

OpenTP1 Version 7
分散トランザクション処理機能

TP1/EE/Message Control Extension 使用の手引

解説・手引・文法書

3000-3-F57-60

前書き

■ 対象製品

- ・適用 OS : Red Hat Enterprise Linux Server 6 (64-bit x86_64), Red Hat Enterprise Linux Server 7 (64-bit x86_64), Red Hat Enterprise Linux Server 8 (64-bit x86_64)

P-8264-9311 uCosminexus TP1/EE/Message Control Extension 02-05

- ・適用 OS : Red Hat Enterprise Linux Server 9 (64-bit x86_64)

P-8864-9311 uCosminexus TP1/EE/Message Control Extension 02-07

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「リースノート」でご確認ください。

■ 輸出時の注意

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

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

記載の会社名、製品名などは、それぞれの会社の商標もしくは登録商標です。

■ 発行

2025 年 4 月 3000-3-F57-60

■ 著作権

All Rights Reserved. Copyright (C) 2008, 2025, Hitachi, Ltd.

変更内容

変更内容 (3000-3-F57-60) uCosminexus TP1/EE/Message Control Extension 02-07,
uCosminexus TP1/EE/Message Control Extension 02-05

追加・変更内容	変更箇所
対象製品を更新した。 <ul style="list-style-type: none">• P-1M64-9311 および P-9W64-9311 を削除した。• Red Hat Enterprise Linux Server 8 (64-bit x86_64)をサポートした (P-8264-9311)。• Red Hat Enterprise Linux Server 9 (64-bit x86_64)をサポートした (P-8864-9311 の追加)。	—
ee_mcp_sendrecv()の ANSI C, C++の形式を変更しました。	5.2 ee_mcp_sendrecv - 同期型メッセージの送受信

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは、次に示すプログラムプロダクトの概要、定義方法、運用方法、および操作方法について説明したものであります。

- uCosminexus TP1/EE/Message Control Extension

以降、このマニュアルでは、「uCosminexus TP1/EE/Message Control Extension」を「MCP」と表記します。

■ 対象読者

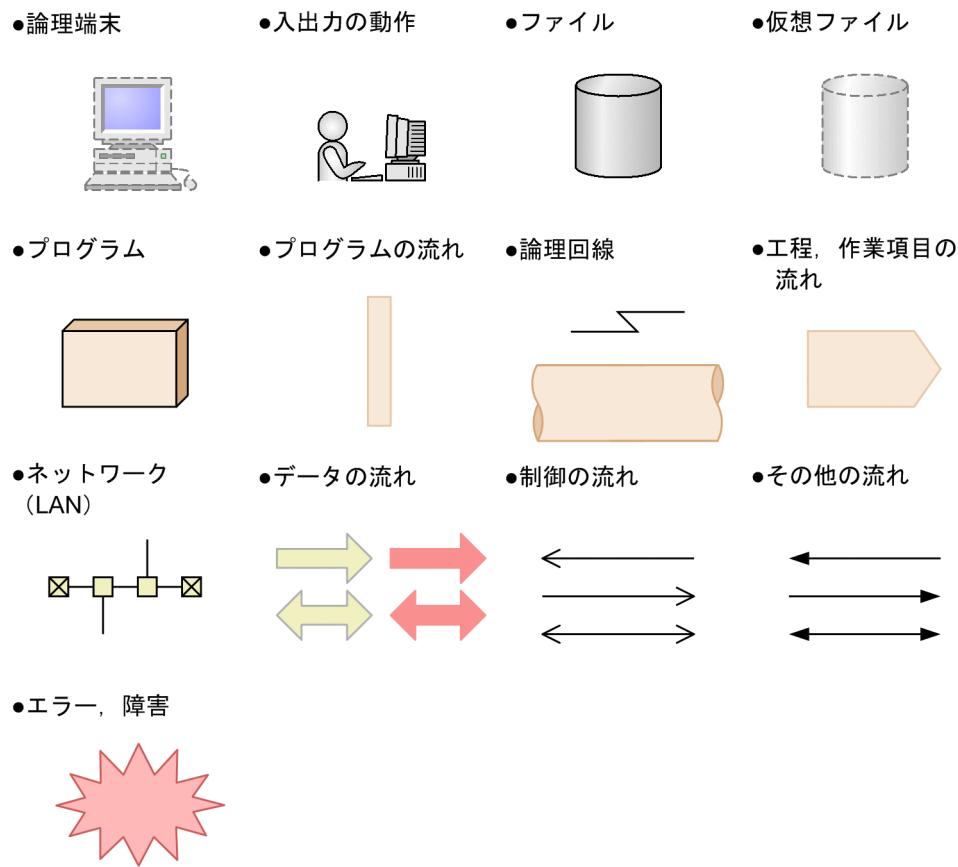
システム管理者、システム設計者、プログラマ、およびオペレータの方を対象としています。

また、次に示す知識があることを前提としています。

- Linux のシステム管理の基礎的な知識
- オペレーティングシステムとオンラインシステムの基礎的な知識
- TP1/Server Base および TP1/Server Base Enterprise Option の知識

■ 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。



■ 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)	'~' (チルダ)

■ 文法の記号

システム定義の説明、およびコマンドの説明に使用する各種の記号を説明します。

ここで説明する文法記述記号、属性表示記号および構文要素記号は、実際の定義には記述しません。

(1) 文法記述記号

システム定義、コマンドのオプションおよび引数の指定方法について説明する記号です。

文法記述記号	意味
[]	<p>この記号で囲まれている項目は省略できることを意味しています。</p> <p>(例) [-r ラン ID] この場合, -r オプションの指定を省略できます。</p>
...	<p>この記号で示す直前の項目を繰り返し指定できることを意味しています。</p> <p>(例) eemcped MCP トレースファイル名 [MCP トレースファイル名…] この場合, eemcped コマンドに MCP トレースファイル名を複数指定できます。</p>
{} {}}	<p>この記号で囲まれた複数の項目が一つの繰り返し項目の単位であることを示します。</p> <p>(例) {{[eemcple -l 論理端末名 -r "portno=自システムのポート番号"]}} これは, 次のように繰り返し指定できることを示します。 eemcple -l 論理端末名 -r "portno=自システムのポート番号" eemcple -l 論理端末名 -r "portno=自システムのポート番号"</p>
{ }	<p>この記号で囲まれている複数の項目のうちから一つを選択することを示します。項目の区切りは で示します。</p> <p>(例) {00000000 00000010} これは, 00000000 と 00000010 の二つの項目のうち, どちらかを指定することを示します。</p>
	<p>この記号で区切られた項目は選択できることを意味しています。</p> <p>(例) -i auto <u>manual</u> この場合, auto または manual のどちらかを指定できます。</p>
— (下線)	<p>この記号で示す項目は, 該当オペランド, またはオプションを省略した場合の省略時解釈値を示します。</p> <p>(例) multicast= yes <u>no</u> この場合, multicast オペランドを省略すると, no が仮定されます。</p>

(2) 属性表示記号

システム定義, およびコマンドに指定できる値の範囲を説明する記号です。

属性表示記号	意味
~	この記号のあとに指定値の属性を示します。
《 》	指定を省略したときに仮定される値を示します。
〈 〉	指定値の構文要素を示します。
(())	指定値の指定範囲を示します。

(3) 構文要素記号

システム定義, およびコマンドに指定できる文字または値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)
〈符号なし整数〉	数字列 (0~9)
〈16進数〉	数字 (0~9), A~F, および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
〈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)
〈ホスト名〉	英数字, および!, #, \$, %, &, ', (,), *, +, -, . (ピリオド), /, ;, <, =, >, ?, @, [, ¥,], ^, _ (アンダスコア), {, , }, ~, , (コンマ) ただし, OS によって使用できる文字が異なります。詳細は, 使用している OS のマニュアルを参照してください。

使用上の注意

すべて半角文字を使用してください。

目次

前書き 2

変更内容 3

はじめに 4

1 概要 13

- 1.1 MCP とは 14
- 1.2 MCP で通信するメッセージの種類 15
 - 1.2.1 一方送信メッセージ 15
 - 1.2.2 一方受信メッセージ 17
 - 1.2.3 同期型メッセージ 18
 - 1.2.4 送受信するメッセージのデータ形式 23
 - 1.2.5 送受信するメッセージの最大サイズ 24
- 1.3 メッセージ送信先の指定 26
 - 1.3.1 TCP/IP 26
 - 1.3.2 UDP 27
- 1.4 MCP で使用できるユーザオウンコーディング (UOC) 28

2 機能 29

- 2.1 コネクションの管理 30
 - 2.1.1 コネクションの状態と状態遷移 30
 - 2.1.2 コネクションの確立 31
 - 2.1.3 コネクションの解放 35
 - 2.1.4 コネクションリプレース機能 38
 - 2.1.5 無通信状態監視 41
- 2.2 論理端末の管理 43
 - 2.2.1 コネクションと論理端末の関係 43
 - 2.2.2 論理端末の状態と状態遷移 45
 - 2.2.3 論理端末の閉塞と閉塞解除 46
 - 2.2.4 送信処理中の論理端末に対するメッセージ送信 47
- 2.3 MCP 後処理トランザクション 49
- 2.4 エラートランザクション 51
- 2.5 端末キュー機能 52
- 2.6 時間監視 54
 - 2.6.1 メッセージの送信完了待ち時間の監視 (TCP/IP プロトコル使用時) 54
 - 2.6.2 後続メッセージの時間監視 (TCP/IP プロトコル使用時) 55

2.7	TP1 キャッシュ機能との連携 56
2.7.1	CL サーバでの MCP 56
2.7.2	CL 連携時の定義チェック 58

3 環境設定 62

3.1	環境設定の手順 63
3.1.1	インストール 63
3.1.2	システム定義の設定 64

4 システム定義 66

4.1	システム定義の概要 67
4.1.1	定義の作成手順 67
4.1.2	定義の規則 67
4.1.3	MCP 構成定義オブジェクトファイルの生成 70
4.2	MCP 構成定義の一覧 72
4.3	MCP 構成定義の詳細 80 <ul style="list-style-type: none">eemcpbuf (バッファ定義) 80eemcptcpstart (TCP/IP 定義の開始) 80eemcpcn (コネクション定義 (TCP/IP)) 81eemcple (論理端末定義 (TCP/IP)) 91eemcptcpPEND (TCP/IP 定義の終了) 92eemcpudpstart (UDP 定義の開始) 92eemcple (論理端末定義 (UDP)) 93eemcplefunc (論理端末アプリケーション定義) 98eemcpthdgrp (スレッドグループ定義) 101eemcpudpend (UDP 定義の終了) 101eemcpfunc (アプリケーション定義) 102eemcptrc (トレース定義) 107
4.4	コネクションの形態と通信構成定義との関係 110
4.5	システム定義での注意事項 112
4.5.1	論理端末の定義数 112
4.5.2	MCP でのファイル識別子の使用量 112
4.6	定義例 114
4.6.1	MCP を使用したシステム構成例 114
4.6.2	定義のコーディング例 115

5 MCP のライブラリ関数の文法 119

5.1	MCP のライブラリ関数の一覧 120
5.2	ライブラリ関数 (C 言語) 123 <ul style="list-style-type: none">関数の説明形式 123サービス関数 (SPP) の作成 123トランザクションインターフェース情報 126

	ee_mcp_actcn - コネクションの確立 134
	ee_mcp_actcn_to - 相手システムを指定したコネクションの確立 136
	ee_mcp_actle - 論理端末の閉塞解除 139
	ee_mcp_cninfo - コネクション情報の取得 140
	ee_mcp_dctcn - コネクションの解放 143
	ee_mcp_dctle - 論理端末の閉塞 145
	ee_mcp_send - 非同期一方送信メッセージの送信 147
	ee_mcp_sendrecv - 同期型メッセージの送受信 151
	ee_mcp_sendsync - 同期一方送信メッセージの送信 156
	ee_mcp_utrace_put - ユーザトレースの取得 159
	ee_mcp_utrace_put_long - ユーザトレースの取得 (データ長拡張用) 160
5.3	ライブラリ関数 (COBOL 言語) 162
	COBOL-UAP 作成用プログラムの説明形式 162
	サービスプログラム (SPP) の作成 162
	トランザクションインターフェース情報 165
	CBLEEMCP('ACTCN ') - コネクションの確立 172
	CBLEEMCP('ACTCNTO ') - 相手システムを指定したコネクションの確立 174
	CBLEEMCP('ACTLE ') - 論理端末の閉塞解除 177
	CBLEEMCP('CNINFO ') - コネクション情報の取得 179
	CBLEEMCP('DCTCN ') - コネクションの解放 181
	CBLEEMCP('DCTLE ') - 論理端末の閉塞 184
	CBLEEMCP('SEND ') - 非同期一方送信メッセージの送信 185
	CBLEEMCP('SENDRECV ') - 同期型メッセージの送受信 190
	CBLEEMCP('SENDSYNC') - 同期一方送信メッセージの送信 196
	CBLEEMCP('UTPUT ') - ユーザトレースの取得 201
	CBLEEMCP('UTPUTL ') - ユーザトレースの取得 (データ長拡張用) 203
5.4	ユーザオウンコーディング (UOC) 205
5.4.1	UOC インタフェース 205
5.4.2	UOC で使用できる関数 205
5.4.3	UOC の実行タイミング 206
5.4.4	入力セグメント判定 UOC 206
5.4.5	入力セグメント判定 UOC インタフェース 207
5.4.6	入力メッセージ編集 UOC 214
5.4.7	入力メッセージ編集 UOC インタフェース 216
5.4.8	出力メッセージ編集 UOC 223
5.4.9	出力メッセージ編集 UOC インタフェース 224
5.4.10	コネクション確立 UOC 230
5.4.11	コネクション確立 UOC インタフェース 230
6	運用 233
6.1	開始と終了 234
6.2	ファイルの運用 235
6.2.1	MCP トレースファイルの運用 235

6.3	運用コマンド 239
	運用コマンドの記述形式 239
	運用コマンドの一覧 239
	eemcpactcn (コネクションの確立) 240
	eemcpactcnto (相手システムを指定したコネクションの確立) 241
	eemcpactle (論理端末の閉塞解除) 243
	eemcpdctcn (コネクションの解放) 244
	eemcpdctle (論理端末の閉塞) 246
	eemcpdump (MCP トレースの出力) 247
	eemcped (MCP トレースの編集) 254
	eemcpfput (MCP トレースの強制ファイル出力) 262
	eemcplscn (コネクション状態表示) 263
	eemcplsle (論理端末の状態表示) 266

7 障害対策 268

7.1	TCP/IP プロトコルに関連する障害 269
7.1.1	コネクション障害 269
7.1.2	受信障害 269
7.1.3	送信障害 271
7.1.4	トランザクションの起動の失敗 (リソース不足によって処理キュー登録に失敗) 272
7.1.5	ハーフコネクションに関連する障害 273
7.2	UDP プロトコルに関連する障害 274
7.2.1	論理端末の閉塞解除の失敗 274
7.2.2	受信障害 274
7.2.3	送信障害 275
7.2.4	トランザクションの起動の失敗 (リソース不足によって処理キュー登録に失敗) 276

付録 277

付録 A	インストールディレクトリとファイル 278
付録 B	MCP のエラーコード一覧 280
付録 C	運用コマンド実行機能での MCP コマンドの実行可否 283
付録 D	各バージョンの変更内容 285
付録 D.1	変更内容 (3000-3-F57-50) 285
付録 D.2	変更内容 (3000-3-F57-40) 285
付録 D.3	変更内容 (3000-3-F57-30) 285
付録 D.4	変更内容 (3000-3-F57-20) 287
付録 D.5	変更内容 (3000-3-F57-10) 289
付録 E	このマニュアルの参考情報 291
付録 E.1	関連マニュアル 291
付録 E.2	このマニュアルでの表記 293
付録 E.3	英略語 293

付録 E.4 KB (キロバイト) などの単位表記について 294

付録 F 用語解説 295

索引 300

1

概要

この章では、MCP の概要について説明します。

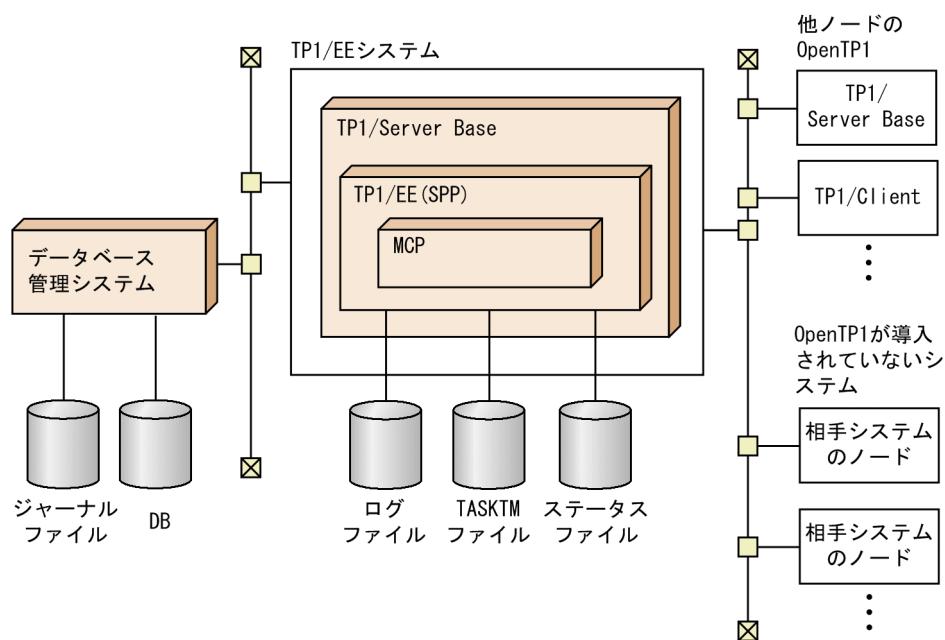
1.1 MCP とは

MCP は、TP1/EE 上で動作するプログラムです。システム間でのメッセージ送受信を行うために必要なメッセージ制御機能を実現します。

通信先の相手システムに OpenTP1 が導入されていない場合は、通信プロトコルに従ったメッセージ通信を行う必要があります。このような場合に MCP を使用すると、TP1/EE で TCP/IP プロトコルおよび UDP プロトコルを使用したメッセージ通信ができるようになります。

MCP を組み込んだ OpenTP1 システムの構成例を次の図に示します。

図 1-1 MCP を組み込んだ OpenTP1 システムの構成例



1.2 MCP で通信するメッセージの種類

MCP では、TCP/IP プロトコルおよび UDP プロトコルを使用したメッセージ通信ができます。使用するプロトコルによるメッセージの種類は次のとおりです。

■TCP/IP プロトコルを使用する場合

TCP/IP プロトコルを使用した一方送信、一方受信および同期送受信ができます。

■UDP プロトコルを使用する場合

次に示すメッセージの送受信ができます。

- 不特定の相手に対する一方送信（ブロードキャスト）
- 特定の一つの相手に対する一方送信（ユニキャスト）
- 特定の複数の相手に対する一方送信（マルチキャスト）
- 一方受信（ブロードキャスト、ユニキャスト、マルチキャスト）

1.2.1 一方送信メッセージ

一方送信メッセージには、非同期一方送信と同期一方送信の 2 種類があります。

■非同期一方送信

非同期一方送信は、トランザクションと連動して相手システムにメッセージを送信する場合に使用します。一つのトランザクションで複数のメッセージを非同期一方送信した場合でも、トランザクションが決着するまで相手システムへメッセージは送信されません。つまり、トランザクションがコミットした場合にだけメッセージを送信し、トランザクションがロールバックした場合にはメッセージを送信しません。

■同期一方送信

同期一方送信は、メッセージの送信要求と相手システムへのメッセージ送信処理を同期させる場合に使用します。一つのトランザクションで複数のメッセージを同期一方送信した場合は、メッセージの送信要求のたびに相手システムへメッセージが送信されます。つまり、トランザクションのコミット、ロールバックに関係なく送信要求のたびに相手システムへメッセージを送信します。

使用するプロトコルに関係なく、デフォルトで送信できるメッセージのサイズは、32 キロバイトまでです。

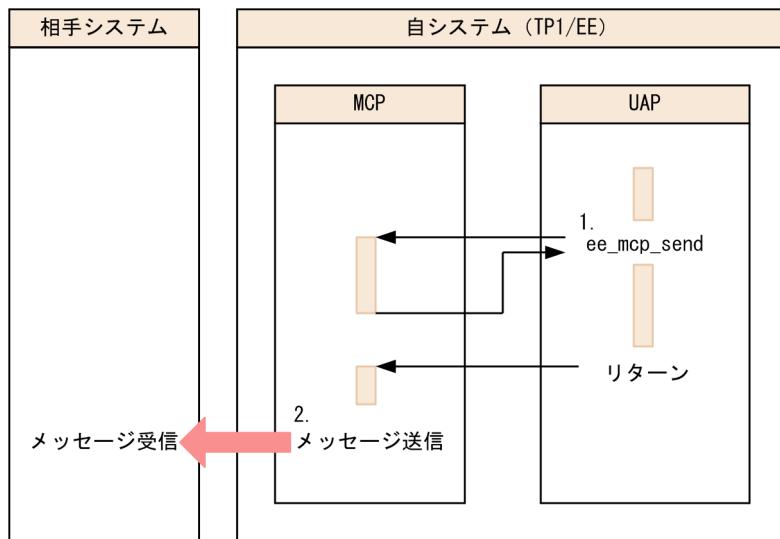
ただし、送信メッセージの最大サイズは、TCP/IP プロトコルを使用する場合は 8 メガバイト、UDP プロトコルを使用する場合は 64 キロバイトまで拡張できます。送信メッセージの最大サイズを拡張する場合は、eemcpfunc 定義コマンドの msglen オペランドに extend を指定してください。

(1) 非同期一方送信

UAP からのメッセージ送信要求を MCP が受け付けたあと、UAP がリターンしてからメッセージを送信する形態です。メッセージを相手システムに非同期一方送信した場合に、障害（コネクション障害など）を検知したときは、送信障害トランザクション（ERRTRNS）を起動して出力メッセージを破棄します。

非同期一方送信でメッセージを送信する場合の処理の流れを次の図に示します。

図 1-2 非同期一方送信でメッセージを送信する場合の処理の流れ



(凡例)

→ : メッセージの流れ

→ : 制御の流れ

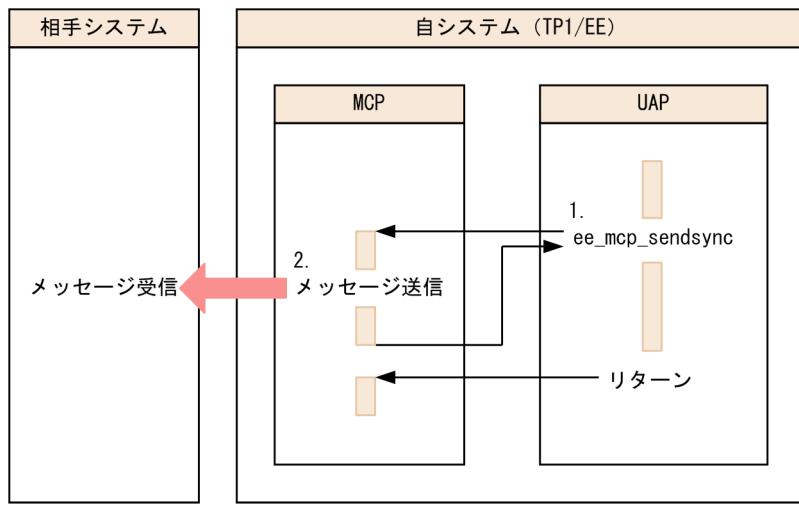
1. UAP は、MCP に対してメッセージ送信要求を行います。

2. MCP は、UAP がリターン（トランザクションがコミット）したことを契機に、相手システムにメッセージを送信します。

(2) 同期一方送信

UAP からのメッセージ送信要求を MCP が受け付けるタイミングに合わせて、メッセージを送信する形態です。メッセージを相手システムに同期一方送信した場合に、障害（コネクション障害など）を検知したときは、呼び出し元にエラーリターンします。同期一方送信でメッセージを送信する場合の処理の流れを次の図に示します。

図 1-3 同期一方送信でメッセージを送信する場合の処理の流れ



(凡例)

→ : メッセージの流れ

→ : 制御の流れ

1. UAP は、MCP に対してメッセージ送信要求を行います。

2. MCP は、UAP からのメッセージ送信要求を受け付けたことを契機に、相手システムにメッセージを送信します。

1.2.2 一方受信メッセージ

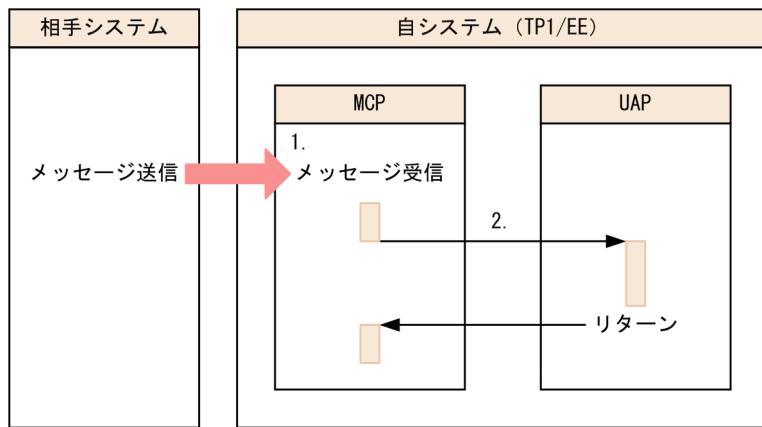
一方受信メッセージは、相手システムからの一方送信メッセージを自システムに受信する形態です。

使用するプロトコルに関係なく、デフォルトで受信できるメッセージのサイズは、受信バッファサイズ (TP1/EE サービス定義の `recv_message_buf_size` オペランドで指定) に指定したサイズまでです。

ただし、受信メッセージの最大サイズは、TCP/IP プロトコルを使用する場合は 8 メガバイト、UDP プロトコルを使用する場合は 64 キロバイトまで拡張できます。受信メッセージの最大サイズを拡張する場合は、`eemcpfunc` 定義コマンドの `msglen` オペランドに `extend` を指定してください。

一方受信メッセージを受信した場合の処理の流れを次の図に示します。

図 1-4 一方受信メッセージを受信した場合の処理の流れ



(凡例)

→ : メッセージの流れ

→ : 制御の流れ

1. 相手システムからの一方送信メッセージを、 MCP が受信します。
2. MCP は、受信したメッセージを UAP に引き渡します。

1.2.3 同期型メッセージ

(1) 機能概要

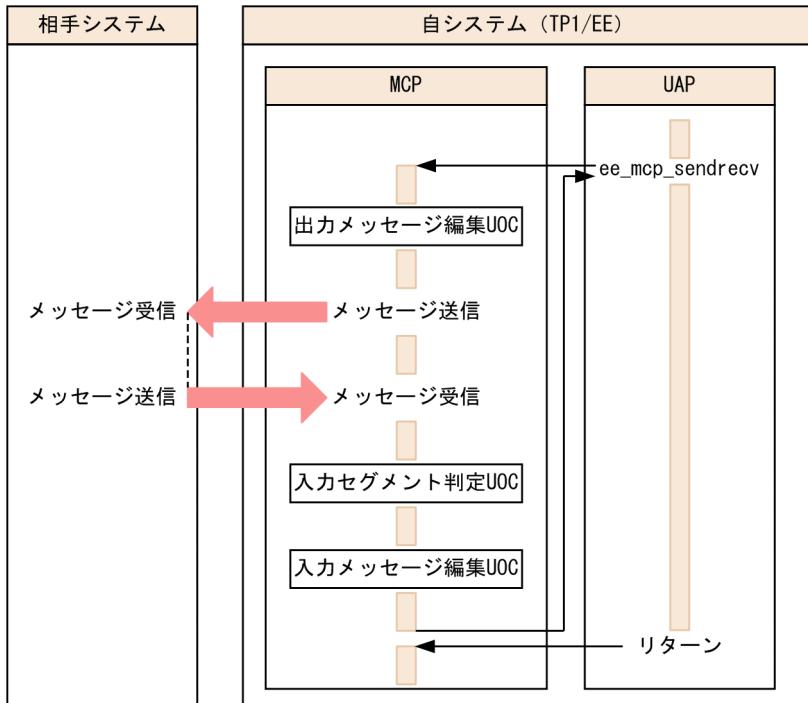
同期型メッセージ送受信は、自システムからのメッセージ送信と、相手システムからの応答メッセージの受信を連続で行う形態です。この形態は、自システムの UAP と相手システムとの間でメッセージの同期合わせ、順序管理をする場合に使用します。一つのトランザクションで複数のメッセージを送信した場合は、メッセージの送信要求のたびに相手システムへメッセージが送信されます。つまり、トランザクションのコミット、ロールバックに関係なく送信要求のたびに相手システムへメッセージを送信します。

同期型メッセージの送受信機能は、TCP/IP プロトコルで使用できます。UDP プロトコルでは、同期型メッセージの送受信機能は使用できません。

UAP は、API (ee_mcp_sendrecv 関数または CBLEEMCP('SENDRECV')) を発行して MCP にメッセージの送受信を要求します。MCP は受け取ったメッセージを編集したあと、相手システムに送信処理をします。送信処理が完了したあとは、UAP にリターンしないで、引き続き受信待ち状態となります。相手システムからのメッセージを受信すると、MCP はメッセージを編集して、UAP に引き渡します。

同期型メッセージ送受信でメッセージを送受信する場合の処理の流れを次の図に示します。

図 1-5 同期型メッセージ送受信の処理の流れ



(凡例)

→ : メッセージの流れ

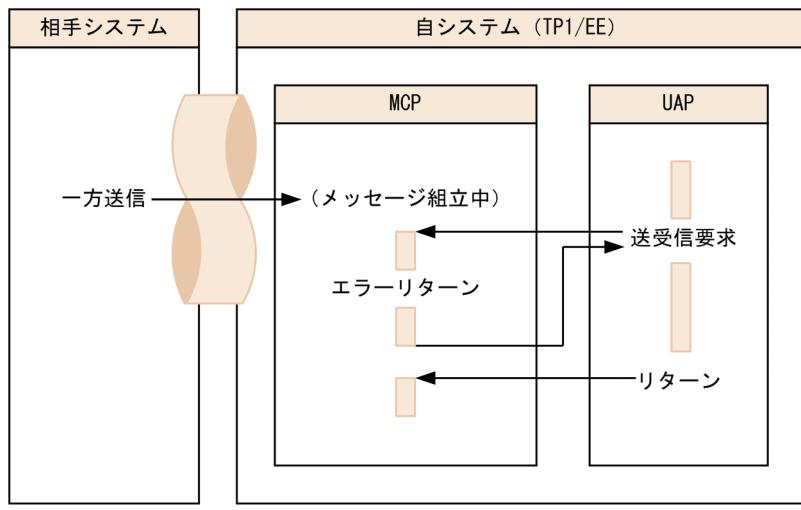
→ : 制御の流れ

(2) メッセージのすれ違い

(a) 送受信要求前に相手システムからの一方送信メッセージを受信

同期送受信の送信要求時、同一コネクションすでに相手システムからの一方送信メッセージを受信し組立中の場合、同期送受信関数はエラーリターンします。

図 1-6 送信要求前に相手システムからの一方送信メッセージを受信



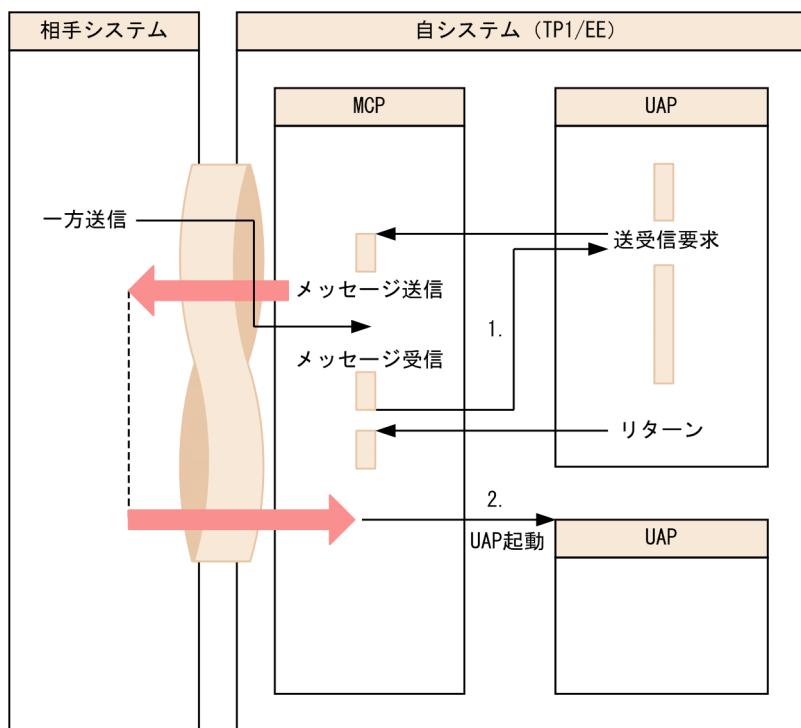
(凡例)

→ : 制御の流れ

(b) 送信処理中に相手システムからの一方送信メッセージを受信

同一コネクションで、自システムからの同期送受信要求に対するメッセージ受信と、相手システムからの一方送信メッセージがネットワーク上ですれ違うことがあります。

図 1-7 送信処理中に相手システムからの一方送信メッセージを受信



(凡例)

→ : メッセージの流れ

→ : 制御の流れ

表 1-1 受信メッセージすれ違い時の MCP の動作

図中の番号	受信したメッセージの種別	MCP の動作
1.	同期送受信の送信要求時、ネットワーク上のメッセージすれ違いで、相手システムからの一方送信メッセージを受信。	相手からの一方送信メッセージを、応答メッセージとして受信します。
2.	図中の 1 で示す相手システムからの一方送信メッセージ受信後、同期送受信の送信に対する応答メッセージを受信。	同期送受信の応答メッセージを、相手システムからの一方送信メッセージとして受信します。

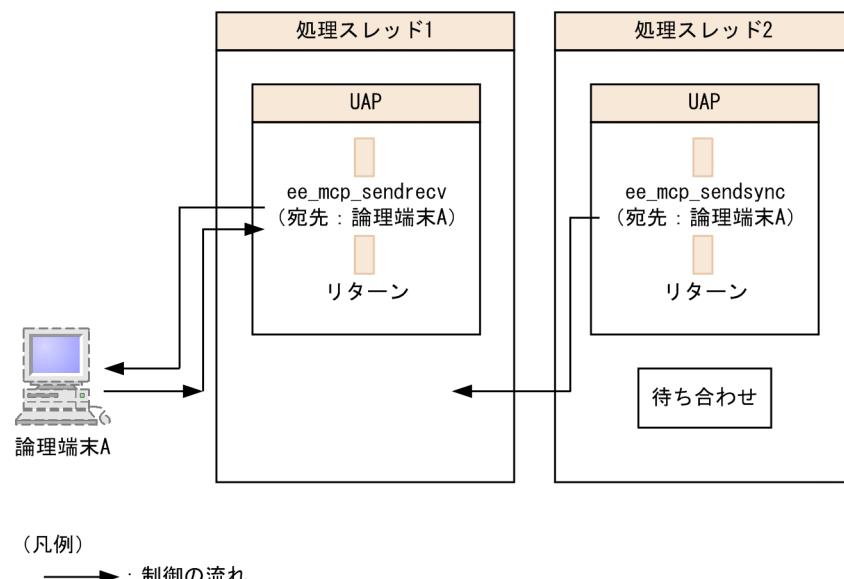
(3) 送信処理中の論理端末に対するメッセージ送信

複数のサービスで同一の論理端末にメッセージを送信した場合、先頭のメッセージによって論理端末が送信処理中になることがあります。

(a) 同期型メッセージ送受信処理中の論理端末に対する一方メッセージ送信

同期型メッセージ送受信は、メッセージの送信から応答メッセージの受信まで同じ論理端末を使用します。そのため先行した同期型メッセージの送受信が応答受信するまで、同一論理端末に対する後続の一方メッセージ送信を待ち合わせます。

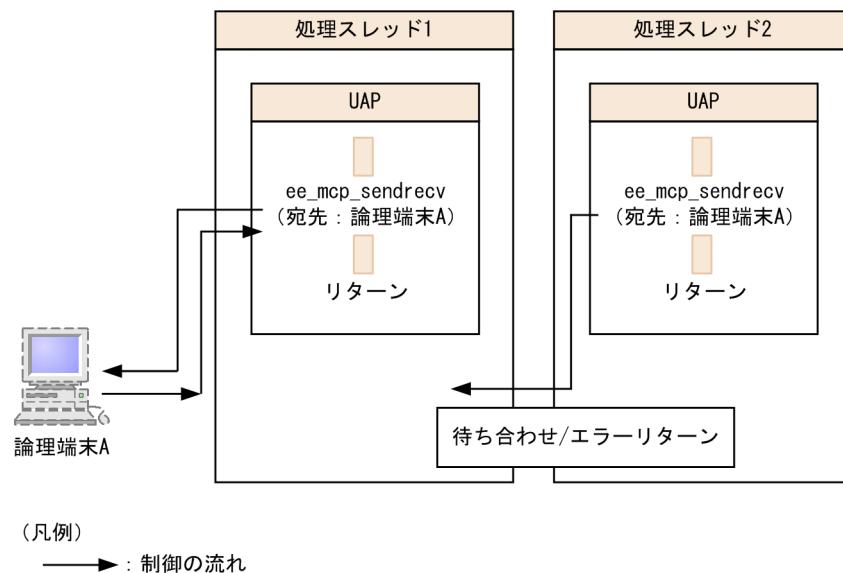
図 1-8 送受信処理中の論理端末に対する一方メッセージ送信



(b) 送信中の論理端末に対する同期型メッセージ送受信

同期型メッセージ送受信中、または一方メッセージ送信中の論理端末に対して同期型メッセージの送受信関数を発行した場合、先行するメッセージの送受信処理が完了するまで待ち合わせて送信するか、または即時エラーリターンとするかは、同期型メッセージ送受信関数の引数で指定することができます。

図 1-9 送受信処理中の論理端末に対する同期型メッセージ送受信



(4) コネクションの切断抑止

同期型メッセージ送受信関数の発行時に指定した監視タイマ（引数：watchtime で指定したメッセージ送受信完了待ち時間）がタイムアウトすると、コネクションは切断されます。

しかし、MCP 構成定義の指定によって、コネクションの切断を抑止することができます。コネクションの切断抑止では、同期型メッセージ送受信関数がタイムアウトした場合、コネクションを切断しません。

コネクションの切断を抑止する場合、入力セグメント判定 UOC で、セグメント未完成のときに後続メッセージ監視タイマの設定を行わないでください（UOC のパラメタ：eemcp_uoctimer_inf の timer_code に EEMCP_TIME_NO_SET を指定してください）。後続メッセージ監視タイマでタイマ監視を行うと、監視タイマがタイムアウトしたときにコネクションは切断されます。

(5) 時間監視

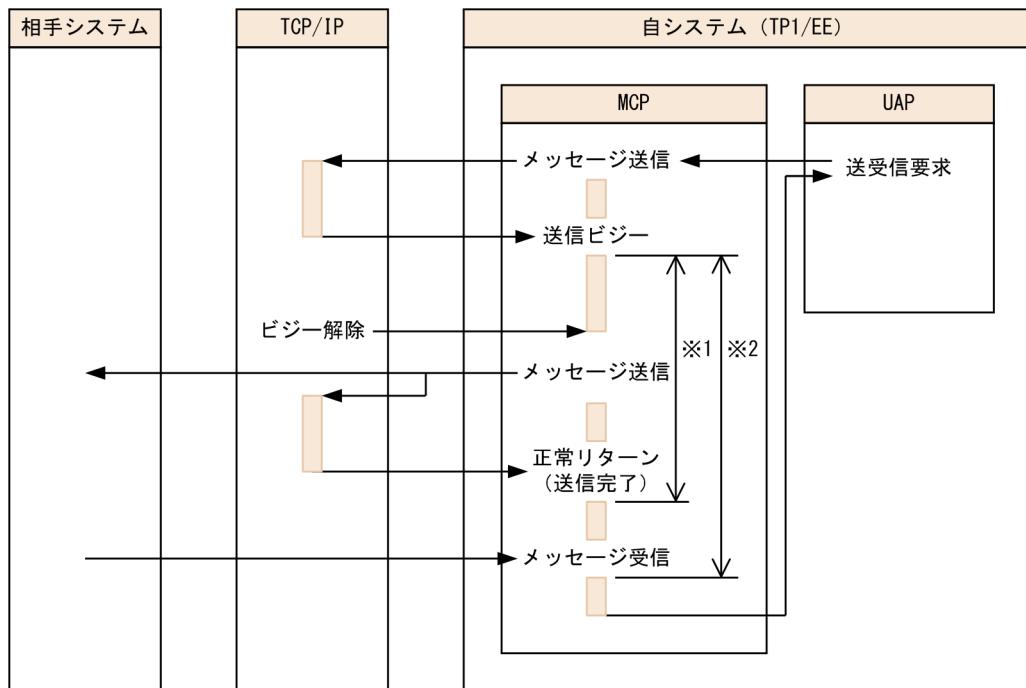
(a) メッセージ送受信完了監視タイマ (TCP/IP プロトコル使用時)

同期型メッセージ送受信関数を発行してから、応答メッセージを受信するまでのタイマ監視を行います。

他スレッドで同一論理端末にメッセージを送信中の場合、メッセージ送受信の待ち合わせを行いますが、この待ち合わせ時間も送信ビジーとしてタイマ監視時間に含みます。

メッセージの送受信完了待ちをする場合の処理の流れを次の図に示します。

図 1-10 メッセージ送受信完了監視タイマ (TCP/IP)



(凡例)

→ : 制御の流れ

注※1 : 送信完了待ちタイマ

注※2 : メッセージ送受信完了タイマ監視

項目	MCP 構成定義 (eemcpcn -w) の指定による動作	
	srtimout オペランドに yes を指定 (コネクションの切断を抑止する)	srtimout オペランドに no を指定 (コネクションの切断を抑止しない)
コネクション	コネクションを切断しません。	コネクションを切断します。
RL (MCP 後処理トランザクション)	起動しません。	起動します。 次に示す内容を通知します。 <ul style="list-style-type: none">• reasonEEMCP_RSN_SRTO
リターン値	EEMCPER_TIME (監視タイマタイムアウト) が返されます。	
ステータスコード	10007 (監視タイマタイムアウト) が返されます。	

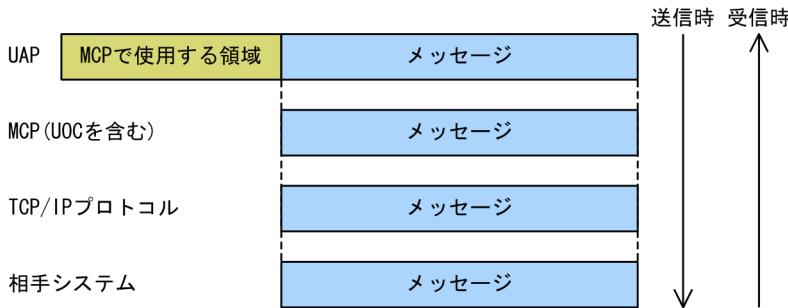
1.2.4 送受信するメッセージのデータ形式

TCP/IP プロトコルまたは UDP プロトコルを使用した通信で送受信するメッセージは、自システムの UAP と相手システムの間の受け渡しの過程でデータ形式が変化します。

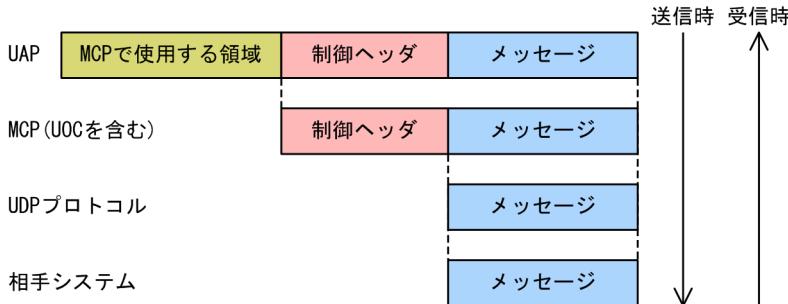
送受信するメッセージのデータ形式の変化を次の図に示します。

図 1-11 送受信するメッセージのデータ形式の変化

●TCP/IPの場合



●UDPの場合



(凡例) —————→: データ形式の変化の流れ

1.2.5 送受信するメッセージの最大サイズ

送受信するメッセージの最大サイズを次の表に示します。UDP プロトコルの場合は制御ヘッダを含みます。

表 1-2 受信メッセージの最大サイズ

メッセージ長拡張有無		プロトコル種別	
		TCP/IP	UDP/IP
メッセージ長拡張	なし	受信バッファサイズ	65531※1
	あり	8388608	65531※1, ※2

注※1

受信バッファサイズ (recv_message_buf_size) の方が小さい場合は受信バッファサイズ

注※2

入力メッセージ編集 UOC での編集後の最大サイズは 8388608

表 1-3 送信メッセージの最大サイズ

メッセージ長拡張有無		プロトコル種別	
		TCP/IP	UDP/IP
メッセージ長拡張	なし	32000*	32000*

メッセージ長拡張有無		プロトコル種別	
		TCP/IP	UDP/IP
メッセージ長拡張	あり	8388608	65531*

注※

送信バッファサイズ (send_message_buf_size) の方が小さい場合は送信バッファサイズ

1.3 メッセージ送信先の指定

TCP/IP プロトコルまたは UDP プロトコルでは、任意の通信先にメッセージを送信するときは、送信先を指定します。

なお、メッセージ送信を行う関数は次のとおりです。

- 非同期一方送信メッセージ：ee_mcp_send 関数、または CBLEEMCP('SEND ')関数
- 同期一方送信メッセージ：ee_mcp_sendsync 関数、または CBLEEMCP('SENDSYNC')関数

1.3.1 TCP/IP

TCP/IP プロトコルでは、コネクション確立時に送信先が決定します。コネクションの確立がクライアント型かサーバ型かによって送信先の指定方法は異なります。

メッセージ送信の API を発行してから実際に送信処理を行うまでの間に、コネクションが切り替わった場合（コネクションの切断と再確立やコネクションリプレースなど）、切り替わった送信先に対しメッセージを送信するときがあります。

(1) クライアント型の場合

クライアント型コネクションは、自システムからコネクションの確立を要求します。

クライアント型の場合、送信先は MCP 構成定義の eemcpcn 定義コマンドで指定します。

- -o オプションの oipaddr オペランドで相手システムのホストの IP アドレス、または ohostname オペランドで相手システムのホスト名を指定
- -o オプションの oportno オペランドで相手システムのホストのポート番号を指定

(2) サーバ型の場合

サーバ型コネクションは、相手システムからのコネクションの確立要求を受け入れます。

サーバ型の場合、送信先は MCP 構成定義の eemcpcn 定義コマンドで指定します。

- -o オプションの oipaddr オペランドで相手システムのホストの IP アドレス、または ohostname オペランドで相手システムのホスト名を指定
- -o オプションの oportno オペランドで相手システムのホストのポート番号を指定

コネクション確立後、コネクションを確立した相手が API を発行する際の送信先となります。

なお、コネクション確立要求受信時に相手アドレス情報のチェックを無効にできます。

1.3.2 UDP

UDP プロトコルでは、関数の引数の指定によって、制御ヘッダまたは MCP 構成定義で送信先を指定します。

(1) 制御ヘッダで指定する場合

引数 senddata (またはデータ名 W) に指定する制御ヘッダで、送信先を指定します。制御ヘッダの領域には、送信先となる相手 IP アドレス、および相手ポート番号を指定します。

(2) MCP 構成定義で指定する場合

引数 opcd (またはデータ名 G) の指定で、MCP 構成定義で指定する送信先にメッセージが送信されるようになります。この場合、制御ヘッダの内容に関係なく、MCP 構成定義で指定した送信先が有効になります。

送信先は、MCP 構成定義の eemcple 定義コマンドで指定します。

- -o オプションの oipaddr オペランドで相手システムのホストの IP アドレス、または ohostname オペランドで相手システムのホスト名を指定
- -o オプションの oportno オペランドで相手システムのホストのポート番号を指定

1.4 MCP で使用できるユーザオウンコーディング (UOC)

UAP によるメッセージの処理をより多様な業務に対応させるために、ユーザオウンコーディング (UOC) を作成できます。UOC は C 言語で作成します。COBOL 言語では作成できません。MCP で使用できる UOC の一覧を次の表に示します。

表 1-4 MCP で使用できる UOC の一覧

項番	UOC の種類	UOC でできる処理
1	入力セグメント判定 UOC [※]	メッセージを受信したとき、そのメッセージでセグメントが完成しているかどうかを判定します。セグメントが未完成のとき（後続メッセージがある場合）は、後続メッセージに対して、監視タイマを設定できます。
2	入力メッセージ編集 UOC	受信した論理メッセージをユーザ任意の形式に編集します。また、受信した論理メッセージを基に、ユーザ任意のサービス名を決定できます。
3	出力メッセージ編集 UOC	送信した論理メッセージをユーザ任意の形式に編集します。
4	コネクション確立 UOC	相手システムからコネクション確立要求を受信するたびに起動し、その確立要求を受け入れるかどうかを判定できます。

注※

TCP/IP プロトコルを使用して通信する場合にだけ使用できます。

各 UOC の詳細については、「[5.4 ユーザオウンコーディング \(UOC\)](#)」を参照してください。

2

機能

この章では、コネクションと論理端末について説明します。さらに、MCP 後処理トランザクション、端末キュー、時間監視、および TP1 キャッシュ機能との連携について説明します。

2.1 コネクションの管理

ここでは、MCP でのコネクションの管理について説明します。コネクションとは、UAP に対する論理的な通信路で、TCP/IP プロトコルを使用して通信する場合だけ使用します。

2.1.1 コネクションの状態と状態遷移

コネクションには、クライアント型コネクションとサーバ型コネクションの 2 種類があります。クライアント型コネクションは自システムからのコネクション確立要求によって確立されるコネクションを指し、サーバ型コネクションは相手システムからのコネクション確立要求によって確立されるコネクションを指します。

コネクションの状態は、コマンドの入力や通信障害などのイベントによって遷移します。このとき、MCP は MCP 後処理トランザクション (RL) を起動し、コネクションの状態変化を UAP に通知します。MCP 後処理トランザクションについては、「[2.3 MCP 後処理トランザクション](#)」を参照してください。コネクションの状態の意味を次の表に示します。

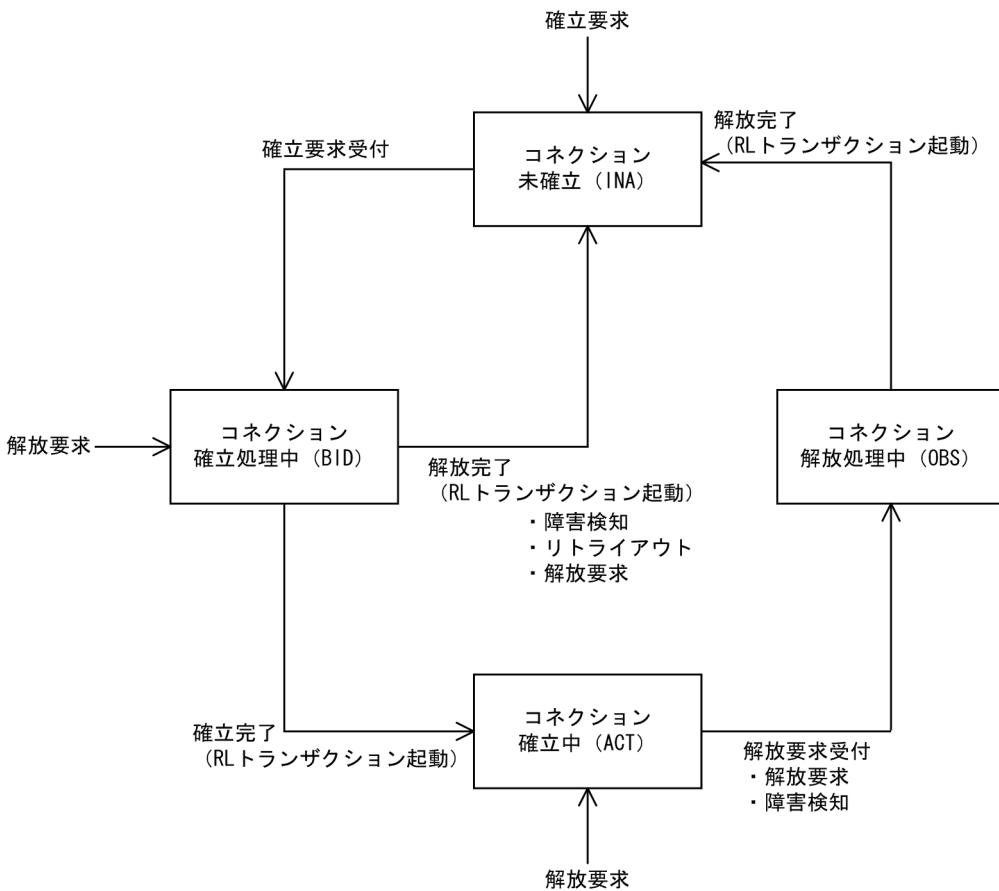
表 2-1 コネクションの状態の意味

項番	状態	意味	
		クライアント型コネクション	サーバ型コネクション
1	コネクション未確立 (INA)	コネクション確立を要求していない状態 (コネクション障害で停止中も含む)	相手システムからコネクション確立要求を受けていない状態 (コネクション障害で停止中も含む)
2	コネクション確立処理中 (BID)	UAP からコネクション確立要求を受け付けてから、コネクション確立完了の MCP 後処理トランザクションが起動するまでの状態	相手システムからコネクション確立要求を受け付けてから、コネクション確立完了の MCP 後処理トランザクションが起動するまでの状態
3	コネクション確立中 (ACT)	コネクション確立完了の MCP 後処理トランザクションが起動している状態 (論理端末でメッセージの送受信ができる状態)	
4	コネクション解放処理中 (OBS)	相手システムがコネクション解放要求 (障害検知も含む) を受け付けてから、コネクション解放の MCP 後処理トランザクションが起動するまでの状態	コネクション障害を検知し、コネクション解放の MCP 後処理トランザクションが起動する前の状態

コネクションにかかるエラーメッセージが出力された場合、コネクションの状態を確認する必要があります。コネクションの状態は、eemcplscn コマンドで表示できます。なお、eemcplscn コマンドに-d オプションを指定すると、コネクションに対応する論理端末の情報も表示できます。

コネクションの状態遷移を次の図に示します。

図 2-1 コネクションの状態遷移



2.1.2 コネクションの確立

ここでは、クライアント型コネクションおよびサーバ型コネクションを確立するときの処理の流れについて説明します。

(1) クライアント型コネクションの確立

クライアント型コネクションの確立には、自動確立と手動確立の2種類があります。

■コネクションの自動確立

オンライン開始前にコネクションを自動確立します。ただし、オンラインはコネクション確立の完了を待たないで開始されます。なお、コネクションの確立が成功した場合は、コネクションの確立を通知するMCP後処理トランザクションは起動しません。

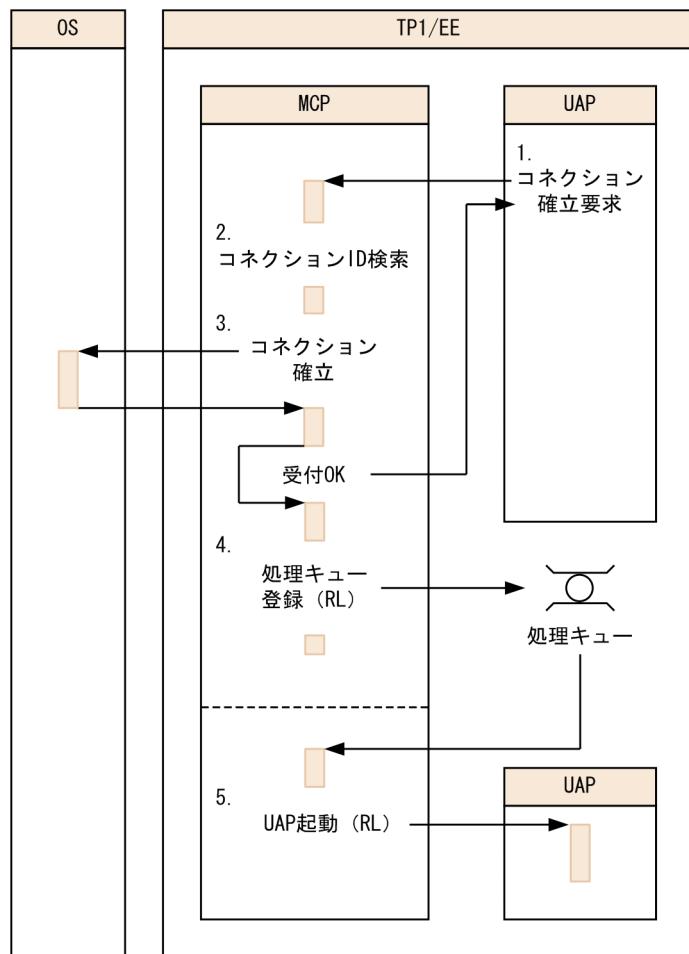
コネクションの確立状態は、KFSB81402-Iメッセージ(eemcpcn定義の-b conmsgオペランドにyesを指定したとき)、またはeemcplscnコマンドで確認してください。

■コネクションの手動確立

ee_mcpactcn コマンドの実行、または ee_mcp_actcn 関数の発行によってコネクションを手動確立します。

ee_mcp_actcn 関数の発行によって、手動でコネクションを確立する場合の処理の流れを次の図に示します。

図 2-2 ee_mcp_actcn 関数の発行によって、手動でコネクションを確立する場合の処理の流れ



(凡例)

→ : 制御の流れ

RL : MCP後処理トランザクション

説明

1. UAP は、MCP に対してコネクションの確立を要求します。
2. MCP は、コネクション管理テーブルでコネクション ID を検索し、そのコネクションが未確立であることをチェックします。
3. コネクションが確立されます。
4. コネクションが確立されると、MCP はコネクション確立の RL (MCP 後処理トランザクション) を処理キューに登録します。

5. UAP にコネクションの確立を通知します。

■コネクションの再確立処理

クライアント型コネクションの確立処理で障害が発生した場合、MCP 構成定義の再試行間隔 (eemcpcn 定義コマンドの-b オプションの bretryint オペランド指定値) および再試行回数 (eemcpcn 定義コマンドの-b オプションの concmptim オペランド指定値) の指定に従い、コネクションの再確立処理が行われます。

なお、コネクション確立障害で再試行リトライアウトとなったコネクションは、MCP 後処理トランザクションを起動してコネクションの障害連絡を行います。

■相手システムを指定したコネクション確立機能

TCP/IP プロトコル（クライアント型コネクション）の論理端末で、コネクション確立 API 発行時、または運用コマンド実行時に、相手システムのアドレス情報（IP アドレス・ポート番号）を指定してコネクションを確立できます。

相手システムを指定したコネクション確立実行時は、MCP 構成定義 eemcpcn の-o オプション指定値の設定値に関係なく、API またはコマンドの引数に指定した相手アドレスに対してコネクションを確立します。この機能で指定した相手アドレス情報はコネクション切断後は引き継ぎません。この機能で確立したコネクションを切断後に、同じコネクション ID に対して eemcpactcn コマンド、または ee_mcp_actcn() でコネクション確立した場合、MCP 構成定義 eemcpcn の-o オプション指定値の設定値に従ってコネクションを確立します。

相手システムを指定したコネクション確立 API の詳細は、「5.2 ライブラリ関数（C 言語）」の「[ee_mcp_actcn_to - 相手システムを指定したコネクションの確立](#)」または「5.3 ライブラリ関数（COBOL 言語）」の「[CBLEEMCP\('ACTCNTO'\) - 相手システムを指定したコネクションの確立](#)」を参照してください。運用コマンド(eemcpactcnto)の詳細は、「5.2 ライブラリ関数（C 言語）」の「[ee_mcp_actcn_to - 相手システムを指定したコネクションの確立](#)」を参照してください。

■コネクション確立時の相手アドレス通知機能

TCP/IP プロトコルの論理端末で、コネクション確立した相手システムのアドレス情報（IP アドレス・ポート番号）を通知します。

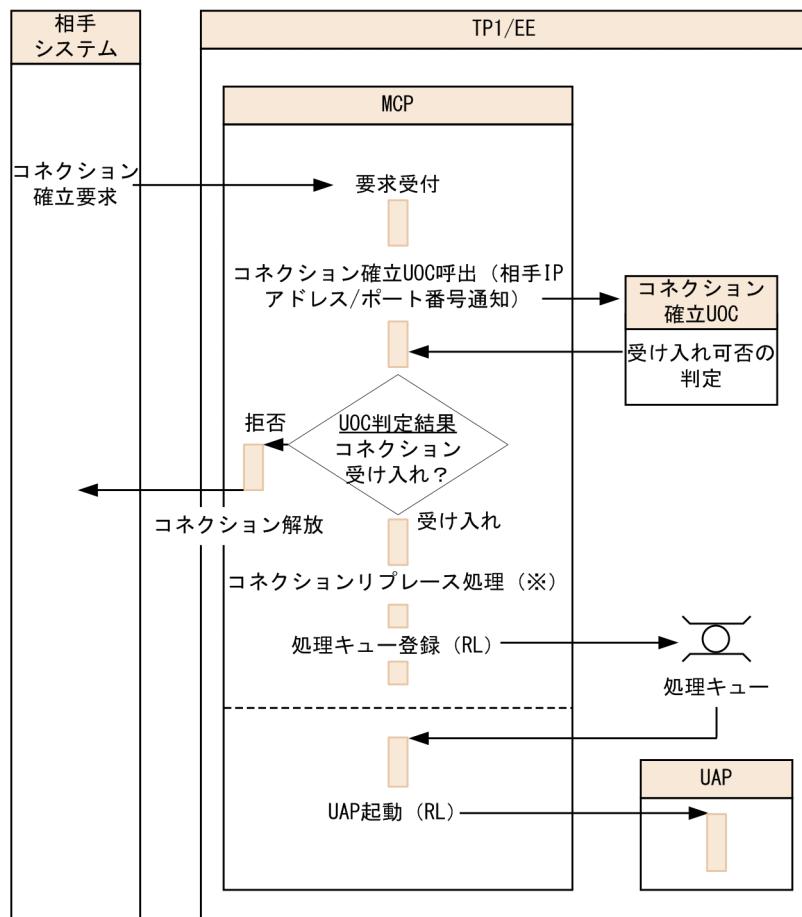
(a) コネクション確立 UOC

サーバ型コネクションで、相手システムからコネクション確立要求を受信するたびにコネクション確立 UOC を起動し、コネクション確立要求を発行した相手システムのアドレス情報を通知します。

コネクション確立 UOC の詳細は、「[5.4.10 コネクション確立 UOC](#)」を参照してください。

コネクション確立 UOC の処理の流れを次の図に示します。

図 2-3 コネクション確立 UOC の処理の流れ



(凡例)

→ : 制御の流れ

RL : MCP後処理トランザクション

注※ : コネクションリプレース機能を使用した場合、割り当て可能な未確立コネクションがないときに動作します。

(b) MCP 拡張トランザクションインターフェース

MCP 後処理トランザクション(RL), または MCP メッセージ受信時のユーザトランザクションの MCP 拡張トランザクションインターフェース情報で, 相手システムのアドレス情報を通知します。

コネクション解放時の MCP 後処理トランザクションでは, 相手システムのアドレス情報は通知しません。

MCP 拡張トランザクションインターフェースの詳細は, 「5.2 ライブラリ関数 (C 言語)」の「MCP 拡張トランザクションインターフェース情報」または「5.3 ライブラリ関数 (COBOL 言語)」の「MCP 拡張トランザクションインターフェース情報」を参照してください。

