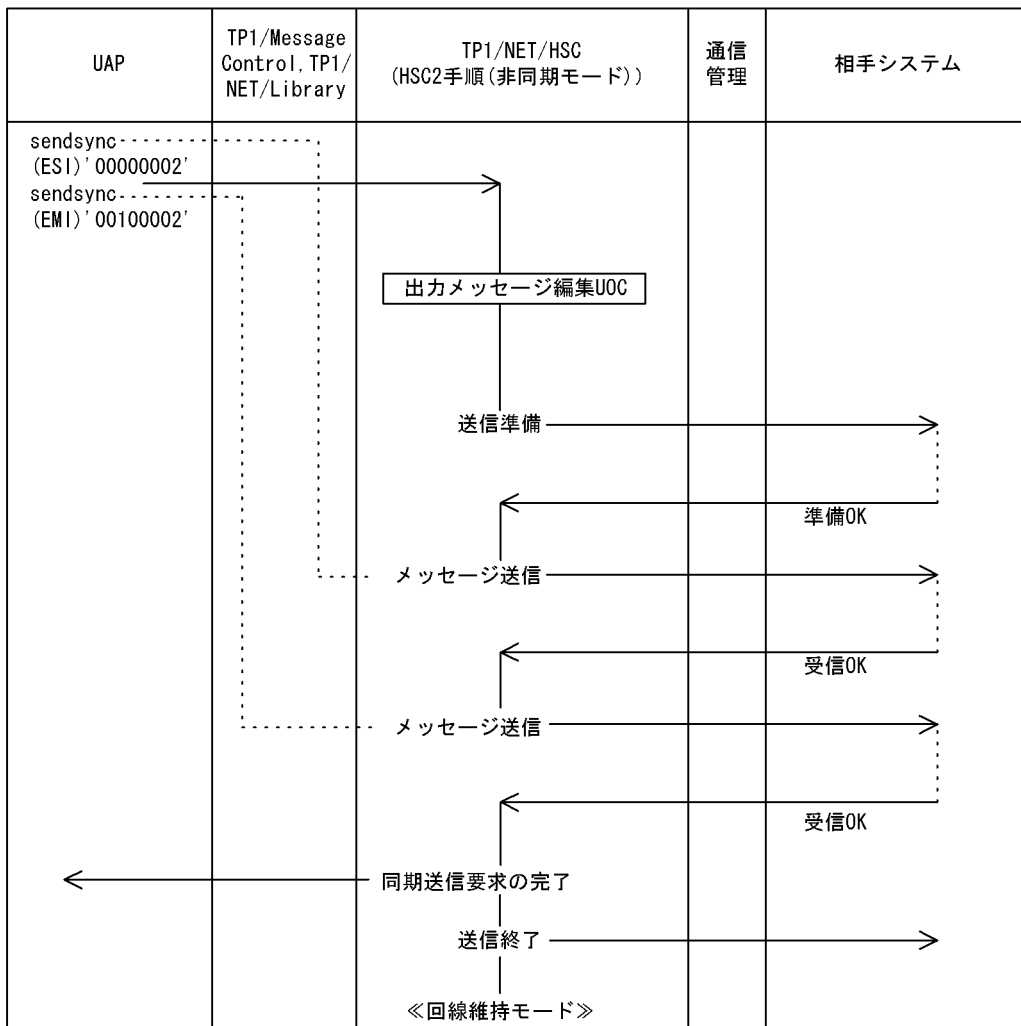
(凡例)

'00000002' : 通常テキストを送信することを表すテキスト形態の数値表現

'00010000' : ET0を送信することを表すテキスト形態の数値表現

(2) 複数セグメントの送信

図 A-17 HSC2 手順 (同期モード) のメッセージ送信の処理の流れ (複数セグメントの場合)



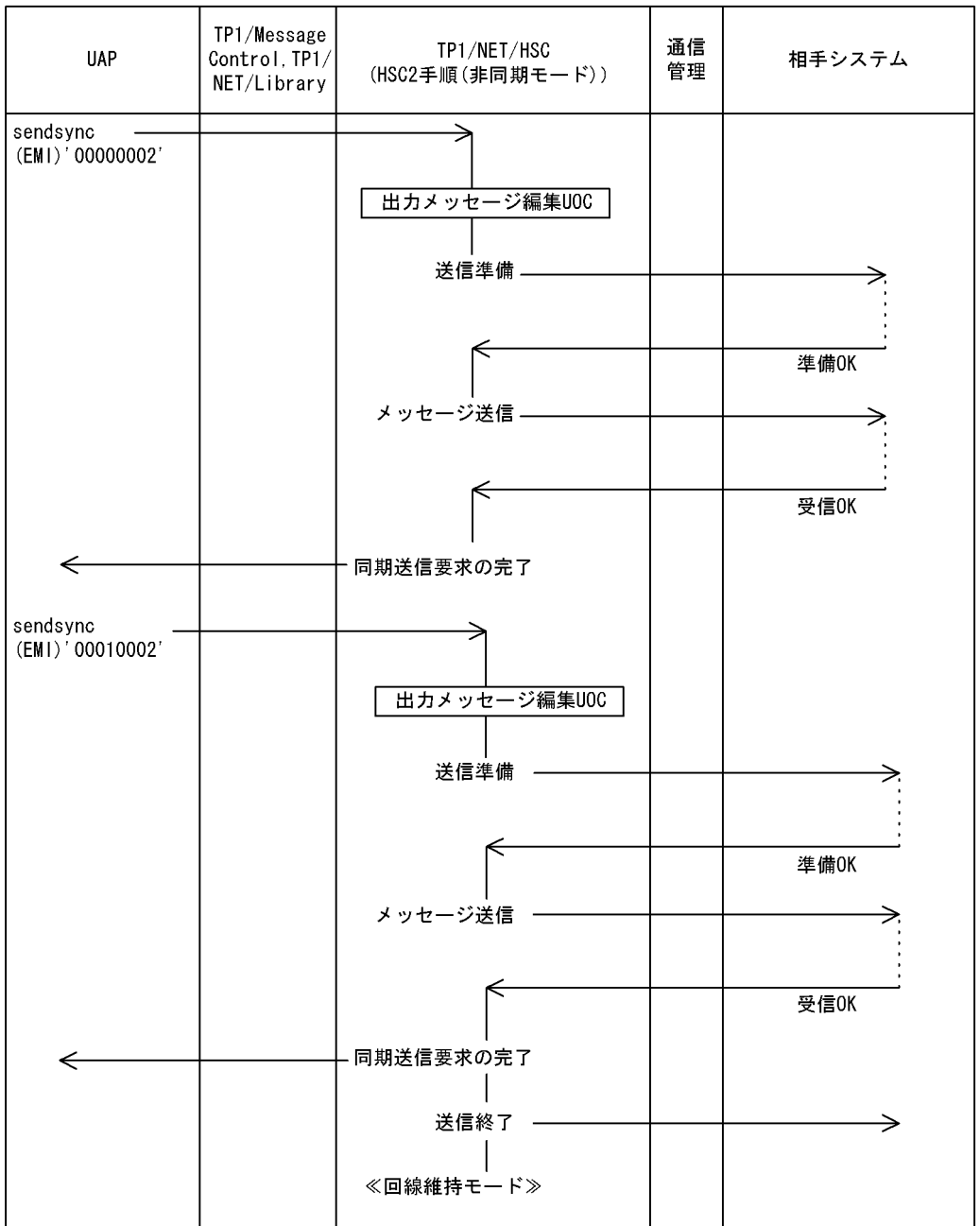
(凡例)

'00000002' : 通常テキストを送信することを表すテキスト形態の数値表現

'00010002' : 通常テキスト送信後, EOTを送信することを表すテキスト形態の数値表現

(3) 列信の場合の送信

図 A-18 HSC2 手順 (同期モード) のメッセージ送信の処理の流れ (列信の場合)



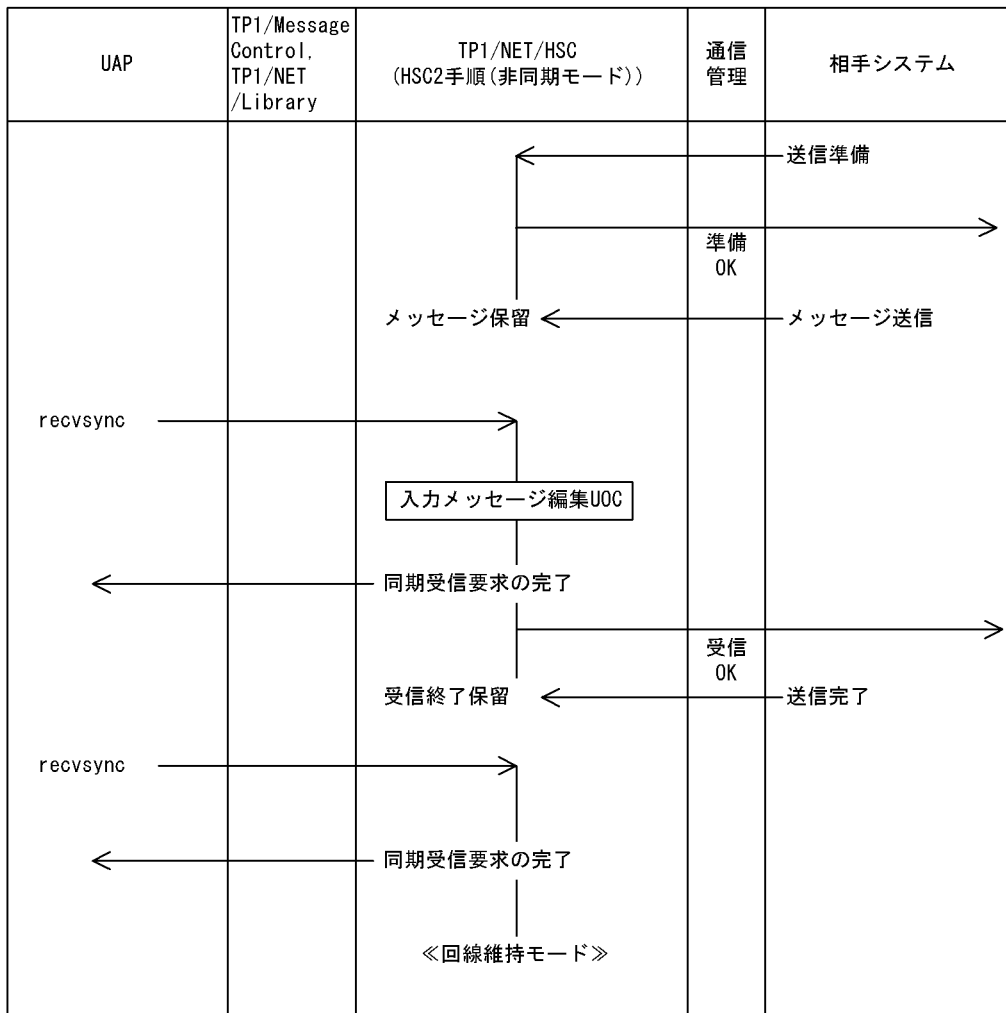
(凡例)

'00000002' : 通常テキストを送信することを表すテキスト形態の数値表現

'00010002' : 通常テキスト送信後, EOTを送信することを表すテキスト形態の数値表現

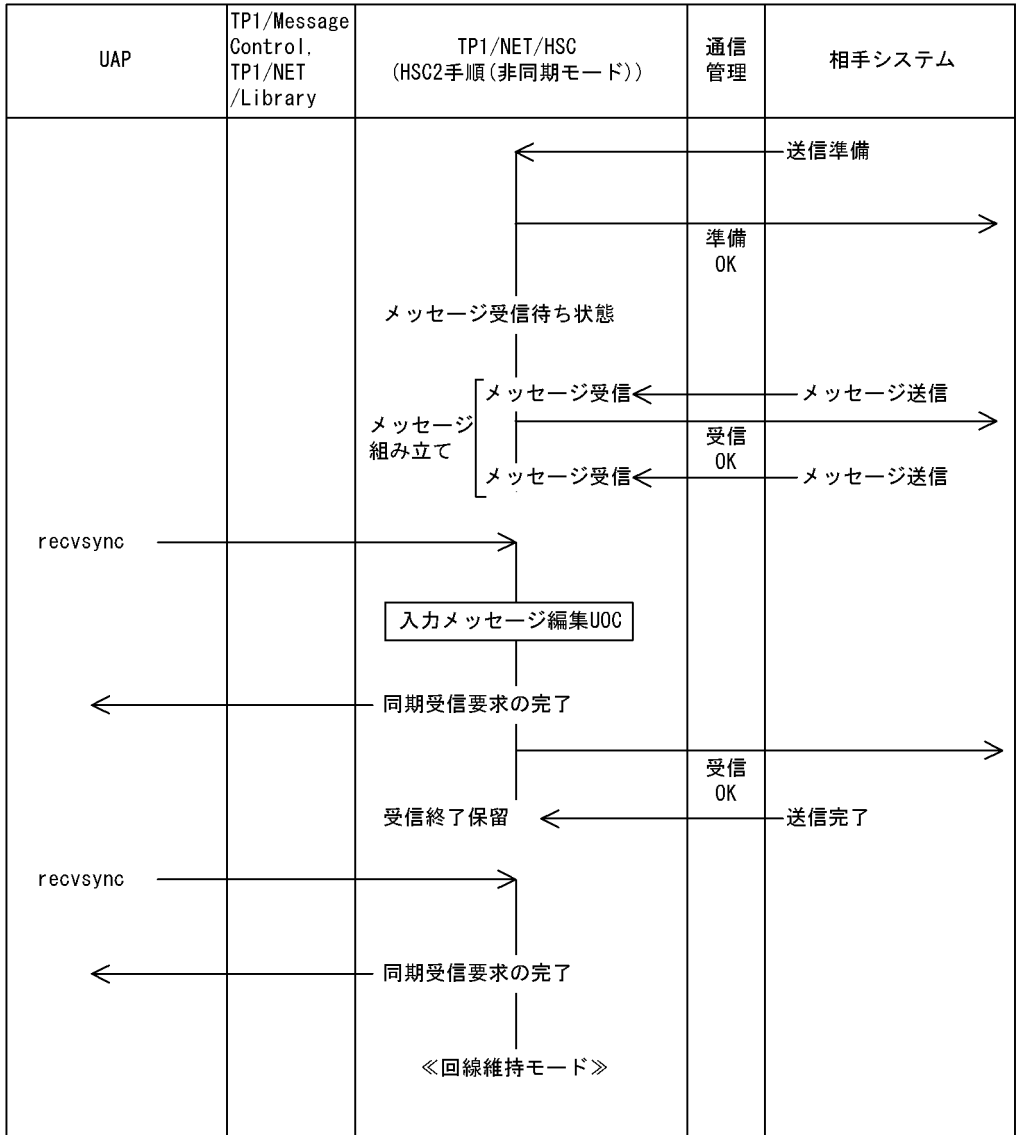
(4) 単一セグメントの受信

図 A-19 HSC2 手順 (同期モード) のメッセージ受信の処理の流れ (単一セグメントの場合)



