

OpenTP1 Version 7
分散トランザクション処理機能

OpenTP1 プロトコル TP1/NET/XMAP3 編

解説・手引・文法・操作書

3000-3-D74-32

前書き

■ 対象製品

- ・適用 OS : AIX V6.1, AIX V7.1, AIX V7.2

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

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

P-F1M64-32415 uCosminexus TP1/NET/XMAP3 07-50

- ・適用 OS : HP-UX 11i V3 (IPF)

P-1J64-3171 uCosminexus TP1/Message Control 07-51

P-1J64-3271 uCosminexus TP1/NET/Library 07-51

P-F1J64-32715 uCosminexus TP1/NET/XMAP3 07-51

- ・適用 OS : Windows 7, Windows 7 x64 Edition, Windows 8, Windows 8 x64 Edition, Windows 8.1, Windows 8.1 x64 Edition, Windows 10, Windows 10 x64 Edition, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019

P-2464-3164 uCosminexus TP1/Message Control 07-51

P-2464-3264 uCosminexus TP1/NET/Library 07-50

P-F2464-32645 uCosminexus TP1/NET/XMAP3 07-51

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

■ 輸出時の注意

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

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI, DCCM, OpenTP1, SEWB, uCosminexus, VOS1, VOS3, XDM, XMAP は、株式会社日立製作所の商標または登録商標です。

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

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

Itanium は、Intel Corporation またはその子会社の商標です。

Microsoft は、マイクロソフト 企業グループの商標です。

UNIX は、The Open Group の登録商標です。

Windows は、マイクロソフト 企業グループの商標です。

Windows Server は、マイクロソフト 企業グループの商標です。

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

■ 発行

2022 年 4 月 3000-3-D74-32

■ 著作権

All Rights Reserved. Copyright (C) 2006, 2022, Hitachi, Ltd.

変更内容

変更内容 (3000-3-D74-32) uCosminexus TP1/Message Control 07-51, uCosminexus TP1/NET/Library 07-51, uCosminexus TP1/NET/XMAP3 07-51

追加・変更内容	変更箇所
コンパイラオプション -VOSCBL,DataComm を指定した場合の説明を追加した。	4. COBOL-UAP 作成用プログラムインタフェース COBOL-UAP 作成用プログラムインタフェースの一覧
マッピングサービス属性定義のヌルクリアコード、データ有無コード、初期設定値の先頭「0x」指定の説明を追加した。	6. OpenTP1 マッピングサービス機能 マッピングサービス属性定義

単なる誤字・脱字などはお断りなく訂正しました。

変更内容 (3000-3-D74-31) uCosminexus TP1/Message Control 07-51, uCosminexus TP1/NET/Library 07-51, uCosminexus TP1/NET/XMAP3 07-51

追加・変更内容
マニュアル訂正の内容を反映した。
マッピングサービスホスト名の説明に IP アドレスとのマッピングが必要であることを追加した。これに伴い、複数のマッピングサービスを使用する場合の定義例を追加した。
出力形式の dddd に、ほかのステータスが出力されたときの説明を追加した。

変更内容 (3000-3-D74-30) uCosminexus TP1/Message Control 07-50, uCosminexus TP1/NET/Library 07-50, uCosminexus TP1/NET/XMAP3 07-50

追加・変更内容
マニュアル訂正の内容を反映した。

変更内容 (3000-3-D74-20) uCosminexus TP1/Message Control 07-50, uCosminexus TP1/NET/Library 07-50, uCosminexus TP1/NET/XMAP3 07-50

追加・変更内容
接続の確立の動作モードを指定できるようにした。これに伴い、XMAP3 共通定義 (mcftxp) に -c オプションを追加した。
論理端末画面の閉じるボタンを有効化する場合の留意事項を追加した。
問い合わせ応答形態の説明を追加した。
継続問い合わせ応答形態の説明を追加した。
強制確定キーの動作の注意事項を追加した。

追加・変更内容
NULL またはヌル文字列設定時のコーディング例を追加した。
リターン値 DCMCFRTN_NOMSG およびステータスコード 70904 の説明を変更した。
初期状態での受け取り領域の説明を追加した。
アプリケーション属性定義との関連について説明を追加した。
ERREVT3 に設定するトランザクションブランチ ID の形式の説明を追加した。
次のオペランドの指定値の上限を拡張した。 <ul style="list-style-type: none"> システムサービス共通情報定義 <pre>max_socket_descriptors max_open_fds</pre>
MCF 性能検証用トレースの説明を追加した。
定義オブジェクトファイルの出力先に読み取り権限を持つファイルがすでに存在する場合、定義オブジェクトファイルを上書きするかどうかを指定できるようにした。
OpenTP1 システムを変更する場合に、見直しが必要な定義と、変更手順について説明を追加した。
アプリケーション起動サービスに関する説明を追加した。
スタート関数を呼び出したあとの注意事項を追加した。
次に示すバージョンの変更点を記載した。 <ul style="list-style-type: none"> TP1/NET/XMAP3 07-50
バージョン 6 以前から移行する場合のインタフェースの変更一覧を追加した。
メッセージを送受信するときのデータの流れ、ジャーナルの取得タイミングなどの説明を追加した。

変更内容 (3000-3-D74-10) uCosminexus TP1/Message Control 07-03, uCosminexus TP1/NET/Library 07-04, uCosminexus TP1/NET/XMAP3 07-01

追加・変更内容
<p>接続の確立、解放、および状態表示を、ライブラリ関数および COBOL-UAP 作成用プログラムインタフェースでできるようにした。</p> <p>これに伴い、次の関数を追加した。</p> <ul style="list-style-type: none"> dc_mcf_tactcn dc_mcf_tdctcn dc_mcf_tlscn CBLDCMCF('TACTCN△△') CBLDCMCF('TDCTCN△△') CBLDCMCF('TLSCN△△△')
<p>論理端末の閉塞、閉塞解除、および状態表示を、ライブラリ関数および COBOL-UAP 作成用プログラムインタフェースでできるようにした。</p>

追加・変更内容

これに伴い、次の関数を追加した。

- dc_mcf_tactle
- dc_mcf_tdctle
- dc_mcf_tlsle
- CBLDCMCF('TACTLE△△')
- CBLDCMCF('TDCTLE△△')
- CBLDCMCF('TLSLE△△△')

論理端末画面の閉じるボタンを有効化できるようにした。

これに伴い、XMAP3 共通定義 (mcftxp) に-g オプションと closebutton オペランドを追加した。

送受信できる一つのセグメントの最大長について説明を追加した。

アプリケーション名の決定について説明を追加した。

システムサービス情報定義に、環境変数 XPCLOSEWAITTIME を追加した。

リターン値 (数値) の説明を追加した。

TP1/NET/XMAP3 の論理端末画面で入出力されたメッセージを送受信する場合の説明を追加した。

マップ名を引き継がない場合の説明を追加した。

出力通番の設定に関する説明を追加した。

データ操作言語の通信文と C 言語のライブラリ関数の対応の説明を追加した。

UAP で値を設定するデータ領域、および OpenTP1 から値が返されるデータ領域について、説明を追加した。また、送信するセグメントを設定する領域、およびセグメントを受信する領域の形式とその内容を追加した。

次のデータ操作言語の通信文の通信記述項に、SYNCHRONOUS MODE 句を追加した。

- RECEIVE - メッセージの受信
- SEND - メッセージの送信

ステータスコード「72016」および「72042」を追加した。

送信メッセージ通番編集 UOC インタフェースを追加した。

MCF イベントの VCLSEVT, および VERREVT が起動される条件の説明を追加した。

MCF イベント情報のマップ名に関する説明を追加した。

マッピングサービス属性定義の PADCHR オペランドの初期設定値について説明を追加した。

注意事項の、バッファグループ定義 (mcftbuf) で接続ごとに割り当てる資源の量について、説明を変更した。

システムサービス情報定義の環境変数 XRESPONSETIME の指定範囲を変更し、デフォルト値を明記した。

MCF 定義オブジェクトの解析コマンドを追加した。

MCF トレースファイルの見積もり式の説明を追加した。

マッピングサービス定義ファイルとプロトコル固有定義のファイルの制約事項について、説明を追加した。

追加・変更内容
出力形式の SYNC および IO に、未処理送信メッセージに関する説明を追加した。
オンラインコマンド使用時の注意事項，入力形式の説明を追加した。
バージョンアップ時の，定義，関数，コマンドおよびデフォルト値の変更点を追加した。
MCF 性能検証用トレースの出力情報，取得タイミング，および取得量の説明を追加した。

第 2 版では，第 1 版の目次の構成を変更しました。第 1 版との対応は次のようになっています。

第 1 版	第 2 版
1. 概要	1. 概要
1.1 対話オンライン処理の概要	1.1 対話オンライン処理の概要
1.2 対話オンライン処理の通信形態	1.2 対話オンライン処理の通信形態
1.3 ソフトウェアの構成	削除
1.3.1 前提プログラム	削除
1.3.2 ソフトウェア構成の例	1.3 ソフトウェア構成の例
2. 機能	2. 機能
2.1 TP1/NET/XMAP3 の論理端末構成	1.2.1 コネクションと論理端末の関係
2.2 オンラインの開始と終了	2.1(1) オンラインの開始形態と終了形態
2.3 対話オンライン処理の仕組み	削除
2.3.1 表示印刷サービスとの接続と切り離し	2.2 コネクションに関する機能 2.2.1 コネクションの確立 2.2.2 コネクションの解放
2.3.2 論理端末の起動と閉塞	2.3 論理端末に関する機能 2.3.2 論理端末のキュー状態の閉塞と閉塞解除 2.3.3 論理端末の端末状態の閉塞と閉塞解除
2.3.3 論理端末とアプリケーションの属性との関係	2.3.1 論理端末とアプリケーションの型の関係
2.3.4 メッセージの分割と組み立て	2.4.5 メッセージの分割と組み立て
2.4 対話オンライン処理のメッセージ送受信	2.4 メッセージ送受信に関する機能
2.4.1 マッピングによるメッセージの編集	2.4.6 マッピングによるメッセージの編集
2.4.2 通信形態とメッセージの流れ	2.4.1 非問い合わせ応答形態のメッセージ 2.4.2 分岐送信形態のメッセージ 2.4.3 問い合わせ応答形態のメッセージ 2.4.4 継続問い合わせ応答形態のメッセージ
2.5 メッセージグループ送信機能	2.4.8 メッセージグループ送信機能

第1版	第2版
2.5.1 メッセージグループ送信の流れ	2.4.8 メッセージグループ送信機能
2.5.2 プリンタの連続使用	
2.6 ページ制御機能	2.4.9 ページ制御機能 (XMAP3/PRINTER UTILITY) の利用
2.6.1 ページ制御機能の概要	
2.7 代行送信	2.4.10 メッセージの代行送信
2.8 XP/W を使用する場合の他システムとの接続	2.2.4 XP/W を使用する場合の他システムとの接続
2.9 キー入力サービス	2.4.11 キー入力サービス
2.9.1 送信要求発生時の処理	
2.9.2 印刷キー処理	
2.9.3 キーボード状態の変更	
2.10 システムコマンド	2.4.12 オンラインコマンド
2.11 オペレータインジケータ表示メッセージ	2.4.13 オペレータインジケータのメッセージ表示
2.12 クライアント PC/WS 応答時間タイムアウト発生時の障害メッセージ出力	2.4.14 クライアント PC/WS 応答時間タイムアウト発生時の障害メッセージ出力
2.12.1 機能の説明	
2.12.2 環境変数	
2.13 注意事項	削除
2.13.1 XMAP3 サーバの起動に関する注意事項	2.1(2) XMAP3 サーバの起動
2.13.2 旧製品からの移行に関する注意事項	付録 B 旧製品からの移行に関する注意事項
3. メッセージ送受信インタフェース	3. C 言語のライブラリ関数 4. COBOL-UAP 作成用プログラムインタフェース 2.4.7 アプリケーションの決定 付録 G ユーザアプリケーションプログラムの作成例
4. ユーザOWNコーディング, MCF イベントインタフェース	5. ユーザOWNコーディング, MCF イベントインタフェース
5. OpenTP1 マッピングサービス機能	6. OpenTP1 マッピングサービス機能
6. システム定義	7. システム定義
7. 運用コマンド	8. 運用コマンド
8. オンラインコマンド	9. オンラインコマンド
9. 組み込み方法	10. 組み込み方法
10. 障害対策	11. 障害対策

第 1 版	第 2 版
付録 A XMAP3 Server サービス名との関係	付録 D XMAP3 Server サービス名との関係
付録 B 表示印刷サービスの起動でコネクション確立をするための環境設定	付録 E 表示印刷サービスの起動でコネクション確立をするための環境設定
付録 C OpenTP1 マッピングサービス機能の定義例 (PAGEC 機能を使用する場合)	付録 F OpenTP1 マッピングサービス機能の定義例 (PAGEC 機能を使用する場合)
付録 D オペレータインジケータ表示メッセージ一覧	付録 H オペレータインジケータ表示メッセージ一覧
付録 E 理由コード一覧	付録 I 理由コード一覧
付録 F マッピングサービスの障害コードの詳細	付録 J マッピングサービスの障害コードの詳細
付録 G 用語解説	付録 K 用語解説

はじめに

このマニュアルは、TP1/NET/XMAP3を使用した対話オンライン処理システムの概要、機能、作成方法、および操作・運用方法について説明したものです。

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

■ 対象読者

システム管理者、システム設計者、およびプログラマを対象としています。次のマニュアルを理解していることを前提としています。

- OpenTP1 解説 (3000-3-D50)
- OpenTP1 プログラム作成の手引 (3000-3-D51)
- OpenTP1 システム定義 (3000-3-D52)
- OpenTP1 運用と操作 (3000-3-D53)
- OpenTP1 プログラム作成リファレンス C 言語編 (3000-3-D54)
- OpenTP1 プログラム作成リファレンス COBOL 言語編 (3000-3-D55)
- OpenTP1 使用の手引 Windows(R)編 (3000-3-D64)

また、XMAP3 の知識をお持ちであることを前提としています。

■ マニュアルの構成

このマニュアルは、次に示す章と付録から構成されています。

第 1 章 概要

TP1/NET/XMAP3 の概要について説明しています。

第 2 章 機能

TP1/NET/XMAP3 の機能について説明しています。

第 3 章 C 言語のライブラリ関数

TP1/NET/XMAP3 で使用できる、C 言語のライブラリ関数について説明しています。

第 4 章 COBOL-UAP 作成用プログラムインタフェース

TP1/NET/XMAP3 で使用できる、COBOL-UAP 作成用プログラムインタフェースについて説明しています。

第 5 章 ユーザOWNコーディング, MCF イベントインタフェース

ユーザOWNコーディング, および MCF イベントインタフェースについて説明しています。

第 6 章 OpenTP1 マッピングサービス機能

OpenTP1 マッピングサービス機能について説明しています。

第 7 章 システム定義

TP1/NET/XMAP3 のシステム定義について説明しています。

第 8 章 運用コマンド

TP1/NET/XMAP3 の運用コマンドについて説明しています。

第 9 章 オンラインコマンド

TP1/NET/XMAP3 のオンラインコマンドについて説明しています。

第 10 章 組み込み方法

TP1/NET/XMAP3 の組み込み方法について説明しています。

第 11 章 障害対策

TP1/NET/XMAP3 の障害時の処理について説明しています。

付録 A バージョンアップ時の変更点

各バージョンでの関数, 定義およびコマンドの変更点について説明しています。

付録 B 旧製品からの移行に関する注意事項

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

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

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

付録 D MCF 性能検証用トレースの取得

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

付録 E メッセージ送受信の流れ

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

付録 F XMAP3 Server サービス名との関係

TP1/NET/XMAP3 の定義と XMAP3 Server サービス名との関係について説明しています。

付録 G 表示印刷サービスの起動でコネクション確立をするための環境設定

表示印刷サービスの起動でコネクション確立をするための環境設定について説明しています。

付録 H ユーザアプリケーションプログラムの作成例

TP1/NET/XMAP3 のユーザアプリケーションプログラムの作成例について説明しています。

付録 I オペレータインジケータ表示メッセージ一覧

オペレータインジケータ表示メッセージについて説明しています。

付録 J 理由コード一覧

障害通知イベントが発生した場合の理由コードについて説明しています。

付録 K マッピングサービスの障害コードの詳細

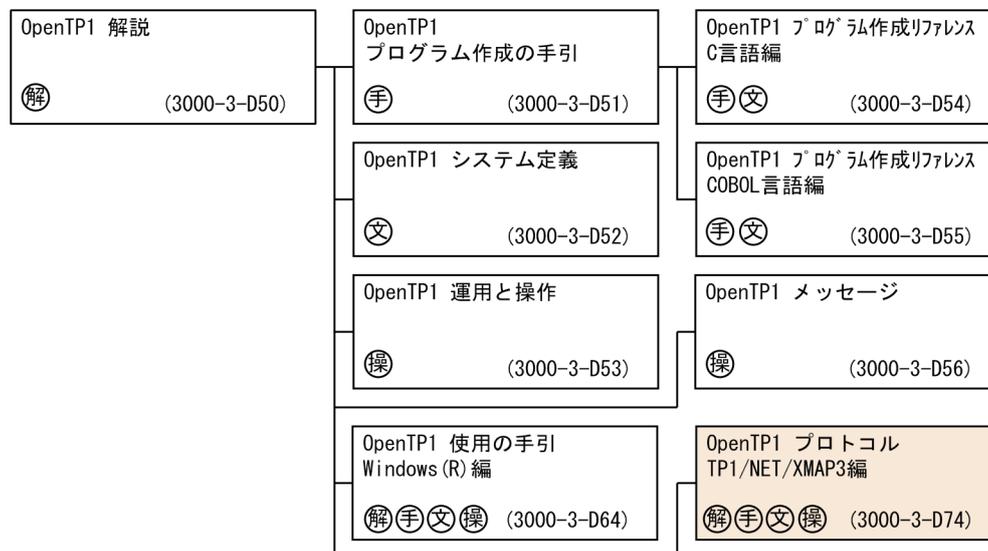
マッピングサービスの障害コードの詳細について説明しています。

付録 L 用語解説

TP1/NET/XMAP3 で使用する用語について説明しています。

■ 関連マニュアル

●OpenTP1 Version 7



<記号>

解 : 解説書

手 : 手引書

文 : 文法書

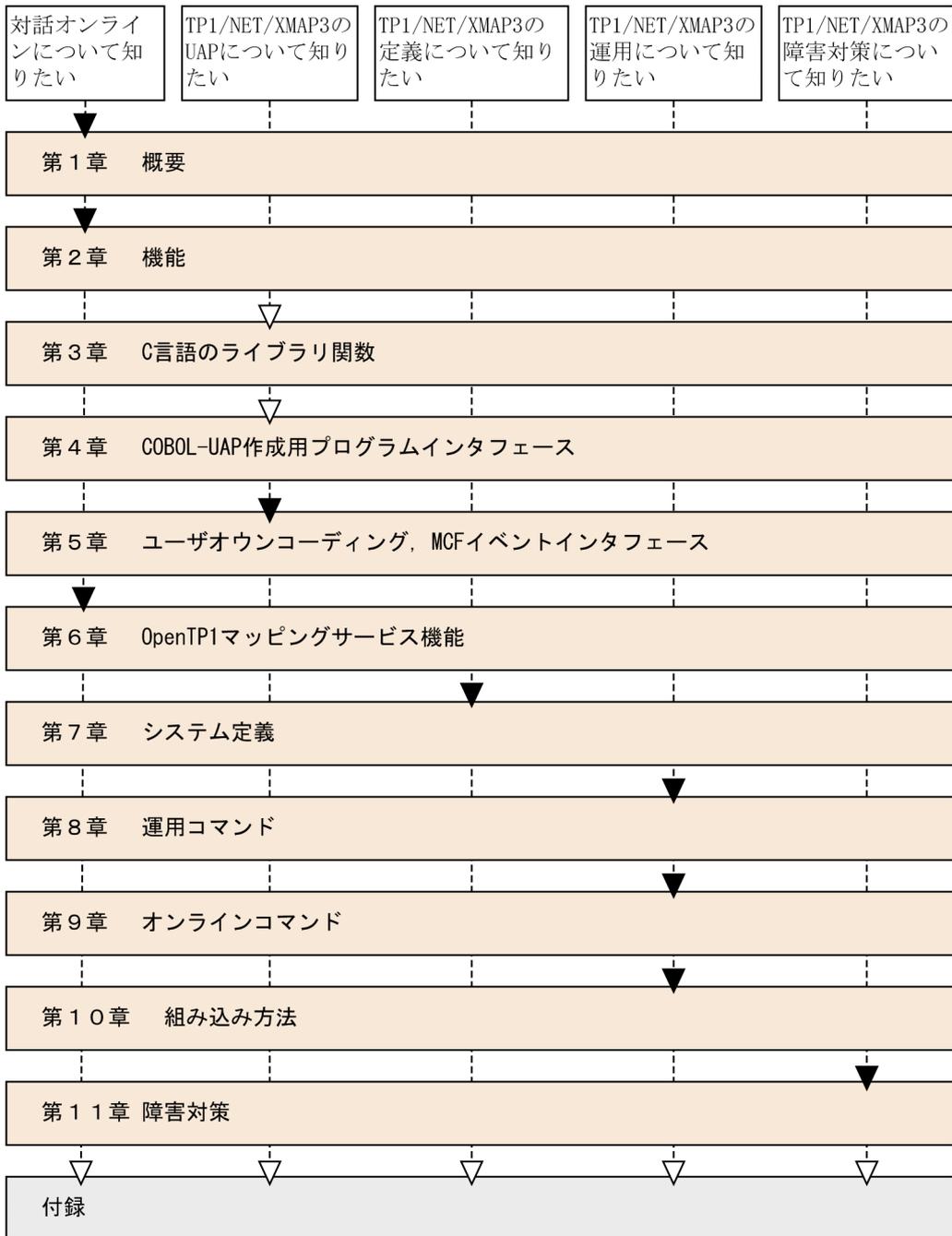
操 : 操作書

●サービスプログラム

- COBOL2002 使用の手引 手引編 (3000-3-D08)
- XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド (3020-7-512)
- XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド (3020-7-513)
- XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド (3020-7-514)

■ 読書手順

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



(凡例) ▼ : 必ず読む項目 ▽ : 必要に応じて読む項目

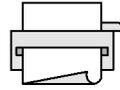
■ 図中で使用する記号

このマニュアルの図中で使用する記号を、次のように定義します。

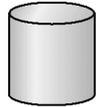
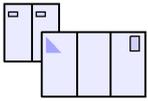
●ワークステーション、
●論理端末
●論理端末画面



●プリンタ



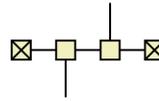
●ホストコンピュータ ●ファイル



●ネットワーク (WAN)



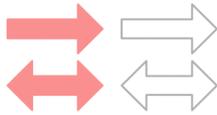
●ネットワーク (LAN)



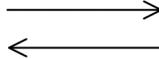
●プログラムの流れ



●データの流れ,
メッセージの流れ



●制御の流れ



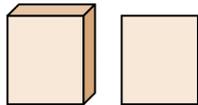
●通信回線



●入出力の動作



●プログラム



■ 文法の記号

このマニュアルで使用する各種の記号を説明します。

(1) 文法記述記号

文法の記述形式について説明する記号です。

文法記述記号	意味
[]	この記号で囲まれている項目は省略できることを示します。 (例) [-s MCF 通信プロセス識別子] -s オプションとそのオペランドを指定するか、何も指定しないことを示します。
 (ストローク)	この記号で仕切られた項目は選択できることを示します。 (例) -t reply request -t オプションに reply または request を指定できることを示します。 ただし、C 言語のインタフェースの説明でこの記号を使用した場合は、C 言語の文法規則に従います。
{ }	この記号で囲まれている複数の項目のうちから一つを選択できることを示します。

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

(2) 属性表示記号

ユーザ指定値の範囲などを説明する記号です。

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

(3) 構文要素記号

ユーザ指定値の内容を説明する記号です。

構文要素記号	意味
英字	アルファベット (A~Z, a~z) と_ (アンダスコア)
英字記号	アルファベット (A~Z, a~z) と#, @, ¥
英数字	英字と数字 (0~9)
英数字記号	英字記号と数字 (0~9)
符号なし整数	数字列 (0~9)
10 進数字	数字 (0~9)

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

■ このマニュアルでの表記

(1) 製品名

このマニュアルで使用する製品名称の略称を次に示します。

製品名称	略称	
AIX V6.1	AIX	
AIX V7.1		
AIX V7.2		
COBOL2002	COBOL	
COBOL85		
HP-UX 11i V2 (IPF)	HP-UX	
HP-UX 11i V3 (IPF)		
Itanium Processor Family	IPF	
Solaris 8	Solaris	
Solaris 9		
Solaris 10		
uCosminexus TP1/Message Control	TP1/Message Control	
uCosminexus TP1/NET/Library	TP1/NET/Library	
uCosminexus TP1/NET/XMAP3	TP1/NET/XMAP3	
uCosminexus TP1/Server Base	TP1/Server Base	
Windows 7 Enterprise (x86)	Windows 7 x86	Windows 7
Windows 7 Professional (x86)		
Windows 7 Ultimate (x86)		

製品名称	略称	
Windows 7 Enterprise (x64)	Windows 7 x64	Windows 7
Windows 7 Professional (x64)		
Windows 7 Ultimate (x64)		
Windows 8 Enterprise (x86)	Windows 8 x86	Windows 8
Windows 8 Pro (x86)		
Windows 8 Enterprise (x64)	Windows 8 x64	
Windows 8 Pro (x64)		
Windows 8.1 Enterprise (x86)	Windows 8.1 x86	Windows 8.1
Windows 8.1 Pro (x86)		
Windows 8.1 Enterprise (x64)	Windows 8.1 x64	
Windows 8.1 Pro (x64)		
Windows 10 Enterprise (x86)	Windows 10 x86	Windows 10
Windows 10 Pro (x86)		
Windows 10 Enterprise (x64)	Windows 10 x64	
Windows 10 Pro (x64)		
Windows Server 2008 R2 Datacenter	Windows Server 2008 R2	
Windows Server 2008 R2 Enterprise		
Windows Server 2008 R2 Standard		
Windows Server 2012 Datacenter	Windows Server 2012	
Windows Server 2012 Standard		
Windows Server 2012 R2 Datacenter	Windows Server 2012 R2	
Windows Server 2012 R2 Standard		
Windows Server 2016 Datacenter	Windows Server 2016	
Windows Server 2016 Standard		
Windows Server 2019 Datacenter	Windows Server 2019	
Windows Server 2019 Standard		
Extended Mapping Service 3	XMAP3	

- Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, および Windows Server 2016 で機能差がない場合、Windows と表記しています。
- AIX, HP-UX, および Solaris を総称して UNIX と表記しています。

(2) 適用 OS による違いについて

Windows 版の製品をご使用になる場合、マニュアルの記述を次のように読み換えてください。

項目	マニュアルの表記	読み換え
環境変数の表記	\$aaaaaa 例 \$DCDIR	%aaaaaa% 例 %DCDIR%
複数のパス名を列挙するときの区切り文字	:	;
ディレクトリの区切り文字	/	¥
完全パス名	ルートディレクトリから指定します。 例 /tmp	先頭にドライブ文字を付加して、ルートディレクトリから指定します。 例 C:¥tmp
実行形式ファイル名	ファイル名だけを指定します。 例 mcfmngrd	ファイル名に拡張子を付加して指定します。 例 mcfmngrd.exe
make コマンド	make	nmake

(3) 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	'ー' (オーバーライン)	'`' (チルダ)

(4) C 言語の関数と COBOL-UAP 作成用プログラムの表記について

このマニュアルでは、C 言語の関数と COBOL-UAP 作成用プログラムとで説明に差異がない場合、C 言語の関数名に統一して説明しています。

■ 略語一覧

このマニュアルで使用する英略語の一覧を次に示します。

英略語	英字での表記
CUI	<u>C</u> haracter <u>U</u> ser <u>I</u> nterface
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
LAN	<u>L</u> ocal <u>A</u> rea <u>N</u> etwork
LRU	<u>L</u> east <u>R</u> ecently <u>U</u> sed
MCF	<u>M</u> essage <u>C</u> ontrol <u>F</u> acility
MHP	<u>M</u> essage <u>H</u> andling <u>P</u> rogram
OS	<u>O</u> perating <u>S</u> ystem
PC	<u>P</u> ersonal <u>C</u> omputer
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram
UAP	<u>U</u> ser <u>A</u> pplication <u>P</u> rogram
UOC	<u>U</u> ser <u>O</u> wn <u>C</u> oding
WAN	<u>W</u> ide <u>A</u> rea <u>N</u> etwork
WS	<u>W</u> ork <u>s</u> tation

■ KB (キロバイト) などの単位表記について

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

■ 謝辞

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

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

COBOL は産業界の言語であり, 特定の団体や組織の所有物ではない。

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

次に示す著作権表示付資料の著作者及び著作権者

FLOW-MATIC (Sperry Rand Corporation の商標),

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

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

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

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

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

目次

前書き	2
変更内容	4
はじめに	10

1 概要 27

1.1	対話オンライン処理の概要	28
1.2	対話オンライン処理の通信形態	30
1.2.1	コネクションと論理端末の関係	30
1.2.2	メッセージ送受信の形態	31
1.3	ソフトウェア構成の例	33

2 機能 34

2.1	オンラインの開始と終了	35
2.1.1	オンラインの開始形態と終了形態	35
2.1.2	XMAP3 サーバの起動	36
2.2	コネクションに関する機能	37
2.2.1	コネクションの確立	37
2.2.2	コネクションの解放	40
2.2.3	コネクションの状態表示	40
2.3	論理端末に関する機能	42
2.3.1	論理端末とアプリケーションの型の関係	42
2.3.2	論理端末のキュー状態の閉塞と閉塞解除	43
2.3.3	論理端末の端末状態の閉塞と閉塞解除	44
2.3.4	論理端末の状態表示	48
2.3.5	論理端末画面の閉じるボタンの有効化	48
2.4	メッセージ送受信に関する機能	50
2.4.1	非問い合わせ応答形態のメッセージ	50
2.4.2	分岐送信形態のメッセージ	50
2.4.3	問い合わせ応答形態のメッセージ	53
2.4.4	継続問い合わせ応答形態のメッセージ	55
2.4.5	メッセージの分割と組み立て	62
2.4.6	マッピングによるメッセージの編集	63
2.4.7	アプリケーションの決定	64
2.4.8	メッセージグループ送信機能	65
2.4.9	メッセージの代行送信	70

- 2.4.10 キー入力サービス 71
- 2.4.11 オンラインコマンド 75
- 2.4.12 オペレータインジケータのメッセージ表示 75
- 2.4.13 クライアント PC/WS 応答時間タイムアウト発生時の障害メッセージ出力 75

3 C言語のライブラリ関数 77

- C言語のライブラリ関数の一覧 78
- dc_mcf_contend - 継続問い合わせ応答の終了 (C言語) 79
- dc_mcf_receive - メッセージの受信 (C言語) 81
- dc_mcf_reply - 応答メッセージの送信 (C言語) 86
- dc_mcf_resend - メッセージの再送 (C言語) 91
- dc_mcf_send - メッセージの送信 (C言語) 96
- dc_mcf_tactcn - コネクションの確立 (C言語) 100
- dc_mcf_tacttle - 論理端末の閉塞解除 (C言語) 104
- dc_mcf_tdctcn - コネクションの解放 (C言語) 107
- dc_mcf_tdcttle - 論理端末の閉塞 (C言語) 111
- dc_mcf_tempget - 一時記憶データの受け取り (C言語) 114
- dc_mcf_tempput - 一時記憶データの更新 (C言語) 117
- dc_mcf_tlscn - コネクションの状態取得 (C言語) 119
- dc_mcf_tlsle - 論理端末の状態取得 (C言語) 124

4 COBOL-UAP 作成用プログラムインタフェース 128

- COBOL-UAP 作成用プログラムインタフェースの一覧 129
- CBLDCMCF('CONTEND ') - 継続問い合わせ応答の終了 (COBOL言語) 133
- CBLDCMCF('RECEIVE ') - メッセージの受信 (COBOL言語) 135
- CBLDCMCF('REPLY ') - 応答メッセージの送信 (COBOL言語) 140
- CBLDCMCF('RESEND ') - メッセージの再送 (COBOL言語) 146
- CBLDCMCF('SEND ') - メッセージの送信 (COBOL言語) 152
- CBLDCMCF('TACTCN ') - コネクションの確立 (COBOL言語) 159
- CBLDCMCF('TACTTLE ') - 論理端末の閉塞解除 (COBOL言語) 163
- CBLDCMCF('TDCTCN ') - コネクションの解放 (COBOL言語) 166
- CBLDCMCF('TDCTTLE ') - 論理端末の閉塞 (COBOL言語) 170
- CBLDCMCF('TEMPGET ') - 一時記憶データの受け取り (COBOL言語) 173
- CBLDCMCF('TEMPPUT ') - 一時記憶データの更新 (COBOL言語) 177
- CBLDCMCF('TLSCN ') - コネクションの状態取得 (COBOL言語) 180
- CBLDCMCF('TLSLE ') - 論理端末の状態取得 (COBOL言語) 184
- DISABLE - 継続問い合わせ応答の終了 (データ操作言語) 187
- RECEIVE - メッセージの受信 (データ操作言語) 189
- RECEIVE - 一時記憶データの受け取り (データ操作言語) 194
- SEND - メッセージの送信 (データ操作言語) 197
- SEND - 一時記憶データの更新 (データ操作言語) 204

5 ユーザOWNコーディング, MCF イベントインタフェース 206

- 5.1 ユーザOWNコーディングインタフェース 207

- 5.1.1 入力メッセージの編集とアプリケーション名の決定 207
- 5.1.2 入力メッセージ編集 UOC インタフェース 209
- 5.1.3 出力メッセージの編集 215
- 5.1.4 出力メッセージ編集 UOC インタフェース 217
- 5.1.5 送信メッセージの通番編集 221
- 5.1.6 送信メッセージ通番編集 UOC インタフェース 223
- 5.1.7 UOC 作成上の注意事項 224
- 5.2 MCF イベントインタフェース 226
- 5.2.1 MCF イベントの種類 226
- 5.2.2 MCF イベント通知時のセグメント構成 228
- 5.2.3 MCF イベント情報の形式 (C 言語) 228
- 5.2.4 MCF イベント情報の形式 (COBOL 言語) 236

6 OpenTP1 マッピングサービス機能 248

- OpenTP1 マッピングサービス機能の利用形態 249
- OpenTP1 マッピングサービス機能の詳細 250
- OpenTP1 マッピングサービスに関する定義 253
- マッピングサービス定義 254
- マッピングサービス属性定義 256
- OpenTP1 マッピングサービス機能の定義例 259
- マッピングサービスコマンド 262
- dcmaphg (読み込みパス名の変更) 263
- dcmaphs (ロード済み資源の表示) 265
- 注意事項 267

7 システム定義 268

- TP1/NET/XMAP3 の定義の概要 269
- TP1/NET/XMAP3 固有のシステム定義の種類 271
- mcaalcap (アプリケーション属性定義) 276
- mcfmcomn (MCF マネージャ共通定義) 279
- mcftalccn (コネクション定義の開始) 280
- mcftalced (コネクション定義の終了) 284
- mcftalcle (論理端末定義) 285
- mcftxp (XMAP3 共通定義) 293
- システムサービス情報定義 296
- システムサービス共通情報定義 299
- MCF 定義オブジェクトの生成 303
- MCF 定義オブジェクトの解析 304
- MCF トレースファイルの見積もり式 307
- 各種定義数に関する制約事項 310
- OpenTP1 システムの変更に影響する定義 311
- 定義例 313

8 運用コマンド 316

- TP1/NET/XMAP3 の運用コマンド 317
- mcftactcn (コネクションの確立) 318
- mcftactle (論理端末の閉塞解除) 320
- mcftdctcn (コネクションの強制解放) 323
- mcftdctle (論理端末の閉塞) 326
- mcftedalt (代行送信の終了) 329
- mcftendct (継続問い合わせ応答処理の強制終了) 331
- mcftlscn (コネクションの状態表示) 334
- mcftlsle (論理端末の状態表示) 337
- mcftstalt (代行送信の開始) 341

9 オンラインコマンド 343

- TP1/NET/XMAP3 のオンラインコマンド 344
- ACT ALTSEND (代行送信の開始) 346
- DEACT ALTSEND (代行送信の終了) 347
- init (初期画面の表示) 348

10 組み込み方法 349

- 10.1 TP1/NET/XMAP3 の組み込みの流れ 350
 - 10.1.1 MCF メイン関数の作成 350
 - 10.1.2 MCF サービス名の登録 350
 - 10.1.3 システムサービス情報定義ファイルの作成 350
 - 10.1.4 定義オブジェクトファイルの作成 350
- 10.2 MCF メイン関数の作成 351
- 10.3 定義オブジェクトファイルの生成 354

11 障害対策 357

- 11.1 障害の種類と対応処理 358
 - 11.1.1 コネクション障害 358
 - 11.1.2 論理端末障害 358
 - 11.1.3 メッセージ障害 359
 - 11.1.4 マッピング障害 359
 - 11.1.5 プリンタ障害 359
 - 11.1.6 入力メッセージ編集 UOC の障害 360
 - 11.1.7 入力キュー, スケジュールサービス, ネームサービス, RPC の障害 361
 - 11.1.8 出力キュー障害 361
 - 11.1.9 ジャーナル障害 362
 - 11.1.10 バッファ障害 362
 - 11.1.11 UAP 障害 362
 - 11.1.12 MCF の障害 363
- 11.2 通信形態と障害の処理 364

- 11.2.1 非問い合わせ応答形態の障害 364
- 11.2.2 分岐送信形態の障害 365
- 11.2.3 問い合わせ応答形態の障害 365
- 11.2.4 継続問い合わせ応答形態の障害 367

付録 371

- 付録 A バージョンアップ時の変更点 372
 - 付録 A.1 07-50 での変更点 372
 - 付録 A.2 07-01 での変更点 373
 - 付録 A.3 07-00 での変更点 374
- 付録 B 旧製品からの移行に関する注意事項 375
- 付録 C インタフェースの変更一覧 (バージョン 6 以前から移行する場合) 377
 - 付録 C.1 メッセージ送受信インタフェース 377
 - 付録 C.2 ユーザOWNコーディング 382
 - 付録 C.3 MCF イベントインタフェース 384
 - 付録 C.4 MCF メイン関数のコーディング概要 386
- 付録 D MCF 性能検証用トレースの取得 388
 - 付録 D.1 MCF 固有情報の出力情報 388
 - 付録 D.2 MCF 性能検証用トレースの取得タイミング 389
 - 付録 D.3 MCF 性能検証用トレースの取得量 391
- 付録 E メッセージ送受信の流れ 392
- 付録 F XMAP3 Server サービス名との関係 394
- 付録 G 表示印刷サービスの起動でコネクション確立をするための環境設定 396
 - 付録 G.1 TP1/NET/XMAP3 の運用コマンドを発行するプログラムの作成 396
 - 付録 G.2 XMAP3 Server サービス名ファイルの設定 396
 - 付録 G.3 TP1/NET/XMAP3 プロトコル固有定義の設定 396
 - 付録 G.4 TP1/NET/XMAP3 運用コマンドを発行するプログラムと XMAP3 Server サービス名ファイルの例 397
- 付録 H ユーザアプリケーションプログラムの作成例 399
 - 付録 H.1 コーディング例 399
 - 付録 H.2 提供するサンプルコーディング 406
- 付録 I オペレーティングデータ表示メッセージ一覧 408
- 付録 J 理由コード一覧 411
- 付録 K マッピングサービスの障害コードの詳細 413
- 付録 L 用語解説 417

索引 420

1

概要

TP1/NET/XMAP3 は、OpenTP1 システムを構成するプログラムの一つです。TP1/NET/XMAP3 は、XMAP3 Server を使って、OpenTP1 システム内の UAP から論理的な画面入出力をする対話オンライン処理機能を提供します。

この章では、TP1/NET/XMAP3 の概要を説明します。

1.1 対話オンライン処理の概要

TP1/NET/XMAP3 は、XMAP3 Server を使って、OpenTP1 システム内の UAP から論理的な画面入出力ができる対話オンライン処理機能を提供するプログラムです。