(c) コネクション情報取得

コネクション情報を取得する API (ee_mcp_cninfo(), CBLEEMCP('CNINFO')) で, 相手システムのアドレス情報を通知します。

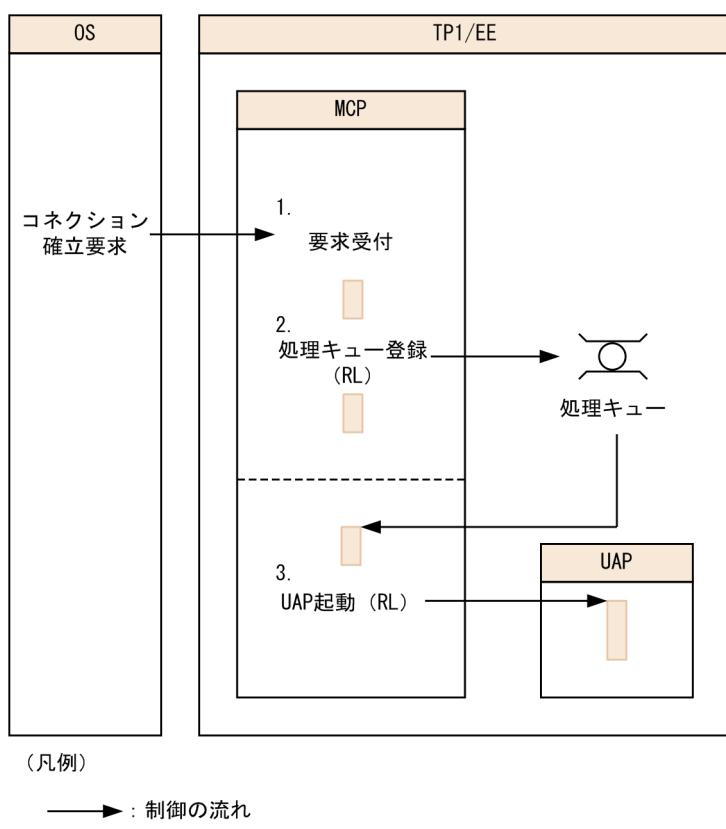
コネクション情報取得 API の詳細は, 「5.2 ライブラリ関数 (C 言語)」の「ee_mcp_cninfo - コネクション情報の取得」または「5.3 ライブラリ関数 (COBOL 言語)」の「CBLEEMCP('CNINFO ') - コネクション情報の取得」を参照してください。

(2) サーバ型コネクションの確立

サーバ型コネクションの確立では、オンライン開始時にポートをオープンしておくことで、相手システムからのコネクション確立要求を待ちます。

サーバ型コネクションを確立する場合の処理の流れを次の図に示します。

図 2-4 サーバ型コネクションを確立する場合の処理の流れ



説明

1. 相手システムからのコネクション確立要求を受け付けます。
2. コネクション確立の RL (MCP 後処理トランザクション) を処理キューに登録します。
3. UAP にコネクションの確立を通知します。

2.1.3 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の 2 種類があります。

(1) コネクションの自動解放

MCP がコネクションを自動解放するタイミングは、次に示すとおりです。

- メッセージ受信で障害を検知した時

TCP/IP プロトコルを使用時に、入力セグメント判定 UOC でエラーリターンまたはパラメタ不正があったときは、MCP によってコネクションが解放されます。このとき、入力メッセージは破棄され、コネクション障害の MCP 後処理トランザクションが起動されます。

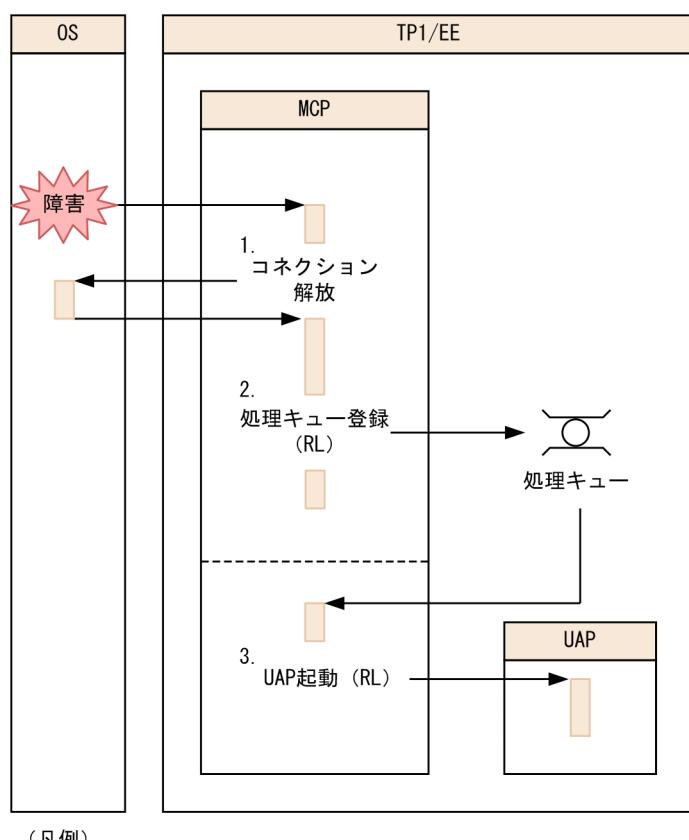
- オンラインを終了した時

オンライン終了時には、MCP によってすべてのコネクションが解放されます。

- コネクション障害が発生した時

コネクション障害時には、MCP によってコネクションが解放されます。コネクション障害が発生した場合のコネクションの解放の流れを次の図に示します。

図 2-5 コネクション障害が発生した場合のコネクションの解放の流れ



説明

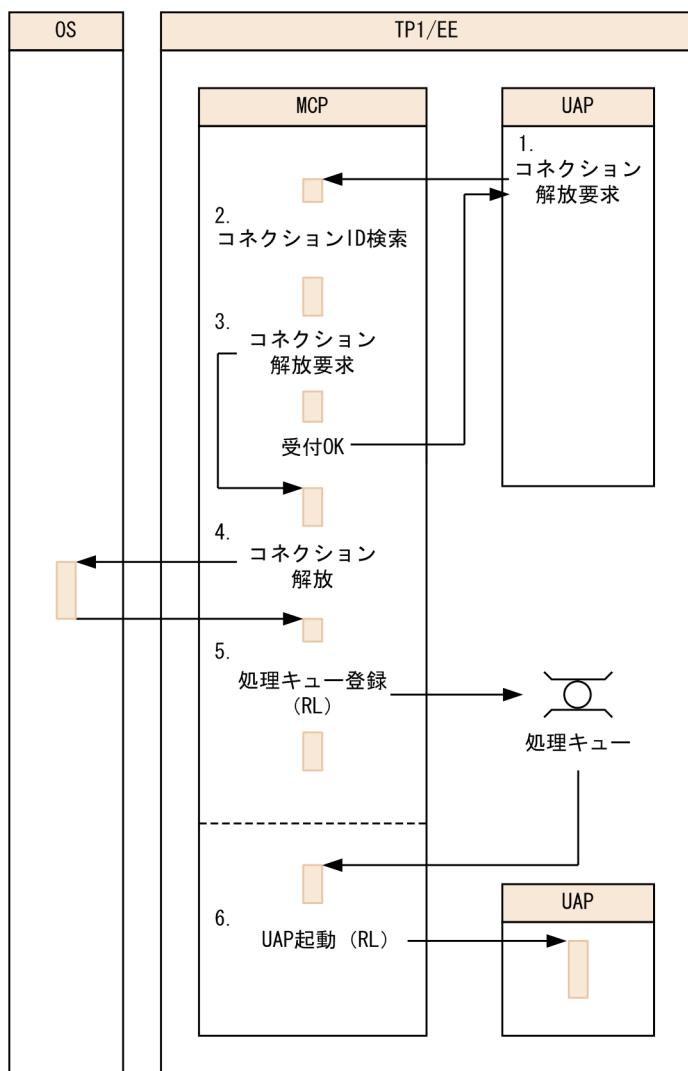
1. 障害を検知した場合、MCP はコネクションを解放します。
2. コネクション解放の RL (MCP 後処理トランザクション) を処理キューに登録します。
3. UAP にコネクションの解放を通知します。

(2) コネクションの手動解放

eemcpdctcn コマンドの実行、または ee_mcp_dctcn 関数の発行によって、コネクションを手動解放できます。

ee_mcp_dctcn 関数の発行によって、手動でコネクションを解放する場合の処理の流れを次の図に示します。

図 2-6 ee_mcp_dctcn 関数の発行によって、手動でコネクションを解放する場合の処理の流れ



(凡例)

→ : 制御の流れ

RL : MCP後処理トランザクション

説明

1. UAP は、MCP に対してコネクションの解放要求をします。
2. MCP は、コネクション管理テーブルでコネクション ID を検索し、解放要求のあったコネクションが確立中であることをチェックします。
3. MCP は、コネクションの解放要求をします。
4. コネクションが解放されます。

5. 解放要求のあったコネクションが解放されると、コネクション確立の RL (MCP 後処理トランザクション) を処理キューに登録します。
6. UAP にコネクションの解放を通知します。

■コネクションの解放形態

自システム側からコネクションを解放する場合、MCP では eemcpcn 定義コマンドの-f オプションの cnrelease オペランドの指定値によって、コネクションの解放形態を決定します。

なお、TP1/EE 終了（正常終了、計画停止 A、および計画停止 B）時のコネクション解放は、eemcpcn 定義コマンドの-f オプションの endrelease オペランドの指定値によって、コネクションの解放形態を決定します。ただし、CL 構成の場合は、cnrelease オペランドで指定したコネクション解放形態でコネクションを解放します。

解放形態には、FIN パケットを送信してコネクションを解放する形態と、RST パケットを送信して強制的にコネクションを解放する形態があります。

■ 注意事項

MCP は OS が提供する TCP/IP のソケットインターフェースを利用しています。そのため、FIN パケットを送信するように指定した場合でも、ネットワーク上のすれ違いなどによって、MCP の指示とは関係なく RST パケットを送信する場合があります。

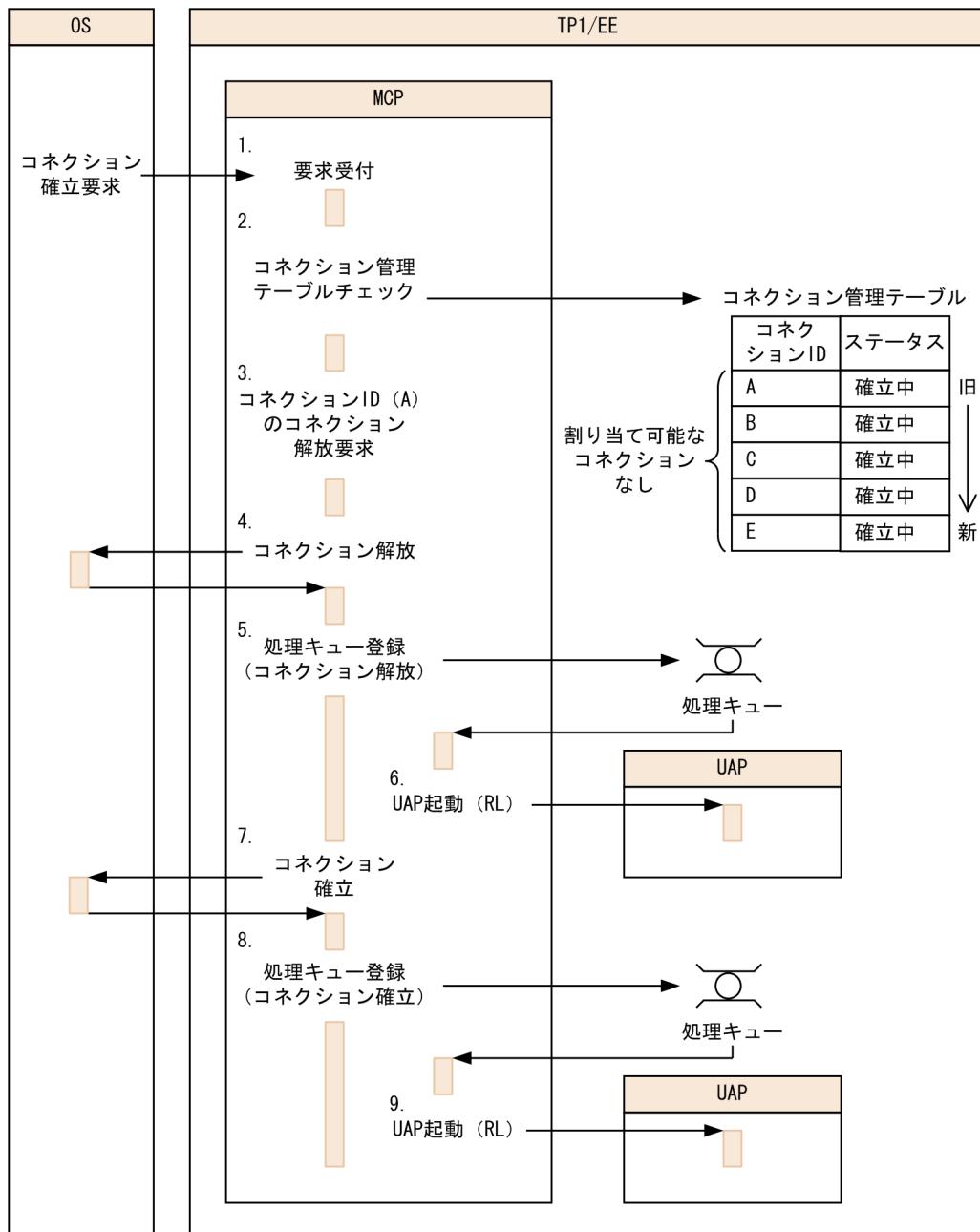
2.1.4 コネクションリプレース機能

コネクションリプレース機能とは、割り当て可能な未確立コネクションがない状態でコネクションの確立要求を受け付けたときに、確立状態にあり、かつ送信処理で使用していない最も古いコネクションを解放することで新たなコネクションの確立要求を受け付ける機能です。この機能を使用していない場合に、割り当て可能な未確立コネクションがない状態でコネクションの確立要求を受け付けたときは、コネクションの確立要求を拒否します。

コネクションリプレース機能は、自システムの IP アドレスとポート番号が同じであるコネクションが、すべて確立状態にあるときだけ使用できます。

コネクションリプレース機能を使用するときの処理の流れを次の図に示します。

図 2-7 コネクションリプレース機能を使用するときの処理の流れ



(凡例)

→ : 制御の流れ

RL : MCP後処理トランザクション

説明

1. 相手システムからのコネクション確立要求を受け付けます。
2. MCPは、コネクション管理テーブルを検索し、未確立のコネクションがあるかどうかをチェックします。
3. コネクション管理テーブルを調査した結果、割り当て可能なコネクションがない場合、送信処理で使用していない最も古いコネクション（コネクションID（A））の解放要求をします。

4. コネクションを解放します。
5. コネクション ID (A) のコネクション解放による RL (MCP 後処理トランザクション) を処理キューに登録します。
6. UAP を起動してコネクション解放を通知します。
7. 確立要求のあったコネクションをコネクション ID (A) で確立します。
8. コネクション ID (A) のコネクション確立による RL (MCP 後処理トランザクション) を処理キューに登録します。
9. UAP を起動してコネクションの確立を通知します。

■ 注意事項

eemcpcn 定義コマンドの指定値によっては、コネクションリプレース機能を使用できません。コネクションリプレース機能を使用できない場合は、次に示すとおりです。

- -o オプションの oportno オペランドに 1~65535 の値を指定した場合
- -r オプションの portno オペランドが一致するすべてのサーバ型コネクションで、-h オプションの chgconn オペランドの値が異なる場合

次の場合には、自システムの IP アドレスおよびポート番号が同じコネクションを複数定義できます。

- ポートフリーのコネクション
- 相手アドレスのチェックを抑止しているコネクション

■ ポートフリーのコネクション

eemcpcn 定義コマンドの-o オプションの oportno オペランドに free を指定した、ポートフリーのコネクションに対してコネクションリプレース機能を使用できます。この場合は、自システムの IP アドレスおよびポート番号が同じコネクションの中から、確立状態にあり、かつ送信処理で使用していない最も古いコネクションをコネクションリプレースの対象として選択します。

送信処理で使用していないコネクションが一つもない場合は、新たなコネクションの確立要求を拒否します。

■ 相手アドレスのチェックを抑止しているコネクション

eemcpcn 定義コマンドの-h オプションの addrchk オペランドに no を指定して、相手アドレスチェックを抑止したコネクションに対してコネクションリプレース機能を使用できます。この場合は、次に示す二つの条件と一致し、かつ確立状態にあり、送信処理で使用していない最も古いコネクションをコネクションリプレースの対象として選択します。

- 接続している既存のコネクションの相手アドレスと、確立要求をした相手のアドレスが同じである
- 自システムの IP アドレスおよびポート番号が同じである

なお、該当するコネクションがない場合、ポートフリーのコネクションの場合と同様に自システムのIPアドレスおよびポート番号が同じコネクションの中から、送信処理で使用していない最も古いコネクションをコネクションリプレースの対象として選択します。

送信処理で使用していないコネクションが一つもない場合は、新たなコネクションの確立要求を拒否します。

2.1.5 無通信状態監視

無通信状態とは、任意のコネクションに、相手システムとの間で、メッセージの送信または受信がない状態です。MCPは、この無通信状態を監視します。無通信状態の監視時間は、MCP構成定義のコネクション定義eemcpcnオペランドの-kオプションのnotrftimeで指定します。指定した時間を超過した場合、何らかの障害が発生しているものとして、コネクションを強制解放します。

コネクション障害を検出する機能としてキープアライブがあります。キープアライブがソケットオプション「SO_KEEPALIVE」を使用して一定間隔でパケットを送信しながらコネクションの状態を監視するのに対し、無通信状態監視機能はパケットを送信しないでコネクションの状態を監視します。

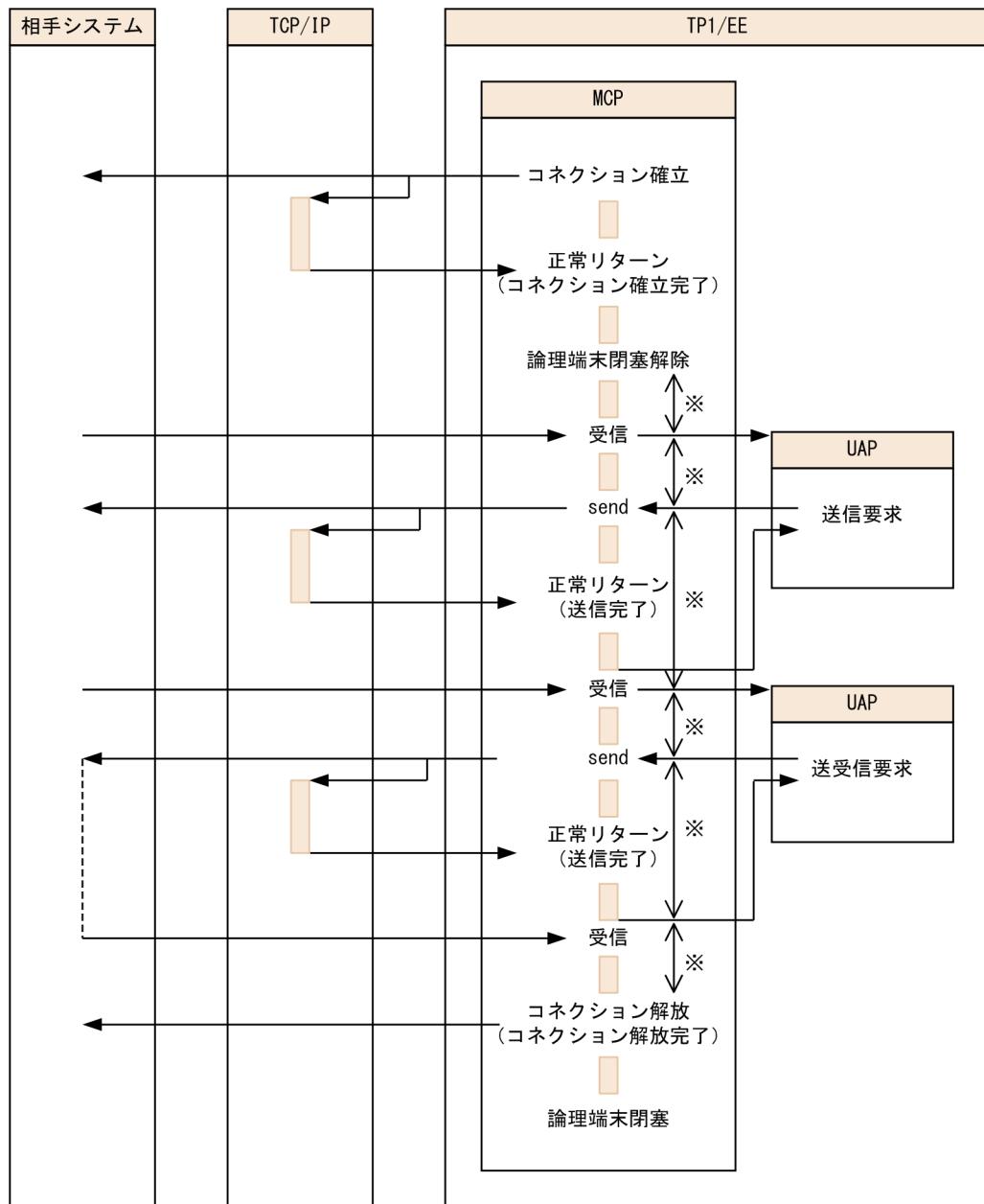
なお、無通信状態の監視時間がタイムアウトした場合は、KFSB41481-Eが出力されます。

MCPは、各コネクションに対して該当の論理端末が閉塞解除された時点から、論理端末の閉塞またはTP1/EEの終了（正常終了/計画停止A/計画停止B）までを監視します。また、メッセージを受信、または送信すると、状態監視をリセットし、改めて次のメッセージの受信または送信までの監視を開始します。

無通信状態監視タイマは、ほかの各種タイマ（後続メッセージタイマやメッセージ送信完了待ちタイマ、メッセージ送受信完了タイマなど）と並行して、無通信状態を監視します。

無通信状態を監視する範囲を、一方受信メッセージを受信、一方送信メッセージを送信、同期メッセージを送受信した場合を例に、次の図に示します。

図 2-8 無通信状態監視タイマ (TCP/IP)



(凡例)

→ : 制御の流れ

注※ : 無通信状態監視タイマ

2.2 論理端末の管理

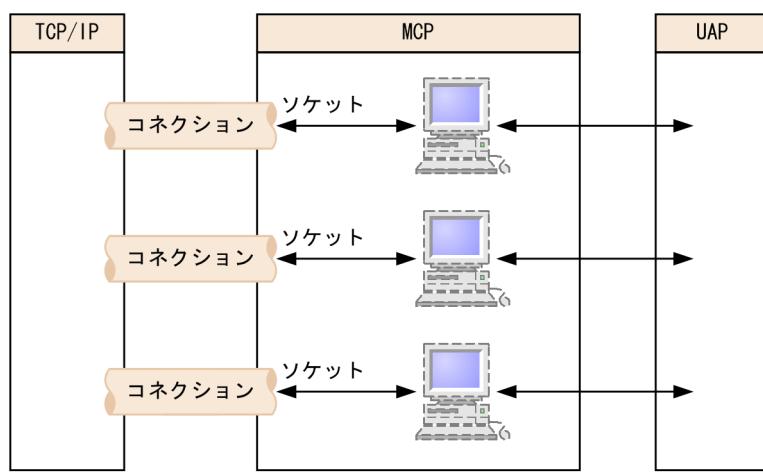
ここでは、MCP で TCP/IP プロトコルおよび UDP プロトコルを使用して通信する場合の、論理端末の管理について説明します。

2.2.1 コネクションと論理端末の関係

TCP/IP プロトコルでは、コネクションへのインターフェースとして論理端末を提供します。コネクションは、TCP/IP プロトコルでのソケットに対応し、論理端末とは 1 対 1 の関係にあります。

コネクションと論理端末の関係を次の図に示します。

図 2-9 コネクションと論理端末の関係



なお、UDP プロトコルを使用したメッセージ通信では、論理端末を通じて自システムの UAP と相手システムとの間でメッセージを送受信します。コネクションレスの通信を前提にしているため、ユーザがコネクションを意識する必要はありません。

また、TP1/EE 終了時（正常終了、計画停止 A、計画停止 B）は、MCP 構成定義（eemcpcn 定義コマンドの-f オプションの endrelease オペランド）の指定によって、FIN パケットまたは RST パケットを送信してコネクションを解放します。

コネクションの解放契機ごとの解放形態を次の表に示します。

表 2-2 コネクションの解放契機ごとの解放形態

解放契機	eemcpcn 定義コマンドの-f オプションの指定値		解放状態
	cnrelease	endrelease	
コネクション障害	fin	fin	FIN
	fin	rst	FIN

解放契機	eempcn 定義コマンドの-f オプションの指定値		解放状態
	cnrelease	endrelease	
コネクション障害	rst	fin	RST
	rst	rst	RST
関数 (ee_mcp_dctcn(), または CBLEEMCP('DCTCN')) の発行	fin	fin	FIN
	fin	rst	FIN
	rst	fin	RST
	rst	rst	RST
eemcpdctcn コマンド (-f オプション指定なし) の実行	fin	fin	FIN
	fin	rst	FIN
	rst	fin	FIN
	rst	rst	FIN
eemcpdctcn コマンド (-f オプション指定あり) の実行	fin	fin	FIN
	fin	rst	FIN
	rst	fin	RST
	rst	rst	RST
TP1/EE 正常終了	fin	fin	FIN
	fin	rst	RST
	rst	fin	FIN
	rst	rst	RST
TP1/EE 計画停止 (計画停止 A, または計画停止 B)	fin	fin	FIN
	fin	rst	RST
	rst	fin	FIN
	rst	rst	RST
TP1/EE 異常終了 (プロセスダウン)	fin	fin	FIN
	fin	rst	FIN
	rst	fin	RST
	rst	rst	RST

(凡例)

FIN : FIN パケットを送信してコネクションを解放します。

RST : RST パケットを送信してコネクションを解放します。

2.2.2 論理端末の状態と状態遷移

論理端末の状態は、コマンドの入力や通信障害などのイベントによって遷移します。このとき、MCP は MCP 後処理トランザクションを起動し、論理端末の状態変化を UAP に通知します。MCP 後処理トランザクションについては、「[2.3 MCP 後処理トランザクション](#)」を参照してください。論理端末は、次の表に示す状態で管理されます。

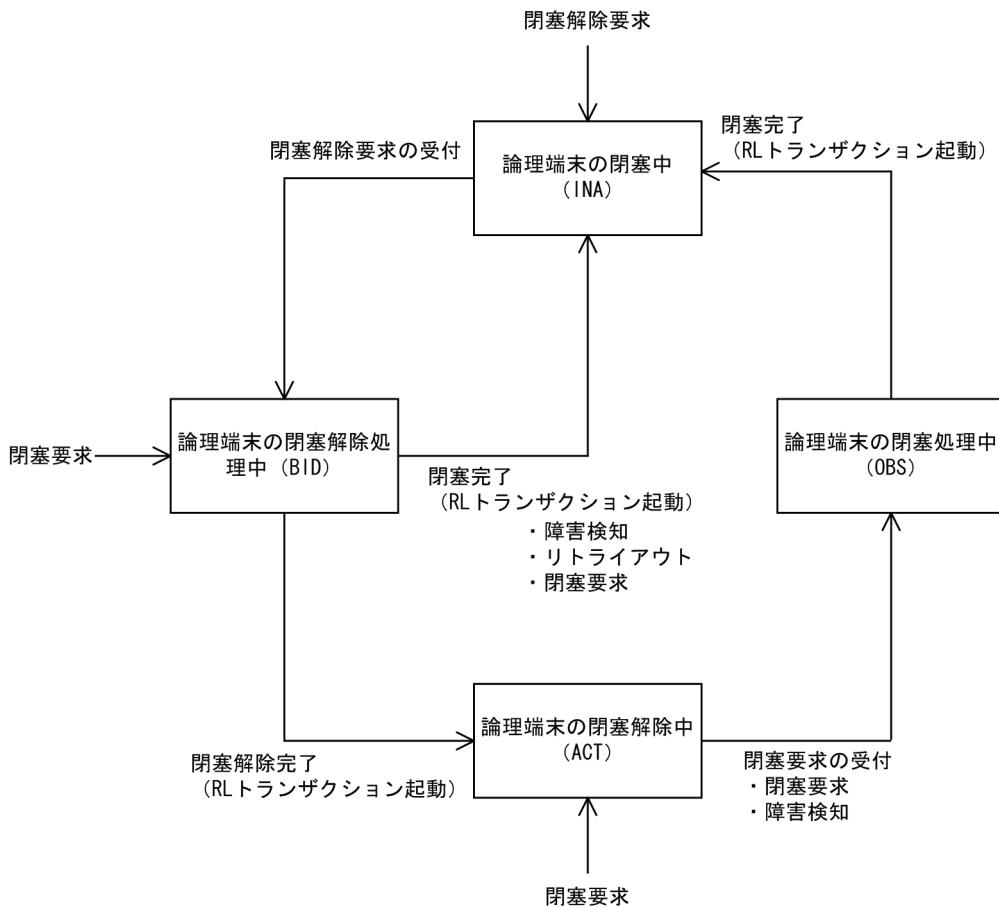
表 2-3 論理端末の状態の意味

項目番	状態名	意味
1	論理端末の閉塞中 (INA)	論理端末の閉塞解除要求をしていない状態（論理端末障害で停止中も含む）
2	論理端末の閉塞解除処理中 (BID)	論理端末の閉塞解除要求を受け付けてから、論理端末の閉塞解除完了の MCP 後処理トランザクションが起動する前までの状態
3	論理端末の閉塞解除中 (ACT)	論理端末の閉塞解除完了の MCP 後処理トランザクションが起動している状態（論理端末でメッセージ送受信ができる状態）
4	論理端末の閉塞処理中 (OBS)	論理端末の閉塞要求（障害検知も含む）を受け付けてから、論理端末閉塞の MCP 後処理トランザクションが起動する前までの状態

論理端末にかかるエラーメッセージが出力された場合、論理端末の状態を確認する必要があります。論理端末の状態は、eemcplsls コマンドで表示できます。また、eemcplscn コマンドに-d オプションを指定すると、コネクションに対応する論理端末の情報が表示できます。

論理端末の状態遷移を次の図に示します。

図 2-10 論理端末の状態遷移



2.2.3 論理端末の閉塞と閉塞解除

自システムの UAP は、論理端末を通じてメッセージの送受信を行います。ここでは、論理端末の閉塞と閉塞解除について説明します。

(1) 論理端末の状態

論理端末の閉塞状態とは、相手システムにメッセージを送信できない状態を指します。

論理端末が閉塞している状態で UAP がメッセージの送信を要求した場合、出力先論理端末障害としてエラーリターンします。また、非同期一方送信をする場合、メッセージ送信時に論理端末が閉塞状態になったときは、送信障害トランザクション (ERRTRNS) を起動します。

一方、論理端末の閉塞解除状態とは、論理端末が持つすべての機能が実行できる状態を指します。

(2) 論理端末の閉塞

論理端末を閉塞するには、次に示す方法があります。

- ・ コネクションを解放する (TCP/IP プロトコルの場合)

コネクションを解放すると、論理端末は自動閉塞されます。

- 運用コマンドを実行する、または API を発行する

eemcpdctle コマンドの実行、または ee_mcp_dctle 関数の発行によって、論理端末を手動閉塞できます。

注意事項

次の場合、論理端末は MCP によって自動閉塞されます。

- 入力メッセージ編集 UOC がエラーリターンまたはパラメタ不正を検知した場合
- 出力メッセージ編集 UOC がエラーリターンまたはパラメタ不正を検知した場合

なお、UOC の異常で MCP が論理端末を自動閉塞する場合は、論理端末障害の MCP 後処理 トランザクションを起動します。この場合、入出力メッセージは破棄されます。

(3) 論理端末の閉塞解除

論理端末を閉塞解除するには、次に示すとおり、自動で閉塞解除する方法と、手動で閉塞解除する方法があります。

- コネクション確立時に自動的に閉塞解除する (TCP/IP プロトコルの場合)

eemcple (論理端末定義 (TCP/IP)) 定義コマンドの-i オプションに auto を指定します。これによって、コネクション確立時に自動的に論理端末が閉塞解除されます。

- オンライン開始時および再開始時に自動的に閉塞解除する (UDP プロトコルの場合)

eemcple (論理端末定義 (UDP)) 定義コマンドの-i オプションに auto を指定します。これによって、オンライン開始時および再開始時に自動的に論理端末が閉塞解除されます。なお、論理端末の閉塞解除が成功した場合は、論理端末の閉塞解除を通知する MCP 後処理トランザクションは起動しません。論理端末の状態は、KFSB81404-I メッセージまたは、eemcplsle コマンドで確認してください。

- 運用コマンドまたは API を発行して手動で閉塞解除する

eemcpactle コマンドの実行、または ee_mcp_actle 関数の発行によって、手動で論理端末を閉塞解除します。

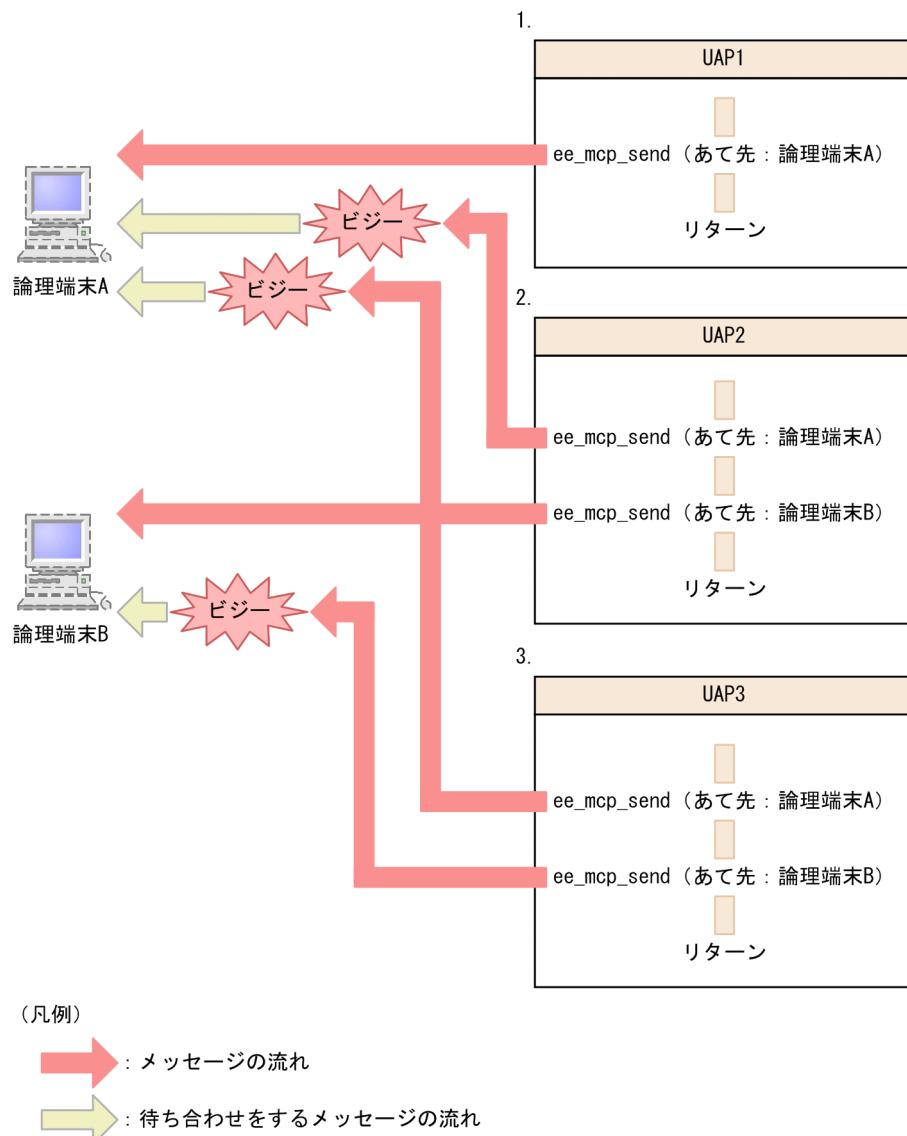
2.2.4 送信処理中の論理端末に対するメッセージ送信

複数のサービスで同一の論理端末にメッセージを送信した場合、先着のメッセージによって論理端末が送信処理中になるおそれがあります。このとき、送信対象の論理端末が複数ある場合は、送信処理中の論理端末に対するメッセージ送信をスキップして、先に送信可能な別の論理端末へメッセージを送信します。

送信対象の論理端末が一つしかない場合、または送信対象の論理端末がすべて送信処理中になっている場合は、論理端末がメッセージを受け付けられるようになった時点でメッセージを送信します。

同一の論理端末に対して複数のサービスがメッセージを送信した場合の処理の流れを次の図に示します。

図 2-11 同一の論理端末に対して複数のサービスがメッセージを送信した場合の処理の流れ



説明

1. UAP1 は、論理端末 A に対してメッセージを送信します。
2. UAP1 のメッセージ送信によって論理端末 A が送信処理中になっているため、UAP2 は、先に論理端末 B に対してメッセージを送信します。論理端末 A が空き次第、メッセージを送信します。
3. UAP1、および UAP2 によってあて先の論理端末すべてが送信処理中になっているため、UAP3 は、論理端末 A、および論理端末 B が空き次第、メッセージを送信します。

2.3 MCP 後処理トランザクション

MCP 後処理トランザクションは、コネクションおよび論理端末の状態が変化した場合に、その変化を UAP に通知するための機能です。この機能を使用すると、MCP は次に示す場合に MCP 後処理トランザクションを起動して、状態の変化を UAP に通知します。

- ・コネクションの確立または解放
- ・論理端末の閉塞または閉塞解除

MCP 後処理トランザクションの起動は、MCP 構成定義で設定します。論理端末ごとに MCP 後処理トランザクションを設定する場合は、eemcpfunc 定義コマンドの-u オプションの rltn オペランドにサービス名 (TP1/EE サービス定義の service オペランドで指定) を指定します。論理端末ごとに設定しない場合は、eemcpfunc 定義コマンドの-u オプションの rltn オペランドにサービス名を指定します。

トランザクションレベル方式指定時に MCP 後処理トランザクション起動サービス名を指定する場合は、eemcpfunc 定義コマンド、または eemcpfunc 定義コマンドの-u オプションの rltrnlevel オペランドに、MCP 後処理トランザクションのトランザクションレベル名を必ず指定してください。

MCP 後処理トランザクションは、該当するサービスが閉塞中であってもサービスを起動します。ただし、MCP 後処理トランザクションで発生した障害に対するエラートランザクション (ERRTRN3) は起動しません。

また、サービス閉塞による処理キュー引き出し禁止 (TP1/EE のユーザサービス関連定義の forbid_draw_service オペランドに Y を指定) を指定している場合、引き出し禁止が解除されるまで MCP 後処理トランザクションは起動しません。

MCP 後処理トランザクションの起動タイミングと内容について次の表に示します。

表 2-4 MCP 後処理トランザクションの起動タイミングと内容

項目番号	MCP 後処理トランザクションの起動タイミング	内容
1	コネクション確立通知*	<p>クライアント側 コマンドまたは API によってコネクションを確立したあと、コネクションが確立されたことを UAP に通知します。</p> <p>サーバ側 相手システムからのコネクション確立要求を受け付けたあと、コネクションが確立されたことを UAP に通知します。</p>
2	コネクション解放通知	コマンドまたは API によってコネクションを解放したあと、コネクションが解放されたことを UAP に通知します。
3		コネクション障害を検知したあと、コネクションが解放されたことを UAP に通知します。
4		入力セグメント判定 UOC の結果、不正を検知したあと、コネクションが解放されたことを UAP に通知します。

項目番号	MCP 後処理トランザクションの起動タイミング	内容
5	論理端末閉塞通知	コマンドまたは API によって論理端末を閉塞したあと、論理端末が閉塞されたことを UAP に通知します。
6		入力メッセージ編集 UOC の結果、不正を検知したあと、論理端末が閉塞されたことを UAP に通知します。
7		出力メッセージ編集 UOC の結果、不正を検知したあと、論理端末が閉塞されたことを UAP に通知します。
8	論理端末閉塞解除通知*	コマンドおよび API によって閉塞を解除したあと、論理端末の閉塞が解除されたことを UAP に通知します。

注※

オンライン開始時にコネクションの自動起動を定義した場合 (eemcpcn 定義コマンドの-i オプションに auto を指定)、または論理端末の自動閉塞解除を定義した場合 (eemcple 定義コマンドの-i オプションに auto を指定)、オンライン開始時の自動起動による MCP 後処理トランザクションは起動しません。

注意事項

TP1/EE、および高速メッセージ通信を可能にする XTC の運用コマンドの中には、MCP 後処理トランザクションで発行できないコマンドがあります。詳細については、「[付録 C 運用コマンド実行機能での MCP コマンドの実行可否](#)」を参照してください。

2.4 エラートランザクション

MCP は、メッセージ送受信の処理でエラーや障害が起こった場合、ユーザによる決着処理ができるようにエラートランザクションを起動します。

エラートランザクションを起動するアプリケーションプログラムは MCP 構成定義で設定します。論理端末ごとにエラートランザクションを設定する場合は、eemcpfunc 定義コマンドの-e オプションの errtrn1, errtrn2, errtrn3, errtrnr, および errtrns オペランドにそれぞれのエントリポインタ名 (C 言語の関数名) を指定します。

論理端末ごとに設定しない場合は、TP1/EE サービス定義の errtrn1, errtrn2, errtrn3, errtrnr, および errtrns オペランドの指定値に従います。

なお、論理端末ごとに設定しない場合は、eemcpfunc 定義コマンドの-e オプションの errtrns オペランドで、送信障害トランザクション (ERRTRNS) を起動するかどうかを指定できます。

エラートランザクションの詳細については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

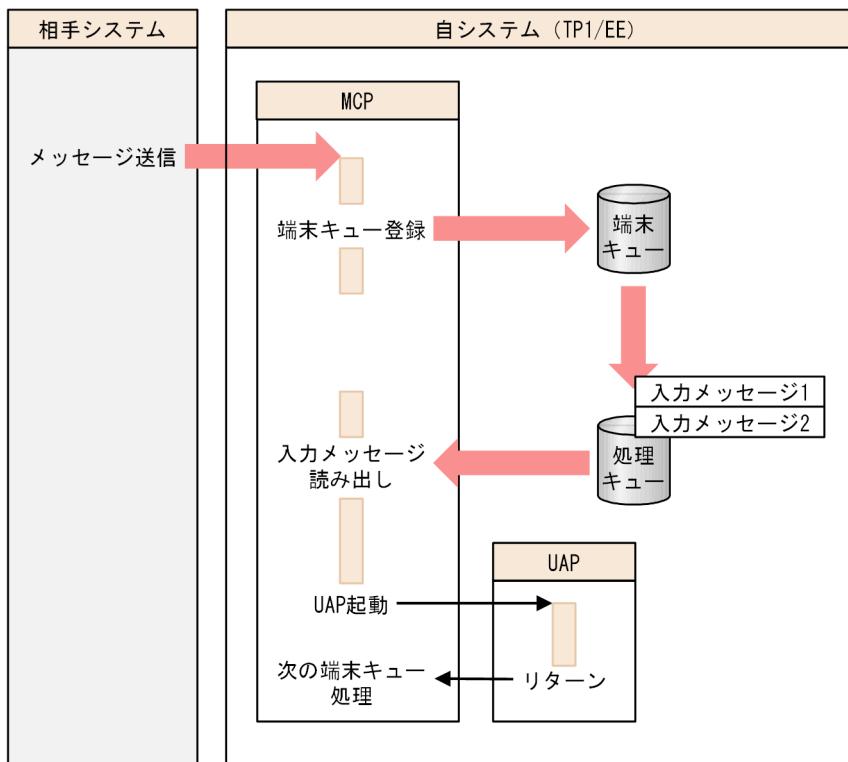
2.5 端末キュー機能

MCP では、端末キュー機能によって、論理端末単位のイベントをシリアルに処理できます。論理端末単位のイベントには、メッセージ受信、コネクション確立・解放、および論理端末の閉塞・閉塞解除があります。

端末キュー機能を使用するかどうかは、eemcpfunc 定義コマンドの-q オプションで指定します。

端末キュー機能を使用した場合の処理の流れを次の図に示します。

図 2-12 端末キュー機能を使用した場合の処理の流れ



(凡例)

→ : メッセージの流れ

説明

1. 相手システムから受信した入力メッセージは、一度端末キューに登録されます。
2. 入力メッセージを端末キューに登録したあと、入力メッセージを該当するサービスの処理キューに登録します。
3. 処理キューからメッセージを読み出し、UAP を起動します。
4. 3.で起動された UAP がリターンすると、次のメッセージが端末キューから処理キューに登録されます。

コネクション、論理端末などの状態によって、端末キューの扱いは異なります。状態別の端末キューの扱いについて、次に説明します。

■コネクション障害時

コネクション障害が発生した場合は、該当するコネクションに対応する論理端末の端末キューに滞留しているすべてのメッセージを破棄します。ただし、すでに処理キューに登録されている入力メッセージは破棄されないで、UAP の処理を続行します。

■論理端末閉塞時 (TCP/IP プロトコル使用時)

TCP/IP プロトコルを使用している場合に論理端末が閉塞しているときは、該当する論理端末の端末キューを引き出し禁止にします。端末キューの引き出しは、論理端末を閉塞解除したタイミングで再開されます。

なお、論理端末が閉塞中のまま eesvstop コマンドを実行してオンラインを終了させた場合、端末キューに滞留中の入力メッセージは、すべて破棄されます。

■論理端末閉塞時 (UDP プロトコル使用時)

UDP プロトコルを使用している場合に論理端末が閉塞しているときは、TCP/IP プロトコル使用時のコネクション障害時と同様に、論理端末の端末キューに滞留しているメッセージを破棄します。

■オンライン終了時

端末キューの使用有無に関係なく、論理端末が閉塞している場合に eesvstop コマンドを実行してオンラインを終了させたときは、滞留しているメッセージを破棄したあとにオンラインを終了させます。

2.6 時間監視

ここでは、MCP での時間監視について説明します。

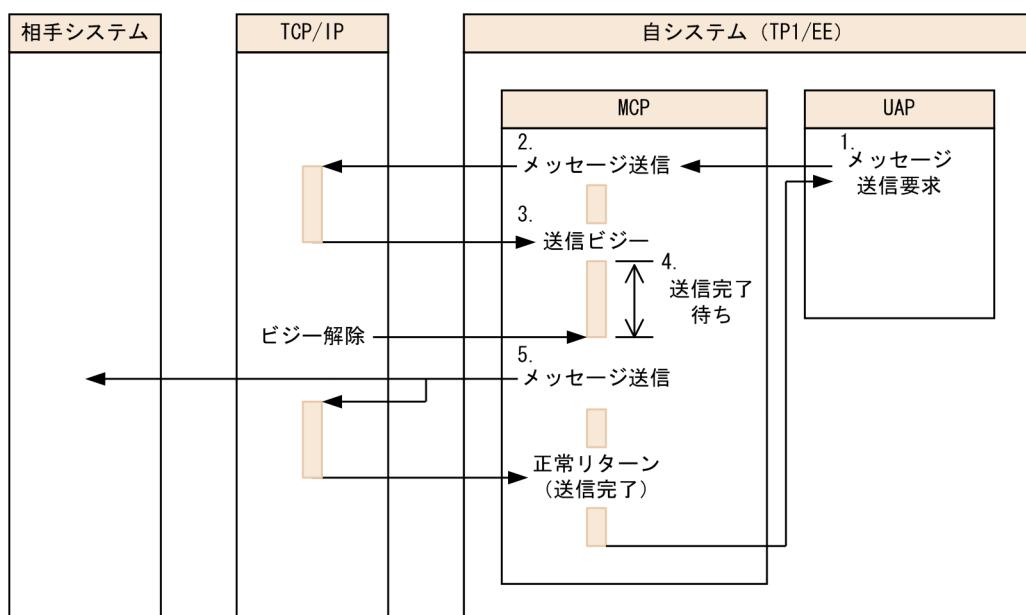
2.6.1 メッセージの送信完了待ち時間の監視 (TCP/IP プロトコル使用時)

メッセージの送信完了とは、ソケットの送信バッファにメッセージを格納完了した時点を指します。ee_mcp_send 関数でのメッセージ送信時に障害を検知した場合は、送信バッファの空き待ちによる送信バッファビジー状態であるため、バッファの空きを監視します。

送信完了待ち時間は、eemcpcn 定義コマンドの-b オプションの sndsynctim オペランド（非同期一方送信メッセージの場合は sndtim オペランド）で指定します。

メッセージの送信完了待ちをする場合の処理の流れを次の図に示します。

図 2-13 メッセージの送信完了待ちをする場合の処理の流れ



(凡例)

→ : 制御の流れ

説明

1. UAP は、MCP に対してメッセージの送信要求をします。
2. MCP は、UAP からのメッセージ送信要求に従い、相手システムにメッセージを送信します。
3. 相手先システムがビジー状態でメッセージが送信できなかった場合、相手システムが送信ビジー状態であることが MCP に通知されます。
4. MCP は、相手先システムの送信ビジーによってメッセージを送信できなかった場合、送信完了待ちタイマを設定し、相手システムのビジー状態が解除されるのを待ちます。

5. ビジー状態が解除されると、MCP は再度メッセージを送信します。

2.6.2 後続メッセージの時間監視 (TCP/IP プロトコル使用時)

入力セグメント判定 UOC で後続メッセージありと判断した場合に、UOC での設定内容に従って後続メッセージを監視します。監視時間は UOC で設定するため、定義での指定はできません。

2.7 TP1 キャッシュ機能との連携

TP1 キャッシュ機能を使用している場合、MCP を HA サーバ、および CL サーバの両方に組み込むことができます。

HA サーバに組み込んだ場合の MCP の機能は、TP1 キャッシュ機能を使用しない場合と同じです。ここでは、CL サーバに組み込んだ場合の MCP の機能について説明します。

なお、TP1 キャッシュ機能については、次に示すマニュアルを参照してください。

- TP1/Server Base Enterprise Option 使用の手引
- TP1/EE/Extended Transaction Controller 使用の手引

2.7.1 CL サーバでの MCP

CL サーバに組み込んだ場合は、HA サーバに MCP を組み込んだ場合と比較して、次に示すときの動作が異なります。

- コネクションの確立および論理端末の閉塞解除のタイミング
- CL サーバでのメッセージ送受信時

それぞれの動作について説明します。

(1) コネクションの確立および論理端末の閉塞解除

待機系で自動起動を設定したクライアント型コネクションは、オンライン開始前ではなく、実行系に切り替わった時点で確立されます。同様に、UDP プロトコルを使用している場合の論理端末では、実行系に切り替わった時点で閉塞が解除されます。

このとき、系切り替え前の実行系のコネクション、および論理端末の状態は引き継ぎません。

(2) CL サーバでのメッセージ送受信

CL サーバで、メッセージを送受信する場合の動作について説明します。

(a) メッセージ送信

ここでは、CL サーバでメッセージを送信する場合の動作について、非同期一方送信の場合と同期一方送信の場合に分けて説明します。

■非同期一方送信

CL サーバでのメッセージの送信では、送信するメッセージを待機系で保証するかどうかを選択できます。このとき、保証するメッセージを永続メッセージ、保証しないメッセージを非永続メッセージといいます。

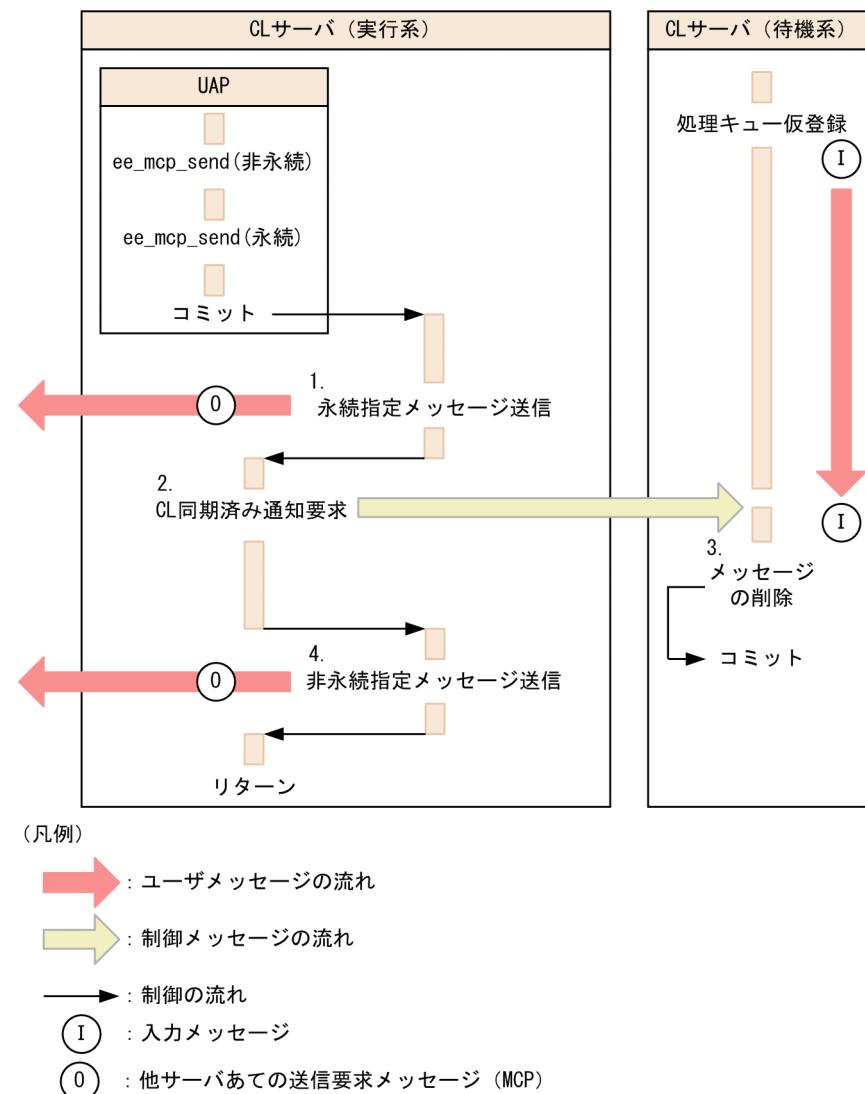
非同期一方送信の場合、`ee_mcp_send` 関数の発行時に、メッセージを永続、非永続のどちらにするか指定します。

永続メッセージの場合、メッセージが消失することはありませんが、メッセージを二重送信することができます。一方、非永続メッセージの場合、メッセージが消失することがあります、メッセージを二重送信することはありません。

永続メッセージを指定する場合は、永続トランザクション上で指定する必要があります。非永続トランザクション上で永続メッセージを指定すると、メッセージが消失することがあります。

CL サーバで、メッセージを非同期一方送信した場合の処理の流れを、次の図に示します。

図 2-14 CL サーバでメッセージを非同期一方送信した場合の処理の流れ



説明

1. 実行系は、永続メッセージを送信します。
2. 実行系は、待機系に対し CL 同期済み通知メッセージを送信します。
3. 待機系は、処理キューに仮登録された入力メッセージを削除します。
4. 実行系は、非永続メッセージを送信します。

図中の 1.から 3.までの処理で実行系がダウンした場合、系切り替え後に永続メッセージを再度送信することがあります。

また、図中の 3.から実行系での処理がリターンするまでに実行系がダウンした場合、送信メッセージが非永続メッセージであったときは、メッセージが消失します。

■同期一方送信

TP1 キャッシュ機能を使用しない場合と同じです。

(b) メッセージ受信

CL サーバでメッセージを受信した場合は、入力メッセージ編集 UOC の実行後に待機系に受信メッセージを転送します。これをメッセージの永続化といいます。受信メッセージを永続化するかどうかは、eemcpfunc 定義コマンドの-m オプションの recvpermanence オペランドで選択できます。

メッセージの転送処理については、マニュアル「TP1/EE/Extended Transaction Controller 使用の手引」を参照してください。

2.7.2 CL 連携時の定義チェック

CL 連携する場合 (XTC サービス定義の cluster_mode オペランドに Y を指定した場合) は、TP1/EE の開始時に実行系と待機系との間で、MCP 構成定義オブジェクトファイルの整合性をチェックします。チェックでの不一致を避けるため、実行系と待機系で同じ MCP 構成定義オブジェクトファイルを使用することをお勧めします。

整合性のチェックを行う定義の一覧を次の表に示します。

表 2-5 整合性のチェックを行う定義の一覧

定義名	コマンド	オプション・オペランド		定義チェックの有無
バッファ定義	eemcpbuf	-e	emsgbuf	○
TCP/IP 定義の開始	eemcptcpstart	-o	con_addr	○
論理端末アプリケーション定義	eemcplefunc	-u	ownlsg	○
			ownlmc	○
			ownlra	○
			ownlcn	○
			owntrn	○
			rltrn	○
		-e	errtrn1	○

定義名	コマンド	オプション・オペランド		定義チェックの有無
論理端末アプリケーション定義	eemcplefunc	-e	errtrn2	○
			errtrn3	○
			errtrnr	○
			errtrns	○
コネクション定義 (TCP/IP)	eemcpcn	-c	—	○
			-i	○
		-b	bretry	○
			bretrycnt	○
			bretryint	○
			concmptim	○
			conmsg	○
			sndsynctim	○
			sndtim	○
			sndrcvtim	○
			sndcmptim	○
		-w	srtimout	○
		-y	mode	○
		-r	ipaddr	○※
			hostname	○※
			portno	○※
		-o	oipaddr	○
			ohostname	○
			oportno	○
		-k	nodelay	○
			keepalive	○
			notrftime	○
		-f	cnrelease	○
			endrelease	○
		-h	addrchk	○
			chgconn	○

定義名	コマンド	オプション・オペランド		定義チェックの有無
論理端末定義 (TCP/IP)	eemcple	-l	—	○
		-i	—	○
		-v	—	○
TCP/IP 定義の終了	eemcptcpend	—	—	×
UDP 定義の開始	eemcpudpstart	—	—	×
論理端末定義 (UDP)	eemcple	-l	—	○
		-i	—	○
		-v	—	○
		-b	sndtim	○
		-s	syssndsize	○
			sysrcvsize	○
		-r	ipaddr	○
			hostname	○
			portno	○
		-o	oipaddr	○
			ohostname	○
			oportno	○
		-k	reuseaddr	○
		-m	multicast	○
			hostgroupaddr	○
			hostgroupname	○
			ripaddr	○
			rhostname	○
			multicastttl	○
スレッドグループ定義	eemcpthdgrp	—	—	×
UDP 定義の終了	eemcpudpPEND	—	—	×
アプリケーション定義	eemcpfunc	-u	ownlsg	○
			ownlmc	○
			ownlra	○
			ownlcn	○

定義名	コマンド	オプション・オペランド		定義チェックの有無
アプリケーション定義	eemcpfunc	-u	rltrn	○
			rltrnlevel	○
		-e	errtrns	○
		-c	uocexc	○
		-q	—	○
		-m	msglen	○
			recvpermanence	○
			reconnectsend	○
トレース定義	eemcptrc	-f	fileno	○
			filesz	○
		-b	bufferno	○
			buffersz	○
		-x	entrysize	○
		-s	sendresult	○
		-m	filter	○
			msgsize	○

(凡例)

- ：定義チェックをします。
- ×：定義オブジェクトファイルに含まれないため、定義チェックの対象外です。
- －：該当しません。

注※

クライアントの場合 (eemcpcn 定義コマンドの-y オプションの mode オペランドに client を指定) は、定義チェックしません。

注意事項

片方の系だけに定義を指定し、その指定した定義の値が省略値と同じ場合、システムとして同じ値が適用されるので定義チェックエラーとはなりません。

3

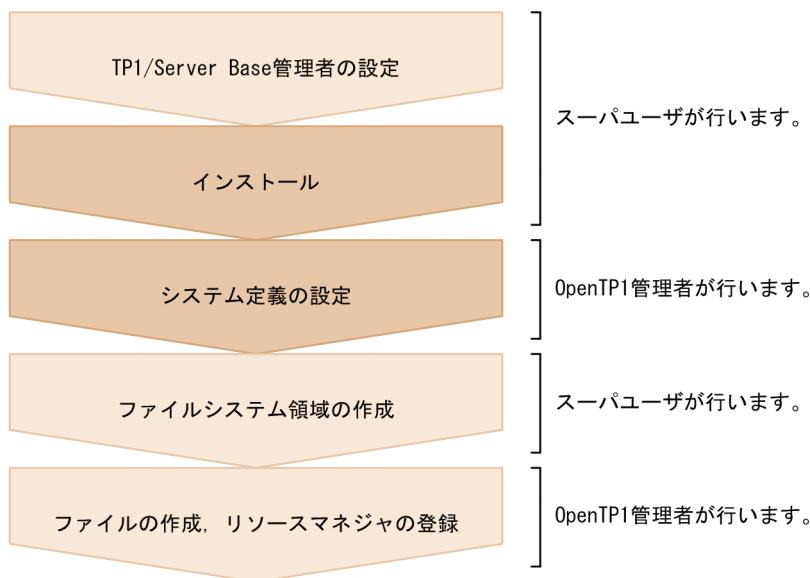
環境設定

この章では、MCP の環境設定とその手順について説明します。

3.1 環境設定の手順

MCP の環境設定の手順を次の図に示します。

図 3-1 MCP の環境設定の手順



(凡例)

: MCP固有の作業がない手順を示します。

: MCP固有の作業がある手順を示します。

ここでは、MCP を使用する場合に必要なインストール、およびシステム定義の設定の手順だけを説明します。ここで説明していない手順については、マニュアル「OpenTP1 運用と操作」、およびマニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。

3.1.1 インストール

インストールの手順を次の図に示します。

図 3-2 インストールの手順



(凡例)

: MCP固有の作業がない手順を示します。

: MCP固有の作業がある手順を示します。

1. TP1/Server Base のインストールについては、マニュアル「OpenTP1 運用と操作」を参照してください。
2. TP1/EE のインストールについては、マニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。
3. MCP をインストールします。MCP のインストールについては、リリースノートを参照してください。MCP をインストールすると、/opt/OpenTP1 ディレクトリ下にファイルがコピーされます。/opt/OpenTP1 ディレクトリの構成については、「[付録 A インストールディレクトリとファイル](#)」を参照してください。

3.1.2 システム定義の設定

システム定義の設定手順を次の図に示します。

図 3-3 システム定義の設定手順



(凡例)

: MCP固有の作業がある手順を示します。

1. TP1/Server Base のユーザサービス定義に、MCP 用の定義ファイル名称を定義します。

3. 環境設定

TP1/Server Base システムサービス定義の作成については、マニュアル「OpenTP1 システム定義」およびマニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。

2. MCP 組み込み定義、および MCP で使用するメモリを定義します。

TP1/EE サービス定義の作成については、マニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。

3. コネクション、論理端末などを定義します。

MCP 構成定義については、「[4. システム定義](#)」を参照してください。

4

システム定義

この章では、MCP を使用するために必要なシステム定義、定義オブジェクト、および定義例の作成方法について説明します。

4.1 システム定義の概要

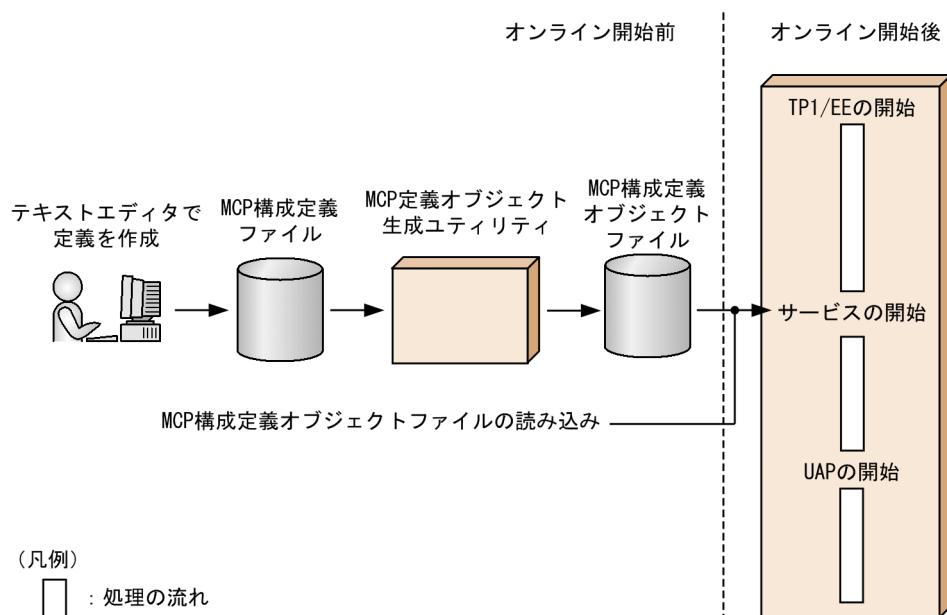
MCP のシステム定義を MCP 構成定義と呼びます。ここでは、MCP 構成定義の作成手順、定義の規則、および MCP 構成定義オブジェクトファイルの生成について説明します。