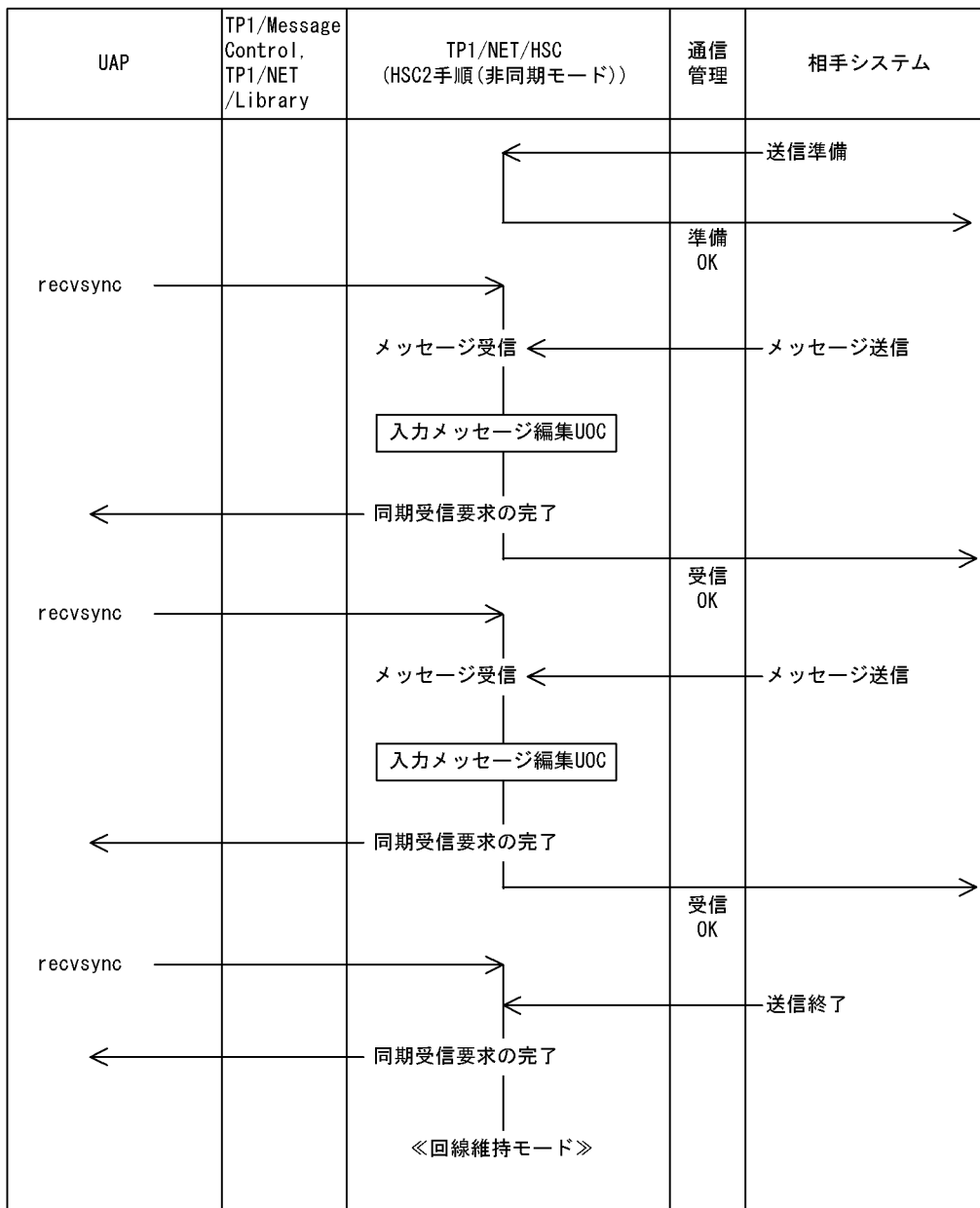
(5) 複数セグメントの受信

図 A-20 HSC2 手順 (同期モード) のメッセージ受信の処理の流れ (複数セグメントの場合)



(6) 列信の場合の受信

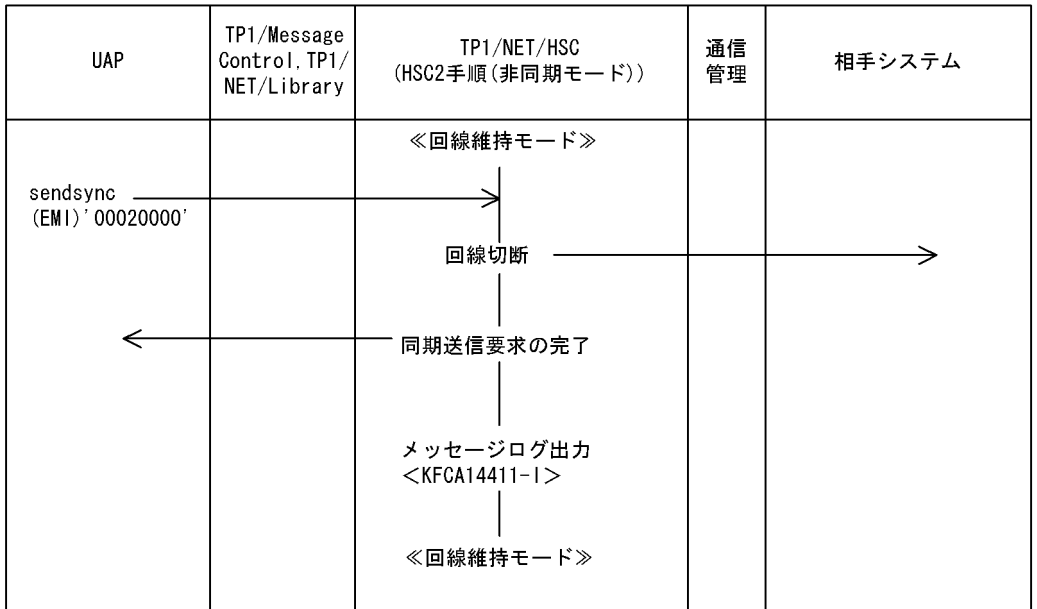
図 A-21 HSC2 手順 (同期モード) のメッセージ受信の処理の流れ (列信の場合)



(7) 回線の切断

回線の切断の処理の流れを回線切断要求送信, MHP の終了, および回線切断の受信の場合に分けて, 図 A-22 ~ 図 A-24 に示します。

図 A-22 HSC2 手順 (同期モード) の回線切断の処理の流れ (回線切断要求 (DLE・EOT) 送信の場合)



(凡例)

'00020000' : 回線切断要求のDLE・EOTを送信することを表すテキスト形態の数値表現

図 A-23 HSC2 手順 (同期モード) の回線切断の処理の流れ (MHP の終了の場合)

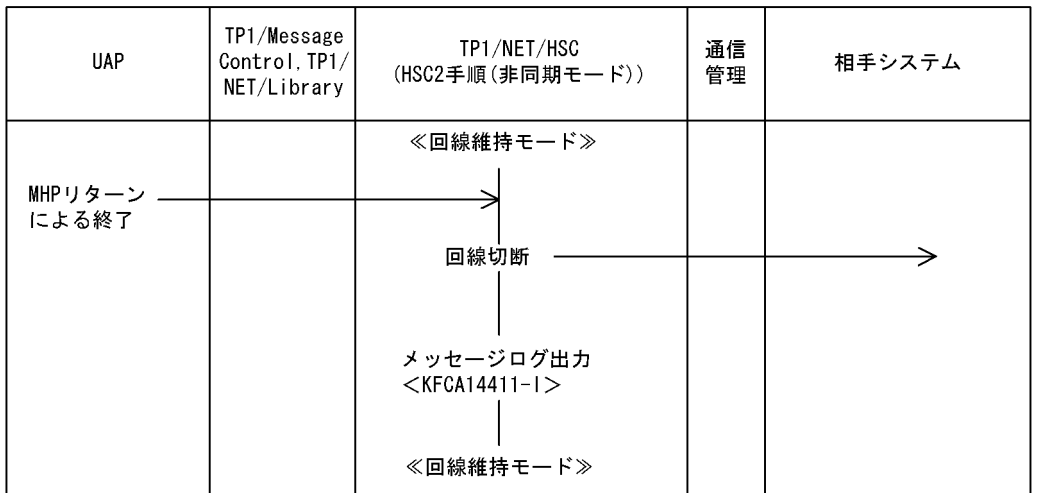
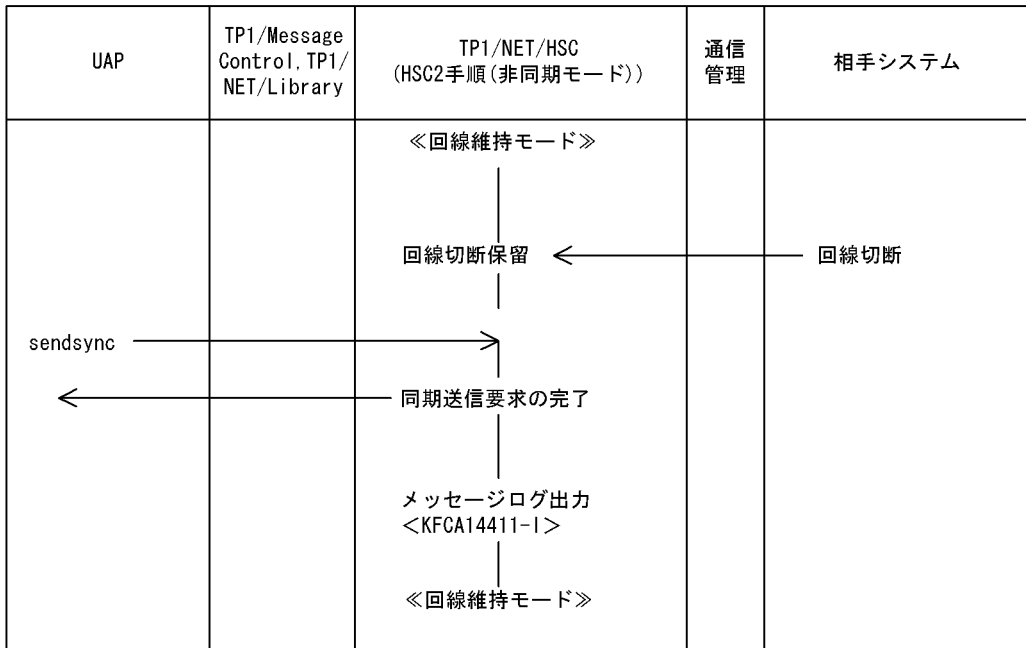


図 A-24 HSC2 手順（同期モード）の回線切断の処理の流れ（回線切断の受信の場合）



付録 B 障害発生時の処理の流れ

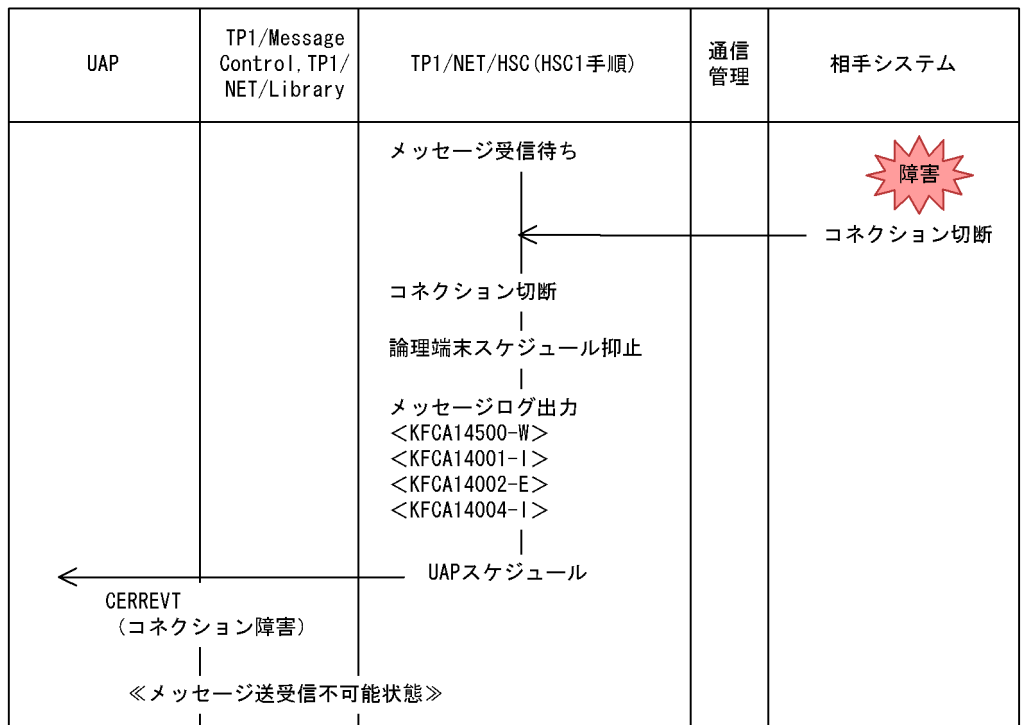
メッセージを送受信するときに発生した障害処理の流れを、HSC1 手順、HSC2 手順（非同期モード）および HSC2 手順（同期モード）に分けて説明します。