XMAP3 Server とは、ワークステーション上で物理マッピング処理（表示・印刷）を支援するプログラムです。XMAP3 Server は、論理データ（画面・帳票メッセージ）と、画面・帳票定義情報※によって、物理マッピング処理をします。XMAP3 Server は、UAP とのインタフェースを提供する画面・帳票用のライブラリと、実際に帳票を印刷する印刷サービスと、サービスを制御するサーバ機能に分かれます。これに対して、パーソナルコンピュータ上で物理マッピング処理を支援するプログラムを、XMAP3 と呼びます。実際に画面表示をする表示サービスが追加されています。

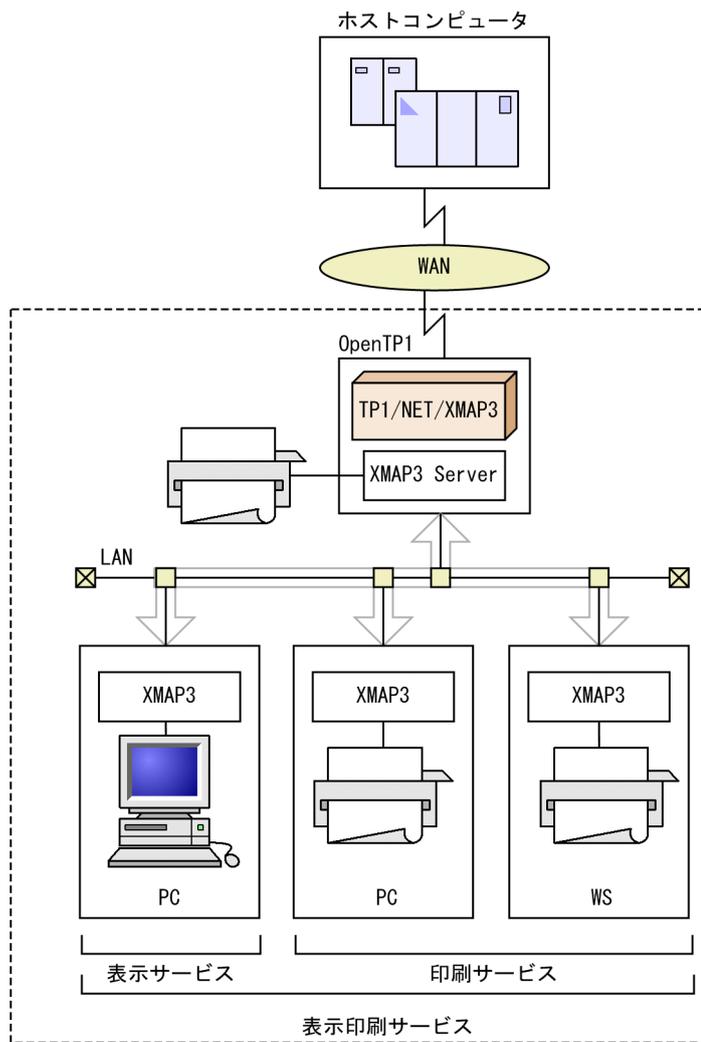
以降、このマニュアルでは、表示サービスと印刷サービスを総称して表示印刷サービスと呼びます。また、サーバ機能を XMAP3 サーバと呼びます。

注

画面・帳票定義情報は、画面や帳票の固定データ（固定文字、けい線など）や入力領域を定義しています。

TP1/NET/XMAP3 を使用したネットワーク構成の例を、次の図に示します。

図 1-1 TP1/NET/XMAP3 を使用したネットワーク構成の例



(凡例)

-----: 対話オンライン処理の範囲

1.2 対話オンライン処理の通信形態

ここでは、コネクションと論理端末の関係、およびメッセージ送受信の形態について説明します。

1.2.1 コネクションと論理端末の関係

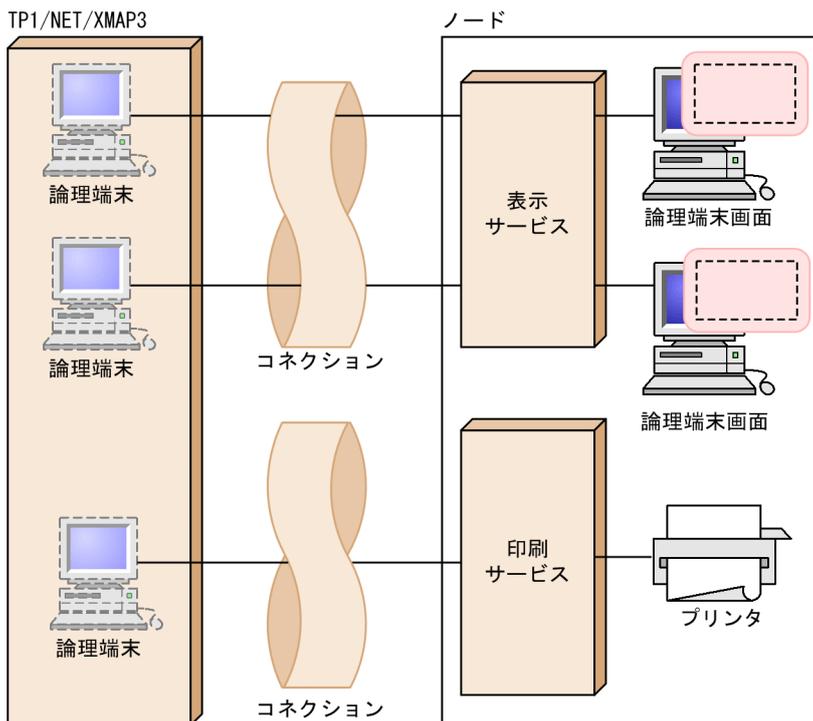
TP1/NET/XMAP3では、一つのコネクションが一つの表示印刷サービスに対応します。コネクションとは、TP1/NET/XMAP3と表示印刷サービス間に確立された論理的な通信路です。

コネクションと論理端末の対応はコネクションの属性によって異なります。論理端末とは、TP1/NET/XMAP3とOpenTP1システム内のUAPとのインタフェースです。表示サービスの場合、一つのコネクションには複数の論理端末を対応づけられます。この場合、論理端末と論理端末画面は1対1で対応します。印刷サービスの場合、一つのコネクションには一つの論理端末しか対応づけられません。

これらの論理端末構成は、ユーザがMCF通信構成定義で定義します。

コネクションと論理端末の関係を、次の図に示します。

図 1-2 コネクションと論理端末の関係



複数のコネクションが一つの表示印刷サービスに対応する構成を作成する場合、一つの表示印刷サービスに対応できるコネクション数は15までです。

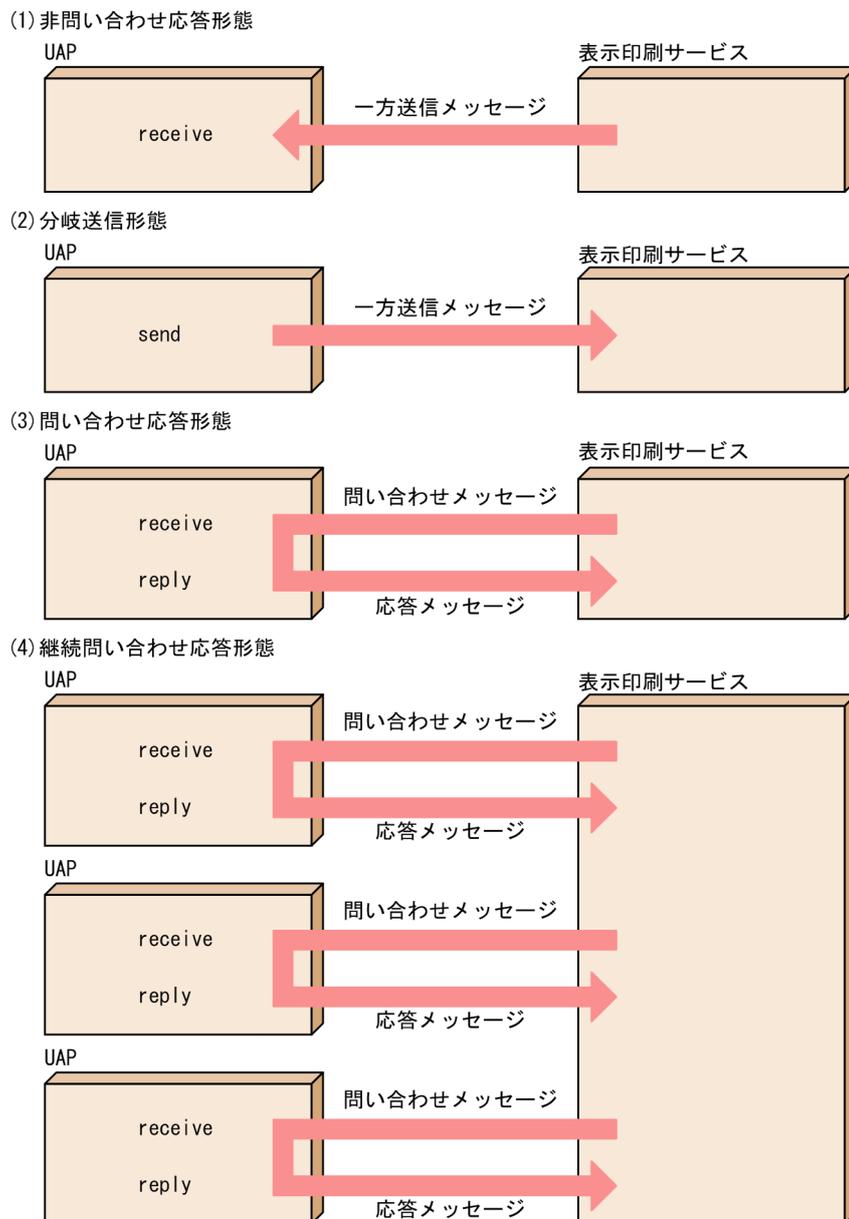
1.2.2 メッセージ送受信の形態

TP1/NET/XMAP3 では、UAP と表示印刷サービスとの間でメッセージを送受信します。TP1/NET/XMAP3 を使用したメッセージ送受信の形態には、非問い合わせ応答形態、分岐送信形態、問い合わせ応答形態、および継続問い合わせ応答形態の四つがあります。

四つの通信形態のうち、MHP で扱えるのは、非問い合わせ応答形態、問い合わせ応答形態、継続問い合わせ応答形態の三つです。分岐送信形態は、この三つの通信形態の中で使用し、単独では使用できません。また、SPP で扱う形態は、分岐送信形態だけです。

TP1/NET/XMAP3 を使用したメッセージ送受信の例を、次の図に示します。

図 1-3 TP1/NET/XMAP3 を使用したメッセージ送受信の例



(1) 非問い合わせ応答形態

非問い合わせ応答形態とは、UAP が表示印刷サービスから一方送信メッセージを受信する通信形態です。

(2) 分岐送信形態

分岐送信形態とは、UAP が表示印刷サービスに対して一方送信メッセージを送信する通信形態です。

(3) 問い合わせ応答形態

問い合わせ応答形態とは、UAP が表示印刷サービスから問い合わせメッセージを受信し、UAP で処理したあと、表示印刷サービスに対して応答メッセージを送信する通信形態です。一つの問い合わせメッセージに対して、複数の応答メッセージを送信できます。

(4) 継続問い合わせ応答形態

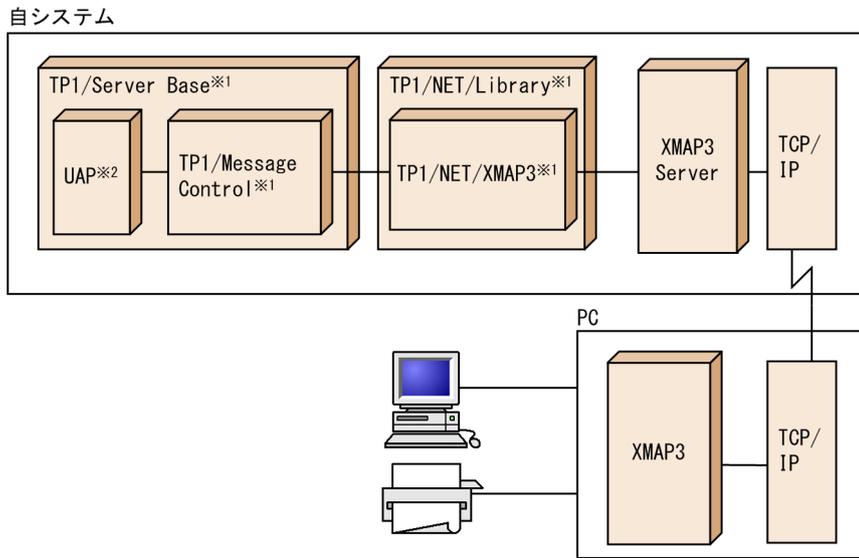
継続問い合わせ応答形態とは、複数の UAP が、表示印刷サービスからの問い合わせメッセージの受信、および応答メッセージの送信を継続して行う通信形態です。

1.3 ソフトウェア構成の例

TP1/NET/XMAP3 は、OpenTP1 システムに組み込まれて動作するプログラムです。OpenTP1 のメッセージ送受信機能（TP1/Message Control, TP1/NET/Library）と連携して、メッセージ制御機能（MCF）を実現します。

TP1/NET/XMAP3 を組み込んだソフトウェア構成の例を、次の図に示します。

図 1-4 TP1/NET/XMAP3 を組み込んだソフトウェア構成の例



注※1

表示印刷サービスと画面・帳票メッセージを送受信する機能を提供します。

注※2

TP1/NET/XMAP3 で扱う UAP は、MHP と SPP です。UAP については、マニュアル「OpenTP1 プログラム作成の手引」を参照してください。

2

機能

TP1/NET/XMAP3 は、対話オンライン処理機能を提供し、メッセージの送受信をします。

この章では、TP1/NET/XMAP3 で使用できるコネクションに関する機能、論理端末に関する機能、およびメッセージ送受信に関する機能について説明します。

2.1 オンラインの開始と終了

TP1/NET/XMAP3 は、オンライン開始時、システム定義の指定に従って表示印刷サービスにサービス開始要求や論理端末画面の生成などをします。

2.1.1 オンラインの開始形態と終了形態

TP1/NET/XMAP3 で対応しているオンラインの開始形態を、次の表に示します。

表 2-1 オンラインの開始形態

開始形態	前回の終了形態	TP1/NET/XMAP3 の処理
正常開始	<ul style="list-style-type: none">システム構築時正常終了強制正常終了	MCF 定義ファイルに従ってオンライン環境を構築します。 <ul style="list-style-type: none">表示印刷サービスにサービス開始要求、論理端末画面の生成
再開始	<ul style="list-style-type: none">計画停止 A, B強制停止	MCF 定義ファイルに従ってオンライン環境を構築します。 ただし、前回のオンラインの入力キュー、出力キューに残っているメッセージは引き継がれます。 また、継続問い合わせ応答形態の情報は引き継がれません。 <ul style="list-style-type: none">表示印刷サービスにサービス開始要求、論理端末画面の生成

また、TP1/NET/XMAP3 で対応しているオンラインの終了形態を、次の表に示します。

表 2-2 オンラインの終了形態

終了形態	TP1/NET/XMAP3 の処理
正常終了 強制正常 終了	<ol style="list-style-type: none">出力キュー内の未処理送信メッセージをすべて送信します。 ただし、一定時間内に出力キュー内の全メッセージを送信できなかった場合は、UAP に未処理送信メッセージ廃棄イベント (ERREVT A) を通知します。論理端末画面を廃棄し、表示印刷サービスにサービス終了要求をします。オンラインを終了します。
計画停止 A, B	<ol style="list-style-type: none">送信中のメッセージを送信完了まで処理します (出力キュー内の未処理送信メッセージは次回の再開時に処理します)。論理端末画面を廃棄し、表示印刷サービスにサービス終了要求をします。オンラインを終了します。
強制停止	<ol style="list-style-type: none">直ちにオンラインを終了します。 このとき、表示印刷サービス側の資源は解放され、論理端末画面は廃棄されます。そのため、次回のオンライン開始時には表示印刷サービスのサービス開始要求から処理します。

2.1.2 XMAP3 サーバの起動

XMAP3 サーバは、OpenTP1 システムを開始する前に起動してください。

XMAP3 サーバの起動については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

2.2 コネクションに関する機能

TP1/NET/XMAP3 では、コネクションの確立と解放によって、論理端末と表示印刷サービスとの接続、および切り離しをします。

ここでは、コネクションに関する機能について説明します。

2.2.1 コネクションの確立

TP1/NET/XMAP3 では、コネクションを確立して、論理端末と表示印刷サービスとを接続します。

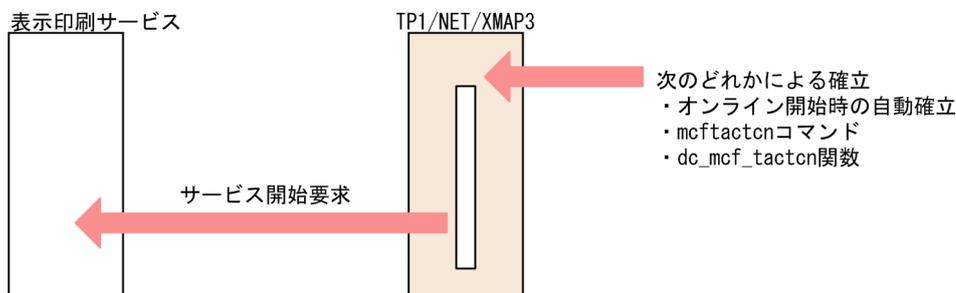
コネクションは、次のどれかの方法で確立します。

- オンライン開始時の自動確立
ユーザがあらかじめ、自動確立させるコネクションを MCF 通信構成定義で指定しておきます。TP1/NET/XMAP3 は、オンラインを開始するとき、指定されたコネクションを確立します。自動確立の指定がないコネクションは、オンライン開始後、解放状態となります。
- コネクションの確立コマンド (mcftactcn) による確立
ユーザが運用コマンドで、コネクションの確立を指示します。TP1/NET/XMAP3 は、指示されたコネクションを確立します。
- コネクションの確立 API (dc_mcf_tactcn 関数) による確立
UAP から dc_mcf_tactcn 関数を発行することで、コネクションの確立を指示します。TP1/NET/XMAP3 は、指示されたコネクションを確立します。

コネクションを確立するとき、TP1/NET/XMAP3 は、表示印刷サービスに要求を出してそのコネクションに対応するサービスを開始させます。該当するコネクションに自動起動指定の論理端末がある場合は、その論理端末を起動します。

コネクションの確立の流れを、次の図に示します。

図 2-1 コネクションの確立の流れ



表示印刷サービスの起動でコネクション確立をすることができます。詳細については「付録 G 表示印刷サービスの起動でコネクション確立するための環境設定」を参照してください。

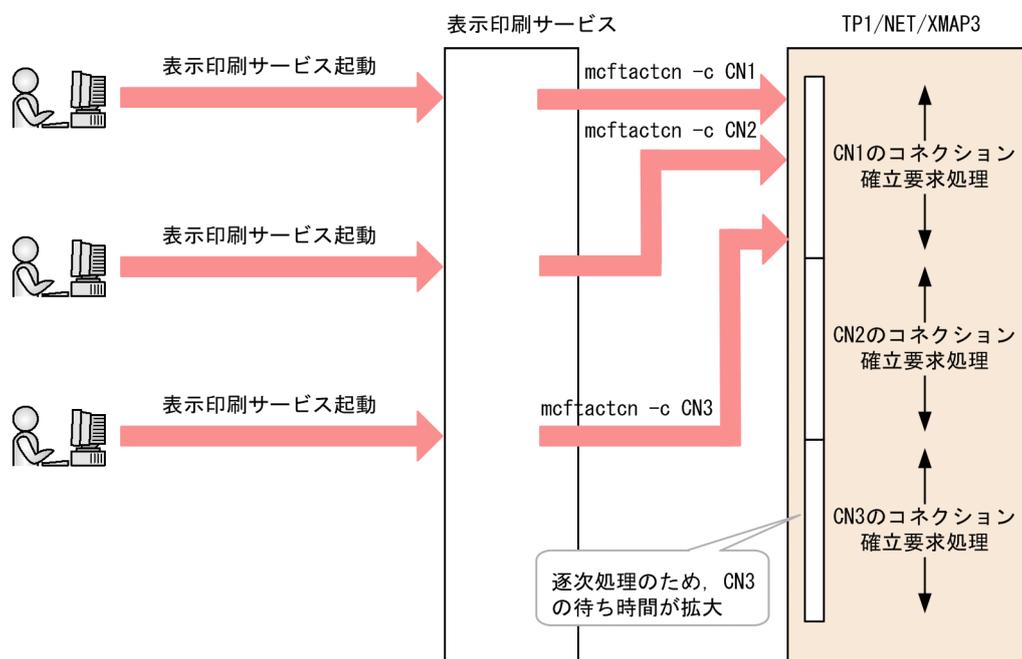
(1) コネクションの確立の動作モード

コネクションの確立の動作モードを XMAP3 共通定義 (mcftxp -c) の actcnaction オペランドで指定できます。動作モードにはコネクション確立要求を逐次処理する形態とコネクション確立要求を並行処理する形態があります。

(a) コネクション確立要求を逐次処理する場合

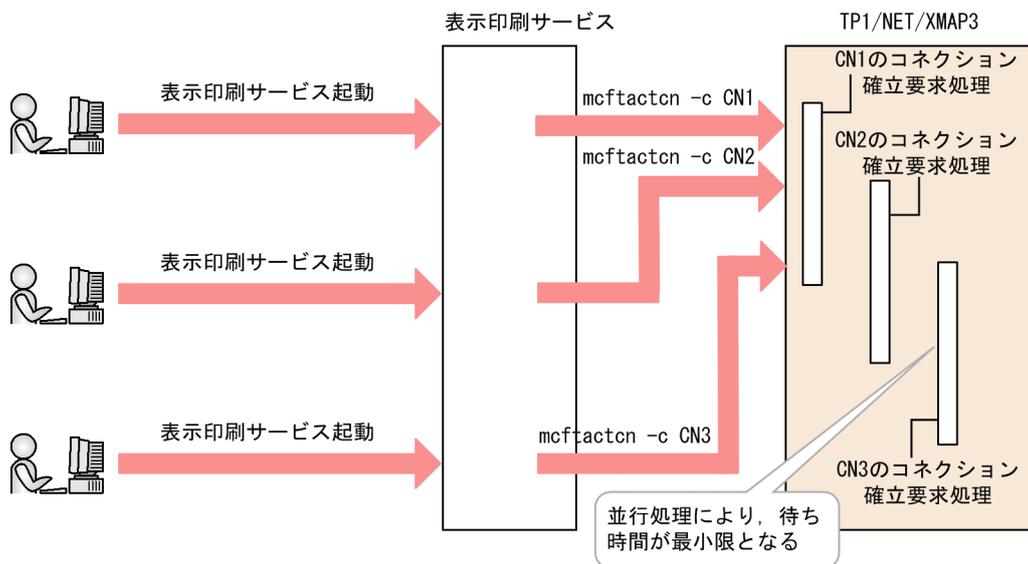
逐次処理する指定をした場合、TP1/NET/XMAP3 はコネクション確立要求が完了するまで次の運用コマンドまたは API によるコネクション確立要求を待ち合わせます。

コネクション確立要求を行う運用コマンドまたは API がリターンしたとき、TP1/NET/XMAP3 がコマンドまたは API を正常に受け付けた場合、コネクションは確立状態となっていますが、運用コマンドを多数入力する場合には、順番待ちが発生して、入力数が増えるほど、待ち時間が拡大します。



(b) コネクション確立要求を並行処理する場合

並行処理する指定をした場合、TP1/NET/XMAP3 はコネクション確立要求の処理中に、次の運用コマンドまたは API によるコネクション確立要求を受け付けることができます。画面表示を行うシステムにおいて、表示印刷サービスの起動でコネクション確立をするための環境設定を行っている場合、この指定をすることを推奨します。



ただし、接続確立要求を並行処理する場合、次の点に注意してください。

1. 接続確立要求を行う運用コマンドまたはAPIがリターンしたとき、TP1/NET/XMAP3で正常に受け付けられていたとしても、接続の状態が確立状態とは限りません。接続確立要求を行う運用コマンドまたはAPIがリターンしたあとに、接続が確立していることを前提とした処理がある場合は、接続の状態確認を行う運用コマンドまたはAPIで、接続の状態を確認してください。
2. すべての接続の確立要求処理が完了する時間は、最大処理多重度定義の最大処理多重度 (mcfttred -m) に依存します。最大処理多重度は、負荷試験を実施したうえで最適な値を指定するようにしてください。多重処理状況は運用コマンド (mcftlstrd) で確認できます。運用コマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

(c) 逐次処理と並行処理の動作差異

逐次処理と並行処理の動作差異を次の表に示します。

表 2-3 逐次処理と並行処理の動作差異

定義内容	接続確立要求を行う運用コマンドまたはAPIがリターンしたとき、TP1/NET/XMAP3がコマンドまたはAPIを正常に受け付けた場合の接続状態	全接続の確立要求処理が完了するまでに要する時間
接続確立要求を逐次処理する場合	確立状態	各接続の確立処理に要した時間の総和 (接続数が多いほど、時間を要します)
接続確立要求を並行処理する場合	確立状態とはかぎらない (通信障害が発生したとき、解放状態または解放処理中状態の場合もあります)	各接続の接続確立処理に要した時間のうちの最大値 (実際の所要時間は最大処理多重度に依存します)

2.2.2 コネクションの解放

TP1/NET/XMAP3 では、コネクションを解放して、論理端末と表示印刷サービスを切り離します。

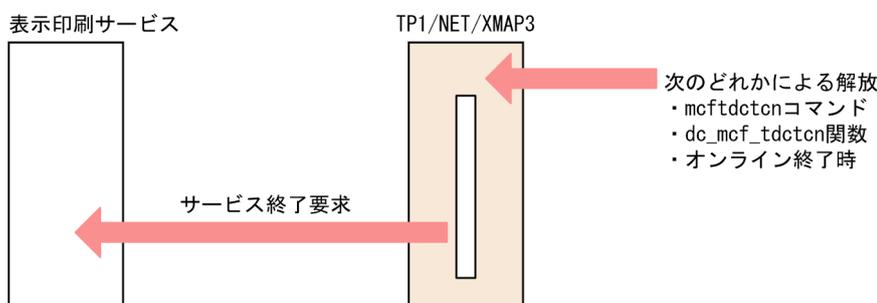
コネクションは、次の場合に解放されます。

- 表示印刷サービス終了時
TP1/NET/XMAP3 は、対応するコネクションを解放します。
- コネクションの強制解放コマンド (mcftdctcn) 入力時
ユーザが運用コマンドで、コネクションの強制解放を指示します。TP1/NET/XMAP3 は、指示されたコネクションを解放します。
- コネクションの解放 API (dc_mcf_tdctcn 関数) 発行時
UAP から dc_mcf_tdctcn 関数を発行することで、コネクションの解放を指示します。TP1/NET/XMAP3 は、指示されたコネクションを解放します。
- オンライン終了時 (正常終了, 強制正常終了, 計画停止 A, および計画停止 B)
オンラインを終了するとき、TP1/NET/XMAP3 は、すべてのコネクションを解放します。

コネクションを解放するとき、TP1/NET/XMAP3 は、そのコネクションに対応するすべての論理端末を閉塞します。それから、表示印刷サービスに要求を出して、そのコネクションに対応するサービスを終了させます。

コネクションの解放の流れを、次の図に示します。

図 2-2 コネクションの解放の流れ



2.2.3 コネクションの状態表示

コネクションが確立状態か解放状態かといったコネクションの状態や、コネクション ID、プロトコル種別などを表示できます。コネクションの状態を表示するには、運用コマンド (mcftlscn) を入力するか、または API (dc_mcf_tlscn 関数) を発行します。

運用コマンド (mcftlscn) を使用した場合、コネクションの状態は、標準出力に表示されます。一方、API (dc_mcf_tlscn 関数) を使用した場合、コネクションの状態は UAP 中で指定した領域に格納されます。

運用コマンド (mcftlscn) のオプションの指定によっては、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。

2.3 論理端末に関する機能

論理端末とは、TP1/NET/XMAP3 と OpenTP1 システム内の UAP とのインタフェースです。TP1/NET/XMAP3 の論理端末には、入力キューおよび出力キューを管理するキュー状態と、表示印刷サービスを管理する端末状態があります。

ここでは、論理端末に関する機能について説明します。

2.3.1 論理端末とアプリケーションの型の関係

TP1/NET/XMAP3 で扱う論理端末の端末タイプは any (任意型) です。MCF 通信構成定義の論理端末定義 (mcfalcle) で、この端末タイプを指定してください。

次に、MCF アプリケーション定義のアプリケーション属性定義 (mcfaalcap) で指定するアプリケーションの型を示します。

- 応答型 (ans)
- 非応答型 (noans)
- 継続問い合わせ応答型 (cont)

アプリケーションは、ユーザが MHP で作成して提供する業務サービスです。その通信形態に合わせて、上記のどれかの型をアプリケーション属性定義 (mcfaalcap) で指定してください。論理端末の端末タイプ、アプリケーションの型、通信形態、メッセージの種類、および UAP インタフェースの関係を次の表に示します。

表 2-4 論理端末の端末タイプ、アプリケーションの型、通信形態、メッセージの種類、および UAP インタフェースの関係

論理端末の端末タイプ	アプリケーションの型	通信形態	メッセージの種類	UAP インタフェース
any	ans	問い合わせ応答	問い合わせメッセージ	dc_mcf_receive
			応答メッセージ	dc_mcf_reply
		分岐送信	一方送信メッセージ	dc_mcf_send
	noans	非問い合わせ応答	一方送信メッセージ	dc_mcf_receive
			分岐送信	一方送信メッセージ
	cont	継続問い合わせ応答	問い合わせメッセージ	dc_mcf_receive
			応答メッセージ	dc_mcf_reply
		分岐送信	一方送信メッセージ	dc_mcf_send

それぞれのアプリケーションの型での、dc_mcf_reply, dc_mcf_send 呼び出しの可否を次の表に示します。

表 2-5 reply, send 呼び出しの可否

アプリケーションの型	UAP インタフェース	送信先の論理端末	
		入力元	入力元以外
ans	dc_mcf_reply	◎	×
	dc_mcf_send	○	○
noans	dc_mcf_reply	○*	×
	dc_mcf_send	○	○
cont	dc_mcf_reply	◎	×
	dc_mcf_send	○	○

(凡例)

◎：次の場合は、必ず呼び出してください。

- dc_mcf_execap 関数を呼び出してほかのアプリケーションに応答の権利を譲渡しない場合
- 応答メッセージを送信したかどうかをチェックする（アプリケーション属性定義（mcfaalcap - n）の replychk オペランドを省略、または yes を指定）応答型のアプリケーションの場合

dc_mcf_execap 関数を呼び出して、応答の権利をほかのアプリケーションに譲渡した場合は呼び出せません。

○：

呼び出せます。

×：

呼び出せません。

注※

非応答型のアプリケーションからの問い合わせ応答をする（UAP 共通定義（mcfmuap）の -c オプションの noansreply オペランドに yes を指定）場合に呼び出せます。

一方送信メッセージとして送信します。

2.3.2 論理端末のキュー状態の閉塞と閉塞解除

ここでは、入力キューおよび出力キューを管理するキュー状態の閉塞と閉塞解除について説明します。

TP1/NET/XMAP3 は、閉塞されているキューに対応する UAP への送信メッセージを破棄します。また、閉塞されているキューに対応する UAP から表示印刷サービスへの送信メッセージは、表示印刷サービスへ送信されないで、出力キューに蓄えられます。

キュー状態は、コネクションが確立しているかどうかに関係なく、閉塞したり、閉塞解除したりできます。

オンライン開始時に TP1/NET/XMAP3 が管理するすべてのキュー状態は、閉塞解除状態になります。

オンライン中に運用コマンド (mcftdctle) で閉塞した場合は、運用コマンド (mcftactle) で閉塞解除できます。オンライン中に状態が閉塞解除のときは、運用コマンド (mcftdctle) で閉塞できます。

なお、UAP から API (dc_mcf_tdctle 関数, dc_mcf_tactle 関数) を発行すれば、論理端末のキュー状態と端末状態をまとめて閉塞したり閉塞解除したりできます。API の詳細については、「3. C 言語のライブラリ関数」, または「4. COBOL-UAP 作成用プログラムインタフェース」を参照してください。

2.3.3 論理端末の端末状態の閉塞と閉塞解除

ここでは、表示印刷サービスを管理する端末状態の閉塞と閉塞解除について説明します。

画面属性の論理端末の端末状態が閉塞されている場合は、論理端末画面は表示されません。また、プリンタ属性の論理端末の端末状態が閉塞されている場合は、送信メッセージはプリンタに出力されません。

TP1/NET/XMAP3 は、閉塞されている表示印刷サービスへの送信メッセージを OpenTP1 の出力キューに蓄えます。

(1) 端末状態の閉塞解除

端末状態は、次のどれかの方法で閉塞解除します。

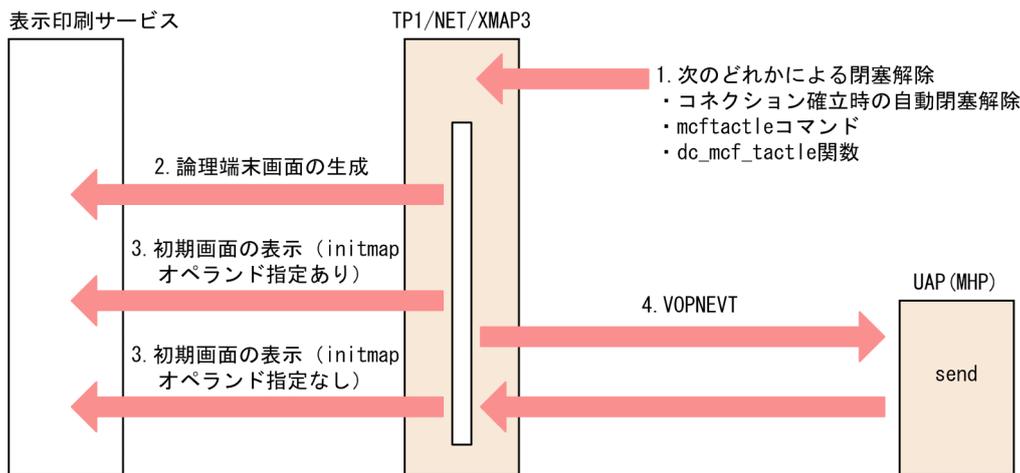
- コネクション確立時の自動閉塞解除
コネクションを確立するとき、TP1/NET/XMAP3 が端末状態を閉塞解除します。
ユーザは、自動閉塞解除する論理端末を MCF 通信構成定義で指定しておきます。
自動閉塞解除の指定がない論理端末は、コネクション確立時に端末状態が閉塞状態になります。
- 論理端末の閉塞解除コマンド (mcftactle) による閉塞解除
ユーザが運用コマンドで、論理端末の閉塞解除を指示します。TP1/NET/XMAP3 は、閉塞解除を指示された論理端末の端末状態を閉塞解除します。この場合、論理端末に対応するコネクションが確立されている必要があります。
- 論理端末の閉塞解除 API (dc_mcf_tactle 関数) による閉塞解除
UAP から dc_mcf_tactle 関数を発行して論理端末の閉塞解除を指示します。TP1/NET/XMAP3 は、閉塞解除を指示された論理端末の端末状態を閉塞解除します。この場合、論理端末に対応するコネクションが確立されている必要があります。なお、API で閉塞解除する場合は、論理端末のキュー状態もあわせて閉塞解除されます。

論理端末の閉塞解除処理は、該当する論理端末の属性が画面かプリンタかによって異なります。論理端末の属性は、コネクション定義 (mcftalccn -S) の type オペランドで指定します。

(a) 画面属性の端末状態の閉塞解除処理

画面属性の端末状態の閉塞解除処理を、次の図に示します。

図 2-3 画面属性の端末状態の閉塞解除処理



次に、図に示した流れの詳細について説明します。番号は、図中の番号と対応しています。

1. 端末状態を、次のどれかによって閉塞解除します。

- コネクション確立時の自動閉塞解除
- 運用コマンド (mcftactle) の入力
- API (dc_mcf_tactle 関数) の発行

2. 論理端末画面を生成します。

3. 論理端末定義 (mcftalcle -p) で initmap オペランドの指定がある場合、初期画面を表示します。

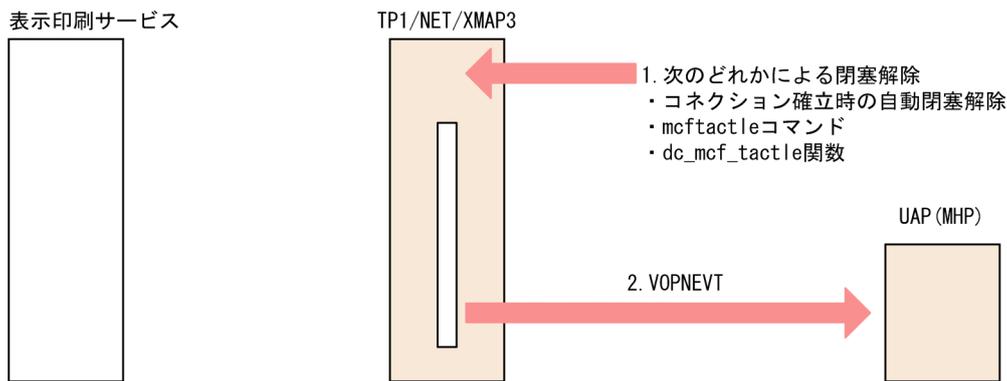
論理端末定義 (mcftalcle -p) で initmap オペランドの指定がない場合、状態通知イベント (VOPNEVT) で起動される UAP から send 関数を呼び出して表示できます (reply 関数は呼び出せません)。初期画面表示の send 関数を呼び出さないと論理端末画面は表示されないため注意してください。

4. UAP に状態通知イベント (VOPNEVT) を発行します。

(b) プリンタ属性の端末状態の閉塞解除処理

プリンタ属性の端末状態の閉塞解除処理を、次の図に示します。

図 2-4 プリンタ属性の端末状態の閉塞解除処理



次に、図に示した流れの詳細について説明します。番号は、図中の番号と対応しています。

1. 端末状態を、次のどれかによって閉塞解除します。

- ・ コネクション確立時の自動閉塞解除
- ・ 運用コマンド (mcftactle) の入力
- ・ API (dc_mcf_tactle 関数) の発行

2. UAP に状態通知イベント (VOPNEVT) を発行します。

(2) 端末状態の閉塞

端末状態は、次の場合に閉塞されます。

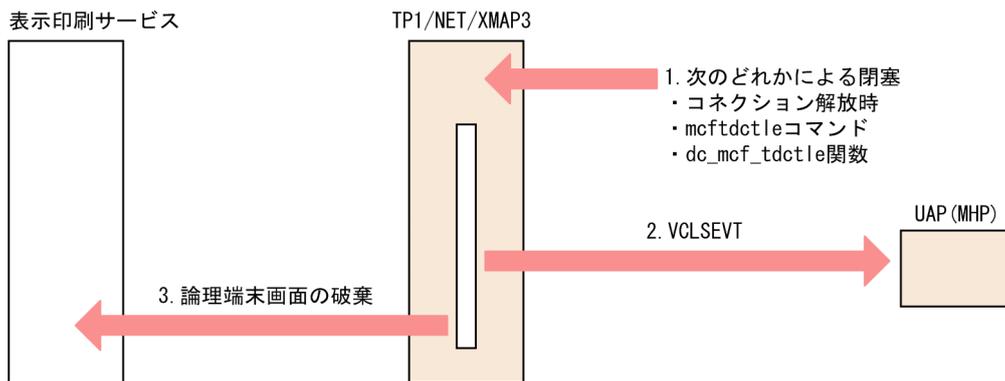
- ・ コネクション解放時
コネクションを解放するとき、TP1/NET/XMAP3 は、それに対応するすべての端末状態を閉塞します。
- ・ 論理端末の閉塞コマンド (mcftdctle) 入力時
ユーザが運用コマンドで、端末状態の閉塞を指示します。TP1/NET/XMAP3 は、指示された端末状態を閉塞します。
- ・ 論理端末の閉塞 API (dc_mcf_tdctle 関数) 発行時
UAP から dc_mcf_tdctle 関数を発行して端末状態の閉塞を指示します。TP1/NET/XMAP3 は、指示された端末状態を閉塞します。なお、API で閉塞する場合は、論理端末のキュー状態もあわせて閉塞されます。

端末状態の閉塞処理は、該当する論理端末の属性が画面かプリンタかによって異なります。論理端末の属性は、コネクション定義 (mcftalccn -S) の type オペランドで指定します。