4.1.1 定義の作成手順

MCP 構成定義を作成するには、オンライン開始前にオブジェクトファイルを生成しておく必要があります。このオブジェクトファイルを MCP 構成定義オブジェクトファイルといいます。MCP 構成定義オブジェクトファイルは、テキストエディタで作成したソースファイルを、MCP オブジェクト生成ユーティリティで変換して生成します。

MCP 構成定義オブジェクトファイルの作成手順とオンライン開始時の処理の流れを次の図に示します。

図 4-1 MCP 構成定義オブジェクトファイルの作成手順とオンライン開始時の処理の流れ



4.1.2 定義の規則

定義の基本事項について説明します。

(1) 定義ファイルの格納場所

■MCP 構成定義ファイル

MCP 構成定義ファイルは、任意のディレクトリに作成してください。ファイル名は任意です。

■MCP 構成定義オブジェクトファイル

MCP 構成定義オブジェクトファイルは、環境変数 DCCONFPATH または DCUAPCONFPATH に指定したディレクトリに格納してください。ファイル名は任意です。

なお、環境変数 MCPDEFFILE には、MCP 構成定義オブジェクトファイルの名称を指定する必要があります。

■ 注意事項

DCCONFPATH、およびDCUAPCONFPATH の両方が定義されていて、両方のディレクトリに MCP 構成定義オブジェクトファイルが格納されている場合は、DCCONFPATH に指定されたディレクトリ下の MCP 構成定義オブジェクトファイルが有効になります。

(2) 記述形式

MCP 構成定義ファイルの記述形式は、TP1/EE サービス定義ファイルの記述形式に従います。TP1/EE サービス定義ファイルの記述形式の詳細については、マニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。

(3) 記述順序

MCP 構成定義では、次の図に示すように、プロトコル (TCP/IP および UDP) 定義の開始・終了、コネクション定義、論理端末定義、およびスレッドグループ定義の記述順序が決まっています。

図 4-2 MCP 構成定義の記述順序



(凡例)

：枠の中の記述順序は変更できません。

：コネクションや論理端末を繰り返し定義する場合の記述順序です。

注※

eemcpudpstart (UDP定義の開始) 定義コマンドの直下では指定を省略できます。

論理端末アプリケーション定義 (eemcplefunc) の記述順序には、次に示すルールがあります。

- 論理端末アプリケーション定義 (eemcplefunc) は、TCP/IP 定義の開始 (eemcptcpstart) と TCP/IP 定義の終了 (eemcptcpPEND) の間、または UDP 定義の開始 (eemcpudpstart) と UDP 定義の終了 (eemcpudpend) の間に定義します。コネクション定義 (eemcpcn) と論理端末定義 (eemcple) の間には定義できません。
- 論理端末アプリケーション定義 (eemcplefunc) は、論理端末アプリケーション定義 (eemcplefunc) 以降の論理端末に対して有効になります。これは次の論理端末アプリケーション定義 (eemcplefunc) が出てくるまで、または TCP/IP 定義の終了 (eemcptcpPEND) もしくは UDP 定義の終了 (eemcpudpend) が出てくるまで有効です。
- 論理端末アプリケーション定義 (eemcplefunc) の前に定義した論理端末 (TCP/IP 定義の開始 (eemcptcpstart) または UDP 定義の開始 (eemcpudpstart) 後、論理端末アプリケーション定義 (eemcplefunc) を定義するまでの論理端末) はアプリケーション定義 (eemcpfunc) の値、または TP1/EE サービス定義の値が有効になります。

4.1.3 MCP 構成定義オブジェクトファイルの生成

MCP 定義オブジェクト生成ユーティリティを使用すると、MCP 構成定義の定義ファイルの構文チェック、および定義オブジェクトファイルの作成ができます。ここでは、MCP 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。

(1) 形式

```
eemcpogen -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
```

(2) 機能

MCP 構成定義の定義ファイルの構文をチェックして、定義オブジェクトファイルを作成します。なお、定義オブジェクトファイルを変更する場合は、必ず TP1/EE を終了させてください。TP1/EE の開始から再開始するまでの間に定義オブジェクトファイルを変更した場合、再開始ができないことがあります。再開始ができた場合でも動作の保証はできません。

(3) オプション

●-i [パス名] 入力ファイル名 ~ <パス名>

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>

定義オブジェクトを格納するファイル名を指定します。

(4) 出力メッセージ

メッセージ ID	内容	出力先
KFSB91401-E	領域の確保に失敗しました。	標準エラー出力
KFSB91403-E	コマンドの形式が不正です。	標準エラー出力
KFSB91404-E	フラグ引数が不正です。	標準エラー出力
KFSB91406-E	内部矛盾が発生しました。	標準エラー出力
KFSB91460-I	ヘルプメッセージ	標準出力
KFSB91461-E	定義解析中に領域の確保に失敗しました。	標準エラー出力
KFSB91462-E	定義解析中にファイルのオープンに失敗しました。	標準エラー出力
KFSB91463-E	定義解析中に定義ファイルの読み込みに失敗しました。	標準エラー出力
KFSB91464-E	定義形式が不正です。	標準エラー出力
KFSB91465-E	コマンド形式定義が不正です。	標準エラー出力
KFSB91466-E	定義間に不正があります。	標準エラー出力

メッセージID	内容	出力先
KFSB91467-E	省略できない定義が未定義です。	標準エラー出力
KFSB91468-I	定義オブジェクトファイルを出力しました。	標準出力
KFSB91469-E	ファイルの書き込みに失敗しました。	標準エラー出力
KFSB91470-Q	定義オブジェクトファイルの上書き確認。	標準出力
KFSB91472-E	オブジェクトファイルが不正です。	標準エラー出力

(5) 注意事項

MCP では MCP 定義オブジェクト生成ユティリティ (eemcpgen コマンド) を使用して、MCP 構成定義ファイルの構文をチェックします。eedefchk コマンドでは、MCP と TP1/EE の定義の関連はチェックしません。

CL 連携する場合 (XTC サービス定義の cluster_mode オペランドに Y を指定した場合) は、TP1/EE の開始時に実行系と待機系との間で MCP 構成定義オブジェクトファイルの整合性をチェックします。詳細については、「[2.7.2 CL 連携時の定義チェック](#)」を参照してください。

4.2 MCP 構成定義の一覧

MCP 構成定義の一覧を次の表に示します。

表 4-1 MCP 構成定義の一覧

定義名	コマンド	オプション・オペランド		定義内容	指定値((値範囲)) 《省略時解釈値》
バッファ定義	eemcpbuf	-e	emsgbuf	入力メッセージ編集 UOC および出力メッセージ編集 UOC で使用するメッセージ編集用バッファサイズ	符号なし整数 (単位: バイト) ((0~2147483647)) 《0》
TCP/IP 定義 の開始	eemcpcpstart	-o	con_addr	クライアント型コネクションの場合、相手システムを動的に指定してコネクションを確立するかどうか	dynamic 《static》
コネクション定義 (TCP/IP)	eemcpcn	-c	—	コネクション ID	1~8 文字の識別子
		-i	—	オンライン開始時および再開始時のコネクションの確立方法	auto 《manual》
		-b	bretry	コネクション確立時に障害が発生した場合にコネクション確立再試行をするかどうか	《yes》 no
			bretrycnt	コネクション確立時に障害が発生した場合のコネクション確立再試行の回数	符号なし整数 (単位: 回) ((0~65535)) 《0》
			bretryint	コネクション確立時に障害が発生した場合のコネクション確立再試行の間隔	符号なし整数 (単位: 秒) ((0~2550)) 《60》
			concmptim	コネクション確立時の監視時間	符号なし整数 (単位: 秒) ((0~65535)) 《0》
			conmsg	コネクション確立時または解放時、メッセージログに確立または解放のメッセージを出力するかどうか	《yes》 no

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》
コネクション定義 (TCP/IP)	eempcn	-b	sndsynctim	同期一方送信時のメッセージ送信完了待ち時間	符号なし整数 (単位:秒) ((0~65535)) 《0》
			sndtim	非同期一方送信時のメッセージ送信完了待ち時間	符号なし整数 (単位:秒) ((0~65535)) 《0》
			sndrcvtim	同期型送受信時のメッセージ送受信完了待ち時間	符号なし整数 (単位:秒) ((0~65535)) 《0》
			sndcmptim	同期型送受信時のメッセージ送信完了監視時間	符号なし整数 (単位:秒) ((0~65535)) 《0》
		-w	srtimout	同期型メッセージ送受信関数がタイムアウトした場合に、コネクションの解放を抑止するかどうかを指定	yes 《no》
		-y	mode	コネクション確立モード	client server
		-r	ipaddr*	自システムのホストのIPアドレス	符号なし整数 ((0~255)) (nnn.nnn.nnn.nnn)
			hostname*	自システムのホスト名	1~255文字のホスト名
			portno	メッセージ送受信に使用する自システムのホストのポート番号	符号なし整数 ((1024~65535))
		-o	oipaddr	相手システムのホストのIPアドレス	符号なし整数 ((0~255)) (nnn.nnn.nnn.nnn)
			ohostname	相手システムのホスト名	1~255文字のホスト名
			oportno	相手システムのホストのポート番号	符号なし整数 ((1~65535)) 《free》
		-k	nodelay	ソケットオプション「TCP_NODELAY」を使用するかどうか	yes 《no》

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》
コネクション定義 (TCP/IP)	eemcpcn	-k	keepalive	ソケットオプション「SO_KEEPALIVE」を使用するかどうか	yes 《no》
			notrftime	無通信監視時間	符号なし整数 (単位:秒) ((0~65535)) 《0》
		-f	cnrelease	コネクションの解放形態	《fin》 rst
			endrelease	終了時のコネクションの解放形態	《fin》 rst
		-h	addrchk	コネクション確立要求を受け付けた場合、相手アドレス情報をチェックするかどうか	《yes》 no
			chgconn	コネクションの確立要求を受け付けた場合、割り当てできる未確立コネクションがないときの動作	replace 《keep》
		-l	—	論理端末名	1~8 文字の識別子
		-i	—	論理端末の閉塞解除方法	《auto》 manual
		-v	—	入力メッセージを受信した場合に起動するエントリポインタ名に対応したサービス名	1~31 文字の識別子
TCP/IP 定義の終了	eemcptcpnd	—	—	TCP/IP 定義の終了	—
UDP 定義の開始	eemcpudpstart	—	—	UDP 定義の開始	—
論理端末定義 (UDP)	eemcple	-l	—	論理端末名	1~8 文字の識別子
		-i	—	オンライン開始時および再開始時に論理端末を自動的に閉塞解除するかどうか	auto 《manual》
		-v	—	入力メッセージを受信した場合に起動するエントリポインタ名に対応したサービス名	1~31 文字の識別子
		-b	sndtim	メッセージ送信完了待ち時間	符号なし整数 (単位:秒)

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》
論理端末定義 (UDP)	eemcple	-b	sndtim	メッセージ送信完了待ち時間	((0~65535)) 《0》
		-s	syssndsize	システムのメッセージ送信バッファ長	符号なし整数 (単位:バイト) ((1~2147483647))
		sysrcvsize		システムのメッセージ受信バッファ長	符号なし整数 (単位:バイト) ((1~2147483647))
		-r	ipaddr*	自システムがホットスタンバイ構成の場合、または複数の LAN アダプタを持つ場合に、メッセージの送受信に使用する LAN アダプタの IP アドレス	符号なし整数 ((0~255)) (nnn.nnn.nnn.nnn)
			hostname*	自システムがホットスタンバイ構成の場合、または複数の LAN アダプタを持つ場合に、メッセージの送受信に使用する LAN アダプタのホスト名	1~255 文字のホスト名
			portno	自システムのホストのポート番号	符号なし整数 ((1024~65535))
		-o	oipaddr	相手システムのホストの IP アドレス	符号なし整数 ((0~255)) (nnn.nnn.nnn.nnn)
			ohostname	相手システムのホスト名	1~255 文字のホスト名
			oportno	相手システムのホストのポート番号	符号なし整数 ((1~65535))
		-k	reuseaddr	ソケットオプション「SO_REUSEADDR」を使用するかどうか	yes 《no》
		-m	multicast	マルチキャストを使用したメッセージの送受信をするかどうか	yes 《no》
			hostgroupaddr	マルチキャストメッセージを受信する場合、参加するホストグループの IP アドレス	符号なし整数 ((0~255)) (nnn.nnn.nnn.nnn)

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》
論理端末定義 (UDP)	eemcple	-m	hostgroupname	マルチキャストメッセージを受信する場合、 参加するホストグループのホスト名	1~255 文字のホスト名
			ripaddr*	自システムがホットスタンバイ構成の場合、 または複数の LAN アダプタを持つ場合、 マルチキャストメッセージの受信に使用する LAN アダプタの IP アドレス	符号なし整数 ((0~255)) (nnn.nnn.nnn.nnn)
			rhostname*	自システムがホットスタンバイ構成の場合、 または複数の LAN アダプタを持つ場合、 マルチキャストメッセージの受信に使用する LAN アダプタのホスト名	1~255 文字のホスト名
			multicastttl	マルチキャストメッセージ送信時のマルチキャストパケットの生存期間 (TTL)	符号なし整数 ((0~255))
論理端末アプリケーション定義	eemcplefunc	-u	ownlsg	入力セグメント判定 UOC のエントリポインタ名	1~31 文字の識別子
			ownlmc	入力メッセージ編集 UOC のエントリポインタ名	1~31 文字の識別子
			ownlra	出力メッセージ編集 UOC のエントリポインタ名	1~31 文字の識別子
			owntrn	トランザクションレベル名判定 UOC のエントリポインタ名	1~31 文字の識別子
			rltrn	MCP 後処理トランザクション起動サービス名	1~31 文字の識別子
			rltrnlevel	MCP 後処理トランザクションのトランザクションレベル名	1~31 文字の識別子
			ownlcn	コネクション確立 UOC のエントリポインタ名	1~31 文字の識別子

定義名	コマンド	オプション・オペランド		定義内容	指定値((値範囲))《省略時解釈値》
論理端末アプリケーション定義	eemcplefunc	-u	ownojed	OJ 編集 UOC エントリ ポインタ名	1~31 文字の識別子
		-e	errtrn1	ERRTRN1 のエントリ ポインタ名	1~31 文字の識別子
			errtrn2	ERRTRN2 のエントリ ポインタ名	1~31 文字の識別子
			errtrn3	ERRTRN3 のエントリ ポインタ名	1~31 文字の識別子
			errtrnr	ERRTRNR のエントリ ポインタ名	1~31 文字の識別子
			errtrns	ERRTRNS のエントリ ポインタ名	1~31 文字の識別子
スレッドグループ定義	eemcpthdgrp	—	—	スレッドグループ定義	—
UDP 定義の終了	eemcpudpend	—	—	UDP 定義の終了	—
アプリケーション定義	eemcpfunc	-u	ownlsg	入力セグメント判定 UOC のエントリポインタ名	1~31 文字の識別子
			ownlmc	入力メッセージ編集 UOC のエントリポインタ名	1~31 文字の識別子
			ownlra	出力メッセージ編集 UOC のエントリポインタ名	1~31 文字の識別子
			rltrn	MCP 後処理トランザクションを起動するサービス名	1~31 文字の識別子
			rltrnlevel	MCP 後処理トランザクションのトランザクションレベル名	1~31 文字の識別子
			ownlcn	コネクション確立 UOC のエントリポインタ名	1~31 文字の識別子
		-e	errtrns	メッセージ送信時に通信障害が発生した場合、送信障害トランザクションを起動するかどうか	《yes》 no
		-c	uoceexc	入力メッセージ編集 UOC および出力メッ	yes 《no》

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》
アプリケーション定義	eemcpfunc	-c	uocexc	セージ編集 UOC を端末単位にシリアル化するかどうか	yes 《no》
		-q	-	端末キュー機能を使用して、論理端末単位のイベントをシリアルに処理するかどうか	《serial》 parallel
		-m	msglen	送受信メッセージの最大サイズを拡張するかどうか	《normal》 extend
			recvpermanence	受信メッセージを永続化するかどうか	yes 《no》
			reconnectsend	メッセージ受信時のコネクション以外の場合に、応答メッセージを送信するかどうか	yes no fsp_use=N の場合 : 《yes》 fsp_use=Y の場合 : 《no》
		-o	outputmessage	OJ の取得先 ID	符号なし整数 ((0~255)) 《0》
トレース定義	eemcptrc		outputfunction	OJ の機能拡張レベル	16 進数 《00000000》 00000001
		-f	fileno	MCP トレースのファイル数	符号なし整数 ((3~100)) 《3》
			filesz	MCP トレースの最大ファイルサイズ	符号なし整数 (単位 : キロバイト) ((400~4000000)) 《40000》
		-b	bufferno	MCP トレースファイルの出力用バッファ数	符号なし整数 ((3~200)) 《3》
			buffersz	MCP トレースファイルの出力用最大バッファサイズ	符号なし整数 (単位 : キロバイト) ((41~20000)) 《200》
		-x	entrysize	スレッドごとの MCP トレースのバッファサイズ	符号なし整数 (単位 : キロバイト) ((40~10000)) 《80》

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》
トレース定義	eemcptrc	-s	sendresult	送信結果トレースを取得するかどうか	yes 《no》
		-m	filter	MCP トレース情報の取得量の変更	16 進数 《00000000》 00000010
			msgsize	MCP トレース情報の取得最大サイズ	符号なし整数 (単位: バイト) ((0~32000)) 《128》

(凡例)

– : 該当しません。

注※

CL 構成では、IP アドレスまたはホスト名と、ポート番号を実行系と待機系で同じ値にする必要があります。そのため、IP アドレスまたはホスト名には、必ずエイリアス IP アドレスを指定してください。

なお、IP アドレスの代わりにホスト名を指定する場合は、ホスト名とエイリアス IP アドレスの対応づけを/etc/hosts に指定するなどの対応が必要です。エイリアス IP アドレスの指定方法については、マニュアル「高信頼化システム監視機能 HA モニタ Linux(R)編」(OS が Linux の場合)、または「高信頼化システム監視機能 HA モニタ AIX(R)編」(OS が AIX の場合)を参照してください。

4.3 MCP 構成定義の詳細

ここでは、MCP 構成定義の詳細について説明します。

eemcpbuf (バッファ定義)

機能

メッセージ編集用バッファに関する環境を定義します。

コマンドの形式

```
[eemcpbuf [-e " [emsgbuf=編集用バッファサイズ] "]]
```

オプションの説明

●-e

(オペランド)

emsgbuf=編集用バッファサイズ

～〈符号なし整数〉((0～2147483647))《0》(単位:バイト)

入力メッセージ編集 UOC および出力メッセージ編集 UOC で使用するメッセージ編集用バッファサイズを指定します。

入力メッセージ編集 UOC または出力メッセージ編集 UOC で使用するメッセージ編集用バッファサイズのうち、どちらか大きい方を指定してください。

eemcptcpstart (TCP/IP 定義の開始)

機能

TCP/IP プロトコルを使用した通信に関する定義の開始を示します。

コマンドの形式

```
[eemcptcpstart [-o " [con_addr=dynamic | static] "]]
```

オプションの説明

●-o

(オペランド)

con_addr=dynamic | static

～《static》

クライアント型コネクション (MCP 構成定義 (eemcpcn -y) の mode オペランドで client を指定) の場合、相手システムを動的に指定しコネクションを確立するかどうかを指定します。

dynamic

相手システムを動的に指定しコネクションを確立します。

クライアント型コネクションの eemcpcn 定義の-o オプションを省略可能です。ただし-o オプションを省略した eemcpcn 定義のコネクション ID に対し、eemcpactcn コマンド、ee_mcp_actcn (COBOL 言語は CBLEEMCP('ACTCN')) API を実行するとエラーになります。

static

相手システムを静的に指定しコネクションを確立 (定義で指定した相手システムに対しコネクションを確立) します。クライアント型コネクションの eemcpcn 定義の-o オプションを省略できません。

注意事項

この定義は、一度だけ定義してください。なお、定義の記述順序については、「4.1.2(3) 記述順序」を参照してください。

eemcpcn (コネクション定義 (TCP/IP))

機能

TCP/IP プロトコルを使用して通信する場合のコネクションに関する環境を定義します。

コマンドの形式

```
{ { [eemcpcn -c コネクションID
      [-i auto | manual]
      [-b " [bretry= yes | no]
            [bretrycnt=コネクション確立障害時の確立再試行回数]
            [bretryint=コネクション確立障害時の確立再試行間隔]
            [concmptim=コネクション確立時の監視時間]
            [conmsg= yes | no]
            [sndsynctim=同期一方送信時のメッセージ送信完了待ち時間]
            [sndtim=非同期一方送信時のメッセージ送信完了待ち時間]
            [sndrcvtim=同期型送受信時のメッセージ送受信完了待ち時間]
            [sndcmptim=同期型送受信時のメッセージ送信完了監視時間]"]
      [-w " [srtimout=yes | no] "]
      -y "mode=client | server"
      [-r " [ipaddr=自システムのホストのIPアドレス]
            [hostname=自システムのホスト名]
            [portno=自システムのポート番号] "]
      [-o " [oipaddr=相手システムのホストのIPアドレス]
            [ohostname=相手システムのホスト名]
            [oportno=相手システムのホストのポート番号 | free] "]
      [-k " [nodeelay= yes | no] "]
    ] }
```

```

[keepalive= yes|no]
[notrftime=無通信監視時間]"]
[-f " [cnorelease= fin|rst]
[endrelease= fin|rst]"]
[-h " [addrchk= yes|no]
[chgconn= replace|keep]"] } }

```

オプションの説明

●-c コネクション ID

～〈1～8 文字の識別子〉

OpenTP1 システム内で、一意となるコネクション ID を指定します。

●-i auto | manual

～《manual》

オンライン開始時および再開始時のコネクションの確立方法を指定します。

auto

オンライン開始前にコネクションを自動確立します。

manual

eemcpactcn コマンドの実行、または ee_mcp_actcn 関数の発行によってコネクションを手動確立します

●-b

(オペランド)

bretry=yes | no

～《yes》

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

yes

コネクションの確立再試行をします。

no

コネクションの確立再試行をしません。

bretrycnt=コネクション確立障害時の確立再試行回数

～〈符号なし整数〉 ((0～65535)) 〈0〉 (単位：回)

コネクションの確立再試行をする場合の確立再試行回数を指定します。bretry オペランドで no を指定した場合、このオペランドの指定値は無視されます。

bretry オペランドで yes を指定して、このオペランドに 0 を指定した場合は、無限にコネクションの確立再試行をします。

bretryint=コネクション確立障害時の確立再試行間隔

～〈符号なし整数〉((0～2550))《60》(単位:秒)

コネクションの確立再試行をする場合の確立再試行間隔を指定します。bretry オペランドで no を指定した場合、このオペランドの指定値は無視されます。確立再試行間隔は、最大で 1 秒早く施行される場合があります。

concmptim=コネクション確立時の監視時間

～〈符号なし整数〉((0～65535))《0》(単位:秒)

コネクション確立時の監視時間を指定します。0 を指定した場合は、時間監視をしません。コネクション確立時の監視時間がタイムアウトした場合、コネクション確立を再試行できる場合 (bretry オペランドに yes を指定)、コネクション確立障害時の確立再試行間隔のあとに、コネクション確立を再試行します。コネクション確立を再試行できない場合 (bretry オペランドに no を指定)、MCP 後処理トランザクションを起動します。監視時間は最大で 1 秒早くタイムアウトする場合があります。

conmsg=yes | no

～《yes》

コネクション確立時または解放時に、メッセージログにコネクションの確立または解放のメッセージを出力するかどうかを指定します。

yes

コネクションの確立時または解放時にメッセージを出力します。

no

コネクションの確立時または解放時にメッセージを出力しません。

このオペランドに no を指定した場合でも、リソース不足などで MCP 後処理トランザクションの起動に失敗したときはメッセージを出力します。

sndsynctim=同期一方送信時のメッセージ送信完了待ち時間

～〈符号なし整数〉((0～65535))《0》(単位:秒)

ee_mcp_sendsync 関数を呼び出してから終了するまでの最大時間を指定します (ソケット空き待ち時間、および送信開始から自システムの TCP/IP の送信バッファに書き込みが完了するまでの時間の合計の最大時間を指定します)。

0 を指定した場合は、無限に送信完了を待ちます。

sndtim=非同期一方送信時のメッセージ送信完了待ち時間

～〈符号なし整数〉((0～65535))《0》(単位:秒)

ee_mcp_send 関数を呼び出したトランザクション完了後の送信開始から終了するまでの最大時間を指定します (自システムの TCP/IP の送信バッファに書き込みが完了するまでの最大時間を指定します)。

0 を指定した場合は、無限に送信完了待ちをします。

sndrcvttim=同期型送受信時のメッセージ送受信完了待ち時間

～〈符号なし整数〉((0～65535))《0》(単位:秒)

ee_mcp_sendrecv 関数を呼び出してから終了するまでの最大時間を設定します。0 を指定した場合、無限に送信完了待ちをします。

sndcmptim=同期型送受信時のメッセージ送信完了監視時間

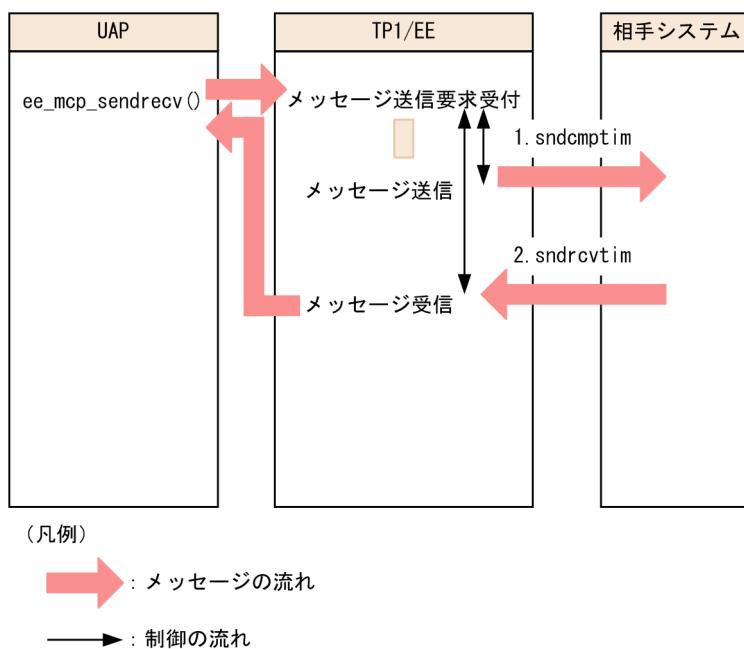
～〈符号なし整数〉((0～65535))〈0〉(単位:秒)

ee_mcp_sendrecv 関数を呼び出してからメッセージ送信完了(ソケット空き待ち時間と、送信開始から自システムの TCP/IP の送信バッファへの書き込み完了までの時間の合計)までの監視時間を指定します。0 を指定した場合、無限に送信完了待ちをします。

このオペランドには、次の値より小さな値を指定してください。次の値より大きい値を指定した場合、値の小さい方が有効になります。

- eemcpn コマンドの-b オプションの sndrcvtime オペランドで指定する同期送受信時のメッセージ送信完了待ち時間
- 同期型メッセージの送受信関数の発行時 (ee_mcp_sendrecv 関数または CBLEEMCP('SENDRECV')) に指定する最大時間 (ee_mcp_sendrecv 関数の場合、watchtime に指定する最大時間。CBLEEMCP('SENDRECV') の場合、データ名 M5 に指定する最大時間)

図 4-3 sndrcvtime と sndcmptim による対象監視時間



●-W

(オペランド)

srtimout=yes | no

～《no》

同期型メッセージの送受信関数で次の監視時間がタイムアウトした場合に、コネクションの解放を抑止するかどうかを指定します。

- eemcpn コマンドの-b オプションの sndrcvtime オペランドで指定する同期送受信時のメッセージ送信完了待ち時間
- 同期型メッセージの送受信関数の発行時 (ee_mcp_sendrecv 関数または CBLEEMCP('SENDRECV')) に指定する最大時間

yes

同期型メッセージの送受信関数がタイムアウトした場合、コネクションを解放しません。

no

同期型メッセージの送受信関数がタイムアウトした場合、コネクションを解放します。

●-y

(オペランド)

mode=client | server

自システムのコネクション確立モードを指定します。

client

クライアントとして、コネクションの確立要求をします。

server

サーバとして、コネクションの確立要求を受けます。

●-r

自システムのアドレス情報を設定します。

クライアント型コネクション (MCP 構成定義 (eemcpcn -y) の mode オペランドで client を指定) の定義間では、-r オプションに同一の IP アドレスまたはホスト名/ポート番号の組み合わせを指定できます。ただし、クライアント型コネクションとサーバ型コネクション (MCP 構成定義 (eemcpcn -y) の mode オペランドで server を指定) の定義間では、-r オプションに同一の IP アドレスまたはホスト名/ポート番号の組み合わせを指定できません。

また、クライアント型コネクションの定義間では、次の条件を満たす場合も、-r オプションに同一の IP アドレスまたはホスト名と、ポート番号の組み合わせを指定できません。

- -o オプションに同一の IP アドレスまたはホスト名と、ポート番号の組み合わせを指定する場合

CL 構成では、IP アドレスまたはホスト名と、ポート番号を実行系と待機系で同じ値にする必要があります。そのため、IP アドレスまたはホスト名には、必ずエイリアス IP アドレスを指定してください。

なお、IP アドレスの代わりにホスト名を指定する場合は、ホスト名とエイリアス IP アドレスの対応づけを/etc/hosts に指定するなどの対応が必要です。エイリアス IP アドレスの指定方法については、マニュアル「高信頼化システム監視機能 HA モニタ Linux(R)編」(OS が Linux の場合)、または「高信頼化システム監視機能 HA モニタ AIX(R)編」(OS が AIX の場合) を参照してください。

(オペランド)

ipaddr=自システムのホストの IP アドレス

~ (nnn.nnn.nnn.nnn) 〈符号なし整数〉 ((0~255))

自システムのホストの IP アドレスを指定します。

指定形式は nnn.nnn.nnn.nnn です。nnn には、0～255 の値を指定できます。ただし、0.0.0.0 および 255.255.255.255 は指定できません。

このオペランドを指定した場合、指定したアドレスをバインドします。

ipaddr オペランドまたは hostname オペランドのどちらか一方を指定してください。どちらも省略した場合は、OS が使用する LAN アダプタが仮定されます。

hostname=自システムのホスト名

～〈1～255 文字のホスト名〉

自システムのホスト名を指定します。

自システムがサーバで、TP1/EE 開始時に IP アドレスを取得できなかった場合、開始処理を中断します。

ipaddr オペランドを指定した場合、このオペランドの指定値は無視されます。ipaddr オペランドまたは hostname オペランドのどちらか一方を指定してください。

このオペランドを指定していて、かつ ipaddr オペランドの指定を省略した場合、このオペランドで指定したホスト名のアドレスをバインドします。どちらも省略した場合は、OS が使用する LAN アダプタが仮定されます。

portno=自システムのポート番号

～〈符号なし整数〉((1024～65535))

メッセージ送受信に使用する、自システムのポート番号を指定します。

サーバ型コネクション (-y オプションの mode オペランドに server を指定) の場合は必ず指定してください。

クライアント型コネクション (-y オプションの mode オペランドに client を指定) の場合は省略できます。省略した場合、自システムのポート番号はコネクション確立時に OS によって割り振られます。ポート番号を指定する場合は、ほかのアプリケーションとの重複を避けるため、OS が任意に割り当てるポート番号（動的ポートまたは短命ポートと呼ばれるポート番号）は使用しないでください。OS が任意に割り当てるポート番号は、OS の種別やバージョンによって異なります。詳細については、ご使用の OS のマニュアルを参照してください。

また、-r オプションの portno オペランドで同じサーバ型コネクションを複数指定する場合、次に示すコネクションを混在させないでください。

- -r オプションの ipaddr オペランドおよび hostname オペランドの指定を省略したコネクション
- -r オプションの ipaddr オペランドまたは hostname オペランドを指定したコネクション

●-o

相手システムのアドレス情報を設定します。

-o オプションを省略または指定した場合の eemcpogen 結果、コネクション確立方法ごとの確立可否を次の表に示します。

表 4-2 クライアント型コネクションでの eemcpcn -o オプション指定時/省略時

項目番号	指定条件 (定義)				eemcp gen 結果	コネクション確立方法	
	eemcpcpstart -o con_addr		eemcp cn -o	eemcpcn -i		eemcpactcpto コマンドまたは ee_mcp_actcn_to (COBOL 言語は BLEEMCP('ACTCNT O')) API	eemcpactcn コマンドまたは ee_mcp_actcn (COBOL 言語は CBLEEMCP('ACTCN ')) API
	dynamic	static		manual	auto		
1	○	—	省略	○	—	OK	使用可
2	○	—	省略	—	○	NG	—
3	○	—	指定	○	—	OK	使用可
4	○	—	指定	—	○	OK	使用可
5	—	○	省略	—	—	NG	—
6	—	○	指定	○	—	OK	使用可
7	—	○	指定	—	○	OK	使用可

(凡例)

○ : 指定します (省略値が同値の場合も含みます)。

— : 該当しません。

OK : eemcpgen が正常終了します。

NG : eemcpgen に失敗します (不正な定義の指定方法です)。

使用可 : コマンド・API を使用可能です。

使用不可 : コマンド・API を実行するとエラーになります。

表 4-3 サーバ型コネクションでの eemcpcn -o オプション指定時/省略時

項目番号	指定条件 (定義)			eemcp gen 結果	
	eemcpcn -h addrchk		eemcpcn -o		
	yes	no			
1	○	—	—	省略	NG
2	○	—	—	指定	OK
3	—	○	—	省略	OK
4	—	○	—	指定	OK

(凡例)

○ : 指定します (省略値が同値の場合も含みます)。

— : 該当しません。

OK : eemcpgen が正常終了します。

NG : eemcpgen に失敗します (不正な定義の指定方法です)。

(オペランド)

oipaddr=相手システムのホストの IP アドレス

～ (nnn.nnn.nnn.nnn) 〈符号なし整数〉 ((0～255))

相手システムのホストの IP アドレスを指定します。

指定形式は nnn.nnn.nnn.nnn です。nnn には、0～255 の値を指定できます。ただし、0.0.0.0 と 255.255.255.255 は指定できません。

ohostname=相手システムのホスト名

～ 〈1～255 文字のホスト名〉

相手システムのホスト名を指定します。

-o オプションを指定する場合は、oipaddr オペランドか ohostname オペランドのどちらか一方を指定してください。oipaddr オペランドを指定した場合は、このオペランドの指定値は無視されます。

oportno=相手システムのホストのポート番号 | free

～ 〈符号なし整数〉 ((1～65535)) 《free》

相手システムのホストのポート番号を指定します。free を指定できるのは、サーバ型コネクション (-y オプションの mode オペランドで server を指定) の場合だけです。

free を指定した場合、oipaddr オペランドまたは ohostname オペランドで指定した相手システムと一致する、任意のポート番号からのコネクション確立要求を受け入れます。このとき、一つの相手システムから複数のコネクションを受け入れるためには、受け入れるコネクション数の分だけコネクション定義を指定する必要があります。それぞれのコネクション定義には、同一の oipaddr オペランドまたは ohostname オペランドを指定してください。指定したコネクション定義数以上のコネクション確立要求を受けた場合、要求を拒否します。ただし、free を指定し、-r オプションの ipaddr オペランドおよび portno オペランドの指定値が同じコネクションに対しては、コネクションリプレース機能を使用できます。コネクションリプレース機能については、「[2.1.4 コネクションリプレース機能](#)」を参照してください。

●-k

(オペランド)

nodelay=yes | no

～ 《no》

TCP/IP が提供するソケットオプション「TCP_NODELAY」を使用するかどうかを指定します。

TCP_NODELAY を使用することで、送信済みデータの応答待ちの状態でも、遅延させることなくデータ送信ができるようになりますが、ネットワークの負荷は高くなります。そのため、この機能の必要性を十分に検討した上で使用してください。

yes

TCP_NODELAY を使用します。

no

TCP_NODELAY を使用しません。

keepalive=yes | no

～《no》

TCP/IP が提供するソケットオプション「SO_KEEPALIVE」を使用するかどうかを指定します。

SO_KEEPALIVE を使用することで、コネクションがメッセージ送受信中でない場合に、相手システムに対して一定間隔でパケットを送信し、その間に応答がなければ障害と見なしてコネクションを切断することができます。これをキープアライブといいます。

ただし、キープアライブを行うとネットワークの負荷を高める場合があります。そのため、この機能の必要性を十分に検討した上で使用してください。

yes

キープアライブを使用します。

no

キープアライブを使用しません。

notrftime=無通信監視時間

～《符号なし整数》 ((0～65535)) 《0》 (単位：秒)

コネクションの無通信状態を監視するタイマ値を設定します。0 を指定した場合は、時間監視をしません。監視時間は最大で 1 秒早くタイムアウトする場合があります。

●-f

(オペランド)

cnrelease=fin | rst

～《fin》

コネクションの解放形態を指定します。

fin

FIN パケットを送信してコネクションを解放します。

rst

RST パケットを送信してコネクションを強制的に解放します。

-f オプションを指定していない場合に、eemcpdctcn コマンドを実行するときは、FIN パケットを送信してコネクションを解放します。TP1/EE の正常終了、計画停止 A、または計画停止 B の場合のコネクション解放形態は、endrelease オペランドで指定してください。

MCP は OS が提供する TCP/IP のソケットインターフェース (SO_LINGER) を利用しています。そのため、このオペランドに fin を指定しても、ネットワーク上でのそれ違いなどによって、このオペランドの指定値とは関係なく RST パケットを送信する場合があります。

endrelease=fin | rst

～《fin》

TP1/EE の正常終了、計画停止 A、または計画停止 B の場合のコネクションの解放形態を指定します。ただし、CL 構成の場合は cnrelease オペランドで指定したコネクション解放形態でコネクションを解放します。

fin

FIN パケットを送信してコネクションを解放します。

rst

RST パケットを送信してコネクションを強制的に解放します。

MCP は OS が提供する TCP/IP のソケットインターフェース (SO_LINGER) を利用しています。そのため、このオペランドに fin を指定しても、ネットワーク上でのすれ違いなどによって、このオペランドの指定値とは関係なく RST パケットを送信する場合があります。

●-h

(オペランド)

addrchk=yes | no

～《yes》

コネクション確立要求の受信時に相手アドレス情報をチェックするかどうかを指定します。

一つの MCP 構成定義内では、すべてのサーバ型コネクションで、このオペランドに同じ値を指定する必要があります。

クライアント型コネクションの場合、このオペランドを指定する必要はありません。指定しても無視されます。

yes

コネクション確立要求受信時に相手のアドレス情報をチェックします。コネクション定義で指定された相手のアドレスと確立要求発行元のアドレスが一致する場合は、確立要求を受け入れます。一致しない場合は、確立要求を拒否します。ただし、相手のアドレス情報をチェックするためには、-o オプションの oipaddr オペランド、ohostname オペランドおよび oportno オペランドで相手システムのアドレス情報を定義しておく必要があります。

no

コネクション確立要求受信時に相手のアドレス情報をチェックしません。未確立コネクションがない場合は、確立要求を拒否します。したがって、相手システムのアドレス情報を定義する必要はありません。定義した場合は、無視されます。ただし、no を指定した場合でも、-r オプションの ipaddr オペランドおよび portno オペランドの指定値が同じコネクションに対しては、コネクションリプレース機能を使用できます。コネクションリプレース機能については、「[2.1.4 コネクションリプレース機能](#)」を参照してください。

chgconn=replace | keep

～《keep》

相手システムからコネクションの確立要求を受け付けた場合に、割り当てできる未確立コネクションがないときの動作を指定します。

replace

コネクションリプレース機能を使用します。相手システムと確立状態にある最も古いコネクションを切断して、新たなコネクション確立要求を受け付けます。

keep

コネクションリプレース機能を使用しません。相手システムからの確立要求を拒否します。

一つの MCP 構成定義内で、同じ自システムのポート番号 (-r オプションの portno オペランドで指定) を持つすべてのサーバ型コネクション (-y オプションの mode オペランドで server を指定) で、このオペランドに同じ値を指定する必要があります。

クライアント型コネクション (-y オプションの mode オペランドで client を指定) および、相手ポート番号が固定 (-o オプションの oportno オペランドに 1~65535 を指定) のサーバ型コネクションにこのオペランドを指定した場合、このオペランドの指定値は無視されます。なお、コネクションリプレース機能については、「[2.1.4 コネクションリプレース機能](#)」を参照してください。

eemcple (論理端末定義 (TCP/IP))

機能

TCP/IP プロトコルを使用して通信する場合の論理端末に関する環境を定義します。

コマンドの形式

```
{ { [eemcple -l 論理端末名  
      [-i auto | manual]  
      [-v 入力メッセージを受信した場合に起動する  
          エントリポイント名に対応したサービス名] ] } }
```

オプションの説明

●-l 論理端末名

～ 〈1~8 文字の識別子〉

OpenTP1 システム内で、一意となる論理端末名を指定します。

●-i auto | manual

～ 《auto》

論理端末の閉塞解除方法を指定します。

auto

コネクション確立時に自動的に論理端末が閉塞解除されます。

manual

eemcpactle コマンドの実行、または ee_mcp_actle 関数の発行によって、手動で論理端末を閉塞解除します。

●-v 入力メッセージを受信した場合に起動するエントリポインタ名に対応したサービス名

～ <1～31 文字の識別子>

入力メッセージを受信した場合に起動するエントリポインタ名に対応したサービス名を指定します。TP1/EE サービス定義で定義したサービス名を指定してください。このオプションを省略した場合は、入力メッセージ編集 UOC で指定してください。

このオプションを省略した論理端末定義が一つでもある場合、入力メッセージ編集 UOC (eemcpfunc -u ownlmc) を指定しないと eemcpgen コマンドが失敗します。

注意事項

この定義は、必ず eemcpcn 定義コマンドの次に定義してください。なお、定義の記述順序については、[\[4.1.2\(3\) 記述順序\]](#) を参照してください。

eemcptcpPEND (TCP/IP 定義の終了)

機能

TCP/IP プロトコルを使用した通信に関する定義の終了を示します。

コマンドの形式

```
[eemcptcpPEND]
```

注意事項

この定義は eemcptcpstart 定義コマンドと対で定義してください。なお、定義の記述順序については、[\[4.1.2\(3\) 記述順序\]](#) を参照してください。

eemcpudpstart (UDP 定義の開始)

機能

UDP プロトコルを使用した通信に関する定義の開始を示します。

コマンドの形式

```
[eemcpudpstart]
```

注意事項

この定義は一度だけ定義してください。なお、定義の記述順序については、[\[4.1.2\(3\) 記述順序\]](#) を参照してください。

eemcple (論理端末定義 (UDP))

機能

UDP プロトコルを使用して通信する場合の論理端末に関する環境を定義します。

コマンドの形式

```
{ { [eemcple -l 論理端末名
      [-i auto | manual]
      [-v 入力メッセージを受信した場合に起動する
          エントリポイント名に対応したサービス名]
      [-b " [sndtim=メッセージ送信完了待ち時間] "]
      [-s " [syssndsize=システムのメッセージ送信バッファ長]
          [sysrcvsize=システムのメッセージ受信バッファ長] "]
      [-r " [ipaddr=メッセージの送受信に使用するLANアダプタの
          IPアドレス]
          [hostname=メッセージの送受信に使用するLANアダプタの
          ホスト名]
          [portno=自システムのポート番号]
      [-o " [oipaddr=相手システムのホストのIPアドレス]
          [ohostname=相手システムのホスト名]
          [oportno=相手システムのホストのポート番号] "]
      [-k " [reuseaddr= yes | no] "]
      [-m " [multicast= yes | no]
          [hostgroupaddr=マルチキャストメッセージの受信に使用する
          ホストグループのIPアドレス]
          [hostgroupname=マルチキャストメッセージの受信に使用する
          ホストグループのホスト名]
          [ripaddr=マルチキャストメッセージの受信に使用する
          LANアダプタのIPアドレス]
          [rhostname=マルチキャストメッセージの受信に使用する
          LANアダプタのホスト名]
          [multicastttl=送信するマルチキャストパケットの
          生存期間 (TTL) ] "]] } }
```

オプションの説明

●-l 論理端末名

～ 〈1～8 文字の識別子〉

OpenTP1 システム内で、一意となる論理端末名を指定します。

●-i auto | manual

～ 《manual》

オンライン開始時、および再開始時の論理端末の閉塞解除方法を指定します。

auto

オンライン開始時および再開始時に自動的に論理端末の閉塞が解除されます。

manual

eemcpactle コマンドの実行、または ee_mcp_actle 関数の発行によって、手動で論理端末の閉塞を解除します。

●-v 入力メッセージを受信した場合に起動するエントリポインタ名に対応したサービス名

～ 〈1～31 文字の識別子〉

入力メッセージを受信した場合に起動するエントリポインタ名に対応したサービス名を指定します。TP1/EE サービス定義で定義したサービス名を指定してください。このオプションを省略した場合は、入力メッセージ編集 UOC で指定してください。

●-b

(オペランド)

sndtim=メッセージ送信完了待ち時間

～ 〈符号なし整数〉 ((0～65535)) 〈0〉 (単位：秒)

同一の論理端末に複数の UAP (スレッド) から同時に送信要求をする場合の送信完了待ち時間の最大時間を指定します。0 を指定した場合は、無限に送信完了を待ちます。

●-s

(オペランド)

syssndsize=システムのメッセージ送信バッファ長

～ 〈符号なし整数〉 ((1～2147483647)) (単位：バイト)

システムのメッセージ送信バッファ長を指定します。

送信メッセージ長が、システムのメッセージ送信バッファ長より大きい場合、送信が失敗する場合があります。送信に失敗した場合は、論理端末を閉塞し、エラーリターンします。そのため、このオペランドの指定値は、次の条件式を満たすことをお勧めします。

システムのメッセージ送信バッファ長 > 送信メッセージ長

sysrcvsize=システムのメッセージ受信バッファ長

～ 〈符号なし整数〉 ((1～2147483647)) (単位：バイト)

システムのメッセージ受信バッファ長を指定します。

受信できるメッセージ長は、受信バッファ (IBF) サイズまでです。受信バッファサイズを超えるメッセージの場合は、受信バッファサイズに入りきらない部分が切り捨てられ、受信バッファサイズのメッセージとして扱います。また、受信メッセージ長が、システムのメッセージ受信バッファ長より大きい場合、メッセージは UDP プロトコルによって破棄される場合があります。したがって、このオペランドの指定値は、次の条件式を満たすことをお勧めします。

システムのメッセージ受信バッファサイズ > 受信バッファサイズ > 受信メッセージ長

syssndsize オペランドおよび sysrcvsize オペランドで指定できるバッファ長の範囲は OS に依存します。指定を省略した場合は、OS のデフォルト値が仮定されます。詳しくは、ご使用の OS のマニュアルで、UDP プロトコルについて説明している個所を参照してください。

●-r

CL 構成では、IP アドレスまたはホスト名と、ポート番号を実行系と待機系で同じ値にする必要があります。そのため、IP アドレスまたはホスト名には、必ずエイリアス IP アドレスを指定してください。

なお、IP アドレスの代わりにホスト名を指定する場合は、ホスト名とエイリアス IP アドレスの対応づけを/etc/hosts に指定するなどの対応が必要です。エイリアス IP アドレスの指定方法については、マニュアル「高信頼化システム監視機能 HA モニタ Linux(R)編」(OS が Linux の場合)、または「高信頼化システム監視機能 HA モニタ AIX(R)編」(OS が AIX の場合) を参照してください。

(オペランド)

ipaddr=メッセージの送受信に使用する LAN アダプタの IP アドレス

～ (nnn.nnn.nnn.nnn) 〈符号なし整数〉 ((0～255))

自システムがホットスタンバイ構成である場合、または複数の LAN アダプタを持つ場合に、メッセージの送受信に使用する LAN アダプタの IP アドレスを指定します。

指定形式は nnn.nnn.nnn.nnn です。nnn には、0～255 の値を指定できます。ただし、0.0.0.0 と 255.255.255.255 は指定できません。

ipaddr オペランドまたは hostname オペランドのどちらか一方を指定してください。どちらも省略した場合は、OS が使用する LAN アダプタが仮定されます。

マルチキャストおよびブロードキャストのメッセージ受信に使用する論理端末の場合、ipaddr オペランドを指定してはいけません。

マルチキャストのメッセージ受信時は、-m オプションの hostgroupname オペランドまたは hostgroupaddr オペランドのどちらか一方を指定してください。

hostname=メッセージの送受信に使用する LAN アダプタのホスト名

～ 〈1～255 文字のホスト名〉

自システムがホットスタンバイ構成である場合、または複数の LAN アダプタを持つ場合に、メッセージの送受信に使用する LAN アダプタのホスト名を指定します。

hostname オペランドまたは ipaddr オペランドのどちらか一方を指定してください。どちらも省略した場合は、OS が使用する LAN アダプタが仮定されます。ipaddr オペランドを指定した場合は、このオペランドの指定値は無視されます。

マルチキャストおよびブロードキャストのメッセージ受信に使用する論理端末の場合、hostname オペランドを指定してはいけません。

マルチキャストのメッセージ受信時は、-m オプションの hostgroupname オペランドまたは hostgroupaddr オペランドのどちらか一方を指定してください。

portno=自システムのポート番号

～ 〈符号なし整数〉 ((1024～65535))

メッセージ送受信に使用する、自システムのポート番号を指定します。

portno オペランドで指定するポート番号は、重複を避けるため次のポート番号とは異なる値を指定してください。

- ほかの eemcple (論理端末定義 (UDP)) 定義コマンドの-r オプションで指定するポート番号
- OS が任意に割り当てるポート番号 (動的ポートまたは短命ポートと呼ばれるポート番号)
OS が任意に割り当てる番号は、OS の種別やバージョンによって異なります。詳細については、ご使用の OS のマニュアルを参照してください。

●-o

(オペランド)

oipaddr=相手システムのホストの IP アドレス

~ (nnn.nnn.nnn.nnn) 〈符号なし整数〉 ((0~255))

ee_mcp_send 関数 (CBLEEMCP('SEND')), および ee_mcp_sendsync 関数 (CBLEEMCP('SENDSYNC')) の相手システムのホストの IP アドレスを指定します。

指定形式は nnn.nnn.nnn.nnn です。nnn には、0~255 の値を指定できます。ただし、0.0.0.0 および 255.255.255.255 は指定できません。

-o オプションを指定する場合は、oipaddr オペランドか ohostname オペランドのどちらか一方を指定してください。

ohostname=相手システムのホスト名

~ 〈1~255 文字のホスト名〉

ee_mcp_send 関数 (CBLEEMCP('SEND')), および ee_mcp_sendsync 関数 (CBLEEMCP('SENDSYNC')) の相手システムのホスト名を指定します。

-o オプションを指定する場合は、oipaddr オペランドか ohostname オペランドのどちらか一方を指定してください。oipaddr オペランドを指定した場合は、このオペランドの指定値は無視されます。

oportno=相手システムのホストのポート番号

~ 〈符号なし整数〉 ((1~65535))

ee_mcp_send 関数 (CBLEEMCP('SEND')), および ee_mcp_sendsync 関数 (CBLEEMCP('SENDSYNC')) の相手システムのホストのポート番号を指定します。

●-k

(オペランド)

reuseaddr=yes | no

~ 〈no〉

ソケットオプション「SO_REUSEADDR」を使用するかどうかを指定します。

SO_REUSEADDR を使用した場合は、複数のプロセスで同一のポート番号を共用できます。

yes

SO_REUSEADDR を使用します。

no

SO_REUSEADDR を使用しません。

●-m

CL 構成では、IP アドレスまたはホスト名と、ポート番号を実行系と待機系で同じ値にする必要があります。そのため、IP アドレスまたはホスト名には、必ずエイリアス IP アドレスを指定してください。

なお、IP アドレスの代わりにホスト名を指定する場合は、ホスト名とエイリアス IP アドレスの対応づけを/etc/hosts に指定するなどの対応が必要です。エイリアス IP アドレスの指定方法については、マニュアル「高信頼化システム監視機能 HA モニタ Linux(R)編」(OS が Linux の場合)、または「高信頼化システム監視機能 HA モニタ AIX(R)編」(OS が AIX の場合) を参照してください。

(オペランド)

multicast=yes | no

～ 《no》

マルチキャストを使用してメッセージを送受信するかどうかを指定します。

yes

マルチキャストを使用してメッセージを送受信します。

no

マルチキャストを使用してメッセージを送受信しません。

hostgroupaddr=マルチキャストメッセージの受信に使用するホストグループの IP アドレス

～ (nnn.nnn.nnn.nnn) 〈符号なし整数〉 ((0～255))

マルチキャストメッセージを受信する場合、参加するホストグループの IP アドレスを指定します。

指定形式は nnn.nnn.nnn.nnn です。nnn には、0～255 の値を指定できます。ただし、0.0.0.0 と 255.255.255.255 は指定できません。

multicast オペランドに yes を指定した場合、hostgroupaddr オペランドまたは hostgroupname オペランドのどちらか一方を必ず指定してください。

hostgroupname=マルチキャストメッセージの受信に使用するホストグループのホスト名

～ 〈1～255 文字のホスト名〉

マルチキャストメッセージを受信する場合、参加するホストグループのホスト名を指定します。

multicast オペランドに yes を指定した場合、hostgroupname オペランドまたは hostgroupaddr オペランドのどちらか一方を必ず指定してください。hostgroupaddr オペランドを指定した場合は、このオペランドの指定値は無視されます。

ripaddr=マルチキャストメッセージの受信に使用する LAN アダプタの IP アドレス

～ (nnn.nnn.nnn.nnn) 〈符号なし整数〉 ((0～255))

自システムがホットスタンバイ構成である場合、または複数の LAN アダプタを持つ場合に、マルチキャストメッセージの受信に使用する LAN アダプタの IP アドレスを指定します。

指定形式は nnn.nnn.nnn.nnn です。nnn には、0~255 の値を指定できます。ただし、0.0.0.0 と 255.255.255.255 は指定できません。

multicast オペランドに yes を指定した場合、ripaddr オペランドまたは rhostname オペランドのどちらか一方を指定してください。どちらも省略した場合は、OS が使用する LAN アダプタが仮定されます。

rhostname=マルチキャストメッセージの受信に使用する LAN アダプタのホスト名

～〈1～255 文字のホスト名〉

自システムがホットスタンバイ構成である場合、または複数の LAN アダプタを持つ場合に、マルチキャストメッセージの受信に使用する LAN アダプタのホスト名を指定します。

multicast オペランドに yes を指定した場合、rhostname オペランドまたは ripaddr オペランドのどちらか一方を指定してください。どちらも省略した場合は、OS が使用する LAN アダプタが仮定されます。ripaddr オペランドを指定した場合は、このオペランドの指定値は無視されます。

multicastttl=送信するマルチキャストパケットの生存期間 (TTL)

～〈符号なし整数〉((0～255))

マルチキャストメッセージ送信時のマルチキャストパケットの生存期間 (TTL) を指定します。TTL のデフォルト値については、ご使用の OS のマニュアルで、定義ファイルについて説明している個所を参照してください。

注意事項

UDP プロトコルでマルチキャストおよびブロードキャストのメッセージ受信に使用する論理端末には、メッセージの送受信に使用する LAN アダプタの IP アドレス (-r オプションの ipaddr オペランドで指定)、またはメッセージの送受信に使用する LAN アダプタのホスト名 (-r オプションの hostname オペランドで指定) を指定しないでください。指定した場合、マルチキャストおよびブロードキャストのメッセージを受信できません。

eemcplefunc (論理端末アプリケーション定義)

機能

論理端末のアプリケーションに関する環境を定義します。

コマンドの形式

```
[eemcplefunc [-u " [ownlsg=入力セグメント判定UOCのエントリポインタ名]
[ownlmc=入力メッセージ編集UOCのエントリポインタ名]
[ownlra=出力メッセージ編集UOCのエントリポインタ名]
[ownlcn=コネクション確立UOCのエントリポインタ名]
[owntrn=トランザクションレベル名判定UOCの
エントリポインタ名]
[ownojed=OJ編集UOCエントリポインタ名]
[rltrn=MCP後処理トランザクション起動サービス名]
[rltrnlevel=MCP後処理トランザクションの
```

```

        [トランザクションレベル名] "]"
[-e " [errtrn1=ERRTRN1のエントリポインタ名]
        [errtrn2=ERRTRN2のエントリポインタ名]
        [errtrn3=ERRTRN3のエントリポインタ名]
        [errtrnr=ERRTRNRのエントリポインタ名]
        [errtrns=ERRTRNSのエントリポインタ名] "]"

```

オプションの説明

各オペランドで指定するエントリポインタ名は、TP1/EE のユーザサービス関連定義の module オペランドに指定した UAP 共用ライブラリの中に作成してください。

●-u

(オペランド)

ownlsg=入力セグメント判定 UOC のエントリポインタ名

～〈1～31 文字の識別子〉

入力セグメント判定 UOC の C 言語関数名を指定します。

省略した場合はアプリケーション定義 (eemcpfunc) の値が有効になります。

ownlmc=入力メッセージ編集 UOC のエントリポインタ名

～〈1～31 文字の識別子〉

入力メッセージ編集 UOC の C 言語関数名を指定します。

省略した場合はアプリケーション定義 (eemcpfunc) の値が有効になります。

ownlra=出力メッセージ編集 UOC のエントリポインタ名

～〈1～31 文字の識別子〉

出力メッセージ編集 UOC の C 言語関数名を指定します。

省略した場合はアプリケーション定義 (eemcpfunc) の値が有効になります。

ownlcn=コネクション確立 UOC エントリポインタ名

～〈1～31 文字の識別子〉

コネクション確立 UOC の C 言語関数名を指定します。

省略した場合はアプリケーション定義 (eemcpfunc) の値が有効になります。

owntrn=トランザクションレベル名判定 UOC のエントリポインタ名

～〈1～31 文字の識別子〉

トランザクションレベル名判定 UOC の C 言語関数名を指定します。

省略した場合は TP1/FSP のユーザサービス関連定義の uoc_func オペランドの-p オプションの指定値に従います。

このオペランドは、TP1/FSP 使用時にだけ指定できます。

ownojed=OJ 編集 UOC エントリポインタ名

～〈1～31 文字の識別子〉

OJ 編集 UOC の C 言語関数名を指定します。

省略した場合は、TP1/EE サービス定義での uoc_func の-o オプションの定義値に従います。

このオペランドは、TP1/FSP 使用時にだけ指定できます。

rltrn=MCP 後処理トランザクション起動サービス名

～〈1～31 文字の識別子〉

MCP 後処理トランザクションを起動するサービス名を指定します。

ここで指定したサービス名は、TP1/EE のユーザサービス関連定義の service オペランドの値を指定してください。

省略した場合はアプリケーション定義 (eemcpfunc) の値が有効になります。

rltrnlevel=MCP 後処理トランザクションのトランザクションレベル名

～〈1～31 文字の識別子〉

MCP 後処理トランザクションのトランザクションレベル名を指定します。

トランザクションレベル方式で MCP 後処理トランザクション起動サービス名を指定する場合、このオペランドを必ず指定してください。

指定を省略した場合はアプリケーション定義 (eemcpfunc) の指定値が有効になります。

トランザクションレベル方式については、TP1/FSP の関連ドキュメントを参照してください。

このオペランドは、TP1/FSP 使用時にだけ指定できます。

●-e

(オペランド)

errtrn1=ERRTRN1 のエントリポインタ名

～〈1～31 文字の識別子〉

エラートランザクション (ERRTRN1) を提供するエントリポインタ名を指定します。

省略した場合は TP1/EE サービス定義のユーザサービス関連定義の errtrn1 オペランドの指定値に従います。

errtrn2=ERRTRN2 のエントリポインタ名

～〈1～31 文字の識別子〉

エラートランザクション (ERRTRN2) を提供するエントリポインタ名を指定します。

省略した場合は TP1/EE サービス定義のユーザサービス関連定義の errtrn2 オペランドの指定値に従います。

errtrn3=ERRTRN3 のエントリポインタ名

～〈1～31 文字の識別子〉

エラートランザクション (ERRTRN3) を提供するエントリポインタ名を指定します。

省略した場合は TP1/EE サービス定義のユーザサービス関連定義の errtrn3 オペランドの指定値に従います。

errtrnr=ERRTRNR のエントリポインタ名

～〈1～31 文字の識別子〉

エラートランザクション (ERRTRNR) を提供するエントリポインタ名を指定します。
省略した場合は TP1/EE サービス定義のユーザサービス関連定義の errtrnr オペランドの指定値に従います。

errtrns=ERRTRNS のエントリポインタ名

～ <1～31 文字の識別子>

エラートランザクション (ERRTRNS) を提供するエントリポインタ名を指定します。
省略した場合は TP1/EE サービス定義のユーザサービス関連定義の errtrns オペランドの指定値に従います。

eemcphdgrp (スレッドグループ定義)

機能

この定義以降に定義した論理端末定義を新しいスレッドグループとします。

スレッドグループは受信スレッドの作成単位であり、同じスレッドグループに属する論理端末は、同じ受信スレッドでメッセージを受信します。各スレッドグループには、一つ以上の論理端末を定義します。直前のスレッドグループに eemcple 定義コマンドがない場合、この定義は無視されます。

eemcpudpstart 定義コマンドの直後に定義した論理端末は、この定義の有無に関係なく、一つ目のスレッドグループに属します。

この定義を省略した場合、定義したすべての論理端末は、同じスレッドグループに属します。

コマンドの形式

```
{ { [eemcphdgrp] } }
```

注意事項

この定義は、eemcpudpstart 定義コマンドと eemcpudpend 定義コマンドの間で定義してください。なお、定義の記述順序については、「[4.1.2\(3\) 記述順序](#)」を参照してください。

eemcpudpend (UDP 定義の終了)

機能

UDP プロトコルを使用した通信に関する定義の終了を示します。

コマンドの形式

```
[eemcpudpend]
```

注意事項

この定義は eemcpudpstart 定義コマンドと対で定義してください。なお、定義の記述順序については、「4.1.2(3) 記述順序」を参照してください。

eemcpfunc (アプリケーション定義)

機能

アプリケーションに関する環境を定義します。

コマンドの形式

```
[eemcpfunc      [-u " [ownlsg=入力セグメント判定UOCのエントリポインタ名]
                  [ownlmc=入力メッセージ編集UOCのエントリポインタ名]
                  [ownlra=出力メッセージ編集UOCのエントリポインタ名]
                  [ownlcn=コネクション確立UOCのエントリポインタ名]
                  [rltrn=MCP後処理トランザクション起動サービス名]
                  [rltrnlevel=MCP後処理トランザクションの
                               トランザクションレベル名]"]
                  [-e " [errtrns=yes | no]"]
                  [-c " [uocexc=yes | no]"]
                  [-q serial | parallel]
                  [-m " [msglen=normal | extend]
                      [recvpermanence=yes | no]
                      [reconnectsend=yes | no]"]
                  [-o " [outputmessage=0Jの取得先ID]
                      [outputfunction=0Jの機能拡張レベル]"] ]]
```

オプションの説明

●-u

このオプションの各オペランドで指定するエントリポインタ名は TP1/EE のユーザサービス関連定義の module オペランドに指定した UAP 共用ライブラリの中に作成してください。

(オペランド)

ownlsg=入力セグメント判定 UOC のエントリポインタ名

～ <1～31 文字の識別子>

入力セグメント判定 UOC の C 言語関数名を指定します。

省略した場合は入力セグメント判定 UOC を呼び出しません。

ownlmc=入力メッセージ編集 UOC のエントリポインタ名

～ <1～31 文字の識別子>

入力メッセージ編集 UOC の C 言語関数名を指定します。

省略した場合は入力メッセージ編集 UOC を呼び出しません。

eemcple コマンドで-v オプションを省略した論理端末定義が一つでもある場合、このオプションは省略できません。

ownlra=出力メッセージ編集 UOC のエントリポインタ名

～〈1～31 文字の識別子〉

出力メッセージ編集 UOC の C 言語関数名を指定します。

省略した場合は出力メッセージ編集 UOC を呼び出しません。

ownlcn=コネクション確立 UOC エントリポインタ名

～〈1～31 文字の識別子〉

コネクション確立 UOC の C 言語関数名を指定します。

省略した場合はコネクション確立 UOC を呼び出しません。

rltrn=MCP 後処理トランザクション起動サービス名

～〈1～31 文字の識別子〉

MCP 後処理トランザクションを起動するサービス名を指定します。

ここで指定するサービス名には TP1/EE のユーザサービス関連定義の service オペランドで指定したサービス名を指定してください。

省略した場合は MCP 後処理トランザクションを起動しません。

rltrnlevel=MCP 後処理トランザクションのトランザクションレベル名

～〈1～31 文字の識別子〉

MCP 後処理トランザクションのトランザクションレベル名を指定します。

トランザクションレベル方式で MCP 後処理トランザクション起動サービス名を指定する場合、このオペランドを必ず指定してください。

トランザクションレベル方式については、TP1/FSP の関連ドキュメントを参照してください。

このオペランドは、TP1/FSP 使用時にだけ指定できます。

●-e

(オペランド)

errtrns=yes | no

～《yes》

メッセージ送信時に通信障害が発生した場合に送信障害トランザクションを起動するかどうかを指定します。

yes

送信障害トランザクション (ERRTRNS) を起動します。

no

送信障害トランザクション (ERRTRNS) を起動しません。通信障害が発生した場合、送信メッセージは破棄されます。

●-c

(オペランド)

uocexc=yes | no

～《no》

入力メッセージ編集 UOC および出力メッセージ編集 UOC を論理端末単位にシリアル化するかどうかを指定します。

yes

論理端末単位にシリアル化します。入力メッセージ編集 UOC と出力メッセージ編集 UOC が同一排他内で実行されます。

no

論理端末単位にシリアル化しません。入力メッセージ編集 UOC と出力メッセージ編集 UOC がパラレルに実行されます。

●-q serial | parallel

～《serial》

端末キュー機能を使用して、論理端末単位のイベントをシリアルに処理するかどうかを指定します。論理端末単位のイベントには、メッセージ受信、接続確立・解放、および論理端末の閉塞・閉塞解除があります。