付録 B.1 HSC1 手順の障害

HSC1 手順の障害発生時の処理の流れを、図 B-1 ~ 図 B-6 に示します。

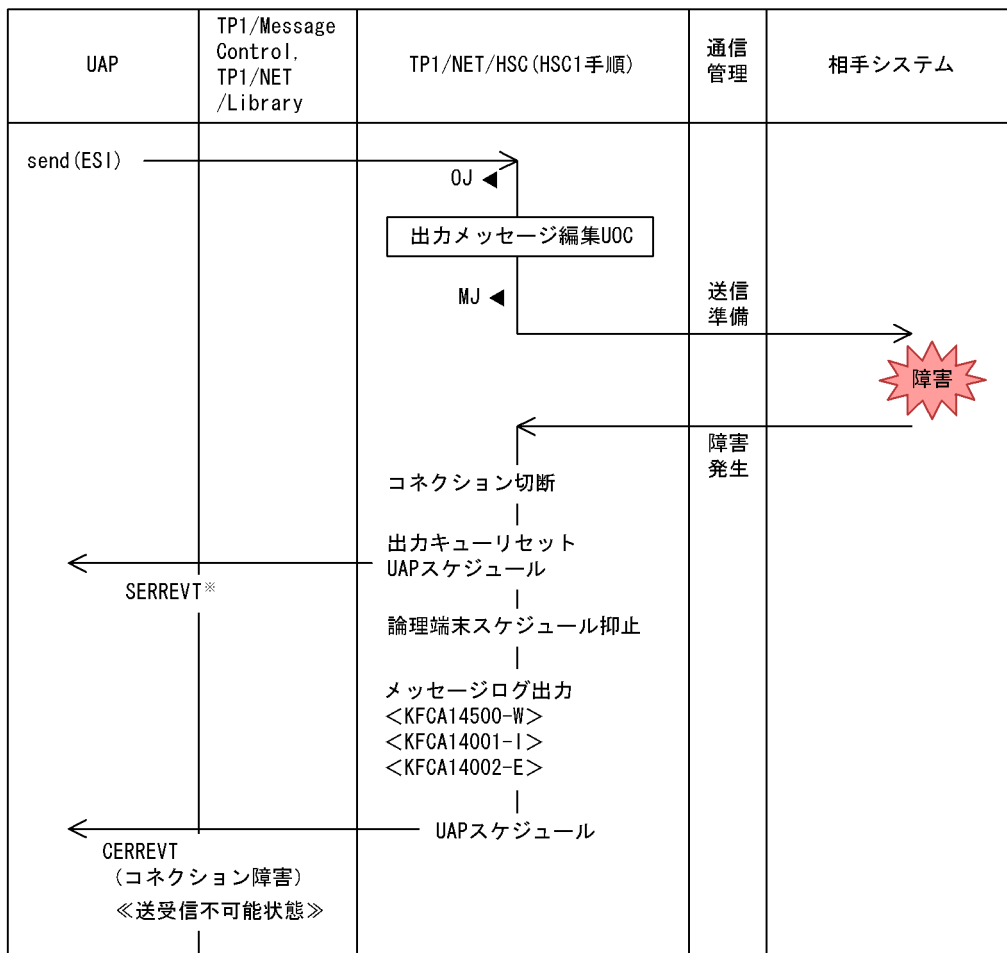
(1) コネクション切断

図 B-1 コネクション切断



(2) メッセージ送信時の障害

図 B-2 メッセージ送信時の障害



(凡例) 0J ◀: メッセージ出力ジャーナル取得

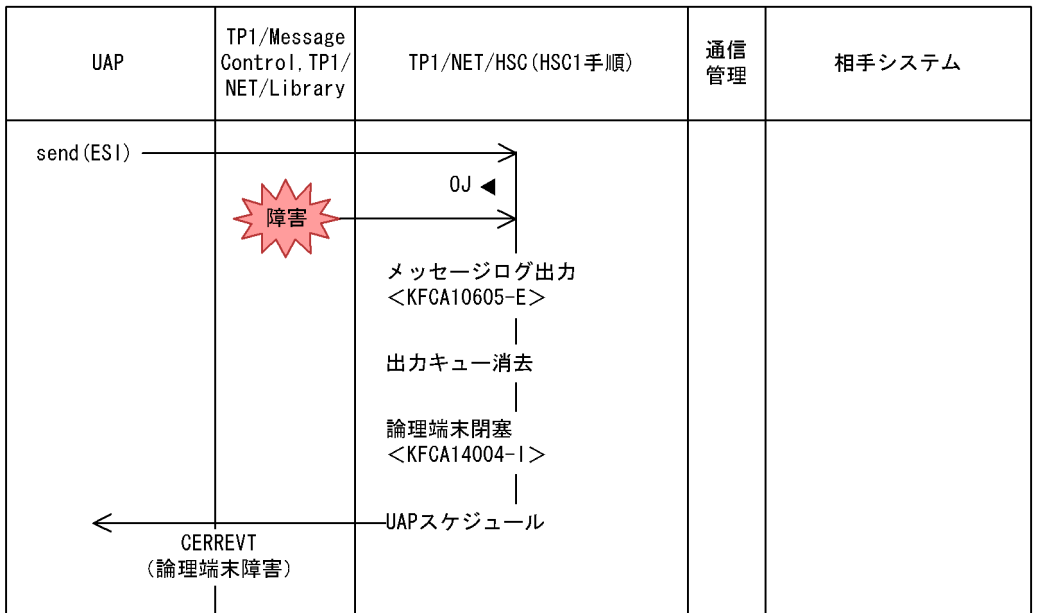
MJ ◀: メッセージジャーナル取得

注

send 関数で送信障害通知イベントを通知させることを指定した場合

(3) 出力キュー読み込み障害

図 B-3 出力キュー読み込み障害

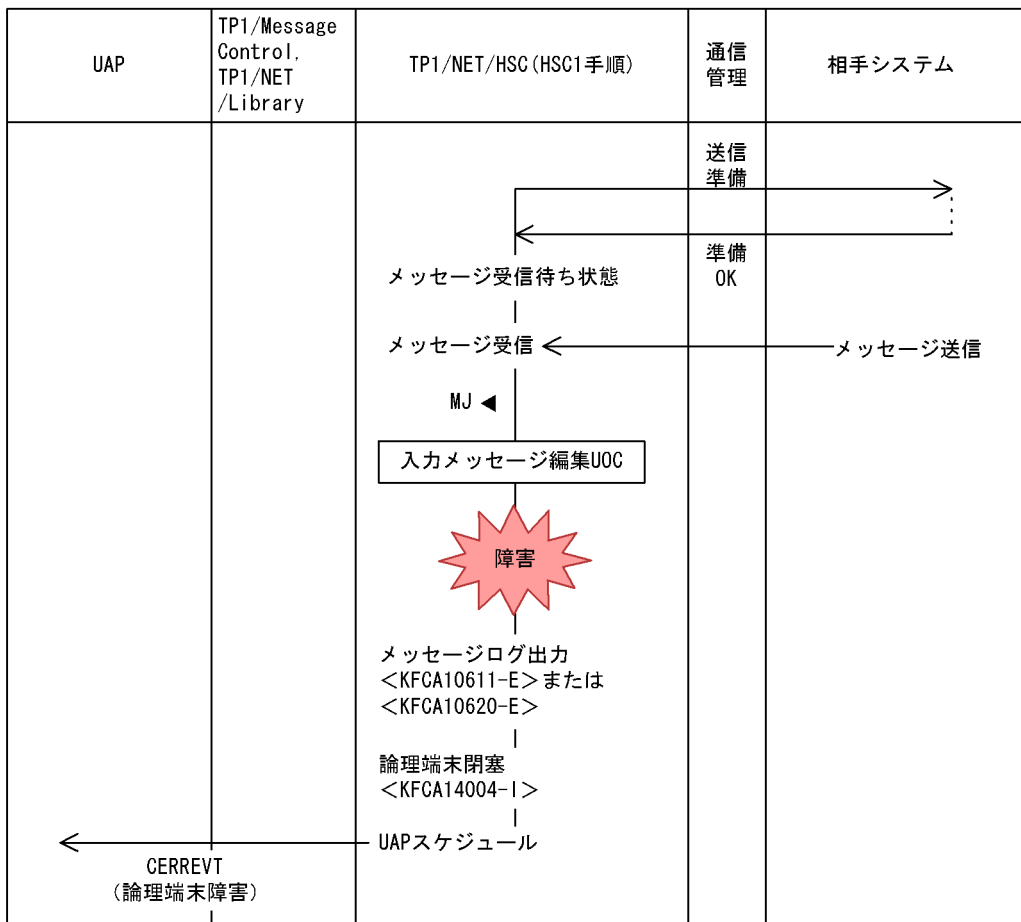


(凡例)

0J ◀: メッセージ出力ジャーナル取得

(4) 入力メッセージ編集 UOC 障害

図 B-4 HSC1 手順の入力メッセージ編集 UOC 障害

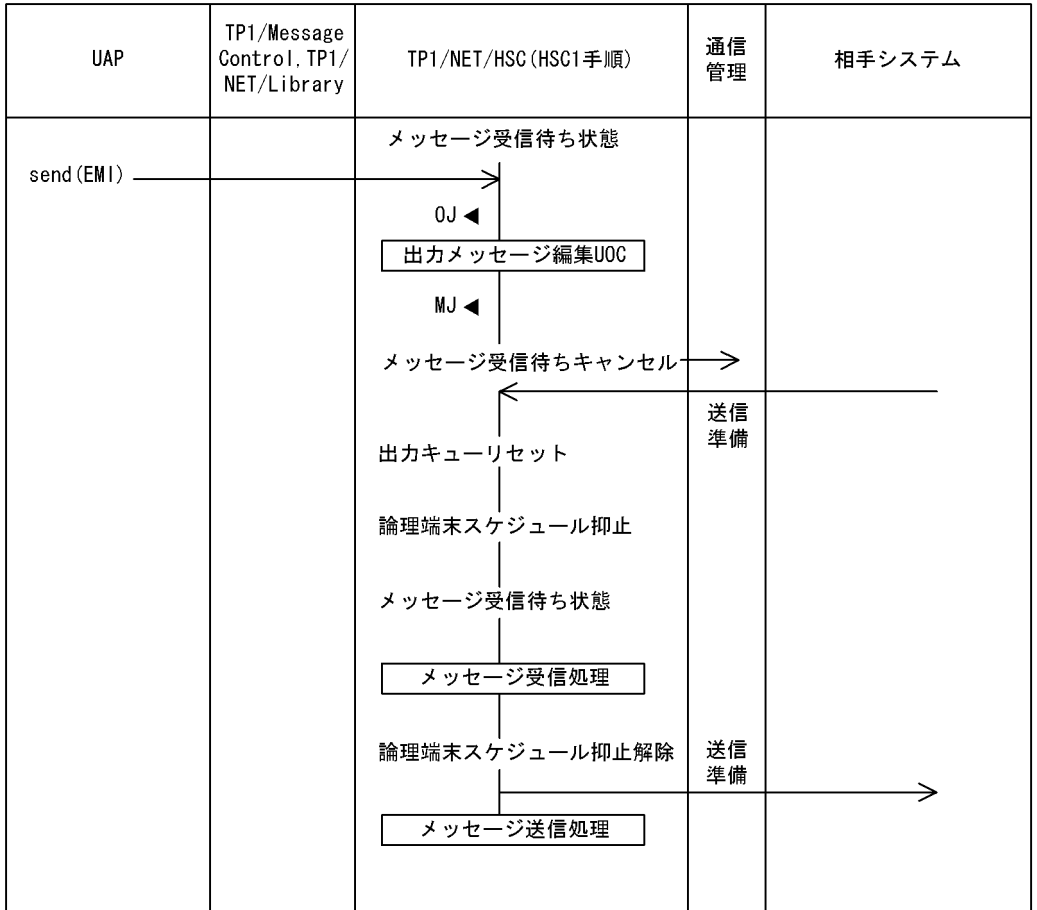


(凡例)

MJ ◀: メッセージジャーナル取得

(5) コンテンション

図 B-5 HSC1 手順のコンテンツン



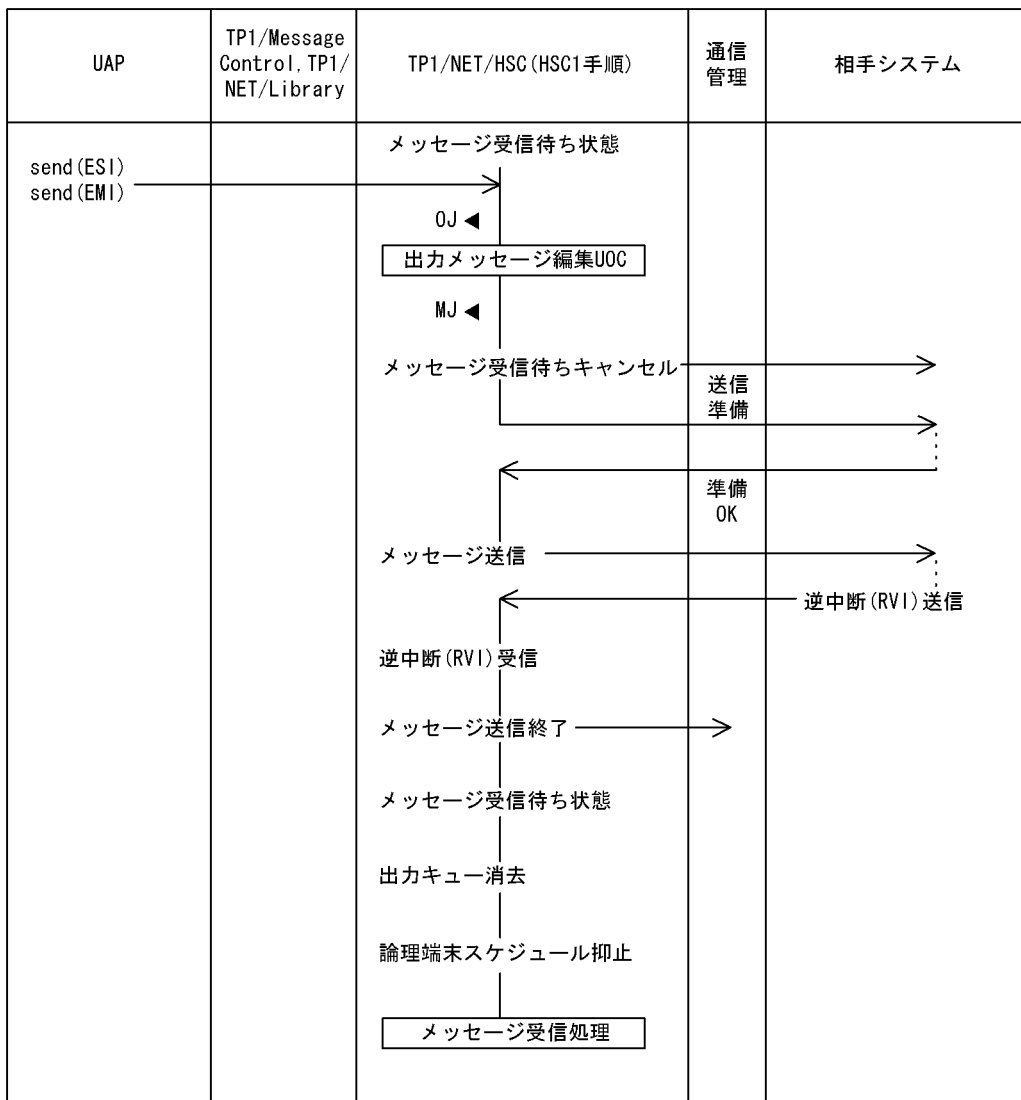
(凡例)

0J ◀: メッセージ出力ジャーナル取得

MJ ◀: メッセージジャーナル取得

(6) 逆中断 (RVI) 受信

図 B-6 HSC1 手順の逆中断 (RVI) 受信 (mcftalccn コマンドの -u オプションで accept を指定した場合)



(凡例)