(a) 画面属性の端末状態の閉塞処理

画面属性の端末状態の閉塞処理を、次の図に示します。

図 2-5 画面属性の端末状態の閉塞処理



次に、図に示した流れの詳細について説明します。番号は、図中の番号と対応しています。

1. 端末状態を、次のどれかによって閉塞します。

- ・ コネクション解放時
- ・ 運用コマンド (mcftdctle) の入力
- ・ API (dc_mcf_tdctle 関数) の発行

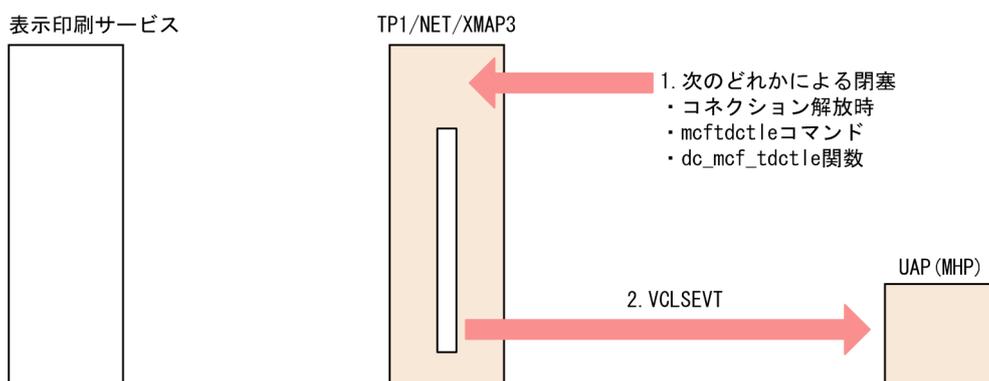
2. UAP に状態通知イベント (VCLSEVT) を発行します。

3. 論理端末画面を破棄します。

(b) プリンタ属性の端末状態の閉塞処理

プリンタ属性の端末状態の閉塞処理を、次の図に示します。

図 2-6 プリンタ属性の端末状態の閉塞処理



次に、図に示した流れの詳細について説明します。番号は、図中の番号と対応しています。

1. 端末状態を、次のどれかによって閉塞します。

- ・ コネクション解放時
- ・ 運用コマンド (mcftdctle) の入力
- ・ API (dc_mcf_tdctle 関数) の発行

2. UAP に状態通知イベント (VCLSEVT) を発行します。

2.3.4 論理端末の状態表示

論理端末の状態は、運用コマンド (mcftlsle) の入力、または API (dc_mcf_tlsle 関数) の発行によって表示できます。表示できる内容は、MCF 識別子、論理端末名称、論理端末状態 (閉塞状態、または閉塞解除状態) などです。

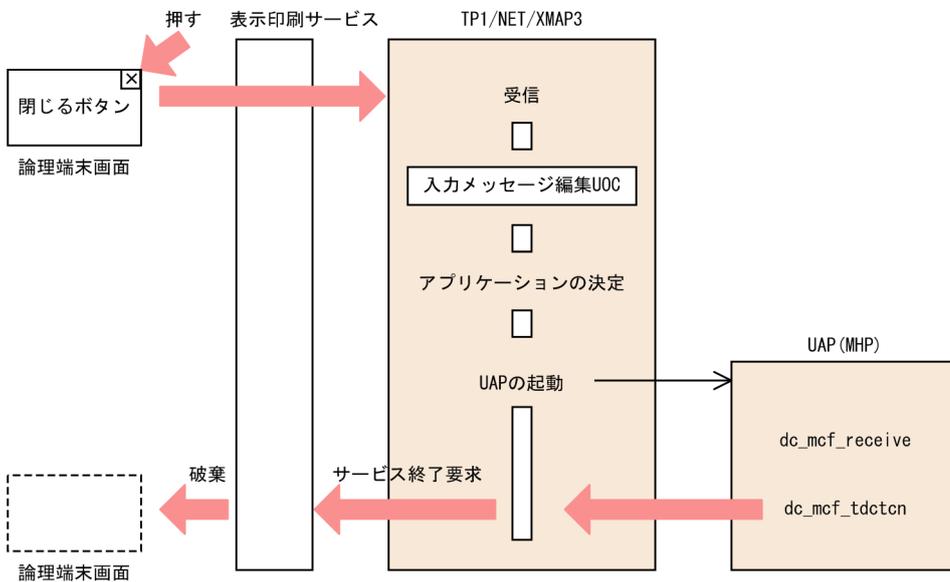
運用コマンド (mcftlsle) を使用した場合、論理端末の状態は、標準出力に表示されます。一方、API (dc_mcf_tlsle 関数) を使用した場合、論理端末の状態は UAP 中で指定した領域に格納されます。

2.3.5 論理端末画面の閉じるボタンの有効化

論理端末画面の閉じるボタン (ウィンドウ右上隅の閉じるボタン) を押した場合、通常は XMAP3 クライアント端末側で無効となります (入力エラーとしてブザーを鳴らします)。XMAP3 共通定義 (mcftxp -g) の closebutton オペランドに on を指定することで、この閉じるボタンを有効にできます。

この機能を使用して閉じるボタンを押すと、ほかのキー入力と同様に、アプリケーション決定後に UAP が起動されます。UAP から接続を解放したり、論理端末を閉塞したりして論理端末画面を閉じてください。閉じるボタンによる論理端末画面の破棄を、次の図に示します。

図 2-7 閉じるボタンによる論理端末画面の破棄



この機能を使用する場合の留意事項を次に示します。

- XMAP3 Server 表示・印刷環境ファイルまたは XMAP3 の表示・印刷セットアップダイアログで、閉じるボタンを表示しておく必要があります。

- 閉じるボタンを押すと、「割込」(オンライン用キー)または「Break Ctrl+Pause」(AIX仕様キー)として通知されます。閉じるボタンを押したときの処理と割込キーを押したときの処理は使い分けられません。
送信要求の発生イベントについては、「2.4.10(1) 送信要求発生時の処理」を参照してください。
- 次のアプリケーションの実行中は、キーボード操作で論理端末画面のキーボードロックを解除したあとに閉じるボタンを押しても、KFCA17800-EまたはKFCA13664-Eメッセージを出力して、アプリケーションは起動されません。論理端末画面のキーボードロックが長時間解除されないときに論理端末画面を閉じる場合は、XMAP3クライアントを終了させてください。
 - noans型アプリケーション(論理端末定義(mcftalcle -K)でkeyboardオペランドにaplockを指定し、かつunlockオペランドにautoを指定または省略した場合)
 - ans型アプリケーション
 - cont型アプリケーション
- 継続問い合わせ応答形態でこの機能を使用する場合は、次の点に留意してください。
 - 継続問い合わせ応答中に閉じるボタンを押した場合、XMAP3のドロワーの隠しフィールドなどの設定値にかかわらず、予約されている次起動アプリケーション名でUAPが起動されます。すべての継続問い合わせ応答型のアプリケーションに閉じるボタンを押したときの処理を追加するか、入力メッセージ編集UOCで閉じるボタンに対応するアプリケーションを指定してください。
 - 継続問い合わせ応答中の場合、継続問い合わせ応答を終了するまで論理端末を閉塞できません。継続問い合わせ応答中に閉じるボタンを押して起動したUAPでは、contend関数で継続問い合わせ応答を終了してから論理端末を閉塞するか、コネクションを解放するようにしてください。

2.4 メッセージ送受信に関する機能

ここでは、メッセージ送受信に関する機能について説明します。

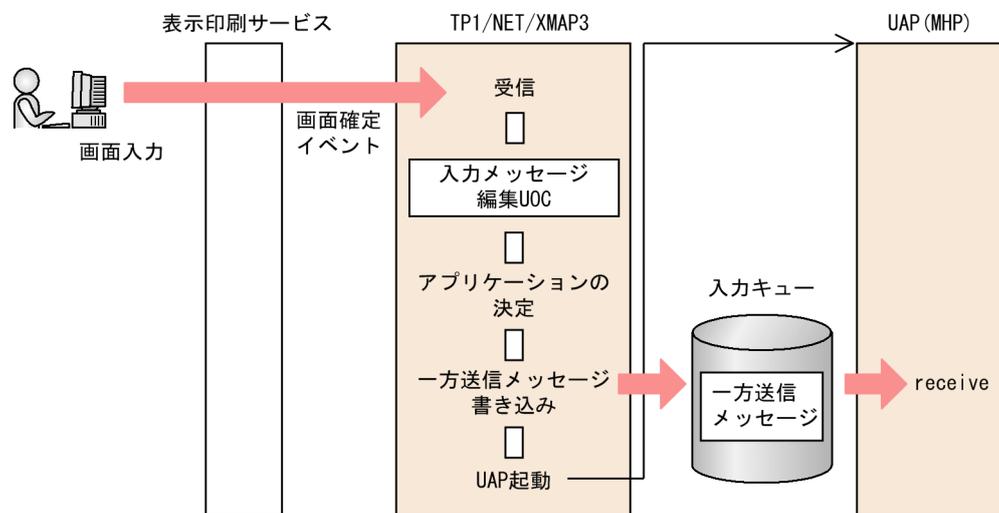
2.4.1 非問い合わせ応答形態のメッセージ

TP1/NET/XMAP3では、非応答型 (noans) アプリケーションに発生したメッセージを一方送信メッセージとして入力キューに登録して、UAP を起動します。

起動された UAP では、receive 関数を呼び出して一方送信メッセージを受け取ります。

非問い合わせ応答形態を、次の図に示します。

図 2-8 非問い合わせ応答形態



非応答型のアプリケーションからの問い合わせ応答をする (UAP 共通定義 (mcfmuap) の-c オプションの noansreply オペランドに yes を指定) 場合、非応答型のアプリケーションから reply 関数を呼び出して一方送信メッセージを送信できます。この一方送信メッセージは、出力通番を付けない一般の一方送信メッセージとして扱います。

2.4.2 分岐送信形態のメッセージ

send 関数によって指定されたメッセージを一方送信メッセージとして出力キューに登録します。

TP1/NET/XMAP3 は、この一方送信メッセージを取り出して表示印刷サービスに送信します。

入力元論理端末へメッセージを送信する場合、応答メッセージを優先します。一方送信メッセージは、応答メッセージをすべて送信したあとに送信します。

画面に対するメッセージの場合は、一方送信メッセージが送信済みであることを出力キューに連絡します。これを送信完了処理と呼びます。プリンタに対するメッセージの場合は、TP1/NET/XMAP3 は表示印刷

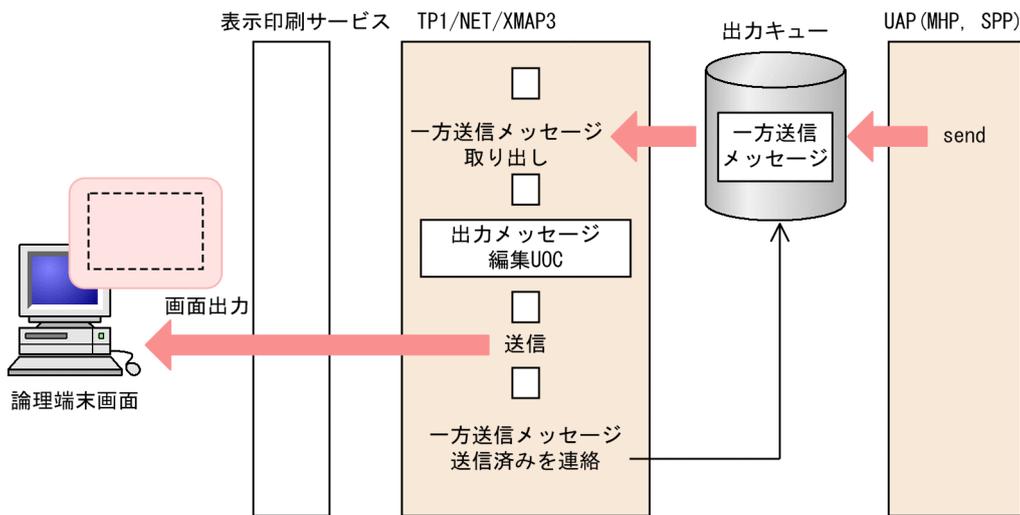
サービスからの送信後印刷完了イベントを待ちます。そのイベントを受け取ったあと、プリンタの排他を解除してから一方送信メッセージが送信済みであることを出力キューに連絡します。UAP は、表示印刷サービスの属性を意識しないで送信要求できます。

次の場合、その論理端末に対する一方送信メッセージの送信が抑止されます。

- 論理端末から cont 型, ans 型のアプリケーションに対するメッセージを受信した場合
- 論理端末定義 (mcftalcle -K) の keyboard オペランドで aplock を指定し、かつ unlock オペランドで auto を指定した論理端末から noans 型のアプリケーションに対するメッセージを受信した場合

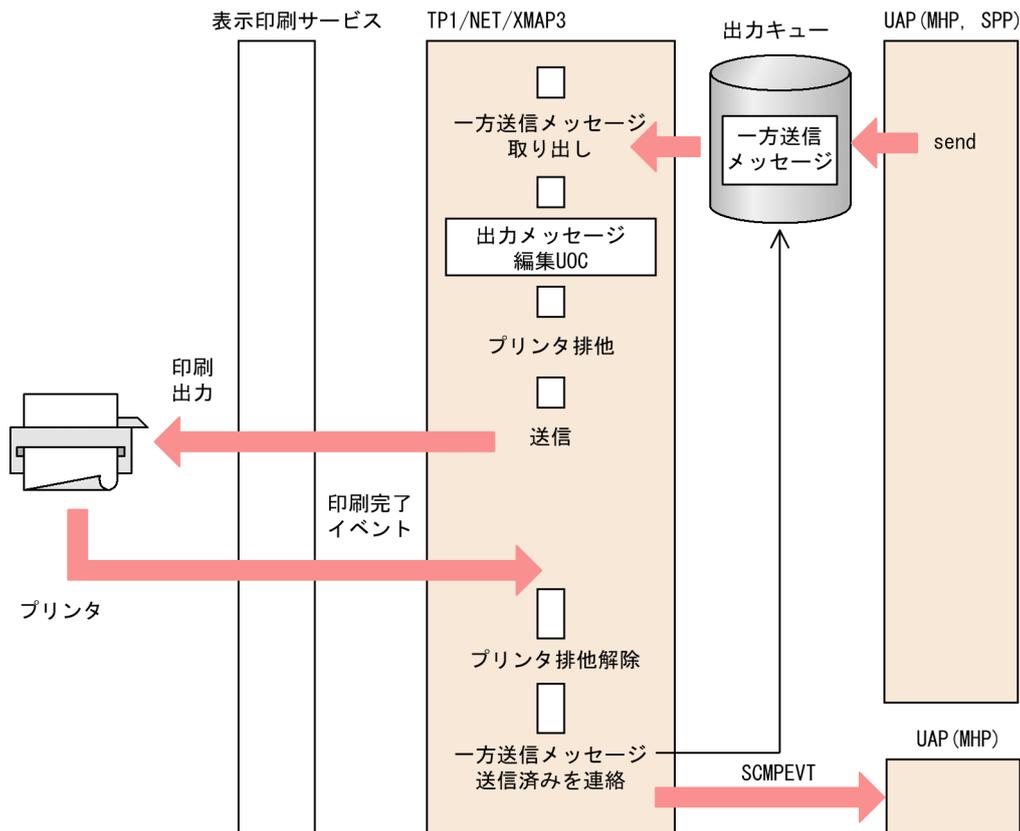
画面に対する分岐送信形態を次の図に示します。

図 2-9 画面に対する分岐送信形態



プリンタに対する分岐送信形態を次の図に示します。

図 2-10 プリンタに対する分岐送信形態



プリンタに対する分岐送信形態の場合、send 関数で送信完了通知イベント (SCMP EVT) および送信障害通知イベント (SERREVT) を通知することを指定すると、送信完了および送信障害を通知します。送信が正常に終了したとき、SCMP EVT を通知します。また、一方送信メッセージ取り出しから印刷完了イベント受信までの間で障害となった場合、SERREVT を通知します。

送信完了および送信障害を通知する場合の注意事項を次に示します。

- 本来は、すべての送信メッセージに対するイベント通知の指定が必要ですが、イベントスケジュール処理集中の原因となりますので、次のように運用してください。
 - グループ送信で、論理端末定義 (mcftalcle) の-y オプションの指定値が 1 グループの最大送信メッセージ数以上で、かつ-x オプションで excnt=0 を指定した場合、グループ最終メッセージでイベント通知を指定すれば、SCMP EVT で全メッセージの送信完了を知ることができます。ただし、SERREVT は通知されない場合があります。
 - グループ先頭メッセージにもイベント通知を指定すれば、SERREVT で送信障害の発生を知ることができます。この場合、SCMP EVT は先頭と最終のメッセージに対応して二つ通知されます。また、SERREVT も二つ通知される場合があります。
- 再送要求のとき、イベント通知の有無は変更できません。
- オンライン終了中は、イベントは通知されません。
- イベント通知順は、必ずしも send 関数の発行順になるとは限りません。

- メッセージセグメントは設定されません。イベントを処理する MHP では、イベント情報内の出力通番を参照して処理を行ってください。
出力通番は、送信メッセージ通番編集 UOC であらかじめ取得しておく必要があります。なお、出力通番は send 関数の発行順に採番されるため、グループ送信時でもグループ内で連番になるとは限りません。
- 送信障害通知されたメッセージも、障害回復によって正常に印刷されることがあります。また、プリンタによっては、正常に印刷されても送信障害通知となることがあります。
- 代行送信した場合、メッセージ受信関数の入力元論理端末名称には、代行先論理端末名称が設定されます。また、イベント情報の出力先論理端末名称およびコネクション名は、代行先論理端末名称になります。
- 画面属性の論理端末に対するイベント通知の指定は無効になります。

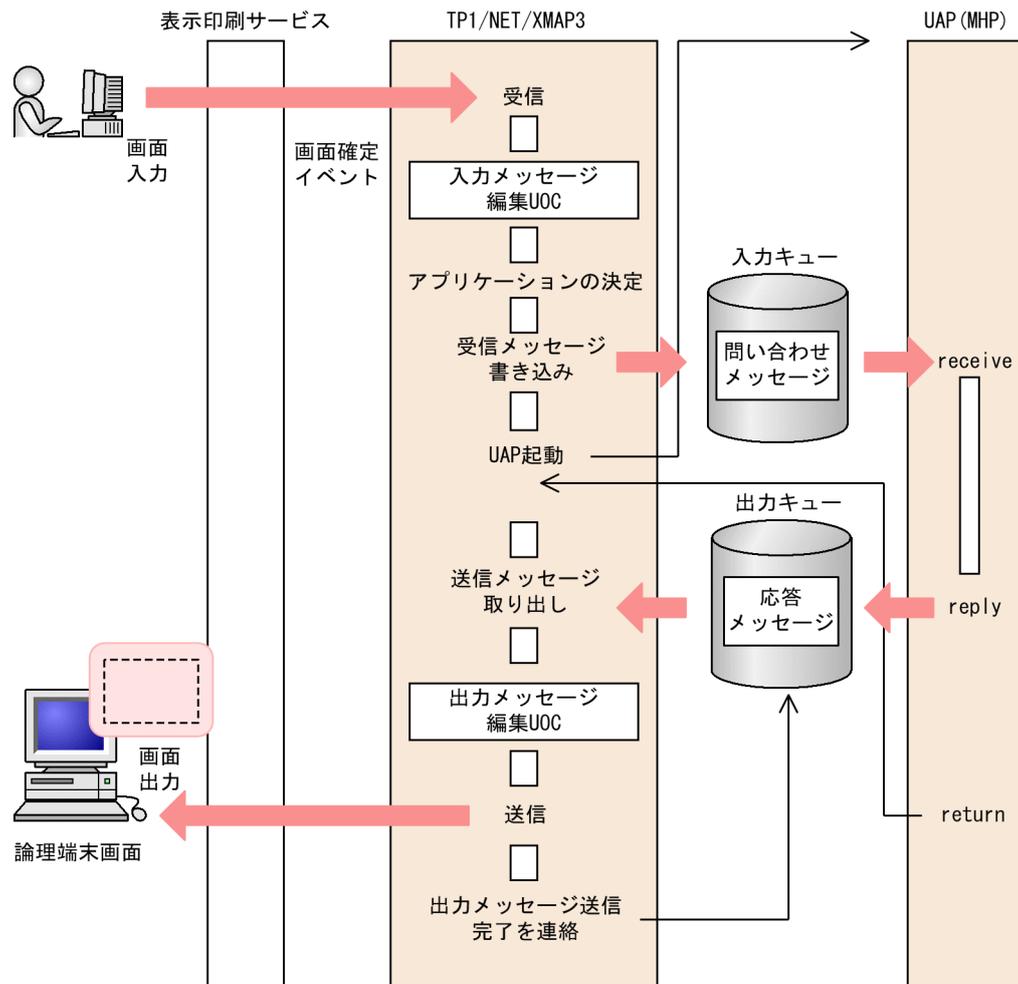
2.4.3 問い合わせ応答形態のメッセージ

TP1/NET/XMAP3 では、応答型 (ans) アプリケーションに発生したメッセージを問い合わせメッセージとして入力キューに登録して、UAP の起動をします。起動された UAP では、receive 関数を呼び出して問い合わせメッセージを受け取ります。また、UAP から reply 関数によって指定されたメッセージを応答メッセージとして出力キューに登録します。TP1/NET/XMAP3 は、この応答メッセージを取り出して表示印刷サービスに送信します。

その後、応答メッセージが送信済みであることを出力キューに連絡します。

問い合わせ応答形態を、次の図に示します。

図 2-11 問い合わせ応答形態



以降、問い合わせ応答形態の詳細を説明します。

(1) 問い合わせ応答の開始と終了

問い合わせ応答は、表示印刷サービスから問い合わせメッセージを受信し、応答型のアプリケーションを起動することで開始します。そして、UAP から応答メッセージを受け付け、表示印刷サービスへの応答メッセージの送信が完了すると問い合わせ応答を終了します。また、障害などによる接続の解放および論理端末の閉塞となった場合にも問い合わせ応答を終了します。

問い合わせ応答の開始から問い合わせ応答の終了までを問い合わせ応答中と呼びます。

(2) 問い合わせ応答中のアプリケーション起動

問い合わせ応答中にアプリケーション起動要求をする場合、非応答型または応答型のアプリケーション名を指定できます。ただし、応答型のアプリケーションを起動できるのは一つのサービスで1回だけです。

応答型のアプリケーションを起動した場合、そのサービスでは応答メッセージを送信できません。起動先の応答型アプリケーションで応答メッセージを送信してください。

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

(3) 問い合わせ応答中の論理端末へのメッセージ受信

問い合わせ応答中の論理端末が表示印刷サービスからメッセージを受信した場合、TP1/NET/XMAP3はメッセージログ (KFCA13664-E) を出力し、メッセージを破棄します。

(4) 問い合わせ応答中の論理端末へのメッセージ送信

問い合わせ応答中の論理端末に対して一方送信メッセージを送信した場合、一方送信メッセージは出力キューに登録され、APIは正常にリターンします。登録された一方送信メッセージは、TP1/NET/XMAP3の取り出しを待ちます。応答メッセージの送信完了後に出力キューから読み出し、表示印刷サービスに送信します。

2.4.4 継続問い合わせ応答形態のメッセージ

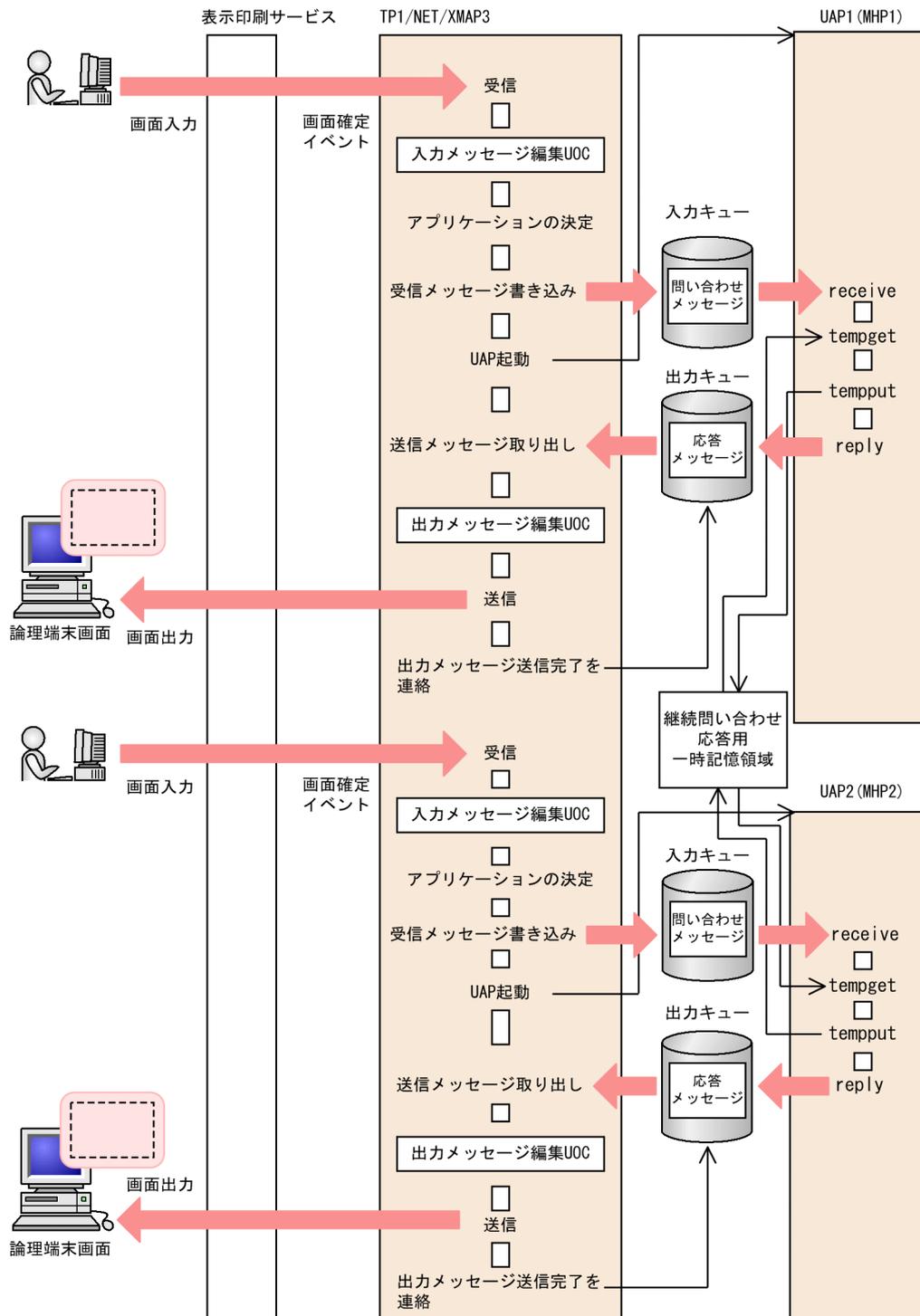
TP1/NET/XMAP3では、継続問い合わせ応答型 (cont) アプリケーションに発生したメッセージによって継続問い合わせ応答の開始を認識します。

起動されたUAPでは、receive関数を呼び出して問い合わせメッセージを受け取ります。また、UAPからreply関数によって指定されたメッセージを応答メッセージとして出力キューに登録します。TP1/NET/XMAP3は、この応答メッセージを取り出して表示印刷サービスに送信します。

継続問い合わせ応答は、UAPがcontend関数(データ操作言語:DISABLE文)を呼び出すか、または運用コマンド(mcftendct)が入力されることによって終了します。

継続問い合わせ応答形態を、次の図に示します。

図 2-12 継続問い合わせ応答形態



以降、継続問い合わせ応答形態の詳細を説明します。

(1) 継続問い合わせ応答の開始

継続問い合わせ応答は、継続問い合わせ応答中でないときに表示印刷サービスから問い合わせメッセージを受信し、継続問い合わせ応答型のアプリケーションを起動することで開始します。

問い合わせ応答と異なり、継続問い合わせ応答では表示印刷サービスへの応答メッセージの送信が完了しても継続問い合わせ応答を終了しません。

継続問い合わせ応答の開始から継続問い合わせ応答の終了までを継続問い合わせ応答中と呼びます。また、継続問い合わせ応答中に UAP を起動してから送信メッセージを読み出すまでを継続問い合わせ応答の UAP 実行中と呼びます。

(2) 次起動アプリケーションの予約

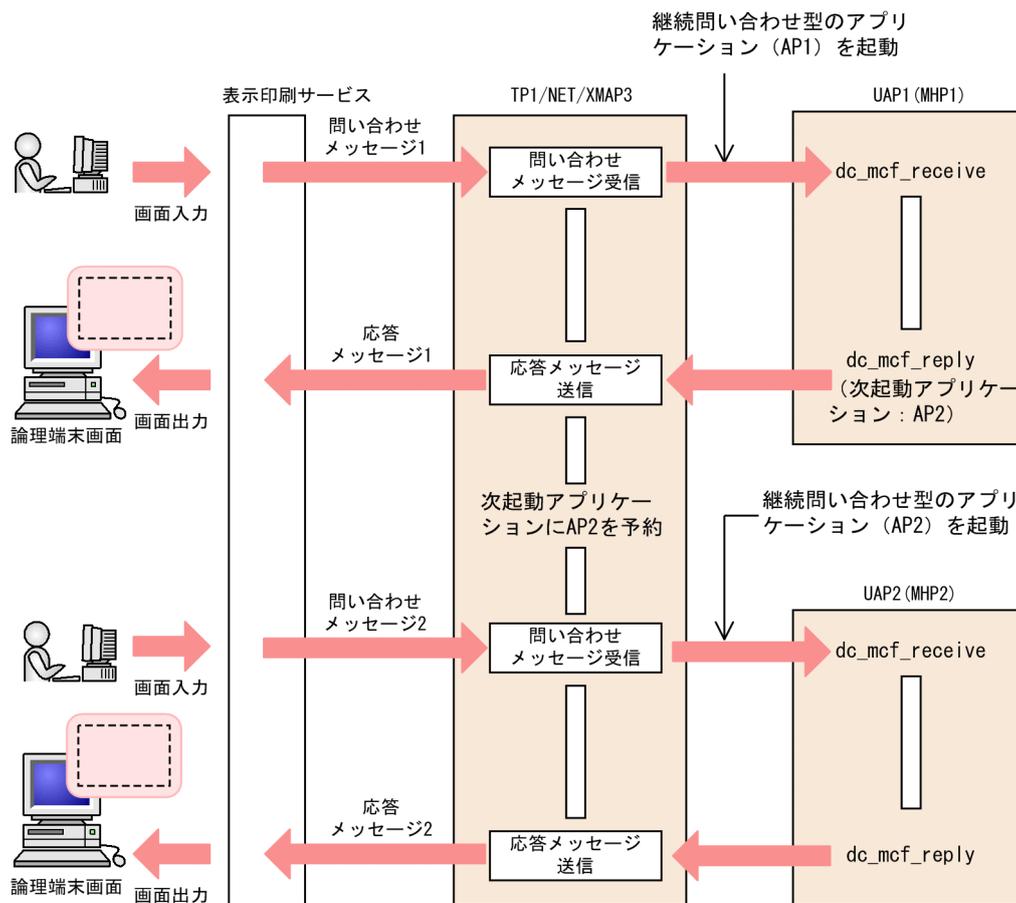
応答メッセージ送信要求時、次に起動する継続問い合わせ応答型のアプリケーション（次起動アプリケーション）を予約できます。

次起動アプリケーションの指定を省略した場合、実行中のアプリケーション名が指定されたものとします。ただし、エラーイベントで次に起動するアプリケーション名の指定を省略した場合は、継続問い合わせ応答を終了します。詳細については、「2.4.4(6) エラーイベント」を参照してください。

次起動アプリケーションの予約を次の図に示します。

使用方法の詳細については、「3. C 言語のライブラリ関数」または「4. COBOL-UAP 作成用プログラムインタフェース」を参照してください。

図 2-13 次起動アプリケーションの予約



(3) 継続問い合わせ応答中のアプリケーション起動

継続問い合わせ応答中にアプリケーション起動要求をする場合、非応答型または継続問い合わせ応答型のアプリケーション名を指定できます。ただし、継続問い合わせ応答型のアプリケーションを起動できるのは一つのサービスで1回だけです。

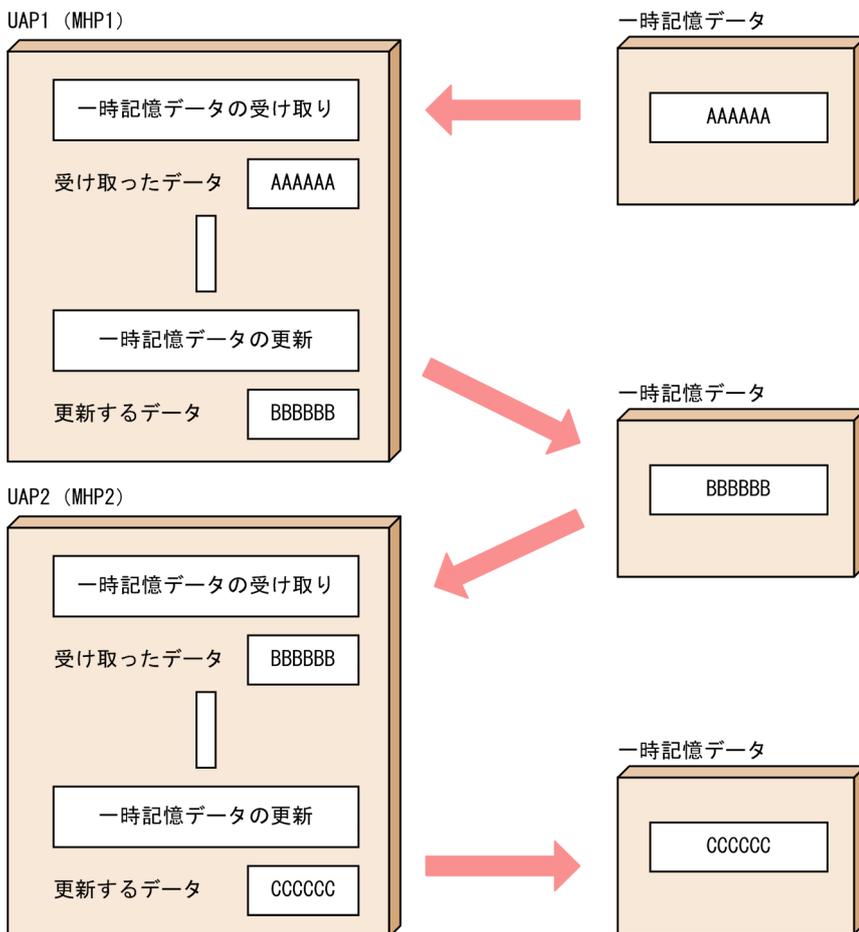
継続問い合わせ応答型のアプリケーションを起動した場合、そのサービスでは応答メッセージを送信できません。また、継続問い合わせ応答の終了要求もできません。起動先の継続問い合わせ応答型アプリケーションで応答メッセージを送信したり、継続問い合わせ応答の終了要求をしてください。

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

(4) 継続問い合わせ応答用一時記憶データ

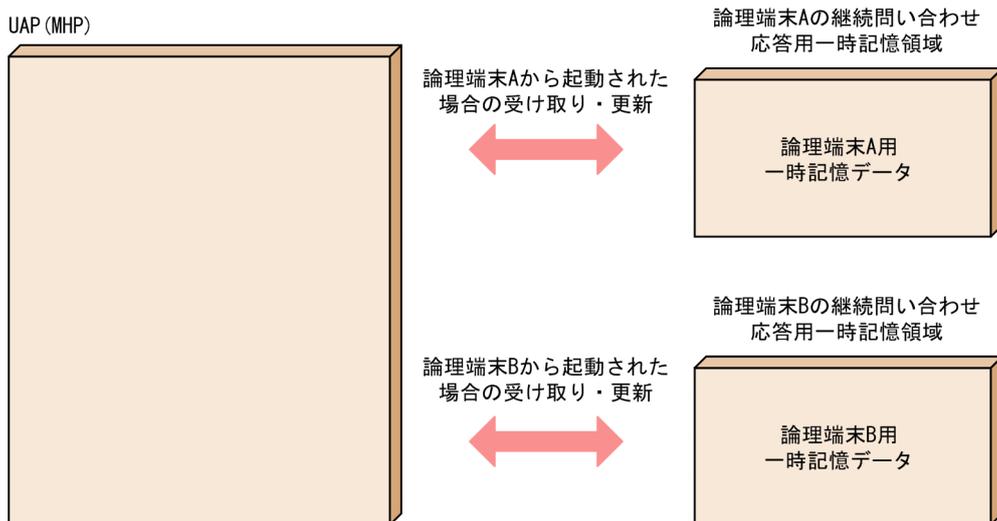
継続問い合わせ応答用一時記憶データ（以降、一時記憶データと呼びます）を使用して、次起動アプリケーションヘデータの引き継ぎができます。データの引き継ぎを、次の図に示します。

図 2-14 一時記憶データの引き継ぎ



一時記憶データを格納する継続問い合わせ応答用一時記憶領域は論理端末ごとに確保するため、複数の論理端末で同時に一つのアプリケーションを使用して継続問い合わせ応答ができます。論理端末と一時記憶データの関係は次の図に示します。

図 2-15 論理端末と一時記憶データの関係



(a) 継続問い合わせ応答用一時記憶領域の確保

一時記憶データを格納する継続問い合わせ応答用一時記憶領域は、更新用領域と回復用領域の2面を共有メモリに確保します。

アプリケーション属性定義 (mcfaalcap) でアプリケーションごとに継続問い合わせ応答用一時記憶領域のサイズを定義できます。

ただし、実際に確保する継続問い合わせ応答用一時記憶領域のサイズは、継続問い合わせ応答を開始したあと最初に一時記憶データの更新要求をしたアプリケーションのアプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドの指定値となります。確保済みの継続問い合わせ応答用一時記憶領域のサイズが、一時記憶データの更新要求を行う場合のデータ長よりも小さいとき、実行中のアプリケーションのアプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドの指定値のサイズで、継続問い合わせ応答用一時記憶領域を再確保します。したがって、継続問い合わせ応答で使用する各アプリケーションの更新する一時記憶データの領域長が異なる場合、最初に起動するアプリケーションのアプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドにデータ長の最大値を指定すると、途中で再確保をしないため、性能が向上します。

(b) 一時記憶データの受け渡し

一時記憶データの受け渡しには「受け取り要求」と「更新要求」があります。

- 一時記憶データの受け取り要求

指定したデータ長の一時記憶データを受け取れます。初期値は、アプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドで指定したサイズ分の初期値(00)₁₆が渡されます。指定した受け取り領域長が一時記憶データより小さい場合、指定した領域長の長さだけ受け取り、残りは切り捨てられます。

- 一時記憶データの更新要求

指定したデータ長のデータを用いて一時記憶データを更新します。最大でアプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドで指定した領域長の更新ができます。

使用方法の詳細については、「3. C 言語のライブラリ関数」または「4. COBOL-UAP 作成用プログラムインタフェース」を参照してください。

(c) 継続問い合わせ応答用一時記憶領域の解放

継続問い合わせ応答を終了すると継続問い合わせ応答用一時記憶領域を解放します。

(d) 障害発生時の一時記憶データ

障害発生時の一時記憶データの扱いについて次に示します。

- OpenTP1 システム障害による全面回復の場合
一時記憶データは失われます。
- UAP 障害による部分回復の場合
異常終了した UAP での一時記憶データの更新要求が取り消され、異常終了した UAP が一時記憶データの受け取り要求を行ったときの状態に戻ります。

(5) 継続問い合わせ応答の終了

継続問い合わせ応答は次のどれかで終了します。

(a) MHP からの終了要求 (contend 関数)

MHP から API (contend 関数) を発行して継続問い合わせ応答の終了を要求します。

ただし、次に示す条件に該当する場合はエラーリターンします。

- 次起動アプリケーションを予約して、応答メッセージの送信を要求している。
- 継続問い合わせ応答型のアプリケーションの起動を要求している。

使用方法の詳細については、「3. C 言語のライブラリ関数」または「4. COBOL-UAP 作成用プログラムインタフェース」を参照してください。

(b) 運用コマンドによる終了要求 (mcftendct)

運用コマンド (mcftendct) で論理端末名称を指定して、その論理端末に対する継続問い合わせ応答を強制終了できます。また、継続問い合わせ応答を行っていない UAP から運用コマンドを発行できます。詳細については、「8. 運用コマンド」を参照してください。