端末キュー機能については、「[2.5 端末キュー機能](#)」を参照してください。

serial

端末キュー機能を使用して、論理端末単位のイベントをシリアルに処理します。

parallel

端末キュー機能を使用しないで、論理端末単位のイベントをパラレルに処理します。サービス単位にシリアル処理をする場合に指定します。

●-m

(オペランド)

msglen=normal | extend

～《normal》

送受信メッセージの最大サイズを拡張するかどうかを指定します。

normal

送受信メッセージの最大サイズを拡張しません。送信メッセージの最大サイズは 32 キロバイト、受信メッセージの最大サイズは受信バッファサイズです。

extend

送受信メッセージの最大サイズを拡張します。送受信メッセージの最大サイズは、TCP/IP プロトコルを使用する場合は 8 メガバイト、UDP プロトコルを使用する場合は 64 キロバイトになります。

msglen オペランドの指定値による送受信メッセージの最大サイズの違いについて、次の表に示します。

表 4-4 msglen オペランドの指定値による送受信メッセージの最大サイズの違い

msglen オペランド指定値	送受信メッセージ	送受信メッセージの最大サイズ (単位: バイト)	
		TCP/IP プロトコル使用時	UDP プロトコル使用時
normal	送信メッセージ	32000 ^{※1}	32000 ^{※1}
	受信メッセージ	TP1/EE サービス定義の recv_message_buf_size オペラン ド指定値	65531 ^{※2}
extend	送信メッセージ	8388608	65531 ^{※1}
	受信メッセージ		65531 ^{※2, ※3}

注※1

TP1/EE サービス定義の send_message_buf_size オペランドで指定した送信バッファサイズの方が小さい場合は、
send_message_buf_size オペランドの指定値が設定されます。

注※2

TP1/EE サービス定義の recv_message_buf_size オペランドで指定した受信バッファサイズの方が小さい場合は、
recv_message_buf_size オペランドの指定値が設定されます。

注※3

入力メッセージ編集 UOC での編集後の最大サイズは 8388608 になります。

recvpermanence=yes | no

～ 《no》

クラスタを構成する実行系と待機系の間で、受信メッセージを永続化するかどうかを指定します。この
オペランドは、XTC サービス定義の cluster_mode オペランドに Y を指定している場合にだけ有効に
なります。

yes

受信メッセージを永続化します。

no

受信メッセージを永続化しません。

reconnectsend=yes | no

～ 《プロセス関連定義の fsp_use オペランドが N の場合は yes, Y の場合は no》

メッセージ受信時のコネクション以外の場合（送信元でコネクションが再確立された場合）でも、メッセージ
送信元の論理端末（TCP/IP）に対する応答メッセージを送信するかどうかを指定します。

yes

メッセージ受信時のコネクション以外の場合でも、応答メッセージを送信します。

no

メッセージ受信時のコネクション以外の場合は、応答メッセージを送信しません。

通信プロトコルを持つプログラムが MCP を使用して通信する場合、プロトコルで再確立されたコネクションを利用する応答メッセージの送信を許可していないときは no を指定します。再確立されたコネクションを利用する応答メッセージの送信を許可しているときは yes を指定します。

受信メッセージを永続化する指定で系切り替えが発生した場合、このオペランドに no を指定したときでも、系切り替え前に受信したメッセージに対して応答メッセージを送信します。系切り替え後に受信したメッセージに対する応答メッセージの送信は、このオペランドの指定に従います。

また、このオペランドに no を指定した場合、トランザクション内で任意の論理端末 (TCP/IP) に対する一方送信関数 (ee_mcp_send) を発行しても、コミットする前に送信対象とのコネクションが再確立されると、トランザクションがコミットしてもメッセージは送信されません。

xtc_use=Y の場合はこのオペランドを指定しても無視し、「yes」を指定したものとして動作します。

●-o

このオプションは、TP1/FSP 使用時にだけ指定できます。

(オペランド)

outputmessage=OJ の取得先 ID

～ 〈符号なし整数〉 ((0～255)) 《0》

MCP での送信処理で OJ を取得する場合に、OJ の取得先 ID を指定します。

outputmessage オペランドを省略した場合、またはこのオペランドに 0 を指定した場合は OJ を取得しません。

outputmessage オペランドで指定した取得先 ID によって、OJ を書き込む UAP 履歴情報グループの表を決定します。取得先 ID が指定されていない場合 (TP1/FSP のトランザクション関連定義の trnaphputid 定義コマンドの-i オプションを指定しない場合) は、プロセスダウンします。

UAP 履歴情報取得機能を使用しない場合 (TP1/FSP のトランザクション関連定義の trn_aph_use オペランドに N を設定した場合)、outputmessage オペランドの指定値は無効となります。

TP1/FSP のユーザサービス関連定義の service_attr 定義コマンドの-m オプションを指定している場合、-v オプションで指定したサービス名、または-x オプションで指定したトランザクション種別では、outputmessage オペランドの指定値は無効となります。

outputmessage オペランドと TP1/FSP のユーザサービス関連定義の service_attr 定義コマンドの-m オプションとの関連性、および UAP 履歴情報取得機能については、TP1/FSP の関連ドキュメントを参照してください。

outputfunction=OJ の機能拡張レベル

～ 〈16 進数〉 {00000000 | 00000001} 《00000000》

MCP での OJ 取得の機能拡張レベルを指定します。

TP1/FSP のユーザサービス関連定義の service_attr 定義コマンドの-p オプションを指定している場合、-v オプションで指定したサービス名、または-x オプションで指定したトランザクション種別では、outputfunction オペランドの指定値は無効となります。

00000000

OJ 取得の機能を拡張しません。

00000001

OJ を取得するときに、ユーザデータを出力データとして取得しません。

eemcptrc (トレース定義)

機能

MCP トレースに関する環境を定義します。

コマンドの形式

```
[eemcptrc      [-f  "[fileno=MCP トレースのファイル数]
                  [filesz=MCP トレースの最大ファイルサイズ]"]
                  [-b  "[bufferno=MCP トレースファイル出力用バッファ数]
                  [buffersz=MCP トレースファイル出力用最大
                  バッファサイズ]"]
                  [-x  "[entrysize=スレッドごとのMCP トレースの
                  バッファサイズ]"]
                  [-s  "[sendresult=yes | no]"]
                  [-m  "[filter=MCP トレース情報の取得量の変更]
                  [msgsize=MCP トレース情報の取得最大サイズ]"]]
```

オプションの説明

●-f

(オペランド)

fileno=MCP トレースのファイル数

～ 〈符号なし整数〉 ((3～100)) 《3》

MCP トレース情報を出力する最大ファイル数を指定します。

filesz=MCP トレースの最大ファイルサイズ

～ 〈符号なし整数〉 ((400～4000000)) 《40000》 (単位：キロバイト)

MCP トレース情報を取得するファイルの最大サイズを指定します。次に示す条件式を満たすように指定してください。

-fオプションのfileszオペランド > -bオプションのbufferszオペランド > -xオプションのentrysizeオペランド

●-b

(オペランド)

bufferno=MCP トレースファイル出力用バッファ数

～ 〈符号なし整数〉 ((3～200)) 《3》

MCP トレース情報をバッファにスタックするバッファ数を指定します。

MCP トレース編集時に破棄したレコード数が output された場合は、このオペランドの設定値を増やしてください。このオペランドに最大値を設定しても破棄したレコード数が output された場合は、buffersz オペランドの設定値を増やしてください。

buffersz=MCP トレースファイル出力用最大バッファサイズ

～ 〈符号なし整数〉 ((41～20000)) 《200》 (単位：キロバイト)

MCP トレース情報をバッファにスタックするバッファサイズを指定します。次に示す条件式を満たすように指定してください。

-f オプションのfilesz オペランド > -b オプションのbuffersz オペランド > -x オプションのentrysize オペランド

MCP トレース編集時に破棄したレコード数が output された場合は、最初に bufferno オペランドの設定値を増やしてください。bufferno オペランドに最大値を設定しても破棄したレコード数が output される場合は、このオペランドの設定値を増やしてください。

●-x

(オペランド)

entrysize=スレッドごとの MCP トレースのバッファサイズ

～ 〈符号なし整数〉 ((40～10000)) 《80》 (単位：キロバイト)

スレッド単位で格納する一時バッファ領域のサイズを指定します。次に示す条件式を満たすように指定してください。

-f オプションのfilesz オペランド > -b オプションのbuffersz オペランド > -x オプションのentrysize オペランド

●-s

(オペランド)

sendresult=yes | no

～ 《no》

出力メッセージ編集 UOC でユーザトレースを取得した場合、送信結果トレースを取得するかどうかを指定します。

yes

送信結果トレースを取得します。

no

送信結果トレースを取得しません。

●-m

(オペランド)

filter=MCP トレース情報の取得量の変更

～ 〈16 進数〉 {00000000 | 00000010} 《00000000》

MCP トレース情報の取得量を変更する場合に、変更する項目を指定します。

00000000

すべての送受信イベントを MCP トレース情報に取得します。送受信データは、1 回の送受信イベントごとに最大で 32000 バイトを取得します。

00000010

すべての送受信イベントを MCP トレース情報に取得します。送受信データは、1 回の送受信イベントごとに-m オプションの msgsize オペランドに指定した最大サイズまでを取得します。

msgsize=MCP トレース情報の取得最大サイズ

～ 〈符号なし整数〉 ((0～32000)) 《128》 (単位：バイト)

送受信データのうち、MCP トレース情報に取得する最大サイズを指定します。

送受信データが最大サイズより小さい場合は、送受信データのすべての内容を MCP トレース情報として取得します。0 を指定した場合、送受信データ長 0 として MCP トレース情報を取得します。

このオペランドは、-m オプションの filter オペランドに 00000010 を指定している場合だけ有効です。

4.4 コネクションの形態と通信構成定義との関係

TCP/IP プロトコルを使用して通信する場合のコネクションの形態と通信構成定義との関係を次の表に示します。

表 4-5 コネクションの形態と通信構成定義との関係

コマンド	オプション	オペランド	コネクション確立モード	
			クライアント	サーバ
eemcpcn (コネクション定義)	-c	-	○	○
	-i	-	△	×
	-b	bretry	△	×
		bretrycnt	△	×
		bretryint	△	×
		concmptim	△	×
		conmsg	△	△
		sndsynctim	△	△
		sndtim	△	△
	-w	srtimout	△	△
	-y	mode	○	○
	-r	ipaddr	△	△
		hostname	△	△
		portno	△	○
	-o	oipaddr	○	△
		ohostname	○	△
		oportno	○	△
	-k	nodelay	△	△
		notrftime	△	△
		keepalive	△	△
	-f	cnrelease	△	△
		endrelease	△	△
	-h	addrchk	×	△
		chgconn	×	△
eemcple (論理端末定義)	-l	-	○	○

コマンド	オプション	オペランド	コネクション確立モード	
			クライアント	サーバ
eemcple (論理端末定義)	-i	-	△	△
	-v	-	△	△
eemcptcpstart (TCP/IP 定義の開始)	-o	con_addr	△	×

(凡例)

－：該当しません。

○：指定が必要です。

△：指定は任意です。または、定義の組み合わせによって省略できます。

×：指定できません、または、指定しても無視されます。

4.5 システム定義での注意事項

ここでは、MCP のシステム定義での注意事項について説明します。

4.5.1 論理端末の定義数

TP1/EE の受信スレッド作成数に関係して、MCP の論理端末の定義数には上限があります。

■MCP の受信スレッド作成数の上限

- UDP 全体：スレッドグループ数
- TCP/IP (クライアント) 全体：1
- TCP/IP (サーバ) : サーバ型コネクション (eemcpcn 定義-y の mode オペランドで server を指定) である eemcpcn 定義-r オプションで指定した IP アドレスまたはホスト名と、ポート番号の組み合わせ数

TP1/EE のプロセス全体で作成できるスレッドの上限は 1000 です。MCP 構成定義での指定範囲内の定義数であっても、TP1/EE のスレッド（処理スレッド、受信スレッド数など）の関係で TP1/EE が起動できなくなることがあるため、注意してください。

■MCP の論理端末数の上限

TCP/IP プロトコル使用時のサーバ型コネクション、クライアント型コネクションおよび UDP プロトコル使用時を含めた論理端末の上限は 32000 です。ただし、論理端末は、一つ以上定義してください。

4.5.2 MCP でのファイル識別子の使用量

TP1/EE のプロセス関連定義の max_descriptors オペランドの指定値に、MCP でのファイル識別子の使用量を加算する必要があります。

■MCP でのファイル識別子の使用量

- UDP 全体：(2×受信スレッド数) + ポート数
- TCP/IP (クライアント) 全体：2 + コネクション数
- TCP/IP (サーバ) : (3×受信スレッド数) + コネクション数

論理端末の定義数に応じてファイルディスクリプタ数 (TP1/EE の max_descriptors 定義値、TP1/SB の max_socket_descriptors,max_open_fds 定義値、および OS のファイルディスクリプタ数) を変更してください。

また、MCP 定義で範囲内の定義数でも、TP1/EE のスレッド（処理スレッド、受信スレッド数など）の関係や、ファイルディスクリプタ数の関係で TP1/EE の起動ができなくなるおそれがあります。TP1/EE のファイルディスクリプタ数については、マニュアル「TP1/Server Base Enterprise Option 使用の手

引」を、TP1/SB のファイルディスクリプタ数に関してはマニュアル「OpenTP1 システム定義」を参照してください。

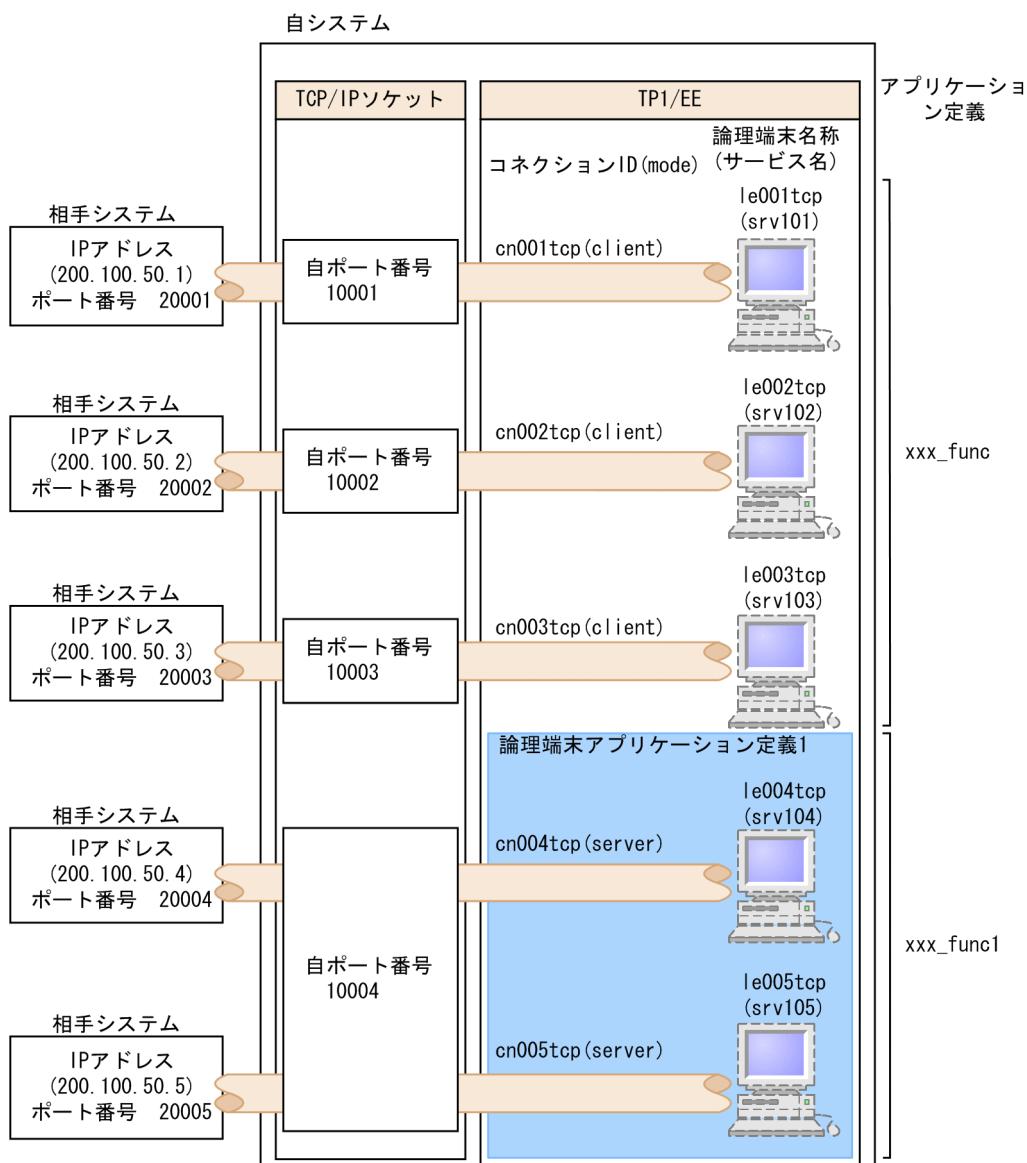
4.6 定義例

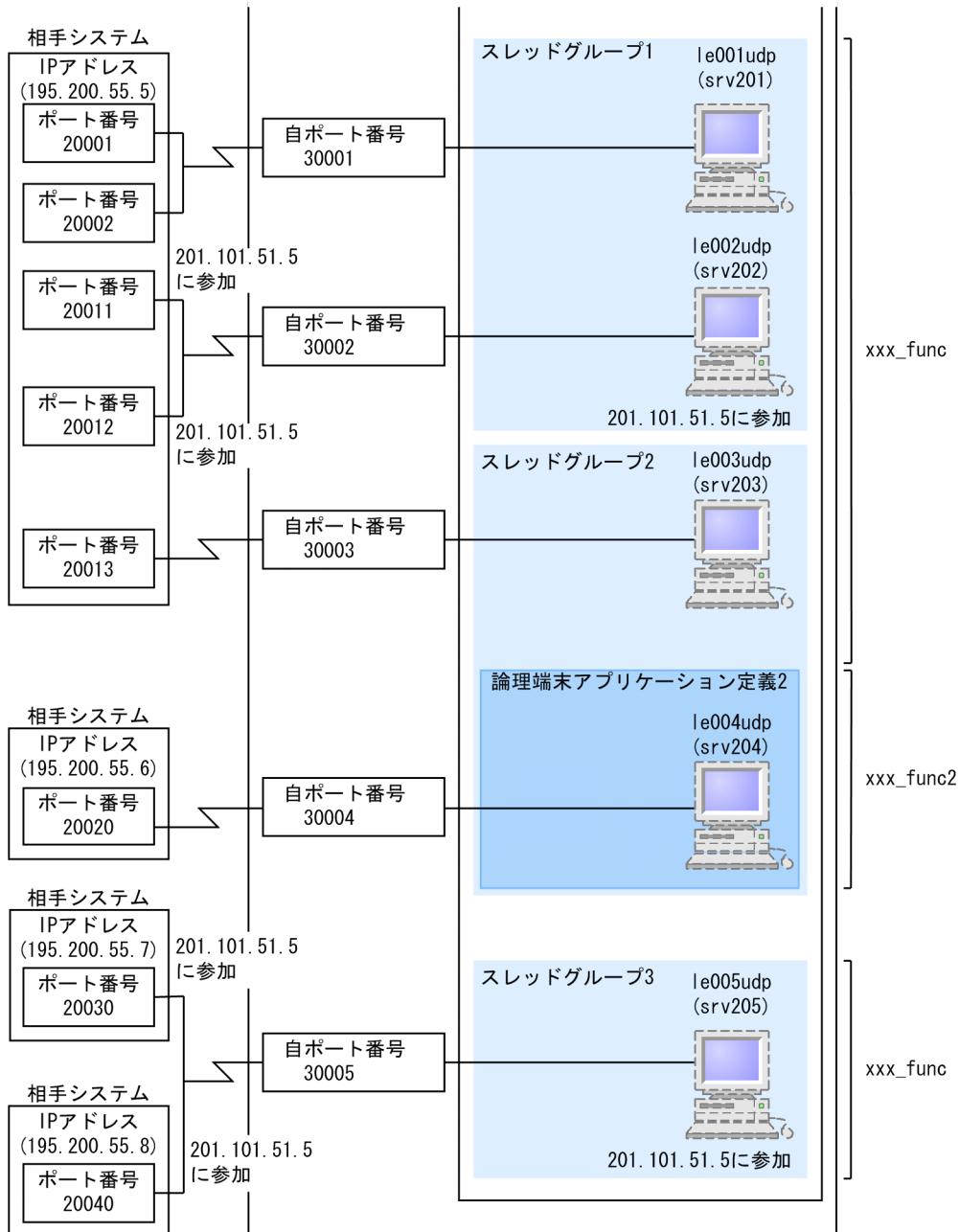
ここでは、システム構成例に対する定義のコーディング例を示します。

4.6.1 MCP を使用したシステム構成例

MCP を使用したシステム構成例を次の図に示します。

図 4-4 MCP のシステム構成例





4.6.2 定義のコーディング例

システム構成例に沿った定義のコーディング例を次に示します。

```

#-----#
# MCP構成定義
#-----#
#-----#
eemcptcpstart          #TCP/IP定義の開始
#-----#
# コネクション定義(cn001tcp : クライアント)
eemcpcn -c cn001tcp -y "mode=client" -i auto      ¥
    -b "retry=yes retrycnt=10 retryint=3 concmptim=5" ¥

```

```

        conmsg=yes sndsynctim=3 sndtim=5"                                ¥
- r "portno=10001"                                                 ¥
- o "oipaddr=200.100.50.1 oportno=20001"                            ¥
#-----
# 論理端末定義 (le001tcp)
eemcple -l le001tcp -i auto -v srv101

#-----
# コネクション定義(cn002tcp : クライアント)
eemcpcn -c cn002tcp -y "mode=client"                                ¥
- r "portno=10002"                                                 ¥
- o "oipaddr=200.100.50.2 oportno=20002"                            ¥
#-----
# 論理端末定義 (le002tcp)
eemcple -l le002tcp -v srv102

#-----
# コネクション定義(cn003tcp : クライアント)
eemcpcn -c cn003tcp -y "mode=client"                                ¥
- r "portno=10003"                                                 ¥
- o "oipaddr=200.100.50.3 oportno=20003"                            ¥
#-----
# 論理端末定義 (le003tcp)
eemcple -l le003tcp -v srv103

#-----
# 論理端末アプリケーション定義1 (xxx_func1)
eemcplefunc -u "ownlsg=lsg_func1"                                     ¥
  ownlmc=lmc_func1                                                 ¥
  ownlra=lra_func1                                                 ¥
  ownlcn=lcn_func1                                                 ¥
  owntrn=trn_func1                                                 ¥
  ownojed=oj_func1                                                 ¥
  rltrn=rl_func1                                                 ¥
  rltrnlevel=trnlevel1"                                             ¥
- e "errtrn1=err1_func1"                                              ¥
  errtrn2=err2_func1"                                              ¥
  errtrn3=err3_func1"                                              ¥
  errtrnr=errr_func1"                                              ¥
  errtrns=errs_func1"                                              ¥
#-----
# コネクション定義(cn004tcp : サーバ)
eemcpcn -c cn004tcp -y "mode=server"                                ¥
- r "portno=10004"                                                 ¥
- o "oipaddr=200.100.50.4 oportno=20004"                            ¥
- h "addrchk=yes chgconn=replace"                                    ¥
#-----
# 論理端末定義 (le004tcp)
eemcple -l le004tcp -v srv104

#-----
# コネクション定義(cn005tcp : サーバ)
eemcpcn -c cn005tcp -y "mode=server"                                ¥
- r "portno=10004"                                                 ¥
- o "oipaddr=200.100.50.5 oportno=20005"                            ¥
- h "chgconn=replace"                                              ¥
#-----
```

```

# 論理端末定義 (le005tcp)
eemcple -l le005tcp -v srv105
#-----
eemcptcpPEND                                # TCP/IP定義の終了

#-----
# eemcpudpstart                                # UDP定義の開始
#-----
# スレッドグループ定義 (スレッドグループ1)
eemcpthdgrp
#-----
# 論理端末定義 (le001udp)
eemcple -l le001udp -i auto -v srv201          ¥
  -b "sndtim=0"                                ¥
  -s "syssndsize=10240 sysrcvsize=10240"        ¥
  -r "portno=30001"                            ¥
#-----
# 論理端末定義 (le002udp)
eemcple -l le002udp -v srv202          ¥
  -r "portno=30002"                            ¥
  -m "multicast=yes hostgroupaddr=201.101.51.5"  ¥
#-----
# スレッドグループ定義 (スレッドグループ2)
eemcpthdgrp
#-----
# 論理端末定義 (le003udp)
eemcple -l le003udp -v srv203          ¥
  -r "portno=30003"                            ¥
#-----
# 論理端末アプリケーション定義2 (xxx_func2)
eemcplefunc -u "ownlsg=lsg_func2          ¥
  ownlmc=lmc_func2                            ¥
  ownlra=lra_func2                            ¥
  ownlcn=lcn_func2                            ¥
  owntrn=trn_func2                            ¥
  ownojed=oj_func2                            ¥
  rltrn=rl_func2                            ¥
  rltrnlevel=trnlevel2"                      ¥
  -e "errtrn1=err1_func2                      ¥
  errtrn2=err2_func2                            ¥
  errtrn3=err3_func2                            ¥
  errtrnr=errr_func2                            ¥
  errtrns=errs_func2"                          ¥
#-----
# 論理端末定義 (le004udp)
eemcple -l le004udp -v srv204          ¥
  -r "portno=30004"                            ¥
#-----
# スレッドグループ定義 (スレッドグループ3)
eemcpthdgrp
#-----
# 論理端末アプリケーション定義をデフォルトに戻す
eemcplefunc
#-----
# 論理端末定義 (le005udp)
eemcple -l le005udp -v srv205          ¥
  -r "portno=30005"                            ¥

```

```

-m "multicast=yes hostgroupaddr=201.101.51.5"
#----- eemcpudpend # UDP定義の終了

#---- トレース定義 -----
eemcptrc -f "fileno=3" filesz=4000" ¥
#---- バッファ定義 -----
eemcpbuf -e "emsgbuf=10240" # UOC編集用バッファサイズ10240

#---- アプリケーション定義 (xxx_func) -----
eemcpfunc -u "ownlsg=lsg_func" ¥
    ownlmc=lmc_func ¥
    ownlra=lra_func ¥
    ownlcn=lcn_func ¥
    rltrn=rl_func ¥
    rltrnlevel=trnlevel" ¥
-e "errtrns=yes" ¥
-c "uocexc=yes" ¥

```

5

MCP のライブラリ関数の文法

この章では、MCP のライブラリ関数と UOC の文法について説明します。

5.1 MCP のライブラリ関数の一覧

C 言語, COBOL 言語のライブラリ関数の対応を次の表に示します。

表 5-1 MCP のライブラリ関数の対応

機能	ライブラリ関数名	
	C 言語	COBOL 言語
コネクションの確立	ee_mcp_actcn	CBLEEMCP('ACTCN ')
相手システムを指定したコネクションの確立	ee_mcp_actcn_to	CBLEEMCP('ACTCNOTO ')
論理端末の閉塞解除	ee_mcp_actle	CBLEEMCP('ACTLE ')
コネクション情報の取得	ee_mcp_cninfo	CBLEEMCP('CNINFO ')
コネクションの解放	ee_mcp_dctcn	CBLEEMCP('DCTCN ')
論理端末の閉塞	ee_mcp_dctle	CBLEEMCP('DCTLE ')
非同期一方送信メッセージの送信	ee_mcp_send	CBLEEMCP('SEND ')
同期型メッセージの送受信	ee_mcp_sendrecv	CBLEEMCP('SENDRECV ')
同期一方送信メッセージの送信	ee_mcp_sendsync	CBLEEMCP('SENDSYNC ')
ユーザトレースの取得	ee_mcp_utrace_put	CBLEEMCP('UTPUT ')
ユーザトレースの取得 (データ長拡張用)	ee_mcp_utrace_put_long	CBLEEMCP('UTPUTL ')

SPP で使用できる MCP のライブラリ関数は、TP1/EE の各製品のトランザクション種別によって異なります。トランザクションで使用できるライブラリ関数を、製品別に以降の表に示します。なお、トランザクション種別については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

表 5-2 トランザクションで使用できるライブラリ関数 (TP1/EE)

ライブラリ関数名	トランザクション種別									
	MI	ME	MN	E1	E2	E3	E4	ES	ER	TM
ee_mcp_actcn CBLEEMCP('ACTCN ')	×	×	○	○	○	○	○	○	○	○
ee_mcp_actcn_to CBLEEMCP('ACTCNOTO ')	×	×	○	○	○	○	○	○	○	○
ee_mcp_actle CBLEEMCP('ACTLE ')	×	×	○	○	○	○	○	○	○	○
ee_mcp_cninfo CBLEEMCP('CNINFO ')	○	○	○	○	○	○	○	○	○	○
ee_mcp_dctcn	○	○	○	○	○	○	○	○	○	○

ライブラリ関数名	トランザクション種別									
	MI	ME	MN	E1	E2	E3	E4	ES	ER	TM
CBLEEMCP('DCTCN')	○	○	○	○	○	○	○	○	○	○
ee_mcp_dctle	○	○	○	○	○	○	○	○	○	○
CBLEEMCP('DCTLE')										
ee_mcp_send	×	○	○	○	○	○	○	○	○	○
CBLEEMCP('SEND')										
ee_mcp_sendrecv	×	×	○	○	○	○	○	○	○	○
CBLEEMCP('SENDRECV')										
ee_mcp_sendsync	×	○	○	○	○	○	○	○	○	○
CBLEEMCP('SENDSYNC')										
ee_mcp_utrace_put	○	○	○	○	○	○	○	○	○	○
CBLEEMCP('UTPUT')										
ee_mcp_utrace_put_long	○	○	○	○	○	○	○	○	○	○
CBLEEMCP('UTPUTL')										

(凡例)

○：使用できます。

×：使用できません。

表 5-3 トランザクションで使用できるライブラリ関数 (MCP, TP1/FSP)

ライブラリ関数名	トランザクション種別						
	UI	MV	RL	UT	CU	JI	BS
ee_mcp_actcn	×	○	○	×	○	×	×
CBLEEMCP('ACTCN')							
ee_mcp_actcn_to	×	○	○	×	○	×	×
CBLEEMCP('ACTCNTO')							
ee_mcp_actle	×	○	○	×	○	×	×
CBLEEMCP('ACTLE')							
ee_mcp_cninfo	○	○	○	○	○	○	×
CBLEEMCP('CNINFO')							
ee_mcp_dctcn	○	○	○	○	○	○	×
CBLEEMCP('DCTCN')							
ee_mcp_dctle	○	○	○	○	○	○	×
CBLEEMCP('DCTLE')							
ee_mcp_send	×	○	○	×	○	×	×
CBLEEMCP('SEND')							

ライブラリ関数名	トランザクション種別						
	UI	MV	RL	UT	CU	JI	BS
ee_mcp_sendrecv CBLEEMCP('SENDRECV')	×	○	○	×	○	×	×
ee_mcp_sendsync CBLEEMCP('SENDSYNC')	×	○	○	×	○	×	×
ee_mcp_utrace_put CBLEEMCP('UTPUT')	○	○	○	○	○	○	×
ee_mcp_utrace_put_long CBLEEMCP('UTPUTL')	○	○	○	○	○	○	×

(凡例)

○ : 使用できます。

× : 使用できません。

5.2 ライブライリ関数 (C 言語)

C 言語のライブラリ関数について説明します。

関数の説明形式

MCP のライブラリ関数の説明形式については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

サービス関数 (SPP) の作成

SPP はサービスを提供する UAP です。ここでは、MCP での SPP の文法について説明します。

形式

ANSI C, C++の形式

```
#include <eerpccsr.h>
#include <eemcp.h>

void 関数名(char *in,  EEULONG *in_len,  char *out,  EEULONG *out_len,
EERPC_INTERFACE_TBL *trninf)
{
    サービスの処理
}

/* EERPC_INTERFACE_TBL構造体の形式 */
EERPC_INTERFACE_TBL {
    EEULONG    trn_len;
    EEULONG    group_len;
    char      servicegroup[32];
    EEULONG    service_len;
    char      service[32];
    char      trn_id[2];
    char      auto_block;
    char      commit_inf;
    EEULONG    thread_no;
    EEULONG    err_code;
    EEULONG    uifa_len;
    char      yobi_1[4];
    char      *uifa;
    EEULONG    usat_len;
    char      yobi_2[4];
    char      *usat;
    char      ans_inf;
    char      msg_inf;
    char      start_inf;
    char      before_end_inf;
    char      end_inf;
    char      thread_down_inf;
```

```

char      uap_errtrn_inf;
char      trninf_ex;
EEULONG  rm_no;
EEULONG  rm_inf;
EEULONG  es_group_len;
char      es_servicegroup[32];
EEULONG  es_service_len;
char      es_service[32];
EEULONG  abn_thread_no;
char      yobi_3[4];
union {
    TRNINF_SERVICE_EX  *trninf_service_ex;
    TRNINF_DBQ_EX      *trninf_dbq_ex;
    TRNINF_DBQ_OBS_EX *trninf_dbq_obs_ex;
} ex_inf;
void      *xtc_pt;
void      *mcp_pt;
:
:
}

```

機能

サービスを実行する SPP のサービス関数です。サービス関数は、上記の形式で任意に作成してください。

サービス関数名は、サービス関数のエントリポインタ（入り口点）の名称に対応します。この対応づけは、UAP の実行環境を設定するときに指定します。UAP の実行環境を設定する手順を次に示します。

1. TP1/EE での設定

ユーザサービス関連定義で指定します。詳細については、マニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。

2. MCP での設定

MCP 後処理トランザクション起動サービス名は、eemcpfunc 定義コマンドで指定します。詳細については、「[4.3 MCP 構成定義の詳細](#)」の「[eemcpfunc \(アプリケーション定義\)](#)」を参照してください。

引数の指定

ここでは、引数の指定のうち、TP1/EE と異なる部分について説明します。そのほかの引数情報については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

トランザクション種別が RL の場合にサービス関数に値が渡される引数を次の表に示します。

表 5-4 トランザクション種別が RL の場合にサービス関数に値が渡される引数

項番	引数情報	参照可否
1	入力パラメタ	×
2	入力パラメタ長	×
3	サービス関数の応答	×

項目番	引数情報	参照可否
4	応答の長さ	×
5	トランザクションインターフェース情報	○

(凡例)

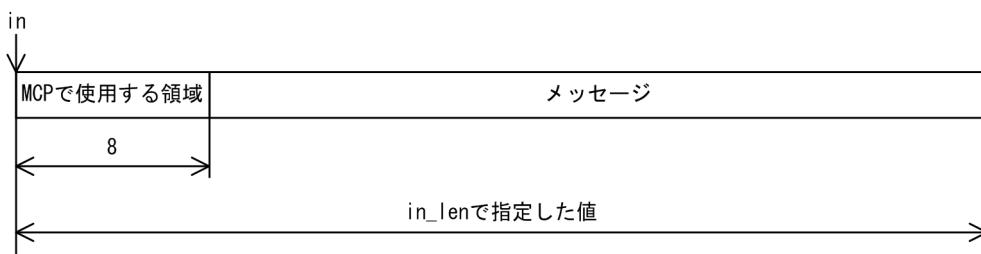
○：参照できます。

×：参照できません。

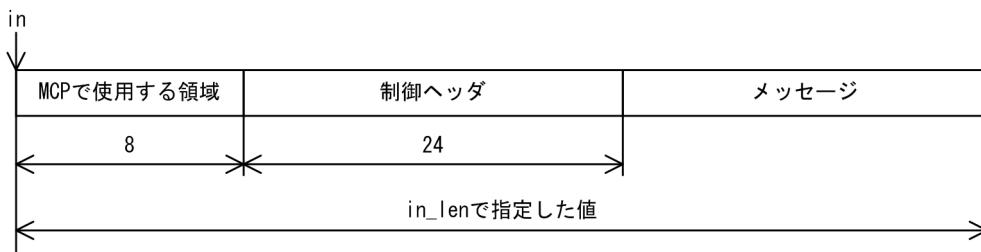
■入力パラメタと入力パラメタ長の関係

入力パラメタ (in) と入力パラメタ長 (in_len) の関係を次の図に示します (単位：バイト)。

TCP/IP の場合



UDP の場合



制御ヘッダについては、「[5.2 ee_mcp_send - 非同期一方送信メッセージの送信](#)」の「UAP で値を設定する引数」を参照してください。

なお、MCP ではサービス関数の応答 (out) を使用した応答送信はできません。

クライアント UAP から値が渡される引数

トランザクション種別が ES の場合に TP1/EE と設定値が異なる引数を次に示します。そのほかの引数については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」およびマニュアル「TP1/EE/Extended Transaction Controller 使用の手引」を参照してください。

●in

トランザクションと同期して送信する非同期一方送信のメッセージ送信 API (ee_mcp_send 関数) の入力パラメタ (senddata) に設定した値 (出力メッセージ編集 UOC 指定時は編集後の値※) が渡されます。

注※

メッセージの最大サイズを拡張している場合に、出力メッセージ編集 UOC で不正なサイズを指定したときは、編集後の値を 0 バイトとします。また、入力メッセージ編集 UOC で編集後に送信バッファ不足が発生した場合、編集後の値の末尾を切り捨てます。

●in_len

トランザクションと同期して送信する非同期一方送信のメッセージ送信 API (ee_mcp_send 関数) の入力パラメタ (sdataleng) に設定した値 (出力メッセージ編集 UOC 指定時は編集後の値※) に 8 を加算した値が渡されます。

注※

メッセージの最大サイズを拡張している場合に、出力メッセージ編集 UOC で不正なサイズを指定したときは、編集後の値を 0 バイトとします。また、入力メッセージ編集 UOC で編集後に送信バッファ不足が発生した場合、編集後の値の末尾を切り捨てます。

●out_len

参照できません。

その他

次に示す内容については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

- TP1/EE から値が返される引数
- UAP で値を設定する引数
- サービス関数の処理での注意
- トランザクションとサービス関数の関係
- リターン値

トランザクションインターフェース情報

ここでは、次に示す内容について説明します。

- MCP 後処理トランザクション
- エラートランザクション (ERRTRNS) ※
- MCP 拡張トランザクションインターフェース情報

注※

TP1/EE と異なる部分についてだけ説明します。

上記以外のトランザクションインターフェース情報の詳細については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

MCP 後処理トランザクション

MCP 後処理トランザクションは MCP 固有のトランザクション種別です。MCP 後処理トランザクションは、次に示すタイミングで起動します。

- ・コネクションの確立または解放
- ・論理端末の閉塞または閉塞解除

トランザクション種別が RL の場合に参照できるトランザクションインターフェース情報を次の表に示します。各トランザクションインターフェース情報については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

表 5-5 トランザクション種別が RL の場合のトランザクションインターフェース情報

項目番	トランザクションインターフェース情報	参照可否
1	情報長 (trn_len)	○
2	サービスグループ名長 (group_len)	○
3	サービスグループ名 (servicegroup)	○
4	サービス名長 (service_len)	○
5	サービス名 (service)	○
6	トランザクション種別 (trn_id)	○
7	自動閉塞有無 (auto_block)	×
8	コミット有無 (commit_inf)	×
9	処理スレッド通番 (thread_no)	○
10	エラー要因コード (err_code)	×
11	ユーザ IFA エリアサイズ (uifa_len)	○
12	ユーザ IFA エリアポインタ (*uifa)	○
13	ユーザ SAT エリアサイズ (usat_len)	○
14	ユーザ SAT エリアポインタ (*usat)	○
15	応答要否 (ans_inf)	×
16	メッセージ状態 (msg_inf)	○
17	プロセス起動要因 (start_inf)	×
18	前回プロセス終了要因 (before_end_inf)	×
19	プロセス終了要因 (end_inf)	×

項目番	トランザクションインターフェース情報	参照可否
20	スレッドダウン種別 (thread_down_inf)	×
21	異常発生トランザクション種別 (uap_errtrn_inf)	×
22	拡張トランザクション種別 (trninf_ex)	×
23	リソースマネジャ数 (rm_no)	○
24	リソースマネジャ接続状態 (rm_inf)	○
25	送信サービスグループ名長 (es_group_len)	×
26	送信サービスグループ名 (es_servicegroup)	×
27	送信サービス名長 (es_service_len)	×
28	送信サービス名 (es_service)	×
29	異常発生処理スレッド通番 (abn_thread_no)	×
30	拡張トランザクションインターフェース情報 (ex_inf)	×
31	XTC 拡張トランザクションインターフェース情報 (*xtc_pt)	△
32	MCP 拡張トランザクションインターフェース情報 (*mcp_pt)	○
33	メッセージ種別 (msg_type)	×
34	エラートランザクション要因 (errtrn_factor)	×
35	拡張ユーザIFA エリアサイズ (uifa_large_len)	○
36	拡張ユーザSAT エリアサイズ (usat_large_len)	○
37	トランザクション起動要因 (run_inf)	×
38	サーバ再起動回数 (rerun_cnt)	×
39	トランザクション起動時刻 (start_time)	○
40	送信元のノード識別子 (node_id)	×
41	トランザクションレベル名長 (trnlevel_len)	○
42	トランザクションレベル名 (trnlevel)	○

(凡例)

○：参照できます。

×：参照できません。

△：XTC を使用している場合だけ参照できます。

エラートランザクション (ERRTRNS)

トランザクション種別が ES の場合に、TP1/EE と設定値が異なるエラー要因コード (err_code) を次に示します。

■エラー要因コード

EEMCPER_MEMORY_SHORT

メモリ不足が発生しました。

EEMCPER_SENDER

送信障害が発生しました。

EEMCPER_TERM_HOLD

論理端末が閉塞中です。

EEMCPER_TIME

送信タイムアウトが発生しました。

EEMCPER_UOC_ERR

出力メッセージ編集 UOC がエラーリターンしました。

EEMCPER_UOC_INVALID

出力メッセージ編集 UOC で障害が発生しました。

次のトランザクションインターフェース情報については TP1/EE から SPP に渡されません。

- 送信サービスグループ名長
- 送信サービスグループ名
- 送信サービス名長
- 送信サービス名

MCP 拡張トランザクションインターフェース情報

MCP 拡張トランザクションインターフェース情報は、MCP エリアポインタ mcp_pt に設定される情報です。次に示す場合に参照できます。

- トランザクション種別が「MN」「E1」「E2」「E3」「ES」「ER」であり、かつメッセージ種別 (msg_type) が EEMCP_MSGTYPE_TCP または EEMCP_MSGTYPE_UDP の場合
- トランザクション種別が RL の場合

MCP エリアポインタ mcp_pt が示す構造体 (TRNINF_SERVICE_MCP) の形式を次に示します。

```
TRNINF_SERVICE_MCP {
    EELONG      cn_name_len;
    char        cn_name[9];
    char        yobi_1[3];
    EELONG      le_name_len;
    char        le_name[9];
    char        yobi_2[3];
    EELONG      pro_kind;
    EELONG      kind;
    EELONG      reason;
    char        yobi_3[4];
```

```

void      *data1;
void      *lebuf;
EEULONG   lebuf_size;
char      yobi_4[4];
unsigned char r_ipaddr[16];
unsigned char o_ipaddr[16];
unsigned short r_port;
unsigned short o_port;
char      yobi[28];
}

```

●cn_name_len

コネクション ID のサイズが設定されます。

●cn_name

コネクション ID が設定されます。

●le_name_len

メッセージの入出力論理端末名のサイズが設定されます。

●le_name

メッセージの入出力論理端末名が設定されます。

●pro_kind

プロトコル種別として、次の値が設定されます。

EEMCP_UOC_PRO_TCP : TCP/IP プロトコル

EEMCP_UOC_PRO_UDP : UDP プロトコル

●kind

種別として、次の値が設定されます。

EEMCP_CN_OPEN : コネクションの確立通知

EEMCP_CN_CLOSE : コネクションの解放通知

EEMCP_LE_FREE : 論理端末の閉塞解除通知

EEMCP_LE_HOLD : 論理端末の閉塞通知

●reason

kind で設定される値の reason 値の意味を次の表に示します。

表 5-6 各 kind で設定される値の reason 値の意味

項目番	kind 設定値	reason 値	reason 値 (数値)	意味	プロト コル
1	EEMCP_CN_OPEN	EEMCP_BY_API	1	ee_mcp_actcn 関数 または ee_mcp_actcn_to 関 数によるコネクション の確立	T
2		EEMCP_BY_COMMAND	2	eemcpactcn コマン ドまたは eemcpactcnto コマ ンドによるコネクショ ンの確立	T
3		EEMCP_BY_SYSTEM	5	システムによるコネク ションの確立	T
4		EEMCP_BY_API	1	ee_mcp_dctcn 関数 (EEMCPNRM 設定 時) によるコネクショ ンの解放	T
5		EEMCP_BY_COMMAND	2	eemcpdctcn コマン ドによるコネクション の解放	T
6		EEMCP_BY_API_F	3	ee_mcp_dctcn 関数 (EEMCPFRC 設定 時) によるコネクショ ンの強制解放	T
7		EEMCP_BY_COMMAND_F	4	eemcpdctcn コマン ド (-f オプションを指 定時 ^{※1}) によるコネ クションの強制解放	T
8		EEMCP_BY_SYSTEM	5	システムによるコネク ションの解放	T
9		EEMCP_BY_SYSTEM_F	6	システムによるコネク ションの強制解放	T
10		EEMCP_RSN_CNNC	101	コネクション確立不可 ^{※2}	T
11		EEMCP_RSN_CNER	102	コネクション解放 (相 手システムからの解 放)	T
12		EEMCP_RSN_CNCL	103	コネクション解放 (自 システムからの解放)	T

項目番	kind 設定値	reason 値	reason 値 (数値)	意味	プロト コル
13	EEMCP_CN_CLOSE	EEMCP_RSN_RBOV	104	受信バッファオーバーフロー	T
14		EEMCP_RSN_NOBF	105	受信バッファ不足	T
15		EEMCP_RSN_NTTO	106	後続メッセージ受信タイムアウト	T
16		EEMCP_RSN_RPLC	107	コネクションリプレースによる切断	T
17		EEMCP_RSN_INTV	108	タイマ値不正	T
18		EEMCP_RSN_TICD	109	タイマセット指示種別不正	T
19		EEMCP_RSN_SGED	110	セグメント完/未完種別不正	T
20		EEMCP_RSN_NTSZ	111	次メッセージのサイズ不正	T
21		EEMCP_RSN_NTAD	112	次メッセージの先頭アドレス不正	T
22		EEMCP_RSN_NWSZ	113	該当メッセージのサイズ不正	T
23		EEMCP_RSN_NWSS	114	残留該当メッセージのサイズ不正	T
24		EEMCP_RSN_NRTO	121	無通信状態監視タイムアウト	T
25		EEMCP_RSN_SRTO	122	同期型メッセージの送受信タイムアウト	T
26		UOC からのリターンコード	UOC からの リターンコー ド	入力セグメント判定 UOC エラーリターン	T
27	EEMCP_LE_FREE	EEMCP_BY_API	1	ee_mcp_actle 関数による論理端末の閉塞解除	T, U
28		EEMCP_BY_COMMAND	2	eemcpactle コマンドによる論理端末の閉塞解除	T, U
29		EEMCP_BY_SYSTEM	5	システムによる論理端末の閉塞解除	T, U
30	EEMCP_LE_HOLD	EEMCP_BY_API	1	ee_mcp_dctle 関数による論理端末の閉塞	T, U

項目番	kind 設定値	reason 値	reason 値 (数値)	意味	プロト コル
31	EEMCP_LE_HOLD	EEMCP_BY_COMMAND	2	eemcpdctle コマンドによる論理端末の閉塞	T, U
32		EEMCP_BY_SYSTEM	5	システムによる論理端末の閉塞	T, U
33		EEMCP_RSN_RBOV	104	受信バッファオーバフロー	T, U
34		EEMCP_RSN_NOBF	105	送受信バッファ不足	T, U
35		EEMCP_RSN_SBOV	115	送信バッファオーバフロー	T, U
36		EEMCP_RSN_BCNT	116	使用バッファ数不正	T, U
37		EEMCP_RSN_SEGM	117	有効セグメント不正	T, U
38		EEMCP_RSN_BADR	118	編集バッファアドレス不正	T, U
39		EEMCP_RSN_LENF	119	論理端末の閉塞解除不可※2	T, U
40		EEMCP_RSN_LECL	120	送受信失敗による論理端末の閉塞	T, U
41		UOC からのリターンコード	UOC からの リターンコー ド	入力メッセージ編集 UOC または出力メ セージ編集 UOC エ ラーリターン	T, U

(凡例)

T : pro_kind が EEMCP_UOC_PRO_TCP (TCP/IP)

U : pro_kind が EEMCP_UOC_PRO_UDP (UDP)

注※1

eemcpn 定義コマンドの-f オプションの cnrelease オペランドに rst を指定する必要があります。

注※2

前に出力されたメッセージを確認してください。

●data1

MCP が使用する領域です。

●lebuf

MCP が使用する領域です。

●lebuf_size

MCP が使用する領域です。

●r_ipaddr

自システムの IP アドレスが設定されます。

下位 4 バイトに次の形式で設定されます。

dotted-decimal 形式 : 194. 11. 42. 20
↓ ↓ ↓ ↓
16進数形式 (4バイト) : C2 0B 2A 14 …この形式で設定してください。

r_ipaddr[16]

00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]

●o_ipaddr

相手システムの IP アドレスが設定されます。

設定形式は r_ipaddr を参照してください。

●r_port

自システムのポート番号が設定されます。

ポート番号はホストバイトオーダー形式の数値で設定されます。

●o_port

相手システムのポート番号が設定されます。

ポート番号はホストバイトオーダー形式の数値で設定されます。

r_ipaddr, o_ipaddr, r_port, o_port は次の条件をすべて満たした場合に設定されます。

- pro_kind が EEMCP_UOC_PRO_TCP (TCP/IP プロトコル)
- kind が EEMCP_CN_CLOSE (コネクション解放通知) 以外

ee_mcp_actcn - コネクションの確立

機能

コネクションを確立します。

コネクションの確立後、MCP 後処理トランザクションを起動します。

コネクションの確立は非同期で実行されるため、この関数で返されるリターン値が EE_OK の場合でも、コネクションの確立が失敗していることがあります。コネクションの確立結果は、MCP 後処理トランザクションによって通知されます。

この関数で指定したコネクション ID 中に、相手システム (eemcpcn オペランド -o) を省略した eemcpcn オペランドのコネクション ID が含まれていた場合は、指定した全コネクション ID のコネクション確立処理は実行されません。

形式

ANSI C, C++の形式

```
#include <eemcp.h>
int ee_mcp_actcn(EEMCPCNINF *info, int flags)
```

UAP で値を設定する引数

●info

コネクションを確立するコネクション ID を設定した EEMCPCNINF 構造体ポインタのアドレスを設定します。

```
typedef struct ee_mcp_cninf {
    int      con_num;
    char    con_id1[9];
    char    con_id2[9];
    char    con_id3[9];
    char    con_id4[9];
    char    con_id5[9];
    char    con_id6[9];
    char    con_id7[9];
    char    con_id8[9];
}EEMCPCNINF;
```

con_num にはコネクションを確立するコネクション ID 数を 1~8 の範囲で設定します。

con_id1~con_id8 にはコネクションを確立するコネクション ID を設定します。con_id1~con_id8 に設定できるのは、eemcpcn 定義コマンドの-y オプションの mode オペランドに client を指定したコネクションだけです。コネクション ID の長さは最大 8 バイトです。コネクション ID の後ろは NULL 文字を付けてください。

コネクション ID は、*を使って一括指定できます。一括指定は一つだけ指定できます。一括指定する場合は、con_num に 1 を設定し con_id1 に「*」または「先行文字列*」の形式で設定してください。

「*」を設定した場合はすべてのコネクションを確立します。「先行文字列*」を設定した場合は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。

●flags

EENOFLAGS を設定します。

リターン値

リターン値	意味
EE_OK	正常に終了しました。
EECOMER_CNDUOC	UOC から発行しているため、この機能は使用できません。
EECOMER_ENVIRON	TP1/EE の環境下にありません。
EEMCPER_CON_NAME	指定したコネクション ID の中に不正なコネクション ID があります。
EEMCPER_CONDITION	発行条件が不正です。
EEMCPER_INFO	info に設定した値が間違っています。
EEMCPER_MEMORY_SHORT	メモリ不足が発生しました。
EEMCPER_NUM	con_num に設定した値が間違っています。
EEMCPER_OPTION	flags に設定した値が間違っています。
EEMCPER_PART	一部のコネクションの確立に失敗しました。
EEMCPER_SYSERR	システムエラーが発生しました。
EEMCPER_ALREADY	指定したコネクションはすでに使用中です。

注意事項

コネクションの確立は非同期で行われます。したがって、この関数でのリターン値が EE_OK の場合でも、コネクションの確立に失敗していることがあります。コネクションの確立結果は MCP 後処理トランザクションで通知されます。

ee_mcp_actcn_to - 相手システムを指定したコネクションの確立

機能

相手システムを指定してコネクションを確立します。

コネクション確立後、MCP 後処理トランザクションを起動します。

eemcpcn 定義の-o オプション指定値（相手システム情報）を無視します。

コネクションの確立は非同期で行われます。そのため、この関数で返されるリターン値が EE_OK であった場合でも、コネクションの確立が失敗しているときがあります。コネクションの確立結果は MCP 後処理トランザクションによって通知されます。

形式

ANSI C, C++の形式

```
#include <eemcp.h>
int ee_mcp_actcn_to(EEMCPCNINFTO *info, int flags)
```

UAP で値を設定する引数

●info

コネクションの確立を行う相手システムを設定した EEMCPCNINFTO 構造体ポインタのアドレスを設定します。

```
typedef struct ee_mcp_cninf_to {
    char        con_id[9];
    char        resv01[7];
    unsigned char o_ipaddr[16];
    char        o_host[256];
    unsigned short o_port;
    char        resv02[6];
}EEMCPCNINFTO;
```

●con_id

コネクションを確立するコネクション ID を設定します。

コネクション ID の長さは最大 8 バイトです。後ろに NULL 文字を付けてください。

8 バイトに満たないコネクション ID を設定する場合は、コネクション ID の後ろに NULL 文字を付けてください。

●resv01

領域を NULL 文字で埋めます。

●o_ipaddr

相手システムの IP アドレスを設定します。

o_host を設定した場合は o_ipaddr の設定は不要です。o_host, o_ipaddr 共に設定した場合は o_ipaddr 設定値が有効になります。

IP アドレスを設定する場合は、xxx.xxx.xxx.xxx の形式で設定します。xxx は、0～255（10 進数）で設定してください。ただし、0.0.0.0, 255.255.255.255 の設定は不可とします。

IP アドレスは o_ipaddr の下位 4 バイトに次の形式で設定します。

dotted-decimal形式 : 194. 11. 42. 20
 ↓ ↓ ↓ ↓
 C2 0B 2A 14 变換
 16進数形式 (4バイト) : C2 0B 2A 14 …この形式で設定してください。

●o_ipaddr[16]

00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]

●o_host

相手システムのホスト名を設定します。

o_ipaddr を設定した場合は o_host の設定は不要です。o_host, o_ipaddr 共に設定した場合は o_ipaddr 設定値が有効になります。

ホスト名の長さは最大 255 文字です。後ろに NULL 文字を付けてください。255 文字に満たないホスト名を設定する場合は、ホスト名の後ろに NULL 文字を付けてください。

ホスト名は、/etc/hosts ファイル、DNS などで、IP アドレスとマッピングできるように指定する必要があります。

●o_port

相手システムのポート番号を設定します。

ポート番号はホストバイトオーダー形式の数値で設定してください。数値は、1～65535 の符号なし整数で設定してください。

●resv02

領域を NULL 文字で埋めます。

●flags

EENOFLAGS を設定します。

リターン値

リターン値	意味
EE_OK	正常に受け付けました。
EECOMER_ENVIRON	TP1/EE 環境下にありません。
EECOMER_CNDUOC	UOC から発行しているため、この機能は使用できません。
EEMCPER_CONDITION	発行条件が不正です。
EEMCPER_CON_NAME	指定したコネクション ID が不正です。
EEMCPER_ADDRESS	指定した IP アドレスまたはホスト名が不正です。
EEMCPER_PORT	指定したポート番号が不正です。

リターン値	意味
EEMCPER_OPTION	flags 設定した値が間違っています。
EEMCPER_MEMORY_SHORT	メモリ不足が発生しました。
EEMCPER_SYSERR	システムエラーが発生しました。
EEMCPER_INFO	info に設定した値が間違っています。
EEMCPER_ALREADY	指定したコネクションはすでに使用中です。

ee_mcp_actle - 論理端末の閉塞解除

機能

論理端末を閉塞解除します。

論理端末の閉塞解除後、MCP 後処理トランザクションを起動します。

形式

ANSI C, C++の形式

```
#include <eemcp.h>
int ee_mcp_actle(EEMCPLIEINF *info, int flags)
```

UAP で値を設定する引数

●info

閉塞解除する論理端末名を設定した EEMCPLIEINF 構造体ポインタのアドレスを設定します。

```
typedef struct ee_mcp_leinf {
    int          le_num;
    char         le_name1[9];
    char         le_name2[9];
    char         le_name3[9];
    char         le_name4[9];
    char         le_name5[9];
    char         le_name6[9];
    char         le_name7[9];
    char         le_name8[9];
} EEMCPLIEINF;
```

le_num には閉塞解除する論理端末数を 1~8 の範囲で設定します。

le_name1~le_name8 には閉塞解除する論理端末名を設定します。論理端末名の長さは最大 8 バイトです。論理端末名の後ろは NULL 文字を付けてください。

論理端末名は、*を使って一括指定できます。一括指定は一つだけ指定できます。一括指定する場合は、le_num に 1 を設定し le_name1 に「*」または「先行文字列*」の形式で設定してください。

「*」の場合はすべての論理端末を閉塞解除します。「先行文字列*」の場合は先行文字列で始まるすべての論理端末を閉塞解除します。

●flags

EENOFLAGS を設定します。

リターン値

リターン値	意味
EE_OK	正常に終了しました。
EECOMER_CNDUOC	UOC から発行しているため、この機能は使用できません。
EECOMER_ENVIRON	TP1/EE の環境下にありません。
EEMCPER_CONDITION	発行条件が不正です。
EEMCPER_INFO	info に設定した値が間違っています。
EEMCPER_LE_NAME	指定した論理端末名の中に不正な論理端末名があります。
EEMCPER_MEMORY_SHORT	メモリ不足が発生しました。
EEMCPER_NUM	le_num に設定した値が間違っています。
EEMCPER_OPTION	flags に設定した値が間違っています。
EEMCPER_PART	一部の論理端末の閉塞解除に失敗しました。
EEMCPER_SYSERR	システムエラーが発生しました。

注意事項

- 指定した論理端末名が UDP 端末の場合は、該当ポートをオープンします。
- 論理端末の閉塞解除は非同期で行われます。したがって、この関数でのリターン値が EE_OK であった場合でも、論理端末の閉塞解除に失敗している場合があります。論理端末の閉塞解除の結果は MCP 後処理トランザクションで通知されます。

ee_mcp_cninfo - コネクション情報の取得

機能

コネクション情報を取得します。

形式

ANSI C, C++の形式

```
#include <eemcp.h>
int ee_mcp_cninfo(EELONG action, char *idname,
                    EEMCPCONIDINF *info, int flags)
```

UAP で値を設定する引数

●action

情報を取得するコネクションの指定方法を次の形式で設定します。

```
{EEMCPLE | EEMCPCN}
```

EEMCPLE

情報を取得するコネクションを論理端末名称で指定するときに設定します。

EEMCPCN

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

●idname

情報を取得するコネクションの論理端末名称、またはコネクション ID を設定します。論理端末名称、またはコネクション ID は 8 バイト以内で設定し、文字列の最後に NULL 文字を付けます。

●info

コネクション状態を格納する領域 ee_mcp_conid_inf を設定します。

●flags

EENOFLAGS を設定します。

MCP から値が返される引数

●info

この関数の対象となったコネクションの情報が、構造体 ee_mcp_conid_inf で返されます。ただし、コネクション情報が設定されるのはリターン値が EE_OK の場合だけです。

構造体の形式を次に示します。

```
typedef struct ee_mcp_conid_inf {
    char          con_id[9];
    char          resv01[1];
    unsigned short id_len;
    char          pname[4];
    unsigned char r_ipaddr[16];
    unsigned char o_ipaddr[16];
```

```

    unsigned short r_port;
    unsigned short o_port;
    char resv02[4];
}EEMCPCONIDINF;

```

●con_id

要求したコネクションのコネクション ID が設定されます。

●resv01

領域を NULL 文字で埋めます。

●id_len

要求したコネクションのコネクション ID の文字列長が設定されます。

●pname

要求したコネクションのプロトコル種別が文字列で設定されます。

"TCP"

TCP/IP プロトコル

●r_ipaddr

相手システムの IP アドレスを設定します。

o_host を設定した場合は o_ipaddr の設定は不要です。o_host, o_ipaddr 共に設定した場合は o_ipaddr 設定値が有効になります。

IP アドレスを設定する場合は、xxx.xxx.xxx.xxx の形式で設定します。xxx は、0～255（10 進数）で設定してください。ただし、0.0.0.0, 255.255.255.255 の設定は不可とします。

IP アドレスは o_ipaddr の下位 4 バイトに次の形式で設定します。

dotted-decimal 形式 : 194. 11. 42. 20
 ↓ ↓ ↓ ↓
 C2 0B 2A 14
 16進数形式 (4 バイト) : C2 0B 2A 14 …この形式で設定してください。

o_ipaddr[16]

00	00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	

●o_ipaddr

相手システムの IP アドレスを設定します。

IP アドレスの設定形式に関しては r_ipaddr を参照してください。

●r_port

相手システムのポート番号を設定します。

ポート番号はホストバイトオーダー形式の数値で設定してください。数値は、1～65535 の符号なし整数で設定してください。

●o_port

相手システムのポート番号を設定します。

ポート番号の設定形式に関しては r_port を参照してください。

●resv02

領域を NULL 文字で埋めます。

リターン値

リターン値	意味
EE_OK	正常に受け付けました。
EECOMER_ENVIRON	TP1/EE 環境下にありません。
EECOMER_CNDUOC	UOC から発行しているため、この機能は使用できません。
EEMCPER_ACTION	action に設定した値が間違っています。
EEMCPER_IDNAME	idname に設定した値が間違っています。
EEMCPER_OPTION	flags に設定した値が間違っています。
EEMCPER_ACT_BEGIN	指定したコネクションが確立処理中状態です。
EEMCPER_DCT	指定したコネクションが未確立状態です。
EEMCPER_DCT_BEGIN	指定したコネクションが解放処理中状態です。
EEMCPER_INFO	info に設定した値が間違っています。

ee_mcp_dctcn - コネクションの解放

機能

コネクションを解放します。

コネクションの解放後、MCP 後処理トランザクションを起動します。

形式

ANSI C, C++の形式

```
#include <eemcp.h>
int ee_mcp_dctcn(EEMCPCNINF *info, char form, int flags)
```