0J ◀: メッセージ出力ジャーナル取得

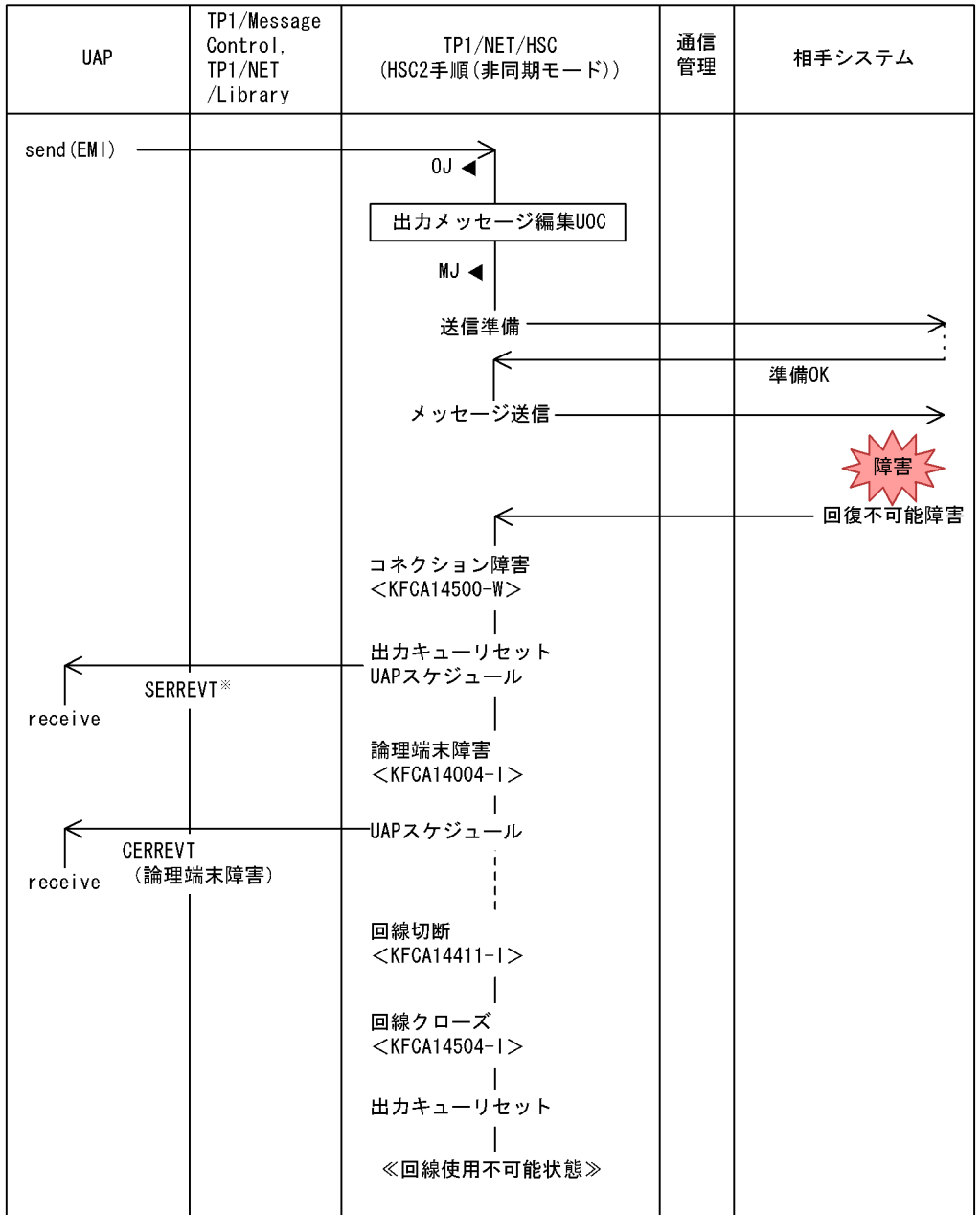
MJ ◀: メッセージジャーナル取得

付録 B.2 HSC2 手順 (非同期モード) の障害

HSC2 手順 (非同期モード) の障害発生時の処理の流れを, 図 B-7 ~ 図 B-11 に示します。

(1) メッセージ送信時の障害

図 B-7 HSC2 手順（非同期モード）のメッセージ送信時の回復不可能障害

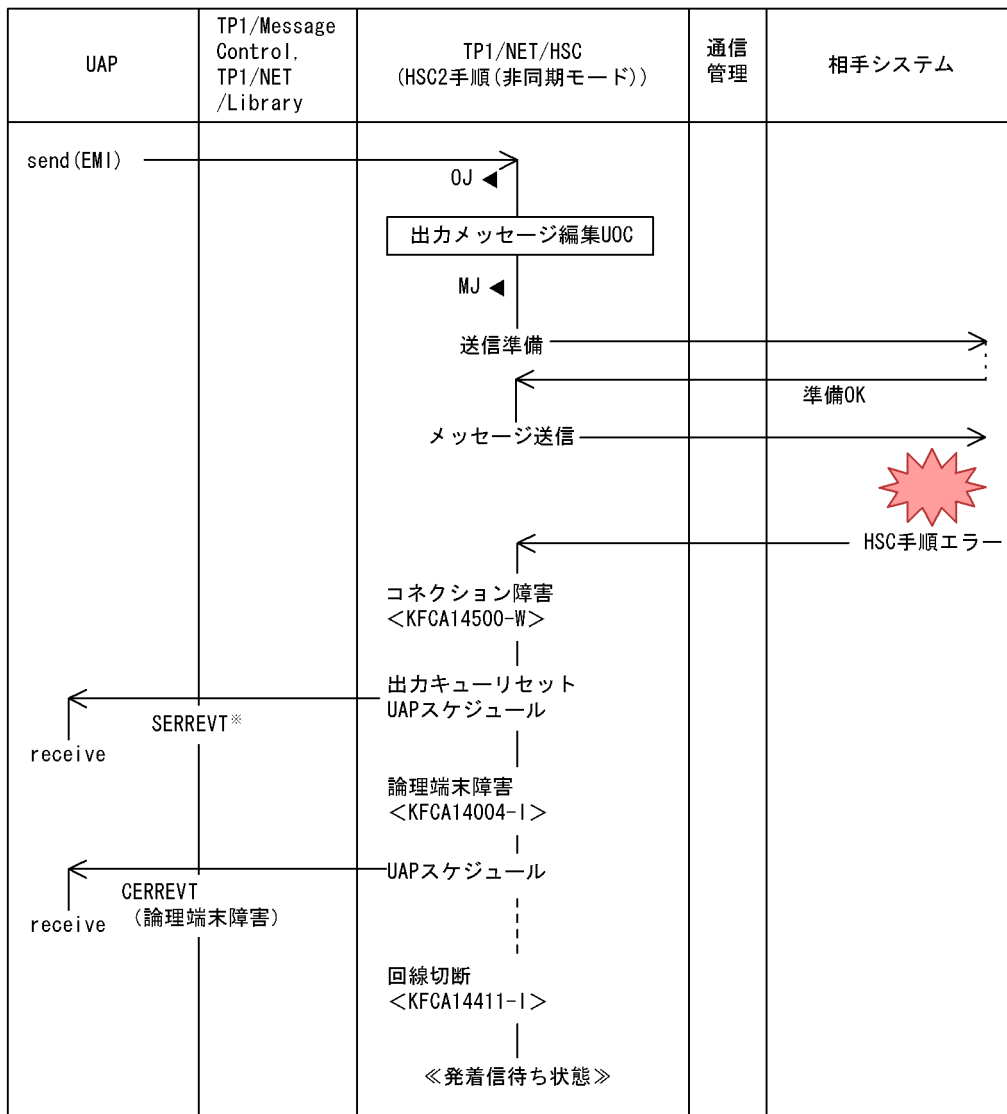


(凡例) 0J ◀: メッセージ出カジャーナル取得
MJ ◀: メッセージジャーナル取得

注

send 関数で送信障害通知イベントを通知させることを指定した場合

図 B-8 HSC2 手順（非同期モード）のメッセージ送信時の HSC 手順エラー



(凡例) 0J ◀: メッセージ出力ジャーナル取得

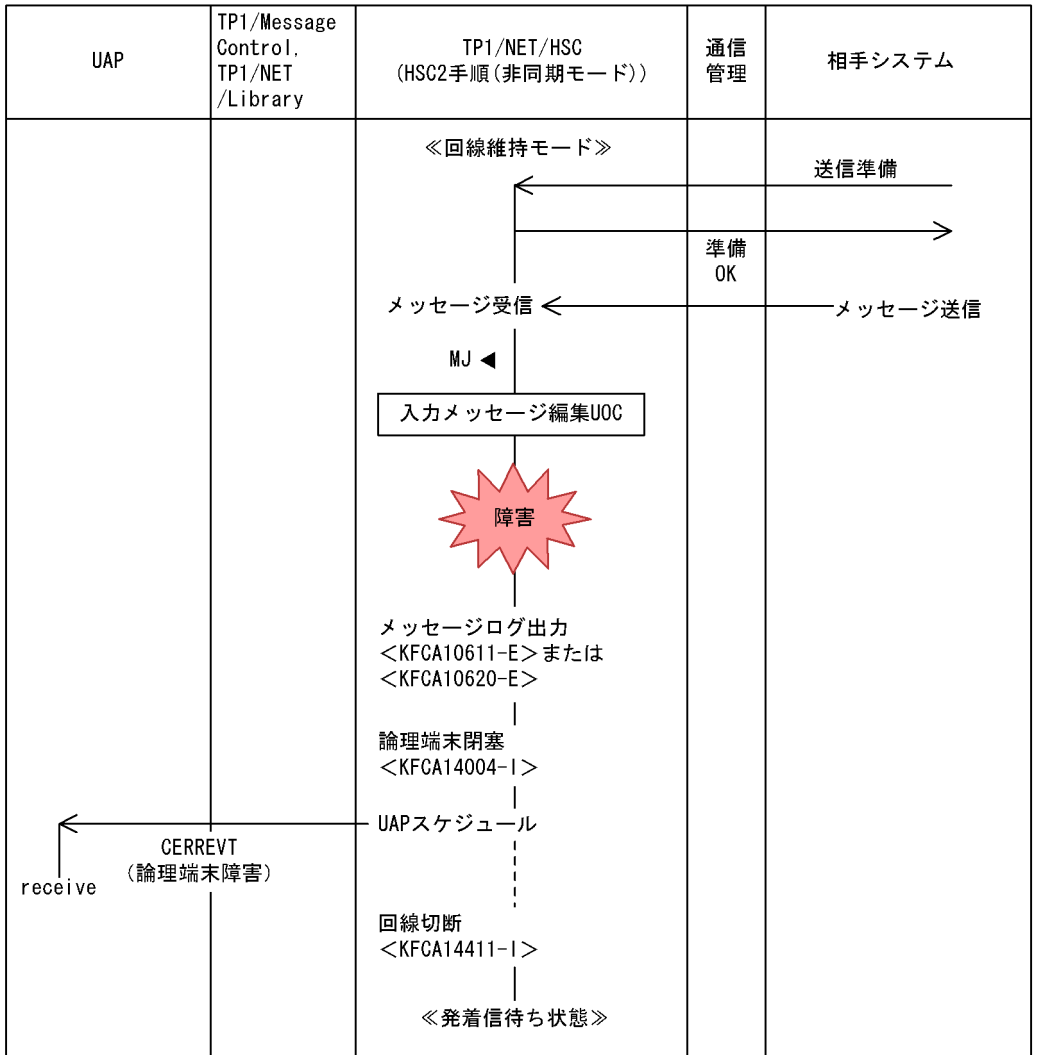
MJ ◀: メッセージジャーナル取得

注

send 関数で送信障害通知イベントを通知させることを指定した場合

(2) 入力メッセージ編集 UOC 障害

図 B-9 HSC2 手順 (非同期モード) の入力メッセージ編集 UOC 障害

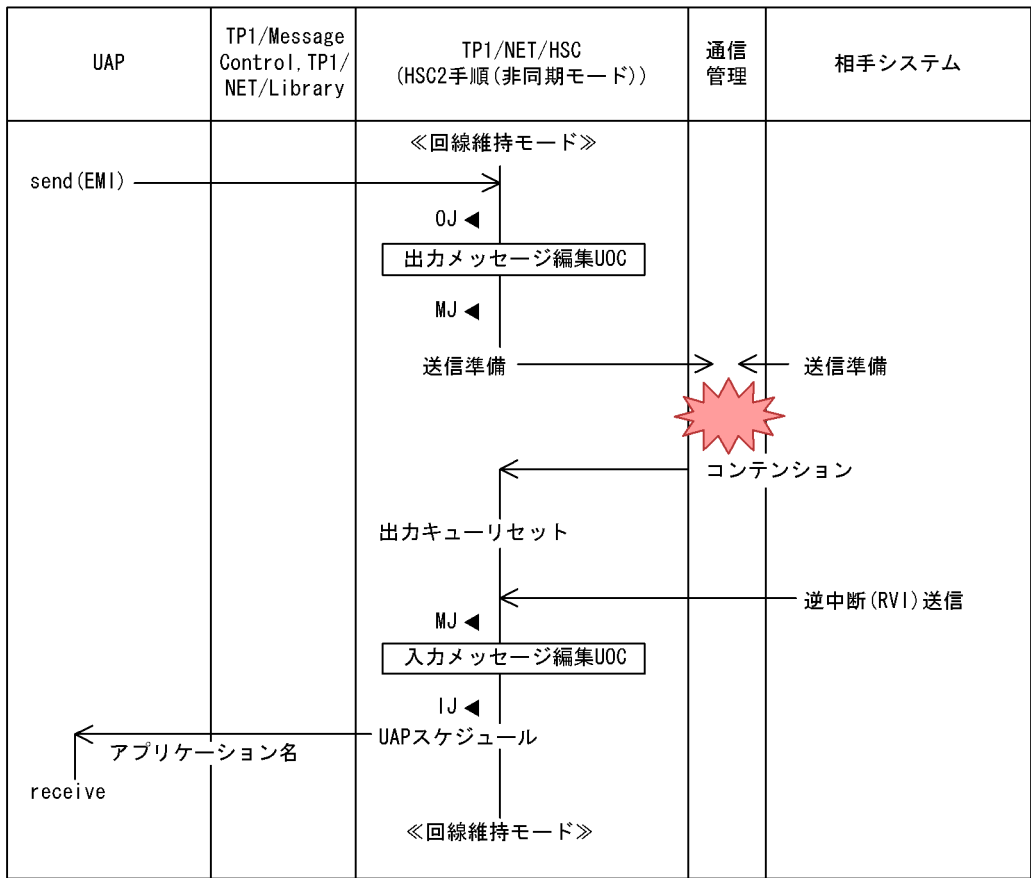


(凡例)

MJ ◀: メッセージジャーナル取得

(3) コンテンション

図 B-10 HSC2 手順 (非同期モード) のコンテンション

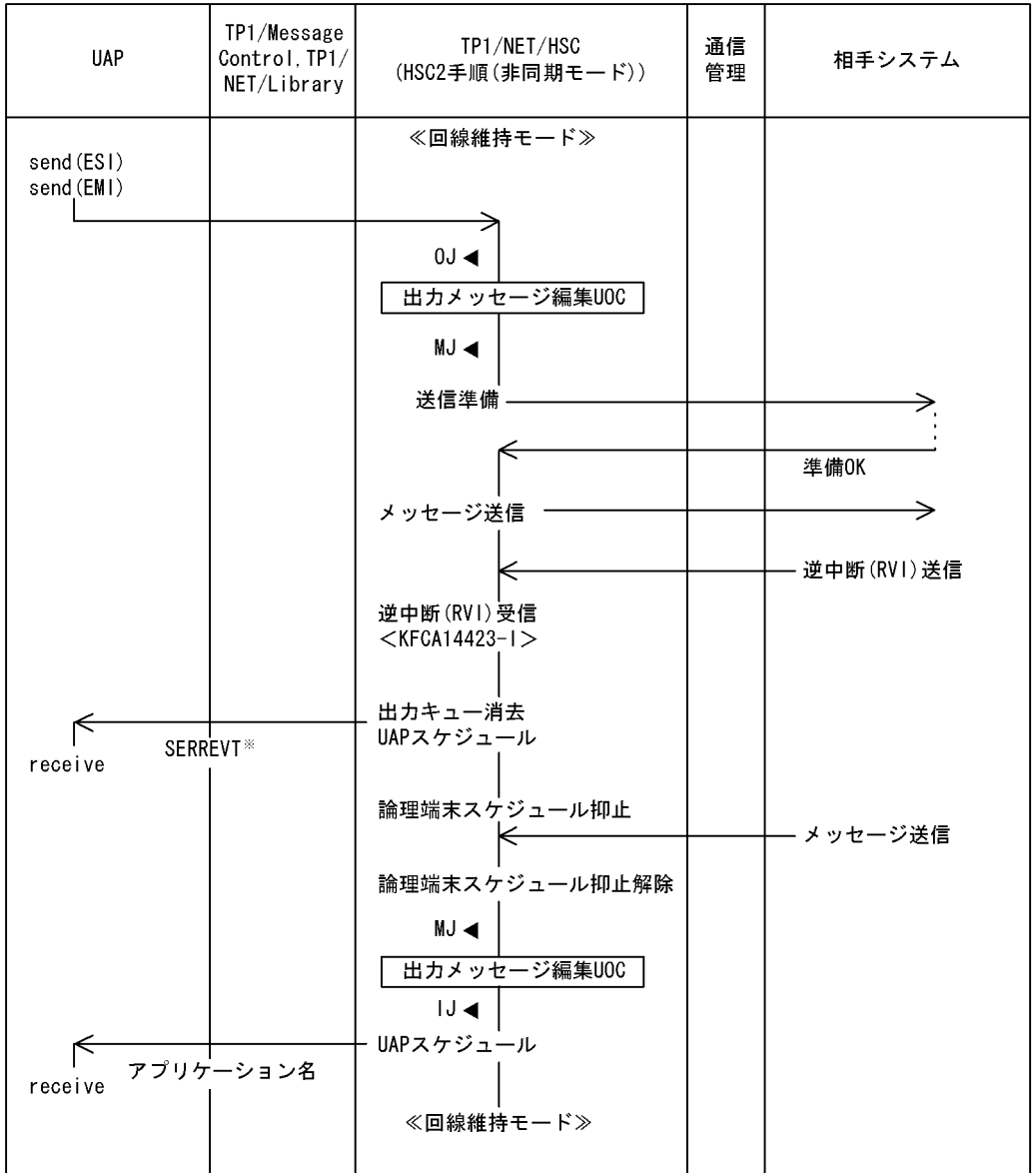


(凡例)