継続問い合わせ応答の UAP 実行中に継続問い合わせ応答を即時強制終了する場合、-f オプションの指定が必要です。

なお、-f オプションを指定した mcftendct コマンドの実行後に MHP が次に示す処理をしたとき、MHP が異常終了します。このとき、エラーイベントは起動しません。

- TP1/Message Control 以外のリソースマネージャにもアクセスする MHP のサービス終了時

- 継続問い合わせ用一時記憶領域アクセス時 (dc_mcf_tempget 関数, dc_mcf_tempput 関数発行時)
- ロールバック要求時 (dc_mcf_rollback 関数発行時 (action : DCMCFRTRY, または DCMCFNRTN の場合))

(c) コネクション解放および論理端末の閉塞

障害などによるコネクションの解放および論理端末の閉塞となった場合に、継続問い合わせ応答を終了します。

相手からのコネクション解放、または運用コマンド (mcftdctcn) や障害によってコネクションを解放する場合、継続問い合わせ応答を処理している MHP が実行中であれば MHP のサービス完了を待ち合わせます。MHP のサービスが完了したあとに継続問い合わせ応答を終了し、コネクションは解放状態となります。

なお、継続問い合わせ応答中の論理端末は、運用コマンド (mcftdctle) による閉塞はできません。

(6) エラーイベント

MHP の処理中に障害が発生した場合、MCF イベント処理用 MHP にエラーイベントを通知します。継続問い合わせ応答中に障害が発生した場合、エラーイベントを継続問い合わせ応答型として起動します。

(a) エラーイベントを定義した場合

MCF イベント処理用 MHP で次のどちらかを行うことで継続問い合わせを続行できます。

- 次起動アプリケーションを指定して、応答メッセージの送信要求をする。
- 継続問い合わせ応答型のアプリケーションの起動要求をする。

MCF イベント処理用 MHP で継続問い合わせ応答型のアプリケーションの起動要求を行わない、かつ、次起動アプリケーションを指定しないで応答メッセージの送信要求をした場合、継続問い合わせ応答処理は終了します。

ただし、MCF イベント処理用 MHP で継続問い合わせ応答型のアプリケーションを起動し、起動先のアプリケーションで次に起動するアプリケーション名を指定しないで応答メッセージの送信要求をした場合、継続問い合わせ応答を終了することなく、MCF イベント処理用 MHP から起動した継続問い合わせ応答型のアプリケーションを次のメッセージ受信時に再び起動します。

(b) エラーイベントを定義しない場合

継続問い合わせ応答処理は終了します。

(7) 継続問い合わせ応答中の論理端末へのメッセージ受信

継続問い合わせ応答中の論理端末が表示印刷サービスからメッセージを受信した場合、応答型と同様に TP1/NET/XMAP3 はメッセージログ (KFCA13664-E) を出力し、メッセージを破棄します。

(8) 継続問い合わせ応答中の論理端末へのメッセージ送信

継続問い合わせ応答中の論理端末に対して一方送信メッセージを送信した場合、一方送信メッセージは出力キューに登録され、APIは正常にリターンします。登録された一方送信メッセージは、TP1/NET/XMAP3の取り出しを待ちます。継続問い合わせ応答の終了後に出力キューから読み出し、表示印刷サービスに送信します。

2.4.5 メッセージの分割と組み立て

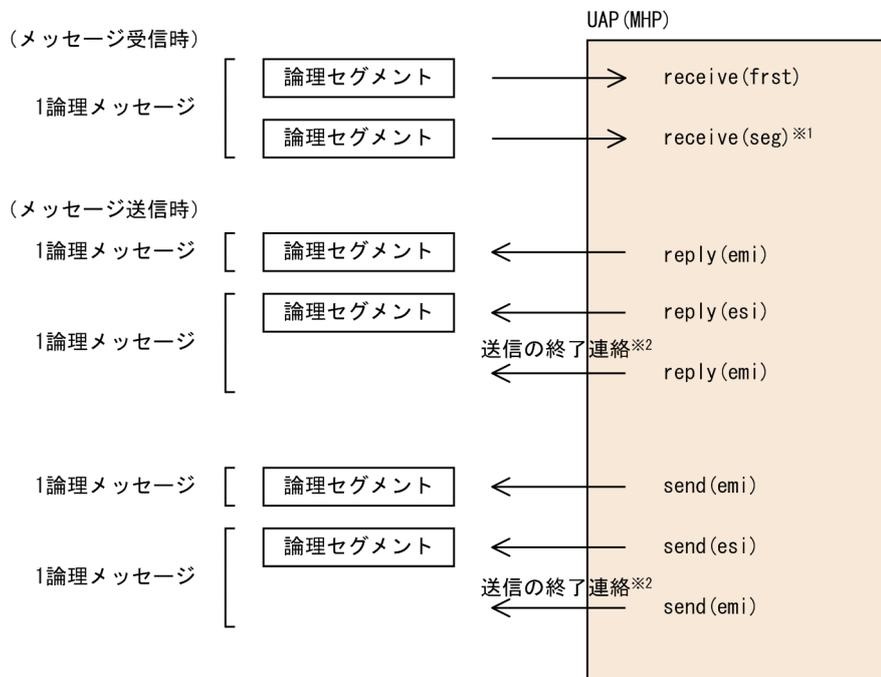
UAPがTP1/NET/XMAP3との間で扱うメッセージの形式を、**論理セグメント群形式**と呼びます。この形式では、一つのメッセージが、一つまたは複数のセグメントで構成されます。ただし、複数のセグメントは、MCFイベントまたはアプリケーション起動で渡されたメッセージを受信する場合だけです。表示印刷サービスから受信するメッセージおよび表示印刷サービスに送信するメッセージは、一つのセグメントで構成されます。

1回のメッセージ送受信では、一つのセグメントの受信や送信をします。したがって、1論理メッセージが複数のセグメントに分割されている場合、UAPでは1論理メッセージの送受信に複数回メッセージ関数呼び出さなければなりません。

メッセージ送信では、1論理メッセージの長さを32000バイト以内にする必要があります。

メッセージ送受信関数の使用方法を、次の図に示します。

図 2-16 メッセージ送受信関数の使用方法



注※1

MCFイベントまたはアプリケーション起動で渡されたメッセージを受信する場合だけです。

注※2

送信するセグメントの長さが0の場合、メッセージの送信の終了がMCFに連絡されます。

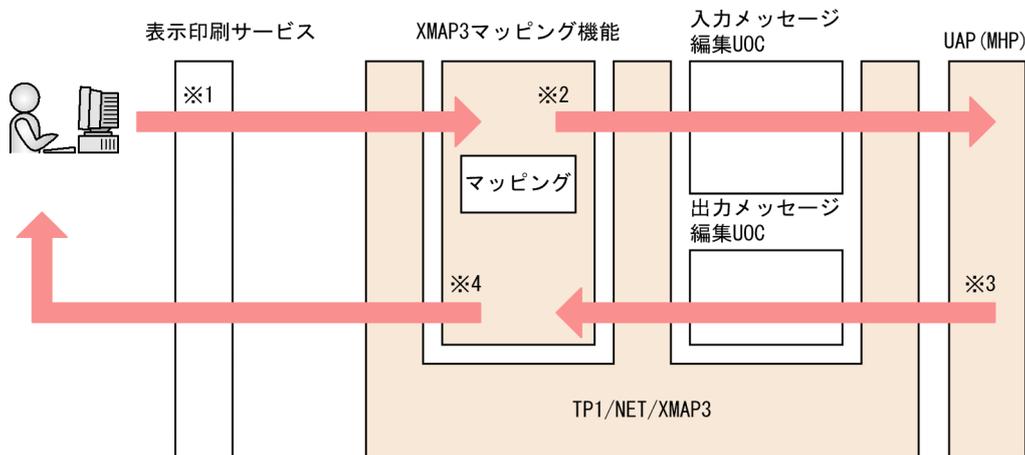
2.4.6 マッピングによるメッセージの編集

表示印刷サービスから TP1/NET/XMAP3 に送られるメッセージは、EBCDIK/KEIS のコード体系を持つ、XP データストリーム形式のメッセージです。XP データストリーム形式のメッセージを物理メッセージと呼びます。一方、OpenTP1 システム内の UAP ではプロセスの環境に応じて、JIS8/シフト JIS、または EUC のコード体系を持つ、論理セグメント群形式のメッセージを扱います。論理セグメント群形式とは、マップ定義に従って生成された論理マップの形式です。

TP1/NET/XMAP3 では、XMAP3 マッピング機能を使って、物理メッセージと論理セグメント群形式のメッセージとの相互変換、および編集をします。メッセージの形式を変換し、メッセージを編集することをマッピングと呼びます。TP1/NET/XMAP3 は、表示印刷サービスからの物理メッセージを論理セグメント群形式のメッセージにマッピングして、OpenTP1 システム内の UAP に渡します。同様に、OpenTP1 システム内の UAP からの論理セグメント群形式のメッセージを、物理メッセージにマッピングして表示印刷サービスに渡します。

マッピングによるメッセージ編集の流れを、次の図に示します。

図 2-17 マッピングによるメッセージ編集の流れ



注※1

入力物理メッセージ (EBCDIK/KEIS)

注※2

入力論理セグメント群 (JIS8/シフト JIS または EUC) ※5

注※3

出力論理セグメント群 (JIS8/シフト JIS または EUC) ※5

注※4

出力物理メッセージ (EBCDIK/KEIS)

注※5

マップ定義で、入力マッピング後の論理セグメント群のメッセージがなくなるようなマップを定義しないでください。定義した場合は、TP1/NET/XMAP3 の動作を保証しません。したがって、入力メッ

ページ編集 UOC では入力マッピング後の論理セグメント群のメッセージがなくなることを意識する必要はありません。

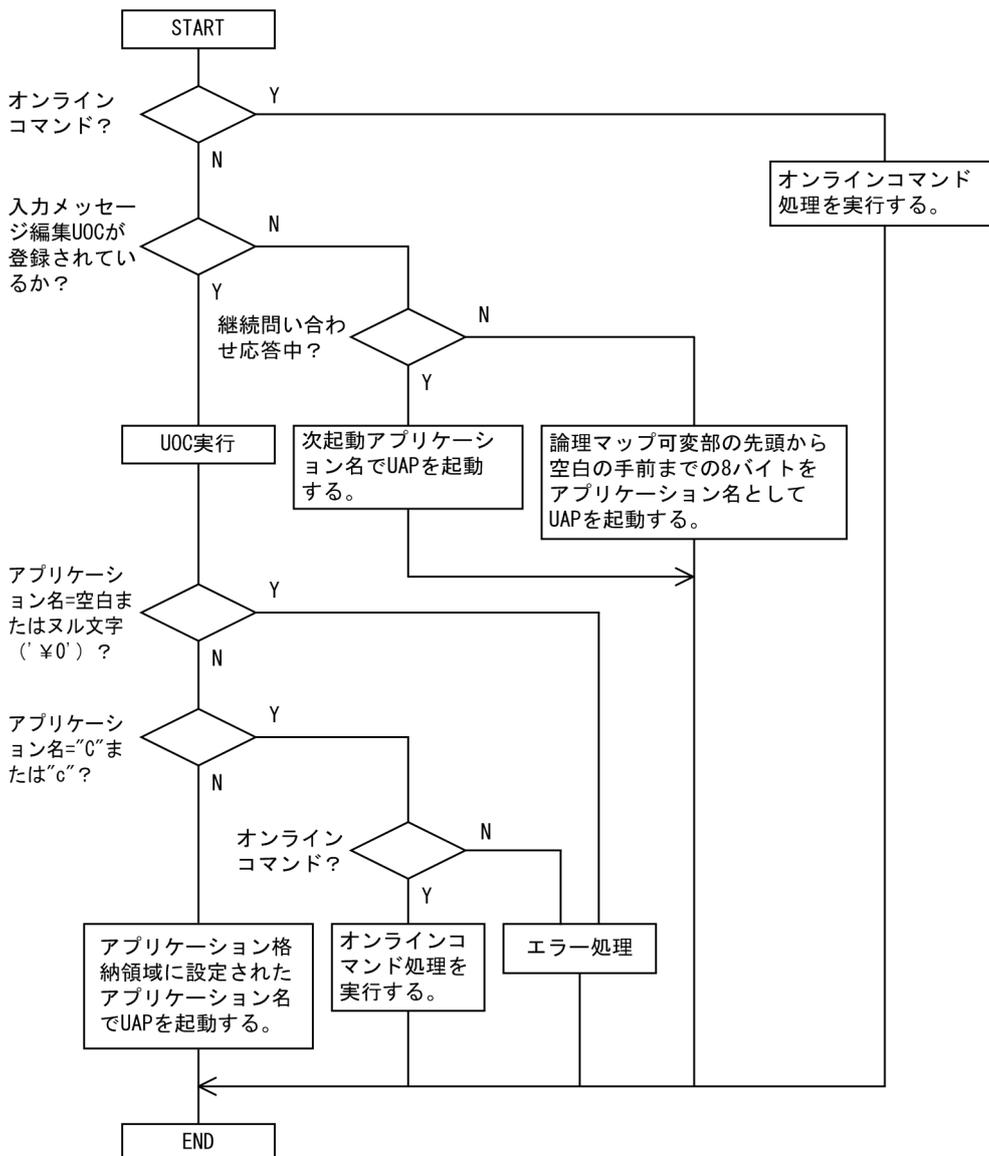
また、UAP からの reply, または send 関数によるメッセージなしの送信はできません。

出力メッセージ編集 UOC では出力マッピング前の論理セグメント群のメッセージがなしの場合を意識する必要はありません。

2.4.7 アプリケーションの決定

TP1/NET/XMAP3 では、メッセージを受信するとアプリケーションを決定して UAP を起動します。TP1/NET/XMAP3 でのアプリケーション決定の流れを次の図に示します。

図 2-18 アプリケーション決定の流れ



初回を除く継続問い合わせ応答中のメッセージ受信時のアプリケーション名は、前回起動された UAP が発行した reply 関数で指定した次起動アプリケーション名になります。この時、UOC で次起動アプリケーション名を更新した場合は、更新後のアプリケーション名が有効となります。

(1) ユーザ任意のアプリケーション名

受信メッセージの先頭にアプリケーション名が設定されるようにしたり、入力メッセージ編集 UOC でアプリケーション名を決定したりすることで、ユーザ任意のアプリケーション名で UAP を起動できます。

アプリケーション名は、受信メッセージの先頭から最初に空白が現れるまでの 8 けた以内の識別子です。先頭から 9 バイト目までに空白がない場合、または先頭に空白がある場合はアプリケーション名不正とし、ERREVT1 を起動します。文字コードは該当する動作マシンの文字コードで識別します。

OpenTP1 が受信するメッセージの形式は、XMAP3 のドローのデータ名編集ダイアログで決定します。データ名編集ダイアログの最上位のデータが、OpenTP1 が受信する入力メッセージの先端になります。

また、XMAP3 のドローの画面属性ダイアログで、隠しフィールドを設定すると、隠しフィールドで設定した値が自動的にデータ名編集ダイアログの最上位となります。

詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド」を参照してください。

(2) システムで使用するアプリケーション名

システムで使用するアプリケーション名を次の表に示します。

このアプリケーション名はユーザのアプリケーション名としては使用できません。

表 2-6 システムで使用するアプリケーション名

アプリケーション名	目的
"C", "c"	"C"または"c"のアプリケーション名（入力メッセージの先端が"C△"または"c△"で始まるアプリケーション名）は、オンラインコマンドの入力として扱います。

例えば、XMAP3 のドローで、"C△"または"c△"を入力できるフィールドを用意し、それをデータ名編集ダイアログで最上位に配置すれば、オンラインコマンドを実行できるマップの生成ができます。

2.4.8 メッセージグループ送信機能

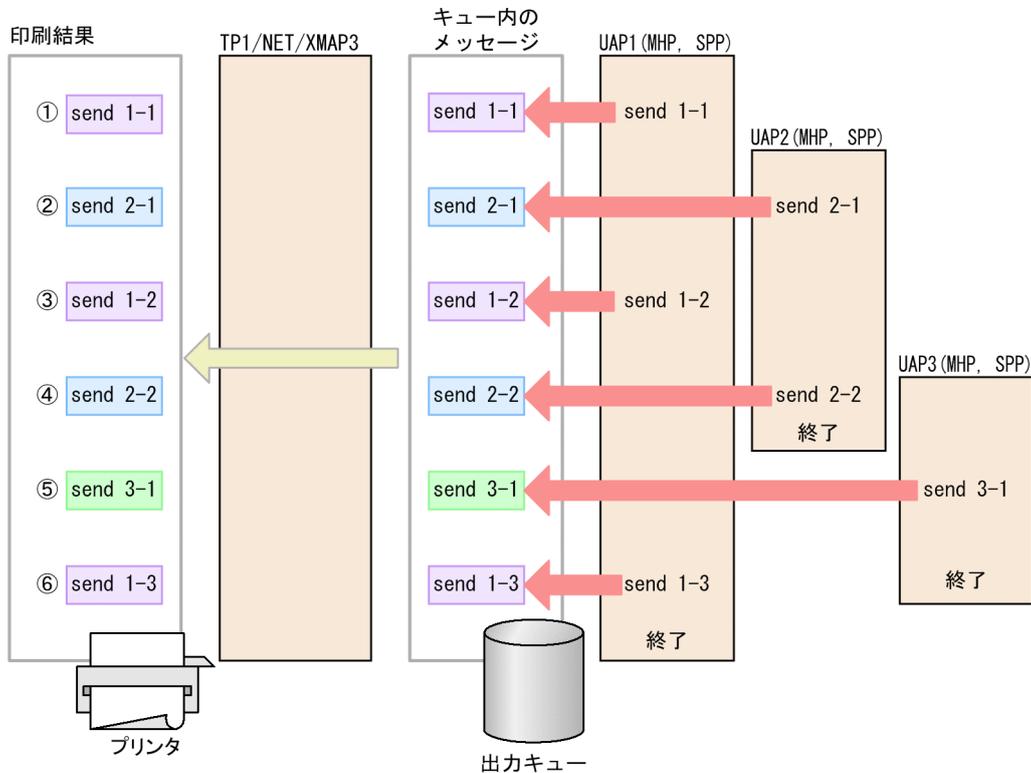
UAP からの複数の一方送信メッセージを、トランザクションブランチ単位で一まとまりのメッセージとして扱えます。これをメッセージグループ送信機能と呼びます。ユーザはメッセージグループ送信するかどうかを、論理端末ごとに論理端末定義 (mcftalcle) の-G オプションで指定します。

メッセージグループ送信機能を使用しない場合、一方送信メッセージは send 関数の発行順に出力キューに登録され、表示印刷サービスに送信されます。そのため、複数の UAP から同一の論理端末に対して同

時に分岐送信を行うと、帳票が混ざって印刷されます。また、複数の UAP から一方送信メッセージを送信する場合、先に一方送信メッセージを送信した UAP がコミットまたは終了するまでの間は、別の UAP の一方送信メッセージは取り出されません。

メッセージグループ送信機能を使用しない場合の印刷結果を次の図に示します。

図 2-19 メッセージグループ送信機能を使用しない場合の印刷結果



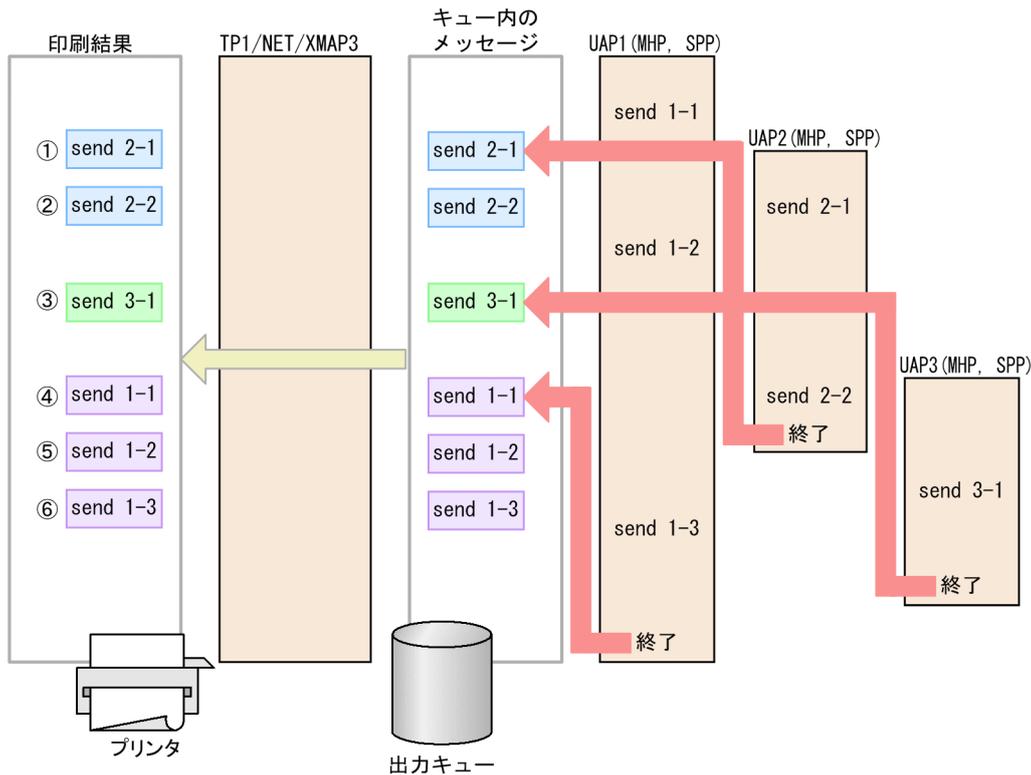
(凡例)

①: 出力キューから取り出されるメッセージの順番

この図に示すように、複数の UAP からの一方送信メッセージが混ざって印刷されます。また、UAP2 および UAP3 の一方送信メッセージは、UAP1 が終了するまで出力キューから取り出されません。

一方、メッセージグループ送信機能を使用すると、UAP のコミットまたは終了順に出力キューに登録されます。次の図に示すように、帳票はトランザクション単位に一まとまりで印刷されます。

図 2-20 メッセージグループ送信機能を使用する場合の印刷結果



(凡例)

①: 出力キューから取り出されるメッセージの順番

注意事項

同一プリンタに複数の論理端末が対応づけられた構成で複数枚の帳票を出力する場合、グループ送信機能を使用しても複数枚の帳票の間に別のトランザクションの帳票が混ざることがあります。また、複数枚の帳票を一つの PDF ファイルに出力する場合、PDF ファイルが分割されたり、上書きによるページ抜けが発生したりすることがあります。

このような問題を解決するために、メッセージグループ送信機能を使用した上で、XMAP3 で次の対応を実施してください。

適用 OS が UNIX の場合

XMAP3 Server の「LP スプールシステムでの帳票一括出力機能」を有効にしてください。「LP スプールシステムでの帳票一括出力機能」の詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

適用 OS が Windows の場合

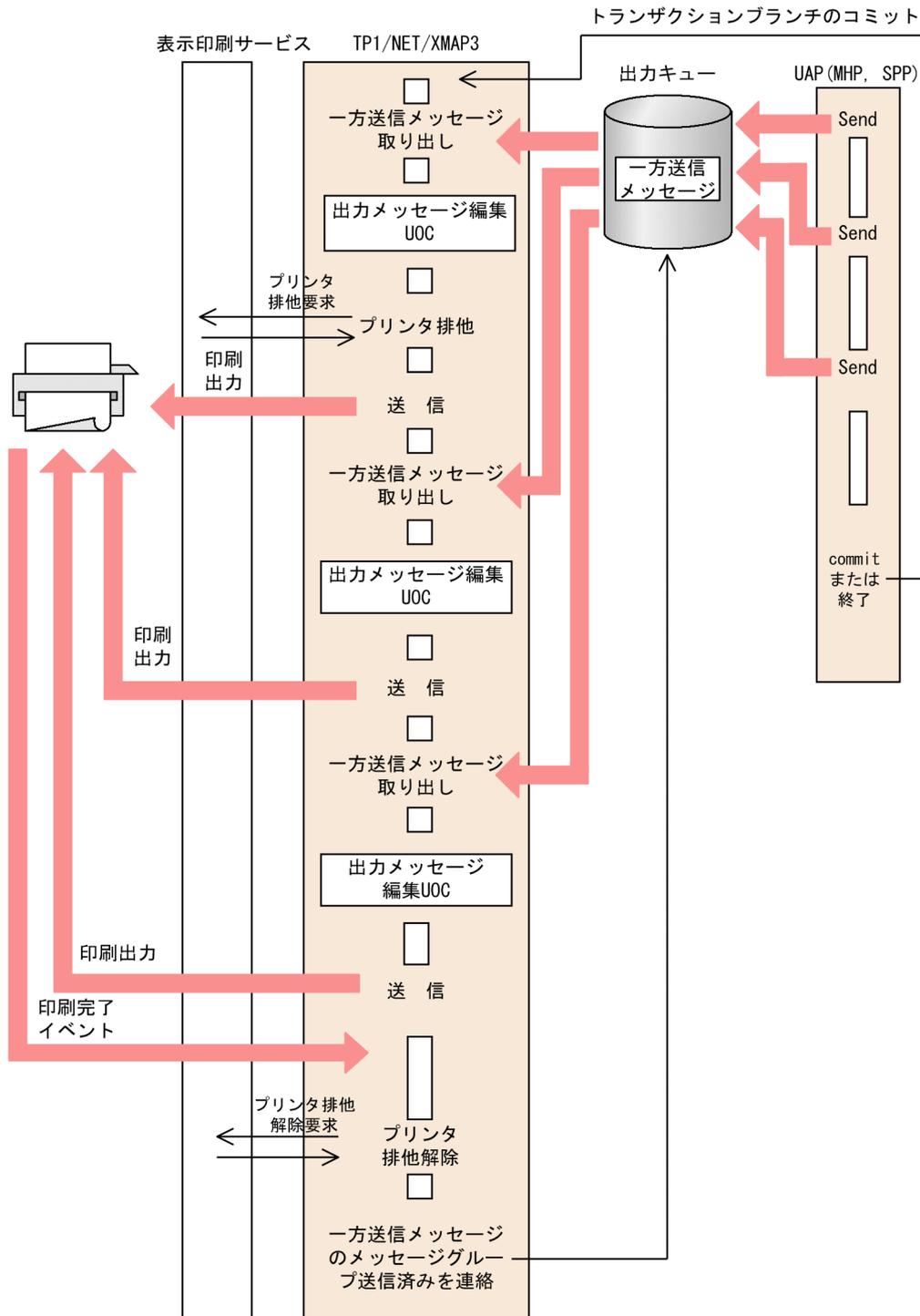
XMAP3 の表示・印刷セットアップダイアログで、スプール書き出し単位に「アプリケーション毎」を指定してください。表示・印刷セットアップの詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

(1) メッセージグループ送信の流れ

メッセージグループ送信の流れについて説明します。UAP は、メッセージグループ送信をすると、定義した論理端末に対する一方送信メッセージを出力キューに登録します。TP1/NET/XMAP3 は、UAP のトランザクションブランチがコミットまたは終了すると、最初のメッセージを出力キューから取り出し、プリンタを排他します。そして、メッセージの出力キューからの取り出しとプリンタへの送信を繰り返します。送信後、TP1/NET/XMAP3 はプリンタからの印刷完了イベントを待ちます。印刷完了イベントを受け取ると、プリンタの排他を解除します。

メッセージグループ送信の流れを次の図に示します。

図 2-21 メッセージグループ送信の流れ



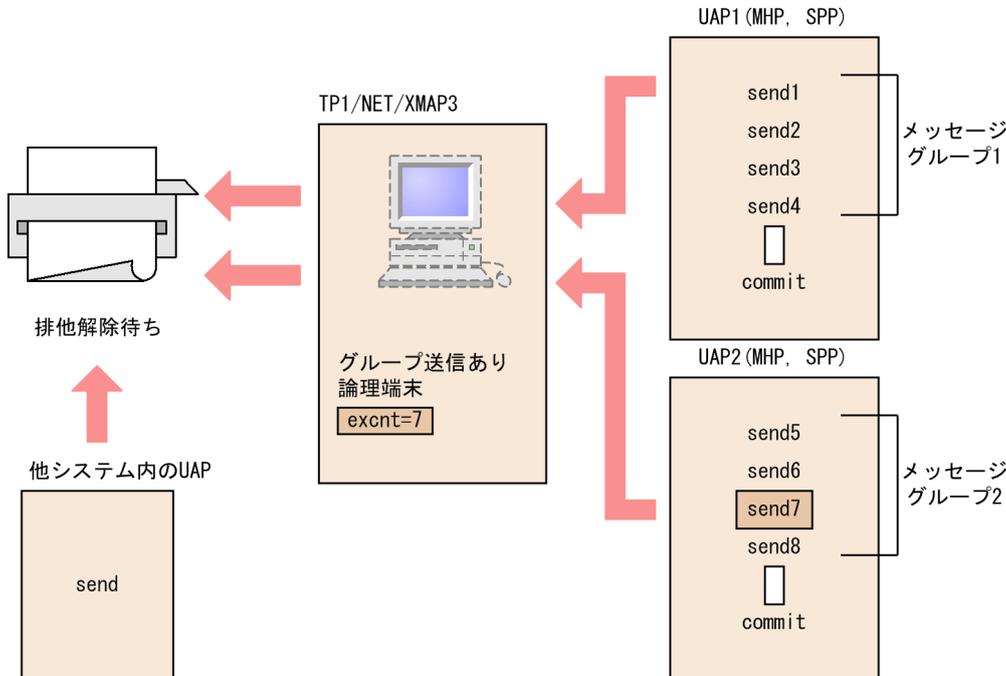
(2) プリンタの連続使用

論理端末定義 (mcftalcle -x) の excnt オペランドで、プリンタの排他を解除するタイミングを、送信メッセージの数で指定できます。指定されたメッセージ数に達するまでプリンタの排他を続けるため、複数のメッセージグループを連続して印刷できます。excnt オペランドに 0 を設定すると、個々のメッセージグループごとにプリンタの排他を解除します。

excncnt オペランドに 0 以外の値を設定している場合、両面印刷の指定をすると、グループの先頭ページが直前のグループの最終ページの裏面に印刷されることがあります。グループの先頭メッセージで改ページするなどの対処をしてください。

プリンタの連続使用を次の図に示します。

図 2-22 プリンタの連続使用



この図では、OpenTP1 システム内の UAP が送信する 7 番目のメッセージ (send7) を含むメッセージグループ 2 の送信が完了したとき、プリンタの排他が解除されます。

2.4.9 メッセージの代行送信

メッセージの代行送信とは、メッセージ送信先論理端末の端末状態が閉塞、またはメッセージ送信先のプリンタが障害の場合に、ほかのプリンタに送信することです。

代行送信は、同じ MCF 通信構成定義で定義したプリンタ属性の論理端末同士の場合にできます。

TP1/NET/XMAP3 では、メッセージ送信の代行を依頼する論理端末を代行元論理端末、メッセージ送信を代行する論理端末を代行先論理端末と呼びます。メッセージグループ送信機能、またはページ制御機能を使用している場合、TP1/NET/XMAP3 では代行送信によるプリンタ出力の結果を保証しません。

(1) 代行送信の開始

オンラインコマンド (ACT ALTSEND)、または運用コマンド (mcfstalt) を入力して開始します。

代行先論理端末の端末状態が閉塞していると代行送信できません。また、代行先のプリンタが障害中の場合は、代行送信できません。

代行元論理端末は端末状態が閉塞，またはプリンタ障害中でなければなりません。

代行元論理端末がプリンタ排他再試行の間は，代行送信できません。プリンタ排他再試行の終了を待つか，運用コマンドで端末状態を閉塞してから再度コマンドを入力してください。

(2) 代行送信の終了

代行送信の終了には，次の五つの場合があります。

- 運用コマンド (mcftedalt) を入力して，代行送信を終了します。
- オンラインコマンド (DEACT ALTSEND) を入力して，代行送信を終了します。
- 代行元論理端末の端末状態が閉塞解除になると，TP1/NET/XMAP3 が代行送信を終了します。
- 代行先論理端末のプリンタが障害になると，TP1/NET/XMAP3 が代行送信を終了します。
- 代行先論理端末の端末状態が閉塞になると，TP1/NET/XMAP3 が代行送信を終了します。

2.4.10 キー入力サービス

TP1/NET/XMAP3 では，XMAP3 Server の機能によって論理端末画面からのキー入力サービスを提供しています。

(1) 送信要求発生時の処理

TP1/NET/XMAP3 での送信要求発生（論理端末画面からの画面確定キー入力）時の処理を，次の表に示します。

表 2-7 送信要求発生時の処理

項番	キー		TP1/NET/XMAP3 の処理	XMAP3 Server のキーシンボル名称 (マクロ値)
	オンライン用キー	AX 仕様キー		
1	実行	Enter (テンキー部)	アプリケーション決定後 UAP 起動	TX_MKLN FED
2	PA1	PageUp		TX_MKPA1
3	PA2	PageDown		TX_MKPA2
4	PA3	Ctrl+PageDown		TX_MKPA3
5	スクリーン	Ctrl+Delete		TX_MKSCREN
6	中断	Ctrl+S		TX_MKBREAK
7	割込	Break Ctrl+Pause		TX_MKINTER

項番	キー		TP1/NET/XMAP3 の処理	XMAP3 Server のキー シンボル名称 (マクロ値)
	オンライン用キー	AX 仕様キー		
8	印刷	Ctrl+P	印刷キー処理 (論理ハードコピーの出力)	TX_MKPRINT
9	PF1	F1	アプリケーション決定後 UAP 起動	TX_MKF1
10	PF2	F2		TX_MKF2
11	PF3	F3		TX_MKF3
12	PF4	F4		TX_MKF4
13	PF5	F5		TX_MKF5
14	PF6	F6		TX_MKF6
15	PF7	F7		TX_MKF7
16	PF8	F8		TX_MKF8
17	PF9	F9		TX_MKF9
18	PF10	F10		TX_MKF10
19	PF11	F11		TX_MKF11
20	PF12	F12		TX_MKF12
21	コントロール+PF1	Shift+F1		TX_MKF13
22	コントロール+PF2	Shift+F2		TX_MKF14
23	コントロール+PF3	Shift+F3		TX_MKF15
24	コントロール+PF4	Shift+F4		TX_MKF16
25	コントロール+PF5	Shift+F5		TX_MKF17
26	コントロール+PF6	Shift+F6		TX_MKF18
27	コントロール+PF7	Shift+F7		TX_MKF19
28	コントロール+PF8	Shift+F8		TX_MKF20
29	コントロール+PF9	Shift+F9		TX_MKF21
30	コントロール +PF10	Shift+F10		TX_MKF22
31	コントロール +PF11	Shift+F11		TX_MKF23
32	コントロール +PF12	Shift+F12		TX_MKF24

その他の画面確定キーの入力については表示印刷サービスが処理しますが、入力エラーとなってブザーが鳴ります。

次メッセージ要求キーとして使用しているキーは、UAP 起動や論理ハードコピーの出力をするキーとしては使用できません。

注意事項

XMAP3 のドロウの画面属性ダイアログで隠しフィールドを設定している場合、XMAP3 Server 表示・印刷環境ファイルまたは XMAP3 の表示・印刷セットアップダイアログで、強制確定キーの動作の「入力済みデータを送らない」をオフにしてください。

強制確定キーの動作の「入力済みデータを送らない」をオンにした場合、隠しフィールドに設定した値が TP1/NET/XMAP3 に通知されません。

(2) 印刷キー処理

TP1/NET/XMAP3 では、表示印刷サービスから印刷キーの入力を受け付けると、表示印刷サービスに対して論理ハードコピーの要求をします。

論理ハードコピーの出力先プリンタは、XMAP3 Server 表示・印刷環境ファイルの表示サービス名 (DCPSNM) の指定に従います。ここで指定する表示サービス名と、コネクション定義 (mcftalccn -S) の svname オペランドで指定する接続対象の表示印刷サービス名とは、重ならないように注意してください。二重に指定した場合、論理ハードコピーが正しく行われなことがあります。

XMAP3 Server 表示・印刷環境ファイルに表示サービス名の指定がない場合は、印刷キーを入力しても論理ハードコピーは取得できません。

また、エラーなどによって論理ハードコピーの取得ができない場合は、オペレータインジケータに表示します。

(3) キーボード状態の変更

論理端末からのメッセージを OpenTP1 システムが受信すると、TP1/NET/XMAP3 は論理端末定義 (mcftalcle) と UAP の型に応じて、論理端末画面のキーボードロックを解除するかどうかを決定します。論理端末画面のキーボード状態を次の表に示します。

表 2-8 論理端末画面のキーボード状態

論理端末定義 (mcftalcle)		アプリケーションの型 ^{*1}	メッセージ入力時のキーボード状態	キーボードロックを解除するタイミング
keyboard オペランド	unlock オペランド			
aplock	send	cont	ロック状態	応答メッセージ送信時 ^{*2*3*4}
		ans	ロック状態	応答メッセージ送信時 ^{*2*3*4}
		noans (MHP)	ロック状態	一方送信メッセージ送信時
		noans (SPP)	ロック解除状態	—

論理端末定義 (mcftalcle)		アプリケーションの型※1	メッセージ入力時のキーボード状態	キーボードロックを解除するタイミング
keyboard オペランド	unlock オペランド			
aplock	auto	cont	ロック状態	応答メッセージ送信時※2※3※4
		ans	ロック状態	応答メッセージ送信時※2※3※4
		noans (MHP)	ロック状態	MHP (正常, 異常) 終了時※5
		noans (SPP)	ロック解除状態	—
aptype	定義不可	cont	ロック状態	応答メッセージ送信時※3※4
		ans	ロック状態	応答メッセージ送信時※3※4
		noans (MHP)	ロック解除状態	—
		noans (SPP)	ロック解除状態	—

(凡例)

— : 該当しません。

注※1

論理端末からのメッセージを処理するために起動されるアプリケーションの型です。

注※2

keyboard オペランドの指定は無効になります。この場合には、aptype が指定されたものとして扱います。

注※3

アプリケーション起動によって応答権をほかの UAP に委譲した場合、キーボードロック解除のタイミングは、応答権を委譲されたアプリケーションが応答メッセージを送信したときになります。

注※4

起動されたアプリケーション (MHP) の障害などによってエラーイベントが起動された場合、このエラーイベントに応答権が委譲されます。キーボードロック解除のタイミングはエラーイベントが応答メッセージを送信したときになります。ただし、エラーイベント未定義時はアプリケーション (MHP) 終了時にキーボードロックが解除されます。

注※5

起動されたアプリケーション (MHP) の障害などによってエラーイベントが起動された場合、キーボードロックはエラーイベントが終了するまで解除されません。

なお、キーボードロック解除のタイミングがメッセージ送信時の場合、ロックを解除するかどうかは、メッセージ送信時に指定したマップのマップ定義に従います。ロックを解除する場合のマップ定義には、PHSEG 文の CNTRL オペランドに FRKB を指定してください。PHSEG 文の CNTRL オペランドを省略した場合、または PHSEG 文の CNTRL オペランドに NOFR を指定した場合は、ロックが解除されません。

キーボードロックを [取消] キーなどで解除して、新たなメッセージを入力した場合には、入力された新たなメッセージは破棄されます。ただし、次の場合には入力された新たなメッセージは破棄されず、正常に受け付けられます。

- 論理端末定義 (mcftalcle -K) の keyboard オペランドで aplock を指定し、かつ unlock オペランドで send を指定した論理端末からの、非応答型アプリケーションに対するメッセージ入力によって、キーボードがロック状態になっていた場合。

2.4.11 オンラインコマンド

TP1/NET/XMAP3 では、論理端末画面からオンラインコマンドを入力できます。アプリケーション名が"C"または"c"の受信メッセージをオンラインコマンドの入力と判断して処理します。

オンラインコマンドの詳細については、「[9. オンラインコマンド](#)」を参照してください。

2.4.12 オペレータインジケータのメッセージ表示

TP1/NET/XMAP3 は、次の場合、画面にメッセージを表示します。

- オンラインコマンドの処理を終了した場合（途中で処理を打ち切った場合を含む）
- 論理ハードコピーの取得に失敗した場合
- システム終了中や、論理端末のキュー状態が閉塞中のため、入力メッセージの受け付けを拒否した場合
- 運用コマンドによって、継続問い合わせ応答を強制終了した場合

メッセージは、CUI 画面、または GUI 画面のオペレータインジケータ[※]に表示されます。なお、オペレータインジケータ表示メッセージは、次の操作で消去されます。