UAP で値を設定する引数

●info

コネクションを解放するコネクション ID を設定した EEMCPCNINF 構造体ポインタのアドレスを設定します。

```
typedef struct ee_mcp_cninf {
    int      con_num;
    char    con_id1[9];
    char    con_id2[9];
    char    con_id3[9];
    char    con_id4[9];
    char    con_id5[9];
    char    con_id6[9];
    char    con_id7[9];
    char    con_id8[9];
}EEMCPCNINF;
```

con_num にはコネクションを解放するコネクション ID 数を 1~8 の範囲で設定します。

con_id1~con_id8 にはコネクションを解放するコネクション ID を設定します。コネクション ID の長さは最大 8 バイトです。コネクション ID の後ろは NULL 文字を付けてください。

コネクション ID は、*を使って一括指定できます。一括指定は一つだけ指定できます。一括指定する場合は、con_num に 1 を設定し con_id1 に「*」または「先行文字列*」の形式で設定してください。

「*」の場合はすべてのコネクションを解放します。「先行文字列*」の場合は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。

●form

コネクションの解放形態を設定します。この設定値は、eemcpcn 定義コマンドの-f オプションの cnrelease オペランドに rst を指定した場合に有効になります。fin を指定した場合は、form の設定値に関係なく FIN パケットを送信してコネクションを解放します。

EEMCPFRC

RST パケットを送信して、コネクションを強制的に解放する場合に設定します。

EEMCPNRM

FIN パケットを送信して、コネクションを解放する場合に設定します。

●flags

EENOFLAGS を設定します。

リターン値

リターン値	意味
EE_OK	正常に終了しました。
EECOMER_CNDUOC	UOC から発行しているため、この機能は使用できません。
EECOMER_ENVIRON	TP1/EE の環境下にありません。
EEMCPER_CON_NAME	指定したコネクション ID の中に不正なコネクション ID があります。
EEMCPER_CONDITION	発行条件が不正です。
EEMCPER_FORM	form に設定した値が間違っています。
EEMCPER_INFO	info に設定した値が間違っています。
EEMCPER_NUM	con_num に設定した値が間違っています。
EEMCPER_OPTION	flags に設定した値が間違っています。
EEMCPER_PART	一部のコネクションの解放に失敗しました。
EEMCPER_SYSERR	システムエラーが発生しました。

注意事項

コネクションの解放は非同期で行われます。したがって、この関数でのリターン値が EE_OK であった場合でも、コネクションの解放に失敗している場合があります。コネクションの解放結果は MCP 後処理トランザクションで通知されます。

ee_mcp_dctle - 論理端末の閉塞

機能

論理端末を閉塞します。

論理端末の閉塞後、MCP 後処理トランザクションを起動します。

形式

ANSI C, C++の形式

```
#include <eemcp.h>
int ee_mcp_dctle(EEMCPLINE *info, int flags)
```

UAP で値を設定する引数

●info

閉塞する論理端末名を設定した EEMCPELEINF 構造体ポインタのアドレスを設定します。

```
typedef struct ee_mcp_leinf {
    int      le_num;
    char    le_name1[9];
    char    le_name2[9];
    char    le_name3[9];
    char    le_name4[9];
    char    le_name5[9];
    char    le_name6[9];
    char    le_name7[9];
    char    le_name8[9];
}EEMCPELEINF;
```

le_num には閉塞する論理端末数を 1~8 の範囲で設定します。

le_name1~le_name8 には閉塞する論理端末名を設定します。論理端末名の長さは最大 8 バイトです。論理端末名の後ろは NULL 文字を付けてください。

論理端末名は、*を使って一括指定できます。一括指定は一つだけ指定できます。一括指定する場合は、le_num に 1 を設定し le_name1 に「*」または「先行文字列*」の形式で設定してください。

「*」の場合はすべての論理端末を閉塞します。「先行文字列*」の場合は先行文字列で始まるすべての論理端末を閉塞します。

●flags

EENOFLAGS を設定します。

リターン値

リターン値	意味
EE_OK	正常に終了しました。
EECOMER_CNDUOC	UOC から発行しているため、この機能は使用できません。
EECOMER_ENVIRON	TP1/EE の環境下にありません。
EEMCPEL_CONDITION	発行条件が不正です。
EEMCPEL_INFO	info に設定した値が間違っています。
EEMCPEL_NAME	指定した論理端末名の中に不正な論理端末名があります。
EEMCPEL_NUM	le_num に設定した値が間違っています。
EEMCPEL_OPTION	flags に設定した値が間違っています。
EEMCPEL_PART	一部の論理端末の閉塞に失敗しました。

リターン値	意味
EEMCPER_SYSERR	システムエラーが発生しました。

注意事項

- 指定した論理端末名が UDP 端末の場合は、該当ポートをクローズします。
- 論理端末の閉塞は非同期で行われます。したがって、この関数で返されるリターン値が EE_OK であった場合でも、論理端末の閉塞に失敗している場合があります。論理端末の閉塞結果は MCP 後処理トランザクションで通知されます。

ee_mcp_send - 非同期一方送信メッセージの送信

機能

相手システムへ非同期一方送信メッセージを送信します。非同期一方送信メッセージは、一つのセグメントで構成されます。

TP1/FSP を使用する場合、eemcpfunc 定義コマンドの-o オプションの outputmessage オペランドが設定されているとき、または TP1/EE のユーザサービス関連定義の service_attr 定義コマンドが設定されているときは、送信データを UAP 履歴情報のデータ種別「OJ (出力メッセージ)」として取得します。オペランドの関連性については、TP1/FSP の関連ドキュメントを参照してください。

形式

ANSI C, C++の形式

```
#include <eemcp.h>
int ee_mcp_send(int action, int commform, char *termnam,
                 char *resv01, char *senddata, int sdata leng,
                 char *resv02, int opcd)
```

UAP で値を設定する引数

●action

送信する論理メッセージのセグメントを次の形式で指定します。

EEMCPEMI

单一セグメントを示す EEMCPEMI を設定します。

●commform

一方送信を示す EEMCPOUT を設定します。

●termnam

出力先の論理端末名を設定します。論理端末名の長さは最大 8 バイトです。論理端末名の後ろは NULL 文字を付けてください。

●resv01

NULL 文字を設定します。

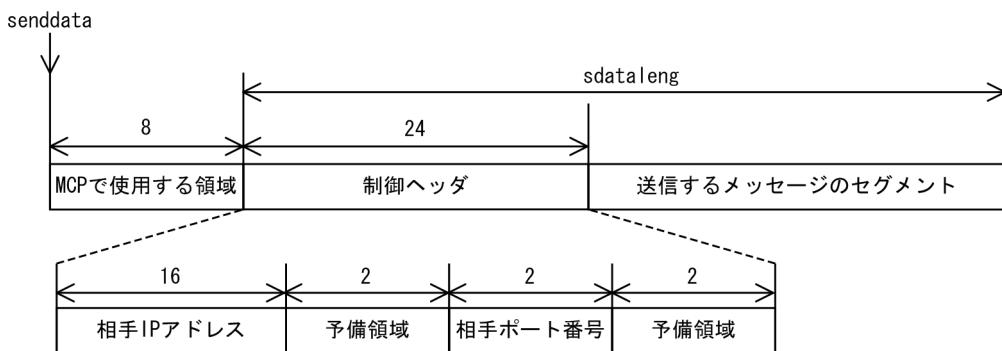
●senddata

送信するセグメントの内容を設定した領域を設定します。送信できる最大長は、sdataleng を参照してください。

TCP/IP の場合



UDP の場合



■制御ヘッダ

制御ヘッダ (EEMCPUDPHD) は、<eemcp.h>に次のように定義されています。

```
typedef struct ee_mcp_udp_header {
    unsigned char r_ipaddr[16];    ... 相手IPアドレス
    unsigned char rsv1[2];        ... 予備
    unsigned short r_portno;     ... 相手ポート番号
    unsigned char rsv2[4];        ... 予備
} EEMCPUDPHD;
```

■相手IPアドレス

相手IPアドレスは`r_ipaddr`の下位4バイトに次に示す形式で設定します。

dotted-decimal形式 : 194. 11. 42. 20

↓ ↓ ↓ ↓

変換

16進数形式 (4バイト) : C2 0B 2A 14 ...この形式で設定してください。

`r_ipaddr[16]`

00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]

■相手ポート番号

ポート番号は次の形式で設定します。

10進数形式 :

1024
↓
変換

16進数形式 (2バイト) : 0x0400 ...この形式 (ビッグエンディアン) で設定してください。

●`sdata leng`

送信するセグメントの長さを設定します。設定できる最大値は、メッセージの最大サイズを拡張しているかどうか (eemcpfunc 定義コマンドの-m オプションの`msglen` オペランド指定値) によって、次の表に示すとおり異なります。

表 5-7 メッセージの最大サイズ拡張有無による送信メッセージ長の違い

メッセージの最大サイズ拡張有無	送信メッセージ長に設定できる最大値 (単位: バイト)	
	TCP/IP プロトコル使用時	UDP プロトコル使用時
なし (<code>msglen</code> オペランドに <code>normal</code> を指定)	32000*	32000*
あり (<code>msglen</code> オペランドに <code>extend</code> を指定)	8388608	65531*

注※

TP1/EE サービス定義の send_message_buf_size オペランドで指定した送信バッファサイズの方が小さい場合は、 send_message_buf_size オペランドの指定値が設定されます。

●resv02

NULL 文字を設定します。

●opcd

一方送信メッセージの送信属性を指定します。

EENOFLAGS 以外を複数指定する場合は、論理和で指定します。

EENOFLAGS

メッセージを非永続属性で、かつ UDP 送信時の送信先を senddata の制御ヘッダで指定する場合に設定します。

EEMCP_PERMANENCE

永続メッセージとする場合に設定します。

MCP をクラスタ構成に組み込んだ場合 (XTC サービス定義の cluster_mode オペランドに Y を指定) にだけ有効となります。

EEMCP_DEFADDR

UDP 送信時の送信先を eemcple 定義コマンドの-o オプションで指定する場合に設定します。

なお、この値を指定した場合は、制御ヘッダの内容に関係なく、MCP 構成定義で指定した送信先が有効になります。

リターン値

リターン値	意味
EE_OK	正常に終了しました。
EECOMER_CNDUOC	UOC から発行しているため、この機能は使用できません。
EECOMER_ENVIRON	TP1/EE の環境下にありません。
EEMCPER_ACTION	action に設定した値が間違っています。
EEMCPER_ARGUMENT	senddata に設定した値が間違っています。
EEMCPER_COMMFORM	commform に設定した値が間違っています。
EEMCPER_CONDITION	発行条件が不正です。 ee_trn_rollback_mark 関数が発行済みです。
EEMCPER_LE_NAME	termnam に設定した論理端末名が間違っています。
EEMCPER_MEMORY_SHORT	メモリ不足が発生しました。
EEMCPER_MESSAGE_TOO_BIG	sdataleng に、送信できる最大サイズを超える値を設定しています。

リターン値	意味
EEMCPER_MESSAGE_TOO_SHORT	sdata leng に 0 (UDP の場合は 24) 以下の値を設定しています。
EEMCPER_OPTION	opcd に設定した値が間違っています。
EEMCPER_TERM_HOLD	termnam に設定した論理端末名は閉塞中です。

注意事項

- この関数発行後に、ee_mcp_sendsync 関数を発行した場合、ee_mcp_sendsync 関数で送信したメッセージが先に送信されます。
- 送信要求したトランザクションがロールバックした場合、送信メッセージは破棄されます。
- ERRTRNS 上でこの関数を発行して、送信障害などのエラーが発生した場合、ERRTRNS は起動されないで、送信メッセージは破棄されます。
- この関数を発行したトランザクションでは、トランザクションの最適化を行いません。
- 非永続メッセージのあとに、永続メッセージを送信した場合、永続メッセージが先に送信されます。
- この関数は、トランザクションのコミット時に実行されます。したがって、トランザクションの同期点を取得する処理でエラーが起こった場合、この関数の処理は無効となります。
- TP1/FSP を使用する場合、次のリターン値のときは、「OJ (出力メッセージ)」は取得されません。
EECOMER_ENVIRON, EECOMER_CNDUOC, EEMCPER_MESSAGE_TOO_BIG,
EEMCPER_LE_NAME, EEMCPER_ACTION, EEMCPER_OPTION,
EEMCPER_COMMFORM, EEMCPER_MESSAGE_TOO_SHORT, EEMCPER_ARGUMENT

ee_mcp_sendrecv - 同期型メッセージの送受信

機能

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

TP1/FSP を使用する場合、アプリケーション定義の eemcpfunc の-o オプションの outputmessage が設定されている場合、または、ユーザサービス関連定義 service_attr の-m オプションが設定されている場合に、送信データを UAP 履歴情報のデータ種別「OJ (出力メッセージ)」として取得します。オペランドの関連性については、TP1/FSP の関連ドキュメントを参照してください。

形式

ANSI C, C++の形式

P-8264-9311 の場合

```
#include <eemcp.h>
int ee_mcp_sendrecv(int action, int commform, char *termnam, char *resv01,
```

```
char *senddata, int sdata leng, char *recvdata,  
int *rdata leng, int inbufleng, int *time,  
int watchtime, int opcd)
```

P-8864-9311 の場合

```
#include <eemcp.h>  
int ee_mcp_sendrecv(int action, int commform, char *termnam, char *resv01,  
                     char *senddata, int sdata leng, char *recvdata,  
                     int *rdata leng, int inbufleng, unsigned int *time,  
                     int watchtime, int opcd)
```

UAP で値を設定する引数

●action

送信する論理メッセージのセグメントを次の形式で指定します。

EEMCPEMI

单一セグメントを示す EEMCPEMI を設定します。

●commform

同期型メッセージ送受信を示す EEMCPIO を設定します。

●termnam

出力先の論理端末名を設定します。論理端末名の長さは最大 8 バイトです。論理端末名の後ろは NULL 文字を付けてください。

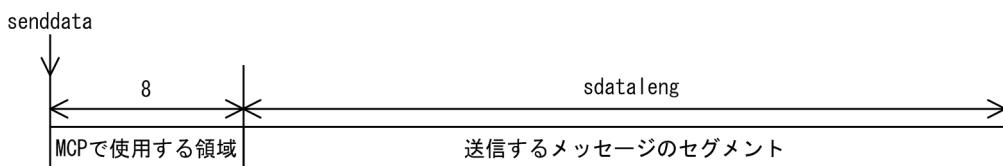
●resv01

NULL 文字を設定します。

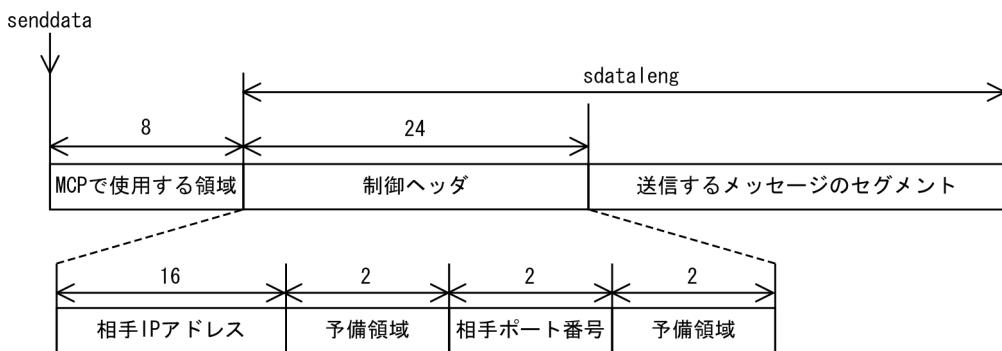
●senddata

送信するセグメントの内容を設定した領域を設定します。送信できる最大長は、sdata leng を参照してください。

TCP/IP の場合



UDP の場合



■制御ヘッダ

制御ヘッダ (EEMCPUDPHD) は、<eemcp.h>に次のように定義されています。

```
typedef struct ee_mcp_udp_header {
    unsigned char r_ipaddr[16];    … 相手IPアドレス
    unsigned char rsv1[2];        … 予備
    unsigned short r_portno;     … 相手ポート番号
    unsigned char rsv2[4];        … 予備
} EEMCPUDPHD;
```

■相手IPアドレス

相手IPアドレスはr_ipaddrの下位4バイトに次に示す形式で設定します。

dotted-decimal形式 : 194. 11. 42. 20

↓ ↓ ↓ ↓

変換

16進数形式 (4バイト) : C2 0B 2A 14 …この形式で設定してください。

r_ipaddr[16]

00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]

■相手ポート番号

ポート番号は次の形式で設定します。

10進数形式 :

1024
↓
変換

16進数形式 (2バイト) : 0x0400 …この形式 (ビッグエンディアン) で設定してください。

●sdata leng

送信するセグメントの長さを設定します。設定できる最大値は、メッセージの最大サイズを拡張しているかどうか (eemcpfunc 定義コマンドの-m オプションの msglen オペランド指定値) によって、次の表に示すとおり異なります。

表 5-8 メッセージの最大サイズ拡張有無による送信メッセージ長の違い

メッセージの最大サイズ拡張有無	送信メッセージ長に設定できる最大値 (単位: バイト)	
	TCP/IP プロトコル使用時	UDP プロトコル使用時
なし (msglen オペランドに normal を指定)	32000*	32000*
あり (msglen オペランドに extend を指定)	8388608	65531*

注※

TP1/EE サービス定義の send_message_buf_size オペランドで指定した送信バッファサイズの方が小さい場合は、 send_message_buf_size オペランドの指定値が設定されます。

●recvdata

セグメントを受信する領域を設定します。この領域は関数発行元で用意してください。

データ形式については、次の図を参照してください。

図 5-1 受信のデータ形式



●rdata leng

受信したセグメントの長さを格納する領域を設定します。この領域は関数発行元で用意してください。

受信セグメントの長さは rdata leng で指定します。rdata leng で設定される最大値を次の表に示します。

表 5-9 rdata leng で指定できる最大値

メッセージの最大サイズ拡張有無 (eemcpfunc コマンドの-m オプション msglen オペランドの指定)	受信メッセージ長に設定される最大値 (単位: バイト)
なし (normal を指定)	TP1/EE サービス定義の recv_message_buf_size (受信バッファサイズ) 指定値
あり (extend を指定)	8388608 (8M)

●inbuflen

セグメントを受信する領域の長さを設定します。領域の長さは、図 5-1 に示す inbuflen を参照してください。

領域の長さは、rdata leng + 8 バイト以上の長さを設定してください。

●time

メッセージを受信した時刻を格納する領域を設定します。この領域は関数発行元で用意してください。

●watchtime

ee_mcp_sendrecv を呼び出してから終了するまでの最大時間 (1~65535 単位: 秒) を設定します。0 を設定した場合、MCP 共通定義 (eemcpn -b sndrcvtime) で指定した同期送受信監視時間が仮定されます。負の値を設定した場合は、無限に待ち合わせをします。

EENOFLAGS

指定した論理端末がすでに送信処理中の場合、送信可能となるまで待ち合わせるときに設定します。待ち合わせている間の時間は、watchtime に設定した時間に含まれます。

EEMCP_NOWAIT

指定した論理端末がすでに送信処理中の場合、エラーリターンするときに設定します。

TP1/EE から値が返される引数

リターン値が EE_OK または EEMCPER_OVERFLOW の場合だけ、次に示す recvdata, rdata leng, time の領域は参照可能です。これ以外のリターン値の場合は参照しないでください。

●recvdata

受信したメッセージの先頭アドレスが返されます。

●rdata leng

受信したメッセージ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
EE_OK	正常に終了しました。
EECOMER_ENVIRON	TP1/EE 環境下にありません。
EECOMER_CNDUOC	UOC から発行しているため、この機能は使用できません。
EEMCPER_CONDITION	発行条件が不正です。
EEMCPER_MESSAGE_TOO_BIG	sdata leng に送信できるサイズを超える値を設定しています。
EEMCPER_MEMORY_SHORT	メモリ不足が発生しました。 • 送信バッファ数または受信バッファ数が不足しました。メッセージサイズが OBF サイズまたは IBF サイズを超過しました。
EEMCPER_LE_NAME	termnam に設定した論理端末名が間違っています。
EEMCPER_ACTION	action に設定した値が間違っています。
EEMCPER_OPTION	opcd に設定した値が間違っています。
EEMCPER_COMMFORM	commform に設定した値が間違っています。
EEMCPER_MESSAGE_TOO_SHORT	sdata leng に 0 以下の値を設定しています。

リターン値	意味
EEMCPER_TERM_HOLD	termnam に設定した論理端末名は閉塞中です。または、メッセージ送信中にコネクションの解放を検知しました。
EEMCPER_TIME	watchtime に設定した時間が経過しました。
EEMCPER_SENDER	システムに対する送信時に障害が発生しました。
EEMCPER_UOC_ERR	UOC がエラーリターンしました。
EEMCPER_UOC_INVALID	UOC で障害が発生しました。
EEMCPER_ARGUMENT	senddata, recvdata, rdataleng, time, watchtime に設定した値が間違っています。
EEMCPER_OVERFLOW	inbuflen の指定値を超えるセグメントを受信しました。inbuflen の指定値を超えた部分は切り捨てされました。
EEMCPER_INBUFLENG	inbuflen の指定値が不足しています。9 バイト以上の領域を確保してください。
EEMCPER_SENDER_RECV	メッセージ受信が仕掛けかり中です。
EEMCPER_TERM_IN_USE	出力先の論理端末は、ほかの UAP で仕掛けかり中です。
EEMCPER_SENDTIME	メッセージ送信完了監視タイマ (MCP 構成定義 eemcpcn の-b オプション sndcmptim オペランドに指定) のタイムアウトが発生しました。
EEMCPER_RECVVER	受信待ちでコネクション障害が発生しました。

注意事項

- ee_mcp_send 関数発行後、同一 UAP 内で ee_mcp_sendrecv 関数を発行した場合、ee_mcp_sendrecv 関数のメッセージが先に送信されます。
- リターン値が EEMCPER_OVERFLOW の場合、rdataleng には inbuflen の領域に格納できた受信データのサイズが設定されます。
- TP1/FSP を使用する場合、次のリターン値のときは、OJ は取得されません。
EECOMER_ENVIRON, EECOMER_CNDUOC, EEMCPER_CONDITION, EEMCPER_MESSAGE_TOO_BIG, EEMCPER_LE_NAME, EEMCPER_ACTION, EEMCPER_OPTION, EEMCPER_COMMFORM, EEMCPER_MESSAGE_TOO_SHORT, EEMCPER_ARGUMENT, EEMCPER_INBUFLENG

ee_mcp_sendsync - 同期一方送信メッセージの送信

機能

相手システムへ同期一方送信メッセージを送信します。同期一方送信メッセージは、一つのセグメントで構成されます。

TP1/FSP を使用する場合、eemcpfunc 定義コマンドの-o オプションの outputmessage オペランドが設定されているとき、または TP1/EE のユーザサービス関連定義の service_attr 定義コマンドが設定されているときは、送信データを UAP 履歴情報のデータ種別「OJ (出力メッセージ)」として取得します。オペランドの関連性については、TP1/FSP の関連ドキュメントを参照してください。

形式

ANSI C、C++の形式

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

UAP で値を設定する引数

●action

送信する論理メッセージのセグメントを次の形式で指定します。

EEMCPEMI

单一セグメントを示す EEMCPEMI を設定します。

●commform

EENOFLAGS を設定します。

●termnam

出力先の論理端末名を設定します。論理端末名の長さは最大 8 バイトです。論理端末名の後ろは NULL 文字を付けてください。

●resv01

NULL 文字を設定します。

●senddata

送信するセグメントの内容を設定した領域を設定します。データ形式については、「[5.2 ee_mcp_send - 非同期一方送信メッセージの送信](#)」の「[UAP で値を設定する引数](#)」を参照してください。

●sdataleng

送信するセグメントの長さを設定します。設定できる最大値については、「[5.2 ee_mcp_send - 非同期一方送信メッセージの送信](#)」の「[UAP で値を設定する引数](#)」を参照してください。

●resv02

NULL 文字を設定します。

●watchtime

この関数を呼び出してから終了するまでの最大時間（単位：秒）を設定します。ここでの最大時間とは、ソケット空き待ち時間および送信開始から自システムの TCP/IP の送信バッファへの書き込み完了までの時間の合計をいいます。65535 を最大値として、値を設定できます。

0 を設定した場合、eemcpn 定義コマンドの -b オプションの sndsynctim オペランドで指定した、同期一方送信時のメッセージ送信完了待ち時間が仮定されます。負の値を設定した場合は、時間を監視しません。無限に送信完了を待ちます。UDP プロトコルを使用した通信時は無視されます。

●opcd

同期型メッセージの送信属性を指定します。

EENOFLAGS

UDP 送信時の送信先を senddata の制御ヘッダで指定する場合に設定します。

EEMCP_DEFADDR

UDP 送信時の送信先を eemcple 定義コマンドの -o オプションで指定する場合に設定します。

なお、この値を指定した場合は、制御ヘッダの内容に関係なく、MCP 構成定義で指定した送信先が有効になります。

リターン値

リターン値	意味
EE_OK	正常に終了しました。
EECOMER_CNDUOC	UOC から発行しているため、この機能は使用できません。
EECOMER_ENVIRON	TP1/EE の環境下にありません。
EEMCPER_ACTION	action に設定した値が間違っています。
EEMCPER_ARGUMENT	senddata、および watchtime に設定した値が間違っています。
EEMCPER_COMMFORM	commform に設定した値が間違っています。
EEMCPER_CONDITION	発行条件が不正です。
EEMCPER_LE_NAME	termnam に設定した論理端末名が間違っています。
EEMCPER_MEMORY_SHORT	メモリ不足が発生しました。
EEMCPER_MESSAGE_TOO_BIG	sdata leng に、送信できる最大サイズを超える値を設定しています。
EEMCPER_MESSAGE_TOO_SHORT	sdata leng に 0 (UDP の場合は 24) 以下の値を設定しています。
EEMCPER_OPTION	opcd に設定した値が間違っています。
EEMCPER_SENDER	システムに対する送信時に障害が発生しました。
EEMCPER_TERM_HOLD	termnam に設定した論理端末名は閉塞中です。
EEMCPER_TIME	watchtime に設定した時間が経過しました。

リターン値	意味
EEMCPER_UOC_ERR	出力メッセージ編集 UOC がエラーリターンしました。
EEMCPER_UOC_INVALID	出力メッセージ編集 UOC で障害が発生しました。

注意事項

- ee_mcp_send 関数発行後に、この関数を発行した場合、この関数で送信したメッセージが先に送信されます。
- TP1/FSP を使用する場合、次のリターン値のときは、「OJ (出力メッセージ)」は取得されません。
EECOMER_ENVIRON, EECOMER_CNDUOC, EEMCPER_MESSAGE_TOO_BIG,
EEMCPER_LE_NAME, EEMCPER_ACTION, EEMCPER_OPTION,
EEMCPER_COMMFORM, EEMCPER_MESSAGE_TOO_SHORT, EEMCPER_ARGUMENT

ee_mcp_utrace_put - ユーザトレースの取得

機能

ユーザ固有のトレースデータを MCP トレースに出力します。

出力したトレースデータは、eemcped コマンドまたは eemcpdump コマンドで編集できます。

形式

ANSI C, C++の形式

```
#include <eemcp.h>
int ee_mcp_utrace_put(unsigned short event_id,
                      unsigned short datalen,
                      char *buffaddr,
                      EELONG flags)
```

UAP で値を設定する領域

●event_id

トレースを取得するイベントのイベント ID を設定します。0x0700～0x0740 の範囲のイベント ID を設定してください。

●datalen

トレースのデータ長を設定します（単位：バイト）。4～32000 の範囲のデータ長を設定してください。

●buffaddr

トレースデータが格納されているバッファの先頭ポインタを設定します。

●flags

EENOFLAGS を設定します。

リターン値

リターン値	意味
EE_OK	正常に終了しました。
EECOMER_CNDUOC	ee_mcp_utrace_put 関数を使用できない UOC から発行しているため、この機能は使用できません。
EECOMER_ENVIRON	TP1/EE の環境下にありません。
EEMCPER_ARGUMENT	データ長以外の引数に設定した値が間違っています。
EEMCPER_CONDITION	発行条件が不正です。
EEMCPER_MESSAGE_TOO_BIG	データ長に 32000 を超える値を設定しています。
EEMCPER_MESSAGE_TOO_SHORT	データ長に 4 未満の値を設定しています。
EEMCPER_STOP	トラブルシュート機能が縮退中です。

ee_mcp_utrace_put_long - ユーザトレースの取得（データ長拡張用）

機能

データ長拡張時に、ユーザ固有のトレースデータを MCP トレースに出力します。

出力したトレースデータは、eemcped コマンドまたは eemcpdump コマンドで編集できます。

形式

ANSI C, C++の形式

```
#include <eemcp.h>
int ee_mcp_utrace_put_long(unsigned short event_id,
                           EEULONG datalen,
                           char *buffaddr,
                           EELONG flags)
```

UAP で値を設定する領域

●event_id

トレースを取得するイベントのイベント ID を設定します。0x0700~0x0740 の範囲のイベント ID を設定してください。

●datalen

トレースのデータ長を設定します（単位：バイト）。4～8388608 の範囲のデータ長を設定してください。

●buffaddr

トレースデータが格納されているバッファの先頭ポインタを設定します。

●flags

EENOFLAGS を設定します。

リターン値

リターン値	意味
EE_OK	正常に終了しました。
EECOMER_CNDUOC	ee_mcp_utrace_put_long 関数を使用できない UOC から発行しているため、この機能は使用できません。
EECOMER_ENVIRON	TP1/EE の環境下にありません。
EEMCPER_ARGUMENT	データ長以外の引数に設定した値が間違っています。
EEMCPER_CONDITION	発行条件が不正です。
EEMCPER_MESSAGE_TOO_BIG	データ長に 8388608 を超える値を設定しています。
EEMCPER_MESSAGE_TOO_SHORT	データ長に 4 未満の値を設定しています。
EEMCPER_STOP	トラブルシュート機能が縮退中です。

注意事項

トレースのデータ長が MCP トレースファイル出力用最大バッファサイズ（eemcptrc 定義コマンドの-b オプションの buffersz オペランドで指定）よりも大きい場合、データの末尾を切り捨てて出力します。切り捨てが起きた場合に取得されるデータのサイズ（単位：バイト）は、次に示す計算式で求められます。

$$((\text{MCP トレースファイル出力用最大バッファサイズ} \times 1024) - 140)$$

5.3 ライブライリ関数 (COBOL 言語)

COBOL 言語のライブライリ関数について説明します。

COBOL-UAP 作成用プログラムの説明形式

COBOL-UAP 作成用プログラムの説明形式については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

サービスプログラム (SPP) の作成

SPP はサービスを提供する UAP です。ここでは、MCP での SPP の文法について説明します。

形式

```
PROGRAM-ID. プログラム名.  
LINKAGE SECTION.  
01—意名1.  
 02 データ名A PIC X(n).  
01—意名2.  
 02 データ名B PIC S9(9) COMP.  
01—意名3.  
 02 データ名C PIC X(n).  
01—意名4.  
 02 データ名D PIC S9(9) COMP.  
01—意名5.  
 02 データ名E PIC 9(9) COMP.      /*情報長  
 02 データ名F PIC 9(9) COMP.      /*サービスグループ名長  
 02 データ名G PIC X(32).          /*サービスグループ名  
 02 データ名H PIC 9(9) COMP.      /*サービス名長  
 02 データ名I PIC X(32).          /*サービス名  
 02 データ名J PIC X(2).          /*トランザクション種別  
 02 データ名K PIC X(1).          /*自動閉塞有無  
 02 データ名L PIC X(1).          /*コミット有無  
 02 データ名M PIC 9(9) COMP.      /*処理スレッド通番  
 02 データ名N PIC 9(9) COMP.      /*エラー要因コード  
 02 データ名O PIC 9(9) COMP.      /*ユーザIFAエリアサイズ  
 02 FILLER PIC X(4).            /*予備  
 02 データ名P ADDRESS.          /*ユーザIFAエリアポインタ  
 02 データ名Q PIC 9(9) COMP.      /*ユーザSATエリアサイズ  
 02 FILLER PIC X(4).            /*予備  
 02 データ名R ADDRESS.          /*ユーザSATエリアポインタ  
 02 データ名S PIC X(1).          /*応答要否  
 02 データ名T PIC X(1).          /*メッセージ状態  
 02 データ名U PIC X(1).          /*プロセス起動要因  
 02 データ名V PIC X(1).          /*前回プロセス終了要因  
 02 データ名W PIC X(1).          /*プロセス終了要因  
 02 データ名X PIC X(1).          /*スレッドダウン種別  
 02 データ名Y PIC X(1).          /*異常発生トランザクション種別  
 02 データ名Z PIC X(1).          /*拡張トランザクション種別
```

```

02 データ名AA PIC 9(9) COMP.      /*リソースマネージャ数      */
02 データ名AB PIC S9(9) COMP.      /*リソースマネージャ接続状態  */
02 データ名AC PIC 9(9) COMP.      /*送信サービスグループ名長    */
02 データ名AD PIC X(32).          /*送信サービスグループ名    */
02 データ名AE PIC 9(9) COMP.      /*送信サービス名長          */
02 データ名AF PIC X(32).          /*送信サービス名          */
02 データ名AG PIC 9(9) COMP.      /*異常発生処理スレッド通番  */
02 FILLER PIC X(4).              /*予備                      */
02 データ名AH ADDRESS.           /*拡張トランザクションインタフェース情報 */
02 データ名AI ADDRESS.           /*XTC拡張トランザクションインタフェース情報*/
02 データ名AJ ADDRESS.           /*MCP拡張トランザクションインタフェース情報*/
:
:

```

```
PROCEDURE DIVISION USING 一意名1 一意名2 一意名3 一意名4 一意名5
```

```
:
サービスの処理
:
EXIT PROGRAM.
```

機能

サービスを実行する SPP のサービスプログラムです。サービスプログラムは、上記の形式で任意に作成してください。

サービス名は、サービスプログラムのプログラム ID に対応させます。この対応づけは、UAP の実行環境を設定するときに指定します。UAP の実行環境を設定する手順を次に示します。

1. TP1/EE での設定

ユーザサービス関連定義で指定します。詳細については、マニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。

2. MCP での設定

MCP 後処理トランザクション起動サービス名は、eemcpfunc 定義コマンドで指定します。詳細については、「[4.3 MCP 構成定義の詳細](#)」の「eemcpfunc (アプリケーション定義)」を参照してください。

データ領域の設定

ここでは、データ領域の設定のうち、TP1/EE と異なる部分について説明します。そのほかのデータ領域情報については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

トランザクション種別 (データ名 J) が RL の場合にサービスプログラムに値が渡されるデータ領域を次の表に示します。

表 5-10 トランザクション種別 (データ名 J) が RL の場合のサービスプログラムに値が渡されるデータ領域

項目番	データ領域	参照可否
1	入力パラメタ	×

項目番	データ領域	参照可否
2	入力パラメタ長	×
3	サービスプログラムの応答	×
4	応答の長さ	×
5	トランザクションインターフェース情報	○

(凡例)

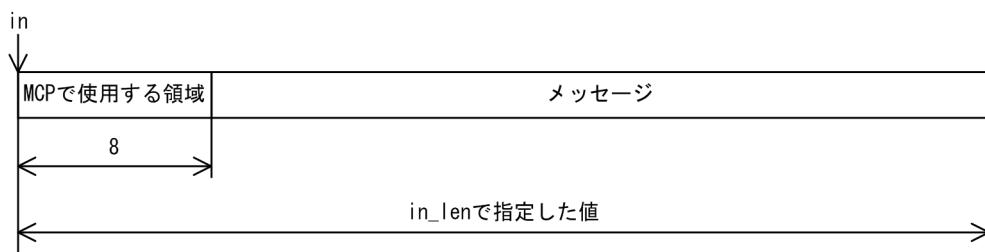
○：参照できます。

×：参照できません。

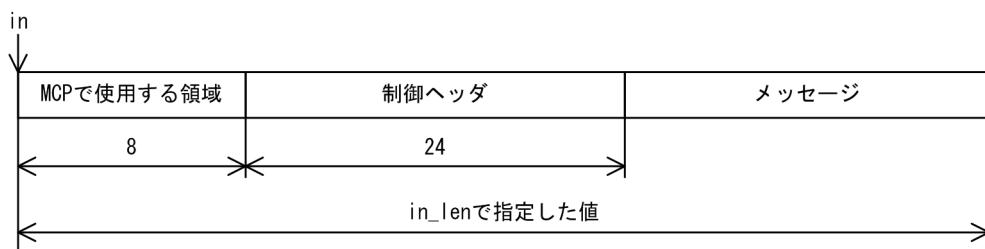
■入力パラメタと入力パラメタ長の関係

入力パラメタ (in) と入力パラメタ長 (in_len) の関係を次の図に示します (単位：バイト)。

TCP/IP の場合



UDP の場合



制御ヘッダについては、「5.2 ee_mcp_send - 非同期一方送信メッセージの送信」の「UAP で値を設定する引数」を参照してください。

なお、MCP ではサービスプログラムの応答 (データ名 C) を使用した応答送信はできません。

クライアント UAP から値が渡されるデータ領域

トランザクション種別が ES の場合に TP1/EE と設定値が異なるデータ領域を次に示します。そのほかのデータ領域については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」およびマニュアル「TP1/EE/Extended Transaction Controller 使用の手引」を参照してください。

●データ名 A

トランザクションと同期して送信する非同期一方送信のメッセージ送信 API (CBLEEMCP('SEND')) の入力パラメタ (データ名 W) に設定した値 (出力メッセージ編集 UOC 指定時は編集後の値※) が渡されます。

注※

メッセージの最大サイズを拡張している場合に、出力メッセージ編集 UOC で不正なサイズを指定したときは、編集後の値を 0 バイトとします。また、入力メッセージ編集 UOC で編集後に送信バッファ不足が発生した場合、編集後の値の末尾を切り捨てます。

●データ名 B

トランザクションと同期して送信する非同期一方送信のメッセージ送信 API (CBLEEMCP('SEND')) の入力パラメタ (データ名 U) に設定した値 (出力メッセージ編集 UOC 指定時は編集後の値※) に 8 を加算した値が渡されます。

注※

メッセージの最大サイズを拡張している場合に、出力メッセージ編集 UOC で不正なサイズを指定したときは、編集後の値を 0 バイトとします。また、入力メッセージ編集 UOC で編集後に送信バッファ不足が発生した場合、編集後の値の末尾を切り捨てます。

●データ名 D

参照できません。

その他

次に示す内容については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

- TP1/EE から値が返されるデータ領域
- UAP で値を設定するデータ領域
- サービスプログラムの処理での注意
- トランザクションとサービスプログラムの関係
- ステータスコード

トランザクションインターフェース情報

ここでは、次に示す内容について説明します。

- MCP 後処理トランザクション
- エラートランザクション (ERRTRNS) ※

- MCP 拡張トランザクションインターフェース情報

注※

TP1/EE と異なる部分についてだけ説明します。

上記以外のトランザクションインターフェース情報の詳細については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

MCP 後処理トランザクション

MCP 後処理トランザクション (RL) は MCP 固有のトランザクション種別です。MCP 後処理トランザクションは、次に示すタイミングで起動します。

- コネクションの確立または解放
- 論理端末の閉塞または閉塞解除

トランザクション種別が RL の場合に参照できるトランザクションインターフェース情報を次の表に示します。各トランザクションインターフェース情報については、マニュアル「TP1/Server Base Enterprise Option プログラム作成の手引」を参照してください。

表 5-11 トランザクション種別が RL の場合のトランザクションインターフェース情報

項目番	トランザクションインターフェース情報	参照可否
1	情報長 (データ名 E)	○
2	サービスグループ名長 (データ名 F)	○
3	サービスグループ名 (データ名 G)	○
4	サービス名長 (データ名 H)	○
5	サービス名 (データ名 I)	○
6	トランザクション種別 (データ名 J)	○
7	自動閉塞有無 (データ名 K)	×
8	コミット有無 (データ名 L)	×
9	処理スレッド通番 (データ名 M)	○
10	エラー要因コード (データ名 N)	×
11	ユーザ IFA エリアサイズ (データ名 O)	○
12	ユーザ IFA エリアポインタ (データ名 P)	○
13	ユーザ SAT エリアサイズ (データ名 Q)	○
14	ユーザ SAT エリアポインタ (データ名 R)	○
15	応答要否 (データ名 S)	×
16	メッセージ状態 (データ名 T)	○

項目番	トランザクションインターフェース情報	参照可否
17	プロセス起動要因 (データ名 U)	×
18	前回プロセス終了要因 (データ名 V)	×
19	プロセス終了要因 (データ名 W)	×
20	スレッドダウン種別 (データ名 X)	×
21	異常発生トランザクション種別 (データ名 Y)	×
22	拡張トランザクション種別 (データ名 Z)	×
23	リソースマネジャ数 (データ名 AA)	○
24	リソースマネジャ接続状態 (データ名 AB)	○
25	送信サービスグループ名長 (データ名 AC)	×
26	送信サービスグループ名 (データ名 AD)	×
27	送信サービス名長 (データ名 AE)	×
28	送信サービス名 (データ名 AF)	×
29	異常発生処理スレッド通番 (データ名 AG)	×
30	拡張トランザクションインターフェース情報 (データ名 AH)	×
31	XTC 拡張トランザクションインターフェース情報 (データ名 AI)	△
32	MCP 拡張トランザクションインターフェース情報 (データ名 AJ)	○
33	メッセージ種別 (データ名 AK)	×
34	エラートランザクション要因 (データ名 AL)	×
35	拡張ユーザ IFA エリアサイズ (データ名 AM)	○
36	拡張ユーザ SAT エリアサイズ (データ名 AN)	○
37	トランザクション起動要因 (データ名 AO)	×
38	サーバ再起動回数 (データ名 AP)	×
39	トランザクション起動時刻 (データ名 AQ)	○
40	送信元のノード識別子 (データ名 AR)	×
41	トランザクションレベル名長 (データ名 AS)	○
42	トランザクションレベル名 (データ名 AT)	○

(凡例)

○ : 参照できます。

× : 参照できません。

△ : XTC を使用している場合だけ参照できます。

エラートランザクション (ERRTRNS)

トランザクション種別が ES の場合に、TP1/EE と設定値が異なるエラー要因コード (err_code) を次に示します。

■エラー要因コード

10030

メモリ不足が発生しました。

10025

送信障害が発生しました。

10020

論理端末が閉塞中です。

10007

送信タイムアウトが発生しました。

10021

出力メッセージ編集 UOC がエラーリターンしました。

10022

出力メッセージ編集 UOC で障害が発生しました。

次のトランザクションインターフェース情報については TP1/EE から SPP に渡されません。

- 送信サービスグループ名長
- 送信サービスグループ名
- 送信サービス名長
- 送信サービス名

MCP 拡張トランザクションインターフェース情報

MCP 拡張トランザクションインターフェース情報は、データ名 AJ に設定される情報です。次に示す場合に参照できます。

- トランザクション種別 (データ名 J) が「MN」, 「E1」, 「E2」, 「E3」, 「ES」または「ER」でメッセージ種別 (データ名 AK) が 5 または 6 の場合
- トランザクション種別 (データ名 J) が「RL」の場合

データ名 AJ が示す構造体の形式を次に示します。

01 一意名.

02 データ名MA	PIC S9(9) COMP.	/*コネクションIDサイズ	*/
02 データ名MB	PIC X(9).	/*コネクションID	*/
02 FILLER	PIC X(3).	/*予備	*/
02 データ名MC	PIC S9(9) COMP.	/*メッセージの入出力論理端末名サイズ	*/

02 データ名MD	PIC X(9).	/*メッセージの入出力論理端末名	*/
02 FILLER	PIC X(3).	/*予備	*/
02 データ名ME	PIC S9(9) COMP.	/*プロトコル種別	*/
02 データ名MF	PIC S9(9) COMP.	/*種別コード	*/
02 データ名MG	PIC S9(9) COMP.	/*理由コード	*/
02 FILLER	PIC X(4).	/*MCP使用領域	*/
02 FILLER	ADDRESS.	/*MCP使用領域	*/
02 データ名MH	ADDRESS.	/*MCP使用領域	*/
02 データ名MI	PIC S9(9) COMP.	/*MCP使用領域	*/
02 FILLER	PIC X(4).	/*予備	*/
02 データ名MJ	PIC X(16).	/*自システムのIPアドレス	*/
02 データ名MK	PIC X(16).	/*相手システムのIPアドレス	*/
02 データ名ML	PIC S9(4) COMP.	/*自システムのポート番号	*/
02 データ名MM	PIC S9(4) COMP.	/*自システムのポート番号	*/
02 FILLER	PIC X(28).	/*予備	*/

●データ名 MA

コネクション ID サイズが設定されます。

●データ名 MB

コネクション ID が設定されます。

●データ名 MC

メッセージの入出力論理端末名サイズが設定されます。

●データ名 MD

メッセージの入出力論理端末名が設定されます。

●データ名 ME

プロトコル種別として、次の値が設定されます。

1 : TCP/IP プロトコル

2 : UDP プロトコル

●データ名 MF

種別として、次の値が設定されます。

1 : コネクションの確立通知

2 : コネクションの解放通知

3 : 論理端末の閉塞解除通知

4 : 論理端末の閉塞通知

●データ名 MG

データ名 MF に対する値が設定されます。

データ名 MF に対するデータ名 MG の値の意味を次の表に示します。

表 5-12 データ名 MF に対するデータ名 MG の値の意味

項番	データ名 MF	データ名 MG	意味	プロトコル
1	1	1	CBLEEMCP('ACTCN ')関数または CBLEEMCP('ACTCNTO ')関数によるコネクションの確立	T
2		2	eemcpactcn コマンドまたは eemcpactcnto コマンドによるコネクションの確立	T
3		5	システムによるコネクションの確立	T
4		1	CBLEEMCP('DCTCN ')関数 (データ名 C に「VALUE 'F'」設定時) によるコネクションの解放	T
5		2	eemcpdctcn コマンドによるコネクションの解放	T
6		3	CBLEEMCP('DCTCN ')関数 (データ名 C に「VALUE 'N'」設定時) によるコネクションの強制解放	T
7		4	eemcpdctcn コマンド (-f オプションを指定時 ^{※1}) によるコネクションの強制解放	T
8		5	システムによるコネクションの解放	T
9		6	システムによるコネクションの強制解放	T
10		101	コネクション確立不可 ^{※2}	T
11		102	コネクション解放 (相手システムからの解放)	T
12		103	コネクション解放 (自システムからの解放)	T
13		104	受信バッファオーバフロー	T
14		105	受信バッファ不足	T
15		106	後続メッセージ受信タイムアウト	T
16		107	コネクションリプレースによる切断	T
17		108	タイマ値不正	T
18		109	タイマセット指示種別不正	T
19		110	セグメント完/未完種別不正	T
20		111	次メッセージのサイズ不正	T
21		112	次メッセージの先頭アドレス不正	T
22		113	該当メッセージのサイズ不正	T
23		114	残留該当メッセージのサイズ不正	T
24		121	無通信状態監視タイムアウト	T
25		122	同期型メッセージの送受信タイムアウト	T

項目番号	データ名 MF	データ名 MG	意味	プロトコル
26		2 UOC からのリターン コード	入力セグメント判定 UOC エラーリターン	T
27	3	1	CBLEEMCP('ACTLE ')関数による論理端末の閉塞解除	T, U
28		2	eemcpactle コマンドによる論理端末の閉塞解除	T, U
29		5	システムによる論理端末の閉塞解除	T, U
30	4	1	CBLEEMCP('DCTLE ')関数による論理端末の閉塞	T, U
31		2	eemcpdctle コマンドによる論理端末の閉塞	T, U
32		5	システムによる論理端末の閉塞	T, U
33		104	受信バッファオーバフロー	T, U
34		105	送受信バッファ不足	T, U
35		115	送信バッファオーバフロー	T, U
36		116	使用バッファ数不正	T, U
37		117	有効セグメント不正	T, U
38		118	編集バッファアドレス不正	T, U
39		119	論理端末の閉塞解除不可※2	T, U
40		120	送受信失敗による論理端末の閉塞	T, U
41		UOC からのリターン コード	入力メッセージ編集 UOC または出力メッセージ編集 UOC エラーリターン	T, U

(凡例)

T : pro_kind が EEMCP_UOC_PRO_TCP (TCP/IP)

U : pro_kind が EEMCP_UOC_PRO_UDP (UDP)

注※1

eemcpn 定義コマンドの-f オプションの cnrelease オペランドに rst を指定する必要があります。

注※2

前に出力されたメッセージを確認してください。

●データ名 MH

MCP が使用する領域です。

●データ名 MI

MCP が使用する領域です。

●データ名 MJ

自システムの IP アドレスが設定されます。

下位 4 バイトに次の形式で設定されます。

dotted-decimal 形式 : 194. 11. 42. 20
↓ ↓ ↓ ↓
16進数形式 (4バイト) : C2 0B 2A 14 …この形式で設定してください。

データ名 MJ

00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]

●データ名 MK

相手システムの IP アドレスが設定されます。

設定形式はデータ名 MJ を参照してください。

●データ名 ML

自システムのポート番号が設定されます。

ポート番号はホストバイトオーダー形式の数値で設定されます。

●データ名 MM

相手システムのポート番号が設定されます。

ポート番号はホストバイトオーダー形式の数値で設定されます。

データ名 MJ, データ名 MK, データ名 ML, データ名 MM は次の条件をすべて満たした場合に設定されます。

- ・ データ名 ME が 1 (TCP/IP プロトコル)
- ・ データ名 MF が 2 (コネクション解放通知) 以外

CBLEEMCP('ACTCN ') - コネクションの確立

機能

コネクションを確立します。

コネクションの確立後, MCP 後処理トランザクションを起動します。

コネクションの確立は非同期で実行されるため、この関数で返されるリターン値が 00000 の場合でも、コネクションの確立が失敗しているときがあります。コネクションの確立結果は、MCP 後処理トランザクションによって通知されます。

この関数で指定したコネクション ID 中に、相手システム (eemcpcn オペランド -o) を省略した eemcpcn オペランドのコネクション ID が含まれていた場合は、指定した全コネクション ID のコネクション確立処理は実行されません。

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.
 02 データ名A  PIC X(8)  VALUE 'ACTCN' .
 02 データ名B  PIC X(5).
 02 FILLER      PIC X(3).
 02 データ名Z  PIC S9(9) COMP VALUE ZERO.

01 一意名2.
 02 FILLER      PIC X(4).
 02 データ名C  PIC 9(9)  COMP.
 02 データ名D  PIC X(9).
 02 データ名E  PIC X(9).
 02 データ名F  PIC X(9).
 02 データ名G  PIC X(9).
 02 データ名H  PIC X(9).
 02 データ名I  PIC X(9).
 02 データ名J  PIC X(9).
 02 データ名K  PIC X(9).
```

UAP で値を設定する引数

●データ名 A

コネクションの確立を示す要求コードを「VALUE 'ACTCN△△△'」と設定します。

●データ名 Z

0 を設定します。

●データ名 C

コネクションを確立するコネクション ID 数を設定します。1~8 の範囲で指定してください。

●データ名 D, E, F, G, H, I, J, K

コネクションを確立するコネクション ID を設定します。コネクション ID は eemcpcn 定義コマンドの-y オプションの mode オペランドに client を指定したコネクションだけ設定できます。

コネクション ID の長さは最大 8 バイトです。コネクション ID の後ろは空白で埋めてください。

コネクション ID は、*を使って一括指定できます。一括指定は一つだけ指定できます。一括指定する場合は、データ名 C に 1 を設定して、データ名 D に「*」または「先行文字列*」の形式で設定してください。

「*」の場合はすべてのコネクションを確立します。「先行文字列*」の場合は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。

TP1/EE 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。
00001	TP1/EE の環境下にありません。
00004	UOC から発行しているため、この機能は使用できません。
10003	データ名 A に設定した値が間違っています。
10005	データ名 Z に設定した値が間違っています。
10008	データ名 C に設定した値が間違っています。
10009	発行条件が不正です。
10010	指定したコネクション ID の中に不正なコネクション ID があります。
10027	システムエラーが発生しました。
10028	一部のコネクションの確立に失敗しました。
10030	メモリ不足が発生しました。
10048	指定したコネクションはすでに使用中です。

注意事項

コネクションの確立は非同期で行われます。したがって、この関数で返されるステータスコードが「00000」であった場合でも、コネクションの確立に失敗している場合があります。コネクションの確立結果は MCP 後処理トランザクションで通知されます。

CBLEEMCP('ACTCNTO ') - 相手システムを指定したコネクションの確立

機能

コネクションを確立します。

コネクション確立後、MCP 後処理トランザクションを起動します。

eemcpn 定義の-o オプション指定値（相手システム情報）を無視します。

コネクションの確立は非同期で行われます。そのため、この関数のステータスコード値が 00000 であった場合でも、コネクションの確立が失敗しているときがあります。コネクションの確立結果は MCP 後処理トランザクションによって通知されます。

形式

PROCEDURE DIVISION の指定

```
CALL 'CBLEEMCP' USING 一意名1 一意名2

01 一意名1.
  02 データ名A  PIC X(8) VALUE 'ACTCNT0'.
  02 データ名B  PIC X(5).
  02 FILLER     PIC X(3).
  02 データ名Z  PIC S9(9) COMP VALUE ZERO.

01 一意名2.
  02 データ名C  PIC X(9).
  02 FILLER     PIC X(7).
  02 データ名D  PIC X(16).
  02 データ名E  PIC X(256).
  02 データ名F  PIC 9(4) COMP-X.
  02 FILLER     PIC X(6).
```

UAP で値を設定するデータ領域

●データ名 A

コネクションの確立を示す要求コードを「VALUE 'ACTCNT0△'」と設定します。

●データ名 Z

0 を設定します。

●データ名 C

コネクションを確立するコネクション ID を設定します。コネクション ID は MCP 関連定義の eemcpn オペランドの-y mode に client を指定したものに限ります。コネクション ID の長さは最大 8 バイトです。後ろに空白を設定してください。8 バイトに満たないコネクション ID 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 D

相手システムの IP アドレスを設定します。

データ名 E を設定した場合はデータ名 D の設定は不要です。データ名 E, データ名 D 共に設定した場合はデータ名 D 設定値が有効になります。

IP アドレスを設定する場合は、xxx.xxx.xxx.xxx の形式で設定します。xxx は、0～255（10 進数）で設定してください。ただし、0.0.0.0, 255.255.255.255 の設定は不可とします。

IP アドレスはデータ名 D の下位 4 バイトに次の形式で設定します。

dotted-decimal 形式：
194. 11. 42. 20
↓ ↓ ↓ ↓
C2 0B 2A 14
16進数形式（4 バイト）：
…この形式で設定してください。

データ名 D

00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]

●データ名 E

相手システムのホスト名を設定します。

データ名 D を設定した場合はデータ名 E の設定は不要です。データ名 D, データ名 E 共に設定した場合はデータ名 D 設定値が有効になります。

ホスト名の長さは最大 255 文字です。ホスト名の後ろに空白文字を付けてください。

ホスト名は、/etc/hosts ファイル、DNS などで、IP アドレスとマッピングできるように指定する必要があります。

●データ名 F

相手システムのポート番号を設定します。

ポート番号はホストバイトオーダー形式の数値で設定してください。数値は、1～65535 の符号なし整数で設定してください。

TP1/EE 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	正常に受け付けました。
00001	TP1/EE 環境下にありません。
00004	UOC から発行しているため、この機能は使用できません。
10003	データ名 A に設定した値が間違っています。
10009	発行条件が不正です。

ステータスコード	意味
10005	データ名 Z に設定した値が間違っています。
10040	指定した IP アドレスまたはホスト名が不正です。
10041	指定したポート番号が不正です。
10010	指定したコネクション ID が不正です。
10030	メモリ不足が発生しました。
10027	システムエラーが発生しました。
10048	指定したコネクションはすでに使用中です。

CBLEEMCP('ACTLE ') - 論理端末の閉塞解除

機能

論理端末を閉塞解除します。

論理端末の閉塞解除後、MCP 後処理トランザクションを起動します。

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.
  02 データ名A  PIC X(8)  VALUE 'ACTCN' .
  02 データ名B  PIC X(5).
  02 FILLER      PIC X(3).
  02 データ名Z  PIC S9(9) COMP ZERO.

01 一意名2.
  02 データ名C  PIC 9(9)  COMP.
  02 データ名D  PIC X(9).
  02 データ名E  PIC X(9).
  02 データ名F  PIC X(9).
  02 データ名G  PIC X(9).
  02 データ名H  PIC X(9).
  02 データ名I  PIC X(9).
  02 データ名J  PIC X(9).
  02 データ名K  PIC X(9).
```

UAP で値を設定する引数

●データ名 A

論理端末の閉塞解除を示す要求コードを「VALUE 'ACTLE△△△'」と設定します。

●データ名 Z

0 を設定します。

●データ名 C

閉塞解除する論理端末数を設定します。

1~8 の範囲で設定してください。

●データ名 D, E, F, G, H, I, J, K

閉塞解除する論理端末名を設定します。論理端末名の長さは最大 8 バイトです。論理端末名の後ろは空白で埋めてください。

論理端末名は、*を使って一括指定できます。一括指定は一つだけ指定できます。一括指定する場合は、データ名 C に 1 を設定しデータ名 D に「*」または「先行文字列*」の形式で設定してください。

「*」の場合はすべての論理端末を閉塞解除します。「先行文字列*」の場合は先行文字列で始まるすべての論理端末を閉塞解除します。

TP1/EE 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。
00001	TP1/EE の環境下にありません。
00004	UOC から発行しているため、この機能は使用できません。
10003	データ名 A に設定した値が間違っています。
10005	データ名 Z に設定した値が間違っています。
10008	データ名 C に設定した値が間違っています。
10009	発行条件が不正です。
10011	指定した論理端末名の中に不正な論理端末名があります。
10027	システムエラーが発生しました。
10028	一部の論理端末の閉塞解除に失敗しました。
10030	メモリ不足が発生しました。

注意事項

- 指定した論理端末名が UDP 端末の場合は、該当ポートをオープンします。
- 論理端末の閉塞解除は非同期で行われます。したがって、この関数で返されるステータスコードが「00000」であった場合でも、論理端末の閉塞解除に失敗している場合があります。論理端末閉塞解除の結果は、MCP 後処理トランザクションで通知されます。

CBLEEMCP('CNINFO ') - コネクション情報の取得

機能

コネクション情報を取得します。

形式

PROCEDURE DIVISION の指定

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

01 一意名1.
  02 データ名A  PIC X(8) VALUE 'CNINFO '.
  02 データ名B  PIC X(5).
  02 FILLER      PIC X(3).
  02 データ名Z  PIC S9(9) COMP VALUE ZERO.

01 一意名2.
  02 データ名C  PIC X(3).
  02 データ名D  PIC X(9).
  02 FILLER      PIC X(4).

01 一意名3.
  02 データ名E  PIC X(9).
  02 FILLER      PIC X(1).
  02 データ名F  PIC S9(4) COMP.
  02 データ名G  PIC X(4).
  02 データ名H  PIC X(16).
  02 データ名I  PIC X(16).
  02 データ名J  PIC 9(4) COMP-X.
  02 データ名K  PIC 9(4) COMP-X.
  02 FILLER      PIC X(4).
```

UAP で値を設定するデータ領域

●データ名 A

コネクションの確立を示す要求コードを「VALUE 'CNINFO△△'」と設定します。

●データ名 Z

0 を設定します。

●データ名C

情報を取得するコネクションの指定方法を次の形式で設定します。

VALUE 'LE△'

情報を取得するコネクションを論理端末名称で指定するときに設定します。

VALUE 'CN△'

情報を取得するコネクションをコネクションIDで指定するときに設定します。

●データ名D

情報を取得するコネクションの論理端末名称、またはコネクションIDを設定します。論理端末名称、またはコネクションIDは8バイト以内で設定し、文字列の最後に空白文字を付けます。

TP1/EE から値が返されるデータ領域

データ名Eからデータ名Kが設定されるのはステータスコードが00000の場合だけです。

●データ名E

要求したコネクションのコネクションIDが設定されます。

コネクションIDは8バイト以内で設定し、文字列の最後にNULL文字を付けます。

●データ名F

要求したコネクションのコネクションIDの文字列長が設定されます。

●データ名G

要求したコネクションのプロトコル種別が設定されます。

VALUE 'TCP△'

TCP/IPプロトコル

●データ名H

自システムのIPアドレスが設定されます。

IPアドレスはデータ名Hの下位4バイトに次の形式で設定されます。

dotted-decimal形式 : 194. 11. 42. 20
↓ ↓ ↓ ↓
C2 0B 2A 14
16進数形式(4バイト) : この形式で設定されます。

データ名H

00	00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]	

●データ名I

相手システムの IP アドレスが設定されます。

IP アドレスの設定形式に関してはデータ名 H を参照してください。

●データ名J

自システムのポート番号が設定されます。

ポート番号はホストバイトオーダー形式の数値で設定されます。

●データ名K

相手システムのポート番号が設定されます。

ポート番号はホストバイトオーダー形式の数値で設定されます。

●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	正常に受け付けました。
00001	TP1/EE 環境下にありません。
00004	UOC から発行しているため、この機能は使用できません。
10003	データ名 A に設定した値が間違っています。
10005	データ名 Z に設定した値が間違っています。
10004	データ名 C に設定した値が間違っています。
10044	データ名 D に設定した値が間違っています。
10045	指定したコネクションが確立処理中状態です。
10046	指定したコネクションが解放処理中状態です。
10047	指定したコネクションが未確立状態です。

CBLEEMCP('DCTCN ') - コネクションの解放

機能

コネクションを解放します。

コネクションの解放後、MCP 後処理トランザクションを起動します。

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.
  02 データ名A  PIC X(8)  VALUE 'DCTCN' .
  02 データ名B  PIC X(5).
  02 FILLER      PIC X(3).
  02 データ名Z  PIC S9(9) COMP VALUE ZERO.

01 一意名2.
  02 データ名C  PIC X(1).
  02 FILLER      PIC X(3).
  02 データ名D  PIC 9(9)  COMP.
  02 データ名E  PIC X(9).
  02 データ名F  PIC X(9).
  02 データ名G  PIC X(9).
  02 データ名H  PIC X(9).
  02 データ名I  PIC X(9).
  02 データ名J  PIC X(9).
  02 データ名K  PIC X(9).
  02 データ名L  PIC X(9).
```

UAP で値を設定する引数

●データ名 A

コネクションの解放を示す要求コードを「VALUE 'DCTCN△△△'」と設定します。

●データ名 Z

0 を設定します。

●データ名 C

コネクション解放形態を設定します。この設定値は、eemcpcn 定義コマンドの-f オプションの cnrelease オペランドに rst を指定した場合に有効になります。fin を指定した場合は、データ名 C の設定値に関係なく FIN パケットを送信してコネクションを解放します。

VALUE 'F'

RST パケットを送信して、コネクションを強制的に解放する場合に設定します。

VALUE 'N'

FIN パケットを送信して、コネクションを解放する場合に設定します。

●データ名 D

コネクションを解放するコネクション ID 数を設定します。1~8 の範囲で設定してください。

●データ名 E, F, G, H, I, J, K, L

コネクションを解放するコネクション ID を設定します。コネクション ID の長さは最大 8 バイトです。コネクション ID の後ろは空白で埋めてください。

コネクション ID は、*を使って一括指定できます。一括指定は一つだけ指定できます。一括指定する場合は、データ名 C に 1 を設定しデータ名 D に「*」または「先行文字列*」の形式で設定してください。

「*」の場合はすべてのコネクションを解放します。「先行文字列*」の場合は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。

TP1/EE 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。
00001	TP1/EE の環境下にありません。
00004	UOC から発行しているため、この機能は使用できません。
10003	データ名 A に設定した値が間違っています。
10005	データ名 Z に設定した値が間違っています。
10008	データ名 D に設定した値が間違っています。
10009	発行条件が不正です。
10010	指定したコネクション ID の中に不正なコネクション ID があります。
10012	データ名 C に設定した値が間違っています。
10027	システムエラーが発生しました。
10028	一部のコネクションの解放に失敗しました。

注意事項

コネクションの解放は非同期で行われます。したがって、この関数で返されるステータスコードが「00000」であった場合でも、コネクションの解放に失敗している場合があります。コネクションの解放結果は MCP 後処理トランザクションで通知されます。

CBLEEMCP('DCTLE ') - 論理端末の閉塞

機能

論理端末を閉塞します。

論理端末の閉塞後、MCP 後処理トランザクションを起動します。

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.
  02 データ名A  PIC X(8)  VALUE 'DCTCN' .
  02 データ名B  PIC X(5).
  02 FILLER      PIC X(3).
  02 データ名Z  PIC S9(9) COMP ZERO.