- 0J ◀: メッセージ出力ジャーナル取得
- MJ ◀: メッセージジャーナル取得
- IJ ◀: メッセージ送信完了ジャーナル取得

(4) 逆中断 (RVI) 受信

図 B-11 HSC2 手順 (非同期モード) の逆中断 (RVI) 受信



(凡例)

- 0J ◀: メッセージ出力ジャーナル取得
- MJ ◀: メッセージジャーナル取得
- IJ ◀: メッセージ送信完了ジャーナル取得

注

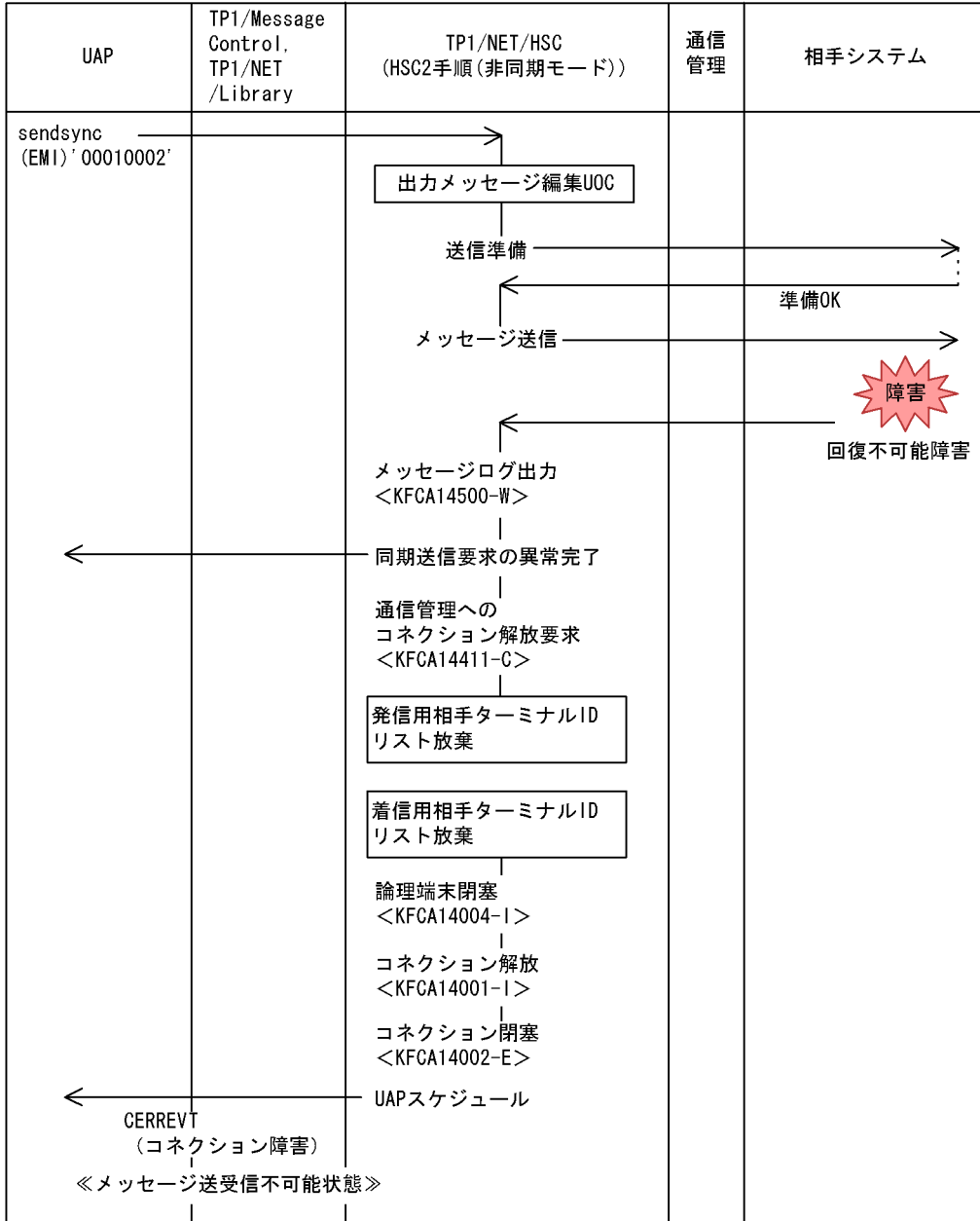
send 関数で送信障害通知イベントを通知させることを指定した場合

付録 B.3 HSC2 手順（同期モード）の障害

HSC2 手順（同期モード）の障害発生時の処理の流れを、図 B-12 ~ 図 B-16 に示します。

(1) メッセージ送信時の障害

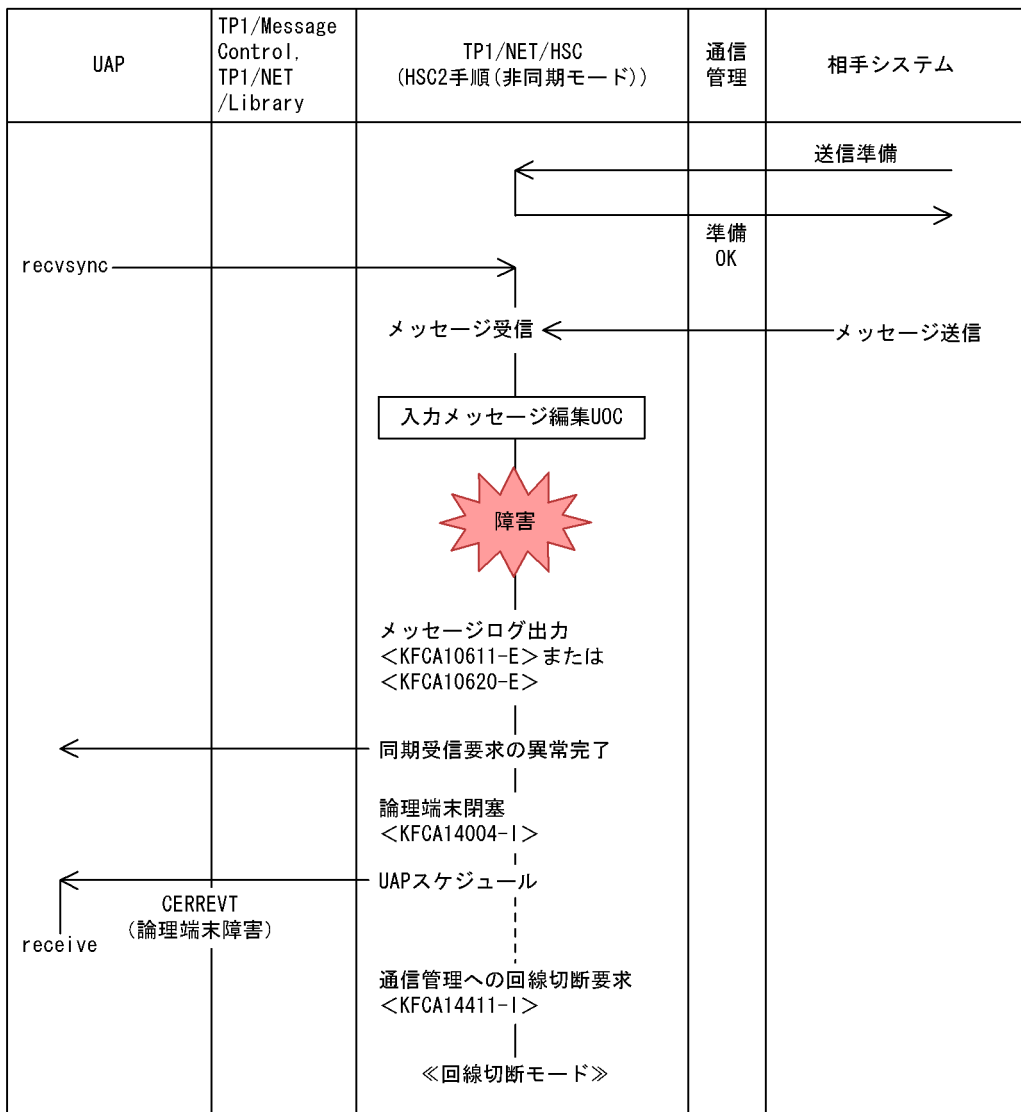
図 B-12 HSC2 手順（同期モード）のメッセージ送信時の回復不可能障害



(凡例) '00010002': 通常テキスト送信後, EOTを送信することを表すテキスト形態の数値表現

(2) 入力メッセージ編集 UOC 障害

図 B-14 HSC2 手順 (同期モード) の入力メッセージ編集 UOC 障害

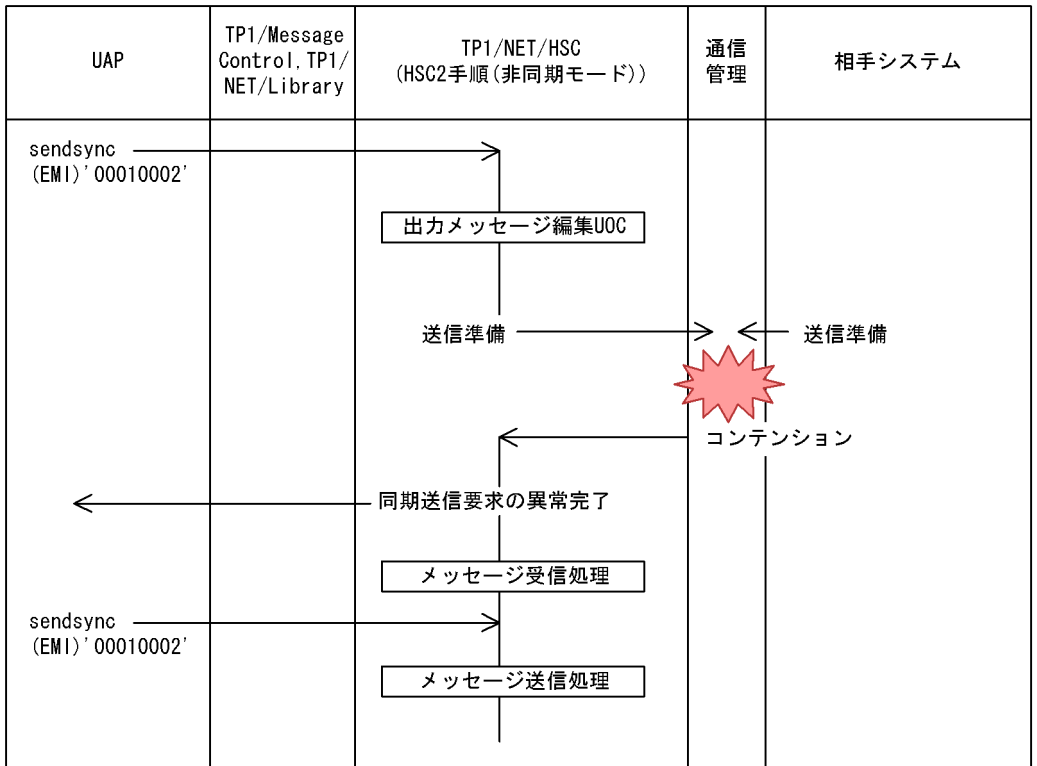


(凡例)

MJ ◀: メッセージジャーナル取得

(3) コンテンション

図 B-15 HSC2 手順 (同期モード) のコンテンション

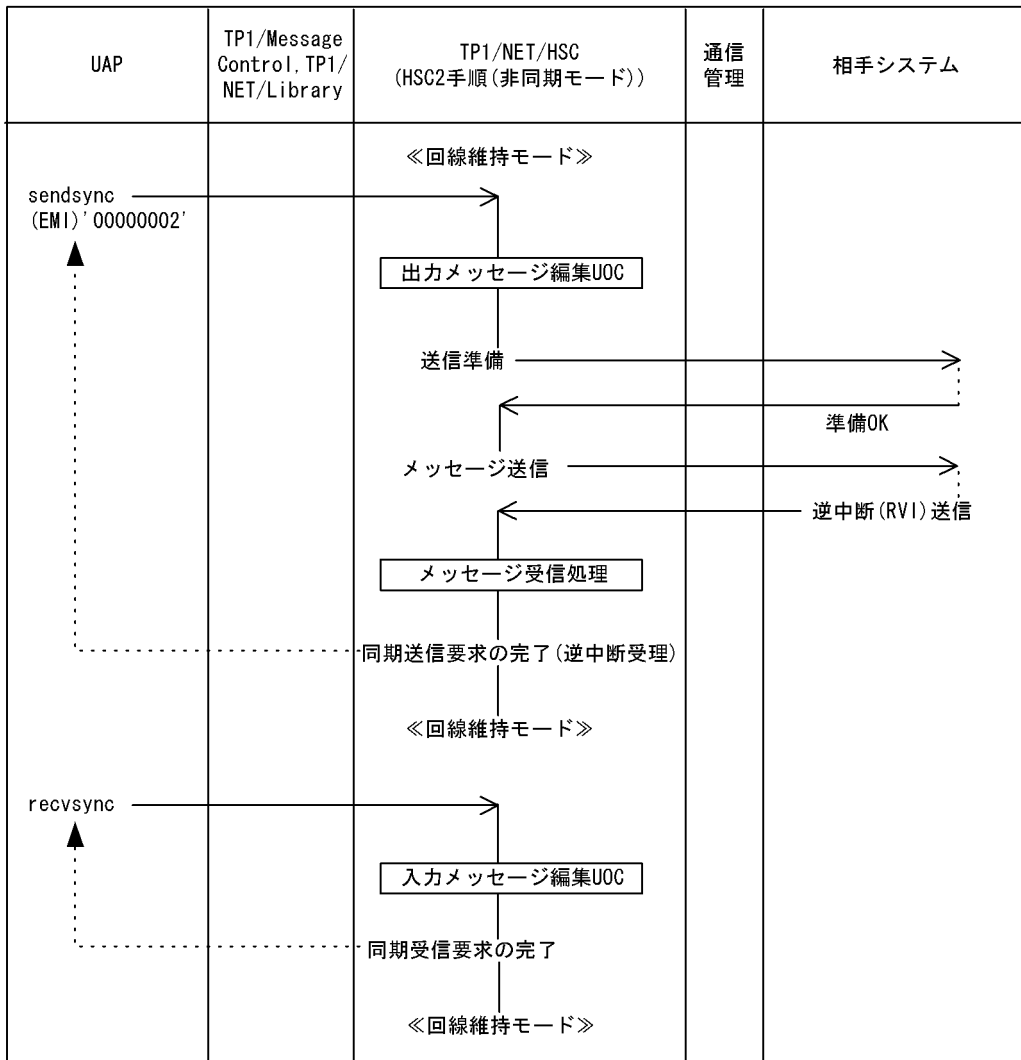


(凡例)