- 画面からメッセージを入力した場合
- 次メッセージ要求キーを押した場合
- 印刷キーを押した場合

GUI 画面で複数ウィンドウを開いている場合は、すべてのウィンドウにメッセージを表示します。

オペレータインジケータ表示メッセージの詳細については、「[付録 I オペレータインジケータ表示メッセージ一覧](#)」を参照してください。

注※

オペレータインジケータの 45 けたから 80 けたまでの 36 けたを使用します。なお、画面のサイズによっては、メッセージが最後まで表示されないことがあります。

2.4.13 クライアント PC/WS 応答時間タイムアウト発生時の障害メッセージ出力

XMAP3 Server では、クライアント PC/WS の応答時間の監視ができます。この機能によって、クライアント PC の異常終了、および TCP/IP ドライバのデータ破壊による、無限待ちの状態を回避できます。

TP1/NET/XMAP3 では、この監視時間のタイムアウトの発生を、TP1/NET/XMAP3 の障害メッセージの出力によって通知できます。

クライアント応答時間監視機能を使用する場合、TP1/NET/XMAP3 通信サービスのシステムサービス情報定義で、環境変数 XRESPONSETIME または環境変数 XPCLOSEWAITTIME を指定する必要があります。システムサービス情報定義の詳細については、「7. システム定義」の「システムサービス情報定義」を参照してください。

クライアント PC/WS 応答時間タイムアウト発生時に障害メッセージを出力させるには、XMAP3 共通定義 (mcftxp -m) の timeoutlog オペランドに yes を指定します。mcftxp 定義コマンドの詳細については、「7. システム定義」の「mcftxp (XMAP3 共通定義)」を参照してください。

3

C 言語のライブラリ関数

この章では、TP1/NET/XMAP3 で使用できる、C 言語のライブラリ関数について説明します。

C 言語のライブラリ関数の一覧

TP1/NET/XMAP3 で使用する C 言語のライブラリ関数の一覧を、次の表に示します。

表 3-1 C 言語のライブラリ関数の一覧

関数名	機能
dc_mcf_contend	継続問い合わせ応答の終了
dc_mcf_receive	メッセージの受信
dc_mcf_reply	応答メッセージの送信
dc_mcf_resend	メッセージの再送
dc_mcf_send	メッセージの送信
dc_mcf_tactcn	接続の確立
dc_mcf_tactle	論理端末の閉塞解除
dc_mcf_tdctcn	接続の解放
dc_mcf_tdctle	論理端末の閉塞
dc_mcf_tempget	一時記憶データの受け取り
dc_mcf_tempput	一時記憶データの更新
dc_mcf_tlscn	接続の状態取得
dc_mcf_tlsle	論理端末の状態取得

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

NULL またはヌル文字列設定時のコーディング例

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

NULL を設定する場合

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

ヌル文字列を設定する場合

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

注

resv01 以外の dc_mcf_receive 関数の引数は省略しています。

dc_mcf_contend – 継続問い合わせ応答の終了 (C 言語)

形式

ANSI C, C++の形式

```
#include <dcmcf.h>
int dc_mcf_contend (DCLONG action, char *resv01)
```

K&R 版 C の形式

```
#include <dcmcf.h>
int dc_mcf_contend (action, resv01)
DCLONG          action;
char             *resv01;
```

機能

継続問い合わせ応答を終了します。継続問い合わせ応答を終了させるときは、その MHP で呼び出した dc_mcf_reply 関数の nextap がヌル文字列であること、および継続問い合わせ応答型のアプリケーションを起動させる dc_mcf_execap 関数を呼び出していないことが前提です。次起動アプリケーションを dc_mcf_reply 関数の nextap に設定している場合、または継続問い合わせ応答型のアプリケーションを起動させる dc_mcf_execap 関数を呼び出している場合は、dc_mcf_contend 関数はエラーリターンします。

dc_mcf_contend 関数を呼び出したあとは、一時記憶データにアクセスする関数 (dc_mcf_tempget 関数、dc_mcf_tempput 関数) は使用できません。

UAP で値を設定する引数

●action

DCNOFLAGS を設定します。

●resv01

ヌル文字列を設定します。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_72000	-13000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、 dc_mcf_contend 関数を呼び出しています。
		< SPP の実行でリターンした場合 >

リターン値	リターン値 (数値)	意味
DCMCFRTN_72000	-13000	SPP では dc_mcf_contend 関数を呼び出せません。
DCMCFRTN_72016	-13016	action に設定した値が間違っています。
		resv01 に設定した値がヌル文字列になっていません。
DCMCFRTN_72101	-13101	継続問い合わせ応答型でないアプリケーションで、dc_mcf_contend 関数を呼び出しています。
DCMCFRTN_72107	-13107	dc_mcf_contend 関数を 2 回以上呼び出しています。
DCMCFRTN_72111	-13111	次起動アプリケーション (nextap) を設定して dc_mcf_reply 関数を呼び出したあと、dc_mcf_contend 関数を呼び出しています。
		継続問い合わせ応答型のアプリケーション名を設定して dc_mcf_execap 関数を呼び出したあと、dc_mcf_contend 関数を呼び出しています。
上記以外	—	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

— : 該当しません。

dc_mcf_receive – メッセージの受信 (C 言語)

形式

ANSI C, C++の形式

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

K&R 版 C の形式

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

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

機能

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

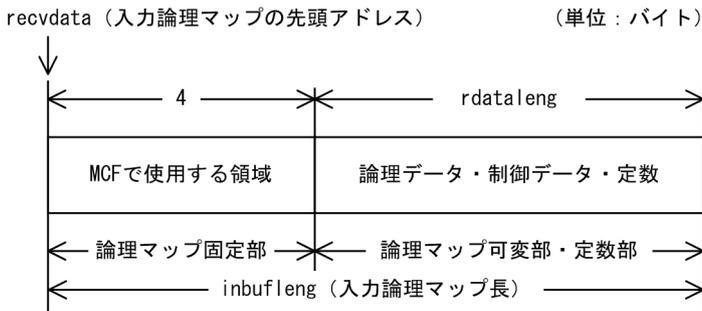
受信できるメッセージの一つのセグメントの最大長は、32763 バイトまでです。

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

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

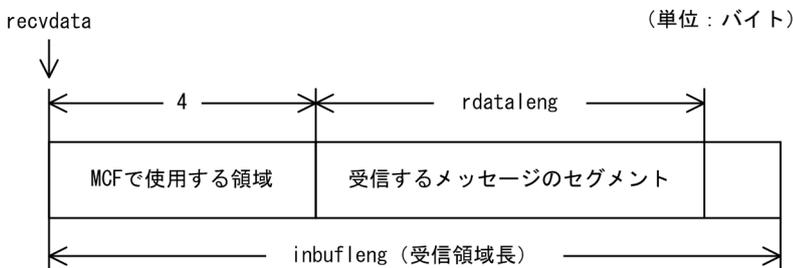
セグメントを受信する領域 (recvdata で示す領域) の形式を次に示します。

TP1/NET/XMAP3 の論理端末画面から入力されたメッセージを受信する場合



XMAP3 で作成した入力論理マップ (拡張子が [.h] のファイル) を #include で取り込み, 入力論理マップの先頭アドレスを recvdata に指定します。入力論理マップの詳細については, マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」, またはマニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド」を参照してください。コーディング例については, 「付録 H.1 コーディング例」を参照してください。

その他のメッセージを受信する場合



UAP で値を設定する引数

●action

受信する論理メッセージのセグメントと, 使用するバッファ形式を次の形式で指定します。

```
{DCMCFRST|DCMCFSEG}|DCMCFBUF2
```

DCMCFRST

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

DCMCFSEG

中間セグメントまたは最終セグメントを受信する場合に設定します。

DCMCFBUF2

バッファ形式 2 のバッファを示す DCMCFBUF2 を設定します。

●commform

DCNOFLAGS を設定します。

●termnam

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

処理終了後、`termnam` には OpenTP1 から値が返されます。

中間セグメントまたは最終セグメントを受信する場合は、先頭セグメントの受信時に返されたメッセージ入力元の論理端末名称を設定します。論理端末名称は最大 8 バイトの長さです。論理端末名称の最後にはヌル文字を付けてください。

中間セグメントまたは最終セグメントを受信した場合、値は返されません。

●recvdata

セグメントを受信する領域を設定します。最大セグメントより 4 バイト大きい領域を確保しておく必要があります。TP1/NET/XMAP3 の論理端末画面から入力されたメッセージを受け取る場合は、入力論理マップの先頭アドレスを設定します。

処理終了後、`recvdata` には OpenTP1 から値が返されます。

●inbufleng

セグメントを受信する領域の長さを設定します。TP1/NET/XMAP3 の論理端末画面から入力されたメッセージを受け取る場合は、入力論理マップの長さを設定します。

OpenTP1 から値が返される引数

●termnam

先頭セグメントまたは単一セグメントを受信する場合、入力元の論理端末名称が返されます。論理端末名称は最大 8 バイトの長さです。論理端末名称の最後にはヌル文字が付けられます。中間セグメントおよび最終セグメントを受信する場合、ここで返された論理端末名称を `termnam` に設定します。

●mapname

XMAP3 のドローで指定した次画面名（マップ名）が返されます。次画面名は最大 6 バイトの長さです。最後にはヌル文字が付けられます。ただし、次の場合は、ヌル文字列が返されます。

- 次画面名を指定していない場合
- アプリケーション起動機能（`dc_mcf_execap` 関数）で起動された MHP から、`dc_mcf_receive` 関数を発行する場合

●recvdata

受信したセグメントの内容が返されます。TP1/NET/XMAP3 の論理端末画面から入力されたメッセージを受信した場合、論理マップ可変部・定数部に設定されるデータが返されます。

●rdataleng

受信したセグメントの長さが返されます。TP1/NET/XMAP3 の論理端末画面から入力されたメッセージを受信した場合、論理マップ可変部・定数部の長さが返されます。

●time

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

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71000	-12000	先頭セグメントを受信する dc_mcf_receive 関数を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、action に DCMCFSEG を設定して dc_mcf_receive 関数を呼び出してください。
DCMCFRTN_71001	-12001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する dc_mcf_receive 関数を呼び出しています。直前に呼び出した dc_mcf_receive 関数でメッセージはすべて受信しました。 このリターン値が返されたあとに、再び dc_mcf_receive 関数を呼び出した場合は、リターン値 DCMCFRTN_72000 が返されます。
DCMCFRTN_71002	-12002	メッセージキューからの入力処理中に障害が発生しました。
		メッセージキューが閉塞されています。
DCMCFRTN_72000	-13000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none">先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、中間セグメントまたは最終セグメントを受信する dc_mcf_receive 関数を呼び出しています。先頭セグメントを受信する場合は、action に DCMCFRST を設定して dc_mcf_receive 関数を呼び出してください。リターン値 DCMCFRTN_71001 が返されたあとに、再び dc_mcf_receive 関数を呼び出しています。
		< SPP の実行でリターンした場合 > SPP では dc_mcf_receive 関数を呼び出せません。
DCMCFRTN_72001	-13001	termnam に設定した論理端末名称が間違っています。
DCMCFRTN_72013	-13013	inbufleng の指定値を超えるセグメントを受信しました。inbufleng の指定値を超えた部分は切り捨てられました。
DCMCFRTN_72016	-13016	action に設定した値が間違っています。
		引数に設定した値に間違いがあります。
DCMCFRTN_72024	-13024	commform に設定した値が間違っています。
DCMCFRTN_72025	-13025	action に設定したセグメント種別 (DCMCFRST または DCMCFSEG) の値が間違っています。

リターン値	リターン値 (数値)	意味
DCMCFRTN_72036	-13036	inbufleng の指定値が不足しています。5 バイト以上の領域を確保してください。
上記以外	—	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

—：該当しません。

dc_mcf_reply – 応答メッセージの送信 (C 言語)

形式

ANSI C, C++の形式

```
#include <dcmcf.h>
int dc_mcf_reply(DCLONG action, DCLONG commform, char *resv01,
                char *mapname, char *senddata, DCLONG sdataleng,
                char *nextap, DCLONG opcd)
```

K&R 版 C の形式

```
#include <dcmcf.h>
int dc_mcf_reply(action, commform, resv01, mapname, senddata,
                sdataleng, nextap, opcd)
DCLONG    action;
DCLONG    commform;
char      *resv01;
char      *mapname;
char      *senddata;
DCLONG    sdataleng;
char      *nextap;
DCLONG    opcd;
```

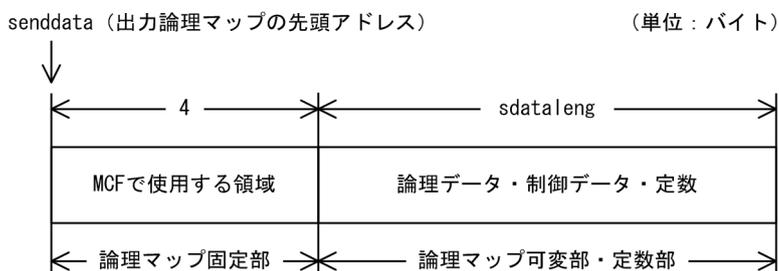
機能

メッセージを入力した論理端末に応答メッセージを送信します。応答メッセージは、一つのセグメントで構成されます。

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

継続問い合わせ応答の処理形態で、nextap の指定は、最初のセグメントだけ有効です。また、dc_mcf_contend 関数を呼び出す場合、nextap はヌル文字を指定してください。

セグメントを送信する領域の形式を次に示します。



TP1/NET/XMAP3 の論理端末に回答する場合は、XMAP3 で作成した出力論理マップ (拡張子が「.h」のファイル) を#include で取り込み、出力論理マップの先頭アドレスを senddata に指定します。出力論理マップの詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」、またはマニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発

ガイド」を参照してください。コーディング例については、「付録 H.1 コーディング例」を参照してください。

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と、使用するバッファ形式などを次の形式で指定します。

```
{DCMCFESI|DCMCFEMI} [|{DCMCFSEQ|DCMCFNSEQ}] |DCMCFBUF2 [|DCMCFMAP]
```

DCMCFESI

先頭セグメントを応答送信する場合に設定します。

DCMCFEMI

単一セグメントを応答送信する場合や、先頭セグメントの応答送信後に応答メッセージの送信の終了を連絡する場合に設定します。

DCMCFSEQ

出力通番を付ける場合に設定します。

ただし、非応答型のアプリケーションの場合、出力通番を付けません。

DCMCFNSEQ

出力通番を付けない場合に設定します。

DCMCFBUF2

バッファ形式 2 のバッファを示す DCMCFBUF2 を設定します。

DCMCFMAP

論理マップの情報だけを使用してマッピングをする場合に設定します。論理マップと物理マップの両方の情報を使用する場合は設定しません。

●commform

DCNOFLAGS を設定します。

●resv01

ヌル文字列を設定します。

●mapname

マップ名を設定します。マップ名は最大 6 バイトの長さです。マップ名の最後にはヌル文字を付けてください。

NULL またはヌル文字列を設定した場合、論理端末定義 (mcftalcle -p) の dflmap オペランドで指定したデフォルトマップ名を使用します。

●senddata

応答送信領域を設定します。TP1/NET/XMAP3 の論理端末画面に応答送信する場合は、出力論理マップの先頭アドレスを設定します。先頭セグメントの応答送信後、応答メッセージの送信の終了を連絡する場合にも必ず設定してください。

●sdata Leng

応答送信するセグメント長を設定します。TP1/NET/XMAP3 の論理端末画面に応答送信する場合は、応答送信する論理マップ可変部・定数部の長さを設定します。先頭セグメントの応答送信後、応答メッセージの送信の終了を連絡する場合には、0 を設定してください。

●nextap

<継続問い合わせ応答形態の場合>

次起動アプリケーションを設定します。次起動アプリケーションは最大 8 バイトの長さです。次起動アプリケーションの最後にはヌル文字を付けてください。

ヌル文字列を設定した場合、実行中のアプリケーションを次のメッセージ受信時に再び起動します。

dc_mcf_reply 関数を呼び出すサービスで dc_mcf_contend 関数を呼び出す場合、nextap にはヌル文字列を設定してください。

継続問い合わせ応答を引き継いだエラーイベントで、dc_mcf_reply 関数を呼び出す際に、nextap にヌル文字列を設定した場合、継続問い合わせ応答を終了します。ただし、継続問い合わせ応答を引き継いだエラーイベントで、dc_mcf_execap 関数を呼び出して継続問い合わせ応答型のアプリケーションを起動し、起動先のアプリケーションで dc_mcf_reply 関数を呼び出す際に、nextap にヌル文字列を設定した場合、継続問い合わせ応答を終了しないで、dc_mcf_execap 関数を呼び出したときに設定した継続問い合わせ応答型のアプリケーションを次のメッセージ受信時に再び起動します。

<継続問い合わせ応答形態以外の場合>

ヌル文字列を設定してください。

●opcd

DCNOFLAGS を設定します。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71002	-12002	メッセージキューへの出力処理中に障害が発生しました。
		メッセージキューが閉塞されています。
		メッセージキューが割り当てられていません。
		sdata Leng に 32000 バイトを超える値を設定しています。
		MCF が終了処理中のため、メッセージの送信を受け付けられません。

リターン値	リターン値 (数値)	意味
DCMCFRTN_71003	-12003	メッセージキューが満杯です。
DCMCFRTN_71004	-12004	メッセージを格納するバッファをメモリ上に確保できませんでした。
DCMCFRTN_71108	-12108	メッセージを送信しようとしたのですが、送信先の管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	-13000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none"> 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_reply 関数を呼び出しています。 非応答型のアプリケーションからの問い合わせ応答をしない (UAP 共通定義 (mcfmuap -c) の noansreply オペランドに no を指定) 場合に、非応答型のアプリケーションから dc_mcf_reply 関数を呼び出しています。 < SPP の実行でリターンした場合 > SPP では dc_mcf_reply 関数を呼び出せません。
DCMCFRTN_72001	-13001	入力元論理端末は、応答メッセージの送信をサポートしていません。
DCMCFRTN_72005	-13005	< action で DCMCFESI を設定した場合 > sdata Leng に 0 バイト、またはマイナス値を設定しています。
DCMCFRTN_72008	-13008	応答型のアプリケーションから dc_mcf_execap 関数を呼び出して応答型のアプリケーションを起動したあとで、dc_mcf_reply 関数を呼び出しています。 継続問い合わせ応答型のアプリケーションから dc_mcf_execap 関数を呼び出して継続問い合わせ応答型のアプリケーションを起動したあとで、dc_mcf_reply 関数を呼び出しています。
DCMCFRTN_72011	-13011	継続問い合わせ応答型でないアプリケーションが、次起動アプリケーションを設定して、dc_mcf_reply 関数を呼び出しています。
DCMCFRTN_72016	-13016	resv01 に設定した値が間違っています。 mapname に設定したマップ名が 8 文字を超えています。 nextap に設定した値が間違っています。 opcd に設定した値が間違っています。 引数に設定した値に間違いがあります。
DCMCFRTN_72026	-13026	action に設定したセグメント種別 (DCMCFESI または DCMCFEMI) の値が間違っています。
DCMCFRTN_72041	-13041	< action で DCMCFEMI を設定した場合 > <ul style="list-style-type: none"> sdata Leng に 0 バイト、またはマイナス値を設定しています。 action に DCMCFESI を設定した dc_mcf_reply 関数を呼び出さないので、メッセージの送信の終了を連絡しています。

リターン値	リターン値 (数値)	意味
DCMCFRTN_72044	-13044	dc_mcf_contend 関数を呼び出したあとで、次起動アプリケーションを設定して dc_mcf_reply 関数を呼び出しています。
DCMCFRTN_72045	-13045	nextap に、継続問い合わせ応答型でないアプリケーションのアプリケーション名を設定して dc_mcf_reply 関数を呼び出しています。
DCMCFRTN_72046	-13046	nextap に設定したアプリケーション名が異なる dc_mcf_reply 関数を、2 回以上呼び出しています。
DCMCFRTN_72047	-13047	nextap に、アプリケーション属性定義 (mcfaalcap) に定義されていないアプリケーション名を設定して dc_mcf_reply 関数を呼び出しています。
上記以外	—	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

— : 該当しません。

dc_mcf_resend – メッセージの再送 (C 言語)

形式

ANSI C, C++の形式

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

K&R 版 C の形式

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

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

機能

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

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

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

マップ名は、以前に送信した場合の名称を引き継ぎます。

UAP で値を設定する引数

●action

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

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

DCMCFSEQ

再送するメッセージに出力通番を付け直す場合に設定します。

DCMCFNSEQ

再送するメッセージに出力通番を付け直さない場合に設定します。

DCMCFNORM

一般の一方送信メッセージとして再送する場合に設定します。

DCMCFPRIO

優先の一方送信メッセージとして再送する場合に設定します。

DCMCFMAP

論理マップの情報だけで、マッピングする場合に設定します。論理マップと物理マップの両方の情報でマッピングする場合は設定しません。

DCMCFLAST

再送する対象のメッセージを検索するキーとして、最終出力通番を持つメッセージを再送する場合に設定します。この値を設定した場合は、orgseq に設定した値は無効となります。

●commform

分岐送信を示す、DCMCFOUT を設定します。

●rtermnam

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

●resv01

NULL を設定します。

●oseqid

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

DCMCFRID_NORM

一般の一方送信メッセージを対象とする場合に設定します。

DCMCFRID_PRIO

優先の一方送信メッセージを対象とする場合に設定します。

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

●orgseq

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

●otermnam

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

●resv02, resv03, resv04

NULLを設定します。

●opcd

DCNOFLAGSを設定します。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_NOMSG	-11904	出力通番使用論理端末数 (MCF マネージャ共通定義 (mcfmcomn) の-n オプション) を省略, または 0 を指定しています。 otermnam, oseqid, または orgseq に設定した値が間違っています。 再送するメッセージは次に示す理由によって, 再送できるメッセージの条件を満たしていません。 <ul style="list-style-type: none">再送対象とするメッセージの送信時に出力通番を付けていません。出力メッセージの割り当て先にメモリキューを使用しています (論理端末定義 (mcftalcle -k) の quekind オペランドを省略, または memory を指定)。論理端末が閉塞しているなどの要因によって, 送信メッセージが送信済みになっていません。送信済みのメッセージがディスクキューに保持されていません (保持メッセージ数はメッセージキューサービス定義の quegrp コマンドの-m オプションで指定します)。 otermnam で指定した論理端末に割り当てられているメッセージキューは, システム開始時に障害が発生したため, メモリキューを代用して縮退運転をしています。
DCMCFRTN_BUF_SHORT	-11905	再送するメッセージのセグメントの長さが, UAP 共通定義 (mcfmuap -e) で指定した値を超えています。

リターン値	リターン値 (数値)	意味
DCMCFRTN_71002	-12002	メッセージキューへの出力処理中に障害が発生しました。
		メッセージキューが閉塞されています。
		メッセージキューが割り当てられていません。
		MCF が終了処理中のため、メッセージの再送を受け付けられません。
DCMCFRTN_71003	-12003	メッセージキューが満杯です。
DCMCFRTN_71004	-12004	メッセージキューから取り出したメッセージを格納するバッファ（作業領域）を、メモリ上に確保できませんでした。
DCMCFRTN_71108	-12108	メッセージを再送しようとしたますが、再送先の管理テーブルが確保できませんでした。
		プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	-13000	<p>< MHP の実行でリターンした場合 ></p> <ul style="list-style-type: none"> 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_resend 関数を呼び出しています。 非トランザクション属性の MHP から、dc_mcf_resend 関数を呼び出しています。
		<p>< SPP の実行でリターンした場合 ></p> <p>トランザクションでない SPP の処理から、dc_mcf_resend 関数を呼び出しています。</p>
DCMCFRTN_72001	-13001	rtermnam または otermnam に設定した論理端末名称が間違っています。
		rtermnam に設定した論理端末名称は、定義されていません。
		該当する論理端末には、メッセージを再送できません。
DCMCFRTN_72016	-13016	action に設定した一方送信メッセージの種別（DCMCFNORM または DCMCFPRIO）の値が間違っています。
		action に設定した値が間違っています。
		oseqid に設定した値が間違っています。
		resv01, resv02, resv03, または resv04 に設定した値が間違っています。
		opcd に設定した値が間違っています。
		引数に設定した値に間違いがあります。
DCMCFRTN_72017	-13017	action に設定した出力通番の要否（DCMCFSEQ または DCMCFNSEQ）の値が間違っています。
DCMCFRTN_72024	-13024	commform に設定した値が間違っています。
上記以外	—	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

—：該当しません。

注意事項

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

-e オプション

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

-l オプション

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

dc_mcf_send – メッセージの送信 (C 言語)

形式

ANSI C, C++の形式

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

K&R 版 C の形式

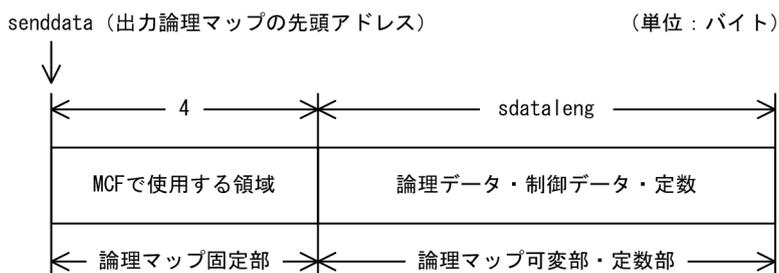
```
#include <dcmcf.h>
int dc_mcf_send(action, commform, termnam, mapname, senddata,
               sdataleng, pagecdat, opcd)
DCLONG action;
DCLONG commform;
char *termnam;
char *mapname;
char *senddata;
DCLONG sdataleng;
char *pagecdat;
DCLONG opcd;
```

機能

論理端末に一方送信メッセージを送信します。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。

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

セグメントを送信する領域の形式を次に示します。



TP1/NET/XMAP3 の論理端末に送信する場合は、XMAP3 で作成した出力論理マップ (拡張子が「.h」のファイル) を#include で取り込み、出力論理マップの先頭アドレスを senddata に指定します。出力論理マップの詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」、またはマニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド」を参照してください。コーディング例については、「付録 H.1 コーディング例」を参照してください。

UAP で値を設定する引数

●action

送信するセグメント，優先か一般か，出力通番を付けるかどうか，使用するバッファ形式，および通知イベントを通知させるかどうかを，次の形式で設定します。

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

DCMCFESI

先頭セグメントを送信する場合に設定します。

DCMCFEMI

単一セグメントを送信する場合に設定します。

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

DCMCFNORM

一般の一方送信メッセージとして送信する場合に設定します。

DCMCFPRIO

優先の一方送信メッセージとして送信する場合に設定します。

DCMCFSEQ

出力通番を付ける場合に設定します。

DCMCFNSEQ

出力通番を付けない場合に設定します。

DCMCFBUF2

バッファ形式 2 のバッファを示す DCMCFBUF2 を設定します。

DCMCFMAP

論理マップの情報だけを使用してマッピングをする場合に設定します。論理マップと物理マップの両方を使用する場合は設定しません。

DCMCFSEVT

送信完了通知イベントおよび送信障害通知イベントを通知させるときに設定します。ただし，これらのイベントを処理する MHP をアプリケーション属性定義 (mcfaalcap) で指定していない場合は無効です。メッセージが複数のセグメントで構成される場合，先頭セグメントの送信時の指定が有効になります。

●commform

分岐送信を示す，DCMCFOUT を設定します。

●termnam

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

●mapname

マップ名を設定します。マップ名は最大 6 バイトの長さです。マップ名の最後にはヌル文字を付けてください。

NULL またはヌル文字列を設定した場合、論理端末定義 (mcfstalcle -p) の dflmap オペランドで指定したデフォルトマップ名を使用します。

●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。TP1/NET/XMAP3 の論理端末に送信する場合は、出力論理マップの先頭アドレスを設定します。先頭セグメントの送信後、メッセージの送信の終了を連絡する場合にも、必ず設定してください。

●sdataleng

送信するセグメントの長さを設定します。TP1/NET/XMAP3 の論理端末画面に送信する場合は、送信する論理マップ可変部・定数部の長さを設定します。先頭セグメントの送信後、メッセージの送信の終了を連絡する場合には、0 を設定してください。

●pagecdat

NULL またはヌル文字列を設定します。

●opcd

DCNOFLAGS を設定します。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71002	-12002	メッセージキューへの出力処理中に障害が発生しました。
		メッセージキューが閉塞されています。
		メッセージキューが割り当てられていません。
		sdataleng に 32000 バイトを超える値を設定しています。
		MCF が終了処理中のため、メッセージの送信を受け付けられません。
DCMCFRTN_71003	-12003	メッセージキューが満杯です。
DCMCFRTN_71004	-12004	メッセージを格納するバッファをメモリ上に確保できませんでした。

リターン値	リターン値 (数値)	意味
DCMCFRTN_71108	-12108	メッセージを送信しようとしたのですが、送信先の管理テーブルが確保できませんでした。 プロセスのローカルメモリが不足しています。
DCMCFRTN_72000	-13000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_send 関数を呼び出しています。 < SPP の実行でリターンした場合 > トランザクションでない SPP の処理から、dc_mcf_send 関数を呼び出しています。
DCMCFRTN_72001	-13001	termnam に設定した論理端末名称が間違っています。 termnam に設定した論理端末名称は、定義されていません。 dc_mcf_send 関数を呼び出せない論理端末を設定しています。
DCMCFRTN_72005	-13005	< action で DCMCFESI を設定した場合 > sdataleng に 0 バイト、またはマイナス値を設定しています。
DCMCFRTN_72016	-13016	action に設定したメッセージ種別 (DCMCFNORM または DCMCFPRIO) の値が間違っています。 action に設定した値が間違っています。 mapname で設定したマップ名が 8 文字を超えています。 opcd に設定した値が間違っています。 引数に設定した値に間違いがあります。
DCMCFRTN_72017	-13017	action に設定した出力通番の要否 (DCMCFSEQ または DCMCFNSEQ) の値が間違っています。
DCMCFRTN_72024	-13024	commform に設定した値が間違っています。
DCMCFRTN_72026	-13026	action に設定したセグメント種別 (DCMCFESI, DCMCFEMI) の値が間違っています。
DCMCFRTN_72041	-13041	< action で DCMCFEMI を設定した場合 > <ul style="list-style-type: none"> • sdataleng に 0 バイト、またはマイナス値を設定しています。 • action に DCMCFESI を設定した dc_mcf_send 関数を呼び出さずに、メッセージの送信の終了を連絡しています。
上記以外	—	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

— : 該当しません。

dc_mcf_tactcn – コネクションの確立 (C 言語)

形式

ANSI C, C++の形式

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

K&R 版 C の形式

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

機能

コネクションを確立します。

なお、dc_mcf_tactcn 関数の正常終了は、コネクション確立要求を TP1/NET/XMAP3 が正常に受け付けたことを意味します。このため、相手システムとのコネクションの確立が正常に完了したことを示すものではありません。

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

UAP で値を設定する引数

●action

確立するコネクションの指定方法を次の形式で設定します。

```
{DCMCFLE | DCMCFCN}
```

DCMCFLE

確立するコネクションを論理端末名称で指定するときに設定します。

DCMCFCN

確立するコネクションをコネクション ID で指定するときに設定します。

●cnopt

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

構造体の形式を次に示します。

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

• mcfid

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

論理端末名称を使用して接続の確立を要求する場合は、無効となります。

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

注※

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

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

• resv01

領域をヌル文字で埋めます。

• idnam

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

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

領域をヌル文字で埋めます。

●proinf, resv02, resv03, resv04

NULL を設定します。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71001	-12001	MCF が開始処理中のため、dc_mcf_tactcn 関数が受け付けられません。
DCMCFRTN_71002	-12002	MCF が終了処理中のため、dc_mcf_tactcn 関数が受け付けられません。
DCMCFRTN_71004	-12004	dc_mcf_tactcn 関数の処理中にメモリ不足が発生しました。
DCMCFRTN_71005	-12005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71006	-12006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71007	-12007	指定されたコネクション ID は登録されていません。
DCMCFRTN_71008	-12008	指定された論理端末名称は登録されていません。
DCMCFRTN_71009	-12009	dc_mcf_tactcn 関数が、該当する MCF 通信プロセスではサポートされていません。
DCMCFRTN_71010	-12010	MCF 通信プロセスにコネクションの確立を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71011	-12011	コネクションが削除されているため、dc_mcf_tactcn 関数が受け付けられません。
DCMCFRTN_71014	-12014	TP1/NET/NCSB, または TP1/NET/X25-Extended の論理端末名称を指定しています。または TP1/NET/OSI-TP のコネクショングループ名を指定しています。
DCMCFRTN_72050	-13050	action に未サポートのフラグを設定しています。
DCMCFRTN_72051	-13051	cnopt に NULL が設定されています。
DCMCFRTN_72052	-13052	proinf に NULL が設定されていません。
DCMCFRTN_72053	-13053	resv02 に NULL が設定されていません。
DCMCFRTN_72054	-13054	resv03 に NULL が設定されていません。
DCMCFRTN_72055	-13055	resv04 に NULL が設定されていません。
DCMCFRTN_72060	-13060	action には DCMCFLE と DCMCFCN を同時に設定できません。
DCMCFRTN_72061	-13061	dcmcf_tactcnopt の mcfid に 0 未満または 240 以上の値が設定されています。
DCMCFRTN_72062	-13062	dcmcf_tactcnopt の resv01 がヌル文字で埋められていません。
DCMCFRTN_72063	-13063	dcmcf_tactcnopt の idnam の先頭がヌル文字です。
DCMCFRTN_72064	-13064	dcmcf_tactcnopt の resv02 がヌル文字で埋められていません。

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

dc_mcf_tactle – 論理端末の閉塞解除 (C 言語)

形式

ANSI C, C++の形式

```
#include <dcmcf.h>
int dc_mcf_tactle (DCLONG action, dcmcf_tactleopt *leopt,
                  char *proinf, DCLONG *resv02,
                  char *resv03, char *resv04)
```

K&R 版 C の形式

```
#include <dcmcf.h>
int dc_mcf_tactle (action, leopt, proinf, resv02, resv03, resv04)
DCLONG          action;
dcmcf_tactleopt *leopt;
char            *proinf;
DCLONG          *resv02;
char            *resv03;
char            *resv04;
```

機能

論理端末の閉塞を解除します。

なお、dc_mcf_tactle 関数の正常終了は、論理端末の閉塞解除要求を TP1/NET/XMAP3 が正常に受け付けたことを意味します。このため、論理端末の閉塞解除が正常に完了したことを示すものではありません。

dc_mcf_tactle 関数の呼び出し後に論理端末に関する何らかの処理をする場合は、dc_mcf_tlsle 関数を用いて論理端末の状態を確認してください。

UAP で値を設定する引数

●action

閉塞解除する論理端末の指定方法を次の形式で設定します。

```
DCMCFLE
```

DCMCFLE

閉塞解除する論理端末を論理端末名称で指定するときに設定します。

●leopt

この関数の対象となった論理端末の情報を、構造体 dcmcf_tactleopt に設定します。

構造体の形式を次に示します。

```
typedef struct {
    DCLONG    mcfid;           …MCF通信プロセス識別子
    char      resv01[4];       …予備領域
    char      idnam[9];        …論理端末名称
    char      resv02[7];       …予備領域
    char      resv03[112];     …予備領域
    char      resv04[376];     …予備領域
} dcmcf_tactleopt;
```

- mcfid

処理対象の論理端末を持つ MCF 通信サービスの MCF 通信プロセス識別子^{*}を設定します。設定できる範囲は 0~239 です。

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

注^{*}

MCF 環境定義 (mcftenv -s) で指定する MCF 通信プロセス識別子は 16 進数と見なしてください。
例えば、MCF 通信プロセス識別子が 10 の場合、16 を設定してください。

- resv01

領域をヌル文字で埋めます。

- idnam

閉塞解除する論理端末の名称を設定します。論理端末名称は最大 8 バイトの長さです。論理端末名称の最後にはヌル文字を付けてください。

- resv02, resv03, resv04

領域をヌル文字で埋めます。

- proinf, resv02, resv03, resv04

NULL を設定します。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71001	-12001	MCF が開始処理中のため、dc_mcf_tactle 関数が受け付けられません。
DCMCFRTN_71002	-12002	MCF が終了処理中のため、dc_mcf_tactle 関数が受け付けられません。
DCMCFRTN_71004	-12004	dc_mcf_tactle 関数の処理中にメモリ不足が発生しました。
DCMCFRTN_71005	-12005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。

リターン値	リターン値 (数値)	意味
DCMCFRTN_71006	-12006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71008	-12008	指定された論理端末名称は登録されていません。
DCMCFRTN_71009	-12009	dc_mcf_tactile 関数が、該当する MCF 通信プロセスではサポートされていません。
DCMCFRTN_71010	-12010	MCF 通信プロセスに論理端末の閉塞の解除を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71011	-12011	論理端末が削除されているため、dc_mcf_tactile 関数が受け付けられません。
DCMCFRTN_72050	-13050	action に未サポートのフラグを設定しています。 action に DCMCFLE が指定されていません。
DCMCFRTN_72051	-13051	leopt に NULL が設定されています。
DCMCFRTN_72052	-13052	proinf に NULL が設定されていません。
DCMCFRTN_72053	-13053	resv02 に NULL が設定されていません。
DCMCFRTN_72054	-13054	resv03 に NULL が設定されていません。
DCMCFRTN_72055	-13055	resv04 に NULL が設定されていません。
DCMCFRTN_72061	-13061	dcmcf_tactleopt の mcfid に 0 未満または 240 以上の値が設定されています。
DCMCFRTN_72062	-13062	dcmcf_tactleopt の resv01 がヌル文字で埋められていません。
DCMCFRTN_72063	-13063	dcmcf_tactleopt の idnam の先頭がヌル文字です。
DCMCFRTN_72064	-13064	dcmcf_tactleopt の resv02 がヌル文字で埋められていません。
DCMCFRTN_72065	-13065	dcmcf_tactleopt の resv03 がヌル文字で埋められていません。
DCMCFRTN_72067	-13067	dcmcf_tactleopt の resv04 がヌル文字で埋められていません。
DCMCFRTN_72073	-13073	dcmcf_tactleopt の idnam に設定された文字数が 9 以上です。
DCMCFRTN_72074	-13074	dcmcf_tactleopt の idnam に設定された文字列中に不正な文字があります。

dc_mcf_tdctcn – コネクションの解放 (C 言語)

形式

ANSI C, C++の形式

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

K&R 版 C の形式

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

機能

コネクションを解放します。

なお、dc_mcf_tdctcn 関数の正常終了は、コネクション解放要求を TP1/NET/XMAP3 が正常に受け付けたことを意味します。このため、相手システムとのコネクションの解放が正常に完了したことを示すものではありません。

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

UAP で値を設定する引数

●action

解放するコネクションの指定方法を次の形式で設定します。

```
{DCMCFLE | DCMFCFN} | DCMCFFRC
```

DCMCFLE

解放するコネクションを論理端末名称で指定するときに設定します。

DCMFCFN

解放するコネクションをコネクション ID で指定するときに設定します。

DCMCFFRC

コネクションを強制的に解放するときに設定します。

●cnopt

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

構造体の形式を次に示します。

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

• mcfid

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

論理端末名称を使用してコネクションの解放を要求する場合は、無効となります。

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

注※

MCF 環境定義 (mcftenv -s) で指定する MCF 通信プロセス識別子は 16 進数と見なしてください。
例えば、MCF 通信プロセス識別子が 10 の場合、16 を設定してください。