01 一意名2.
  02 データ名C  PIC 9(9)  COMP.
  02 データ名D  PIC X(9).
  02 データ名E  PIC X(9).
  02 データ名F  PIC X(9).
  02 データ名G  PIC X(9).
  02 データ名H  PIC X(9).
  02 データ名I  PIC X(9).
  02 データ名J  PIC X(9).
  02 データ名K  PIC X(9).
```

UAP で値を設定する引数

●データ名 A

論理端末の閉塞を示す要求コードを「VALUE 'DCTLE△△△'」と設定します。

●データ名 Z

0 を設定します。

●データ名 C

閉塞する論理端末数を設定します。

1~8 の範囲で設定してください。

●データ名 D, E, F, G, H, I, J, K

閉塞する論理端末名を設定します。論理端末名の長さは最大 8 バイトです。論理端末名の後ろは空白で埋めてください。

論理端末名は、*を使って一括指定できます。一括指定は一つだけ指定できます。一括指定する場合は、データ名 C に 1 を設定しデータ名 D に「*」または「先行文字列*」の形式で設定してください。

「*」の場合はすべての論理端末を閉塞します。「先行文字列*」の場合は先行文字列で始まるすべての論理端末を閉塞します。

TP1/EE 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。
00001	TP1/EE の環境下にありません。
00004	UOC から発行しているため、この機能は使用できません。
10003	データ名 A に設定した値が間違っています。
10005	データ名 Z に設定した値が間違っています。
10008	データ名 C に設定した値が間違っています。
10009	発行条件が不正です。
10011	指定した論理端末名の中に不正な論理端末名があります。
10027	システムエラーが発生しました。
10028	一部の論理端末の閉塞に失敗しました。

注意事項

- 指定した論理端末名が UDP 端末の場合は、該当ポートをクローズします。
- 論理端末の閉塞は非同期で行われます。したがって、この関数で返されるステータスコードが「00000」であった場合でも、論理端末の閉塞に失敗している場合があります。論理端末閉塞の結果は MCP 後処理トランザクションで通知されます。

CBLEEMCP('SEND ') - 非同期一方送信メッセージの送信

機能

相手システムへ非同期一方送信メッセージを送信します。非同期一方送信メッセージは、一つのセグメントで構成されます。

TP1/FSP を使用する場合、eemcpfunc 定義コマンドの -o オプションの outputmessage オペランドが設定されているとき、または TP1/EE のユーザサービス関連定義の service_attr 定義コマンドが設定されているときは、送信データを UAP 履歴情報のデータ種別「OJ（出力メッセージ）」として取得します。オペランドの関連性については、TP1/FSP の関連ドキュメントを参照してください。

形式

PROCEDURE DIVISION の指定

CALL 'CBLEEMCP' 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(9)	COMP.		
02	データ名V	PIC	X(8)	.		
02	データ名W	PIC	X(n)	.		

UAP で値を設定する引数

●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, D

空白を設定します。

●データ名 E, F

MCP で使用する領域です。

●データ名 G

一方送信メッセージの送信属性を指定します。

0 以外を複数指定する場合は、指定する数値を加算して設定します。

0

メッセージを非永続属性で、かつ UDP 送信時の送信先をデータ名 W の制御ヘッダで指定する場合に設定します。

1

永続メッセージとする場合に設定します。

MCP をクラスタ構成に組み込んだ場合 (XTC サービス定義の cluster_mode オペランドに Y を指定) にだけ有効となります。

2

UDP 送信時の送信先を eemcple 定義コマンドの -o オプションで指定する場合に設定します。

なお、この値を指定した場合は、制御ヘッダの内容に関係なく、MCP 構成定義で指定した送信先が有効になります。

●データ名 H

单一セグメントの送信を示す「VALUE 'EMI△'」を設定します。

●データ名 I, J, K, L, M1, M2, M3

空白を設定します。

●データ名 M4, M5

0 を設定します。

●データ名 M6, M7

空白を設定します。

●データ名 N

MCP 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を設定します。論理端末名の長さは最大 8 バイトです。論理端末名の後ろは空白で埋めてください。

●データ名 Q, R

空白を設定します。

●データ名 T

MCP で使用する領域です。

●データ名 U

送信するセグメントの長さを設定します。設定できる最大値は、メッセージの最大サイズを拡張しているかどうか (eemcpfunc 定義コマンドの-m オプションの msglen オペランド指定値) によって、次の表に示すとおり異なります。

表 5-13 メッセージの最大サイズ拡張有無による送信メッセージ長の違い

メッセージの最大サイズ拡張有無	送信メッセージ長に設定できる最大値 (単位: バイト)	
	TCP/IP プロトコル使用時	UDP プロトコル使用時
なし (msglen オペランドに normal を指定)	32000*	32000*
あり (msglen オペランドに extend を指定)	8388608	65531*

注※

TP1/EE サービス定義の send_message_buf_size オペランドで指定した送信バッファサイズの方が小さい場合は、send_message_buf_size オペランドの指定値が設定されます。

●データ名 V

空白を設定します。

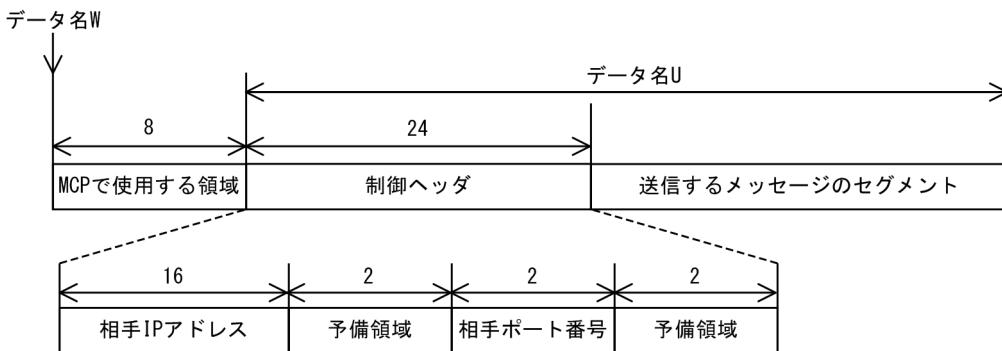
●データ名 W

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。送信できる最大長は、データ名 U を参照してください。

TCP/IP の場合



UDP の場合



■制御ヘッダ

制御ヘッダ (EEMCPUDPHD) は、<eemcp.h>に次のように定義されています。

```
typedef struct ee_mcp_udp_header {
    unsigned char r_ipaddr[16]; ... 相手IPアドレス
    unsigned char rsv1[2]; ... 予備
    unsigned short r_portno; ... 相手ポート番号
    unsigned char rsv2[4]; ... 予備
} EEMCPUDPHD;
```

■相手IPアドレス

相手IPアドレスはr_ipaddrの下位4バイトに次に示す形式で設定します。

dotted-decimal形式 : 194. 11. 42. 20
 ↓ ↓ ↓ ↓
 16進数形式 (4バイト) : C2 0B 2A 14

変換

...この形式で設定してください。

r_ipaddr[16]

00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]

■相手ポート番号

ポート番号は次の形式で設定します。

10進数形式 : 1024
 ↓
 16進数形式 (2バイト) : 0x0400

変換

...この形式 (ビッグエンディアン) で設定してください。

TP1/EE 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。
00001	TP1/EE の環境下にありません。
00004	UOC から発行しているため、この機能は使用できません。
10001	データ名 U に、送信できる最大サイズを超える値を設定しています。

ステータスコード	意味
10002	データ名 U に 0 以下の値を設定しています。
10003	データ名 A, W に設定した値が間違っています。
10004	データ名 H に設定した値が間違っています。
10005	データ名 G に設定した値が間違っています。
10006	データ名 O に設定した値が間違っています。
10009	発行条件が不正です。 CBLEETRN('ROLLMARK')関数が発行済みです。
10011	データ名 P に設定した論理端末名が間違っています。
10020	データ名 P に設定した論理端末名は閉塞中です。
10030	メモリ不足が発生しました。

注意事項

- この関数発行後に、CBLEEMCP('SENDSYNC')関数を発行した場合、CBLEEMCP('SENDSYNC')関数で送信したメッセージが先に送信されます。
- 送信要求したトランザクションがロールバックした場合、送信メッセージは破棄されます。
- ERRTRNS 上でこの関数を発行して、送信障害などのエラーが発生した場合、ERRTRNS は起動されないで、送信メッセージは破棄されます。
- この関数を発行したトランザクションでは、トランザクションの最適化を行いません。
- 非永続メッセージのあとに、永続メッセージを送信した場合、永続メッセージが先に送信されます。
- この関数は、トランザクションのコミット時に実行されます。したがって、トランザクションの同期点を取得する処理でエラーが起こった場合、この関数の処理は無効となります。
- TP1/FSP を使用する場合、次のステータスコードのときは、「OJ (出力メッセージ)」は取得されません。

00001, 00004, 10001, 10002, 10003, 10004, 10005, 10006, 10011

CBLEEMCP('SENDRECV ') - 同期型メッセージの送受信

機能

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

TP1/FSP を使用する場合、アプリケーション定義の eemcpfunc の-o オプションの outputmessage が設定されている場合、または、ユーザサービス関連定義 service_attr の-m オプションが設定されている場合

に、送信データを UAP 履歴情報のデータ種別「OJ (出力メッセージ)」として取得します。オペランドの関連性については、TP1/FSP の関連ドキュメントを参照してください。

形式

PROCEDURE DIVISION の指定

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

01 一意名1.
 02 データ名A  PIC X(8) VALUE 'SENDRECV'.
 02 データ名B  PIC X(5).
 02 FILLER    PIC X(3).
 02 データ名C  PIC X(4) VALUE SPACE.
 02 データ名D  PIC X(4) VALUE SPACE.
 02 データ名E  PIC 9(8).
 02 データ名F  PIC 9(8).
 02 データ名G  PIC 9(9) COMP VALUE ZERO.
 02 データ名H  PIC X(4).
 02 データ名I  PIC 9(9) COMP.
 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 S9(9) COMP.
 02 データ名M6 PIC X(1) VALUE SPACE.
 02 データ名M7 PIC X(1).
 02 データ名N  PIC X(14) VALUE LOW-VALUE.

01 一意名2.
 02 データ名O  PIC X(4) VALUE 'IO '.
 02 データ名P  PIC X(8).
 02 データ名Q  PIC X(8) VALUE SPACE.
 02 データ名R  PIC X(8) VALUE SPACE.
 02 データ名T  PIC X(28) VALUE LOW-VALUE.

01 一意名3.
 02 データ名U  PIC 9(9) COMP.
 02 データ名V  PIC X(8).
 02 データ名W  PIC X(n).

01 一意名4.
 02 データ名X  PIC 9(9) COMP.
 02 データ名Y  PIC X(8)
 02 データ名Z  PIC X(n).
```

UAP で値を設定するデータ領域

●データ名 A

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

●データ名 C,D

空白を設定します。

●データ名 G

指定した論理端末がすでに送信処理中のとき、送信可能となるまで待ち合わせるか、エラーリターンとするかを、次のどちらかの数値で設定します。

0

送信可能となるまで待ち合わせる場合に設定します。

待ち合わせている間の時間は、watcitime に設定した時間に含まれます。

1

エラーリターンする場合に設定します。

●データ名 H

単一セグメントの送信を示す「VALUE 'EMI△'」を設定します。

●データ名 I

セグメントを受信する領域の長さを設定します。領域の長さは、[図 5-2](#) のデータ名 I を参照してください。

領域の長さは、データ名 X + 8 バイト以上の長さを設定してください。

●データ名 J,K,L,M1,M2,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

CBLEEMCP('SENDRECV')を呼び出してから終了するまでの最大時間（1～65535 単位：秒）を設定します。0 を設定した場合、MCP 共通定義（eemcpcn -b sndrcvtime）で指定した同期送受信監視時間が仮定されます。負の値を設定した場合は、無限に待ち合わせをします。

●データ名 M6

空白を設定します。

●データ名 M7

空白を設定します。

●データ名 N

MCP で使用する領域です。

●データ名 O

同期型のメッセージの送受信を示す「VALUE 'IO△△'」を設定します。

●データ名 P

出力先の論理端末名を設定します。論理端末名の長さは最大 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,R

空白を設定します。

●データ名 T

MCP で使用する領域です。

●データ名 U

送信するセグメントの長さを設定します。設定できる最大値は、メッセージの最大サイズを拡張しているかどうか (eemcpfunc 定義コマンドの-m オプションの msglen オペランド指定値) によって、次の表に示すとおり異なります。

表 5-14 メッセージの最大サイズ拡張有無による送信メッセージ長の違い

メッセージの最大サイズ拡張有無	送信メッセージ長に設定できる最大値 (単位: バイト)	
	TCP/IP プロトコル使用時	UDP プロトコル使用時
なし (msglen オペランドに normal を指定)	32000*	32000*
あり (msglen オペランドに extend を指定)	8388608	65531*

注※

TP1/EE サービス定義の send_message_buf_size オペランドで指定した送信バッファサイズの方が小さい場合は、send_message_buf_size オペランドの指定値が設定されます。

●データ名 V

MCP で使用する領域です。

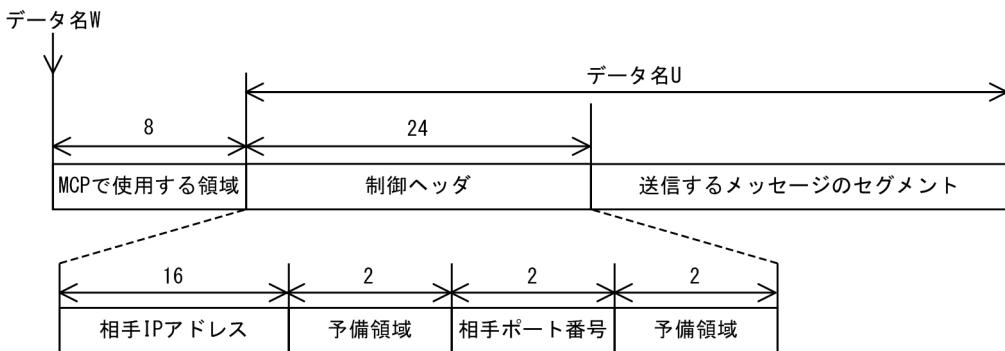
●データ名 W

送信するセグメントの内容を設定した領域を設定します。送信できる最大長は、データ名 U を参照してください。

TCP/IP の場合



UDP の場合



■制御ヘッダ

制御ヘッダ (EEMCPUDPHD) は、<eemcp.h>に次のように定義されています。

```
typedef struct ee_mcp_udp_header {  
    unsigned char    r_ipaddr[16];    ... 相手IPアドレス  
    unsigned char    rsv1[2];        ... 予備  
    unsigned short   r_portno;     ... 相手ポート番号  
    unsigned char    rsv2[4];        ... 予備  
} EEMCPUDPHD;
```

■相手IPアドレス

相手IPアドレスはr_ipaddrの下位4バイトに次に示す形式で設定します。

dotted-decimal形式 : 194. 11. 42. 20
↓ ↓ ↓ ↓
C2 0B 2A 14

16進数形式 (4バイト) : C2 0B 2A 14 ...この形式で設定してください。

r_ipaddr[16]

00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]

■相手ポート番号

ポート番号は次の形式で設定します。

10進数形式 : 1024
↓
0x0400

16進数形式 (2バイト) : 0x0400 ...この形式 (ビッグエンディアン) で設定してください。

●データ名 Y

MCP で使用する領域です。

TP1/EE 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。

ステータスコードが 00000 または 10031 の場合だけ、次に示すデータ名 E, データ名 F, データ名 X, データ名 Z の領域は参照可能です。これ以外のステータスコードの場合は参照しないでください。

●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日付が HHMMSS00(HH:時 MM:分 SS:秒)の形式で返されます。

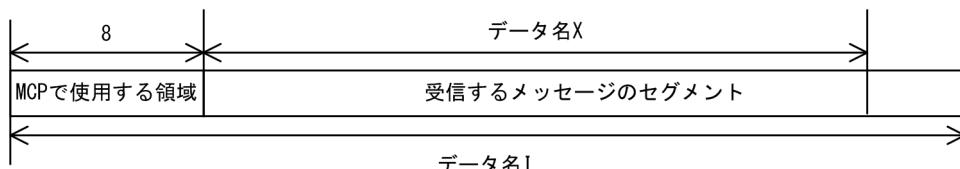
●データ名X

受信したメッセージの長さが返されます。

●データ名Z

受信したセグメント内容が返されます。

図 5-2 受信のデータ形式



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。
00001	TP1/EE 環境下にありません。
00004	UOC から発行しているため、この機能は使用できません。
10009	発行条件が不正です。
10001	データ名 U に送信サイズを超える値を設定しています。
10002	データ名 U に 0 以下の値を設定しています。
10003	データ名 A,E,F, M5,W,X,Z に設定した値が間違っています。
10004	データ名 H に設定した値が間違っています。
10005	データ名 G に設定した値が間違っています。
10006	データ名 O に設定した値が間違っています。
10007	データ名 M5 に設定した時間が経過しました。
10011	データ名 P に設定した論理端末名が間違っています。
10020	データ名 P に設定した論理端末名は閉塞中です。
10021	出力メッセージ編集 UOC がエラーリターンしました。
10022	出力メッセージ編集 UOC で障害が発生しました。
10025	システムに対する送信時に障害が発生しました。
10030	メモリ不足が発生しました。 <ul style="list-style-type: none">送信バッファ数または受信バッファ数が不足しました。メッセージサイズが OBF サイズまたは IBF サイズを超過しました。

ステータスコード	意味
10031	データ名 I の指定値を超えるセグメントを受信しました。データ名 I の指定値を超えた部分は切り捨てされました。
10032	データ名 I の指定値が不足しています。9 バイト以上の領域を確保してください。
10033	メッセージ受信が仕掛けかり中です。
10034	出力先の論理端末は、ほかの UAP で仕掛けかり中です。
10035	メッセージ送信完了監視タイマ(MCP 構成定義 eempcn の-b オプション sndcmptim オペランドに指定)のタイムアウトが発生しました。
10036	受信待ちでコネクション障害が発生しました。

注意事項

- CBLEEMCP('SEND ')関数発行後、CBLEEMCP('SENDRECV ')関数を発行した場合、CBLEEMCP('SENDRECV ')関数のメッセージが先に送信されます。
- リターン値が 10031 の場合、データ名 X にはデータ名 I の領域に格納できた受信データのサイズが設定されます。
- TP1/FSP を使用する場合、以下のステータスコードの場合、OJ は取得されません。
00001, 00004, 10001, 10002, 10003, 10004, 10005, 10006, 10009, 10011, 10032

CBLEEMCP('SEENDSYNC') - 同期一方送信メッセージの送信

機能

相手システムへ同期一方送信メッセージを送信します。

TP1/FSP を使用する場合、eemcpfunc 定義コマンドの-o オプションの outputmessage オペランドが設定されているとき、または TP1/EE のユーザサービス関連定義の service_attr 定義コマンドが設定されているときは、送信データを UAP 履歴情報のデータ種別「OJ (出力メッセージ)」として取得します。オペランドの関連性については、TP1/FSP の関連ドキュメントを参照してください。

形式

PROCEDURE DIVISION の指定

```
CALL 'CBLEEMCP' 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 'SEENDSYNC'.
02 データ名B  PIC X(5).
02 FILLER      PIC X(3).
02 データ名C  PIC X(4)  VALUE SPACE.
```

02	データ名D	PIC X(4) VALUE SPACE.
02	データ名E	PIC 9(8).
02	データ名F	PIC 9(8).
02	データ名G	PIC 9(9) COMP VALUE ZERO.
02	データ名H	PIC X(4).
02	データ名I	PIC X(4) VALUE SPACE.
02	データ名J	PIC X(4) VALUE SPACE.
02	データ名K	PIC X(4) VALUE SPACE.
02	データ名L	PIC X(8) VALUE SPACE.
02	データ名M1	PIC X(4) VALUE SPACE.
02	データ名M2	PIC X(8) VALUE SPACE.
02	データ名M3	PIC X(4) VALUE SPACE.
02	データ名M4	PIC 9(9) COMP VALUE ZERO.
02	データ名M5	PIC S9(9) COMP.
02	データ名M6	PIC X(1) VALUE SPACE.
02	データ名M7	PIC X(1).
02	データ名N	PIC X(14) VALUE LOW-VALUE.
01	一意名2.	
02	データ名O	PIC X(4) VALUE SPACE.
02	データ名P	PIC X(8).
02	データ名Q	PIC X(8) VALUE SPACE.
02	データ名R	PIC X(8) VALUE SPACE.
02	データ名T	PIC X(28) VALUE LOW-VALUE.
01	一意名3.	
02	データ名U	PIC 9(9) COMP.
02	データ名V	PIC X(8).
02	データ名W	PIC X(n).

UAP で値を設定する引数

●データ名 A

同期一方送信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, D

空白を設定します。

●データ名 E, F

MCP で使用する領域です。

●データ名 G

同期型メッセージの送信属性を指定します。

0

UDP 送信時の送信先をデータ名 W の制御ヘッダで指定する場合に設定します。

2

UDP 送信時の送信先を eemcple 定義コマンドの-o オプションで指定する場合に設定します。

なお、この値を指定した場合は、制御ヘッダの内容に関係なく、MCP 構成定義で指定した送信先が有効になります。

●データ名 H

单一セグメントの送信を示す「VALUE 'EMI△'」を設定します。

●データ名 I, J, K, L, M1, M2, 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

この関数を呼び出してから終了するまでの最大時間（単位：秒）を設定します。ここでの最大時間とは、ソケット空き待ち時間および送信開始から自システムの TCP/IP の送信バッファへの書き込み完了までの時間の合計をいいます。65535 を最大値として、値を設定できます。

0 を設定した場合、eemcpcn 定義コマンドの-b オプションの sndsynctim オペランドで指定した同期一方送信監視時間が仮定されます。負の値を設定した場合は、時間を監視しません。無限に送信完了を待ちます。UDP プロトコルを使用した通信時は無視されます。

●データ名 M6, M7

空白を設定します。

●データ名 N

MCP 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を設定します。論理端末名の長さは最大 8 バイトです。論理端末名の後ろは空白で埋めてください。

●データ名 Q, R

空白を設定します。

●データ名 T

MCP で使用する領域です。

●データ名 U

送信するセグメントの長さを設定します。設定できる最大値は、メッセージの最大サイズを拡張しているかどうか (eemcpfunc 定義コマンドの-m オプションの msglen オペランド指定値) によって、次の表に示すとおり異なります。

表 5-15 メッセージの最大サイズ拡張有無による送信メッセージ長の違い

メッセージの最大サイズ拡張有無	送信メッセージ長に設定できる最大値 (単位: バイト)	
	TCP/IP プロトコル使用時	UDP プロトコル使用時
なし (msglen オペランドに normal を指定)	32000*	32000*
あり (msglen オペランドに extend を指定)	8388608	65531*

注※

TP1/EE サービス定義の send_message_buf_size オペランドで指定した送信バッファサイズの方が小さい場合は、send_message_buf_size オペランドの指定値が設定されます。

●データ名 V

MCP で使用する領域です。

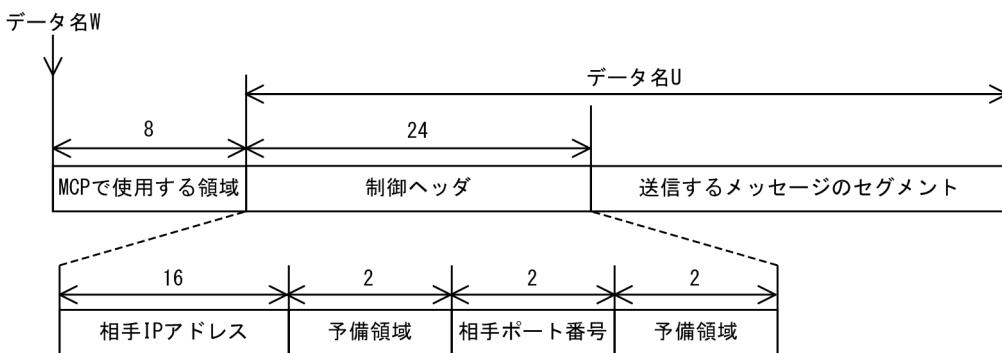
●データ名 W

送信するセグメントの内容を設定した領域を設定します。送信できる最大長は、データ名 U を参照してください。

TCP/IP の場合



UDP の場合



■制御ヘッダ

制御ヘッダ (EEMCPUDPHD) は、<eemcp.h>に次のように定義されています。

```
typedef struct ee_mcp_udp_header {
    unsigned char r_ipaddr[16]; ... 相手IPアドレス
    unsigned char rsv1[2]; ... 予備
    unsigned short r_portno; ... 相手ポート番号
    unsigned char rsv2[4]; ... 予備
} EEMCPUDPHD;
```

■相手IPアドレス

相手IPアドレスはr_ipaddrの下位4バイトに次に示す形式で設定します。

dotted-decimal形式 : 194. 11. 42. 20
 ↓ ↓ ↓ ↓
 16進数形式 (4バイト) : C2 0B 2A 14

変換

...この形式で設定してください。

r_ipaddr[16]

00	00	00	00	00	00	00	00	00	00	00	00	C2	0B	2A	14
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	[14]	[15]

■相手ポート番号

ポート番号は次の形式で設定します。

10進数形式 : 1024
 ↓
 16進数形式 (2バイト) : 0x0400

変換

...この形式 (ビッグエンディアン) で設定してください。

TP1/EE 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。
00001	TP1/EE の環境下にありません。
00004	UOC から発行しているため、この機能は使用できません。
10001	データ名 U に、送信できる最大サイズを超える値を設定しています。

ステータスコード	意味
10002	データ名 U に 0 以下の値を設定しています。
10003	データ名 A, データ名 W, およびデータ名 M5 に設定した値が間違っています。
10004	データ名 H に設定した値が間違っています。
10005	データ名 G に設定した値が間違っています。
10006	データ名 O に設定した値が間違っています。
10007	データ名 M5 に設定した時間が経過しました。
10009	発行条件が不正です。
10011	データ名 P に設定した論理端末名が間違っています。
10020	データ名 P に設定した論理端末名は閉塞中です。
10021	出力メッセージ編集 UOC がエラーリターンしました。
10022	出力メッセージ編集 UOC で障害が発生しました。
10025	システムに対する送信時に障害が発生しました。
10030	メモリ不足が発生しました。

注意事項

- CBLEEMCP('SEND ')関数発行後に、この関数を発行した場合、この関数で送信したメッセージが先に送信されます。
- TP1/FSP を使用する場合、次のステータスコードのときは、「OJ (出力メッセージ)」は取得されません。
00001, 00004, 10001, 10002, 10003, 10004, 10005, 10006, 10011

CBLEEMCP('UTPUT ') - ユーザトレースの取得

機能

ユーザ固有のトレースデータを MCP トレースに出力します。出力したトレースデータは、eemcped コマンドまたは eemcpdump コマンドで編集できます。

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
 02 データ名A PIC X(8) VALUE 'UTPUT' .  
 02 データ名B PIC X(5).  
 02 FILLER PIC X(3).  
 02 データ名C PIC S9(9) COMP VALUE ZERO.  
01 一意名2.  
 02 データ名D PIC 9(9) COMP.  
 02 データ名E PIC 9(9) COMP.  
 02 データ名F PIC X(n).
```

UAP で値を設定する引数

●データ名 A

ユーザ固有のトレースの取得を示す要求コードを「VALUE 'UTPUT△△△'」と設定します。

●データ名 C

0 を設定します。

●データ名 D

トレースを取得するイベントのイベント ID を設定します。0x0700～0x0740 の範囲のイベント ID を設定してください。

●データ名 E

トレースのデータ長を設定します（単位：バイト）。4～32000 の範囲のデータ長を設定してください。

●データ名 F

トレースデータを取得する領域を設定します。トレースデータとして有効なデータは、データ名 E で設定した長さまでです。

TP1/EE 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。
00001	TP1/EE の環境下にありません。
00004	CBLEEMCP('UTPUT')関数を使用できない UOC から発行しているため、この機能は使用できません。
10001	データ名 E に 32000 バイトを超える値を設定しています。

ステータスコード	意味
10002	データ名 E に 4 未満の値を設定しています。
10003	データ名 E 以外のデータ領域に設定した値が間違っています。
10009	発行条件が不正です。
10026	トラブルシュート機能が縮退中です。

CBLEEMCP('UTPUTL ') - ユーザトレースの取得 (データ長拡張用)

機能

データ長拡張時に、ユーザ固有のトレースデータを MCP トレースに出力します。出力したトレースデータは、eemcped コマンドまたは eemcpdump コマンドで編集できます。

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.
  02 データ名A  PIC X(8) VALUE 'UTPUTL '.
  02 データ名B  PIC X(5).
  02 FILLER    PIC X(3).
  02 データ名C  PIC S9(9) COMP VALUE ZERO.
01 一意名2.
  02 データ名D  PIC 9(9) COMP.
  02 データ名E  PIC 9(9) COMP.
  02 データ名F  PIC X(n).
```

UAP で値を設定する領域

●データ名 A

ユーザ固有のトレースの取得を示す要求コードを「VALUE 'UTPUTL△△'」と設定します。

●データ名 C

0 を設定します。

●データ名 D

トレースを取得するイベントのイベント ID を設定します。0x0700～0x0740 の範囲のイベント ID を設定してください。

●データ名 E

トレースのデータ長を設定します（単位：バイト）。4～8388608 の範囲のデータ長を設定してください。

●データ名 F

トレースデータを取得する領域を設定します。トレースデータとして有効なデータは、データ名 E で設定した長さまでです。

TP1/EE 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。

リターン値

ステータスコード	意味
00000	正常に終了しました。
00001	TP1/EE の環境下にありません。
00004	CBLEEMCP('UTPPUTL ')関数を使用できない UOC から発行しているため、この機能は使用できません。
10001	データ名 E に 8388608 バイトを超える値を設定しています。
10002	データ名 E に 4 未満の値を設定しています。
10003	データ名 E 以外のデータ領域に設定した値が間違っています。
10009	発行条件が不正です。
10026	トラブルシュート機能が縮退中です。

注意事項

トレースのデータ長がトレース定義の MCP トレースファイル出力用最大バッファサイズ (eemcptrc 定義コマンドの-b オプションの buffersz オペランドで指定) よりも大きい場合、データの末尾を切り捨てて出力します。切り捨てが起きた場合に取得されるデータのサイズ（単位：バイト）は、次に示す計算式で求められます。

$$((\text{MCP トレースファイル出力用最大バッファサイズ} \times 1024) - 140)$$

5.4 ユーザオウンコーディング (UOC)

UOC は、C 言語で作成します。COBOL 言語では作成できません。ここでは、UOC の使用方法および作成例について説明します。

5.4.1 UOC インタフェース

メッセージの送受信をする 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、UOC といいます。UOC は保護区です。

MCP で使用できる UOC を次に示します。

- 入力セグメント判定 UOC
- 入力メッセージ編集 UOC
- 出力メッセージ編集 UOC
- コネクション確立 UOC

UOC を使用する場合の手順は、次のとおりです。

1. C 言語で UOC を作成する (COBOL 言語では作成できません)。
2. MCP 構成定義に UOC 関数を登録する。
3. UOC 関数のオブジェクトファイルやその他の UAP オブジェクトファイルをリンクエージして UAP 共用ライブラリを作成する。

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 の内で関数の再帰呼び出しはしないでください。

UOC の組み込みは、MCP 構成定義で行います。論理端末ごとに UOC を組み込む場合は、eemcplefunc 定義コマンドの-u オプションの ownlsg, ownlmc, および ownlra オペランドにそれぞれのエントリポインタ名 (C 言語の関数名) を指定します。論理端末ごとに UOC を組み込まない場合は、eemcpfunc 定義コマンドの-u オプションの ownlsg, ownlmc, および ownlra オペランドにそれぞれのエントリポインタ名 (C 言語の関数名) を指定します。

5.4.2 UOC で使用できる関数

UOC では、次の表に示す関数だけを使用できます。表に示す以外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。

表 5-16 UOC で使用できる関数

項目番	関数	
1	TP1/EE および MCP が提供するライブラリ関数	ユーザトレース取得関数 (ee_mcp_utrace_put, および ee_mcp_utrace_put_long) *
2		メッセージログ出力関数 (ee_logprint)
3		現在時刻取得関数 (ee_thd_gettime)
4	標準関数	メモリ管理関数 <ul style="list-style-type: none"> メモリ確保 (例 : malloc) メモリ解放 (例 : free)
5		メモリ操作関数 <ul style="list-style-type: none"> メモリ操作 (例 : memcpy) 文字列操作 (例 : strcpy)
6		時間取得関数

注※

入力メッセージ編集 UOC, および出力メッセージ編集 UOC にだけ使用できます。

5.4.3 UOC の実行タイミング

MCP が起動する UOC は, マルチスレッドでパラレルに実行します。UOC は, パラレルに呼び出されることを想定して, スレッドセーフとなるように作成してください。

ただし, 入力セグメント判定 UOC, 入力メッセージ編集 UOC およびコネクション確立 UOC は論理端末単位ではシリアルに実行しますが, 出力メッセージ編集 UOC はパラレルに実行します。

なお, eemcpfunc 定義コマンドの-c オプションの uocexc オペランドに yes を指定することで, 出力メッセージ編集 UOC も論理端末単位でシリアルに実行できます。その場合は, 入力メッセージ編集 UOC および出力メッセージ編集 UOC が同一論理端末単位でシリアルに実行されます。つまり, 同一排他内で, 入力メッセージ編集 UOC および出力メッセージ編集 UOC が実行されます。

5.4.4 入力セグメント判定 UOC

入力セグメント判定 UOC は, メッセージを受信したとき, そのメッセージでセグメントが完成しているかどうかを判定する UOC です。UOC は, 相手システムからのメッセージを受信するたびに起動します。セグメントが未完成のとき (後続メッセージがある場合) は, 後続メッセージに対して, 監視タイマを設定できます。

ただし, UAP から UAP (タイマトランザクションなど) を起動する場合は, この UOC は起動しません。

入力セグメント判定 UOC は, プロトコル種別が TCP/IP の場合にだけ使用できます。

(1) 入力セグメントの判定

受信したメッセージのセグメントが格納されている受信バッファを引き渡します。UOC では、受信バッファを基にセグメントが完成しているかどうかを判定して、リターン値を設定します。ただし、判定には該当メッセージについての情報を設定する必要があります。そのため、次メッセージが含まれている場合には、次メッセージについての情報も設定してください。なお、セグメントが未完成と判定された場合に後続メッセージを受信した場合は、未完成セグメントのバッファの末尾に後続メッセージを結合して引き渡します。

(2) 後続メッセージ監視タイマの設定

セグメントが未完成の場合、必要に応じて後続メッセージ監視タイマを設定できます。後続メッセージ監視タイマのセットを指示したあとで、後続メッセージのタイムアウトが発生した場合は、コネクションを解放して MCP 後処理トランザクションを起動します。このとき、該当メッセージは破棄されます。また、各パラメタに設定した値が不正の場合、コネクションを解放して MCP 後処理トランザクションを起動します。

MCP 構成定義でコネクション切断抑止を指定した場合は、後続メッセージに対する監視タイマを設定しないでください。

5.4.5 入力セグメント判定 UOC インタフェース

入力セグメント判定 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 <eemcp.h>
int uoc_func(struct eemcp_uoc_segchk *parm)
```

(2) 説明

入力セグメント判定 UOC を呼び出すとき、MCP 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) eemcp_uoc_segchk の内容

```
typedef struct eemcp_uoc_segchk {
    int pro_kind;           ... プロトコル種別
    char le_name[9];        ... 論理端末名称
    char reserve1[7];       ... 予備
    int rcv_prim;          ... 受信サービスプリミティブ
    char *rcv_data_addr;   ... 受信データ先頭アドレス
```

```

int  rcv_data_size;           ... 受信データ有効サイズ
char reserve3[4];           ... 予備
char *uoc_inf_adr;          ... MCP使用領域
int  uoc_inf_size;          ... MCP使用領域
char reserve4[4];           ... 予備
EEMCPSEGUOCPROT *pro_indv_ifa; ... プロトコル個別インターフェース領域アドレス
EEMCPUOCTIMEINF *ptimerinf_adr; ... プロトコルタイマ情報アドレス
int  rtn_detail;            ... 詳細リターンコード
int  reserve2[3];           ... 予備
char *usat_adr;             ... ユーザSATアドレス
EEULONG usat_size;          ... ユーザSATサイズ
EEULONG lebuf_size;          ... MCP使用領域
void *lebuf;                ... MCP使用領域
EEULONG64 usat_large_size;  ... 拡張ユーザSATサイズ
char reserve5[40];           ... 予備
}EEMCPUOCSEGCHK;

```

(b) eemcp_seguoc_prot の内容

```

typedef struct eemcp_seguoc_prot {
    int  rest_data_size;        ... 残っている該当メッセージのサイズ
    int  next_data_size;        ... 次メッセージの有効長
    char *next_data_adr;        ... 次メッセージの先頭アドレス
    int  now_data_size;         ... 該当メッセージの有効長
    int  reserve[3];            ... 予備
}EEMCPSEGUOCPROT;

```

(c) eemcp_uoctimer_inf の内容

```

typedef struct eemcp_uoctimer_inf {
    int timer_code;             ... タイマセット指示
    int timer_value;            ... 後続メッセージ監視タイマ値
    int timer_result;           ... MCP使用領域
    int reserve1[3];            ... 予備
}EEMCPUOCTIMEINF;

```

(d) eemcp_uocbufhead の内容

```

typedef struct eemcp_uocbufhead {
    struct eemcp_uocbufhead *next_bufhead_adr; ... 次バッファ情報のアドレス
    char *buf_adr;               ... 受信データのバッファアドレス
    int  buf_size;                ... 受信データのバッファサイズ
    char reserve[4];              ... 予備
}EEMCPUOCBUFHEAD;

```

(4) MCP が値を設定する項目

(a) eemcp_uoc_segchk

●pro_kind

プロトコル種別として、次の値が設定されます。

EEMCP_UOC_PRO_TCP

TCP/IP プロトコル

●le_name

メッセージを入力した論理端末名が設定されます。

●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。

EEMCP_UOC_RCV_BRD

一方送信メッセージの受信

EEMCP_UOC_RCV REP_SR

同期送受信メッセージの受信

●rcv_data_adr

受信したメッセージの先頭アドレスが設定されます。

●rcv_data_size

受信したメッセージの有効長が設定されます。

●uoc_inf_adr

MCP で使用するパラメタです。

●uoc_inf_size

MCP で使用するパラメタです。

●pro_indv_ifa

プロトコル個別インターフェース領域アドレスです。

●ptimerinf_adr

プロトコルタイマ情報アドレスが設定されます。

●usat_adr

ユーザ SAT エリアのアドレスが設定されます。TP1/EE のサービス定義のメモリ関連定義の system_user_area オペランドに 0 を指定した場合は、値を参照できません。

●usat_size

ユーザ SAT エリアのサイズが設定されます（単位：バイト）。

ユーザ SAT エリアのサイズが 4 ギガバイト以上の場合は、0 が設定されます。0 の場合、ユーザ SAT エリアのサイズは、usat_large_size のエリアを参照してください。なお、ユーザ SAT エリアのサイズが 4 ギガバイト未満の場合でも、usat_large_size のエリアからサイズを取得できます。

●lebuf_size

MCP で使用するパラメタです。

●lebuf

MCP で使用するパラメタです。

●usat_large_size

ユーザ SAT エリアのサイズが設定されます (単位: バイト)。

(b) eemcp_uoctimer_inf (タイマ情報)

●ptimerinf_adr

プロトコルタイマ情報アドレスが設定されます。

●timer_result

MCP で使用するパラメタです。

(c) eemcp_uocbufhead (受信バッファ情報)

メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定) に, eemcp_uoc_segchk の rcv_data_adr が指すパラメタです。受信したメッセージは複数の受信バッファに分割して渡します。

●next_bufhead_adr

次受信バッファ情報のアドレスが設定されます。最終受信バッファの場合, NULL が設定されます。

●buf_adr

受信バッファの先頭アドレスが設定されます。

●buf_size

受信バッファの有効長が設定されます。

(5) ユーザが値を設定する項目

(a) eemcp_uoc_segchk

●rtn_detail ~((-19999~-19000))

詳細リターンコードを設定できます。このコードは UOC が EEMCP_UOC_SGCK_NG をリターンしたときに MCP に返されます。MCP 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) eemcp_seguoc_prot (領域アドレス)

●rest_data_size

残っている該当メッセージのサイズを設定します。

受信したメッセージのセグメントが未完成の場合は、不足分のメッセージサイズを設定します。
EEMCP_UOC_SGCK_OK_NSEG でリターンした場合だけ有効です。

●next_data_size

次のメッセージサイズを設定します。受信したメッセージのセグメントが完成している場合、次のメッセージが含まれているときは、次のメッセージサイズを設定します。EEMCP_UOC_SGCK_OK_LAST でリターンした場合だけ有効です。

●next_data_adr

次のメッセージの先頭アドレスを設定します。next_data_size が 0 でない場合に設定します。

●now_data_size

該当するメッセージのサイズを設定します。next_data_size が 0 でない場合に設定します。

(c) eemcp_uoctimer_inf

●timer_code

後続メッセージ監視タイマを設定します。

受信したメッセージのセグメントが未完成の場合、後続メッセージの受信に対して監視タイマをセットするかどうかを設定します。EEMCP_UOC_SGCK_OK_NSEG でリターンした場合だけ有効です。

EEMCP_TIME_SET

監視タイマをセットします。

EEMCP_TIME_NO_SET

監視タイマをセットしません。

●timer_value ~ <10進数字> ((1~2550)) (単位:秒)

後続メッセージの監視タイマの値を設定します。

この値は、監視タイマをセットする場合に設定します。timer_code で EEMCP_TIME_SET を設定した場合だけ有効になります。監視時間は最大で 1 秒早くタイムアウトする場合があります。

(6) リターン値

入力セグメント判定 UOC は、次のコードでリターンしてください。

リターン値	意味
EEMCP_UOC_SGCK_OK_NSEG	正常リターン (セグメント未完成)
EEMCP_UOC_SGCK_OK_LAST	正常リターン (最終セグメント受信完了)
EEMCP_UOC_SGCK_NG	セグメント判定エラー

■EEMCP_UOC_SGCK_NG がリターンした場合

UOC から EEMCP_UOC_SGCK_NG でリターンした場合、コネクションを解放して MCP 後処理トランザクションを起動します。該当するメッセージは破棄します。

■UOC のパラメタが不正の場合

UOC で設定した値に不正があった場合、MCP はメッセージログを出力して、コネクションを解放します。その後、MCP 後処理トランザクションを起動します。該当するメッセージは破棄します。

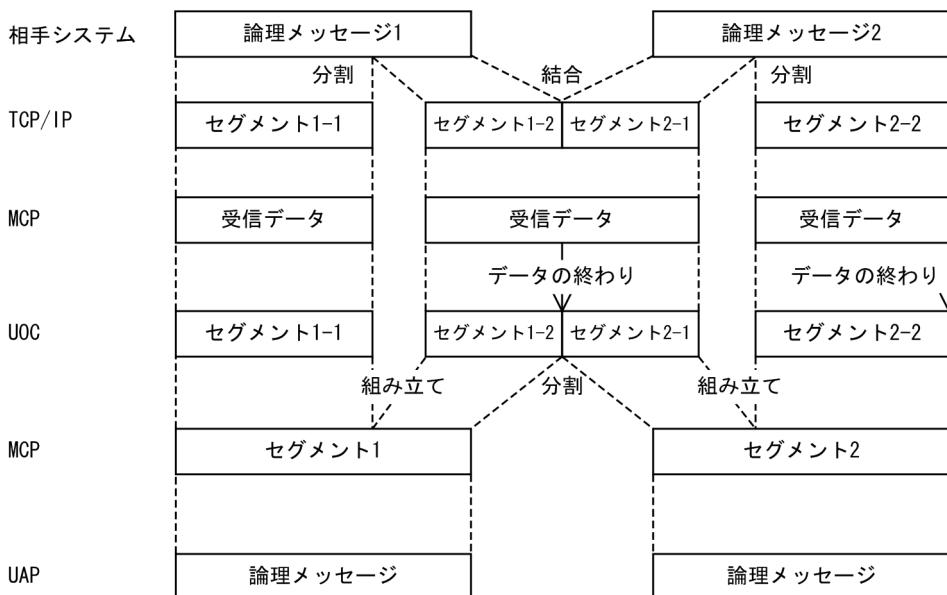
■受信バッファオーバフロー

- メッセージ長エリアに TP1/EE サービス定義で定義した受信バッファのサイズを超える値が設定されていた場合、MCP は受信バッファオーバフローと見なし、コネクションを解放して MCP 後処理トランザクションを起動します。該当するメッセージは破棄します。
- メッセージの最大サイズを拡張していく (eemcpfunc 定義コマンドの msglen オペランドに extend を指定)、かつメッセージ長エリアに設定された値が 8388608 を超える場合、MCP は受信バッファオーバフローと見なします。

(7) セグメントの組み立て

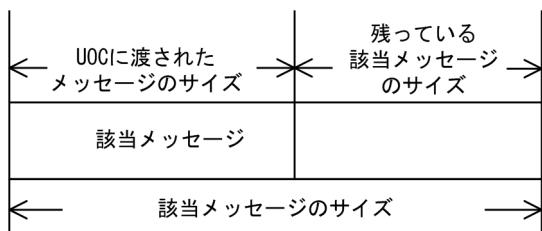
入力セグメント判定 UOC でのセグメントの分割・組み立てを次の図に示します。

図 5-3 入力セグメント判定 UOC でのセグメントの分割・組み立て



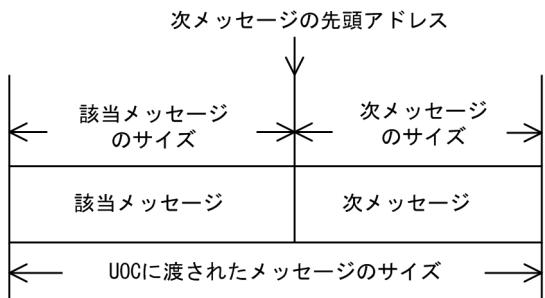
未完成のセグメントを次の図に示します。

図 5-4 未完成のセグメント



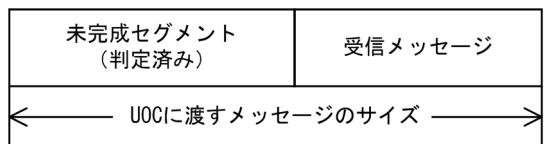
完成済みのセグメント (次のメッセージが含まれている場合) を次の図に示します。

図 5-5 完成済みのセグメント



セグメントが未完成として判定されたあとに、後続メッセージを受信した場合のセグメントを次の図に示します。

図 5-6 セグメントが未完成として判定されたあとに後続メッセージを受信した場合のセグメント



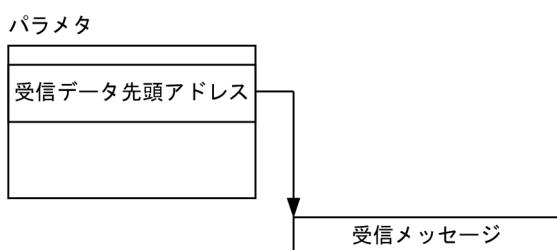
(8) 受信メッセージ

受信メッセージの構成について説明します。

■メッセージの最大サイズを拡張していない場合

メッセージの最大サイズを拡張していない場合の受信メッセージの構成は、次の図に示すとおりです。

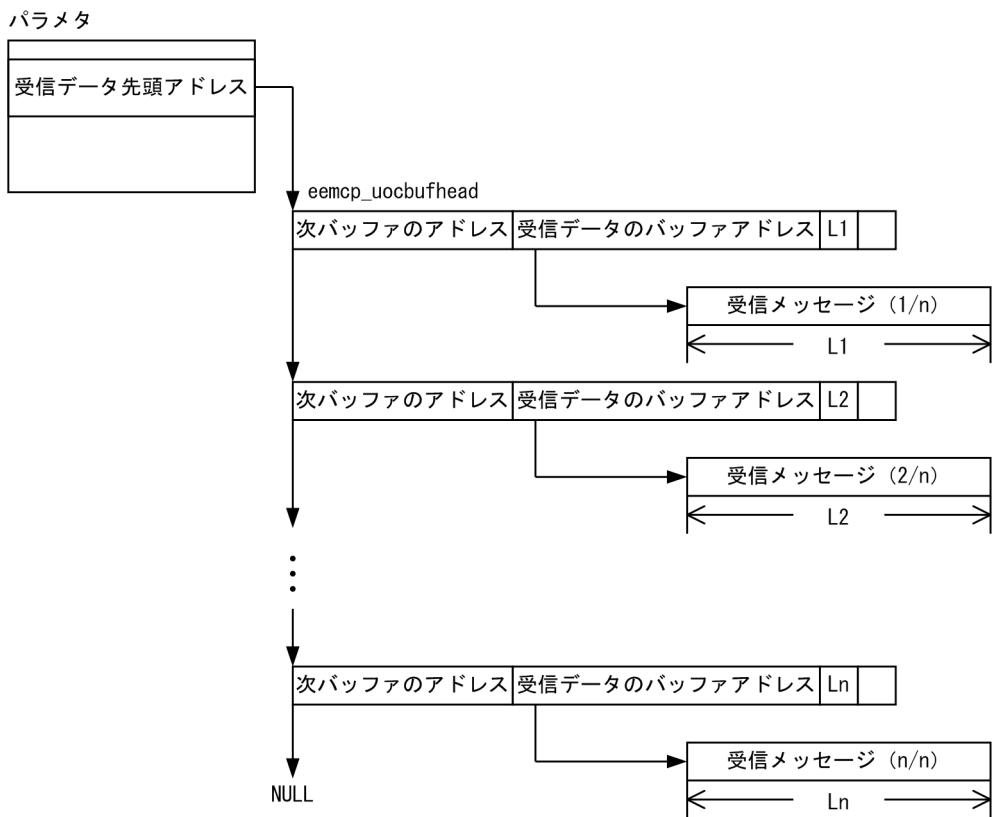
図 5-7 メッセージの最大サイズを拡張していない場合の受信メッセージの構成



■メッセージの最大サイズを拡張している場合

メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定) の受信メッセージの構成は、次の図に示すとおりです。

図 5-8 メッセージの最大サイズを拡張している場合の受信メッセージの構成



(9) 注意事項

入力セグメント判定 UOC が登録されていない場合は、受信した単位（入力バッファに格納された単位）を論理メッセージ（セグメント）として、該当のサービスキューに登録します。

5.4.6 入力メッセージ編集 UOC

入力メッセージ編集 UOC は、一方受信メッセージまたは同期送受信の受信メッセージの編集をする UOC です（論理メッセージをユーザ任意の形式に編集します）。また、受信した論理メッセージを基に、ユーザ任意のサービス名を決定できます。

UOC は、UAP を起動するメッセージのセグメントを受信すると起動します。ただし、UAP から UAP（タイマトランザクションなど）を起動する場合は、この UOC は起動しません。

入力メッセージ編集 UOC はプロトコル種別が TCP/IP、または UDP の場合に使用できます。

(1) 入力メッセージの編集（一方受信メッセージおよび同期送受信の受信メッセージ）

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、入力メッセージを編集できます。

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セグメントを使用できます。どちらを使用するかは、UOC から返されるリターンコードによって選択できます。

また、入力メッセージが不要な場合は、入力メッセージを破棄できます。

入力メッセージは、プロトコル種別が TCP/IP の場合はメッセージの先頭から、UDP の場合は制御ヘッダの先頭から引き渡します。制御ヘッダについては、「[5.2 ee_mcp_send - 非同期一方送信メッセージの送信](#)」の「[UAP で値を設定する引数](#)」を参照してください。

(2) サービス名の決定（一方受信メッセージ）

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サービス名を決定できます。

UOC でサービス名を決定する場合、サービス名の形式は、サービス名格納領域の先頭から、'¥0'の手前までの 1~31 バイトの識別子です。先頭から 32 バイト目までに'¥0'がないときは、サービス名を不正として、エラートランザクション (ERRTRN1) を起動します。ただし、エラートランザクション (ERRTRN1) に通知するサービス名は先頭から 31 文字で 31 文字目に「*」を設定します。

(3) 入力メッセージプライオリティの決定（一方受信メッセージ）

入力メッセージ編集 UOC が登録されている場合、入力メッセージのプライオリティ（優先度）を決定できます。

UOC で優先指定をした場合、該当するサービス（処理キュー）に登録されている通常メッセージよりも、優先的にアプリケーションを起動します。入力メッセージ編集 UOC が登録されていない場合は、すべて通常メッセージ扱いになります。

(4) UOC パラメタ不正の場合の処理

UOC で設定したパラメタに不正があった場合、MCP はメッセージログを出力します。その後、MCP 後処理トランザクションを起動して、論理端末を閉塞します。

(5) UOC リターン処理

UOC から EEMCP_UOC_MSG_NG でエラーリターンした場合、MCP はメッセージログを出力して、MCP 後処理トランザクションを起動します。

(6) メッセージの破棄（一方受信メッセージ）

UOC から EEMCP_UOC_MSG_DESTRUCT でリターンした場合、MCP は入力メッセージを破棄します。サービスは起動しません。

同期送受信の受信メッセージの場合、破棄は指定しないでください。破棄を指定した場合、正常にメッセージを受信したとして扱います (EEMCP_UOC_MSG_OK_RCVと同じ扱いです)。

5.4.7 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 <eemcp.h>
int uoc_func(struct eemcp_uoc_min_n *parm)
```

(2) 説明

入力メッセージ編集 UOC を呼び出すとき、MCP 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) eemcp_uoc_min_n の内容

```
typedef struct eemcp_uoc_min_n {
    int pro_kind;           ... プロトコル種別
    char le_name[9];        ... 論理端末名称
    char priority;          ... プライオリティ
    char reserve1[6];        ... 予備
    int rcv_prim;           ... 受信サービスプリミティブ
    EEMCPUOCBUFLIST *buflist_addr;    ... 受信バッファリストアドレス
    EEMCPUOCBUFLIST *ebuflist_addr;    ... 編集バッファリストアドレス
    char svname[32];         ... サービス名
    char *pro_indv_ifa;      ... MCP使用領域
    int rtn_detail;          ... 詳細リターンコード
    char reserve3[4];         ... 予備
    char *usat_adr;          ... ユーザSATアドレス
    EEULONG usat_size;        ... ユーザSATサイズ
    EEULONG lebuf_size;        ... MCP使用領域
    void *lebuf;              ... MCP使用領域
    EEULONG64 usat_large_size;  ... 拡張ユーザSATサイズ
    char reserve2[40];        ... 予備
} EEMCPUOCMIN;
```

(b) eemcp_uocbuff_list_n の内容

```
typedef struct eemcp_uocbuff_list_n {
    int buf_num;            ... バッファ情報数
    int used_buf_num;        ... 使用バッファ情報数
```

```

char reserve1[8];           ... 予備
EEMCPUOCBUINF buf_array[EEMCP_UOC_BUFF_MAX];
                           ... バッファ情報
}EEMCPUOCBUFLIST;

```

(c) eemcp_uocbufinf_n の内容

```

typedef struct eemcp_uocbufinf_n {
    char *buf_addr;           ... バッファアドレス
    unsigned int buf_size;    ... バッファ最大長
    unsigned int seg_size;    ... バッファ使用長
    char reserve1[8];         ... 予備
    eemcpuoc_w_type buff_id;  ... MCP内部情報1
    char *buff_addr;          ... MCP内部情報2
    char reserve2[8];         ... 予備
}EEMCPUOCBUINF ;

```

(d) eemcp_uocbufhead の内容

```

typedef struct eemcp_uocbufhead {
    struct eemcp_uocbufhead *next_bufhead_addr;
                           ... 次バッファ情報のアドレス
    char *buf_addr;          ... 受信データのバッファアドレス
    int buf_size;            ... 受信データのバッファサイズ
    char reserve[4];          ... 予備
}EEMCPUOCBUFHEAD;

```

(4) MCP が値を設定する項目

(a) eemcp_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。

EEMCP_UOC_PRO_TCP

TCP/IP プロトコル

EEMCP_UOC_PRO_UDP

UDP プロトコル

●le_name

メッセージを入力した論理端末名が設定されます。

●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。

EEMCP_UOC_RCV_BRD

一方送信メッセージの受信

EEMCP_UOC_RCV REP_SR

同期送受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。

メッセージ編集バッファが未定義の場合 (eemcpbuf 定義コマンドの-e オプションを省略した場合), ebuflist_adr には NULL が設定されます。

●svname

eemcple 定義コマンドの-v オプションで指定したサービス名が設定されます。

●pro_indv_ifa

MCP で使用するパラメタです。

●usat_adr

ユーザ SAT エリアのアドレスが設定されます。TP1/EE のサービス定義のメモリ関連定義の system_user_area オペランドに 0 を指定した場合は、値を参照できません。

●usat_size

ユーザ SAT エリアのサイズが設定されます (単位: バイト)。

ユーザ SAT エリアのサイズが 4 ギガバイト以上の場合は、0 が設定されます。0 の場合、ユーザ SAT エリアのサイズは、usat_large_size のエリアを参照してください。なお、ユーザ SAT エリアのサイズが 4 ギガバイト未満の場合でも、usat_large_size のエリアからサイズを取得できます。

●lebuf_size

MCP で使用するパラメタです。

●lebuf

MCP で使用するパラメタです。

●usat_large_size

ユーザ SAT エリアのサイズが設定されます (単位: バイト)。

(b) eemcp_uocbuff_list_n (バッファリスト)

●buf_num

バッファ情報の数として 1 が設定されます。メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定), 1 以上の整数値が設定されます。

●buf_array

バッファ情報の配列が設定されます。バッファ情報が設定される数は一つです。

(c) eemcp_uocbufinf_n (バッファ情報)

●buf_addr

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。メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定), 受信バッファリストでは, eemcp_uocbufhead の buf_addr が示す受信バッファの最大長が設定されます。

●seg_size

バッファの使用長が設定されます。メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定), メッセージの全体長が設定されます。

●buff_id

MCP で使用するパラメタです。

●buff_addr

MCP で使用するパラメタです。

(d) eemcp_uocbufhead (受信バッファ情報)

メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定) に, eemcp_uocbufinf_n (受信用バッファリスト) の buf_addr が指すパラメタです。受信したメッセージは複数の受信バッファに分割して渡します。

●next_bufhead_addr

次受信バッファ情報のアドレスが設定されます。最終受信バッファの場合, NULL が設定されます。

●buf_addr

受信バッファの先頭アドレスが設定されます。

●buf_size

受信バッファの有効長が設定されます。

(5) ユーザが値を設定する項目

(a) eemcp_uoc_min_n

同期送受信メッセージの受信メッセージの場合, priority と svname の指定は無視します。

●priority

入力メッセージの優先度を設定します。

EEMCP_UOC_HI

優先

EEMCP_UOC_LOW (EEMCP_UOC_HI 以外も含む)

通常

●svname

UOC で決定したサービス名を設定します。

サービス名の形式は、サービス名格納領域の先頭から、'¥0'の手前までの 1~31 バイトの識別子で設定してください。

●rtn_detail ~((-19999~-19000))

詳細リターンコードを設定します。

このコードは、UOC が EEMCP_UOC_MSG_NG でリターンした場合に、MCP に渡されます。MCP 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) eemcp_uocbuff_list_n (バッファリスト)

●used_buf_num

使用したバッファ情報の数を設定します。

使用バッファ情報数は、buf_num と同じ値を設定してください。

(c) eemcp_uocbufinf_n (バッファ情報)

●seg_size

バッファの使用長を設定します。メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定)，編集後のメッセージの全体長を設定してください。

(d) eemcp_uocbufhead (受信バッファ情報)

メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定) に、EEMCP_UOC_MSG_OK_RCV でリターンしたときに設定します。

●buf_size

受信バッファの有効長を設定します。UDP プロトコルを使用している場合、先頭の受信バッファには、制御ヘッダ以上のサイズを設定してください。

(6) リターン値

入力メッセージ編集 UOC は次のコードでリターンしてください。

リターン値	意味
EEMCP_UOC_MSG_OK	正常リターン (編集バッファでスケジューリング)
EEMCP_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)
EEMCP_UOC_MSG_DESTRUCT	メッセージ破棄

リターン値	意味
EEMCP_UOC_MSG_NG	メッセージ編集エラー

■メッセージの破棄

UOC から EEMCP_UOC_MSG_DESTRUCT でリターンした場合、MCP は入力メッセージを破棄します。このとき、サービスは起動しません。同期送受信メッセージの受信メッセージについては、正常にメッセージを受信したとして扱います (EEMCP_UOC_MSG_OK_RCV と同じ扱いです)。

■EEMCP_UOC_MSG_NG の場合

UOC から EEMCP_UOC_MSG_NG でエラーリターンした場合、論理端末を閉塞し MCP 後処理トランザクションを起動します。

■UOC パラメタ不正の場合

UOC で設定した値に不正があった場合、MCP はメッセージログ出力後に、論理端末を閉塞して、MCP 後処理トランザクションを起動します。

■受信バッファオーバフロー

- EEMCP_UOC_MSG_OK でリターンした場合に、バッファの使用長 (seg_size) に TP1/EE サービス定義で定義した受信バッファのサイズを超える値が設定されていたとき、MCP は受信バッファオーバフローと見なして、論理端末を閉塞します。その後、MCP 後処理トランザクションを起動します。該当するメッセージは破棄します。
- メッセージの最大サイズを拡張していて (eemcpfunc 定義コマンドの msglen オペランドに extend を指定)、かつバッファの使用長 (seg_size) に設定された値が 8388608 を超える場合、MCP は受信バッファオーバフローと見なします。

■サービス名不正

サービス名の形式は、サービス名格納領域の先頭から、'¥0'の手前までの 1~31 バイトの識別子です。先頭から 32 バイト目までに'¥0'がないときは、サービス名を不正として、エラートランザクション (ERRTRN1) を起動します。

■受信バッファ不足

メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定) に、EEMCP_UOC_MSG_OK でリターンしたとき、受信バッファ不足が発生すると、論理端末を閉塞して MCP 後処理トランザクションを起動します。このとき、該当するメッセージは破棄されます。