'00010002' : 通常テキスト送信後, EOTを送信することを表すテキスト形態の数値表現

(4) 逆中断 (RVI) 受信

図 B-16 HSC2 手順 (同期モード) の逆中断 (RVI) 受信 (mcftalccn コマンドの -u オプションで accept を指定した場合)



(凡例)

'00000002' : 通常テキストを送信することを表すテキスト形態の数値表現

付録 C 理由コード一覧

付録 C.1 ERREVT2 の理由コード一覧

ERREVT2 発生時の理由コードと、コードが示す ERREVT2 発生時の通知理由を次の表に示します。

表 C-1 ERREVT2 の理由コード一覧

C 言語の理由コード (16 進数字)	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 HSC1 手順の理由コード一覧

HSC1 手順の CERREVT の理由コードを示します。

HSC1 手順のコネクション障害の理由コードを表 C-2 に、論理端末障害の理由コードを表 C-3 に示します。

表 C-2 HSC1 手順のコネクション障害の理由コード一覧

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
DCMHSC_RSN1_MCF (00000001)(MCF 障害)	DCMHSC_RSN2_CNNC (00000000)	AP 間通信開始不可能, またはコ ネクション確立不可能
	DCMHSC_RSN2_CNER (00000001)	コネクション切断
	DCMHSC_RSN2_RBFOF (00000002)	受信バッファオーバーフロー
	DCMHSC_RSN2_ERROR (00000003)	MCF 内部論理矛盾
	DCMHSC_RSN2_LINE (00000009)	回線上の HSC 手順エラー
	DCMHSC_RSN2_RDISC (0000000a)	障害回復動作中止
	DCMHSC_RSN2_RBFST (0000000b)	受信バッファ不足

表 C-3 HSC1 手順の論理端末障害の理由コード一覧

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
DCMHSC_RSN1_MCF (00000001)(MCF 障害)	DCMHSC_RSN2_SBFOF (00000004)	送信バッファオーバーフロー
	DCMHSC_RSN2_OTGET (00000005)	出力キュー読み込み障害
	DCMHSC_RSN2_SBFST (00000006)	送信バッファ不足
	DCMHSC_RSN2_TXTER (00000007)	送信テキスト形態不正
	DCMHSC_RSN2_EOTR (00000008)	テキスト送信時 EOT 受信
DCMHSC_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
	DCMHSC_RSN2_BCNT (00000001)	使用バッファ数不正
	DCMHSC_RSN2_SEG (00000002)	有効セグメント不正
	DCMHSC_RSN2_BADR (00000003)	編集バッファリストアドレス不正

付録 C.3 HSC2 手順（非同期モード）の理由コード一覧

HSC2 手順（非同期モード）の理由コードを示します。

HSC2 手順（非同期モード）の接続障害の理由コードを表 C-4 に、論理端末障害の理由コードを表 C-5 に示します。

表 C-4 HSC2 手順（非同期モード）の接続障害の理由コード一覧

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
DCMHS2_RSN1_MCF (00000001)(MCF 障害)	DCMHS2_RSN2_CNNC (00000000)	接続確立不可能
	DCMHS2_RSN2_CNER (00000001)	接続切断
	DCMHS2_RSN2_RBFOF (00000002)	受信バッファオーバーフロー（テキスト受信時）
	DCMHS2_RSN2_ERROR (00000003)	MCF 内部論理矛盾
	DCMHS2_RSN2_RBBSY (00000007)	受信バッファ不足

表 C-5 HSC2 手順（非同期モード）の論理端末障害の理由コード一覧

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
DCMHS2_RSN1_MCF (00000001)(MCF 障害)	DCMHS2_RSN2_SBFOF (00000004)	送信バッファオーバーフロー
	DCMHS2_RSN2_OTGET (00000005)	出力キュー読み込み障害
	DCMHS2_RSN2_SBFST (00000006)	送信バッファ不足
	DCMHS2_RSN2_TXTER (00000009)	送信テキスト形態不正
	DCMHS2_RSN2_IDINV (0000000b)	相手ターミナル ID チェック不正
	DCMHS2_RSN2_IDREJ (0000000c)	発信要求時の接続拒否
	DCMHS2_RSN2_CALE (0000000d)	発信失敗
	DCMHS2_RSN2_DCTLE (00000017)	運用コマンド（mcfctdle）による明示閉塞
	DCMHS2_RSN2_SSEGOV (00000019)	送信最大セグメント長オーバー

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
	DCMHS2_RSN2_SOHTXT (0000001a)	ヘッダ付きテキスト受信
	DCMHS2_RSN2_SERRO (0000001b)	メッセージ送信中の回線回復不可能障害
	DCMHS2_RSN2_SLINE (0000001c)	メッセージ送信中の HSC 手順エラー
	DCMHS2_RSN2_SDISC (0000001d)	メッセージ送信中の DLE・EOT 受信
	DCMHS2_RSN2_SEOT (0000001e)	メッセージ送信中の EOT 受信
	DCMHS2_RSN2_RERRO (0000001f)	メッセージ受信中の回線回復不可能障害
	DCMHS2_RSN2_RLINE (00000020)	メッセージ受信中の HSC 手順エラー
	DCMHS2_RSN2_RDISC (00000021)	メッセージ受信中の DLE・EOT 受信
	DCMHS2_RSN2_REOT (00000022)	メッセージ受信中の EOT 受信
	DCMHS2_RSN2_DCTCN (00000023)	メッセージ送信中の運用コマンド (meftdetcn) 入力
	DCMHS2_RSN2_SDCTLE (00000024)	メッセージ送信中の運用コマンド (meftdctle) 入力
	DCMHS2_RSN2_NDISC (00000025)	回線維持終了
	DCMHS2_RSN2_NOTEL (00000027)	相手ダイヤル番号未定義
DCMHS2_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
	DCMHS2_RSN2_BCNT (00000001)	使用バッファ数不正
	DCMHS2_RSN2_SEG (00000002)	有効セグメント不正
	DCMHS2_RSN2_BADR (00000003)	編集バッファリストアドレス不正

付録 C.4 HSC2 手順 (同期モード) の理由コード一覧

HSC2 手順 (同期モード) の理由コードを示します。

HSC2 手順 (同期モード) のコネクション障害の理由コードを表 C-6 に、論理端末障害の理由コードを表 C-7 に示します。

表 C-6 HSC2 手順 (同期モード) のコネクション障害の理由コード一覧

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
DCMHS2_RSN1_MCF (00000001)(MCF 障害)	DCMHS2_RSN2_CNNC (00000000)	コネクション確立不可能
	DCMHS2_RSN2_CNER (00000001)	コネクション切断
	DCMHS2_RSN2_RBFOF (00000002)	受信バッファオーバーフロー (テキスト受信時)
	DCMHS2_RSN2_ERROR (00000003)	MCF 内部論理矛盾
	DCMHS2_RSN2_RBBSY (00000007)	受信バッファ不足
	DCMHS2_RSN2_RDERR (00000008)	発信, 着信用相手ターミナル ID リストファイル読み込み障害
	DCMHS2_RSN2_NOLEID (00000018)	論理端末未定義

表 C-7 HSC2 手順 (同期モード) の論理端末障害の理由コード一覧

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件
DCMHS2_RSN1_MCF (00000001)(MCF 障害)	DCMHS2_RSN2_SBFOF (00000004)	送信バッファオーバーフロー
	DCMHS2_RSN2_OTGET (00000005)	出力キュー読み込み障害
	DCMHS2_RSN2_SBFST (00000006)	送信バッファ不足
	DCMHS2_RSN2_TXTER (00000009)	送信テキスト形態不正
	DCMHS2_RSN2_DISC (0000000a)	テキスト送受信時処理中の DLE・EOT 受信
	DCMHS2_RSN2_IDINV (0000000b)	相手ターミナル ID チェック不正
	DCMHS2_RSN2_IDREJ (0000000c)	発信要求時の接続拒否
	DCMHS2_RSN2_CALE (0000000d)	発信失敗
	DCMHS2_RSN2_LINE (0000000e)	回線上の HSC 手順エラー
	DCMHS2_RSN2_APERR (0000000f)	テキスト送受信処理中の UAP プロセス終了

理由コード 1 (16進数字)	理由コード 2 (16進数字)	発生条件
	DCMHS2_RSN2_IDRCV (00000010)	発信要求時のコンテンツンション
	DCMHS2_RSN2_NOSTID (00000011)	発信要求時の相手ターミナル ID なし
	DCMHS2_RSN2_ACTINV (00000012)	分岐種別不正
	DCMHS2_RSN2_SEQERR (00000013)	シーケンス不正
	DCMHS2_RSN2_RBFOID (00000014)	受信バッファオーバーフロー (ID 交換時)
	DCMHS2_RSN2_APPMOV (00000015)	アプリケーション起動不可
	DCMHS2_RSN2_SYERR (00000016)	同期要求完了エラーリターン
	DCMHS2_RSN2_DCTLE (00000017)	運用コマンド (mcftdetle) による 明示閉塞
	DCMHS2_RSN2_SSEGOV (00000019)	送信最大セグメント長オーバー
	DCMHS2_RSN2_SOHTXT (0000001a)	ヘッダ付きテキスト受信
	DCMHS2_RSN2_TELER (00000026)	相手ダイヤル番号不正
DCMHS2_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
	DCMHS2_RSN2_BCNT (00000001)	使用バッファ数不正
	DCMHS2_RSN2_SEG (00000002)	有効セグメント不正
	DCMHS2_RSN2_BADR (00000003)	編集バッファリストアドレス不正

付録 D 関数使用の制限 (HSC2 手順 (同期モード))

HSC2 手順 (同期モード) の場合、ある関数のテキスト形態とリターン値によって、その次に呼び出せる関数の制限について、次の表に示します。

表 D-1 関数使用の制限

関数名	リターン値	次の関数名						
		recvsync	sendsync				send	
		-	1, 3	2, 4	5	6	7	
sendsync	1, 3	DCMCFRTN_00000	x					x
	2, 4				x			x
	5				x			x
	6		x	x	x	x		
	1, 2, 3, 4,	DCMCFRTN_73003		x	x	x		x
	1, 2, 3, 4	DCMCFRTN_77901				x		x
	5	DCMCFRTN_77902				x		x
	1, 2, 3, 4,	DCMCFRTN_77903		2	2	x		x
recvsync	1, 3 ¹	DCMCFRTN_00000		x	x	x		x
	5 ¹				x			x
	6 ¹		x	x	x	x	x	
	-	DCMCFRTN_77900	x	x	x	x	x	
receive	8	DCMCFRTN_00000	x					x
	9			x	x	x		x

(凡例)

- : 呼び出しができます。
- x : 呼び出しができません。
- : ありません。

1 から 9 はテキスト形態です。

- 1: 通常テキストの送信または受信をします。
- 2: 通常テキスト送信後, EOT 送信をします。
- 3: 透過モードテキストの送信または受信をします。
- 4: 透過モードテキスト送信後, EOT 送信をします。
- 5: EOT 送信または受信をします。
- 6: DLE・EOT の送信または受信をします。
- 7: 回線接続要求をします。
- 8: 発信による回線接続をします。
- 9: 着信による回線接続をします。

注 1

recvsync 関数がリターン値 DCMCFRTN_00000 で終了したときに通知されるテキスト形態です。

注 2

sendsync 関数を呼び出せますが, リターン値 DCMCFRTN_73003 で終了する場合がありますので, 注意してください。

付録 E 論理端末名称決定 UOC のコーディング例

論理端末名称決定 UOC のコーディング例 (K & R 版 C) を次の図に示します。また、このコーディング例を、/BeTRAN/examples/mcf/HSC/cmlib/c/uoc.c のファイルで提供しています。

```

/*
 * TP1/NET/HSC  HSC2手順 (非同期モード)  論理端末名称決定UOC
 */
#include <stdio.h>
#include <dcmcf.h>
#include <dcmcfuoc.h>
#include <dcmpcm.h>
#include <dcmhsc2.h>

#define NOLE -19000          /* 未使用論理端末なし */

long ledtmn01(parm)
dcmcf_uoc_min_n *parm;
{
    long rtn_code ,
        count ;
    dcmhs3_uoc_prot *le_ptr ;

    le_ptr = (dcmhs2_uoc_prot *)parm->pro_indv_ifa ;
    count = 0 ;
    rtn_code = DCMCF_UOC_MSG_NG ;
    parm->rtn_detail = NOLE ;

    /***** 未使用論理端末検索 *****/

    while(le_ptr->le_num > count) {
        if(le_ptr->le_list[count].le_inf == DCMHS2_LE_NOUSE){
            strcpy(parm->le_name,le_ptr->le_list[count].le_name);
            rtn_code = DCMCF_UOC_MSG_OK ;
            parm->rtn_detail = 0 ;
            break ;
        }
        count++ ;
    }
    return(rtn_code);
}

```

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

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

GJ (メッセージ受信ジャーナル)

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

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

IJ (メッセージ入力ジャーナル)

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

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

MJ (メッセージジャーナル)

メッセージに関する情報です。端末名称、メッセージ種別、入力メッセージ編集前のデータ、出力メッセージ編集後のデータなどで構成されます。

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

OJ (メッセージ出力ジャーナル)

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

OJ の取得タイミングは UAP から send を受け付けたときです。

(力行)

コネクション

通信管理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送受信をします。

(サ行)

出力キュー

処理の結果として出力メッセージを管理する待ち行列です。

(ナ行)

入力キュー

処理要求として入力メッセージを管理する待ち行列です。

(ラ行)

論理端末

TP1/NET/HSC と UAP との通信接点です。TP1/NET/HSC と UAP は、論理端末単位にメッセージの送受信をします。

索引

A

AJ (メッセージ送信完了ジャーナル) 402
AP 間通信 2
AP 間通信〔HSC1 手順〕 14
AP 間通信〔HSC2 手順 (同期モード)〕 36
AP 間通信〔HSC2 手順 (非同期モード)〕 28
AP 間通信の概要 2
AP 間通信の形態 3
AP 間通信の終了 14
AP 間通信の準備 13