• resv01

領域をヌル文字で埋めます。

• idnam

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

• resv02, resv03, scnam, resv04, resv05

領域をヌル文字で埋めます。

●proinf, resv02, resv03, resv04

NULL を設定します。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71001	-12001	MCF が開始処理中のため、dc_mcf_tdctcn 関数が受け付けられません。
DCMCFRTN_71002	-12002	MCF が終了処理中のため、dc_mcf_tdctcn 関数が受け付けられません。
DCMCFRTN_71004	-12004	dc_mcf_tdctcn 関数の処理中にメモリ不足が発生しました。
DCMCFRTN_71005	-12005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71006	-12006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71007	-12007	指定されたコネクション ID は登録されていません。
DCMCFRTN_71008	-12008	指定された論理端末名称は登録されていません。
DCMCFRTN_71009	-12009	dc_mcf_tdctcn 関数が、該当する MCF 通信プロセスではサポートされていません。
DCMCFRTN_71010	-12010	MCF 通信プロセスにコネクションの解放を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71011	-12011	コネクションが削除されているため、dc_mcf_tdctcn 関数が受け付けられません。
DCMCFRTN_71014	-12014	TP1/NET/NCSB, または TP1/NET/X25-Extended の論理端末名称を指定しています。または TP1/NET/OSI-TP のコネクショングループ名を指定しています。
DCMCFRTN_72050	-13050	action に未サポートのフラグを設定しています。
DCMCFRTN_72051	-13051	cnopt に NULL が設定されています。
DCMCFRTN_72052	-13052	proinf に NULL が設定されていません。
DCMCFRTN_72053	-13053	resv02 に NULL が設定されていません。
DCMCFRTN_72054	-13054	resv03 に NULL が設定されていません。
DCMCFRTN_72055	-13055	resv04 に NULL が設定されていません。
DCMCFRTN_72060	-13060	action には DCMCFLE と DCMCFCN を同時に設定できません。
DCMCFRTN_72061	-13061	dcmcf_tdctcnopt の mcfid に 0 未満または 240 以上の値が設定されています。
DCMCFRTN_72062	-13062	dcmcf_tdctcnopt の resv01 がヌル文字で埋められていません。
DCMCFRTN_72063	-13063	dcmcf_tdctcnopt の idnam の先頭がヌル文字です。
DCMCFRTN_72064	-13064	dcmcf_tdctcnopt の resv02 がヌル文字で埋められていません。

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

dc_mcf_tdctle – 論理端末の閉塞 (C 言語)

形式

ANSI C, C++の形式

```
#include <dcpcf.h>
int dc_mcf_tdctle (DCLONG action, dcmcf_tdctleopt *leopt,
                  char *proinf, DCLONG *resv02,
                  char *resv03, char *resv04)
```

K&R 版 C の形式

```
#include <dcpcf.h>
int dc_mcf_tdctle (action, leopt, proinf, resv02, resv03, resv04)
DCLONG          action;
dcmcf_tdctleopt *leopt;
char            *proinf;
DCLONG          *resv02;
char            *resv03;
char            *resv04;
```

機能

論理端末を閉塞します。

なお、dc_mcf_tdctle 関数の正常終了は、論理端末の閉塞要求を TP1/NET/XMAP3 が正常に受け付けたことを意味します。このため、論理端末の閉塞が正常に完了したことを示すものではありません。

dc_mcf_tdctle 関数の呼び出し後に論理端末に関する何らかの処理をする場合は、dc_mcf_tlsle 関数を用いて論理端末の状態を確認してください。

UAP で値を設定する引数

●action

閉塞する論理端末の指定方法を次の形式で設定します。

```
DCMCFLE
```

DCMCFLE

閉塞する論理端末を論理端末名称で指定するときに設定します。

●leopt

この関数の対象となった論理端末の情報を、構造体 dcmcf_tdctleopt に設定します。

構造体の形式を次に示します。

```
typedef struct {
    DCLONG    mcfid;        …MCF通信プロセス識別子
    char      resv01[4];    …予備領域
    char      idnam[9];    …論理端末名称
    char      resv02[7];    …予備領域
    char      resv03[112];  …予備領域
    char      resv04[376];  …予備領域
} dcmcf_tdctleopt;
```

- mcfid

処理対象の論理端末を持つ MCF 通信サービスの MCF 通信プロセス識別子*を設定します。設定できる範囲は 0~239 です。

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

注※

MCF 環境定義 (mcftenv -s) で指定する MCF 通信プロセス識別子は 16 進数と見なしてください。
例えば、MCF 通信プロセス識別子が 10 の場合、16 を設定してください。

- resv01

領域をヌル文字で埋めます。

- idnam

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

- resv02, resv03, resv04

領域をヌル文字で埋めます。

- proinf, resv02, resv03, resv04

NULL を設定します。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71001	-12001	MCF が開始処理中のため、dc_mcf_tdctle 関数が受け付けられません。
DCMCFRTN_71002	-12002	MCF が終了処理中のため、dc_mcf_tdctle 関数が受け付けられません。
DCMCFRTN_71004	-12004	dc_mcf_tdctle 関数の処理中にメモリ不足が発生しました。
DCMCFRTN_71005	-12005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。

リターン値	リターン値 (数値)	意味
DCMCFRTN_71006	-12006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71008	-12008	指定された論理端末名称は登録されていません。
DCMCFRTN_71009	-12009	dc_mcf_tdctle 関数が、該当する MCF 通信プロセスではサポートされていません。
DCMCFRTN_71010	-12010	MCF 通信プロセスに論理端末の閉塞を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71011	-12011	論理端末が削除されているため、dc_mcf_tdctle 関数が受け付けられません。
DCMCFRTN_72050	-13050	action に未サポートのフラグを設定しています。
		action に DCMCFLE が指定されていません。
DCMCFRTN_72051	-13051	leopt に NULL が設定されています。
DCMCFRTN_72052	-13052	proinf に NULL が設定されていません。
DCMCFRTN_72053	-13053	resv02 に NULL が設定されていません。
DCMCFRTN_72054	-13054	resv03 に NULL が設定されていません。
DCMCFRTN_72055	-13055	resv04 に NULL が設定されていません。
DCMCFRTN_72061	-13061	dcmcf_tdctleopt の mcfid に 0 未満または 240 以上の値が設定されています。
DCMCFRTN_72062	-13062	dcmcf_tdctleopt の resv01 がヌル文字で埋められていません。
DCMCFRTN_72063	-13063	dcmcf_tdctleopt の idnam の先頭がヌル文字です。
DCMCFRTN_72064	-13064	dcmcf_tdctleopt の resv02 がヌル文字で埋められていません。
DCMCFRTN_72065	-13065	dcmcf_tdctleopt の resv03 がヌル文字で埋められていません。
DCMCFRTN_72067	-13067	dcmcf_tdctleopt の resv04 がヌル文字で埋められていません。
DCMCFRTN_72073	-13073	dcmcf_tdctleopt の idnam に設定された文字数が 9 以上です。
DCMCFRTN_72074	-13074	dcmcf_tdctleopt の idnam に設定された文字列中に不正な文字があります。

dc_mcf_tempget – 一時記憶データの受け取り (C 言語)

形式

ANSI C, C++の形式

```
#include <dcacf.h>
int dc_mcf_tempget (DCLONG action, char *getdata, DCLONG gtempleng,
                   DCLONG *gdataleng, char *resv01)
```

K&R 版 C の形式

```
#include <dcacf.h>
int dc_mcf_tempget (action, getdata, gtempleng, gdataleng, resv01)
DCLONG    action;
char      *getdata;
DCLONG    gtempleng;
DCLONG    *gdataleng;
char      *resv01;
```

機能

継続問い合わせ応答用一時記憶領域に格納されている一時記憶データを受け取ります。

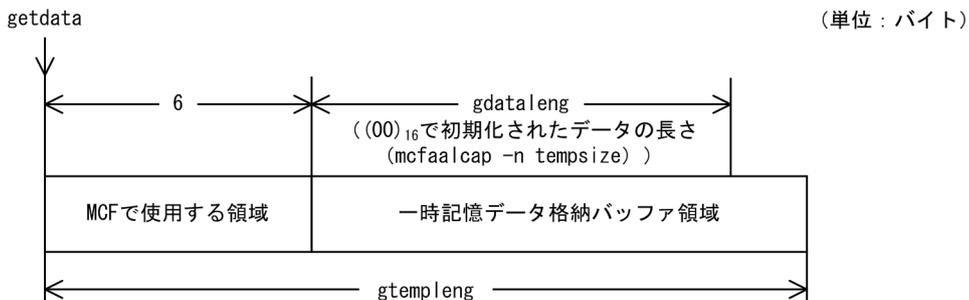
gtempleng の長さ (7~32006 バイト) を超える一時記憶データがある場合、超えた部分については切り捨てます。

dc_mcf_tempget 実行時、gtempleng から 6 を減算した値と比べて一時記憶データ長が短い場合、一時記憶データを受け取り領域に設定します。残りの受け取り領域については何も設定しません。

dc_mcf_tempget 実行時、初期状態 (継続問い合わせ応答開始後、dc_mcf_tempput を 1 回も実行していない状態) の場合、アプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドで指定した長さの $(00)_{16}$ の一時記憶データがあるものとして、dc_mcf_tempget を実行します。

受け取り領域の形式を次に示します。

- dc_mcf_tempput未実行（継続問い合わせ応答開始後、dc_mcf_tempputを実行していない（初期状態））



- dc_mcf_tempput実行済み（継続問い合わせ応答開始後、dc_mcf_tempputを1回以上実行）



UAP で値を設定する引数

●action

DCMCFBUF2 を設定します。

●getdata

受け取り領域を設定します。

処理終了後、`getdata` には OpenTP1 から値が返されます。

●gtempleng

受け取り領域長を 7～32006 バイトで設定します。

●resv01

ヌル文字列を設定します。

OpenTP1 から値が返される引数

●getdata

受け取った一時記憶データが返されます。初期状態の場合、継続問い合わせ応答用一時記憶領域の長さ（アプリケーション属性定義（`mcfaalcap -n`）の `tempsize` オペランドの指定値）分だけ `(00)16` が埋められます。

●gdata Leng

前回更新した一時記憶データの長さが返されます。初期状態の場合、継続問い合わせ応答用一時記憶領域の長さ（アプリケーション属性定義（mcfaalcap -n）の tempsize オペランドの指定値）が返されます。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_72000	-13000	SPP では dc_mcf_tempget 関数を呼び出せません。
DCMCFRTN_72013	-13013	gtempleng の設定値を超える一時記憶データを受け取りました。gtempleng の設定値を超えた部分は切り捨てられました。
DCMCFRTN_72016	-13016	action に設定した値が間違っています。
		resv01 に設定した値が間違っています。
		引数に設定した値に間違いがあります。
DCMCFRTN_72036	-13036	gtempleng の設定値が不足しています。7 バイト以上の領域を確保してください。
DCMCFRTN_72101	-13101	継続問い合わせ応答型でないアプリケーションで、dc_mcf_tempget 関数を呼び出しています。
DCMCFRTN_72106	-13106	先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_tempget 関数を呼び出しています。
DCMCFRTN_72107	-13107	dc_mcf_contend 関数を呼び出したあとで、dc_mcf_tempget 関数を呼び出しています。
上記以外	—	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

—：該当しません。

dc_mcf_tempput – 一時記憶データの更新 (C 言語)

形式

ANSI C, C++の形式

```
#include <dcmcf.h>
int dc_mcf_tempput(DCLONG action, char *putdata, DCLONG pdataleng,
                  char *resv01)
```

K&R 版 C の形式

```
#include <dcmcf.h>
int dc_mcf_tempput(action, putdata, pdataleng, resv01)
DCLONG    action;
char      *putdata;
DCLONG    pdataleng;
char      *resv01;
```

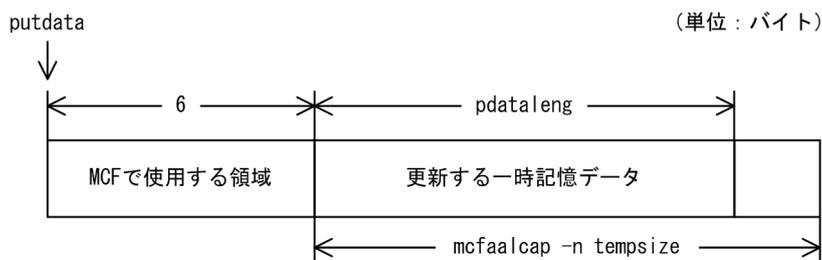
機能

継続問い合わせ応答用一時記憶領域に格納されている一時記憶データを更新します。

更新要求をする前に、必ず受け取り要求をしてください。

アプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドには、更新する一時記憶データ長 (pdataleng) 以上の値を指定してください。

更新する領域の形式を次に示します。



UAP で値を設定する引数

●action

DCMCFBUF2 を設定します。

●putdata

一時記憶データの更新データが格納されている領域を設定します。

●pdataleng

一時記憶データの更新データ長を設定します。

●resv01

ヌル文字列を設定します。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71103	-12103	一時記憶データを更新するための領域をメモリ上に確保できませんでした。
DCMCFRTN_72000	-13000	SPP では dc_mcf_tempput 関数を呼び出せません。
DCMCFRTN_72016	-13016	action に設定した値が間違っています。
		resv01 に設定した値が間違っています。
		引数に設定した値に間違いがあります。
DCMCFRTN_72035	-13035	pdataleng に設定した更新データの長さが、アプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドで定義した長さを超えています。
		pdataleng に 0 バイト、またはマイナス値を設定しています。
DCMCFRTN_72101	-13101	継続問い合わせ応答型でないアプリケーションで、dc_mcf_tempput 関数を呼び出しています。
DCMCFRTN_72105	-13105	dc_mcf_tempget 関数を呼び出す前に、dc_mcf_tempput 関数を呼び出しています。
DCMCFRTN_72106	-13106	先頭セグメントを受信する dc_mcf_receive 関数を呼び出す前に、dc_mcf_tempput 関数を呼び出しています。
DCMCFRTN_72107	-13107	dc_mcf_contend 関数を呼び出したあとで、dc_mcf_tempput 関数を呼び出しています。
上記以外	—	プログラムの破壊などによる、予期しないエラーが発生しました。

(凡例)

— : 該当しません。

dc_mcf_tlscn – コネクションの状態取得 (C 言語)

形式

ANSI C, C++の形式

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

K&R 版 C の形式

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

機能

コネクションの状態を取得します。

UAP で値を設定する引数

●action

状態を取得するコネクションの指定方法を次の形式で設定します。

```
{DCMCFLE | DCMCFCN}
```

DCMCFLE

状態を取得するコネクションを論理端末名称で指定するときに設定します。

DCMCFCN

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

●cnopt

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

構造体の形式を次に示します。

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

- mcfid

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

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

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

注※

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

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

- resv01

領域をヌル文字で埋めます。

- idnam

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

- resv02, resv03, resv04

領域をヌル文字で埋めます。

- resv01, resv02, resv03

NULL を設定します。

- infcnt

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

処理終了後は、該当するコネクションの個数が返されます。

- inf

コネクション状態を格納する領域 dcmcf_cninf を設定します。

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

●resv04

NULL を設定します。

OpenTP1 から値が返される引数

●infcnt

この関数の対象となった接続の個数が返されます。

●inf

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

構造体の形式を次に示します。

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

- idnam

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

- resv01

領域をヌル文字で埋めます。

- pnam

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

XP△

TP1/NET/XMAP3 プロトコル

- status

要求した接続の状態として、次の値が設定されます。

DCMCF_CNST_ACT

確立状態

DCMCF_CNST_ACT_B

確立処理中状態

DCMCF_CNST_DCT

解放状態

DCMCF_CNST_DCT_B

解放処理中状態

- resv02

領域をヌル文字で埋めます。

リターン値

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

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

dc_mcf_tlsle – 論理端末の状態取得 (C 言語)

形式

ANSI C, C++の形式

```
#include <dcpcf.h>
int dc_mcf_tlsle (DCLONG action, dcmcf_tlsleopt *leopt,
                 char *resv01, DCLONG *resv02,
                 char *resv03, DCLONG *infcnt,
                 dcmcf_leinf2 *inf, char *resv04)
```

K&R 版 C の形式

```
#include <dcpcf.h>
int dc_mcf_tlsle (action, leopt, resv01, resv02, resv03, infcnt,
                 inf, resv04)
DCLONG          action;
dcmcf_tlsleopt  *leopt;
char            *resv01;
DCLONG          *resv02;
char            *resv03;
DCLONG          *infcnt;
dcmcf_leinf2    *inf;
char            *resv04;
```

機能

論理端末の状態を取得します。

UAP で値を設定する引数

●action

状態を取得する論理端末の指定方法を次の形式で設定します。

```
DCMCFLE
```

DCMCFLE

状態を取得する論理端末を論理端末名称で指定するときに設定します。

●leopt

この関数の対象となった論理端末の情報を、構造体 dcmcf_tlsleopt に設定します。

構造体の形式を次に示します。

```
typedef struct {
    DCLONG    mcfid;           …MCF通信プロセス識別子
    char      resv01[4];      …予備領域
```

```

char    idnam[9];      …論理端末名称
char    resv02[7];    …予備領域
char    resv03[112];  …予備領域
char    resv04[376];  …予備領域
} dcmcf_tlsleopt;

```

- mcfid

処理対象の論理端末を持つ MCF 通信サービスの MCF 通信プロセス識別子*を設定します。設定できる範囲は 0~239 です。

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

注※

MCF 環境定義 (mcftenv -s) で指定する MCF 通信プロセス識別子は 16 進数と見なしてください。例えば、MCF 通信プロセス識別子が 10 の場合、16 を設定してください。

- resv01

領域をヌル文字で埋めます。

- idnam

状態を取得する論理端末の名称を設定します。論理端末名称は最大 8 バイトの長さです。論理端末名称の最後にはヌル文字を付けてください。

- resv02, resv03, resv04

領域をヌル文字で埋めます。

- resv01, resv02, resv03

NULL を設定します。

- infcnt

論理端末の状態を格納する領域 dcmcf_leinf2 の個数として、1 を設定します。

処理終了後は、該当する論理端末の個数が返されます。

- inf

論理端末の状態を格納する領域 dcmcf_leinf2 を設定します。

「構造体 dcmcf_leinf2 のサイズ×infcnt」バイト数分の領域が必要です。

- resv04

NULL を設定します。

OpenTP1 から値が返される引数

●infcnt

この関数の対象となった論理端末の個数が返されます。

●inf

この関数の対象となった論理端末の情報が構造体 dcmcf_leinf2 で返されます。

構造体の形式を次に示します。

```
typedef struct {
    char    idnam[9];      …論理端末名称
    char    resv01[7];    …予備領域
    char    resv02[4];    …予備領域
    DCLONG  status;       …論理端末状態
    char    resv03[40];   …予備領域
} dcmcf_leinf2;
```

- idnam

要求した論理端末の名称が設定されます。論理端末名称は最大 8 バイトの長さです。論理端末名称の最後にはヌル文字が付けられます。

- resv01, resv02

領域をヌル文字で埋めます。

- status

要求した論理端末の状態として、次の値が設定されます。

DCMCF_LEST_ACT

閉塞解除状態

DCMCF_LEST_DCT

閉塞状態

- resv03

領域をヌル文字で埋めます。

リターン値

リターン値	リターン値 (数値)	意味
DCMCFRTN_00000	0	正常に終了しました。
DCMCFRTN_71001	-12001	MCF が開始処理中のため、dc_mcf_tlsle 関数が受け付けられません。
DCMCFRTN_71004	-12004	dc_mcf_tlsle 関数の処理中にメモリ不足が発生しました。
DCMCFRTN_71005	-12005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。

リターン値	リターン値 (数値)	意味
DCMCFRTN_71006	-12006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71008	-12008	指定された論理端末名称は登録されていません。
DCMCFRTN_71009	-12009	dc_mcf_tlsle 関数が、該当する MCF 通信プロセスではサポートされていません。
DCMCFRTN_71010	-12010	MCF 通信プロセスに論理端末の状態取得を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
DCMCFRTN_71011	-12011	論理端末が削除されているため、dc_mcf_tlsle 関数が受け付けられません。
DCMCFRTN_72050	-13050	action に未サポートのフラグを設定しています。 action に DCMCFLE が指定されていません。
DCMCFRTN_72051	-13051	leopt に NULL が設定されています。
DCMCFRTN_72052	-13052	resv01 に NULL が設定されていません。
DCMCFRTN_72053	-13053	resv02 に NULL が設定されていません。
DCMCFRTN_72054	-13054	resv03 に NULL が設定されていません。
DCMCFRTN_72055	-13055	resv04 に NULL が設定されていません。
DCMCFRTN_72056	-13056	infcnt に NULL が設定されています。
DCMCFRTN_72057	-13057	inf に NULL が設定されています。
DCMCFRTN_72061	-13061	dcmcf_tlsleopt の mcfid に 0 未満または 240 以上の値が設定されています。
DCMCFRTN_72062	-13062	dcmcf_tlsleopt の resv01 がヌル文字で埋められていません。
DCMCFRTN_72063	-13063	dcmcf_tlsleopt の idnam の先頭がヌル文字です。
DCMCFRTN_72064	-13064	dcmcf_tlsleopt の resv02 がヌル文字で埋められていません。
DCMCFRTN_72065	-13065	dcmcf_tlsleopt の resv03 がヌル文字で埋められていません。
DCMCFRTN_72067	-13067	dcmcf_tlsleopt の resv04 がヌル文字で埋められていません。
DCMCFRTN_72073	-13073	dcmcf_tlsleopt の idnam に設定された文字数が 9 以上です。
DCMCFRTN_72074	-13074	dcmcf_tlsleopt の idnam に設定された文字列中に不正な文字があります。
DCMCFRTN_72076	-13076	infcnt に 1 が設定されていません。

4

COBOL-UAP 作成用プログラムインタフェース

この章では、TP1/NET/XMAP3 で使用できる、COBOL-UAP 作成用プログラムインタフェースについて説明します。

COBOL-UAP 作成用プログラムインタフェースの一覧

TP1/NET/XMAP3 で使用する COBOL-UAP 作成用プログラムインタフェースについて、COBOL 言語、およびデータ操作言語に分けて説明します。

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

COBOL 言語のプログラムインタフェース

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

COBOL 言語のプログラムインタフェースの一覧を、次の表に示します。

表 4-1 COBOL 言語のプログラムインタフェースの一覧

プログラム名	データ名	機能
CBLDCMCF	'CONTEND△'	継続問い合わせ応答の終了
	'RECEIVE△'	メッセージの受信
	'REPLY△△△'	応答メッセージの送信
	'RESEND△△'	メッセージの再送
	'SEND△△△△'	メッセージの送信
	'TACTCN△△'	コネクションの確立
	'TACTLE△△'	論理端末の閉塞解除
	'TDCTCN△△'	コネクションの解放
	'TDCTLE△△'	論理端末の閉塞
	'TEMPGET△'	一時記憶データの受け取り
	'TEMPPUT△'	一時記憶データの更新
	'TLSCN△△△'	コネクションの状態取得
	'TLSLE△△△'	論理端末の状態取得

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

データ操作言語のプログラムインタフェース

データ操作言語（DML）を使用した、通信文について説明します。データ操作言語の形式の詳細については、マニュアル「OpenTP1 プログラム作成リファレンス COBOL 言語編」を参照してください。

データ操作言語のプログラムインタフェースの一覧を、次の表に示します。

表 4-2 データ操作言語のプログラムインタフェースの一覧

通信文		機能	対応する C 言語のライブラリ関数
データコミュニケーション機能	RECEIVE	メッセージの受信	「表 4-4 RECEIVE 文（データコミュニケーション機能）の指定と C 言語のライブラリ関数との対応」、および「表 4-5 SEND 文（データコミュニケーション機能）の指定と C 言語のライブラリ関数との対応」を参照してください。
	SEND	メッセージの送信	
		応答メッセージの送信	
サービス機能	DISABLE	継続問い合わせ応答の終了	dc_mcf_contend
	RECEIVE	一時記憶データの受け取り	dc_mcf_tempget
	SEND	一時記憶データの更新	dc_mcf_tempput

注

dc_mcf_resend（メッセージの再送）に対応するデータ操作言語のインタフェースはありません。

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

通信記述項について

TP1/NET/XMAP3 のメッセージ送受信の通信文で、通信記述項に指定できる句の指定要否を、次の表に示します。

表 4-3 通信記述項に指定できる句の指定要否

データ名を指定する句	データ領域の値の設定元								
	1.	2.	3.	4.	5.	6.	7.	8.	9.
STATUS KEY	B	B	B	B	B	B	B	B	B
SYMBOLIC TERMINAL	B	U ₁ ※	U ₁	U ₁	—	—	×	×	×
MESSAGE DATE	B	B	—	—	—	—	—	—	—
MESSAGE TIME	B	B	—	—	—	—	—	—	—
MAP NAME	B	B	U ₂	U ₂	U ₂	U ₂	—	—	—
SYNCHRONOUS MODE	U ₂	U ₂	U ₂	U ₂	U ₂	U ₂	—	—	—
SWITCHING MODE	—	—	U ₂	U ₂	—	—	—	—	—
NEXT TRANSACTION	—	—	—	—	U ₂	U ₂	—	—	—
ACTIVE INTERVAL/TIME	—	—	—	—	—	—	—	—	—
DETAIL MODE	—	—	U ₂	U ₂	U ₂	U ₂	—	—	—
WAITING TIME	—	—	—	—	—	—	—	—	—

(凡例)

1. : 先頭セグメントの非同期受信 (RECEIVE)
 2. : 中間, 最終セグメントの非同期受信 (RECEIVE)
 3. : 一方送信メッセージの先頭セグメントの非同期送信 (SEND)
 4. : 一方送信メッセージの単一セグメントの非同期送信 (SEND)
 5. : 応答メッセージの先頭セグメントの非同期送信 (SEND)
 6. : 応答メッセージの単一セグメントの非同期送信 (SEND)
 7. : 継続問い合わせ応答の終了 (DISABLE)
 8. : 一時記憶データの受け取り (RECEIVE)
 9. : 一時記憶データの更新 (SEND)
- B : OpenTP1 から値が返されます。省略できます。
- U1 : UAP で値を設定します。省略できません。
- U2 : UAP で値を設定します。省略できます。
- : 該当しません。設定しても無効です。
- × : 空白以外の値を設定するとエラーリターンします。

注※

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

コンパイラオプション -VOSCBL,DataComm を指定した場合の通信記述項

次のどちらかを行った場合, 通信記述項の初期値は, メインフレーム (VOS3/VOS1) の XDM/DCCM データコミュニケーション機能を使用するときの値となります。

- 通信記述項の句の指定を省略している
- 通信文の実行前に, 通信記述項の句に指定されたデータ名に初期値を設定していない

コンパイラオプションについては, マニュアル「COBOL2002 使用の手引 手引編」を参照してください。

データコミュニケーション機能と C 言語のライブラリ関数の対応

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

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

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

(凡例)

- : 指定あり
- : 指定なし

注※

TP1/NET/XMAP3 ではサポートしていない関数です。

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

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

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

（凡例）

○：指定あり

—：指定なし

注※

TP1/NET/XMAP3 ではサポートしていない関数です。

CBLDCMCF('CONTEND') – 継続問い合わせ応答の終了 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

```
CALL 'CBLDCMCF' USING 一意名1
```

DATA DIVISION の指定

```
01 一意名1.  
  02 データ名A PIC X(8) VALUE 'CONTEND'.  
  02 データ名B PIC X(5).  
  02 FILLER PIC X(3).  
  02 データ名C PIC X(16) VALUE LOW-VALUE.
```

機能

継続問い合わせ応答を終了します。継続問い合わせ応答を終了するときは、その MHP から呼び出した CBLDCMCF('REPLY△△△')のデータ名 R が空白であること、および継続問い合わせ応答型のアプリケーションを起動する CBLDCMCF('EXECAP△△')を呼び出していないことが前提です。次起動アプリケーションを CBLDCMCF('REPLY△△△')のデータ名 R に設定したり、継続問い合わせ応答型のアプリケーションを起動する CBLDCMCF('EXECAP△△')を呼び出した場合は、CBLDCMCF('CONTEND△')はエラーリターンします。

CBLDCMCF('CONTEND△')を呼び出したあとは、一時記憶データにアクセスするプログラム (CBLDCMCF('TEMPGET△'), CBLDCMCF('TEMPPUT△')) は使用できません。

UAP で値を設定するデータ領域

●データ名 A

継続問い合わせ応答の終了を示す要求コード「VALUE 'CONTEND△」を設定します。

●データ名 C

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。

ステータスコード	意味
72000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する CBLDCMCF('RECEIVE△')を呼び出す前に、 CBLDCMCF('CONTEND△')を呼び出しています。
	< SPP の実行でリターンした場合 > SPP では CBLDCMCF('CONTEND△')を呼び出せません。
72016	データ名 C に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72101	継続問い合わせ応答型でないアプリケーションで、CBLDCMCF('CONTEND△')を呼び出しています。
72107	CBLDCMCF('CONTEND△')を 2 回以上呼び出しています。
72111	次起動アプリケーションを設定して CBLDCMCF('REPLY△△△')を呼び出したあと、 CBLDCMCF('CONTEND△')を呼び出しています。
	継続問い合わせ応答型のアプリケーション名を設定して CBLDCMCF('EXECAP△△')を呼び出したあと、 CBLDCMCF('CONTEND△')を呼び出しています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

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

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

機能

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

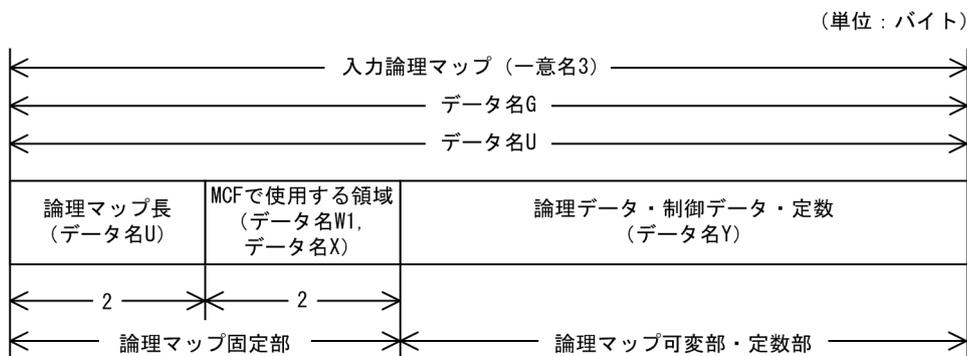
受信できるメッセージの一つのセグメントの最大長は、32763 バイトまでです。

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

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

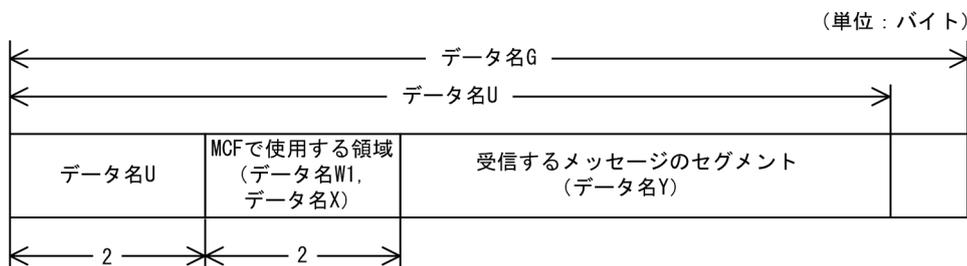
セグメントを受信する領域（一意名 3 で示す領域）の形式を次に示します。

TP1/NET/XMAP3 の論理端末画面から入力されたメッセージを受信する場合



TP1/NET/XMAP3 の論理端末画面から入力されたメッセージを受信する場合、XMAP3 で作成した入力論理マップを COPY 文で取り込み、一意名 3 として指定します。入力論理マップの詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」、またはマニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド」を参照してください。コーディング例については、「付録 H.1 コーディング例」を参照してください。

その他のメッセージを受信する場合



UAP で値を設定するデータ領域

●データ名 A

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

●データ名 C

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

VALUE 'FRST'

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

VALUE 'SEG△'

中間セグメントまたは最終セグメントを受信する場合に設定します。

●データ名 D

空白を設定します。

●データ名 G

セグメントを受信する領域の長さを設定します。TP1/NET/XMAP3 の論理端末画面から入力されたメッセージを受信する場合、入力論理マップの長さを設定します。

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

空白を設定します。

●データ名 M4, データ名 M5

0 を設定します。

●データ名 M6

空白を設定します。

●データ名 M7

使用する送受信バッファの形式を示す「VALUE '2」を設定します。

●データ名 N

MCF で使用する領域です。

●データ名 O

空白を設定します。

●データ名 P

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

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

●データ名 R

空白を設定します。

●データ名 T, データ名 W1, データ名 X

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

●データ名 E

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

●データ名 F

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

●データ名 P

先頭セグメントまたは単一セグメントを受信する場合、入力元の論理端末名称が返されます。論理端末名
称は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称の後ろが空白で埋められます。

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

●データ名 Q

XMAP3 のドローで指定した次画面名 (マップ名) が返されます。次画面名は最大 6 バイトの長さです。
次画面名の後ろは空白で埋められます。

ただし、次の場合は、空白が返されます。

- 次画面名を指定していない場合
- アプリケーション起動機能 (dc_mcf_execap 関数) で起動された MHP から、
CBLDCMCF('RECEIVE△')を発行する場合

●データ名 U

受信したセグメントの長さ + 4 が返されます。TP1/NET/XMAP3 の論理端末画面から入力されたメッ
セージを受信した場合、受信した入力論理マップの長さ (論理マップ固定部の長さ + 論理マップ可変部・
定数部の長さ) が返されます。

●データ名 Y

受信したセグメントの内容が返されます。TP1/NET/XMAP3 の論理端末画面から入力されたメッセー
ジを受信した場合、論理マップ可変部・定数部に設定されるデータが返されます。一つのセグメントで 32763
バイトまで受信できます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する CBLDCMCF('RECEIVE△')を 2 回以上呼び出しています。中間セグメントまたは最終セグメントを受信する場合は、データ名 C に「VALUE 'SEG△」を設定して CBLDCMCF('RECEIVE△')を呼び出してください。
71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する CBLDCMCF('RECEIVE△')を呼び出しています。直前に呼び出した CBLDCMCF('RECEIVE△')でメッセージはすべて受信しました。このステータスコードが返されたあとに、再び CBLDCMCF('RECEIVE△')を呼び出した場合は、ステータスコード 72000 が返されます。
71002	メッセージキューからの入力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
71108	プロセスのローカルメモリが不足しています。
72000	<p>< MHP の実行でリターンした場合 ></p> <ul style="list-style-type: none"> 先頭セグメントを受信する CBLDCMCF('RECEIVE△')を呼び出す前に、中間セグメントまたは最終セグメントを受信する CBLDCMCF('RECEIVE△')を呼び出しています。先頭セグメントを受信する場合は、データ名 C に「VALUE 'FRST」を設定して CBLDCMCF('RECEIVE△')を呼び出してください。 ステータスコード 71001 が返されたあとに、再び CBLDCMCF('RECEIVE△')を呼び出しています。
	<p>< SPP の実行でリターンした場合 ></p> <p>SPP では CBLDCMCF('RECEIVE△')を呼び出せません。</p>
72001	データ名 P に設定した論理端末名称が間違っています。
72013	データ名 G に設定した値を超えるセグメントを受信しました。データ名 G に設定した値を超えた部分は切り捨てられました。
	32763 バイトを超えるセグメントを受信しました。32763 バイトを超えた部分は切り捨てられました。
72016	データ名 D に設定した値が間違っています。
	データ名 N またはデータ名 T に設定した値が間違っています。
	データ名 M7 に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72025	データ名 C に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72036	データ名 G に設定した値が不足しています。5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