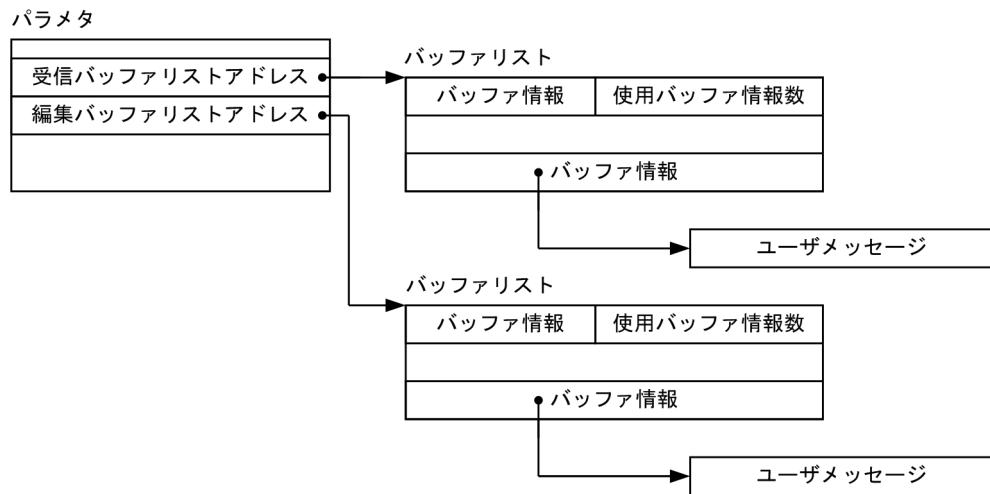
(7) バッファとパラメタの関係

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。

■メッセージの最大サイズを拡張していない場合

メッセージの最大サイズを拡張していない場合の受信メッセージの構成は、次の図に示すとおりです。

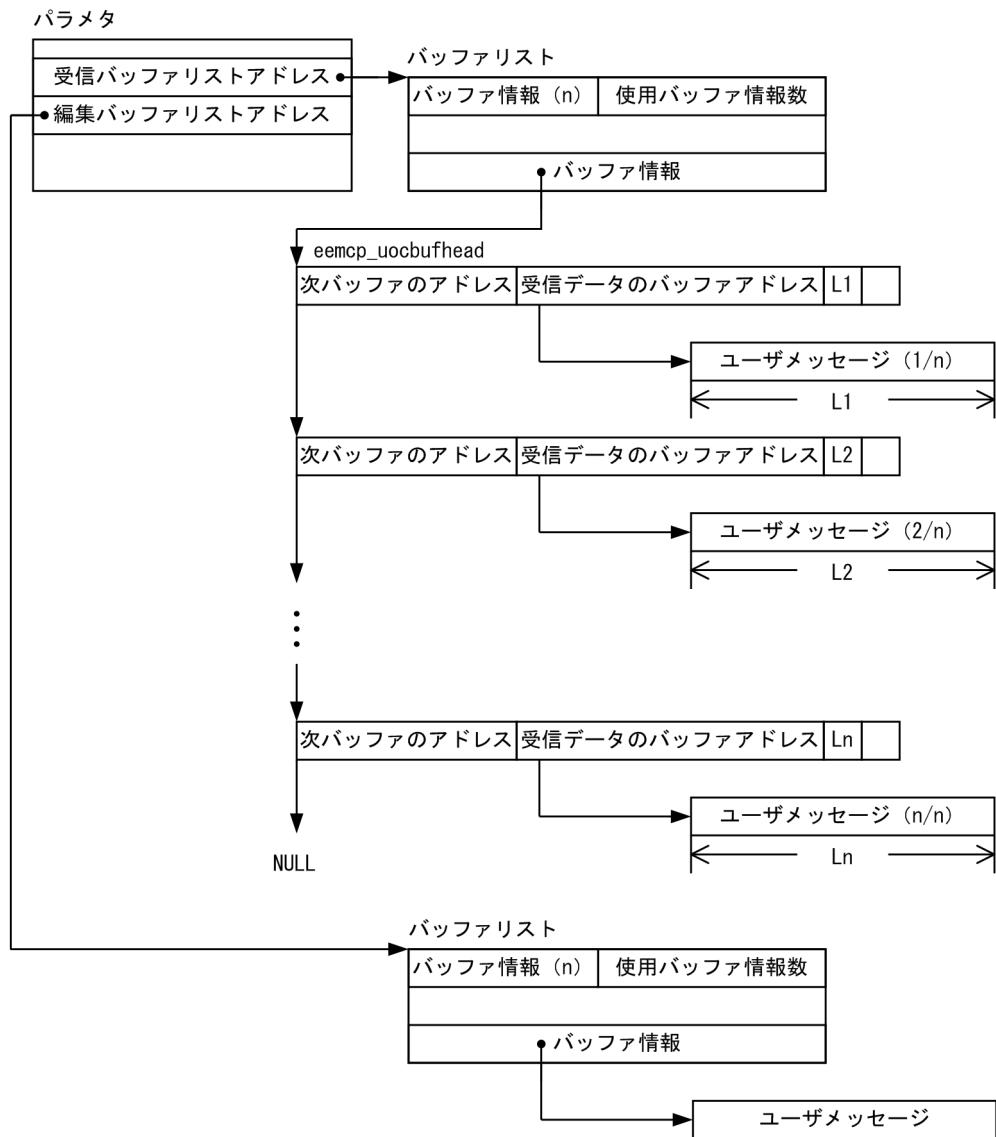
図 5-9 UOC インタフェース用のパラメタとバッファの関係（メッセージの最大サイズを拡張していない場合）



■メッセージの最大サイズを拡張している場合

メッセージの最大サイズを拡張している場合（eemcpfunc 定義コマンドの msglen オペランドに extend を指定）の受信メッセージの構成は、次の図に示すとおりです。

図 5-10 UOC インタフェース用のパラメタとバッファの関係（メッセージの最大サイズを拡張している場合）



5.4.8 出力メッセージ編集 UOC

出力メッセージ編集 UOC は、一方送信メッセージおよび同期送受信の送信メッセージの編集をする UOC です（論理メッセージをユーザ任意の形式に編集します）。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。

出力メッセージ編集 UOC はプロトコル種別が TCP/IP、または UDP の場合に使用できます。

(1) 出力メッセージの編集

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

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

プロトコル種別がTCP/IPの場合に、各パラメタに設定した値が不正なときは、MCP後処理トランザクションを起動して論理端末を閉塞します。

出力メッセージは、プロトコル種別がTCP/IPの場合はメッセージの先頭から、UDPの場合は制御ヘッダの先頭から引き渡します。制御ヘッダについては、「[5.2 ee_mcp_send - 非同期一方送信メッセージの送信](#)」の「UAPで値を設定する引数」を参照してください。

(2) UOCエラーリターン処理

UOCからEEMCP_UOC_MSG_NGでエラーリターンした場合、MCPはメッセージログを出力し、MCP後処理トランザクションを起動して論理端末を閉塞します。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で設定したパラメタに不正があった場合、MCPはメッセージログを出力しMCP後処理トランザクションを起動して論理端末を閉塞します。該当するメッセージは破棄します。

5.4.9 出力メッセージ編集UOCインターフェース

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 <eemcp.h>
int uoc_func(struct eemcp_uoc_mout_n *parm)
```

(2) 説明

出力メッセージ編集UOCを呼び出すとき、MCP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) eemcp_uoc_mout_nの内容

```
typedef struct eemcp_uoc_mout_n {
    int pro_kind;           ... プロトコル種別
    char le_name[9];        ... 論理端末名称
    char reserve1[11];      ... 予備
    EEMCPUOCBUFLIST *buflist_addr; ... 送信バッファリストアドレス
```

```

EEMCPUOCBUFLIST *ebuflist_adr;
int output_no;           ... 編集バッファリストアドレス
char msg_type;          ... メッセージ出力通番
char outputno_flag;     ... メッセージ種別
char resend_flag;       ... メッセージ出力通番有効フラグ
char reserve2[1];       ... MCP使用領域
char *pro_indv_ifa;    ... 予備
char rtn_detail;       ... MCP使用領域
char reserve3[20];      ... 詳細リターンコード
char *lebuf;            ... 予備
EEULONG *usat_adr;     ... ユーザSATアドレス
int usat_size;          ... ユーザSATサイズ
EEULONG lebuf_size;    ... MCP使用領域
void *lebuf;            ... MCP使用領域
EEULONG64 usat_large_size; ... 拡張ユーザSATサイズ
char *uifa_adr;         ... ユーザIFAアドレス
EEULONG64 uifa_large_size; ... ユーザIFAサイズ
char reserve4[32];      ... 予備
}EEMCPUOCMOUT;

```

(b) eemcp_uocbuff_list_n の内容

```

typedef struct eemcp_uocbuff_list_n {
    int buf_num;           ... バッファ情報数
    int used_buf_num;     ... 使用バッファ情報数
    char reserve1[8];      ... 予備
    EEMCPUOCBUFINF buf_array[EEMCP_UOC_BUFF_MAX];
                           ...バッファ情報
}EEMCPUOCBUFLIST;

```

(c) eemcp_uocbufinf_n の内容

```

typedef struct eemcp_uocbufinf_n {
    char *buf_adr;         ... バッファアドレス
    unsigned int buf_size; ... バッファ最大長
    unsigned int seg_size; ... バッファ使用長
    char reserve1[8];      ... 予備
    eemcpuoc_w_type buff_id; ... MCP内部情報1
    char *buff_addr;       ... MCP内部情報2
    char reserve2[8];      ... 予備
}EEMCPUOCBUFINF ;

```

(d) eemcp_uocbufhead の内容

```

typedef struct eemcp_uocbufhead {
    struct eemcp_uocbufhead *next_bufhead_adr;
                           ... 次バッファ情報のアドレス
    char *buf_adr;         ... 送信データのバッファアドレス
    int buf_size;          ... 送信データのバッファサイズ
    char reserve[4];       ... 予備
}EEMCPUOCBUFHEAD;

```

(4) MCP が値を設定する項目

(a) eemcp_uoc_mout_n

●pro_kind

プロトコル種別として、次の値が設定されます。

EEMCP_UOC_PRO_TCP

TCP/IP プロトコル

EEMCP_UOC_PRO_UDP

UDP プロトコル

●le_name

メッセージを入力した論理端末名が設定されます。

●buflist_adr

送信用バッファリストのアドレスが設定されます。

●ebuflist_adr

編集用バッファリストのアドレスが設定されます。

メッセージ編集バッファが未定義の場合 (eemcpbuf 定義コマンドの-e オプションを省略した場合), ebuflist_adr には NULL が設定されます。

●output_no

MCP で使用するパラメタです。

●msg_type

メッセージ種別として、次の値が設定されます。

'n'

非同期一方送信メッセージ

's'

同期一方送信メッセージ

'v'

同期型送受信メッセージの送信メッセージ

●outputno_flag

MCP で使用するパラメタです。

●resend_flag

MCP で使用するパラメタです。

●pro_indv_ifa

MCP で使用するパラメタです。

●usat_adr

ユーザ SAT エリアのアドレスが設定されます。TP1/EE のサービス定義のメモリ関連定義の system_user_area オペランドに 0 を指定した場合は、値を参照できません。

●usat_size

ユーザ SAT エリアのサイズが設定されます（単位：バイト）。

ユーザ SAT エリアのサイズが 4 ギガバイト以上の場合は、0 が設定されます。0 の場合、ユーザ SAT エリアのサイズは、usat_large_size のエリアを参照してください。なお、ユーザ SAT エリアのサイズが 4 ギガバイト未満の場合でも、usat_large_size のエリアからサイズを取得できます。

●lebuf_size

MCP で使用するパラメタです。

●lebuf

MCP で使用するパラメタです。

●usat_large_size

ユーザ SAT エリアのサイズが設定されます（単位：バイト）。

●uifa_adr

ユーザ IFA エリアのアドレスが設定されます。

ユーザ IFA エリアサイズが 0 の場合は、値を参照できません。

●uifa_large_size

ユーザ IFA エリアのサイズが設定されます（単位：バイト）。

(b) eemcp_uocbuff_list_n (バッファリスト)

●buf_num

バッファ情報の数として 1 が設定されます。メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定)、1 以上の整数値が設定されます。

●buf_array

バッファ情報の配列が一つ設定されます。

(c) eemcp_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。メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定)、送信バッファリストでは、eemcp_uocbufhead の buf_adr が示す送信バッファの最大長が設定されます。

●seg_size

送信, または受信用バッファリストの場合だけ, バッファの使用長が設定されます。メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定), メッセージの全体長が設定されます。

●buff_id

MCP で使用するパラメタです。

●buff_addr

MCP で使用するパラメタです。

(d) eemcp_uocbufhead (送信バッファ情報)

メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定) に, eemcp_uocbufinf_n (送信用バッファリスト) の buf_addr が指すパラメタです。送信要求したメッセージは, 複数の送信バッファに分割して渡します。

●next_bufhead_addr

次送信バッファ情報のアドレスが設定されます。最終送信バッファの場合, NULL が設定されます。

●buf_addr

送信バッファの先頭アドレスが設定されます。

●buf_size

送信バッファの有効長が設定されます。

(5) ユーザが値を設定する項目

(a) eemcp_uoc_mout_n

●rtn_detail ~((-19999~-19000))

詳細リターンコードを設定します。

このコードは, UOC が EEMCP_UOC_MSG_NG でリターンしたときに, MCP に渡されます。MCP 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) eemcp_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として, buf_num と同じ値を設定してください。

(c) eemcp_uocbufinf_n (バッファ情報)

●seg_size

バッファの使用長を設定します。メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定), 編集後のメッセージの全体長を設定してください。

(d) eemcp_uocbufhead (送信バッファ情報)

メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定) に, EEMCP_UOC_MSG_OK_SND でリターンしたときに設定します。

●buf_size

送信バッファの有効長を設定します。UDP プロトコルを使用している場合, 先頭の送信バッファには, 制御ヘッダ以上のサイズを設定してください。

(6) リターン値

出力メッセージ編集 UOC は次のコードでリターンしてください。

リターン値	意味
EEMCP_UOC_MSG_OK	正常リターン (編集バッファでスケジューリング)
EEMCP_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)
EEMCP_UOC_MSG_NG	メッセージ編集エラー

■EEMCP_UOC_MSG_NG の場合

UOC から EEMCP_UOC_MSG_NG でエラーリターンした場合, MCP 後処理トランザクションを起動します。

■UOC パラメタ不正の場合

UOC で設定した値に不正があった場合, MCP はメッセージログを出力して, MCP 後処理トランザクションを起動します。

■送信バッファオーバフロー

- EEMCP_UOC_MSG_OK でリターンした場合に, バッファの使用長 (seg_size) に送信できる最大サイズを超える値が設定されていたとき, MCP は送信バッファオーバフローと見なします。その後, 論理端末を閉塞して MCP 後処理トランザクションを起動します。
- メッセージの最大サイズを拡張していて (eemcpfunc 定義コマンドの msglen オペランドに extend を指定), かつバッファの使用長 (seg_size) に設定された値が 8388608 を超える場合, MCP は送信バッファオーバフローと見なします。

■送信バッファ不足

メッセージの最大サイズを拡張している場合 (eemcpfunc 定義コマンドの msglen オペランドに extend を指定) に, EEMCP_UOC_MSG_OK でリターンして, 送信バッファ不足が発生すると, 論理端末を閉塞して MCP 後処理トランザクションを起動します。

(7) バッファとパラメタの関係

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

「5.4.7(7) バッファとパラメタの関係」を参照してください。

5.4.10 コネクション確立 UOC

コネクション確立 UOC は、相手システムからコネクション確立要求を受信するたびに起動し、その確立要求を受け入れるかどうかを判定できます（サーバ型コネクション）。

MCP は、コネクション確立 UOC を呼び出すとき、コネクション確立要求を発行した相手システムのアドレス情報を通知します。コネクション確立 UOC は、通知された相手システムのアドレス情報を基に、コネクション確立要求を受け入れるかどうかを判定します。

コネクション確立要求を受け入れる場合、MCP はコネクション確立の RL トランザクション（MCP 後処理トランザクション）を起動することで、UAP へコネクションの確立を通知します。

コネクション確立要求を拒否する場合はコネクション確立の RL トランザクションを起動しません。この場合、相手システムにはコネクション確立後、MCP から即座にコネクションが解放されたように見えます。

なおコネクションリプレース処理は、UOC でコネクション確立を受け入れた場合に行います。

コネクション確立 UOC は、プロトコル種別が TCP/IP の場合にだけ使用できます。

5.4.11 コネクション確立 UOC インタフェース

(1) 形式

ANSI C、C++の形式

```
#include <eemcp.h>
int uoc_func(struct eemcp_uoc_con_n *parm)
```

(2) 説明

コネクション確立 UOC を呼び出すとき、MCP 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) eemcp_uoc_con_n の内容

```
typedef struct eemcp_uoc_con_n {
    char con_id[9];           ... コネクションID
```

```

char reserve1[7];           ... 予備
unsigned char ripaddr[16];  ... 自システムのIPアドレス
unsigned char oipaddr[16];  ... 相手システムのIPアドレス
unsigned short rportno;    ... 自システムのポート番号
unsigned short oportno;    ... 相手システムのポート番号
EELONG reject_reason;    ... コネクション確立拒否理由
EELONG rtn_detail;       ... 詳細リターンコード
char reserve2[4];          ... 予備
char *usat_adr;           ... ユーザSATアドレス
EEULONG64 usat_large_size; ... ユーザSATサイズ（拡張）
char reserve3[32];         ... 予備
} EEMCPUOCCCONN;

```

(4) MCP が値を設定する項目

(a) eemcp_uoc_con_n

●con_id

確立要求を受け付けたコネクション ID が設定されます。

●ripaddr

確立要求を受け付けた自システムの IP アドレスが設定されます。

IP アドレスは ripaddr の下位 4 バイトに次の形式で設定されます。

dotted-decimal 形式 : 194. 11. 42. 20
 ↓ ↓ ↓ ↓
 C2 0B 2A 14 …この形式で設定してください。
 16進数形式 (4バイト) : C2 0B 2A 14

●oipaddr

確立要求を行った相手システムの IP アドレスが設定されます。

IP アドレスの設定形式は ripaddr を参照してください。

●rportno

確立要求を受け付けた自システムのポート番号が設定されます。

ポート番号はホストバイトオーダー形式の数値で設定されます。

●oportno

確立要求を行った相手システムのポート番号が設定されます。

ポート番号はホストバイトオーダー形式の数値で設定されます。

●usat_adr

ユーザ SAT エリアのアドレスが設定されます。TP1/EE のサービス定義のメモリ関連定義の system_user_area オペランドに 0 を指定した場合は、値を参照できません。該当領域に排他を掛けています。

●usat_large_size

ユーザ SAT エリアのサイズが設定されます (単位: バイト)。

(5) ユーザが値を設定する項目

(a) eemcp_uoc_con_n

●reject_reason

コネクション確立 UOC が EEMCP_UOC_CON_OK_REJECT (コネクション確立拒否) でリターンするときの拒否理由を設定します。値は任意です。この拒否理由はログメッセージ (KFSB81406-I) に出力されます。

●rtn_detail

コネクション確立 UOC が EEMCP_UOC_CON_NG (UOC 異常終了) でリターンするときの詳細リターンコードを設定します。この詳細リターンコードはログメッセージ (KFSB51401-E) に出力されます。値は-19999~-19000 の範囲内で設定します。MCP は、値の範囲をチェックしません。

(6) リターン値

コネクション確立 UOC は次のコードでリターンしてください。

リターン値	意味
EEMCP_UOC_CON_OK_ACCEPT	正常リターン (コネクション確立要求を受け入れ)
EEMCP_UOC_CON_OK_REJECT	正常リターン (コネクション確立要求拒否)
EEMCP_UOC_CON_NG	コネクション確立要求判定エラー

注 上記以外のリターン値を設定した場合、EEMCP_UOC_CON_NG を設定したものとして扱います。

■EEMCP_UOC_CON_OK_REJECT の場合

UOC から EEMCP_UOC_CON_OK_REJECT でリターンした場合、MCP はメッセージログを出力し、コネクションの確立を拒否します。

■EEMCP_UOC_CON_NG の場合

UOC から EEMCP_UOC_CON_NG でリターンした場合、MCP はメッセージログを出力し、コネクションの確立を拒否します。

6

運用

この章では、MCP で取得するトレースファイルの運用、および運用コマンドについて説明します。

6.1 開始と終了

MCP は TP1/EE 上で動作するため、MCP の開始および終了は TP1/EE に依存します。

TP1/EE の開始および終了については、マニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。

6.2 ファイルの運用

ファイルの運用方法について説明します。

■ステータスファイル

TP1/EE で使用するステータスファイル以外に、MCP が使用するステータスファイルはありません。したがって、MCP を使用する場合に、ステータスファイルの設計で考慮する項目はありません。

■TASKTM ファイル

MCP を使用した場合、TASKTM ファイルには、コネクション ID、論理端末名称、および UOC 実行時間が項目として追加されます。MCP 固有の TASKTM ファイルはありません。

■統計情報ファイル

統計情報ファイルには MCP の個別情報はありません。したがって、統計情報ファイルの設計で MCP を使用するときに考慮する項目はありません。

■メモリダンプファイル

メモリダンプファイルには MCP の個別情報はありません。したがって、メモリダンプファイルの設計で MCP を使用するときに考慮する項目はありません。

■メッセージログファイル

MCP は、システムの稼働状況を示すメッセージをログファイルへ出力します。出力先は TP1/EE が出力するメッセージログファイルと同じファイルです。したがって、メッセージログファイルの保存、および障害時の運用については、マニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。

■MCP トレースファイル

MCP は保守情報として送受信イベントの MCP トレースを取得します。また、ユーザ任意のトレースも取得します。MCP トレースの運用については、「6.2.1 MCP トレースファイルの運用」を参照してください。

参考

TASKTM ファイルと MCP トレースファイルを受信 ID で関連づけることで、障害対策に役立てるすることができます。詳細については、マニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。

なお、マニュアル「TP1/Server Base Enterprise Option 使用の手引」では、"回線トレース"を "MCP トレース" に置き換えてお読みください。

6.2.1 MCP トレースファイルの運用

MCP トレースファイルには、送受信メッセージ、状態などの情報がシステムコール単位、または入出力ユーザメッセージ編集 UOC のメッセージ単位で、MCP トレース情報として取得されます。

MCP トレースには、次に示す 3 種類があります。

■メッセージ送受信イベントトレース

MCP トレース送受信イベント情報（MCP 保守用トレース）です。

eemcptrc 定義コマンドの-m オプションの filter オペランドの指定によって取得内容が決まります。

1 回の送受信メッセージをトレースに取得する量は、eemcptrc 定義コマンドの-m オプションの指定によって決まります。

詳細は、「[\(5\) MCP トレース情報の取得量の変更](#)」を参照してください。

■ユーザトレース

入出力メッセージなどのユーザ情報トレースです。ee_mcp_utrace_put 関数、または ee_mcp_utrace_put_long 関数で要求したデータです。

■メッセージ送信結果トレース

出力メッセージ編集 UOC でユーザトレースを取得した場合、メッセージ送信の成否に関係なく取得する結果トレースです。送受信メッセージの内容は取得しません。出力メッセージ編集 UOC でユーザトレースを取得したあとに、UOC パラメタ不正などで実際に送信しなかった場合でも、保守情報として送信結果トレースを取得します。このトレースは、eemcptrc 定義の-s オプションの sendresult オペランドに yes を指定した場合にだけ取得します。

通常、これらの MCP トレース情報は、イベント発生時の要求ごとに取得されます。

なお、MCP トレースは、TP1/EE システム定義の mcp_use オペランドに Y を指定した場合に機能します。

MCP トレースファイルに取得した MCP トレース情報は、eemcped コマンドを実行して編集し、標準出力、または CSV 形式のファイルとして出力します。また、eemcpdump コマンドを実行すると、コアファイルまたはメモリダンプファイルを入力情報として、MCP トレースファイルの編集結果を標準出力、または CSV 形式のファイルとして出力することもできます。

PP トレーススレッドの異常や I/O が間延びしてバッファが不足した場合、eemcped コマンドでは MCP トレース情報を出力できません。ただし、eemcpfput コマンドを実行すると強制出力できます。

(1) MCP トレースファイルの容量設定

MCP トレースファイルは、eemcptrc 定義コマンドの-f オプションの filesz オペランドに指定したサイズで、TP1/EE のプロセスごとに UNIX ファイルとして作成されます。プロセスごとに、eemcptrc 定義コマンドの-f オプションの fileno オペランドに指定した数の MCP トレースファイルが作成され、ラウンドロビン方式で使用されます。MCP トレースファイルのサイズおよびファイル数は、ノードで稼働する TP1/EE のプロセス数やノードのリソース容量などを考慮して、指定してください。

全体の MCP トレースファイルサイズの見積もり式を次に示します。

$$\begin{aligned} \text{全体のMCPトレースファイルサイズ (T)} \\ = & ((L + 40) \times x) + ((M + 40) \times y) \\ & + ((N + 40) \times z) + (40 \times u) \\ & + (↑ (((L + 40) \times x) + ((M + 40) \times y)) \end{aligned}$$

$$+ ((N + 40) \times z) + (40 \times u)) \div b) \uparrow \times 96 \\ + F$$

見積もり式の変数または定数を次の表に示します。

変数または定数	意味
L	メッセージ受信単位のバイト数
M	メッセージ送信単位のバイト数
N	ee_mcp_utrace_put 関数で指定するトレースのデータ長
40	トレースヘッダバイト数
96	1 ブロック単位に付加する情報のバイト数
F※	1 ファイル単位に付加する情報のバイト数
x	受信メッセージ数
y	送信メッセージ数
z	ee_mcp_utrace_put 関数の発行回数
u	出力メッセージ編集 UOC 内で ee_mcp_utrace_put 関数を発行して送信した送信回数
b	eemcptrc 定義コマンドの-b オプションの buffersz オペランドに指定した値

注※

$$F = 128 + 34 \times (\text{サービス数}) + 16 \times (\text{論理端末数}) + 16 \times (\text{コネクションID数}) + 256$$

(2) MCP トレースファイルの作成

MCP トレースファイルは、TP1/EE の起動時、\$DCDIR/spool/dceinf/mcp ディレクトリ下に MCP が作成します。したがって、OpenTP1 管理者は MCP トレースファイルを作成する必要はありません。MCP が作成するファイル名は、次に示す規則で付けられます。

サービスグループ名mcpXXX (XXXは、3けたのファイル通番)

(3) MCP トレースファイルの削除

MCP の起動時に、前回の MCP 稼働時の MCP トレースファイルが存在する場合は、最も古い MCP トレースファイルに MCP トレース情報を上書きします。したがって、OpenTP1 管理者は MCP トレースファイルを削除する必要はありません。

(4) MCP トレースファイルの保存

出力中の MCP トレースファイルのサイズが、eemcptrc 定義コマンドの-f オプションの filesz オペランドに指定した値を超えた場合、出力先が新しい MCP トレースファイルに切り替わります。この場合、出力先が新しい MCP トレースファイルに切り替わることを通知する KFSB85400-I メッセージが出力されま

す。このメッセージが出力されるタイミングで、MCP トレースファイルのバックアップを取得して、同一論理ボリューム上でファイル移動 (mv コマンド) することをお勧めします。

MCP トレースファイル数が、eemcptrc 定義コマンドの-f オプションの fileno オペランドに指定した値を超えた場合、最も古い MCP トレースファイルに MCP トレース情報を上書きします。古い MCP トレースファイルの MCP トレース情報を保存したい場合は、MCP が上書きする前に、OpenTP1 管理者がバックアップしてください。

使用中の MCP トレースファイルをバックアップした場合、編集結果が正しく表示できないことがあります。そのため、MCP トレースファイルを切り替えることを通知するメッセージが出力されてから、バックアップを取得してください。また、使用中の MCP トレースファイルは、移動または削除しないでください。

(5) MCP トレース情報の取得量の変更

送受信データごとに MCP トレースを取得する場合、I/O 処理の増加によって性能が低下したり、重要な MCP トレースが欠落したりすることがあります。これを防止するために、eemcptrc 定義コマンドの-m オプションの filter オペランドによって、送受信時の MCP トレース情報の取得内容を変更できます。

それぞれのオペランドの指定値別に、取得内容および取得サイズの差異について説明します。

00000000 (デフォルト)

すべての送受信イベントを MCP トレース情報に取得します。送受信データは、1 回の送受信イベントごとに最大で 32000 バイトを取得します。

00000010

すべての送受信イベントを MCP トレース情報に取得します。送受信データは、1 回の送受信イベントごとに eemcptrc 定義コマンドの-m オプションの msgsize オペランドに指定された最大サイズまでを取得します。msgsize オペランドに 0 を指定した場合は、データ長 0 として取得します。

6.3 運用コマンド

ここでは、運用コマンドについて説明します。

運用コマンドの記述形式

運用コマンドの記述形式については、マニュアル「TP1/Server Base Enterprise Option 使用の手引」を参照してください。

運用コマンドの一覧

MCP の運用コマンド一覧を次の表に示します。

表 6-1 MCP で使用する運用コマンドの一覧

項目番	機能	運用コマンド	オフライン	オンライン	API
1	コネクションの確立	eemcpactn	×	○	○
2	相手システムを指定したコネクションの確立	eemcpactn to	×	○	○
3	論理端末の閉塞解除	eemcpactle	×	○	○
4	コネクションの解放	eemcpdctn	×	○	○
5	論理端末の閉塞	eemcpdctle	×	○	○
6	MCP トレースの編集、出力 (コアファイルまたはメモリダンプファイル入力)	eemcpdump	○	○	○
7	MCP トレースの編集、出力	eemcped	○	○	○
8	MCP トレースの強制ファイル出力	eemcpfput	×	○	○
9	コネクション状態表示	eemcplscn	×	○	○
10	論理端末の状態表示	eemcplsle	×	○	○

(凡例)

オフライン：TP1/EE が起動していない状態での実行を示します。

オンライン：TP1/EE が起動している状態での実行を示します。

API : ee_adm_call_command 関数での実行を示します。

○：実行できます。

×：実行できません。

eemcpactcn (コネクションの確立)

機能

コネクションを確立します。

コネクションの確立は非同期で実行されるため、このコマンドが正常終了した場合でも、コネクションの確立が失敗していることがあります。コネクションの確立結果は、MCP 後処理トランザクションによって通知されます。

このコマンドの-c オプションに指定したコネクション ID 中に、相手システム (eemcpcn 定義コマンドの-o オプション) を省略した eemcpcn 定義コマンドのコネクション ID が含まれていた場合は、指定した全コネクション ID のコネクション確立処理は実行されません。

コマンドの形式

```
eemcpactcn -g サービスグループ名  
-c コネクションID
```

オプションの説明

●-g サービスグループ名

～〈1～31 文字の識別子〉

処理対象のサービスグループ名を指定します。

●-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 ID を指定します。

クライアント型のコネクション ID (eemcpcn 定義コマンドの-c オプションの指定値) を指定してください。クライアント型のコネクションとは、eemcpcn 定義コマンドの-y オプションの mode オペランドに client を指定したコネクションのことです。また、eemcpcn 定義コマンドの相手システム (-o オプション) を指定したコネクション ID を指定してください。

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

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

*

すべてのコネクションを確立します。

先行文字列*

先行文字列で始まるすべてのコネクションを確立します。

<複数指定の例> cnn1, cnn2, cnn3 を指定する場合

-c "cnn1 cnn2 cnn3"

<一括指定の例> cnn で始まるすべてのコネクションを指定する場合

-c "cnn*"

出力メッセージ

メッセージ ID	内容	出力先
KFSB91401-E	領域の確保に失敗しました。	標準エラー出力
KFSB91402-E	通信障害が発生しました。	標準エラー出力
KFSB91403-E	コマンドの形式が不正です。	標準エラー出力
KFSB91404-E	フラグ引数が不正です。	標準エラー出力
KFSB91406-E	内部矛盾が発生しました。	標準エラー出力
KFSB91420-I	ヘルプメッセージ	標準出力
KFSB91421-I	コネクションの確立を受け付けました。	標準出力
KFSB91422-E	コネクションの確立の受け付けに失敗しました。	標準エラー出力
KFSB91423-E	コネクションの確立に一部失敗しました。	標準エラー出力

eemcpactcno (相手システムを指定したコネクションの確立)

機能

コネクションを確立する相手システムを指定し、コネクションを確立します。

コネクションの確立は非同期で行われます。そのため、このコマンドが正常終了した場合でも、コネクションの確立が失敗していることがあります。コネクションの確立結果は MCP 後処理トランザクションによって通知されます。

eemcpn 定義の-o オプション指定値を無視します。

このコマンドで指定したコネクション ID のコネクションを切断後に、同じコネクション ID に対し eemcpactcn コマンド/ee_mcp_actcn 関数でコネクション確立した場合、eemcpn オペランドの相手システム (-o 指定値) に対しコネクションを確立します。

コマンドの形式

```
eemcpactcnto -g サービスグループ名 -c コネクションID  
      {-i 相手システムのIPアドレス | -n 相手システムのホスト名}  
      -p 相手システムのポート番号
```

オプションの説明

●-g サービスグループ名

～〈1～31文字の識別子〉

処理対象のサービスグループ名を指定します。

●-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 ID を指定します。

クライアント型のコネクション ID (eemcpn 定義コマンドの-c オプションの指定値) を指定してください。クライアント型のコネクションとは, eemcpn 定義コマンドの-y オプションの mode オペランドに client を指定したコネクションのことです。eemcpn 定義コマンドの相手システム (-o オプション) を指定したコネクション ID を指定してください。

●-i 相手システムの IP アドレス

～〈7～15文字の符号無し数字〉

相手システムの IP アドレスを指定します。

IP アドレスを指定する場合は, xxx.xxx.xxx.xxx の形式で指定してください。xxx は, 0～255 (10進数) で指定してください。ただし, 0.0.0.0, 255.255.255.255 の指定は不可とします。

●-n 相手システムのホスト名

～〈1～255文字のホスト名〉

相手システムのホスト名を指定します。

ホスト名は, /etc/hosts ファイル, DNS などで, IP アドレスとマッピングできるように指定する必要があります。

●-p 相手システムのポート番号

～〈1～65535〉

相手システムのポート番号を指定します。

出力メッセージ

メッセージID	内容	出力先
KFSB91401-E	領域の確保に失敗しました。	標準エラー出力

メッセージID	内容	出力先
KFSB91402-E	通信障害が発生しました。	標準エラー出力
KFSB91403-E	コマンドの形式が不正です。	標準エラー出力
KFSB91404-E	フラグ引数が不正です。	標準エラー出力
KFSB91406-E	内部矛盾が発生しました。	標準エラー出力
KFSB91471-I	ヘルプメッセージ	標準出力
KFSB91421-I	コネクションの確立を受け付けました。	標準出力
KFSB91422-E	コネクションの確立の受け付けに失敗しました。	標準エラー出力

eemcpactle (論理端末の閉塞解除)

機能

論理端末を閉塞解除します。

コマンドの形式

```
eemcpactle -g サービスグループ名
          -l 論理端末名
```

オプションの説明

●-g サービスグループ名

～ <1～31文字の識別子>

処理対象のサービスグループ名を指定します。

●-l 論理端末名

～ <1～8文字の識別子>

閉塞解除する論理端末名を指定します。eemcpactle 定義コマンドの-l オプションで指定した論理端末名を指定してください。

論理端末名は、一度に8個まで指定できます。複数の論理端末名を指定するときは、引用符「"」で囲んで、論理端末名と論理端末名との間を空白で区切れます。論理端末名は重複して指定できません。

また、論理端末名は、「*」を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外の論理端末名とは、混在して指定できません。一括指定をするときは、引用符「"」で囲んで指定します。

*

すべての論理端末を閉塞解除します。

先行文字列*

先行文字列で始まるすべての論理端末を閉塞解除します。

<複数指定の例> len1, len2, len3 を指定する場合

-l "len1 len2 len3"

<一括指定の例> len で始まるすべての論理端末を指定する場合

-l "len*"

出力メッセージ

メッセージ ID	内容	出力先
KFSB91401-E	領域の確保に失敗しました。	標準エラー出力
KFSB91402-E	通信障害が発生しました。	標準エラー出力
KFSB91403-E	コマンドの形式が不正です。	標準エラー出力
KFSB91404-E	フラグ引数が不正です。	標準エラー出力
KFSB91406-E	内部矛盾が発生しました。	標準エラー出力
KFSB91445-I	ヘルプメッセージ	標準出力
KFSB91446-I	論理端末の閉塞解除を受け付けました。	標準出力
KFSB91447-E	論理端末の閉塞解除の受け付けに失敗しました。	標準エラー出力
KFSB91448-E	論理端末の閉塞解除に一部失敗しました。	標準エラー出力

eemcpdctcn (コネクションの解放)

機能

コネクションを解放します。該当コネクションが仕掛け中の場合でも、強制的に解放します。

コネクション解放後、MCP 後処理トランザクションを起動します。

コマンドの形式

```
eemcpdctcn -g サービスグループ名  
-c コネクションID [-f]
```

オプションの説明

●-g サービスグループ名

～〈1～31 文字の識別子〉

処理対象のサービスグループ名を指定します。

●-c コネクション ID

～〈1～8 文字の識別子〉

コネクション解放を行うコネクション ID を指定します。

eemcpcn 定義コマンドの-c オプションに指定したコネクション ID を指定してください。

コネクション ID は、一度に 8 個まで指定できます。複数の論理端末名を指定するときは、引用符「"」で囲んで、コネクション ID とコネクション ID との間を空白で区切ります。コネクション ID は重複して指定できません。

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

*

すべてのコネクションを解放します。

先行文字列*

先行文字列で始まるすべてのコネクションを解放します。

＜複数指定の例＞cnn1, cnn2, cnn3 を指定する場合

-c "cnn1 cnn2 cnn3"

＜一括指定の例＞cnn で始まるすべてのコネクションを指定する場合

-c "cnn*"

●-f

eemcpcn 定義コマンドの-f オプションの cnrelease オペランドに rst を指定した場合、RST パケットを送信してコネクションを強制解放し、MCP 後処理トランザクションを起動します。このオプションを指定していない場合は、eemcpcn 定義コマンドの-f オプションの cnrelease オペランドに rst を指定した場合でも、FIN パケットを送信してコネクションを解放します。

出力メッセージ

メッセージ ID	内容	出力先
KFSB91401-E	領域の確保に失敗しました。	標準エラー出力
KFSB91402-E	通信障害が発生しました。	標準エラー出力
KFSB91403-E	コマンドの形式が不正です。	標準エラー出力
KFSB91404-E	フラグ引数が不正です。	標準エラー出力
KFSB91406-E	内部矛盾が発生しました。	標準エラー出力
KFSB91425-I	ヘルプメッセージ	標準出力
KFSB91426-I	コネクションの解放を受け付けました。	標準出力

メッセージID	内容	出力先
KFSB91427-E	コネクションの解放の受け付けに失敗しました。	標準エラー出力

eemcpdctle (論理端末の閉塞)

機能

論理端末を閉塞します。

コマンドの形式

```
eemcpdctle -g サービスグループ名
             -l 論理端末名
```

オプションの説明

●-g サービスグループ名

～〈1～31文字の識別子〉

処理対象のサービスグループ名を指定します。

●-l 論理端末名

～〈1～8文字の識別子〉

閉塞する論理端末名を指定します。eemcpdctle 定義コマンドの-l オプションで指定した論理端末名を指定してください。

論理端末名は、一度に8個まで指定できます。複数の論理端末名を指定するときは、引用符「」で囲んで、論理端末名と論理端末名との間を空白で区切れます。論理端末名は重複して指定できません。

また論理端末名は、「*」を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外の論理端末名は、混在して指定できません。一括指定をするときは、引用符「」で囲んで指定します。

*

すべての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

メッセージ ID	内容	出力先
KFSB91401-E	領域の確保に失敗しました。	標準エラー出力
KFSB91402-E	通信障害が発生しました。	標準エラー出力
KFSB91403-E	コマンドの形式が不正です。	標準エラー出力
KFSB91404-E	フラグ引数が不正です。	標準エラー出力
KFSB91406-E	内部矛盾が発生しました。	標準エラー出力
KFSB91440-I	ヘルプメッセージ	標準出力
KFSB91441-I	論理端末の閉塞を受け付けました。	標準出力
KFSB91422-E	論理端末の閉塞の受け付けに失敗しました。	標準エラー出力

eemcpdump (MCP トレースの出力)

機能

指定されたコアファイルまたはメモリダンプファイルから MCP トレース情報を編集し、標準出力または CSV 形式でファイルに出力します。

TP1/EE プロセスがダウンした場合や特定のスレッドがダウンした場合に、MCP 通信での障害情報を特定のトランザクションまたは IP アドレスなどで編集し、障害を調査するために使用します。

コマンドの形式

```
eemcpdump [-r ランID]
[-s usr | lin]
[-x 中央処理通番の下限値, 中央処理通番の上限値]
[-l 論理端末名 [, 論理端末名…] ]
[-i IPアドレス [, IPアドレス…] ]
[-c CSV出力ファイル名 [-n ファイル出力行] ] ファイル名
```

オプションの説明

●-r ラン ID

指定されたラン ID を持つトレース情報だけ編集、出力します。

●-s usr | lin

～《usr》

usr：ユーザトレース情報だけを編集、出力します。

lin : MCP トレース情報だけを編集、出力します。

●-x 中央処理通番の下限値、中央処理通番の上限値

～ <1～8 文字の 16 進数> ((0～ffffffff))

指定されたカレント中央処理通番の範囲内の情報だけを編集、出力します。

-x オプションを指定する場合、上限値および下限値を必ず指定してください。

中央処理通番を指定する場合、最大値 (4294967295) を考慮して指定してください。

●-l 論理端末名

～ <1～8 文字の識別子>

指定された論理端末名を持つ情報だけ編集、出力します。eemcple 定義コマンドの-l オプションで指定した論理端末名を指定してください。論理端末名は、最大 10 個指定できます。

●-i IP アドレス

～ <7～15 文字の数字>

IP アドレスを指定する場合は、nnn.nnn.nnn.nnn の形式で指定します。nnn は、0～255 (10 進数) で指定します。IP アドレスは、最大 10 個指定できます。

●-c CSV 出力ファイル名

～ <パス名>

編集結果を指定したファイルに CSV 形式で出力します。標準出力には出力しません。CSV ファイルに出力する場合、コンマで区切って出力します。10 進数以外の場合、引用符「"」で囲んで出力します。

-n オプションを指定した場合は、パス名中のファイル名は、245 文字以下としてください。-n オプション指定時にファイル名が 246 文字以上の場合はオプションエラーで、このコマンドが異常終了します。

●-n ファイル出力行

((10000～100000))

-c オプションに指定した csv ファイルを分割して出力する場合に 1 ファイルに出力する行数を指定します。見出し行を含めて-n オプションに指定した行数を出力したときに、出力するファイルを切り替えます。切り替えた先のファイルには見出し行から出力します。

分割したファイルには、10 進数の文字列をファイル名の末尾に付与します。付与する文字列は 1 から 9999 ファイルまでは「0001」からの連番で左に 0 を埋めた 4 けたの文字列とします。10000 ファイルを超えた場合は左に 0 埋めをしません。ファイル名に「.」を含む場合は、最後の「.」の前に付与します。

ファイル名に文字列を付与する例を示します。

<例 1> -c に xxx と指定した場合

```
xxx0001  
xxx0002  
:
```

<例 2> -c に xxx.csv と指定した場合

```
xxx0001.csv  
xxx0002.csv  
:
```

このオプションを省略した場合は、ファイルを分割しないで 1 ファイルに出力します。

コマンド引数

●ファイル名

～ <パス名>

入力元のコアファイルまたはメモリダンプファイル名を指定します。

指定できるオプションの組み合わせを次の表に示します。

表 6-2 eemcpdump コマンドのオプションの組み合わせ

オプション		-r	-s		-x	-l	-i	-c	-n	ファイル指定
			usr	lin						
-r		—	○	○	○	○	○	○	○	○
-s	usr	○	—	×	○	○	×	○	○	○
	lin	○	×	—	×	×	○	○	○	○
-x		○	○	×	—	○	×	○	○	○
-l		○	○	×	○	—	×	○	○	○
-i		○	×	○	×	×	—	○	○	○
-c		○	○	○	○	○	○	—	○	○
-n		○	○	○	○	○	○	○	—	○
ファイル指定		○	○	○	○	○	○	○	○	—

(凡例)

○：指定できます。

×：指定できません。

—：該当しません。

出力形式

コマンド実行時の出力形式について説明します。

■-s に lin を指定した場合

■ -s に usr を指定した場合

出力形式中の、右端の数字の意味を次に示します。

- 1：改行キーを含まないで、1行で出力します。
- 2：レコード数分出力します。
- 3：複数行を改行して出力します。

出力形式中の、各変数の意味を次に示します。

変数	意味
a1-a1-a1	コマンドを実行した TP1/EE のバージョン
a2-a2-a2	コマンドを実行した MCP のバージョン
bb....bb	このコマンドの引数
cc....cc	入力ファイル名

変数	意味
e1-e1-e1	ファイルを作成した TP1/EE のバージョン
e2-e2-e2	ファイルを作成した MCP のバージョン
e3-e3-e3	ファイルを作成した TP1/FSP のバージョン TP1/FSP を使用した場合だけ出力します。
e3	MCP トレース情報取得タイプ
ff....ff	サービスグループ名 (31 文字以内)
gggggggg	TP1/EE のラン ID (16 進数 8 けた)
hhhh/hh/hh hh:hh:hh hhh.hhh	MCP トレースを取得した時刻 「年/月/日△時間：分：秒 ミリ秒. マイクロ秒」の形式で表示します。
ii....ii	イベント名 (15 文字以内) イベントの前後で情報を取得するため、イベント名の後ろに、「前」または「後」が表示されます。また、ee_mcp_utrace_put 関数、または ee_mcp_utrace_put_long 関数で指定した ID を 16 進数で表示します。
jjjjjjjjjj	データ長 (10 進数 10 けた)
oooo/oo/oo oo:oo:oo 000.000	実行時刻 「年/月/日△時間：分：秒 ミリ秒. マイクロ秒」の形式で表示します。
kkkkkkkk	該当レコードを取得した中央処理通番 (16 進数 8 けた)
kkkkkkkk1	該当レコードを取得した受信 ID (16 進数 8 けた)
lll	取得した機能種別 (3 文字) <ul style="list-style-type: none">usr : ユーザトレース API、または MCP で取得lin : MCP で取得
mm....mm	論理端末名 (1~8 文字の識別子)
nn....nn	コネクション ID (1~8 文字の識別子)
ppp.ppp.ppp.ppp	受信/送信先の IP アドレス (10 進数 15 けた以内)
qq....qq	受信/送信先のポート番号 (10 進数 5 けた以内)
rr....rr	サービス名 (1~31 文字の識別子)
0xss....ss	送信 ID (16 進数 16 けた)
0xtt....tt	リターンコード (16 進数 4 けた以内)
tt2....tt2	リターンコード (10 進数 5 けた以内)
uu....uu	実際に送受信したデータ長
vv vv vv vv vv vv vv vvvvvvvvvvvvvvvvvvv	次に示すデータです。 -s オプションに usr を指定した場合 ユーザデータ、または MCP で取得したデータ

6. 運用

■-n オプションを指定した場合の標準出力

日本語と英字共通

EARLIEST TIME NAME	- LATEST TIME	EARLIEST CPN - LATEST CPN FILE
xxxx/xx/xx xx:xx:xx xxx.xxx - yyyy/yy/yy yy:yy:yy yyyy.yyy	nnnnnnnn	mmmmmmmm zzz00
01		
xxxx/xx/xx xx:xx:xx xxx.xxx - yyyy/yy/yy yy:yy:yy yyyy.yyy	nnnnnnnn	mmmmmmmm zzz00
02		
:	:	:

項目番	表示名称	出力形式
1	xxxx/xx/xx xx:xx:xx xxx.xxx	出力した MCP トレース情報の中でいちばん古い起動時刻 出力対象の MCP トレース情報がない場合は 「****/**/** *.*.*.* ***.*.*」 を出力します。
2	yyyy/yy/yy yy:yy:yy yyy.yyy	出力した MCP トレース情報の中でいちばん新しい起動時刻 出力対象の MCP トレース情報がない場合は 「****/**/** *.*.*.* ***.*.*」 を出力します。
3	nnnnnnnn	出力した MCP トレース情報の中でいちばん値の小さい中央処理通番。16 進数 8 けた※ 出力対象の MCP トレース情報がない場合は 「*****」 を出力します。 出力対象の MCP トレース情報の中央処理通番がすべて 0x00000000 の場合は 「*****」 を出力します。

項目番	表示名称	出力形式
4	mmmmmmmm	出力した MCP トレース情報の中でいちばん値の大きい中央処理通番。16 進数 8 けた※ 出力対象の MCP トレース情報がない場合は「*****」 を出力します。 出力対象の MCP トレース情報の中央処理通番がすべて 0x00000000 の場合は「*****」 を出力します。
5	zzz	-c オプションに指定した出力ファイル名

注※

出力対象の MCP トレース情報の中央処理通番が 0x00000000, 0x00000001, 0x00000002 の三つの場合, nnnnnnnnnn に表示される値は 0x00000001 で, mmmmmmmmm に表示される値は 0x00000002 となり, 0x00000000 が nnnnnnnnnn, mmmmmmmmm に表示されることはありません。

出力メッセージ

メッセージ ID	内容	出力先
KFSB91300-E	メモリを確保できません。	標準エラー出力
KFSB91301-E	コマンドの形式が不正です。	標準エラー出力
KFSB91302-E	コマンドのオプションが不正です。	標準エラー出力
KFSB91303-E	コマンドのオプションの組み合わせが不正です。	標準エラー出力
KFSB91308-E	編集できないファイルタイプです。	標準エラー出力
KFSB91310-E	ファイルのオープン処理で障害が発生しました。	標準エラー出力
KFSB91311-E	ファイルからのリード処理で障害が発生しました。	標準エラー出力
KFSB91312-E	ファイルへのライト処理で障害が発生しました。	標準エラー出力
KFSB91313-E	ファイルのクローズ処理で障害が発生しました。	標準エラー出力
KFSB91314-E	ファイルポインタの移動処理で障害が発生しました。	標準エラー出力
KFSB91315-Q	-c オプションに指定したファイルはすでに存在します。	標準出力
KFSB91316-E	指定したファイルに編集するデータがありません。	標準エラー出力
KFSB91317-E	データの編集途中でファイルの終わりに達しました。	標準エラー出力
KFSB91344-I	ヘルプメッセージ	標準出力
KFSB91357-Q	-c オプションに指定したファイルはすでに存在します。(-n オプション指定時)	標準出力

注意事項

- 編集対象のレコードがない場合, ヘッダ情報だけを表示します。

- ・サービス名、論理端末名およびコネクション ID を表示する場合、表示する内容がないときは、「*」を表示します。
- ・該当 MCP ト雷斯情報が MCP ト雷斯ファイルに出力される以前の MCP トレス出力時刻の場合は、「*」を表示します。
- ・MCP トレス情報に欠落があった場合、欠落したレコード数を表示します。欠落したレコード数が 4294967295 を超えた場合、4294967295 と表示します。
- ・CSV ファイルに出力する場合、コンマで区切って出力します。10 進数以外の場合、引用符「"」で囲んで出力します。
- ・CSV ファイルに出力する場合、標準出力に出力しません。
- ・MCP トレス情報取得時に障害が発生した場合など、情報が取得できないときは、0 を表示します。
- ・ファイル名に指定できるコアファイルまたはメモリダンプファイルは、コマンド実行時の環境で作成されたファイルだけです。バージョン、および適用 OS が異なる製品で作成したファイルは指定できません。
- ・CSV ファイルに出力する場合、ファイル作成バージョンは「ファイル作成バージョン：dd-dd-dd」の形式で出力されます。

eemcped (MCP トレスの編集)

機能

指定されたファイルの MCP トレス情報を編集して、標準出力または CSV 形式でファイルに出力します。MCP を利用した通信で障害が発生した場合、特定の論理端末名または IP アドレスなどで編集し、障害を調査するために使用します。

コマンドの形式

```
eemcped [-e 編集種別] [-t [開始時刻] [, 終了時刻]] [-r ランID]
[-s usr | lin]
[-x 中央処理通番の下限値, 中央処理通番の上限値]
[-l 論理端末名 [, 論理端末名…] ]
[-i IPアドレス [, IPアドレス…] ]
[-c CSV出力ファイル名 [-n ファイル出力行] ]
MCPトレスファイル名 [MCPトレスファイル名…]
```

オプションの説明

●-e 編集種別

～《f》

編集種別を指定します。

r：レコード単位に編集、出力します。

f : ファイル情報一覧を出力します。

このオプションに **f** を指定した場合、その他のオプションを指定するとエラーになります。

●-t 開始時刻、終了時刻

指定された時刻内に、出力バッファに出力された MCP トレース情報を編集、出力します。開始時刻、および終了時刻は、1970 年 1 月 1 日 0 時 0 分 0 秒から当年当月当日の現在時刻までの範囲で指定します。

開始時刻または終了時刻のどちらか一方を必ず指定してください。開始時刻の指定を省略した場合は、MCP トレースファイルの先頭から指定した終了時刻までが出力範囲になります。終了時刻の指定を省略した場合は、指定した開始時刻から MCP トレースファイルの最後までが出力範囲になります。

開始時刻、および終了時刻は、「hhmmss [MMDD [YYYY]]」の形式で指定します。

hh : 時 (00≤hh≤23) : 指定を省略できません。

mm : 分 (00≤mm≤59) : 指定を省略できません。

ss : 秒 (00≤ss≤59) : 指定を省略できません。

MM : 月 (01≤MM≤12) : 指定を省略できます。※

DD : 日 (01≤DD≤31) : 指定を省略できます。※

YYYY : 年 (西暦) (1970≤YYYY≤9999) : 指定を省略できます。※

注※

開始時刻、または終了時刻の「年」の指定を省略した場合は、当年の指定月日時刻と見なされます。
「年、月、日」の指定を省略した場合、当年当月当日の指定時刻とみなされます。「月、日」、「月」、
または「日」だけを省略することはできません。省略した場合はエラーになります。「月」または
「日」を省略したい場合は、「年」、「月」、「日」のすべてを省略してください。

-t オプションの指定を省略すると、指定したファイル内のすべての情報を編集、出力します。

●-r ラン ID

指定されたラン ID を持つ MCP トレース情報だけ編集、出力します。

●-s usr | lin

～《usr》

usr : ユーザトレース情報を編集、出力します。

lin : MCP トレース情報を編集、出力します。

●-x 中央処理通番の下限値, 中央処理通番の上限値

～〈1～8 文字の 16 進数〉((0～ffffffff))

指定されたカレント中央処理通番の範囲内の情報だけ編集、出力します。このオプションを指定する場合、上限値および下限値を必ず指定してください。

中央処理通番を指定する場合、最大値 (4294967295) を考慮して指定してください。

●-l 論理端末名

～〈1～8 文字の識別子〉

指定された論理端末名を持つ情報だけ編集、出力します。eemcple 定義コマンドの-l オプションで指定した論理端末名を指定してください。論理端末名は、最大 10 個指定できます。

●-i IP アドレス

～〈7～15 文字の数字〉

IP アドレスを指定する場合は、nnn.nnn.nnn.nnn の形式で指定します。nnn は、0～255 (10 進数) で指定します。IP アドレスは、最大 10 個指定できます。

●-c CSV 出力ファイル名

～〈パス名〉

指定したファイルに編集結果を CSV 形式で出力します。標準出力には出力しません。CSV ファイルに出力する場合、コンマで区切って出力します。10 進数以外の場合、引用符「"」で囲んで出力します。

-n オプションを指定した場合は、パス名中のファイル名は、245 文字以下としてください。-n オプション指定時にファイル名が 246 文字以上の場合はオプションエラーで、このコマンドが異常終了します。

●-n ファイル出力行

((10000～100000))

-c オプションに指定した csv ファイルを分割して出力する場合に 1 ファイルに出力する行数を指定します。見出し行を含めて-n オプションに指定した行数を出力したときに、出力するファイルを切り替えます。切り替えた先のファイルには見出し行から出力します。

分割したファイルには、10 進数の文字列をファイル名の末尾に付与します。付与する文字列は 1 から 9999 ファイルまでは「0001」からの連番で左に 0 を埋めた 4 けたの文字列とします。10000 ファイルを超えた場合は左に 0 埋めをしません。ファイル名に「.」を含む場合は、最後の「.」の前に付与します。

ファイル名に文字列を付与する例を示します。

<例 1> -c に xxx と指定した場合

```
xxx0001
xxx0002
:
```

<例2> -c に xxx.csv と指定した場合

```
xxx0001.csv
xxx0002.csv
:
```

このオプションを省略した場合は、ファイルを分割しないで1ファイルに出力します。

コマンド引数

●MCP トレースファイル名

～ <パス名>

編集するファイル名を指定します。-e オプションに f を指定した場合は、最大 1024 個の MCP トレースファイル名を指定できます。

指定できるオプションの組み合わせを次の表に示します。

表 6-3 eemcped コマンドのオプションの組み合わせ

オプション		-e		-t	-r	-s		-x	-l	-i	-c	-n	ファイル指定	
						usr	lin						单一	複数
-e	r	—	×	○	○	○	○	○	○	○	○	○	○	×
	f	×	—	×	×	×	×	×	×	×	×	×	○	○
-t		○	×	—	○	○	○	○	○	○	○	○	○	×
-r		○	×	○	—	○	○	○	○	○	○	○	○	×
-s	usr	○	×	○	○	—	×	○	○	×	○	○	○	×
	lin	○	×	○	○	×	—	×	×	○	○	○	○	×
-x		○	×	○	○	○	×	—	○	×	○	○	○	×
-l		○	×	○	○	○	×	○	—	×	○	○	○	×
-i		○	×	○	○	×	○	×	×	—	○	○	○	×
-c		○	×	○	○	○	○	○	○	○	—	○	○	×
-n		○	×	○	○	○	○	○	○	○	○	—	○	×
ファイル指定	单一	○	○	○	○	○	○	○	○	○	○	○	—	—
	複数	×	○	×	×	×	×	×	×	×	×	—	—	—

(凡例)

○：指定できます。

×：指定できません。

—：該当しません。

出力形式

コマンド実行時の出力形式について説明します。

■-s に lin を指定した場合

■-s に usr を指定した場合

出力形式中の、右端の数字の意味を次に示します。

- 1：改行キーを含まないで、1行で出力します。
- 2：レコード数分出力します。
- 3：複数行を改行して出力します。

出力形式中の、各変数の意味を次に示します。

変数	意味
al-al-al	コマンドを実行した TP1/EE のバージョン

変数	意味
a2-a2-a2	コマンドを実行した MCP のバージョン
bb....bb	このコマンドの引数
cc....cc	入力ファイル名
dddd/dd/dd dd:dd:dd ddd.ddd	該当ファイルに出力を開始した日付 「年/月/日△時間：分：秒 ミリ秒. マイクロ秒」の形式で表示します。
e1-e1-e1	ファイルを作成した TP1/EE のバージョン
e2-e2-e2	ファイルを作成した MCP のバージョン
e3-e3-e3	ファイルを作成した TP1/FSP のバージョン TP1/FSP を使用した場合だけ出力します。
e3	MCP トレース情報取得タイプ
ff....ff	サービスグループ名 (31 文字以内)
gggggggg	TP1/EE のラン ID (16 進数 8 けた)
hhhh/hh/hh hh:hh:hh hhh.hhh	MCP トレース情報を取得した時刻 「年/月/日△時間：分：秒 ミリ秒. マイクロ秒」の形式で表示します。
ii....ii	イベント名 (15 文字以内) イベントの前後で情報を取得するため、イベント名の後ろに、「前」または「後」が表示されます。また、ee_mcp_utrace_put 関数、または ee_mcp_utrace_put_long 関数で指定した ID を 16 進数で表示します。
jjjjjjjjjj	データ長 (10 進数 10 けた)
oooo/oo/oo oo:oo:oo 000.000	実行時刻 「年/月/日△時間：分：秒 ミリ秒. マイクロ秒」の形式で表示します。
kkkkkkkk	該当レコードを取得した中央処理通番 (16 進数 8 けた)
kkkkkkkk1	該当レコードを取得した受信 ID (16 進数 8 けた)
lll	取得した機能種別 (3 文字) <ul style="list-style-type: none">usr : ユーザトレース API、または MCP で取得lin : MCP で取得
mm....mm	論理端末名 (1~8 文字の識別子)
nn....nn	コネクション ID (1~8 文字の識別子)
ppp.ppp.ppp.ppp	受信/送信先の IP アドレス (10 進数 15 けた以内)
qq....qq	受信/送信先のポート番号 (10 進数 5 けた以内)
rr....rr	サービス名 (1~31 文字の識別子)
0xss....ss	送信 ID (16 進数 16 けた)
0xtt....tt	リターンコード (16 進数 4 けた以内)

変数	意味
tt2....tt2	リターンコード (10 進数 5 けた以内)
uu....uu	実際に送受信したデータ長
vv vv vv vv vv vv vv vvvvvvvvvvvvvvvvvvv	次に示すデータです。 -s オプションに usr を指定した場合 ユーザデータ、または MCP で取得したデータ -s オプションに lin を指定した場合 MCP で取得したデータ
wwww/ww/ww ww:ww:ww www.www	MCP トレース情報の出力を再開した時刻 「年/月/日△時間：分：秒 ミリ秒. マイクロ秒」の形式で表示します。
yy....yy	すべての出力バッファが使用中となり、その間に MCP トレース情報出力要求が発生したため、破棄したレコード数
zzzz/zz/zz zz:zz:zz zzz.zzz	破棄を開始した時刻 「年/月/日△時間：分：秒 ミリ秒. マイクロ秒」の形式で表示します。
yy....yy2	破棄を開始したレコードを取得した中央処理通番 該当しない場合は 0 を表示します。
yy....yy3	破棄を開始したレコードを取得した受信 ID 該当しない場合は 0 を表示します。

■-n オプションを指定した場合の標準出力

日本語と英字共通

EARLIEST TIME NAME	- LATEST TIME	EARLIEST CPN - LATEST CPN FILE
xxxx/xx/xx xx:xx:xx xxx.xxx - yyyy/yy/yy yy:yy:yy yyyy.yyy 01		nnnnnnnnnn mmmmmmmmm zzz00
xxxx/xx/xx xx:xx:xx xxx.xxx - yyyy/yy/yy yy:yy:yy yyyy.yyy 02		nnnnnnnnnn mmmmmmmmm zzz00
:	:	:

項目番	表示名称	出力形式
1	xxxx/xx/xx xx:xx:xx xxx.xxx	出力した MCP トレース情報の中でいちばん古い起動時刻 出力対象の MCP トレース情報がない場合は 「****/**/**:***:***.***」 を出力します。
2	yyyy/yy/yy yy:yy:yy yyy.yyy	出力した MCP トレース情報の中でいちばん新しい起動時刻 出力対象の MCP トレース情報がない場合は 「****/**/**:***:***.***」 を出力します。
3	nnnnnnnn	出力した MCP トレース情報の中でいちばん値の小さい中央処理通番。16 進数 8 けた※ 出力対象の MCP トレース情報がない場合は

項目番	表示名称	出力形式
3	nnnnnnnn	「*****」 を出力します。 出力対象の MCP トレース情報の中央処理通番がすべて 0x00000000 の場合は 「*****」 を出力します。
4	mmmmmmmm	出力した MCP トレース情報の中でいちばん値の大きい中央処理通番。16 進数 8 けた※ 出力対象の MCP トレース情報がない場合は 「*****」 を出力します。 出力対象の MCP トレース情報の中央処理通番がすべて 0x00000000 の場合は 「*****」 を出力します。
5	zzz	-c オプションに指定した出力ファイル名

注※

出力対象の MCP トレース情報の中央処理通番が 0x00000000, 0x00000001, 0x00000002 の三つの場合, nnnnnnnn に表示される値は 0x00000001 で, mmmmmmmmm に表示される値は 0x00000002 となり, 0x00000000 が nnnnnnnn, mmmmmmmmm に表示されることはありません。

出力メッセージ

メッセージ ID	内容	出力先
KFSB91300-E	メモリを確保できません。	標準エラー出力
KFSB91301-E	コマンドの形式が不正です。	標準エラー出力
KFSB91302-E	コマンドのオプションが不正です。	標準エラー出力
KFSB91303-E	コマンドのオプションの組み合わせが不正です。	標準エラー出力
KFSB91305-E	ファイルのヘッダ情報が不正です	標準エラー出力
KFSB91306-E	データブロックが不正です	標準エラー出力
KFSB91308-E	編集できないファイルタイプです。	標準エラー出力
KFSB91309-E	実行したコマンドのバージョンとファイル作成バージョンが不一致です。	標準エラー出力
KFSB91310-E	ファイルのオープン処理で障害が発生しました。	標準エラー出力
KFSB91311-E	ファイルからのリード処理で障害が発生しました。	標準エラー出力
KFSB91312-E	ファイルへのライト処理で障害が発生しました。	標準エラー出力
KFSB91313-E	ファイルのクローズ処理で障害が発生しました。	標準エラー出力
KFSB91315-Q	-c オプションに指定したファイルはすでに存在します。	標準出力
KFSB91331-I	ヘルプメッセージ	標準出力

メッセージID	内容	出力先
KFSB91357-Q	-c オプションに指定したファイルはすでに存在します。(-n オプション指定時)	標準出力

注意事項

- -e オプションに r を指定して、編集対象のレコードがない場合、-e オプションに f を指定したときの表示と同様の表示となります。
- サービス名、論理端末名およびコネクション ID を表示する場合、表示する内容がないときは、「*」を表示します。
- MCP トレース情報に欠落があった場合、欠落したレコード数を表示します。欠落したレコード数が 4294967295 を超えた場合、4294967295 と表示します。
- CSV ファイルに出力する場合、コンマで区切って出力します。10 進数以外の場合、引用符「"」で囲んで出力します。
- CSV ファイルに出力する場合、標準出力に出力しません。
- MCP トレース情報取得時に障害が発生した場合などによって情報が取得できないときは、0 を表示します。
- -e オプションに f を指定した場合、その他のオプションの省略値を無視します。
- CSV ファイルに出力する場合、ファイル作成バージョンは「ファイル作成バージョン：ee-ee-ee」の形式で出力されます。

eemcpfput (MCP トレースの強制ファイル出力)

機能

MCP トレースバッファにスタックしている MCP トレース情報をファイルに強制出力して、ファイルをスワップします。

コマンドの形式

```
eemcpfput -g サービスグループ名
```

オプションの説明

●-g サービスグループ名

～ <1～31 文字の識別子>

TP1/EE プロセスのサービスグループ名を指定します。

出力メッセージ

メッセージID	内容	出力先
KFSB91300-E	メモリを確保できません。	標準エラー出力
KFSB91301-E	コマンドの形式が不正です。	標準エラー出力
KFSB91302-E	コマンドのオプションが不正です。	標準エラー出力
KFSB91325-I	強制出力コマンドが成功しました。	標準出力
KFSB91326-E	強制出力コマンドが失敗しました。	標準エラー出力
KFSB91327-E	強制出力コマンドが失敗しました。	標準エラー出力
KFSB91328-E	コマンド実行中に内部矛盾が発生しました。	標準エラー出力
KFSB91335-I	ヘルプメッセージ	標準出力

eemcplscn (コネクション状態表示)

機能

コネクション状態を表示します。

コマンドの形式