C

CBLDCMCF('RECEIVE') 92
CBLDCMCF('RECVSYNC') 98
CBLDCMCF('RESEND') 104
CBLDCMCF('SEND') 110
CBLDCMCF('SENDSYNC') 116
CCLSEVT 185, 194
CERREVT 183, 193
COBOL 言語のメッセージ送受信 69
COPNEVT 185, 194
C 言語のメッセージ送受信 69

D

dc_mcf_receive 71
dc_mcf_recvsync 75
dc_mcf_resend 79
dc_mcf_send 83
dc_mcf_sendsync 88

E

EOT 交換 57, 60
EOT 交換の抑止〔HSC2 手順 (同期モード)〕 65
EOT 交換の抑止〔HSC2 手順 (非同期モード)〕 63
ERREVT1 179, 185
ERREVT2 180, 186

ERREVT2 の理由コード一覧 393
ERREVT3 180, 188
ERREVT4 181, 189

G

GJ (メッセージ受信ジャーナル) 402

H

HSC1 手順 2
HSC1 手順の AP 間通信の仕組み 12
HSC1 手順の AP 間通信メッセージの送受信 22
HSC1 手順の機能 11, 24
HSC1 手順の機能と TP1/NET/HSC の扱い 24
HSC1 手順のコネクション障害の理由コード一覧 394
HSC1 手順の障害 377
HSC1 手順の送受信の方法 12
HSC1 手順の通信形態 3
HSC1 手順のメッセージ受信の処理の流れ (複数セグメントの場合)〔HSC1 手順〕 358
HSC1 手順のメッセージ受信の流れ 23
HSC1 手順のメッセージ送信の処理の流れ (複数セグメントの場合)〔HSC1 手順〕 355
HSC1 手順のメッセージ送信の流れ 22
HSC1 手順の論理端末障害の理由コード一覧 394
HSC2 呼制御 57
HSC2 手順 2
HSC2 手順 (同期モード) のコネクション障害の理由コード一覧 397
HSC2 手順 (同期モード) の障害 388
HSC2 手順 (同期モード) の送受信の方法 30
HSC2 手順 (同期モード) の通信形態 4
HSC2 手順 (同期モード) の論理端末障害の理由コード一覧 397
HSC2 手順 (同期モード) プロトコル障害 339

HSC2 手順 (同期モード) を使用した通信形態 4
 HSC2 手順 (非同期モード) のコネクション障害の理由コード一覧 395
 HSC2 手順 (非同期モード) の障害 382
 HSC2 手順 (非同期モード) の送受信の方法 26
 HSC2 手順 (非同期モード) の通信形態 4
 HSC2 手順 (非同期モード) の論理端末障害の理由コード一覧 395
 HSC2 手順 (非同期モード) プロトコル障害 317
 HSC2 手順の AP 間通信の仕組み 26
 HSC2 手順の AP 間通信メッセージの送受信 50
 HSC2 手順の機能 25, 57
 HSC 手順障害 352

I

IJ (メッセージ入力ジャーナル) 402

M

max_open_fds 252
 max_socket_descriptors 251
 MCF 8
 mcftactcn 271
 mcftactle 273
 mcftalccn [HSC1 手順] 202
 mcftalccn [HSC2 手順 (同期モード)] 236
 mcftalccn [HSC2 手順 (非同期モード)] 215
 mcftalced [HSC1 手順] 210
 mcftalced [HSC2 手順 (同期モード)] 245
 mcftalced [HSC2 手順 (非同期モード)] 227
 mcftalcle [HSC1 手順] 207
 mcftalcle [HSC2 手順 (同期モード)] 243
 mcftalcle [HSC2 手順 (非同期モード)] 223
 mcftdctcn 275
 mcftdctle 278
 mcfteof 249

mcftleof 230
 mcftllst 229
 mcftlnf 228
 mcftlscn 281
 mcftlsle 284
 mcfttrlst 248
 mcftstf 246
 mcftstlst 247
 MCF アプリケーション定義 197
 MCF イベント 176
 MCF イベントインタフェース 176
 MCF イベント情報の形式 (COBOL 言語) 185
 MCF イベント情報の形式 (C 言語) 178
 MCF イベント処理用 MHP 178
 MCF イベント通知時のセグメント構成 177
 MCF イベントの共通ヘッダ 178
 MCF イベントの種類 176
 MCF サービス名の登録 288
 MCF 通信構成定義 197
 MCF 通信プロセスでアクセスするファイルの最大数 252
 MCF 定義オブジェクトの生成 253
 MCF で使用する定義ファイル 197
 MCF の障害 [HSC1 手順] 304
 MCF の障害 [HSC2 手順 (同期モード)] 347
 MCF の障害 [HSC2 手順 (非同期モード)] 325
 MCF マネージャ定義 197
 MCF メイン関数の作成 288, 289
 MHP 終了での切断 36
 MJ (メッセージジャーナル) 402

O

OJ (メッセージ出力ジャーナル) 402

P

PRC 障害 347

R

RECEIVE 122

S

SCMPEVT 181, 190
 SEND 126
 sendsync 関数・recvsync 関数のリターン障
 害 347
 SERREVT 182, 191

T

TP1/Message Control 8
 TP1/NET/HSC が通知する MCF イベントの
 種類 176
 TP1/NET/HSC 固有のシステム定義の種類
 [HSC1 手順] 199
 TP1/NET/HSC 固有のシステム定義の種類
 [HSC2 手順 (同期モード)] 231
 TP1/NET/HSC 固有のシステム定義の種類
 [HSC2 手順 (非同期モード)] 211
 TP1/NET/HSC の運用コマンド 270
 TP1/NET/HSC の組み込みの流れ 288
 TP1/NET/HSC の定義の概要 197
 TP1/NET/HSC を組み込んだソフトウェア構
 成 9
 TP1/NET/HSC を使用したネットワーク構成
 例 2
 TP1/NET/Library 8

U

UAP 異常終了通知イベント 176
 UAP および通信管理でのデータの扱い
 (HSC1 手順) 20
 UAP および通信管理でのデータの扱い
 (HSC2 手順) 47
 UAP 障害 [HSC1 手順] 304
 UAP 障害 [HSC2 手順 (同期モード)] 346
 UAP 障害 [HSC2 手順 (非同期モード)]
 324
 UOC インタフェース用のパラメタとバッ
 ファの関係 167
 UOC インタフェース用のパラメタとバッ
 ファの関係 (論理端末名称決定 UOC の場
 合) 159
 UOC 作成上の注意事項 174

UOC で使用できる関数 174
 UOC の異常処理 175
 UOC の構造 174
 UOC の実行タイミング 175
 UOC の障害処理 154

あ

相手システムからの切断 [HSC2 手順 (同期
 モード)] 36
 相手システムからの切断 [HSC2 手順 (非同
 期モード)] 29
 相手ターミナル ID リスト定義 255
 相手ターミナル ID リスト定義ユティリティ
 (同期モード) 59
 アプリケーションの型 [HSC1 手順] 15
 アプリケーションの型 [HSC2 手順] 42
 アプリケーション名 [HSC1 手順] 15
 アプリケーション名 [HSC2 手順] 42
 アプリケーション名の決定 161
 アプリケーション名の決定の処理 162
 暗黙閉塞 [HSC1 手順] 16
 暗黙閉塞 [HSC2 手順] 43

い

一時受信抑止 (WACK) [HSC1 手順] 24
 一時受信抑止 (WACK) [HSC2 手順] 57
 一時送信延期 (TTD) [HSC1 手順] 24
 一時送信延期 (TTD) [HSC2 手順] 57
 一方受信形態 [HSC1 手順] 3
 一方受信形態 [HSC2 手順 (同期モード)] 5
 一方送信形態の AP 間通信の例 (HSC1 手順
 の場合) 3
 一方送信メッセージ 3
 一方送信メッセージまたは回線接続要求の送
 信 (COBOL 言語) 110
 一方送信メッセージまたは回線接続要求の送
 信 (C 言語) 83

う

運用コマンド 269

か

回線維持モード 50
 回線接続要求テキスト 48
 回線切断の処理の流れ (MHP の終了の場合)〔HSC2 手順 (同期モード)〕 375
 回線切断の処理の流れ (回線維持タイムアウトの場合)〔HSC2 手順 (非同期モード)〕 366
 回線切断の処理の流れ (回線切断の受信の場合)〔HSC2 手順 (同期モード)〕 376
 回線切断の処理の流れ (回線切断要求 (DLE・EOT) 送信の場合)〔HSC2 手順 (同期モード)〕 375
 回線の接続〔HSC2 手順 (同期モード)〕 33
 回線の接続〔HSC2 手順 (非同期モード)〕 27
 回線の切断〔HSC2 手順 (同期モード)〕 36
 回線の切断〔HSC2 手順 (非同期モード)〕 28
 回線リスト出力ファイル定義 228
 回線リスト出力ファイル定義の終了 230
 回線リスト定義 229, 254
 回線リスト定義ユティリティ (非同期モード) 58
 概要 1
 関数使用の制限 (HSC2 手順 (同期モード)) 399

き

逆中断 (RVI)〔HSC1 手順〕 24
 逆中断 (RVI)〔HSC2 手順〕 57
 逆中断 (RVI) 受信〔HSC1 手順〕 382
 逆中断 (RVI) 受信〔HSC2 手順 (同期モード)〕 392
 逆中断 (RVI) 受信〔HSC2 手順 (非同期モード)〕 387
 キューグループ ID〔HSC1 手順〕 208
 キューグループ ID〔HSC2 手順 (非同期モード)〕 224

く

組み込み方法 287

こ

コネクション 402
 コネクション ID〔HSC1 手順〕 202
 コネクション ID〔HSC2 手順 (同期モード)〕 236
 コネクション ID〔HSC2 手順 (非同期モード)〕 215
 コネクション障害〔HSC1 手順〕 298, 305
 コネクション障害〔HSC2 手順 (同期モード)〕 348
 コネクション障害〔HSC2 手順 (非同期モード)〕 326
 コネクション切断 377
 コネクション切断後の再確立〔HSC2 手順 (同期モード)〕 37
 コネクション切断後の再確立〔HSC2 手順 (非同期モード)〕 30
 コネクション定義の開始〔HSC1 手順〕 202
 コネクション定義の開始〔HSC2 手順 (同期モード)〕 236
 コネクション定義の開始〔HSC2 手順 (非同期モード)〕 215
 コネクション定義の終了〔HSC1 手順〕 210
 コネクション定義の終了〔HSC2 手順 (同期モード)〕 245
 コネクション定義の終了〔HSC2 手順 (非同期モード)〕 227
 コネクションと論理端末の関係〔HSC1 手順〕 14
 コネクションと論理端末の関係〔HSC2 手順〕 38
 コネクションの解放 275
 コネクションの解放〔HSC2 手順 (同期モード)〕 36
 コネクションの解放〔HSC2 手順 (非同期モード)〕 29
 コネクションの確立 271
 コネクションの確立〔HSC2 手順 (同期モード)〕 31
 コネクションの確立〔HSC2 手順 (非同期モード)〕 26
 コネクションの自動回復の抑止〔HSC1 手順〕 14

コネクションの状態表示 281
 コンテンション〔HSC1 手順〕381
 コンテンション〔HSC2 手順(同期モード)〕391
 コンテンション〔HSC2 手順(非同期モード)〕386
 コンテンション制御〔HSC1 手順〕24
 コンテンション制御〔HSC2 手順〕57

し

自システムからの切断 29
 システム構成例 (HSC1 手順) 257
 システム構成例 (HSC 2 手順(同期モード)) 259
 システム構成例 (HSC 2 手順(非同期モード)) 258
 システムサービス共通情報定義 251
 システムサービス情報定義 250
 システムサービス情報定義ファイルの作成 288
 システム定義 195
 自動解除〔HSC1 手順〕16
 自動解除〔HSC2 手順〕44
 自動解放〔HSC2 手順(同期モード)〕36
 自動解放〔HSC2 手順(非同期モード)〕29
 自動確立〔HSC2 手順(同期モード)〕31
 自動確立〔HSC2 手順(非同期モード)〕26
 自動閉塞(暗黙閉塞)〔HSC1 手順〕16
 自動閉塞(暗黙閉塞)〔HSC2 手順〕44
 受信スケジュール関係(入力キュー, 入力メッセージ編集 UOC)の障害〔HSC1 手順〕300
 受信スケジュール関係(入力キュー, 入力メッセージ編集 UOC)の障害〔HSC2 手順(同期モード)〕343
 受信スケジュール関係(入力キュー, 入力メッセージ編集 UOC)の障害〔HSC2 手順(非同期モード)〕320
 受信するテキスト形態の数値表現と内容〔HSC1 手順〕21
 受信するテキスト形態の数値表現と内容〔HSC2 手順〕49
 受信モード 50

出力キュー 402
 出力キュー障害〔HSC1 手順〕312
 出力キュー障害〔HSC2 手順(非同期モード)〕333
 出力キュー読み込み障害 379
 出力ファイル定義 246
 出力ファイル定義の終了 249
 出力メッセージの編集 168
 出力メッセージの編集 UOC 168
 出力メッセージの割り当て先〔HSC1 手順〕208
 出力メッセージの割り当て先〔HSC2 手順(同期モード)〕244
 出力メッセージの割り当て先〔HSC2 手順(非同期モード)〕224
 出力メッセージ編集 UOC インタフェース 169
 手動解除〔HSC1 手順〕17
 手動解除〔HSC2 手順〕44
 手動解放(強制解放)〔HSC2(非同期モード)〕29
 手動解放(正常解放)〔HSC2(非同期モード)〕29
 手動解放〔HSC2 手順(同期モード)〕37
 手動確立〔HSC2 手順(同期モード)〕32
 手動確立〔HSC2 手順(非同期モード)〕26
 手動閉塞(明示閉塞)〔HSC1 手順〕16
 手動閉塞(明示閉塞)〔HSC2 手順〕44
 障害対策〔HSC1 手順〕297
 障害対策〔HSC2 手順(同期モード)〕335
 障害対策〔HSC2 手順(非同期モード)〕313
 障害通知イベント 177
 障害の種類と対応処理〔HSC1 手順〕298
 障害の種類と対応処理〔HSC2 手順(同期モード)〕336
 障害の種類と対応処理〔HSC2 手順(非同期モード)〕314
 障害発生時の処理の流れ 377
 障害閉塞(暗黙閉塞)〔HSC1 手順〕16
 障害閉塞(暗黙閉塞)〔HSC2 手順〕44
 状態通知イベント 177
 処理層とデータの関係 (HSC1 手順) 18
 処理層とデータの関係 (HSC2 手順) 45

せ

制御付き会話〔HSC1 手順〕 24
 制御付き会話〔HSC2 手順〕 57
 接続情報通知テキスト 48

そ

送受信反転制御 367
 送信完了通知イベント 177
 送信障害通知イベント 177
 送信スケジュール可能〔HSC1 手順〕 16
 送信スケジュール関係（出力キュー，出力メッセージ編集 UOC）の障害〔HSC1 手順〕 302
 送信スケジュール関係（出力キュー，出力メッセージ編集 UOC）の障害〔HSC2 手順（同期モード）〕 344
 送信スケジュール関係（出力キュー，出力メッセージ編集 UOC）の障害〔HSC2 手順（非同期モード）〕 322
 送信スケジュール抑止〔HSC1 手順〕 16
 送信するテキスト形態の数値表現と内容（HSC1 手順） 21
 送信するテキスト形態の数値表現と内容（HSC2 手順） 49
 送信メッセージの通番編集 172
 送信メッセージの通番編集 UOC インタフェース 173
 送信モード 50
 ソケット用ファイル記述子の最大数 251
 ソフトウェアの構成 8