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

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
  02 データ名A PIC X(8) VALUE 'REPLY'.  
  02 データ名B PIC X(5).  
  02 FILLER PIC X(3).  
  02 データ名C PIC X(4) VALUE SPACE.  
  02 データ名D PIC X(4) VALUE SPACE.  
  02 データ名E PIC 9(8).  
  02 データ名F PIC 9(8).  
  02 データ名G PIC 9(9) COMP VALUE ZERO.  
  02 データ名H PIC X(4).  
  02 データ名I PIC X(4) VALUE SPACE.  
  02 データ名J PIC X(4) VALUE SPACE.  
  02 データ名K PIC X(4) VALUE SPACE.  
  02 データ名L PIC X(8) VALUE SPACE.  
  02 データ名M1 PIC X(4) VALUE SPACE.  
  02 データ名M2 PIC X(8) VALUE SPACE.  
  02 データ名M3 PIC X(4) VALUE SPACE.  
  02 データ名M4 PIC 9(9) COMP VALUE ZERO.  
  02 データ名M5 PIC 9(9) COMP VALUE ZERO.  
  02 データ名M6 PIC X(1).  
  02 データ名M7 PIC X(1) VALUE '2'.  
  02 データ名N PIC X(14) VALUE LOW-VALUE.  
01 一意名2.  
  02 データ名O PIC X(4) VALUE SPACE.  
  02 データ名P PIC X(8) VALUE SPACE.  
  02 データ名Q PIC X(8).  
  02 データ名R PIC X(8).  
  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(4) COMP.  
  02 データ名V PIC X(2) VALUE LOW-VALUE.  
  02 データ名W PIC X(n).
```

機能

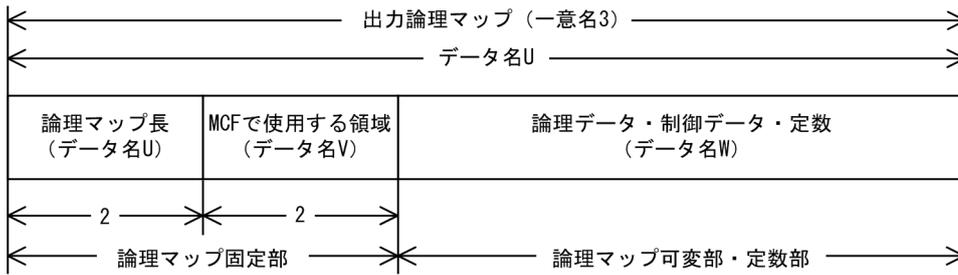
メッセージを入力した論理端末に対して応答メッセージを送信します。応答メッセージは、一つのセグメントで構成されます。

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

継続問い合わせ応答の処理形態で、データ名 R の指定は、最初のセグメントだけ有効です。また、継続問い合わせ応答の終了要求をする場合、データ名 R を空白で埋めてください。

セグメントを送信する領域（一意名 3 で示す領域）の形式を次に示します。

(単位：バイト)



TP1/NET/XMAP3 の論理端末に応答する場合は、XMAP3 で作成した出力論理マップを COPY 文で取り込み、一意名 3 として指定します。出力論理マップの詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」、またはマニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド」を参照してください。コーディング例については、「付録 H.1 コーディング例」を参照してください。

UAP で値を設定するデータ領域

●データ名 A

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

●データ名 C, データ名 D

空白を設定します。

●データ名 E, データ名 F

MCF で使用する領域です。

●データ名 G

0 を設定します。

●データ名 H

送信する応答メッセージのセグメント種別を設定します。

VALUE 'ESI△'

先頭セグメントを応答送信する場合に設定します。

VALUE 'EMI△'

次のどれかの場合に設定します。

- 論理メッセージが単一セグメントの場合
- 先頭セグメントの応答送信後に応答メッセージの送信の終了を連絡する場合

●データ名 I, データ名 J

空白を設定します。

●データ名 K

出力通番を付けるかどうかを設定します。

VALUE 'SEQ△'

出力通番を付ける場合に設定します。

ただし、非応答型のアプリケーションの場合、出力通番を付けません。

VALUE 'NSEQ'

出力通番を付けない場合に設定します。

空白

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

●データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

●データ名 M4, データ名 M5

0 を設定します。

●データ名 M6

マッピングモードを設定します。

VALUE '0'

論理マップと物理マップをマージしてマッピングする場合に設定します。

VALUE '1'

論理マップだけでマッピングする場合に設定します。

空白

省略されたものとして、「VALUE '0'」（論理マップと物理マップをマージしてマッピングする）が設定されます。

●データ名 M7

使用する送受信バッファの形式を「VALUE '2'」と設定します。

●データ名 N

MCF で使用する領域です。

●データ名 O, データ名 P

空白を設定します。

●データ名 Q

マップ名を設定します。マップ名は最大 6 バイトの長さです。マップ名の後ろを空白で埋めて、8 文字になるように設定してください。

空白を設定した場合、論理端末定義 (mcftalcle -p) の dflmap オペランドで指定したデフォルトマップ名を使用します。

●データ名 R

<継続問い合わせ応答形態の場合>

次起動アプリケーションを設定します。次起動アプリケーションは最大 8 バイトの長さです。8 バイトに満たない場合、次起動アプリケーションの後ろを空白で埋めてください。

空白を設定した場合、実行中のアプリケーションを次のメッセージ受信時に再び起動します。

CBLDCMCF('REPLY△△△')を呼び出すサービスで CBLDCMCF('CONTEND△')を呼び出す場合はデータ名 R に空白を設定してください。

継続問い合わせ応答を引き継いだエラーイベントで CBLDCMCF('REPLY△△△')を呼び出す際に、データ名 R に空白を設定した場合、継続問い合わせ応答を終了します。ただし、継続問い合わせ応答を引き継いだエラーイベントで、CBLDCMCF('EXECAP△△')を呼び出して継続問い合わせ応答型のアプリケーションを起動し、起動先のアプリケーションで CBLDCMCF('REPLY△△△')を呼び出す際に、データ名 R に空白を設定した場合、継続問い合わせ応答を終了しないで、CBLDCMCF('EXECAP△△')を呼び出したときに設定した継続問い合わせ応答型のアプリケーションを次のメッセージ受信時に再び起動します。

<継続問い合わせ応答形態以外の場合>

空白を設定します。

データ名 R の設定は、最初のセグメントだけ有効です。継続問い合わせ応答の処理形態で、終了要求をする場合は空白で埋めてください。

●データ名 T

MCF で使用する領域です。

●データ名 U

応答送信する出力論理マップの長さ（論理マップ固定部の長さ+論理マップ可変部・定数部の長さ）を設定します。先頭セグメントの応答送信後、応答メッセージの送信の終了を連絡する場合には、4 を設定してください。

●データ名 V

MCF で使用する領域です。

●データ名 W

送信する応答メッセージのセグメントの内容（論理マップ可変部・定数部に出力する内容）を設定します。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	データ名 U に 32004 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたますが、送信先の管理テーブルを確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none">先頭セグメントを受信する CBLDCMCF('RECEIVE△')を呼び出す前に、CBLDCMCF('REPLY△△△')を呼び出しています。非応答型のアプリケーションからの問い合わせ応答をしない (UAP 共通定義 (mcfmuap-c) の noansreply オペランドに no を指定) 場合に、非応答型のアプリケーションから CBLDCMCF('REPLY△△△')を呼び出しています。
	< SPP の実行でリターンした場合 > SPP では CBLDCMCF('REPLY△△△')を呼び出せません。
72001	入力元論理端末は、応答メッセージの送信をサポートしていません。
72005	<データ名 H で VALUE 'ESI△'を設定した場合 > データ名 U に 0 から 4 バイト、またはマイナス値を設定しています。
72008	応答型のアプリケーションから、CBLDCMCF('EXECAP△△')を呼び出して応答型のアプリケーションを起動したあとで、CBLDCMCF('REPLY△△△')を呼び出しています。
	継続問い合わせ応答型のアプリケーションから、CBLDCMCF('EXECAP△△')で継続問い合わせ応答型のアプリケーションを起動したあとで、CBLDCMCF('REPLY△△△')を呼び出しています。
72011	継続問い合わせ応答型でないアプリケーションが、次起動アプリケーションを設定して、CBLDCMCF('REPLY△△△')を呼び出しています。
72016	データ名 N、データ名 T に設定した値が間違っています。
	データ名 M7 に設定した値が間違っています。

ステータスコード	意味
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
72020	データ名 I に設定した値が間違っています。
72026	データ名 H に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72041	<p><データ名 H で VALUE 'EMI△'を設定した場合></p> <ul style="list-style-type: none"> データ名 U に 0 から 4 バイト, またはマイナス値を設定しています。 データ名 H に VALUE 'ESI△'を設定した CBLDCMCF('REPLY△△△')を呼び出さないで, メッセージの送信の終了を連絡しています。
72044	CBLDCMCF('CONTEND△')を呼び出したあとで, 次起動アプリケーションを設定して CBLDCMCF('REPLY△△△')を呼び出しています。
72045	データ名 R に, 継続問い合わせ応答型でないアプリケーションのアプリケーション名を設定して CBLDCMCF('REPLY△△△')を呼び出しています。
72046	データ名 R に設定したアプリケーション名が異なる CBLDCMCF('REPLY△△△')を, 2 回以上呼び出しています。
72047	データ名 R に, アプリケーション属性定義 (mcfaalcap) に定義されていないアプリケーション名を設定して CBLDCMCF('REPLY△△△')を呼び出しています。
上記以外	プログラムの破壊などによる, 予期しないエラーが発生しました。

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

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

機能

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

- 出力先の論理端末名称

- メッセージ出力通番
- メッセージ種別（一般の一方送信，優先の一方送信）

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

マップ名は，以前に送信した場合の名称を引き継ぎます。

UAP で値を設定するデータ領域

●データ名 A

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

●データ名 C, データ名 D

空白を設定します。

●データ名 E, データ名 F

MCF で使用する領域です。

●データ名 G

0 を設定します。

●データ名 H, データ名 I

空白を設定します。

●データ名 J

一般として再送するか優先として再送するかを設定します。

VALUE 'NORM'

一般の一方送信メッセージとして再送する場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージとして再送する場合に設定します。

空白

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

●データ名 K

再送するメッセージに出力通番を付け直すかどうかを設定します。

VALUE 'SEQ△'

再送するメッセージに出力通番を付け直す場合に設定します。

VALUE 'NSEQ'

再送するメッセージに出力通番を付け直さない場合に設定します。

空白

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

●データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

●データ名 M4, データ名 M5

0 を設定します。

●データ名 M6

マッピングモードの作成方法を設定します。

VALUE '0'

論理マップと物理マップの情報でマッピングする場合に設定します。

VALUE '1'

論理マップの情報だけでマッピングする場合に設定します。

空白

省略されたものとして、「VALUE '0'」（論理マップと物理マップでマッピング）が設定されます。

●データ名 M7

空白を設定します。

●データ名 N

MCF で使用する領域です。

●データ名 O

分岐送信を示す「VALUE 'OUT△'」を設定します。

●データ名 P

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

●データ名 Q, データ名 R

空白を設定します。

●データ名 S

MCF で使用する領域です。

●データ名 T

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

●データ名 U

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

VALUE 'NORM'

一般の一方送信メッセージを対象とする場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージを対象とする場合に設定します。

空白

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

●データ名 V

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

●データ名 W

再送する対象のメッセージを検索するキーとして、最終出力通番を持つメッセージを再送するかどうかを設定します。

VALUE 'LAST'

送信済みのメッセージのうち、最終出力通番を持つメッセージを再送する場合に設定します。この値を設定した場合は、データ名 V に設定した値は無効となります。

空白

省略されたものとして、最終出力通番を持つメッセージを再送対象としないことが設定されます。

●データ名 X

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。

ステータスコード	意味
70904	出力通番使用論理端末数 (MCF マネージャ共通定義 (mcfmcomn) の-n オプション) を省略, または 0 を指定しています。
	データ名 T, データ名 U, またはデータ名 V に設定した値が間違っています。
	再送するメッセージは次に示す理由により, 再送できるメッセージの条件を満たしていません。 <ul style="list-style-type: none"> 再送対象とするメッセージの送信時に出力通番を付けていません。 出力メッセージの割り当て先にメモリキューを使用しています (論理端末定義 (mcftalcle -k) の quekind オペランドを省略, または memory を指定)。 論理端末が閉塞しているなどの要因によって, 送信メッセージが送信済みになっていません。 送信済みのメッセージがディスクキューに保持されていません (保持メッセージ数はメッセージキューサービス定義の quegrp コマンドの-m オプションで指定します)。
	データ名 T で指定した論理端末に割り当てられているメッセージキューは, システム開始時に障害が発生したため, メモリキューを代用して縮退運転をしています。
70905	再送するメッセージのセグメントの長さが, UAP 共通定義 (mcfmuap -e) で指定した値を超えています。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	MCF が終了処理中のため, メッセージの再送を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージキューから取り出したメッセージを格納するバッファ (作業領域) を, メモリ上に確保できませんでした。
71108	メッセージを再送しようとしたますが, 再送先の管理テーブルが確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none"> 先頭セグメントを受信する CBLDCMCF('RECEIVE△')を呼び出す前に, CBLDCMCF('RESEND△△')を呼び出しています。 非トランザクション属性の MHP から, CBLDCMCF('RESEND△△')を呼び出しています。
	< SPP の実行でリターンした場合 > トランザクションでない SPP の処理から, CBLDCMCF('RESEND△△')を呼び出しています。
72001	データ名 P またはデータ名 T に設定した論理端末名称が間違っています。
	データ名 P に設定した論理端末名称は, 定義されていません。
	該当する論理端末には, メッセージを再送できません。
72016	データ名 J に設定した値が間違っています。
	データ名 M4 に設定した値が間違っています。
	データ名 U に設定した値が間違っています。

ステータスコード	意味
72016	データ名 N, データ名 S, またはデータ名 X に設定した値が間違っています。
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
上記以外	プログラムの破壊などによる, 予期しないエラーが発生しました。

注意事項

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

-e オプション

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

-l オプション

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

CBLDCMCF('SEND ') – メッセージの送信 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
02 データ名A PIC X(8) VALUE 'SEND ' .  
02 データ名B PIC X(5).  
02 FILLER PIC X(3).  
02 データ名C PIC X(4) VALUE SPACE.  
02 データ名D PIC X(4) VALUE SPACE.  
02 データ名E PIC 9(8).  
02 データ名F PIC 9(8).  
02 データ名G PIC 9(9) COMP VALUE ZERO.  
02 データ名H PIC X(4).  
02 データ名I PIC X(4) VALUE SPACE.  
02 データ名J PIC X(4).  
02 データ名K PIC X(4).  
02 データ名L PIC X(8) VALUE SPACE.  
02 データ名M1 PIC X(4) VALUE SPACE.  
02 データ名M2 PIC X(8) VALUE SPACE.  
02 データ名M3 PIC X(4) VALUE SPACE.  
02 データ名M4 PIC 9(9) COMP VALUE ZERO.  
02 データ名M5 PIC 9(9) COMP.  
02 データ名M6 PIC X(1).  
02 データ名M7 PIC X(1) VALUE '2'.  
02 データ名M8 PIC X(1).  
02 データ名N PIC X(13) 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).  
02 データ名R PIC X(8) VALUE SPACE.  
02 データ名T1 PIC X(6) VALUE SPACE.  
02 データ名T2 PIC X(22) VALUE LOW-VALUE.  
01 一意名3.  
02 データ名U PIC 9(4) COMP.  
02 データ名V PIC X(2).  
02 データ名W PIC X(n).
```

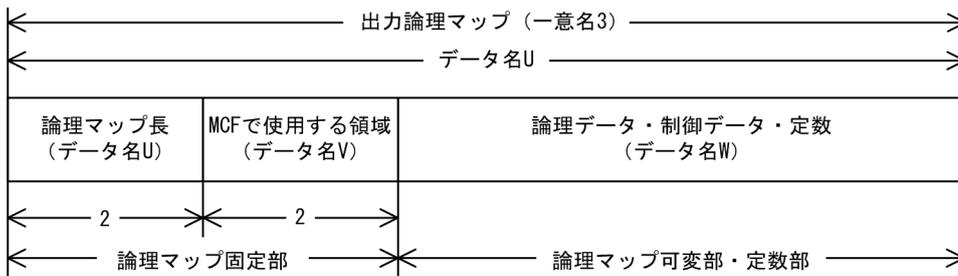
機能

論理端末に一方送信メッセージを送信します。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。

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

セグメントを送信する領域（一意名 3 で示す領域）の形式を次に示します。

(単位：バイト)



TP1/NET/XMAP3 の論理端末に送信する場合は、XMAP3 で作成した出力論理マップを COPY 文で取り込み、一意名 3 として指定します。出力論理マップの詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」、またはマニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド」を参照してください。コーディング例については、「付録 H.1 コーディング例」を参照してください。

UAP で値を設定するデータ領域

●データ名 A

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

●データ名 C, データ名 D

空白を設定します。

●データ名 E, データ名 F

MCF で使用する領域です。

●データ名 G

0 を設定します。

●データ名 H

送信するメッセージのセグメント種別を設定します。

VALUE 'ESI△'

先頭セグメントを送信する場合に設定します。

VALUE 'EMI△'

次のどれかの場合に設定します。

- ・ 論理メッセージが単一セグメントの場合
- ・ 先頭セグメントの応答送信後に応答メッセージの送信の終了を連絡する場合

●データ名 I

空白を設定します。

●データ名 J

一般として送信するか優先として送信するかを設定します。

VALUE 'NORM'

一般の一方送信メッセージとして送信する場合に設定します。

VALUE 'PRIO'

優先の一方送信メッセージとして送信する場合に設定します。

空白

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

●データ名 K

出力通番を付けるかどうかを設定します。

VALUE 'SEQ△'

出力通番を付ける場合に設定します。

VALUE 'NSEQ'

出力通番を付けない場合に設定します。

空白

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

●データ名 L, データ名 M1, データ名 M2, データ名 M3

空白を設定します。

●データ名 M4

0 を設定します。

●データ名 M5

MCF で使用する領域です。

●データ名 M6

マッピングモードを設定します。

VALUE '0'

論理マップと物理マップの情報でマッピングする場合に設定します。

VALUE '1'

論理マップだけでマッピングする場合に設定します。

空白

省略されたものとして、「VALUE '0」（論理マップと物理マップの情報でマッピングする）が設定されます。

●データ名 M7

使用する送受信バッファの形式を「VALUE '2」と設定します。

●データ名 M8

空白または「VALUE '0」を設定します。

●データ名 N

MCF で使用する領域です。

●データ名 O

分岐送信を示す「VALUE 'OUT△」を設定します。

●データ名 P

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

●データ名 Q

マップ名を設定します。マップ名は最大 6 バイトの長さです。マップ名の後ろを空白で埋めて、8 文字になるように設定してください。

空白を設定した場合、論理端末定義 (mcftalcle -p) の dflmap オペランドで指定したデフォルトマップ名を使用します。

●データ名 R, データ名 T1

空白を設定します。

●データ名 T2

MCF で使用する領域です。

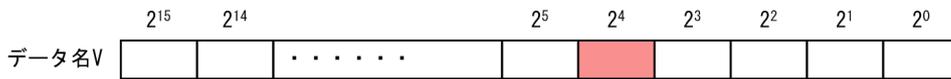
●データ名 U

送信する出力論理マップの長さ（論理マップ固定部の長さ+論理マップ可変部・定数部の長さ）を設定します。先頭セグメントの送信後、メッセージの送信の終了を連絡する場合には、4 を設定してください。

●データ名 V

送信完了通知イベントを設定します。データ名 V とビットの設定値の関係を次に示します。

AIX, HP-UX の場合

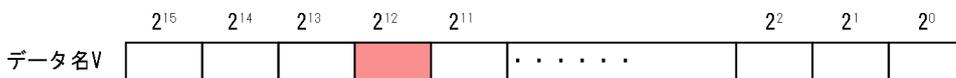


(凡例) : イベントの可否を設定するビット

送信完了通知イベントおよび送信障害通知イベントの可否は、データ名 V の領域の 2^4 ビットに設定します。

- 領域の 2^4 ビットに 1 を設定すると、送信完了通知イベントおよび送信障害通知イベントを通知させます。ただし、これらのイベントを処理する MHP をアプリケーション属性定義 (mcfaalcap) で指定していない場合は無効です。
メッセージが複数のセグメントで構成される場合、先頭セグメント送信時の指定が有効になります。
- 領域の 2^4 ビットに 0 を設定すると、送信完了イベントおよび送信障害通知イベントを通知させません。

Linux, Windows の場合



(凡例) : イベントの可否を設定するビット

送信完了通知イベントおよび送信障害通知イベントの可否は、データ名 V の領域の 2^{12} ビットに設定します。

- 領域の 2^{12} ビットに 1 を設定すると、送信完了通知イベントおよび送信障害通知イベントを通知させます。ただし、これらのイベントを処理する MHP をアプリケーション属性定義 (mcfaalcap) で指定していない場合は無効です。
メッセージが複数のセグメントで構成される場合、先頭セグメント送信時の指定が有効になります。
- 領域の 2^{12} ビットに 0 を設定すると、送信完了イベントおよび送信障害通知イベントを通知させません。

●データ名 W

送信するセグメントの内容 (論理マップ可変部・定数部に出力する内容) を設定します。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。

ステータスコード	意味
71002	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	データ名 U に 32004 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたますが、送信先の管理テーブルを確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する CBLDCMCF('RECEIVE△')を呼び出す前に、CBLDCMCF('SEND△△△△')を呼び出しています。
	< SPP の実行でリターンした場合 > トランザクションでない SPP の処理から、CBLDCMCF('SEND△△△△')を呼び出しています。
72001	データ名 P に設定した論理端末名称が間違っています。
	データ名 P に設定した論理端末名称は、定義されていません。
	CBLDCMCF('SEND△△△△')を呼び出せない論理端末を設定しています。
72005	<データ名 H で VALUE 'ESI△'を設定した場合 > データ名 U に 0 から 4 バイト、またはマイナス値を設定しています。
72016	データ名 N またはデータ名 T2 に設定した値が間違っています。
	データ名 T1 に設定した値が間違っています。
	データ名 J に設定した値が間違っています。
	データ名 M1 に設定した値が間違っています。
	データ名 M7, M8 に設定した値が間違っています。
72017	データ名 K に設定した値が間違っています。
72019	データ名 M6 に設定した値が間違っています。
72020	データ名 I に設定した値が間違っています。
72024	データ名 O に設定した値が間違っています。
72026	データ名 H に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72041	<データ名 H で VALUE 'EMI△'を設定した場合 > <ul style="list-style-type: none"> データ名 U に 0 から 4 バイト、またはマイナス値を設定しています。 データ名 H に VALUE 'ESI△'を設定した CBLDCMCF('SEND△△△△')を呼び出さないので、メッセージの送信の終了を連絡しています。

ステータスコード	意味
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

CBLDCMCF('TACTCN ') – コネクションの確立 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
  02 データ名A PIC X(8) VALUE 'TACTCN '  
  02 データ名B PIC X(5).  
  02 FILLER PIC X(3).  
  02 データ名C PIC X(4).  
  02 データ名D1 PIC X(1) VALUE SPACE.  
  02 データ名D2 PIC X(1) VALUE SPACE.  
  02 データ名D3 PIC X(26) VALUE SPACE.  
  02 データ名E PIC 9(9) COMP.  
  02 データ名F1 PIC X(8).  
  02 データ名F2 PIC X(56) VALUE SPACE.  
  02 データ名G PIC X(8) VALUE SPACE.  
  02 データ名H PIC X(8) VALUE SPACE.  
  02 データ名I PIC X(144) VALUE SPACE.  
  02 データ名J PIC X(184) VALUE SPACE.  
  02 データ名K PIC 9(9) COMP VALUE ZERO.  
  
01 一意名2.  
  02 データ名L PIC 9(9) COMP VALUE ZERO.
```

機能

コネクションを確立します。

なお、CBLDCMCF('TACTCN△△')の正常終了は、コネクション確立要求を TP1/NET/XMAP3 が正常に受け付けたことを意味します。このため、相手システムとのコネクションの確立が正常に完了したことを示すものではありません。

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

UAP で値を設定するデータ領域

●データ名 A

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

●データ名 C

確立するコネクションの指定方法を設定します。

VALUE 'LE△△'

確立するコネクションを論理端末名称で指定するときに設定します。

VALUE 'CN△△'

確立するコネクションをコネクション ID で指定するときに設定します。

空白

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

●データ名 D1, データ名 D2, データ名 D3

空白を設定します。

●データ名 E

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

論理端末名称を使用してコネクションの確立を要求する場合は、無効となります。

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

注※

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

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

●データ名 F1

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

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

空白を設定します。

●データ名 K

0 を設定します。

●データ名 L

0 を設定します。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71001	MCF が開始処理中のため、CBLDCMCF('TACTCN△△')が受け付けられません。
71002	MCF が終了処理中のため、CBLDCMCF('TACTCN△△')が受け付けられません。
71004	CBLDCMCF('TACTCN△△')の処理中にメモリ不足が発生しました。
71005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
71006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
71007	指定された接続 ID は登録されていません。
71008	指定された論理端末名称は登録されていません。
71009	CBLDCMCF('TACTCN△△')が、該当する通信プロセスではサポートされていません。
71010	MCF 通信プロセスに接続の確立を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
71011	接続が削除されているため、CBLDCMCF('TACTCN△△')が受け付けられません。
71014	TP1/NET/NCSEB, または TP1/NET/X25-Extended の論理端末名称を指定しています。または TP1/NET/OSI-TP の接続グループを指定しています。
72028	データ名 A に設定した値が間違っています。
72052	データ名 K に 0 でない値が設定されています。
72053	データ名 L に 0 でない値が設定されています。
72058	データ名 C に 'LE△△', 'CN△△', または空白以外の値が設定されています。
72059	データ名 D1, データ名 D2, またはデータ名 D3 に空白でない値が設定されています。
72061	データ名 E に 0 未満, または 240 以上の値が設定されています。
72063	データ名 F1 に空白が設定されています。
72065	データ名 F2 に空白でない値が設定されています。
72066	データ名 G に空白でない値が設定されています。
72068	データ名 H に空白でない値が設定されています。
72070	データ名 I に空白でない値が設定されています。
72072	データ名 J に空白でない値が設定されています。

ステータスコード	意味
72074	データ名 F1 に設定された文字列中に不正な文字があります。

CBLDCMCF('TACTLE ') – 論理端末の閉塞解除 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

機能

論理端末の閉塞を解除します。

なお、CBLDCMCF('TACTLE△△')の正常終了は、論理端末の閉塞解除要求を TP1/NET/XMAP3 が正常に受け付けたことを意味します。このため、論理端末の閉塞解除が正常に完了したことを示すものではありません。

CBLDCMCF('TACTLE△△')の呼び出し後に論理端末に関する何らかの処理をする場合は、CBLDCMCF('TLSLE△△△')を用いて論理端末の状態を確認してください。

UAP で値を設定するデータ領域

●データ名 A

論理端末の閉塞の解除を示す要求コード「VALUE 'TACTLE△△'」を設定します。

●データ名 C, データ名 D1, データ名 D2, データ名 D3

空白を設定します。

●データ名 E

処理対象の論理端末を持つ MCF 通信サービスの MCF 通信プロセス識別子※を設定します。設定できる範囲は 0～239 です。

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

注※

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

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

●データ名 F1

閉塞解除する論理端末の名称を設定します。論理端末名称は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称の後ろを空白で埋めてください。

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

空白を設定します。

●データ名 K, データ名 L

0 を設定します。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71001	MCF が開始処理中のため、CBLDCMCF('TACTLE△△')が受け付けられません。
71002	MCF が終了処理中のため、CBLDCMCF('TACTLE△△')が受け付けられません。
71004	CBLDCMCF('TACTLE△△')の処理中にメモリ不足が発生しました。
71005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
71006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
71008	指定された論理端末名称は登録されていません。
71009	CBLDCMCF('TACTLE△△')が、該当する通信プロセスではサポートされていません。
71010	MCF 通信プロセスに論理端末の閉塞の解除を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。

ステータスコード	意味
71011	論理端末が削除されているため、CBLDCMCF('TACTLE△△')が受け付けられません。
72028	データ名 A に設定した値が間違っています。
72052	データ名 K に 0 でない値が設定されています。
72053	データ名 L に 0 でない値が設定されています。
72058	データ名 C に空白でない値が設定されています。
72059	データ名 D1, データ名 D2, またはデータ名 D3 に空白でない値が設定されています。
72061	データ名 E に 0 未満または 240 以上の値が設定されています。
72063	データ名 F1 に空白が設定されています。
72065	データ名 F2 に空白でない値が設定されています。
72066	データ名 G に空白でない値が設定されています。
72068	データ名 H に空白でない値が設定されています。
72070	データ名 I に空白でない値が設定されています。
72072	データ名 J に空白でない値が設定されています。
72074	データ名 F1 に設定された文字列中に不正な文字があります。

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

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
02 データ名A PIC X(8) VALUE 'TDCTCN'.  
02 データ名B PIC X(5).  
02 FILLER PIC X(3).  
02 データ名C PIC X(4).  
02 データ名D1 PIC X(1) VALUE '1'.  
02 データ名D2 PIC X(1) VALUE SPACE.  
02 データ名D3 PIC X(26) VALUE SPACE.  
02 データ名E PIC 9(9) COMP.  
02 データ名F1 PIC X(8).  
02 データ名F2 PIC X(56) VALUE SPACE.  
02 データ名G PIC X(8) VALUE SPACE.  
02 データ名H PIC X(8) VALUE SPACE.  
02 データ名I PIC X(144) VALUE SPACE.  
02 データ名J PIC X(184) VALUE SPACE.  
02 データ名K PIC 9(9) COMP VALUE ZERO.  
  
01 一意名2.  
02 データ名L PIC 9(9) COMP VALUE ZERO.
```

機能

コネクションを解放します。

なお、CBLDCMCF('TDCTCN△△')の正常終了は、コネクション解放要求を TP1/NET/XMAP3 が正常に受け付けたことを意味します。このため、相手システムとのコネクションの解放が正常に完了したことを示すものではありません。

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

UAP で値を設定するデータ領域

●データ名 A

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

●データ名 C

解放するコネクションの指定方法を設定します。

VALUE 'LE△△'

解放するコネクションを論理端末名称で指定するときに設定します。

VALUE 'CN△△'

解放するコネクションをコネクション ID で指定するときに設定します。

空白

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

●データ名 D1

コネクションの強制解放指定である「VALUE '1」を設定します。

●データ名 D2, データ名 D3

空白を設定します。

●データ名 E

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

論理端末名称を使用してコネクションの解放を要求する場合は、無効となります。

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

注※

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

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

●データ名 F1

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

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

空白を設定します。

●データ名 K, データ名 L

0 を設定します。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71001	MCF が開始処理中のため、CBLDCMCF('TDCTCN△△')が受け付けられません。
71002	MCF が終了処理中のため、CBLDCMCF('TDCTCN△△')が受け付けられません。
71004	CBLDCMCF('TDCTCN△△')の処理中にメモリ不足が発生しました。
71005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
71006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
71007	指定されたコネクション ID は登録されていません。
71008	指定された論理端末名称は登録されていません。
71009	CBLDCMCF('TDCTCN△△')が、該当する通信プロセスではサポートされていません。
71010	MCF 通信プロセスにコネクションの解放を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
71011	コネクションが削除されているため、CBLDCMCF('TDCTCN△△')が受け付けられません。
71014	TP1/NET/NCSB, または TP1/NET/X25-Extended の論理端末名称を指定しています。または TP1/NET/OSI-TP のコネクショングループ名を指定しています。
72028	データ名 A に設定した値が間違っています。
72052	データ名 K に 0 でない値が設定されています。
72053	データ名 L に 0 でない値が設定されています。
72058	データ名 C に 'LE△△', 'CN△△', または空白以外の値が設定されています。
72059	データ名 D2, またはデータ名 D3 に空白でない値が設定されています。
72061	データ名 E に 0 未満または 240 以上の値が設定されています。
72063	データ名 F1 に空白が設定されています。
72065	データ名 F2 に空白でない値が設定されています。
72066	データ名 G に空白でない値が設定されています。
72068	データ名 H に空白でない値が設定されています。
72070	データ名 I に空白でない値が設定されています。
72072	データ名 J に空白でない値が設定されています。

ステータスコード	意味
72074	データ名 F1 に設定された文字列中に不正な文字があります。
72075	データ名 D1 に 1 以外の値が設定されています。

CBLDCMCF('TDCTLE ') – 論理端末の閉塞 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

機能

論理端末を閉塞します。

なお、CBLDCMCF('TDCTLE△△')の正常終了は、論理端末の閉塞要求を TP1/NET/XMAP3 が正常に受け付けたことを意味します。このため、論理端末の閉塞が正常に完了したことを示すものではありません。

CBLDCMCF('TDCTLE△△')の呼び出し後に論理端末に関する何らかの処理をする場合は、CBLDCMCF('TLSLE△△△')を用いて論理端末の状態を確認してください。

UAP で値を設定するデータ領域

●データ名 A

論理端末の閉塞を示す要求コード「VALUE 'TDCTLE△△」を設定します。

●データ名 C, データ名 D1, データ名 D2, データ名 D3

空白を設定します。

●データ名 E

処理対象の論理端末を持つ MCF 通信サービスの MCF 通信プロセス識別子※を設定します。設定できる範囲は 0～239 です。

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

注※

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

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

●データ名 F1

閉塞する論理端末の名称を設定します。論理端末名称は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称の後ろを空白で埋めてください。

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

空白を設定します。

●データ名 K, データ名 L

0 を設定します。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71001	MCF が開始処理中のため、CBLDCMCF('TDCTLE△△')が受け付けられません。
71002	MCF が終了処理中のため、CBLDCMCF('TDCTLE△△')が受け付けられません。
71004	CBLDCMCF('TDCTLE△△')の処理中にメモリ不足が発生しました。
71005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
71006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
71008	指定された論理端末名称は登録されていません。
71009	CBLDCMCF('TDCTLE△△')が、該当する通信プロセスではサポートされていません。
71010	MCF 通信プロセスに論理端末の閉塞を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。

ステータスコード	意味
71011	論理端末が削除されているため、CBLDCMCF('TDCTLE△△')が受け付けられません。
72028	データ名 A に設定した値が間違っています。
72052	データ名 K に 0 でない値が設定されています。
72053	データ名 L に 0 でない値が設定されています。
72058	データ名 C に空白でない値が設定されています。
72059	データ名 D2, またはデータ名 D3 に空白でない値が設定されています。
72061	データ名 E に 0 未満または 240 以上の値が設定されています。
72063	データ名 F1 に空白が設定されています。
72065	データ名 F2 に空白でない値が設定されています。
72066	データ名 G に空白でない値が設定されています。
72068	データ名 H に空白でない値が設定されています。
72070	データ名 I に空白でない値が設定されています。
72072	データ名 J に空白でない値が設定されています。
72074	データ名 F1 に設定された文字列中に不正な文字があります。
72075	データ名 D1 に空白でない値が設定されています。

CBLDCMCF('TEMPGET ') – 一時記憶データの受け取り (COBOL 言語)

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

機能

継続問い合わせ応答用一時記憶領域に格納されている一時記憶データを受け取ります。

データ名 G の長さ (7~32006 バイト) を超える一時記憶データがある場合、超えた分については切り捨てます。

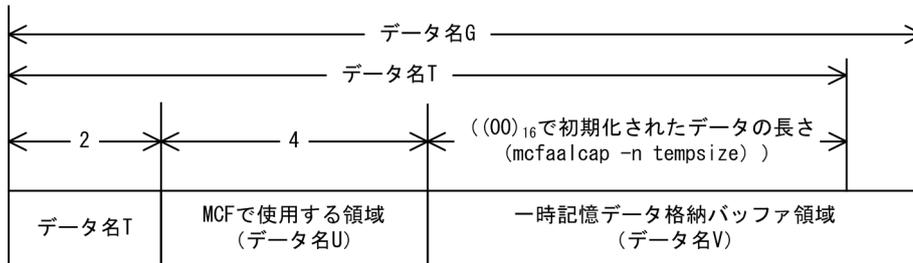
データ名 G から 6 を減算した値と比べて一時記憶データ長の方が短い場合、データ名 V に一時記憶データを設定します。データ名 V の残りの領域については何も設定しません。

受け取り要求実行時、初期状態（継続問い合わせ応答開始後、CBLDCMCF('TEMPPUT△')を1回も実行していない状態）の場合、アプリケーション属性定義（mcfaalcap -n）の tempsize オペランドで指定した長さの(00)₁₆の一時記憶データがあるものとしてCBLDCMCF('TEMPGET△')を実行します。

受け取り領域（一意名3で示す領域）の形式を次に示します。

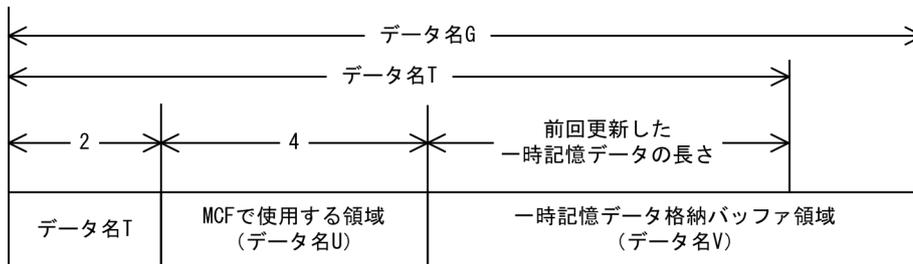
- CBLDCMCF('TEMPPUT△')未実行（継続問い合わせ応答開始後、CBLDCMCF('TEMPPUT△')を実行していない（初期状態））

（単位：バイト）



- CBLDCMCF('TEMPPUT△')実行済み（継続問い合わせ応答開始後、CBLDCMCF('TEMPPUT△')を1回以上実行）

（単位：バイト）



UAP で値を設定するデータ領域

●データ名A

一時記憶データの受け取りを示す要求コード「VALUE 'TEMPGET△」を設定します。

●データ名C, データ名D

空白を設定します。

●データ名E, データ名F

MCF で使用する領域です。

●データ名G

一時記憶データを受け取る領域の長さを7～32006バイトで設定します。

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

空白を設定します。

●データ名 M4, データ名 M5

0 を設定します。

●データ名 M6

空白を設定します。

●データ名 M7

使用するバッファの形式を「VALUE '2」 と設定します。

●データ名 N

MCF で使用する領域です。

●データ名 O, データ名 P, データ名 Q, データ名 R

空白を設定します。

●データ名 S, データ名 U

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

●データ名 T

前回更新した一時記憶データの長さ + 6 が返されます。初期状態の場合、継続問い合わせ応答用一時記憶領域の長さ（アプリケーション属性定義（mcfaalcap -n）の tempsize オペランドの指定値）+ 6 が返されます。

●データ名 V

受け取った一時記憶データが返されます。初期状態の場合、継続問い合わせ応答用一時記憶領域の長さ（アプリケーション属性定義（mcfaalcap -n）の tempsize オペランドの指定値）分だけ(00)₁₆ が埋められます。

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。
72000	SPP では CBLDCMCF('TEMPGET△')を呼び出せません。
72013	一時記憶データ格納バッファ領域の長さ（データ名 G から 6 を減算した値）を超える一時記憶データを受け取りました。一時記憶データ格納バッファ領域の長さを超える一時記憶データを切り捨てました。

ステータスコード	意味
72016	データ名 N またはデータ名 S に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72036	データ名 G の設定値が不足しています。7 バイト以上の領域を確保してください。
72101	継続問い合わせ応答型でないアプリケーションで、CBLDCMCF('TEMPGET△')を呼び出しています。
72106	先頭セグメントを受信する CBLDCMCF('RECEIVE△')を呼び出す前に、CBLDCMCF('TEMPGET△')を呼び出しています。
72107	CBLDCMCF('CONTEND△')を呼び出したあとで、CBLDCMCF('TEMPGET△')を呼び出しています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

CBLDCMCF('TEMPPUT ') – 一時記憶データの更新 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

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

機能

継続問い合わせ応答用一時記憶領域に格納されている一時記憶データを更新します。

アプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドには、更新する一時記憶データ長 (データ名 T から 6 を減算した値) 以上の値を指定してください。

更新する領域 (一意名 3 で示す領域) の形式を次に示します。

(単位: バイト)



UAP で値を設定するデータ領域

●データ名 A

一時記憶データの更新を示す要求コード「VALUE 'TEMPPUT△」を設定します。

●データ名 C, データ名 D

空白を設定します。

●データ名 E, データ名 F

MCF で使用する領域です。

●データ名 G

0 を設定します。

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

空白を設定します。

●データ名 M4, データ名 M5

0 を設定します。

●データ名 M6

空白を設定します。

●データ名 M7

使用するバッファの形式を「VALUE '2」」と設定します。

●データ名 N

MCF で使用する領域です。

●データ名 O, データ名 P, データ名 Q, データ名 R

空白を設定します。

●データ名 S

MCF で使用する領域です。

●データ名 T

一時記憶データの更新データ長を設定します。

●データ名 U

MCF で使用する領域です。

●データ名 V

一時記憶データが格納されている領域を設定します。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71103	一時記憶データを更新するための領域をメモリ上に確保できませんでした。
72000	SPP では CBLDCMCF('TEMPPUT△')を呼び出せません。
72016	データ名 N またはデータ名 S に設定した値が間違っています。
72028	データ名 A に設定した値が間違っています。
72035	データ名 T に設定した更新データの長さが、アプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドで定義した長さを超えています。 データ名 T に 0 から 6 バイト、またはマイナス値を設定しています。
72101	継続問い合わせ応答型でないアプリケーションで、CBLDCMCF('TEMPPUT△')を呼び出しています。
72105	CBLDCMCF('TEMPGET△')を呼び出す前に、CBLDCMCF('TEMPPUT△')を呼び出しています。
72106	先頭セグメントを受信する CBLDCMCF('RECEIVE△')を呼び出す前に、CBLDCMCF('TEMPPUT△')を呼び出しています。
72107	CBLDCMCF('CONTEND△')を呼び出したあとで、CBLDCMCF('TEMPPUT△')を呼び出しています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

CBLDCMCF('TLSCN ') - コネクションの状態取得 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
02 データ名A PIC X(8) VALUE 'TLSCN ' .  
02 データ名B PIC X(5).  
02 FILLER PIC X(3).  
02 データ名C PIC X(4).  
02 データ名D PIC X(28) VALUE SPACE.  
02 データ名E PIC 9(9) COMP.  
02 データ名F1 PIC X(8).  
02 データ名F2 PIC X(56) VALUE SPACE.  
02 データ名G PIC X(8) VALUE SPACE.  
02 データ名H PIC X(8) VALUE SPACE.  
02 データ名I PIC X(144) VALUE SPACE.  
02 データ名J PIC X(184) VALUE SPACE.  
02 データ名K PIC 9(9) COMP VALUE ZERO.  
  
01 一意名2.  
02 データ名L PIC 9(9) COMP VALUE ZERO.  
  
01 一意名3.  
02 データ名M PIC 9(9) COMP.  
02 一意名4.  
03 データ名N PIC X(8).  
03 データ名O PIC X(4).  
03 データ名P PIC X(4).  
03 データ名Q PIC X(40) VALUE LOW-VALUE.
```

機能

コネクションの状態を取得します。

UAP で値を設定するデータ領域

●データ名A

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

●データ名C

状態を取得するコネクションの指定方法を設定します。

VALUE 'LE△△'

状態を取得するコネクションを論理端末名称で指定するときに設定します。

VALUE 'CN△△'

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

空白

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

●データ名 D

空白を設定します。

●データ名 E

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

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

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

注※

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

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

●データ名 F1

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

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

空白を設定します。

●データ名 K, データ名 L

0 を設定します。

●データ名 M

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

処理終了後は、該当するコネクションの個数が返されます。

●データ名 Q

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

●データ名 M

この命令文の対象となった接続の個数が返されます。

●データ名 N

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

●データ名 O

要求した接続のプロトコル種別が設定されます。

VALUE 'XP△△'

TP1/NET/XMAP3 プロトコル

●データ名 P

要求した接続の状態として、次の値が設定されます。

VALUE 'ACT△'

確立状態

VALUE 'ACTB'

確立処理中状態

VALUE 'DCT△'

解放状態

VALUE 'DCTB'

解放処理中状態

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71001	MCF が開始処理中のため、CBLDCMCF('TLSCN△△△')が受け付けられません。
71004	CBLDCMCF('TLSCN△△△')の処理中にメモリ不足が発生しました。
71005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
71006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
71007	指定された接続 ID は登録されていません。

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

CBLDCMCF('TLSLE ') – 論理端末の状態取得 (COBOL 言語)

形式

PROCEDURE DIVISION の指定

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

DATA DIVISION の指定

```
01 一意名1.  
02 データ名A PIC X(8) VALUE 'TLSLE ' .  
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(28) VALUE SPACE.  
02 データ名E PIC 9(9) COMP.  
02 データ名F1 PIC X(8).  
02 データ名F2 PIC X(56) VALUE SPACE.  
02 データ名G PIC X(8) VALUE SPACE.  
02 データ名H PIC X(8) VALUE SPACE.  
02 データ名I PIC X(144) VALUE SPACE.  
02 データ名J PIC X(184) VALUE SPACE.  
02 データ名K PIC 9(9) COMP VALUE ZERO.  
  
01 一意名2.  
02 データ名L PIC 9(9) COMP VALUE ZERO.  
  