```
eemcplscn -g サービスグループ名  
-c コネクションID [-d]
```

オプションの説明

●-g サービスグループ名

～〈1～31文字の識別子〉

処理対象のサービスグループ名を指定します。

●-c コネクションID

～〈1～8文字の識別子〉

状態表示するコネクションIDを指定します。

eemcpcn 定義コマンドの-c オプションに指定したコネクションIDを指定してください。

コネクションIDは、一度に8個まで指定できます。複数のコネクションIDを指定する場合は、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。コネクションIDは重複して指定できません。

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

*

すべてのコネクション状態を表示します。

先行文字列*

先行文字列で始まるすべてのコネクション状態を表示します。

<複数指定の例> cnn1, cnn2, cnn3 を指定する場合

-c "cnn1 cnn2 cnn3"

<一括指定の例> cnn で始まるすべてのコネクションを指定する場合

-c "cnn*"

●-d

コネクションの状態と該当するコネクションに対応する論理端末の情報を表示します。このオプションの指定を省略すると、コネクションの状態だけを表示します。

出力形式

コマンド実行時の出力形式について説明します。

■-d オプションを指定しない場合

```
Service group name = aa....aa
CONID  Type  IPAddress      Port      Mode  S1
bb....bb  ccc  ddd.ddd.ddd.ddd  ee....ee  fff  ggg
bb....bb  ccc  ddd.ddd.ddd.ddd  ee....ee  fff  ggg
bb....bb  ccc  ddd.ddd.ddd.ddd  ee....ee  fff  ggg
                                :

```

■-d オプションを指定する場合

```
Service group name = aa....aa
CONID  Type  IPAddress      Port      Mode  S1  LENAME  S2  QueNum
bb....bb  ccc  ddd.ddd.ddd.ddd  ee....ee  fff  ggg  hh....hh  iii  jj....jj
bb....bb  ccc  ddd.ddd.ddd.ddd  ee....ee  fff  ggg  hh....hh  iii  jj....jj
bb....bb  ccc  ddd.ddd.ddd.ddd  ee....ee  fff  ggg  hh....hh  iii  jj....jj
                                :

```

出力形式中の、各変数の意味を次に示します。

変数	意味
aa....aa	処理対象の TP1/EE サービスグループ名
bb....bb	コネクション ID
ccc	プロトコル種別

変数	意味
ccc	<ul style="list-style-type: none"> TCP : TCP/IP プロトコル
ddd.ddd.ddd.ddd	接続先の IP アドレス コネクション状態が未確立状態の場合は、「*」を表示します。
ee....ee	接続先のポート番号 コネクション状態が未確立状態の場合は、「*」を表示します。
fff	モード <ul style="list-style-type: none"> CLT : クライアント型 SVR : サーバ型
ggg	コネクション状態 <ul style="list-style-type: none"> INA : 未確立状態 BID : 確立処理中 ACT : 確立状態 OBS : 解放処理中
hh....hh	論理端末名
iii	論理端末状態 <ul style="list-style-type: none"> INA : 閉塞中 BID : 閉塞解除処理中 ACT : 閉塞解除中 OBS : 閉塞処理中
jj...jj	端末キュー滞留数

出力メッセージ

メッセージ ID	内容	出力先
KFSB91401-E	領域の確保に失敗しました。	標準エラー出力
KFSB91402-E	通信障害が発生しました。	標準エラー出力
KFSB91403-E	コマンドの形式が不正です。	標準エラー出力
KFSB91404-E	フラグ引数が不正です。	標準エラー出力
KFSB91405-E	コマンド処理でエラーが発生しました。	標準エラー出力
KFSB91406-E	内部矛盾が発生しました。	標準エラー出力
KFSB91430-I	ヘルプメッセージ	標準出力

eemcplsle (論理端末の状態表示)

機能

論理端末の状態を表示します。

コマンドの形式

```
eemcplsle -g サービスグループ名  
-l 論理端末名
```

オプションの説明

●-g サービスグループ名

～ <1～31 文字の識別子>

処理対象のサービスグループ名を指定します。

●-l 論理端末名

～ <1～8 文字の識別子>

状態表示する論理端末名を指定します。eemcple 定義コマンドの-l オプションで指定した論理端末名を指定してください。

論理端末名は、一度に 8 個まで指定できます。複数の論理端末名を指定するときは、引用符「"」で囲んで、論理端末名と論理端末名との間を空白で区切れます。論理端末名は重複して指定できません。

また、論理端末名は、「*」を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外の論理端末名とは、混在して指定できません。一括指定をするときは、引用符「"」で囲んで指定します。

*

すべての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*"

出力形式

```
Service group name = aa....aa
LENAME      Type    Status     QueNum
bb....bb    ccc     ddd        ee....ee
bb....bb    ccc     ddd        ee....ee
:
:
```

出力形式中の、各変数の意味を次に示します。

変数	意味
aa....aa	処理対象のTP1/EEサービスグループ名
bb....bb	論理端末名
ccc	プロトコル種別 <ul style="list-style-type: none">TCP : TCP/IP プロトコルUDP : UDP プロトコル
ddd	論理端末状態 <ul style="list-style-type: none">INA : 閉塞中BID : 閉塞解除処理中ACT : 閉塞解除中OBS : 閉塞処理中
ee....ee	端末キュー滞留数

出力メッセージ

メッセージ ID	内容	出力先
KFSB91401-E	領域の確保に失敗しました。	標準エラー出力
KFSB91402-E	通信障害が発生しました。	標準エラー出力
KFSB91403-E	コマンドの形式が不正です。	標準エラー出力
KFSB91404-E	フラグ引数が不正です。	標準エラー出力
KFSB91405-E	コマンド処理でエラーが発生しました。	標準エラー出力
KFSB91406-E	内部矛盾が発生しました。	標準エラー出力
KFSB91450-I	ヘルプメッセージ	標準出力

7

障害対策

この章では、MCP 運用時の障害、および障害時の処理について説明します。

7.1 TCP/IP プロトコルに関する障害

ここでは、TCP/IP プロトコルを使用して通信する場合の障害について説明します。

7.1.1 コネクション障害

コネクション障害時の MCP の処理について、次の表に示します。

表 7-1 コネクション障害時の MCP の処理 (TCP/IP)

項番	障害の内容	MCP の処理
1	コネクション確立時障害 (クライアント型)	1. コネクション確立時障害を通知するメッセージログを出力します。 2. MCP 後処理トランザクションを起動します。
2	コネクション確立時障害 (サーバ型)	サーバ側コネクション確立時障害を通知するメッセージログを出力します。
3	コネクション切断	1. コネクション障害を通知するメッセージログを出力します。 2. MCP 後処理トランザクションを起動します。 3. 端末キューを破棄します。
4	コネクション確立 UOC エラーリターン	1. UOC エラーリターンを通知するメッセージログを出力します。 2. コネクション受け入れを拒否します。 3. MCP 後処理トランザクションを起動しません。
5	無通信監視時間タイムアウト	1. 無通信監視時間タイムアウトを通知するメッセージログを出力します。 2. コネクションを解放して MCP 後処理トランザクションを起動します (入力メッセージは破棄されます)。 3. 端末キューを破棄します。 4. 同期型送受信 (ee_mcp_sendrecv) 要求の場合は、エラーリターンします。

7.1.2 受信障害

メッセージの受信障害時の MCP の処理について、次の表に示します。

表 7-2 メッセージの受信障害時の MCP の処理 (TCP/IP)

項番	障害の内容	MCP の処理
1	サービス名未定義	1. サービス名不正を通知するメッセージログを出力します。 2. 不正サービス名称検出通知トランザクション (ERRTRN1) を起動します。
2	入力メッセージ編集 UOC パラメタ不正	1. UOC パラメタ不正を通知するメッセージログを出力します。

項番	障害の内容	MCP の処理
2	入力メッセージ編集 UOC パラメタ不正	2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 このとき、入力メッセージは破棄されます。 3. 同期型送受信 (ee_mcp_sendrecv) 要求の場合は、エラーリターンします。
3	入力メッセージ編集 UOC エラーリターン	1. UOC エラーリターンを通知するメッセージログを出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 このとき、入力メッセージは破棄されます。 3. 同期型送受信 (ee_mcp_sendrecv) 要求の場合は、エラーリターンします。
4	受信バッファオーバフロー	1. メッセージ受信失敗を通知するメッセージログを出力します。 2. コネクションを解放して MCP 後処理トランザクションを起動します。このとき、入力メッセージは破棄されます。 3. 端末キューを破棄します。 4. 同期型送受信 (ee_mcp_sendrecv) 要求の場合は、エラーリターンします。
5	受信バッファ数不足	1. メモリ不足を通知するメッセージログを出力します。 2. コネクションを解放して MCP 後処理トランザクションを起動します。このとき、入力メッセージは破棄されます。 3. 端末キューを破棄します。 4. 同期型送受信 (ee_mcp_sendrecv) 要求の場合は、エラーリターンします。
6	入力セグメント判定 UOC パラメタ不正	1. UOC パラメタ不正を通知するメッセージログを出力します。 2. コネクションを解放して MCP 後処理トランザクションを起動します。このとき、入力メッセージは破棄されます。 3. 端末キューを破棄します。 4. 同期型送受信 (ee_mcp_sendrecv) 要求の場合は、エラーリターンします。
7	入力セグメント判定 UOC エラーリターン	1. UOC エラーを通知するメッセージログを出力します。 2. コネクションを解放して MCP 後処理トランザクションを起動します。このとき、入力メッセージは破棄されます。 3. 端末キューを破棄します。 4. 同期型送受信 (ee_mcp_sendrecv) 要求の場合は、エラーリターンします。
8	後続メッセージ時間タイムアウト	1. 後続メッセージ監視タイムアウトを通知するメッセージログを出力します 2. コネクションを解放して MCP 後処理トランザクションを起動します。このとき、入力メッセージは破棄されます。 3. 端末キューを破棄します。 4. 同期型送受信 (ee_mcp_sendrecv) 要求の場合は、エラーリターンします。
9	リソース不足	1. メッセージ受信失敗を通知するメッセージログを出力します。

7. 障害対策

項番	障害の内容	MCP の処理
9	リソース不足	<ol style="list-style-type: none"> 2. コネクションを解放して MCP 後処理トランザクションを起動します。このとき、入力メッセージは破棄されます。 3. 端末キューを破棄します。 4. 同期型送受信 (ee_mcp_sendrecv) 要求の場合は、エラーリターンします。

7.1.3 送信障害

メッセージの送信障害時の MCP の処理について、次の表に示します。

表 7-3 メッセージの送信障害時の MCP の処理 (TCP/IP)

項番	障害の内容	MCP の処理
1	送信障害(コネクション障害)	<ol style="list-style-type: none"> 1. 送信障害を通知するメッセージログを出力します。 2. コネクションを解放して MCP 後処理トランザクションを起動します。 3. 送信形態によって処理が異なります。 <ul style="list-style-type: none"> ・同期一方送信 (ee_mcp_sendsync) 要求、および同期型送受信 (ee_mcp_sendrecv) 要求の場合はエラーリターンします。 ・非同期一方送信 (ee_mcp_send) 要求の場合は送信障害トランザクション (ERRTRNS) を起動します。
2	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を出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。このとき、出力メッセージは破棄されます。 3. 送信形態によって処理が異なります。 <ul style="list-style-type: none"> ・同期一方送信 (ee_mcp_sendsync) 要求、および同期型送受信 (ee_mcp_sendrecv) 要求の場合はエラーリターンします。 ・非同期一方送信 (ee_mcp_send) 要求の場合は送信障害トランザクション (ERRTRNS) を起動します。
3	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を出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。このとき、出力メッセージは破棄されます。 3. 送信形態によって処理が異なります。 <ul style="list-style-type: none"> ・同期一方送信 (ee_mcp_sendsync) 要求、および同期型送受信 (ee_mcp_sendrecv) 要求の場合はエラーリターンします。 ・非同期一方送信 (ee_mcp_send) 要求の場合は送信障害トランザクション (ERRTRNS) を起動します。
4	送信バッファオーバフロー	<ol style="list-style-type: none"> 1. 送信失敗を通知するメッセージログを出力します。 2. 同期一方送信 (ee_mcp_sendsync) 要求、非同期一方送信 (ee_mcp_send) 要求、および同期型送受信 (ee_mcp_sendrecv) 要求すべてエラーリターンします。

項番	障害の内容	MCP の処理
4	送信バッファオーバフロー	3. 出力メッセージ編集後にオーバフローを検知した場合、送信形態によって処理が異なります。 ・同期一方送信 (ee_mcp_sendsync) 要求のときはエラーリターンします。 ・一方送信 (ee_mcp_send) 要求のときは送信障害トランザクション (ERRTRNS) を起動します。
5	送信バッファ数不足	1. メモリ不足を通知するメッセージログを出力します。 2. 同期一方送信 (ee_mcp_sendsync) 要求、非同期一方送信 (ee_mcp_send) 要求、および同期型送受信 (ee_mcp_sendrecv) 要求すべてエラーリターンします。
6	メッセージ送信完了時間のタイムアウト (eemcpcn の sndtim, sndsynctim, sndcmptim 指定時間を超過)	1. 送信失敗を通知するメッセージログを出力します。 2. コネクションを解放して MCP 後処理トランザクションを起動します。 3. 同期型一方送信 (ee_mcp_sendsync) 要求、同期型送受信 (ee_mcp_sendrecv) 要求の場合はエラーリターンします。 4. 一方送信 (ee_mcp_send) 要求の場合は送信障害トランザクション (ERRTRNS) を起動します。
7	メッセージ送受信完了時間のタイムアウト (eemcpcn の sndrcvtime, API の watchtime 指定時間を超過)	1. 送信失敗を通知するメッセージログを出力します。 2. コネクションを解放して MCP 後処理トランザクションを起動します。 3. エラーリターンします。

7.1.4 トランザクションの起動の失敗 (リソース不足によって処理キュー登録に失敗)

トランザクションの起動に失敗した場合の MCP の処理について、次の表に示します。

表 7-4 トランザクションの起動に失敗した場合の MCP の処理 (TCP/IP)

項番	障害の内容	MCP の処理
1	メッセージ入力時	1. メモリ不足を通知するメッセージログを出力します。 2. コネクションを解放して MCP 後処理トランザクションを起動します。このとき、入力メッセージは破棄されます。
2	コネクション確立時	1. メモリ不足を通知するメッセージログを出力します。 2. コネクション確立を通知するメッセージログを出力します。 3. MCP 後処理トランザクションは起動しません。
3	コネクション解放時	1. メモリ不足を通知するメッセージログを出力します。 2. コネクション解放を通知するメッセージログを出力します。 3. MCP 後処理トランザクションは起動しません。
4	論理端末の閉塞時	1. メモリ不足を通知するメッセージログを出力します。

項番	障害の内容	MCP の処理
4	論理端末の閉塞時	2. 論理端末の閉塞を通知するメッセージログを出力します。 3. MCP 後処理トランザクションは起動しません。
5	論理端末の閉塞解除時	1. メモリ不足を通知するメッセージログを出力します。 2. 論理端末の閉塞解除を通知するメッセージログを出力します。 3. MCP 後処理トランザクションは起動しません。
6	送信障害通知時	1. メモリ不足を通知するメッセージログを出力します。 2. 送信障害通知トランザクション (ERRTRNS) の起動失敗を通知するメッセージログを出力します。

7.1.5 ハーフコネクションに関連する障害

OS 障害などで、RST/FIN パケットが送信されないで相手システムがダウンすると、TP1/MCP の該当コネクションはハーフコネクション状態になります。この状態では、新規にコネクションを確立できないため、eemcpdctcn コマンドを実行、または ee_mcp_dctcn 関数を発行し、コネクションを解放する必要があります。

7.2 UDP プロトコルに関する障害

ここでは、 UDP プロトコルを使用して通信する場合の障害について説明します。

7.2.1 論理端末の閉塞解除の失敗

論理端末の閉塞解除に失敗した場合の MCP の処理について、次の表に示します。

表 7-5 論理端末の閉塞解除に失敗した場合の MCP の処理 (UDP)

項番	障害の内容	MCP の処理
1	論理端末の閉塞解除の失敗（ポートオープン失敗）	1. 論理端末の閉塞解除の失敗を通知するメッセージログを出力します。 2. MCP 後処理トランザクションを起動します（閉塞）。

7.2.2 受信障害

メッセージの受信障害時の MCP の処理について、次の表に示します。

表 7-6 メッセージの受信障害時の MCP の処理 (UDP)

項番	障害の内容	MCP の処理
1	サービス名不正	1. サービス名不正を通知するメッセージログを出力します。 2. 不正サービス名称検出通知トランザクション (ERRTRN1) を起動します。
2	受信バッファ数不足	1. バッファ不足を通知するメッセージログを出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 このとき、入力メッセージは破棄されます。 3. 端末キューを破棄します。
3	入力メッセージ編集 UOC パラメタ不正	1. UOC パラメタ不正を通知するメッセージログを出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 このとき、入力メッセージは破棄されます。 3. 端末キューを破棄します。
4	入力メッセージ編集 UOC エラーリターン	1. UOC エラーリターンを通知するメッセージログを出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 このとき、入力メッセージは破棄されます。 3. 端末キューを破棄します。
5	受信バッファオーバフロー	1. メッセージ受信失敗を通知するメッセージログを出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 このとき、入力メッセージは破棄されます。 3. 端末キューを破棄します。

7.2.3 送信障害

メッセージの送信障害時の MCP の処理について、次の表に示します。

表 7-7 メッセージの送信障害時の MCP の処理 (UDP)

項番	障害の内容	MCP の処理
1	送信障害	<ol style="list-style-type: none"> 1. 送信障害を通知するメッセージログを出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 3. 端末キューを破棄します。 4. 送信形態によって処理が異なります。 <ul style="list-style-type: none"> ・同期一方送信 (ee_mcp_sendsync) 要求の場合はエラーリターンします。 ・非同期一方送信 (ee_mcp_send) 要求の場合は送信障害トランザクション (ERRTRNS) を起動します。
2	送信バッファ数不足	<ol style="list-style-type: none"> 1. メモリ不足を通知するメッセージログを出力します。 2. 同期一方送信 (ee_mcp_sendsync) 要求、非同期一方送信 (ee_mcp_send) 要求ともにエラーリターンします。
3	送信バッファオーバフロー	<ol style="list-style-type: none"> 1. 送信失敗を通知するメッセージログを出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 3. 端末キューを破棄します。 4. 同期一方送信 (ee_mcp_sendsync) 要求、非同期一方送信 (ee_mcp_send) 要求ともにエラーリターンします。 5. 出力メッセージ編集後にオーバフローを検知した場合、送信形態によって処理が異なります。 <ul style="list-style-type: none"> ・同期一方送信 (ee_mcp_sendsync) 要求のときはエラーリターンします。 ・非同期一方送信 (ee_mcp_send) 要求のときは送信障害トランザクション (ERRTRNS) を起動します。
4	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を出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 3. 端末キューを破棄します。 4. 送信形態によって処理が異なります。 <ul style="list-style-type: none"> ・同期一方送信 (ee_mcp_sendsync) 要求の場合はエラーリターンします。 ・非同期一方送信 (ee_mcp_send) 要求の場合は送信障害トランザクション (ERRTRNS) を起動します。
5	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を出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 3. 端末キューを破棄します。 4. 送信形態によって処理が異なります。 <ul style="list-style-type: none"> ・同期一方送信 (ee_mcp_sendsync) 要求の場合はエラーリターンします。

項番	障害の内容	MCP の処理
5	出力メッセージ編集 UOC パラメタ不正	・非同期一方送信 (ee_mcp_send) 要求の場合は送信障害トランザクション (ERRTRNS) を起動します。

7.2.4 トランザクションの起動の失敗 (リソース不足によって処理キュー登録に失敗)

トランザクションの起動に失敗した場合の MCP の処理について、次の表に示します。

表 7-8 トランザクションの起動に失敗した場合の MCP の処理 (UDP)

項番	障害の内容	MCP の処理
1	メッセージ入力時	1. メモリ不足を通知するメッセージログを出力します。 2. 論理端末を閉塞して MCP 後処理トランザクションを起動します。 このとき、入力メッセージは破棄されます。
2	論理端末の閉塞時	1. メモリ不足を通知するメッセージログを出力します。 2. 論理端末を閉塞するメッセージログを出力します。 3. 論理端末を閉塞します。 4. MCP 後処理トランザクションを起動しません。
3	論理端末の閉塞解除時	1. メモリ不足を通知するメッセージログを出力します。 2. 論理端末を閉塞するメッセージログを出力します。 3. 論理端末を閉塞します。 4. MCP 後処理トランザクションを起動しません。
4	送信障害通知時	1. メモリ不足を通知するメッセージログを出力します。 2. 送信障害通知トランザクション (ERRTRNS) の起動失敗を通知するメッセージログを出力します。

付録

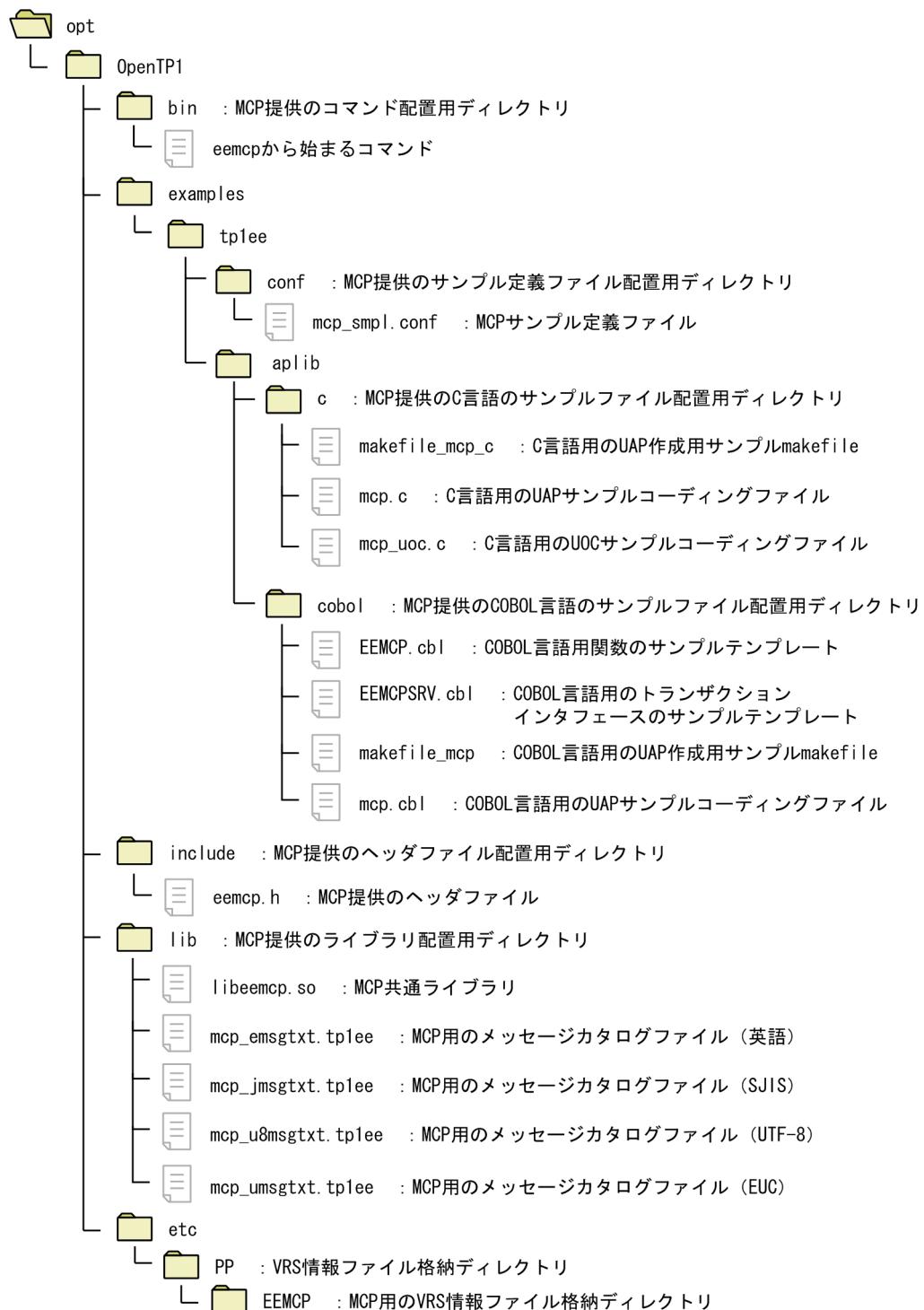
付録 A インストールディレクトリとファイル

MCP のインストールディレクトリの構成とファイルを次の図に示します。

この図の構成は、OS が Linux の場合の構成です。OS が AIX の場合は次のとおり読み変えてください。

- /opt/OpenTP1 は、/BeTRAN に読み変えてください。
- lib ディレクトリ下のライブラリの拡張子は、.so から、.a に読み変えてください。
- mcp_u8msgtxt.tplee (MCP 用のメッセージカタログファイル(UTF-8)) は提供しません。

図 A-1 MCP のインストールディレクトリの構成とファイル



付録 B MCP のエラーコード一覧

MCP の API で返されるエラーコードの一覧と、各エラーコードの意味を説明します。

表 B-1 MCP の API で返されるエラーコード

項目番	COBOL 言語の ステータスコー ド	C 言語のリターン値	意味
1	00000	EE_OK	正常に終了しました。
2	00001	EECOMER_ENVIRON	TP1/EE の環境下にありません。
3	00004	EECOMER_CNDUOC	この関数を実行できない UOC から発行してい るため、この機能は使用できません。
4	10001	EEMCPER_MESSAGE_TOO_BIG	次のどちらかを意味します。 • データ長に指定できる最大サイズを超える値 を設定しています。 • sdata leng に送信できる最大サイズを超える 値を設定しています。
5	10002	EEMCPER_MESSAGE_TOO_SHORT	次のどちらかを意味します。 • データ長に 4 未満を設定しています。 • sdata leng に 0 (UDP の場合は 24) 以下を 設定しています。
6	10003	EEMCPER_ARGUMENT	次のどちらかを意味します。 • データ長以外の引数に設定した値が間違って います。 • senddata, および watchtime に設定した値 が間違っています。
		EEMCPER_INFO	info に設定した値が間違っています。
7	10004	EEMCPER_ACTION	action に設定した値が間違っています。
8	10005	EEMCPER_OPTION	次のどちらかを意味します。 • opcd に設定した値が間違っています。 • flags に設定した値が間違っています。
9	10006	EEMCPER_COMMFORM	commform に設定した値が間違っています。
10	10007	EEMCPER_TIME	watchtime に設定した時間が経過しました。
11	10008	EEMCPER_NUM	次のどちらかを意味します。 • le_num に設定した値が間違っています。 • con_num に設定した値が間違っています。
12	10009	EEMCPER_CONDITION	発行条件が不正です。
13	10010	EEMCPER_CON_NAME	指定したコネクション ID の中に不正なコネク ション ID があります。

項目番号	COBOL 言語の ステータスコー ド	C 言語のリターン値	意味
14	10011	EEMCPER_LE_NAME	<p>次のどちらかを意味します。</p> <ul style="list-style-type: none"> termnam に設定した論理端末名が間違っています。 指定した論理端末名の中に不正な論理端末名があります。
15	10012	EEMCPER_FORM	form に設定した値が間違っています。
16	10020	EEMCPER_TERM_HOLD	termnam に設定した論理端末名は閉塞中です。
17	10021	EEMCPER_UOC_ERR	出力メッセージ編集 UOC がエラーリターンしました。
18	10022	EEMCPER_UOC_INVALID	出力メッセージ編集 UOC で障害が発生しました。
19	10025	EEMCPER_SENDER	システムに対する送信時に障害が発生しました。
20	10026	EEMCPER_STOP	トラブルシュート機能が停止中です。
21	10027	EEMCPER_SYSERR	システムエラーが発生しました。
22	10028	EEMCPER_PART	<p>次のどちらかを意味します。</p> <ul style="list-style-type: none"> 一部の論理端末の閉塞または閉塞解除に失敗しました。 一部のコネクションの確立または解放に失敗しました。
23	10030	EEMCPER_MEMORY_SHORT	メモリ不足が発生しました。
24	10031	EEMCPER_OVERFLOW	inbuflen の指定値を超えるセグメントを受信しました。inbuflen の指定値を超えた部分は切り捨てられました。
25	10032	EEMCPER_INBUFLENG	inbuflen の指定値が不足しています。9 バイト以上の領域を確保してください。
26	10033	EEMCPER_SENDER_RECV	メッセージ受信が仕掛けかり中です。
27	10034	EEMCPER_TERM_IN_USE	出力先の論理端末は、ほかの UAP で仕掛けかり中です。
28	10035	EEMCPER_SENDTIME	メッセージ送信完了監視タイマのタイムアウトが発生しました。
29	10036	EEMCPER_RECV	受信待ちでコネクション障害が発生しました。
30	10040	EEMCPER_ADDRESS	指定した IP アドレスまたはホスト名が不正です。
31	10041	EEMCPER_PORT	指定したポート番号が不正です。
32	10044	EEMCPER_IDNAME	idname に設定した値が間違っています。
33	10045	EEMCPER_ACT_BEGIN	指定したコネクションが確立処理中状態です。

項目番号	COBOL 言語の ステータスコー ド	C 言語のリターン値	意味
34	10046	EEMCPER_DCT_BEGIN	指定したコネクションが解放処理中状態です。
35	10047	EEMCPER_DCT	指定したコネクションが未確立状態です。
36	10048	EEMCPER_ALREADY	指定したコネクションはすでに使用中です。

付録 C 運用コマンド実行機能での MCP コマンドの実行可否

運用コマンド実行機能での、MCP コマンドの自 TP1/EE プロセスに対する実行可否を次の表に示します。

表 C-1 運用コマンド実行機能での MCP コマンドの自 TP1/EE プロセスに対する実行可否

コマンド	トランザクション種別																
	MI	M E	M N	E1	E2	E3	E4	ES	ER	T M	UI	M V	RL	UT	C U	JI	
eemcpactcn	×	×	○	○	○	○	○	○	○	○	×	○	○	○	○	×	
eemcpactcnto	×	×	○	○	○	○	○	○	○	○	×	○	○	○	○	×	
eemcpactle	×	×	○	○	○	○	○	○	○	○	×	○	○	○	○	×	
eemcpdctcn	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○	×	
eemcpdctle	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○	×	
eemcpdump	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
eemcped	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
eemcpfput	×	×	○	○	○	○	○	○	○	○	×	○	○	○	○	×	
eemcplscn	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○	×	
eemcplsle	×	○	○	○	○	○	○	○	○	○	×	○	○	○	○	×	

(凡例)

○：実行できます。

×：実行できません。

■MCP 後処理トランザクションで実行できないコマンド

TP1/EE および XTC が提供するコマンドのうち、次に示すコマンドについては、運用コマンド実行機能での MCP 後処理トランザクションで実行できません。

- eeadmccsetup
- eechgap
- eedbqmstp
- eedbqtblh
- eedbqtbl0
- eemchotqend
- eepcerefer
- eerasget
- eerpcnamdel
- eesvstart

- eetrbwtr

付録 D 各バージョンの変更内容

各バージョンの変更内容を示します。

付録 D.1 変更内容 (3000-3-F57-50)

uCosminexus TP1/EE/Message Control Extension 02-04 の製品の変更内容 (3000-3-F57-50) を表に示します。

追加・変更内容

適用 OS に AIX V7.2 を追加した。

uCosminexus TP1/EE/Message Control Extension Linux5, Linux6 02-02/Linux6, Linux7 02-04 の製品の変更内容 (3000-3-F57-50) を表に示します。

追加・変更内容

eemcpctrl (トレース定義) の-b オプションのオペランドの説明を変更した。

次のライブラリ関数のトランザクション種別の使用有無を変更した。

- ee_mcp_send, CBLEEMCP('SEND')
- ee_mcp_sendsync, CBLEEMCP('SENDSYNC')
- ee_mcp_actcn, CBLEEMCP('ACTCN')
- ee_mcp_actle, CBLEEMCP('ACTLE')

付録 D.2 変更内容 (3000-3-F57-40)

uCosminexus TP1/EE/Message Control Extension 02-03 の製品の変更内容 (3000-3-F57-40) を表に示します。

追加・変更内容

各バージョンの変更内容を追加した。

付録 D.3 変更内容 (3000-3-F57-30)

uCosminexus TP1/EE/Message Control Extension 02-02 の製品の変更内容 (3000-3-F57-30) を表に示します。

追加・変更内容

各バージョンの変更内容を追加した。

uCosminexus TP1/EE/Message Control Extension 02-01 の製品の変更内容 (3000-3-F57-30) を表に示します。

追加・変更内容

回線トレース情報を CSV ファイル形式で出力する際、ファイルを分割できるようにした。

これに伴い、次のコマンドを変更した。

- eemcpdump
- eemcped

uCosminexus TP1/EE/Message Control Extension 02-00 の製品の変更内容 (3000-3-F57-30) を表に示します。

追加・変更内容

自システムからのメッセージ送信と、相手システムからの応答メッセージの受信を連続で行う同期型メッセージの送受信機能を追加した。

これに伴い、次の MCP 構成定義を変更した。

コネクション定義 (TCP/IP)

eemcpn

論理端末定義 (TCP/IP)

eemcpn

次のライブラリ関数 (C 言語) を追加した。

- ee_mcp_sendrecv

次のライブラリ関数 (C 言語) の説明を変更した。

- トランザクションインターフェース情報

次のライブラリ関数 (COBOL 言語) を追加した。

- CBLEEMCP('SENDRECV')

次のライブラリ関数 (COBOL 言語) の説明を変更した。

- トランザクションインターフェース情報
- CBLEEMCP('SEND ')
- CBLEEMCP('SENDSYNC')

MCP で使用できるユーザオウンコーディング (UOC) に、コネクション確立 UOC を追加した。

これに伴い、次の MCP 構成定義を変更した。

論理端末アプリケーション定義

eemcpfunc

アプリケーション定義

eemcpfunc

次のライブラリ関数を追加した。

- ee_mcp_cninfo
- CBLEEMCP('CNINFO ')

クライアント型コネクションでは、相手システムを指定したコネクションを確立できるようにした。

これに伴い、次の MCP 構成定義を変更した。

追加・変更内容

TCP/IP 定義の開始

 eemcptcpstart

コネクション定義 (TCP/IP)

 eemcpn

次のライブラリ関数を追加した。

- ee_mcp_actcn_to
- CBLEEMCP('ACTCNTO ')

次のライブラリ関数 (C 言語) の説明を変更した。

- トランザクションインターフェース情報
- ee_mcp_actcn

次のライブラリ関数 (COBOL 言語) の説明を変更した。

- トランザクションインターフェース情報
- CBLEEMCP('ACTCN ')

次のコマンドの説明を変更した。

- eemcpactcn

次のコマンドを追加した。

- eemcpactcnto

任意のコネクションで、相手システムとの間で、メッセージの送信または受信がない無通信状態を監視できるようにした。

これに伴い、次の MCP 構成定義を変更した。

コネクション定義 (TCP/IP)

 notrftime

次のライブラリ関数 (C 言語) の説明を変更した。

- トランザクションインターフェース情報

論理端末の閉塞解除が成功した場合の処理と、論理端末の状態を確認する方法について説明を追加した。

MCP のシステム定義での注意事項を変更した。

MCP のエラーコード一覧の説明を追加した。

付録 D.4 変更内容 (3000-3-F57-20)

uCosminexus TP1/EE/Message Control Extension 01-04 の製品の変更内容 (3000-3-F57-20) を表に示します。

追加・変更内容

送受信するメッセージのデータ形式の変化、およびメッセージ送信先の指定について、説明を追加した。

UDP プロトコルで、メッセージ送信先を MCP 構成定義で指定できるようにした。

これに伴い、次のオペランドを追加した。

 eemcple (論理端末定義 (UDP))

 oipaddr

追加・変更内容

ohostname
oportno

また、次の関数の引数 opcd、またはデータ名 G の指定値を変更した。

- ee_mcp_send
- ee_mcp_sendsync
- CBLEEMCP('SEND ')
- CBLEEMCP('SENDSYNC')

TP1/EE 終了時のコネクションの解放形態を指定できるようにした。

これに伴い、次のオペランドを追加した。

eemcpcn (コネクション定義 (TCP/IP))

endrelease

また、次のオペランドの説明を変更した。

eemcpcn (コネクション定義 (TCP/IP))

cnrelease

MCP 後処理トランザクションを起動する場合の設定について、説明を変更した。

論理端末ごとにアプリケーションを定義できるようにした。

これに伴い、次の定義を追加した。

- eemcpfunc (論理端末アプリケーション定義)

TP1/EE のオプション製品として、TP1/FSP をサポートした。

これに伴い、次のオペランドを追加した。

eemcpfunc (論理端末アプリケーション定義)

rltrnlevel

eemcpfunc (アプリケーション定義)

rltrnlevel

outputmessage

outputfunction

次のトランザクションインターフェース情報を追加した。

- トランザクション起動要因
- サーバ再起動回数
- トランザクション起動時刻
- 送信元のノード識別子
- トランザクションレベル名長
- トランザクションレベル名

次の関数の説明を変更し、注意事項を追加した。

- ee_mcp_send
- ee_mcp_sendsync
- CBLEEMCP('SEND ')
- CBLEEMCP('SENDSYNC')

次のコマンドの出力形式を変更した。

- eemcpdump (MCP トレースの出力)

追加・変更内容

- eemcped (MCP トレースの編集)

また、次のトランザクション種別を追加した。

UT, CU, JI

オンライン開始時の自動起動による MCP 後処理トランザクションが起動しない場合の条件を追加した。

エラートランザクションの起動について、説明を追加した。

メッセージ受信時のコネクション以外の場合（送信元でコネクションが再確立された場合）でも、応答メッセージを送信するかどうかを指定できるようにした。

これに伴い、次のオペランドを追加した。

eemcpfunc (アプリケーション定義)

reconnectsend

MCP 定義オブジェクト生成ユティリティの出力メッセージを追加した。

また、定義オブジェクトファイルの互換性について、説明を追加した。

定義とオペランドの追加に伴い、システム構成例、および定義のコーディング例を変更した。

TP1/EE の各製品のトランザクションで使用できる MCP のライブラリ関数について、説明を追加した。

UOC の組み込みについて、説明を追加した。

UOC で使用できる関数に、TP1/EE の現在時刻取得関数 (ee_thd_gettime) を追加した。

別プロセスによるトランザクション回復機能を削除した。これに伴い、関連するコマンドを削除した。

各バージョンの変更内容を追加した。

付録 D.5 変更内容 (3000-3-F57-10)

uCosminexus TP1/EE/Message Control Extension 01-02 の製品の変更内容 (3000-3-F57-10) を表に示します。

追加・変更内容

次に示す関数で扱うメッセージの最大サイズを拡張できるようにした。

- ee_mcp_send
- ee_mcp_sendsync
- CBLEEMCP('SEND ')
- CBLEEMCP('SENDSYNC')

これに伴い、メッセージの最大サイズを拡張した場合の次のトレース取得関数を追加した。

- ee_mcp_utrace_put_long
- CBLEEMCP('UTPUTL ')

また、次に示す定義のオペランドを追加、変更した。

eemcpfunc (アプリケーション定義)

msglen

追加・変更内容

eemcptrc (トレース定義)

entrysize

さらに、次に示すコマンドの出力形式を変更した。

- eemcpdump (MCP トレースの出力)
- eemcped (MCP トレースの編集)

TP1 キャッシュ機能使用時の CL 構成に、MCP を組み込めるようにした。

これに伴い、次のオペランドを追加した。

eemcpfunc (アプリケーション定義)

recvpermanence

また、次の関数、および定義に注意事項を追加した。

- ee_mcp_send
- CBLEEMCP('SEND ')
- eemcpcn (コネクション定義 (TCP/IP))
- eemcples (論理端末定義 (UDP))
- eemcpfunc (アプリケーション定義)

MCP トレース情報の取得量を変更できるようにした。

これに伴い、次のオペランドを追加した。

eemcptrc (トレース定義)

filter

msgsize

送信結果トレースの取得有無をユーザが指定できるようにした。

これに伴い、次のオペランドを追加した。

eemcptrc (トレース定義)

sendresult

プロセス管理テーブルユーザエリア、およびスレッド対応インターフェースユーザエリアのサイズを拡張した。

これに伴い、トランザクションインターフェース情報に次の情報を追加した。

- 拡張ユーザ IFA エリアサイズ
- 拡張ユーザ SAT エリアサイズ

トランザクションの同期点を取得する処理でエラーが発生した場合の、注意事項を追加した。

受信 ID を出力することで、MCP トレースファイルと TASKTM ファイルを関連づけられるようにした。

これに伴い、次のコマンドの出力形式を変更した。

- eemcpdump (MCP トレースの出力)
- eemcped (MCP トレースの編集)

MCP トレースファイルの容量算出式を変更した。

システム統計情報として、トラブルシュート情報を取得できるようにした。

これに伴い、次のコマンドの出力形式を変更した。

- eemcpdump
- eemcped

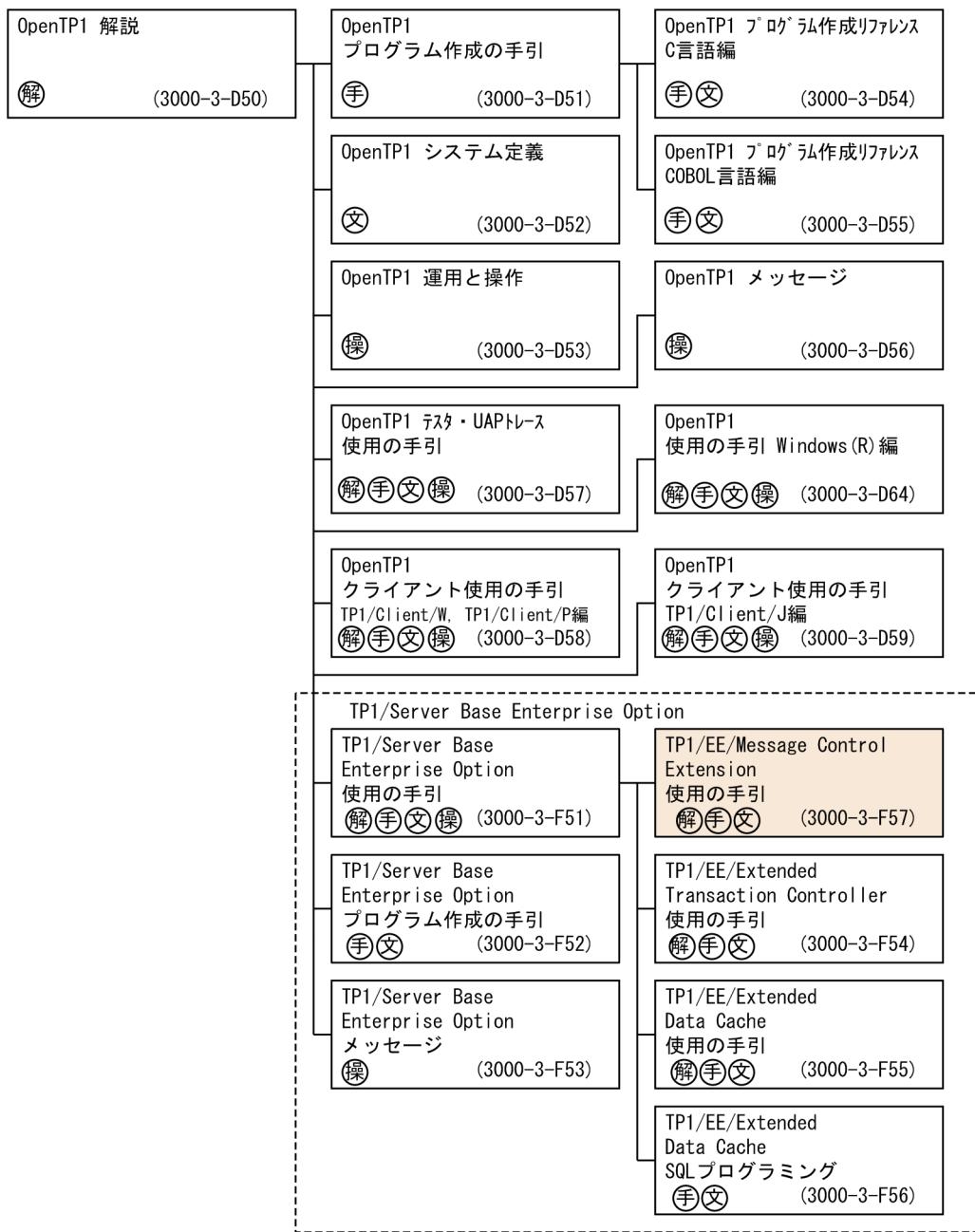
付録 E このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 E.1 関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

- OpenTP1 Version 7



〈記号〉

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

• 関連製品

- COBOL2002 使用の手引 手引編 (3000-3-D42)
- COBOL2002 使用の手引 操作編 (3000-3-D43)
- 高信頼化システム監視機能 HA モニタ Linux(R) 編 (3000-9-132)
- 高信頼化システム監視機能 HA モニタ Linux(R) (x86) 編 (3000-9-201)
- 高信頼化システム監視機能 HA モニタ メッセージ (3000-9-203)

付録 E.2 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。

略称	名称
Linux	Red Hat Enterprise Linux Server 6 (64-bit x86_64)
	Red Hat Enterprise Linux Server 7 (64-bit x86_64)
	Red Hat Enterprise Linux Server 8 (64-bit x86_64)
	Red Hat Enterprise Linux Server 9 (64-bit x86_64)
MCP	uCosminexus TP1/EE/Message Control Extension
TP1/Client	uCosminexus TP1/Client/J
	uCosminexus TP1/Client/P
	uCosminexus TP1/Client/W
TP1/EE	uCosminexus TP1/Server Base Enterprise Option
	uCosminexus TP1/Server Base Enterprise Option(64)
TP1/FSP	uCosminexus TP1/Financial Service Platform
TP1/Server Base	uCosminexus TP1/Server Base
UNIX	UNIX(R)
XDB	uCosminexus TP1/EE/Extended Data Cache
XTC	uCosminexus TP1/EE/Extended Transaction Controller

(2) インストールディレクトリのパス

このマニュアルでは、TP1/EE のインストールディレクトリを次のように表記しています。

このマニュアルでの表記	読み替え
インストールディレクトリ	/opt/OpenTP1

付録 E.3 英略語

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
API	Application Programming Interface
CSV	Comma Separated Value
DB	Database
HA	High Availability
IP	Internet Protocol
LAN	Local Area Network
OS	Operating System
RPC	Remote Procedure Call
SDB	Structured Database
SPP	Service Providing Program
TCP/IP	Transmission Control Protocol/Internet Protocol
TTL	Time To Live
UAP	User Application Program
UDP	User Datagram Protocol
UOC	User Own Coding
XA	Extended Architecture

付録 E.4 KB (キロバイト) などの単位表記について

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

(英字)

CL 連携

XTC と HA モニタの連携によって、高信頼性と高可用性を実現するクラスタリングのことです。

MCP トレース

MCP を使用して送受信したメッセージの状態に関する情報のうち、ユーザトレースの取得関数で取得した情報です。通常はイベント発生時の要求ごとに取得します。ただし、トランザクション実行中に要求された取得項目は、トランザクション終了時に一括して取得します。MCP トレースを編集、出力する場合は、MCP のコマンドを実行します。

OpenTP1 管理者

OpenTP1 を管理する、UNIX OS の利用者のことです。OpenTP1 管理者をどの利用者名称とするかは、スーパーユーザが決めます。OpenTP1 管理者は、OpenTP1 を使う上で重要な権限を持つユーザです。システムの機密保護上、利用者名称にはパスワードを必ず設定して、限られた人だけが OpenTP1 管理者の利用者名称を使えるようにしてください。

RPC (Remote Procedure Call)

→リモートプロシージャコールの説明を参照してください。

SO_KEEPALIVE

OS が提供しているソケットオプションです。SO_KEEPALIVE を使用すると、コネクションがメッセージ送受信中でない場合に、相手システムに対して一定間隔でパケットを送信し、その間に応答がなければ障害と見なしてコネクションを切断します。この機能をキープアライブといいます。

ただし、キープアライブの仕様は OS の実装によって異なり、ネットワークの負荷を高める場合があります。そのため、コネクション障害を確実に検出したい場合だけ使用してください。

SO_REUSEADDR

OS が提供しているソケットオプションです。SO_REUSEADDR を使用すると、別のプロセスによって使用中のポートであっても、ポート番号を共用することで複数のプロセスで使用できるようになります。

SPP (Service Providing Program)

OpenTP1 の UAP のうち、ファイルへのアクセスなどサーバの役割をするプログラムのことです。SPP は、クライアント UAP から要求されたサービスを実行するサービス関数と、サービス関数をまとめたメイン関数から構成されます。

TASKTM

TP1/EE が出力するトランザクションごとの統計情報です。トランザクションの開始からコミットまたはロールバックの間に、各スレッドで処理する事象発生回数や時間が、TASKTM 情報としてファイルに格納されます。TASKTM 情報を編集、出力する場合は、TP1/EE のコマンドを実行します。

TCP/IP

米国国防総省高等研究計画局 (DARPA) のプロジェクトである ARPANET が開発したプロトコルのことです。TCP/IP プロトコルは、主に LAN で使われます。TP1/EE では、クライアント／サーバ形態の通信の通信プロトコルに TCP/IP を使用します。

TCP_NODELAY

OS が提供するソケットオプションです。TCP_NODELAY を使用すると、送信済みデータの応答待ちの状態でも遅延させることなくデータ送信できるようになります。

ただし、TCP_NODELAY の仕様は OS によって異なり、ネットワークの負荷を高める場合があります。TCP_NODELAY を使用する場合は、必要性を十分に検討した上で使用するようにしてください。

UAP (User Application Program)

ユーザの業務をプログラムとして作成したものです。アプリケーションプログラムともいいます。

UOC (User Own Coding)

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ユーザがコーディングするプログラムです。起動するアプリケーション名を決定したり、UAP が送信したメッセージを他システムに送信する前に編集できます。

(ア行)

アプリケーション

TP1/EE の業務処理の総称です。アプリケーションとして作成するプログラムをアプリケーションプログラム（またはユーザアプリケーションプログラム UAP）といいます。

(力行)

コネクション

メッセージ送受信の業務で、ほかのシステムと OpenTP1 システムとの間に確立する論理的な通信路をコネクションといいます。

コミット

トランザクションの同期点を取得できたことをコミットといいます。コミットできた場合、該当するトランザクション処理が有効となります。

(サ行)

サービス

クライアント／サーバシステムでは、クライアントから要求された手続きを総称してサービスといいます。UAP のコーディング時には、C 言語の場合は関数として、COBOL 言語の場合はサブルーチンとして、サービスを作成します。作成したサービスのことを、C 言語の場合はサービス関数、COBOL 言語の場合はサービスプログラムといいます。

サービスグループ

OpenTP1 のサーバ UAP は、クライアントからの手続き要求を処理するサービスの集合です。このことから、サーバ UAP のことをサービスグループといいます。TP1/EE のリモートプロシージャコールの関数 ee_rpc_call では、サービスを要求するときにサービスグループ名とサービス名を引数に設定します。

システムサービス

TP1/EE の個々の機能を指してシステムサービスといいます。

スーパーユーザ (superuser)

UNIX OS の最高権限を持つユーザのことです。UNIX ファイルシステムのすべてのファイルに対してアクセス権を持ちます。スーパーユーザの利用者名称は root で固定されています。

(タ行)

トランザクション

ファイルからデータを読み出して、変更したデータを書き込む処理（更新処理）では、データの一貫性を保持するため、途中で分けられません。このような処理の単位をトランザクションといいます。トランザクションの処理結果は、有効にするか無効にするかのどちらかに必ず決められます。

(ナ行)

ノード

ネットワークでつながれた、TP1/EE や TP1/Server Base が稼働する一つのコンピュータのことです。

(ハ行)

プロセス

ユーザサーバ、またはTP1/EEが、OSの作業領域を使用することで生成される作業領域の処理をプロセスといいます。

ホスト

ネットワークにつながれた、TP1/EEが稼働する一つの計算機のことです。

(マ行)

マルチスレッド

一つのアプリケーションがスレッドと呼ばれる処理単位を複数生成し、並行して複数の処理を行うことをマルチスレッドといいます。TP1/EEは、TP1/Server Base機能の一部をマルチスレッドで実現します。

メッセージログ

TP1/EEシステムの稼働状況に関する情報です。稼働状況を管理したり、障害発生時の切り分けをしたりするために取得します。メッセージログファイルは、TP1/EEの起動時に、TP1/EEのサービスグループごとにUNIXファイルとしてTP1/EEが作成します。

(ラ行)

リソースマネージャ (Resource Manager)

分散処理システムでの、資源を管理する機能の総称です。DBMSはリソースマネージャです。

リモートAPI機能

クライアント側のノードにあるUAPが呼び出したAPIを、TP1/EEのプロセスで代理実行する機能をリモートAPI機能といいます。リモートAPI機能を要求するクライアント側のノードにあるUAPをrapクライアントといいます。rapクライアントが呼び出したAPIをrapサーバで実行します。TP1/EEは、rapクライアントとしても、rapサーバとしても動作します。リモートAPI機能で代理実行できるAPIは、ee_rpc_call関数、およびdc_rpc_call関数です。dc_rpc_call関数は、TP1/EEがrapサーバとして動作する場合だけ使用できます。

リモートプロシージャコール (Remote Procedure Call)

UAPを実行するプロセス間で通信する機能を、リモートプロシージャコール(RPC)といいます。TP1/EEのUAPは、ほかのシステムのUAPとRPCで通信します。TP1/Server Baseのネームサービスで、サーバUAPがあるノードのネットワークアドレスとサービスグループ名を管理しているので、RPCを使用するときは、通信相手がクライアント/サーバシステムのどのノードのUAPかを意識する必要はありません。

ロールバック

トランザクション処理を無効にする同期点処理をロールバックといいます。UAP から関数を呼び出す場合と、TP1/EE から UAP の処理を無効にする場合があります。

索引

C

CBLEEMCP('ACTCNTO ') 174
CBLEEMCP('ACTCN ') 172
CBLEEMCP('ACTLE ') 177
CBLEEMCP('CNINFO ') 179
CBLEEMCP('DCTCN ') 181
CBLEEMCP('DCTLE ') 184
CBLEEMCP('SENDRECV ') 190
CBLEEMCP('SENDSYNC') 196
CBLEEMCP('SEND ') 185
CBLEEMCP('UTPUTL ') 203
CBLEEMCP('UTPUT ') 201
CL サーバでの MCP 56
CL 連携 [用語解説] 295
COBOL-UAP 作成用プログラムの説明形式 162

E

ee_mcp_actcn 134
ee_mcp_actcn_to 136
ee_mcp_actle 139
ee_mcp_cninfo 140
ee_mcp_dctcn 143
ee_mcp_dctle 145
ee_mcp_send 147
ee_mcp_sendrecv 151
ee_mcp_sendsync 156
ee_mcp_utrace_put 159
ee_mcp_utrace_put_long 160
eemcpactcn 240
eemcpactcnto 241
eemcpactle 243
eemcpbuf 80
eemcpcn 81
eemcpdctcn 244
eemcpdctle 246
eemcpdump 247
eemcped 254

eemcpfput 262
eemcpfunc 102
eemcpgen 70
eemcpfunc 98
eemcple [TCP/IP] 91
eemcple [UDP] 93
eemcplscn 263
eemcplsle 266
eemcptcpPEND 92
eemcptcpstart 80
eemcpthdgrp 101
eemcptrc 107
eemcpudpend 101
eemcpudpstart 92

M

MCP 14
MCP 後処理トランザクションの起動タイミングと内容 49
MCP オブジェクト生成ユーティリティ 67
MCP 拡張トランザクションインターフェース情報 (COBOL 言語) 168
MCP 拡張トランザクションインターフェース情報 (C 言語) 129
MCP 構成定義オブジェクトファイル 67
MCP 構成定義オブジェクトファイル [格納場所] 68
MCP 構成定義オブジェクトファイルの生成 70
MCP 構成定義の一覧 72
MCP 構成定義の記述順序 69
MCP 構成定義ファイル 67
MCP で使用できるユーザオウンコーディング (UOC) 28
MCP トレースの強制ファイル出力 [運用コマンド] 262
MCP トレースの出力 [運用コマンド] 247
MCP トレースの編集 [運用コマンド] 254
MCP トレースファイル 235
MCP トレースファイルの運用 235

MCP トレースファイルの削除 237
MCP トレースファイルの作成 237
MCP トレースファイルの保存 237
MCP トレースファイルの容量設定 236
MCP トレース [用語解説] 295
MCP のエラーコード一覧 280
MCP のライブラリ関数の一覧 120

○

OpenTP1 管理者 [用語解説] 295

R

Remote Procedure Call [用語解説] 298
Resource Manager [用語解説] 298
RPC [用語解説] 295

S

SO_KEEPALIVE 89
SO_KEEPALIVE [用語解説] 295
SO_REUSEADDR 96
SO_REUSEADDR [用語解説] 295
SPP [用語解説] 295
superuser [用語解説] 297

T

TASKTM ファイル 235
TASKTM [用語解説] 296
TCP_NODELAY 88
TCP_NODELAY [用語解説] 296
TCP/IP 定義の開始 80
TCP/IP 定義の終了 92
TCP/IP [メッセージ送信先の指定] 26
TCP/IP [用語解説] 296
TP1 キャッシュ機能との連携 56

U

UAP [用語解説] 296
UDP 定義の開始 92
UDP 定義の終了 101

UDP [メッセージ送信先の指定] 27
UOC 205
UOC インタフェース 205
UOC で使用できる関数 205
UOC の組み込み 205
UOC の実行タイミング 206
UOC [用語解説] 296
UOC を使用する場合の手順 205

あ

相手アドレスのチェックを抑止しているコネクション 40
相手システムを指定したコネクション確立機能 33
相手システムを指定したコネクションの確立 241
相手システムを指定したコネクションの確立 [COBOL 言語] 174
相手システムを指定したコネクションの確立 [C 言語] 136
アプリケーション定義 102
アプリケーション [用語解説] 296

い

一方受信メッセージ 17
一方送信メッセージ 15
インストール 63
インストールディレクトリとファイル 278

う

運用コマンド 239
運用コマンド実行機能 283
運用コマンドの一覧 239
運用コマンドの記述形式 239

え

永続メッセージ 56
エラートランザクション 51

お

オブジェクトファイル 67
オンライン終了時 53

か

開始と終了 234
環境設定の手順 63
関数の説明形式 123

き

キープアライブ 89
記述順序 68

く

クライアント型コネクション 30
クライアント型コネクションの確立 31

こ

後続メッセージの時間監視 (TCP/IP プロトコル使用時) 55
コーディング例 115
コネクション 30
コネクション解放処理中 (OBS) 30
コネクション確立 UOC 230
コネクション確立 UOC インタフェース 230
コネクション確立処理中 (BID) 30
コネクション確立中 (ACT) 30
コネクション障害 (TCP/IP) 269
コネクション障害時 53
コネクション状態表示 [運用コマンド] 263
コネクション情報の取得 [COBOL 言語] 179
コネクション情報の取得 [C 言語] 140
コネクション定義 (TCP/IP) 81
コネクションと論理端末の関係 43
コネクションの解放 35
コネクションの解放 [COBOL 言語] 181
コネクションの解放 [C 言語] 143
コネクションの解放 [運用コマンド] 244
コネクションの解放形態 38
コネクションの確立 31
コネクションの確立 [COBOL 言語] 172
コネクションの確立 [C 言語] 134
コネクションの確立 [運用コマンド] 240

コネクションの形態と通信構成定義との関係 110

コネクションの再確立処理 33
コネクションの自動解放 35
コネクションの自動確立 31
コネクションの手動解放 37
コネクションの手動確立 32
コネクションの状態遷移 31
コネクションの状態と状態遷移 30
コネクションの状態の意味 30
コネクション未確立 (INA) 30
コネクション [用語解説] 296
コネクションリプレース機能 38
コミット [用語解説] 297

さ

サーバ型コネクション 30
サーバ型コネクションの確立 35
サービス関数 (SPP) の作成 123
サービスグループ [用語解説] 297
サービスプログラム (SPP) の作成 162
サービス [用語解説] 297
再確立処理 33

し

時間監視 54
システム構成例 114
システムサービス [用語解説] 297
システム定義での注意事項 112
システム定義の設定 64
自動確立 31
受信障害 (TCP/IP) 269
受信障害 (UDP) 274
出力メッセージ編集 UOC 223
出力メッセージ編集 UOC インタフェース 224
手動確立 32

す

スーパユーザ [用語解説] 297
ステータスファイル 235

スレッドグループ定義 101

そ

送受信するメッセージの最大サイズ 24
送受信するメッセージのデータ形式 23
送信障害 (TCP/IP) 271
送信障害 (UDP) 275
送信処理中の論理端末に対するメッセージ送信 47

た

端末キュー機能 52

て

定義例 114

と

同期一方送信 15
同期一方送信 [処理の流れ] 16
同期一方送信メッセージの送信 [COBOL 言語] 196
同期一方送信メッセージの送信 [C 言語] 156
同期型メッセージ 18
同期型メッセージの送受信 [COBOL 言語] 190
同期型メッセージの送受信 [C 言語] 151
統計情報ファイル 235
トランザクションインターフェース情報 [COBOL 言語] 165
トランザクションインターフェース情報 [C 言語] 126
トランザクションの起動の失敗 (TCP/IP) 272
トランザクションの起動の失敗 (UDP) 276
トランザクション [用語解説] 297
トレース定義 107

に

入力セグメント判定 UOC 206
入力セグメント判定 UOC インタフェース 207
入力メッセージ編集 UOC 214
入力メッセージ編集 UOC インタフェース 216

の

ノード [用語解説] 297

は

ハーフコネクションに関連する障害 273
バッファ定義 80

ひ

非永続メッセージ 56
非同期一方送信 15
非同期一方送信 [処理の流れ] 16
非同期一方送信メッセージの送信 [COBOL 言語] 185
非同期一方送信メッセージの送信 [C 言語] 147

ふ

ファイルの運用 235
プロセス [用語解説] 298

へ

閉塞 46
閉塞解除 47

ほ

ポートフリーのコネクション 40
ホスト [用語解説] 298

ま

マルチスレッド [用語解説] 298

む

無通信状態監視 41

め

メッセージ送信先の指定 26
メッセージの永続化 58
メッセージの送信完了待ち時間の監視 (TCP/IP プロトコル使用時) 54
メッセージログファイル 235
メッセージログ [用語解説] 298
メモリダンプファイル 235

Φ

- ユーザオウンコーディング 205
- ユーザトレースの取得 (データ長拡張用) [COBOL 言語] 203
- ユーザトレースの取得 (データ長拡張用) [C 言語] 160
- ユーザトレースの取得 [COBOL 言語] 201
- ユーザトレースの取得 [C 言語] 159

ら

- ライブラリ関数 (COBOL 言語) 162
- ライブラリ関数 (C 言語) 123

り

- リソースマネージャ [用語解説] 298
- リモート API 機能 [用語解説] 298
- リモートプロシージャコール [用語解説] 298

ろ

- ロールバック [用語解説] 299
- 論理端末アプリケーション定義 98
- 論理端末定義 (TCP/IP) 91
- 論理端末定義 (UDP) 93
- 論理端末の管理 43
- 論理端末の状態 46
- 論理端末の状態遷移 46
- 論理端末の状態と状態遷移 45
- 論理端末の状態の意味 45
- 論理端末の状態表示 [運用コマンド] 266
- 論理端末の定義数 112
- 論理端末の閉塞 46
- 論理端末の閉塞 [COBOL 言語] 184
- 論理端末の閉塞 [C 言語] 145
- 論理端末の閉塞 [運用コマンド] 246
- 論理端末の閉塞解除 47
- 論理端末の閉塞解除 [COBOL 言語] 177
- 論理端末の閉塞解除 [C 言語] 139
- 論理端末の閉塞解除 [運用コマンド] 243
- 論理端末の閉塞解除状態 46

論理端末の閉塞解除処理中 (BID) 45

論理端末の閉塞解除中 (ACT) 45

論理端末の閉塞解除の失敗 (UDP) 274

論理端末の閉塞状態 46

論理端末の閉塞処理中 (OBS) 45

論理端末の閉塞中 (INA) 45

論理端末の閉塞と閉塞解除 46

論理端末閉塞時 (TCP/IP プロトコル使用時) 53

論理端末閉塞時 (UDP プロトコル使用時) 53