た

タイマ起動メッセージ廃棄通知イベント 176
 端末タイプ〔HSC1 手順〕 207
 端末タイプ〔HSC2 手順（同期モード）〕 243
 端末タイプ〔HSC2 手順（非同期モード）〕 223

ち

着信の場合の回線の接続〔HSC2 手順（同期モード）〕 35

着信の場合の回線の接続〔HSC2 手順（非同期モード）〕 28
 着信の場合の通信形態の組み合わせ 7
 着信用相手ターミナル ID リスト定義 248
 着信用相手ターミナル ID リストファイル 31

つ

通常テキスト〔HSC1 手順〕 20
 通常テキスト〔HSC2 手順〕 47
 通信管理との定義の関係（HSC1 手順） 257
 通信管理との定義の関係（HSC2 手順（同期モード）） 259
 通信管理との定義の関係（HSC2 手順（非同期モード）） 257
 通信管理プログラムと関連づける内容 257
 通信形態の組み合わせ 5
 通信形態の種類 4
 通信障害〔HSC2 手順（同期モード）〕 336
 通信障害〔HSC2 手順（非同期モード）〕 314

て

定義オブジェクトファイルの生成 288, 293
 定義の指定順序〔HSC1 手順〕 201
 定義の指定順序〔HSC2 手順（同期モード）〕 234
 定義の指定順序〔HSC2 手順（非同期モード）〕 214
 定義例〔HSC1 手順〕 261
 定義例〔HSC2 手順（同期モード）〕 265
 定義例〔HSC2 手順（非同期モード）〕 263
 ディスク出力メッセージ最大格納数〔HSC1 手順〕 207
 ディスク出力メッセージ最大格納数〔HSC2 手順（非同期モード）〕 224
 データ操作言語（COBOL 言語）のメッセージ送受信 70
 テキスト形態と UAP でのデータの扱い〔HSC1 手順〕 18
 テキスト形態と UAP でのデータの扱い〔HSC2 手順〕 46
 テキスト形態の種類（HSC1 手順） 19

テキスト形態の種別〔HSC2 手順〕 47
適用範囲 7

と

透過モードテキスト〔HSC1 手順〕 20
透過モードテキスト〔HSC2 手順〕 48
同期型のメッセージの受信〔COBOL 言語〕 98
同期型のメッセージの受信〔C 言語〕 75
同期型のメッセージの送信〔COBOL 言語〕 116
同期型のメッセージの送信〔C 言語〕 88
同期受信形態〔HSC2 手順（同期モード）〕 5
同期送信形態〔HSC2 手順（同期モード）〕 5
同期モードの EOT 交換 63
同期モードのメッセージの受信 54
同期モードのメッセージの送信 52

に

入力キュー 403
入力キュー障害〔HSC1 手順〕 311
入力キュー障害〔HSC2 手順（非同期モード）〕 332
入力メッセージの編集 160
入力メッセージの編集とアプリケーション名の決定 160
入力メッセージ編集 UOC 160
入力メッセージ編集 UOC インタフェース 163
入力メッセージ編集 UOC 障害〔HSC1 手順〕 380
入力メッセージ編集 UOC 障害〔HSC2 手順（同期モード）〕 390
入力メッセージ編集 UOC 障害〔HSC2 手順（非同期モード）〕 385

は

発信の場合の回線の接続〔HSC2 手順（同期モード）〕 33
発信の場合の回線の接続〔HSC2 手順（非同期モード）〕 27
発信の場合の通信形態の組み合わせ 6

発信相手ターミナル ID リスト定義 247
発信相手ターミナル ID リストファイル 31

ひ

非同期送信スケジュール可能 43
非同期送信スケジュール抑止 43
非同期モードの EOT 交換 60
非同期モードのメッセージの受信 53
非同期モードのメッセージの送信 50

ふ

不正アプリケーション名検出通知イベント 176
ブロック転送〔HSC1 手順〕 24
ブロック転送〔HSC2 手順〕 57
分岐送信形態〔HSC1 手順〕 3
分岐送信形態〔HSC2 手順（同期モード）〕 4
分岐送信形態の AP 間通信の例〔HSC1 手順の場合〕 3

へ

閉塞解除状態〔HSC1 手順〕 16
閉塞解除状態〔HSC2 手順〕 43
閉塞状態〔HSC1 手順〕 16
閉塞状態〔HSC2 手順〕 43
ヘッダ付きテキスト〔HSC1 手順〕 20
ヘッダ付き透過モードテキスト〔HSC1 手順〕 20

み

未処理送信メッセージ廃棄通知イベント 177

め

明示閉塞〔HSC1 手順〕 16
明示閉塞〔HSC2 手順〕 43
メッセージ受信時のコネクション障害〔HSC1 手順〕 306
メッセージ受信時のコネクション障害〔HSC2 手順（非同期モード）〕 327

- メッセージ受信の処理の流れ (単一セグメントの場合) [HSC1 手順] 357
- メッセージ受信の処理の流れ (単一セグメントの場合) [HSC2 手順 (同期モード)] 372
- メッセージ受信の処理の流れ (単一セグメントの場合) [HSC2 手順 (非同期モード)] 363
- メッセージ受信の処理の流れ (複数セグメントの場合) [HSC2 手順 (同期モード)] 373
- メッセージ受信の処理の流れ (複数セグメントの場合) [HSC2 手順 (非同期モード)] 364
- メッセージ受信の処理の流れ (列信の場合) [HSC1 手順] 359
- メッセージ受信の処理の流れ (列信の場合) [HSC2 手順 (同期モード)] 374
- メッセージ受信の処理の流れ (列信の場合) [HSC2 手順 (非同期モード)] 365
- メッセージ受信用バッファグループ番号 [HSC1 手順] 203
- メッセージ受信用バッファグループ番号 [HSC2 手順 (同期モード)] 237
- メッセージ受信用バッファグループ番号 [HSC2 手順 (非同期モード)] 216
- メッセージ制御機能 8
- メッセージ送受信インタフェース 67
- メッセージ送受信インタフェースの一覧 69
- メッセージ送受信の処理の流れ 354
- メッセージ送受信の通信文 (COBOL 言語) 70
- メッセージ送受信の通信文 (データ操作言語) 70
- メッセージ送受信のライブラリ関数 (C 言語) 69
- メッセージ送信時の HSC 手順エラー [HSC2 手順 (同期モード)] 389
- メッセージ送信時の HSC 手順エラー [HSC2 手順 (非同期モード)] 384
- メッセージ送信時の回復不可能障害 [HSC2 手順 (同期モード)] 388
- メッセージ送信時の回復不可能障害 [HSC2 手順 (非同期モード)] 383
- メッセージ送信時のコネクション障害 [HSC1 手順] 305
- メッセージ送信時のコネクション障害 [HSC2 手順 (非同期モード)] 326
- メッセージ送信時の障害 [HSC1 手順] 378
- メッセージ送信時の切断 36
- メッセージ送信の処理の流れ (単一セグメントで通常テキスト送信後, EOT を送信する場合) [HSC2 手順 (同期モード)] 368
- メッセージ送信の処理の流れ (単一セグメントで通常テキスト送信する場合) [HSC2 手順 (同期モード)] 369
- メッセージ送信の処理の流れ (単一セグメントの場合) [HSC1 手順] 354
- メッセージ送信の処理の流れ (単一セグメントの場合) [HSC2 手順 (非同期モード)] 360
- メッセージ送信の処理の流れ (複数セグメントの場合) [HSC2 手順 (同期モード)] 370
- メッセージ送信の処理の流れ (複数セグメントの場合) [HSC2 手順 (非同期モード)] 361
- メッセージ送信の処理の流れ (列信の場合) [HSC1 手順] 356
- メッセージ送信の処理の流れ (列信の場合) [HSC2 手順 (同期モード)] 371
- メッセージ送信の処理の流れ (列信の場合) [HSC2 手順 (非同期モード)] 362
- メッセージ送信用バッファグループ番号 [HSC1 手順] 202
- メッセージ送信用バッファグループ番号 [HSC2 手順 (同期モード)] 237
- メッセージ送信用バッファグループ番号 [HSC2 手順 (非同期モード)] 216
- メッセージの再送 (COBOL 言語) 104
- メッセージの再送 (C 言語) 79
- メッセージの受信 (データ操作言語) 122
- メッセージの受信 [HSC1 手順] 23
- メッセージの受信 [HSC2 手順] 53
- メッセージの送受信 [HSC1 手順] 354
- メッセージの送受信 [HSC2 手順 (同期モード)] 367

メッセージの送受信〔HSC2 手順（非同期モード）〕 360
 メッセージの送信（データ操作言語） 126
 メッセージの送信〔HSC1 手順〕 22
 メッセージの送信〔HSC2 手順〕 50
 メッセージの分割と組み立て〔HSC1 手順〕 17
 メッセージの分割と組み立て〔HSC2 手順〕 44
 メッセージ廃棄通知イベント 176
 メッセージ編集用バッファグループ番号〔HSC1 手順〕 203
 メッセージ編集用バッファグループ番号〔HSC2 手順（同期モード）〕 237
 メッセージ編集用バッファグループ番号〔HSC2 手順（非同期モード）〕 216
 メッセージ編集用バッファ数〔HSC1 手順〕 203
 メッセージ編集用バッファ数〔HSC2 手順（同期モード）〕 237
 メッセージ編集用バッファ数〔HSC2 手順（非同期モード）〕 216
 メッセージまたは接続情報通知の受信（C 言語） 71
 メモリ出力メッセージ最大格納数〔HSC1 手順〕 207
 メモリ出力メッセージ最大格納数〔HSC2 手順（同期モード）〕 243
 メモリ出力メッセージ最大格納数〔HSC2 手順（非同期モード）〕 223

ゆ

ユーザアプリケーションプログラム異常終了〔HSC1 手順〕 308
 ユーザアプリケーションプログラム異常終了〔HSC2 手順（同期モード）〕 349
 ユーザアプリケーションプログラム異常終了〔HSC2 手順（非同期モード）〕 329
 ユーザアプリケーションプログラム作成例 132
 ユーザアプリケーションプログラム閉塞〔HSC1 手順〕 310

ユーザアプリケーションプログラム閉塞〔HSC2 手順（同期モード）〕 351
 ユーザアプリケーションプログラム閉塞〔HSC2 手順（非同期モード）〕 331
 ユーザOWNコーディング 154
 ユーザOWNコーディング，MCF イベントインタフェース 153
 ユーザOWNコーディングインタフェース 154

よ

用語解説 402

り

理由コード一覧 393

れ

列信〔HSC1 手順〕 24
 列信〔HSC2 手順〕 57

ろ

論理端末 403
 論理端末定義〔HSC1 手順〕 207
 論理端末定義〔HSC2 手順（同期モード）〕 243
 論理端末定義〔HSC2 手順（非同期モード）〕 223
 論理端末とアプリケーションの型の関係〔HSC1 手順〕 15
 論理端末とアプリケーションの型の関係〔HSC2 手順〕 41
 論理端末の状態表示 284
 論理端末の属性 42
 論理端末の端末タイプ，メッセージの種類，アプリケーションの型，UAP インタフェース，および通信形態の関係〔HSC1 手順〕 15
 論理端末の端末タイプ，メッセージの種類，アプリケーションの型，UAP インタフェース，および通信形態の関係〔HSC2 手順〕 43
 論理端末の端末タイプ〔HSC1 手順〕 15
 論理端末の端末タイプ〔HSC2 手順〕 41

- 論理端末の閉塞 278
- 論理端末の閉塞〔HSC1 手順〕 16
- 論理端末の閉塞〔HSC2 手順〕 44
- 論理端末の閉塞解除 273
- 論理端末の閉塞解除〔HSC1 手順〕 16
- 論理端末の閉塞解除〔HSC2 手順〕 44
- 論理端末の閉塞と閉塞解除〔HSC1 手順〕 15
- 論理端末の閉塞と閉塞解除〔HSC2 手順〕 43
- 論理端末名称〔HSC1 手順〕 207
- 論理端末名称〔HSC2 手順(同期モード)〕
243
- 論理端末名称〔HSC2 手順(非同期モード)〕
223
- 論理端末名称決定 UOC インタフェース 155
- 論理端末名称決定 UOC のコーディング例
401
- 論理端末名称の形式 154
- 論理端末名称の決定 154
- 論理メッセージの扱い〔HSC1 手順〕 17
- 論理メッセージの扱い〔HSC2 手順〕 44

ソフトウェアマニュアルのサービス ご案内

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

1. マニュアル情報ホームページ

ソフトウェアマニュアルの情報をインターネットで公開しております。

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

ホームページのメニューは次のとおりです。

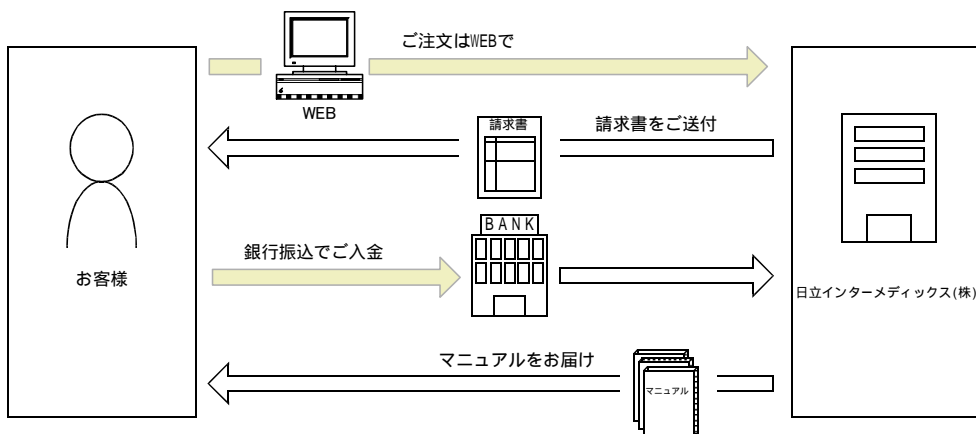
マニュアル一覧	日立コンピュータ製品マニュアルを製品カテゴリ、マニュアル名称、資料番号のいずれかから検索できます。
CD-ROMマニュアル情報	複数マニュアルを格納したCD-ROMマニュアルを提供しています。どの製品に対応したCD-ROMマニュアルがあるか、を参照できます。
マニュアルのご購入	日立インターメディックス(株)の「日立コンピュータ製品マニュアルサイト」からお申し込みできます。 (詳細は「3. マニュアルのご注文」を参照してください。)
Web提供マニュアル一覧	インターネットで参照できるマニュアルの一覧を提供しています。 (詳細は「2. インターネットからのマニュアル参照」を参照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

2. インターネットからのマニュアル参照(ソフトウェアサポートサービス)

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

3. マニュアルのご注文

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



下記 URL にアクセスして必要事項を入力してください。

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

ご注文いただいたマニュアルについて、請求書をお送りします。

請求書の金額を指定銀行へ振り込んでください。なお、送料は弊社で負担します。

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