01 一意名3.  
02 データ名M PIC 9(9) COMP.  
02 一意名4.  
03 データ名N PIC X(8).  
03 データ名O PIC X(4) VALUE LOW-VALUE.  
03 データ名P PIC X(4).  
03 データ名Q PIC X(40) VALUE LOW-VALUE.
```

機能

論理端末の状態を取得します。

UAP で値を設定するデータ領域

●データ名 A

論理端末の状態取得を示す要求コード「VALUE 'TLSLE△△△」を設定します。

●データ名 C, データ名 D

空白を設定します。

●データ名 E

処理対象の論理端末を持つ MCF 通信サービスの MCF 通信プロセス識別子※を設定します。設定できる範囲は 0～239 です。

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

注※

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

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

●データ名 F1

状態を取得する論理端末の名称を設定します。論理端末名称は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称の後ろを空白で埋めてください。

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

空白を設定します。

●データ名 K, データ名 L

0 を設定します。

●データ名 M

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

処理終了後は、該当する論理端末の個数が返されます。

●データ名 O, データ名 Q

MCF で使用する領域です。

OpenTP1 から値が返されるデータ領域

●データ名 B

ステータスコードが、5 けたの数字で返されます。

●データ名 M

この命令文の対象となった論理端末の個数が返されます。

●データ名 N

要求した論理端末の名称が設定されます。8 バイトに満たない場合、論理端末名称の後ろが空白で埋められます。

●データ名P

要求した論理端末の状態として、次の値が設定されます。

VALUE 'ACT△'

閉塞解除状態

VALUE 'DCT△'

閉塞状態

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71001	MCF が開始処理中のため、CBLDCMCF('TLSLE△△△')が受け付けられません。
71004	CBLDCMCF('TLSLE△△△')の処理中にメモリ不足が発生しました。
71005	通信障害が発生しました。原因については、メッセージログファイルを参照してください。
71006	内部障害が発生しました。原因については、メッセージログファイルを参照してください。
71008	指定された論理端末名称は登録されていません。
71009	CBLDCMCF('TLSLE△△△')が、該当する通信プロセスではサポートされていません。
71010	MCF 通信プロセスに論理端末の状態取得を要求しましたが、受け付けられませんでした。原因については、メッセージログファイルを参照してください。
71011	論理端末が削除されているため、CBLDCMCF('TLSLE△△△')が受け付けられません。
72028	データ名 A に設定した値が間違っています。
72052	データ名 K に 0 でない値が設定されています。
72053	データ名 L に 0 でない値が設定されています。
72058	データ名 C に空白でない値が設定されています。
72059	データ名 D に空白でない値が設定されています。
72061	データ名 E に 0 未満または 240 以上の値が設定されています。
72063	データ名 F1 に空白が設定されています。
72065	データ名 F2 に空白でない値が設定されています。
72066	データ名 G に空白でない値が設定されています。
72068	データ名 H に空白でない値が設定されています。
72070	データ名 I に空白でない値が設定されています。
72072	データ名 J に空白でない値が設定されています。
72074	データ名 F1 に設定された文字列中に不正な文字があります。
72076	データ名 M に 1 以外の値が設定されています。

DISABLE – 継続問い合わせ応答の終了（データ操作言語）

形式

DATA DIVISION（通信記述項）の指定

```
CD 通信記述名
   FOR I-O STORAGE
   [STATUS KEY IS データ名1] .
```

PROCEDURE DIVISION（通信文）の指定

```
DISABLE 通信記述名.
```

機能

次に示す CALL インタフェースの機能を実現します。

- 継続問い合わせ応答の終了 CBLDCMCF('CONTEND△')

UAP で値を設定する項目

●FOR 句

継続問い合わせ応答の終了を示す I-O STORAGE を設定します。

OpenTP1 から値が返される項目

●STATUS KEY 句

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

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
72000	< MHP の実行でリターンした場合 > 先頭セグメントを受信する RECEIVE 文を呼び出す前に、DISABLE 文を呼び出しています。 < SPP の実行でリターンした場合 > SPP では DISABLE 文を呼び出せません。
72042	通信記述項に SYMBOLIC TERMINAL 句の設定があり、空白以外が設定されています。
72101	継続問い合わせ応答型でないアプリケーションで、DISABLE 文を呼び出しています。
72107	DISABLE 文を 2 回以上呼び出しています。

ステータスコード	意味
72111	<p>次起動アプリケーションを設定して SEND（応答メッセージの送信）文を呼び出したあと、DISABLE 文を呼び出しています。</p> <p>継続問い合わせ応答型のアプリケーション名を設定して SEND（アプリケーションプログラムの起動）文を呼び出したあと、DISABLE 文を呼び出しています。</p>
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

RECEIVE - メッセージの受信 (データ操作言語)

形式

DATA DIVISION (通信記述項) の指定

```
CD 通信記述名
   FOR {INPUT|I-0}
   [STATUS KEY IS データ名1]
   [SYMBOLIC TERMINAL IS データ名2]
   [MESSAGE DATE IS データ名3]
   [MESSAGE TIME IS データ名4]
   [MAP NAME IS データ名5]
   [SYNCHRONOUS MODE IS {ASYNC|データ名6} ] .
```

DATA DIVISION (データ記述項) の指定

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

PROCEDURE DIVISION (通信文) の指定

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

機能

次に示す CALL インタフェースの機能を実現します。

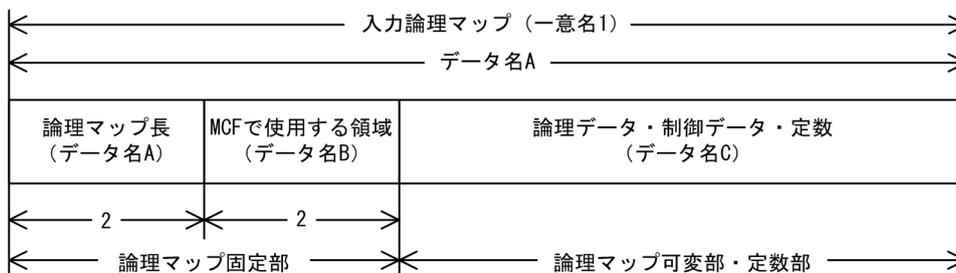
- メッセージの受信 CBLDCMCF('RECEIVE△')

受信できるメッセージの一つのセグメントの最大長は、32763 バイトまでです。

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

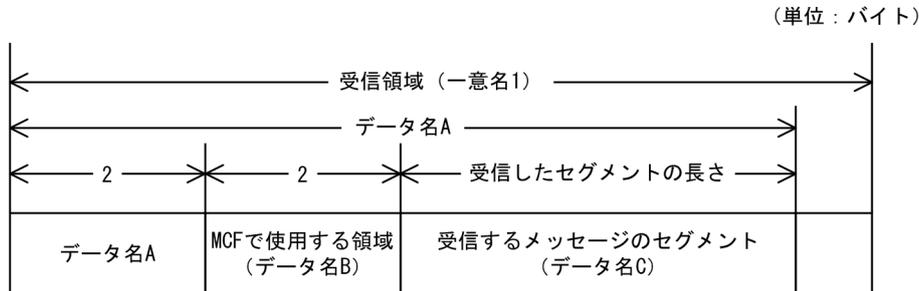
TP1/NET/XMAP3 の論理端末画面から入力されたメッセージを受信する場合

(単位: バイト)



XMAP3 で作成した入力論理マップを COPY 文で取り込み、一意名 1 として指定します。入力論理マップの詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」、またはマニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド」を参照してください。コーディング例については、「付録 H.1 コーディング例」を参照してください。

その他のメッセージを受信する場合



UAP で値を設定する項目

●FOR 句

次のどちらかの値を設定します。

INPUT

一方送信メッセージの受信

I-O

問い合わせメッセージの受信

●SYMBOLIC TERMINAL 句

中間セグメントまたは最終セグメントを受信する場合、データ名 2 に入力元の論理端末名称を設定します。先頭セグメントの受信時に返された論理端末名称を設定してください。論理端末名称は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称の後ろを空白で埋めてください。

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

●SYNCHRONOUS MODE 句

非同期型でのメッセージ受信を示す、次のどちらかの値を指定してください。

ASYNCH

非同期型のメッセージの受信

データ名 6

次の値を設定したデータ項目

'0'または'△': 非同期型のメッセージの受信

省略した場合は、ASYNC（非同期型のメッセージの受信）が設定されます。

●データ名 B

MCF で使用する領域です。

●FIRST

先頭セグメントまたは単一セグメントを受信する場合に設定します。

OpenTP1 から値が返される項目

●STATUS KEY 句

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

●SYMBOLIC TERMINAL 句

先頭セグメントまたは単一セグメントを受信する場合、入力元の論理端末名称を受け取りたいときに設定します。省略した場合は、論理端末名称を受け取れません。データ名 2 にメッセージ入力元の論理端末名称が返されます。論理端末名称は最大 8 バイトの長さです。8 バイトに満たない場合、論理端末名称の後ろが空白で埋められます。

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

●MESSAGE DATE 句

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

●MESSAGE TIME 句

メッセージを受信した時刻を受け取りたい場合に設定します。省略した場合は、メッセージを受信した時刻を受け取れません。データ名 4 にメッセージを受信した時刻が HHMMSS00（HH：時 MM：分 SS：秒 00 は固定）の形式で返されます。

●MAP NAME 句

XMAP3 のドローで指定した次画面名（マップ名）を受け取りたい場合に設定します。省略した場合は、次画面名を受け取れません。データ名 5 に次画面名が返されます。次画面名は最大 6 バイトの長さです。次画面名の後ろは空白で埋められます。

ただし、次の場合は、空白が返されます。

- 次画面名を指定していない場合
- アプリケーション起動機能（dc_mcf_execap 関数）で起動されたアプリケーションから、RECEIVE 文を発行する場合

●データ名 A

受信したセグメントの長さ+4が返されます。TP1/NET/XMAP3の論理端末画面から入力されたメッセージを受信した場合、受信した入力論理マップの長さ（論理マップ固定部の長さ+論理マップ可変部・定数部の長さ）が返されます。

●データ名 C

受信したセグメントが返されます。TP1/NET/XMAP3の論理端末画面から入力されたメッセージを受信した場合、論理マップ可変部・定数部に設定されるデータが返されます。一つのセグメントで32763バイトまで受信できます。

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71000	先頭セグメントを受信する RECEIVE 文を 2 回以上呼び出しています。中間セグメントおよび最終セグメントを受信する場合は、FIRST を設定しないで RECEIVE 文を呼び出してください。
71001	メッセージの最終セグメントを受信したあとで、次のセグメントを受信する RECEIVE 文を呼び出しています。直前に呼び出した RECEIVE 文でメッセージをすべて受信しました。このステータスコードが返されたあとに、再び RECEIVE 文を呼び出した場合は、ステータスコード 72000 が返されます。
71002	メッセージキューからの入力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	MCF 終了処理中のため、メッセージの受信を受け付けられません。
71108	メッセージの受信に必要な管理テーブルを確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none">先頭セグメントを受信する RECEIVE 文を呼び出す前に、中間セグメントまたは最終セグメントを受信する RECEIVE 文を呼び出しています。先頭セグメントを受信する場合は、FIRST を設定して RECEIVE 文を呼び出してください。ステータスコード 71001 が返されたあとで、RECEIVE 文を呼び出しています。
	< SPP の実行でリターンした場合 > SPP では RECEIVE 文を呼び出せません。
72001	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。
	SYMBOLIC TERMINAL 句に設定した論理端末名称は、定義されていません。
	RECEIVE 文を実行できない論理端末を設定しています。
	SYNCHRONOUS MODE 句に SYNC が設定されているか、データ名 6 に 'I' が設定されています。

ステータスコード	意味
72013	データ名 C のサイズを超えるセグメントを受信しました。データ名 C のサイズを超えた部分は切り捨てられました。
	32763 バイトを超えるセグメントを受信しました。32763 バイトを超えた部分は切り捨てられました。
72016	通信文に WAITING 句が設定されています。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72036	データ名 C のサイズが不足しています。5 バイト以上の領域を確保してください。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

RECEIVE – 一時記憶データの受け取り (データ操作言語)

形式

DATA DIVISION (通信記述項) の指定

```
CD 通信記述名
   FOR {INPUT|I-0} STORAGE
   [STATUS KEY IS データ名1] .
```

DATA DIVISION (データ記述項) の指定

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

PROCEDURE DIVISION (通信文) の指定

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

機能

次に示す CALL インタフェースの機能を実現します。

- 一時記憶データの受け取り CBLDCMCF('TEMPGET△')

一時記憶データ格納バッファ領域 (データ名 C) は、1~32000 バイトの長さを確保してください。

データ名 C の長さを超える一時記憶データがある場合、超えた分については切り捨てます。

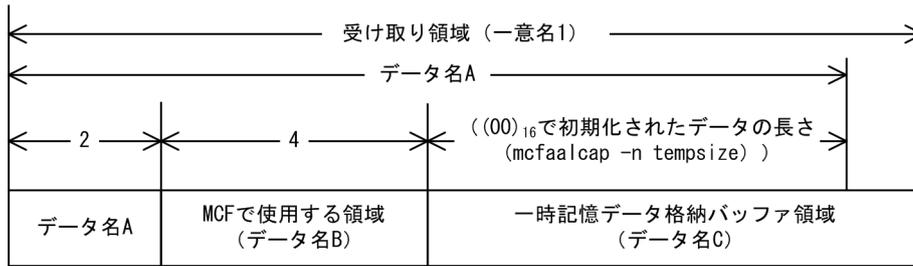
データ名 C の長さ比べて一時記憶データの長さの方が短い場合、データ名 C に一時記憶データを設定します。残りの領域については何も設定しません。

受け取り要求実行時に、初期状態 (継続問い合わせ応答開始後、SEND (一時記憶データの更新) 文を 1 回も実行していない状態) の場合、アプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドで指定した長さの(00)₁₆ の一時記憶データがあるものとして実行します。

受け取り後の一意名 1 の形式を次に示します。

- SEND（一時記憶データの更新）文未実行（継続問い合わせ応答開始後、SEND（一時記憶データの更新）文を実行していない（初期状態））

（単位：バイト）



- SEND（一時記憶データの更新）文実行済み（継続問い合わせ応答開始後、SEND（一時記憶データの更新）文を1回以上実行）

（単位：バイト）



UAP で値を設定する項目

●FOR 句

一時記憶データの受け取りであることを示す、INPUT STORAGE または I-O STORAGE を設定します。

●データ名 B

MCF で使用する領域です。

●MESSAGE, SEGMENT

どちらかを指定します。

OpenTP1 から値が返される項目

●STATUS KEY 句

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

●データ名 A

前回更新した一時記憶データの長さ + 6 が返されます。初期状態の場合、継続問い合わせ応答用一時記憶領域の長さ（アプリケーション属性定義（mcfaalcap -n）の tempsize オペランドの指定値）+ 6 が返されます。

●データ名 C

受け取った一時記憶データが返されます。初期状態の場合、継続問い合わせ応答用一時記憶領域の長さ（アプリケーション属性定義（mcfaalcap -n）の tempsize オペランドの指定値）分だけ(00)₁₆ が埋められます。

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。
72000	SPP では RECEIVE（一時記憶データの受け取り）文を呼び出せません。
72013	一時記憶データ格納バッファ領域の長さ（データ名 C の長さ）を超える一時記憶データを受け取りました。一時記憶データ格納バッファ領域の長さを超える一時記憶データを切り捨てました。
72016	通信文に BEFORE ERASING が設定されています。
72036	受け取り領域の長さ（一意名 1 の長さ）が不足しています。7 バイト以上の領域を確保してください。
72042	通信記述項に SYMBOLIC TERMINAL 句の指定があり、空白以外が設定されています。
72101	継続問い合わせ応答型でないアプリケーションで、RECEIVE（一時記憶データの受け取り）文を呼び出しています。
72106	先頭セグメントを受信する RECEIVE 文を呼び出す前に、RECEIVE（一時記憶データの受け取り）文を呼び出しています。
72107	DISABLE 文を呼び出したあとで、RECEIVE（一時記憶データの受け取り）文を呼び出しています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

SEND – メッセージの送信（データ操作言語）

形式 1（論理端末へ送信するメッセージを設定して、送信要求する場合）

DATA DIVISION（通信記述項）の指定

```
CD 通信記述名
   FOR {OUTPUT|I-0}
   [STATUS KEY IS データ名1]
   [SYMBOLIC TERMINAL IS データ名2]
   [MAP NAME IS データ名5]
   [SYNCHRONOUS MODE IS {ASYNC|データ名6} ]
   [SWITCHING MODE IS {NORMAL|PRIOR|データ名7}]
   [NEXT TRANSACTION IS データ名8]
   [DETAIL MODE IS データ名10]
   [MAPPING MODE IS データ名13] .
```

DATA DIVISION（データ記述項）の指定

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

PROCEDURE DIVISION（通信文）の指定

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

形式 2（先頭セグメントの送信後、メッセージ送信の終了を連絡する場合）

DATA DIVISION（通信記述項）の指定

```
CD 通信記述名
   FOR {OUTPUT|I-0}
   [STATUS KEY IS データ名1]
   [SYMBOLIC TERMINAL IS データ名2]
   [SWITCHING MODE IS {NORMAL|PRIOR|データ名7}] .
```

PROCEDURE DIVISION（通信文）の指定

```
SEND 通信記述名 WITH EMI.
```

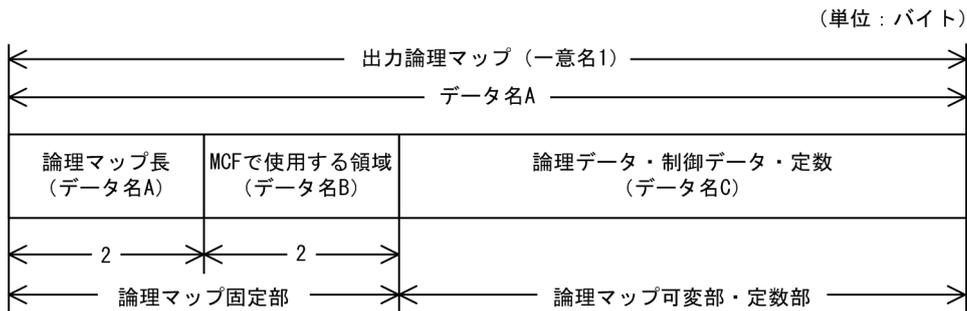
機能

次に示す CALL インタフェースの機能を実現します。

- 応答メッセージの送信 CBLDCMCF('REPLY△△△')
- メッセージの送信 CBLDCMCF('SEND△△△△')

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

セグメントを送信する領域（一意名1で示す領域）の形式を次に示します。



TP1/NET/XMAP3の論理端末に送信する場合は、XMAP3で作成した出力論理マップをCOPY文で取り込み、一意名1として指定します。出力論理マップの詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」、またはマニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド」を参照してください。コーディング例については、「付録 H.1 コーディング例」を参照してください。

UAPで値を設定する項目

●FOR句

次のどちらかの値を設定します。

OUTPUT

一方送信メッセージの送信

I-O

応答メッセージの送信

●SYMBOLIC TERMINAL句

論理端末名称を設定したデータ項目を設定します。データ名2に出力先の論理端末名称を設定します。論理端末名称は最大8バイトの長さです。8バイトに満たない場合、論理端末名称の後ろを空白で埋めてください。

●MAP NAME句

マップ名を設定したデータ項目を設定します。データ名5にマップ名を設定します。マップ名は最大6バイトの長さです。6バイトに満たない場合、マップ名の後ろを空白で埋めてください。

MAP NAME句を省略、または、データ名5に空白を設定した場合は、論理端末定義 (mcftalcle -p) の dflmap オペランドで指定したデフォルトマップ名を使用します。

●SYNCHRONOUS MODE句

非同期型でのメッセージ送信を示す、次のどちらかの値を指定してください。

ASYNCR

非同期型のメッセージの送信

データ名 6

次の値を設定したデータ項目

'0'または'△': 非同期型のメッセージの送信

省略した場合は、ASYNCR（非同期型のメッセージの送信）が設定されます。

●SWITCHING MODE 句

一方送信メッセージの場合に、一般か優先かを設定します。

NORMAL

一般の一方送信メッセージ

PRIOR

優先の一方送信メッセージ

データ名 7

次の値を設定したデータ項目

'0'または'△': 一般の一方送信メッセージ

'1': 優先の一方送信メッセージ

省略した場合、および非応答型のアプリケーションから応答メッセージを送信した場合は、NORMAL（一般の一方送信メッセージ）が設定されます。

●NEXT TRANSACTION 句

<継続問い合わせ応答形態の場合>

次起動アプリケーションを設定したデータ項目を設定します。データ名 8 に次起動アプリケーションを設定します。次起動アプリケーションは最大 8 バイトの長さです。8 バイトに満たない場合、次起動アプリケーションの後ろを空白で埋めてください。

NEXT TRANSACTION 句を省略、または、データ名 8 に空白を設定した場合は、実行中のアプリケーションを次のメッセージ受信時に再び起動します。

応答メッセージの送信を行う SEND（メッセージの送信）文を呼び出すサービスで DISABLE（継続問い合わせ応答の終了）文を呼び出す場合は NEXT TRANSACTION 句を省略、または、データ名 8 に空白を設定してください。

継続問い合わせ応答を引き継いだエラーイベントで応答メッセージの送信を行う SEND（メッセージの送信）文を呼び出す際に、NEXT TRANSACTION 句を省略、または、データ名 8 に空白を設定した場合、継続問い合わせ応答を終了します。ただし、継続問い合わせ応答を引き継いだエラーイベントで、SEND（アプリケーションプログラムの起動）文を呼び出して継続問い合わせ応答型のアプリケーションを起動し、起動先のアプリケーションで応答メッセージの送信を行う SEND（メッセージの送信）文を呼び出す際に、NEXT TRANSACTION 句を省略、または、データ名 8 に空白を設定した場

合、継続問い合わせ応答を終了しないで、SEND（アプリケーションプログラムの起動）文を呼び出したときに設定した継続問い合わせ応答型のアプリケーションを次のメッセージ受信時に再び起動します。

<継続問い合わせ応答形態以外の場合>

NEXT TRANSACTION 句を省略、または、データ名 8 に空白を設定します。

●DETAIL MODE 句

出力通番を付けるかどうかを設定します。データ名 10 に次のどちらかを設定します。

'0'または'△'

出力通番を付けます。

'1'

出力通番を付けません。

省略した場合、および非応答型のアプリケーションから応答メッセージを送信した場合は、出力通番を付けません。

●MAPPING MODE 句

マッピングモードを設定します。データ名 13 に次のどちらかを設定します。

'2'

論理マップだけでマッピングする場合に設定します。

'2'以外

論理マップと物理マップの情報でマッピングする場合に設定します。

省略した場合は、論理マップと物理マップの情報でマッピングします。

●データ名 A

出力論理マップの長さ（論理マップ固定部の長さ+論理マップ可変部・定数部の長さ）を設定します。

●データ名 B

送信完了通知イベントおよび送信障害通知イベントを設定します。データ名 B とビットの設定値の関係を次に示します。

AIX, HP-UX の場合

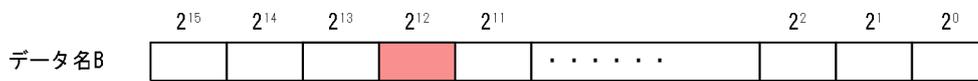


(凡例) : イベントの要否を設定するビット

送信完了通知イベントおよび送信障害通知イベントの要否は、データ名 B の領域の 2⁴ ビットに設定します。

- 領域の 2^4 ビットに 1 を設定すると、送信完了通知イベントおよび送信障害通知イベントを通知させます。ただし、これらのイベントを処理する MHP をアプリケーション属性定義 (mcfaalcap) で指定していない場合は無効です。
- 領域の 2^4 ビットに 0 を設定すると、送信完了イベントおよび送信障害通知イベントを通知させません。

Linux, Windows の場合



(凡例) : イベントの要否を設定するビット

送信完了通知イベントおよび送信障害通知イベントの要否は、データ名 B の領域の 2^{12} ビットに設定します。

- 領域の 2^{12} ビットに 1 を設定すると、送信完了通知イベントおよび送信障害通知イベントを通知させます。ただし、これらのイベントを処理する MHP をアプリケーション属性定義 (mcfaalcap) で指定していない場合は無効です。
メッセージが複数のセグメントで構成される場合、先頭セグメント送信時の指定が有効になります。
- 領域の 2^{12} ビットに 0 を設定すると、送信完了イベントおよび送信障害通知イベントを通知させません。

●データ名 C

送信するセグメントの内容 (論理マップ可変部・定数部に出力する内容) を設定します。

●WITH 句

送信するメッセージのセグメント種別を設定します。

ESI

先頭セグメントの場合に設定します。

EMI

単一セグメントまたは先頭セグメントの送信後にメッセージ送信の終了を連絡する場合に設定します。

一意名 2

次の値を設定したデータ項目

'1': ESI (先頭セグメント)

'2': EMI (単一セグメント)

指定を省略した場合は、EMI (単一セグメント) を設定します。

OpenTP1 から値が返される項目

●STATUS KEY 句

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

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71002	メッセージキューへの出力処理中に障害が発生しました。
	メッセージキューが閉塞されています。
	メッセージキューが割り当てられていません。
	データ名 A に 32004 バイトを超える値を設定しています。
	MCF が終了処理中のため、メッセージの送信を受け付けられません。
71003	メッセージキューが満杯です。
71004	メッセージを格納するバッファをメモリ上に確保できませんでした。
71108	メッセージを送信しようとしたますが、送信先の管理テーブルを確保できませんでした。
	プロセスのローカルメモリが不足しています。
72000	< MHP の実行でリターンした場合 > <ul style="list-style-type: none">先頭セグメントを受信する RECEIVE 文を呼び出す前に、SEND 文を呼び出しています。非応答型のアプリケーションからの問い合わせ応答をしない (UAP 共通定義 (mcfmuap - c) の noansreply オペランドに no を指定) 場合に、非応答型のアプリケーションから応答メッセージを送信しています。
	< SPP の実行でリターンした場合 > <ul style="list-style-type: none">SPP では応答メッセージを送信する SEND 文を呼び出せません。トランザクションでない SPP の処理から、SEND 文を呼び出しています。
72001	SYMBOLIC TERMINAL 句に設定した論理端末名称が間違っています。
	SYMBOLIC TERMINAL 句に設定した論理端末名称は、定義されていません。
	SEND 文を実行できない論理端末を設定しています。
	< 応答メッセージの送信を行う (FOR 句に I-O を設定し、かつ、SYNCHRONOUS MODE 句を省略または SYNCHRONOUS MODE 句に ASYNC を設定または SYNCHRONOUS MODE 句に 0 もしくは空白を設定したデータ名 6 を設定) 場合 > 入力元論理端末は、応答メッセージの送信をサポートしていません。
	SYNCHRONOUS MODE 句に SYNC が設定されているか、データ名 6 に '1' が設定されています。
72005	< WITH 句に ESI を設定または WITH 句に 1 を設定した一意名 2 を設定した場合 > データ名 A に 0 から 4 バイト、またはマイナス値を設定しています。

ステータスコード	意味
72008	<p>応答型のアプリケーションから SEND (アプリケーションプログラムの起動) 文を呼び出して応答型のアプリケーションを起動したあとで、SEND (応答メッセージの送信) 文を呼び出しています。</p> <p>継続問い合わせ応答型のアプリケーションから SEND (アプリケーションプログラムの起動) 文を呼び出して継続問い合わせ応答型のアプリケーションを起動したあとで、SEND (応答メッセージの送信) 文を呼び出しています。</p>
72011	継続問い合わせ応答型でないアプリケーションが、次起動アプリケーションを設定して、SEND 文を呼び出しています。
72017	DETAIL MODE 句に設定した値が間違っています。
72018	SWITCHING MODE 句に設定した値が間違っています。
72020	SYNCHRONOUS MODE 句に設定した値が間違っています。
72024	FOR 句に設定した値が間違っています。
72026	WITH 句に設定した値が間違っています。
72037	通信文に BEFORE 句が設定されています。
72041	<p>< WITH 句を省略または WITH 句に EMI を設定または WITH 句に 2 を設定した一意名 2 を設定した場合 > データ名 A に 0 から 4 バイト、またはマイナス値を設定しています。</p> <p>< メッセージの送信の終了を連絡した場合 > WITH 句に ESI を設定または WITH 句に 1 を設定した一意名 2 を設定した SEND 文を呼び出さないうで、メッセージの送信の終了を連絡しています。</p>
72044	DISABLE 文を呼び出したあとで、次起動アプリケーションを設定して SEND 文を呼び出しています。
72045	NEXT TRANSACTION 句に継続問い合わせ応答型でないアプリケーションのアプリケーション名を設定して応答メッセージを送信する SEND 文を呼び出しています。
72046	NEXT TRANSACTION 句に設定したアプリケーション名が異なる応答メッセージの送信を行う SEND 文を 2 回以上呼び出しています。
72047	NEXT TRANSACTION 句にアプリケーション属性定義 (mcfaalcap) に定義されていないアプリケーション名を設定して応答メッセージを送信する SEND 文を呼び出しています。
73001	メッセージ出力先論理端末に障害が発生しました。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

SEND — 一時記憶データの更新（データ操作言語）

形式

DATA DIVISION（通信記述項）の指定

```
CD 通信記述名
   FOR I-O STORAGE
   [STATUS KEY IS データ名1] .
```

DATA DIVISION（データ記述項）の指定

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

PROCEDURE DIVISION（通信文）の指定

```
SEND 通信記述名 FROM 一意名1.
```

機能

次に示す CALL インタフェースの機能を実現します。

- 一時記憶データの更新 CBLDCMCF('TEMPPUT△')

アプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドには、更新する一時記憶データ長（データ名 A から 6 を減算した値）以上の値を指定してください。

更新する領域（一意名 1 で示す領域）の形式を次に示します。



UAP で値を設定する項目

●FOR 句

一時記憶データの更新であることを示す、I-O STORAGE を設定します。

●データ名 A

一意名 1 の長さ（データ名 A の長さ（2） + データ名 B の長さ（4） + データ名 C の長さ）を設定します。

●データ名 B

MCF で使用する領域です。

●データ名 C

一時記憶データが格納されている領域を設定します。

OpenTP1 から値が返される項目

●STATUS KEY 句

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

ステータスコード

ステータスコード	意味
00000	正常に終了しました。
71103	一時記憶データを更新するための領域をメモリ上に確保できませんでした。
72000	SPP では SEND (一時記憶データの更新) 文を呼び出せません。
72024	FOR 句に設定した値が間違っています。
72035	データ名 A に設定した更新データの長さが、アプリケーション属性定義 (mcfaalcap -n) の tempsize オペランドで定義した長さを超えています。 データ名 A に 0 から 6 バイト、またはマイナス値を設定しています。
72042	通信記述項に SYMBOLIC TERMINAL 句の指定があり、空白以外が設定されています。
72101	継続問い合わせ応答型でないアプリケーションで、SEND (一時記憶データの更新) 文を呼び出しています。
72105	RECEIVE (一時記憶データの受け取り) 文を呼び出す前に、SEND (一時記憶データの更新) 文を呼び出しています。
72106	先頭セグメントを受信する RECEIVE 文を呼び出す前に、SEND (一時記憶データの更新) 文を呼び出しています。
72107	DISABLE 文を呼び出したあとで、SEND (一時記憶データの更新) 文を呼び出しています。
上記以外	プログラムの破壊などによる、予期しないエラーが発生しました。

5

ユーザOWNコーディング，MCF イベントインタフェース

この章では，TP1/NET/XMAP3 に関連するユーザOWNコーディング，およびMCF イベントのインタフェースについて説明します。

5.1 ユーザOWNコーディングインタフェース

メッセージ送受信の UAP を、より多様な業務に対応させるために補助するプログラムを、ユーザOWNコーディング（以降 UOC と略します）と呼びます。独自の処理でメッセージを編集したり、アプリケーション名を決定したりする場合は、ユーザが UOC を作成してください。

TP1/NET/XMAP3 で使用できる UOC を次に示します。

- 入力メッセージ編集 UOC
入力メッセージ編集 UOC は、UAP を呼び出す前に入力メッセージを編集したり、アプリケーション名を決定したりするプログラムです。
- 出力メッセージ編集 UOC
出力メッセージ編集 UOC は、UAP からメッセージが出力されたあとに、出力されたメッセージを編集するプログラムです。
- 送信メッセージ通番編集 UOC
送信メッセージ通番編集 UOC は、受け取った出力通番を基に、ユーザ独自の処理をするプログラムです。

UOC は C 言語で作成します。UOC を使用する場合は、あらかじめ UOC と MCF 提供ライブラリをリンクする必要があります。

5.1.1 入力メッセージの編集とアプリケーション名の決定

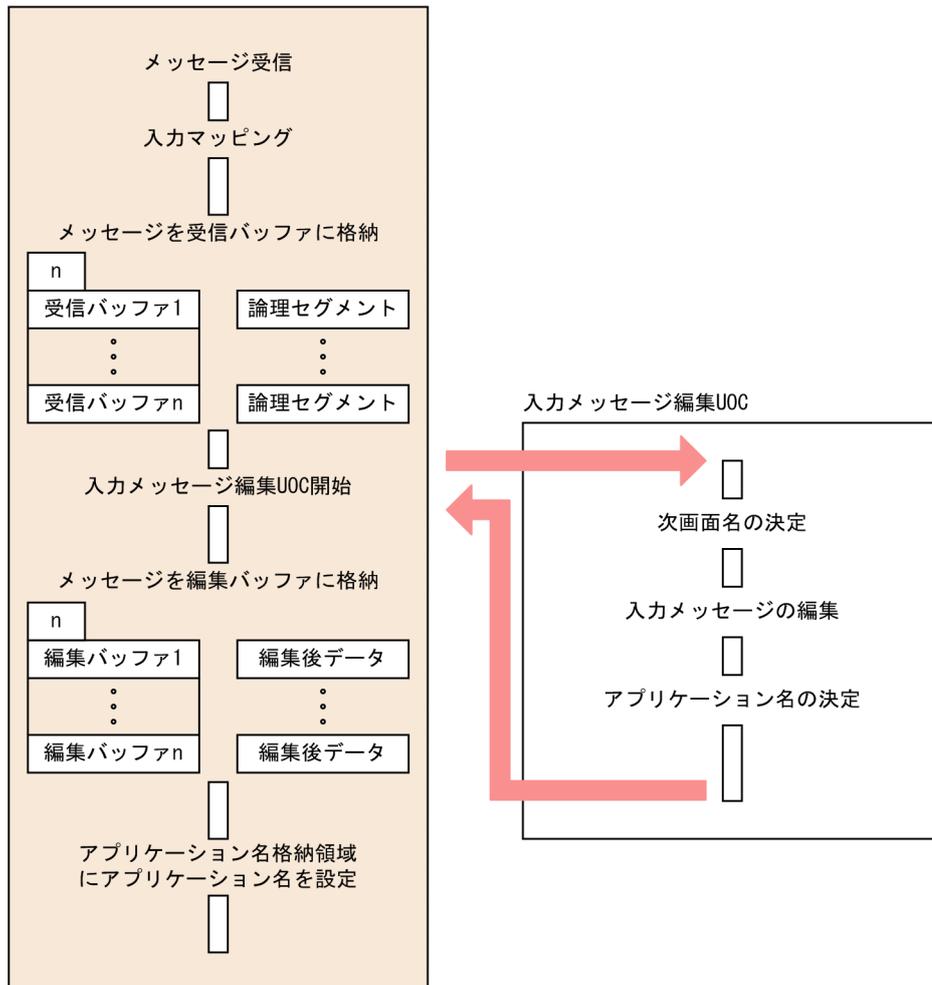
入力メッセージ編集 UOC を作成すると、入力メッセージを編集したり、アプリケーション名を決定したりできます。TP1/NET/XMAP3 は、UAP を起動するメッセージの最終セグメントを受信すると、入力メッセージ編集 UOC を起動します。ただし、MCF イベント発生時と UAP からのアプリケーション起動時は、入力メッセージ編集 UOC を起動しません。また、'C'で始まるオンラインコマンドを受信した場合にも、UOC を起動しません。

ユーザは、MCF メイン関数で UOC 関数アドレスを設定します。また、コネクション定義 (mcftalccn - e) でメッセージ編集用バッファグループ番号と編集バッファ数を定義しておく必要があります。

入力メッセージ編集 UOC の処理の流れを次の図に示します。

図 5-1 入力メッセージ編集 UOC の処理の流れ

TP1/NET/XMAP3



(1) 次画面名の決定

TP1/NET/XMAP3 は、マッピング関数からのリターン情報の次画面名を、次画面名格納領域に設定し、入力メッセージ編集 UOC に渡します。UOC で、次画面名を決定する必要がない場合には、次画面名格納領域に次画面名を設定する必要はありません。次画面名が 6 バイトより長い場合、MCF は次画面名不正としてメッセージログを出力し、該当するメッセージを破棄します。

(2) 入力メッセージの編集

入力メッセージをユーザ任意のメッセージ形式に編集できます。TP1/NET/XMAP3 は、入力メッセージが格納されている受信バッファと、コネクション定義 (mcftalccn -e) で指定した編集バッファを、リスト形式で入力メッセージ編集 UOC に渡します。UOC では、受信バッファと編集バッファを使って入力メッセージを編集します。受信バッファにあるメッセージを UAP に通知するか、編集バッファにあるメッセージを UAP に通知するかは、UOC からのリターンコードで選択します。

(3) アプリケーション名の決定

アプリケーション名格納領域にアプリケーション名を設定することで、ユーザ独自の方法でアプリケーション名を決定できます。メッセージの先頭からアプリケーション名を取り出し、アプリケーション名格納領域に設定します。ユーザ独自の方法でアプリケーション名を決定する必要がない場合には、アプリケーション名格納領域にアプリケーション名を設定する必要はありません。次の表にアプリケーション名格納領域の初期値を示します。

表 5-1 アプリケーション名格納領域の初期値

入力メッセージ編集コール条件	アプリケーション名格納領域の初期値
継続問い合わせ応答中	前回の UAP の dc_mcf_reply 関数で指定された次起動アプリケーション名。UAP が次起動アプリケーション名を省略（ヌル文字を指定）した場合は、その UAP のアプリケーション名を次起動アプリケーション名とします。
上記以外※	TP1/NET/XMAP3 が入力メッセージの先頭から取り出したアプリケーション名。不正なアプリケーション名の場合は、その文字列を最大 9 文字セットします。文字列が 8 文字以内の場合は、文字列のあとにヌル文字 ('¥ 0') が設定されます。

注※

継続問い合わせ応答の初回受信を含みます。

アプリケーション名の形式は、アプリケーション名格納領域の先頭から '¥ 0' の手前までの 1~8 バイトの英数字です。先頭から 9 バイト目までに '¥ 0' がいない場合は、アプリケーション名を不正として、ERREVT1 を起動し、受信メッセージを破棄します。

(4) UOC エラーリターン処理

UOC から DCMCF_UOC_MSG_NG でリターンした場合、TP1/NET/XMAP3 は該当メッセージを破棄し、メッセージログを出力します。また、その内容をオペレータインジケータに表示します。この場合、MCF イベントは起動しません。

(5) OpenTP1 への組み込み方法

スタート関数 (dc_mcf_svstart) を発行する MCF メイン関数に、作成した UOC の関数アドレスを設定します。入力メッセージ編集 UOC の関数アドレスは任意に求められます。UOC のオブジェクトファイルは、スタート関数を実行する MCF メイン関数をコンパイル・リンケージすれば、TP1/NET/XMAP3 の実行形式ファイルに結合されて実行できる状態になります。スタート関数の詳細については、「[10.2 MCF メイン関数の作成](#)」を参照してください。

5.1.2 入力メッセージ編集 UOC インタフェース

入力メッセージ編集 UOC は、次に示す形式で呼び出します。

(1) 形式

ANSI C, C++の形式

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

K&R 版 C の形式

```
#include <dcmfuoc.h>
#include <dcmxpn.h>
DCLONG uoc_func(parm)

dcmcf_uoc_min_n *parm;
```

(2) 説明

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

(3) パラメタの内容

(a) dcmcf_uoc_min_n の内容

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

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

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

(c) dcmcf_uocbufinf_n (バッファ情報) の内容

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

(d) dcmxp_min_ifa (プロトコル個別インタフェース) の内容

```
typedef struct {
    char map_name[9];       …表示中マップ名
    char reserve1[7];      …予備
    char next_map_name[9]; …次画面名
    char reserve2[7];      …予備
    char device[8];        …デバイス
    char cont_status;      …継続問い合わせ応答ステータス
    char reserve3[23];     …予備
} dcmxp_min_ifa;
```

(4) MCF が設定する項目

(a) dcmcf_uoc_min_n

- pro_kind
プロトコル種別として、次の値が設定されます。
DCMCF_UOC_PRO_XP
TP1/NET/XMAP3 プロトコル
- le_name
メッセージを入力した論理端末の名称が設定されます。
- rcv_prim
受信サービスプリミティブとして、次の値が設定されます。
DCMCF_UOC_RCV_USELESS
意味なし
- 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 (コネクション定義 (mcftalccn) の-e オプションを省略した場合)、ebuflist_adr には NULL が設定されます。

(5) ユーザが値を設定する項目

(a) dcmcf_uoc_min_n

- aplname

UOC で決定したアプリケーション名を設定します。

- rtn_detail

詳細リターンコードを設定します。

このコードは、UOC が DCMCF_UOC_MSG_NG でリターンしたときに、MCF がメッセージログファイルに出力します。

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

(b) dcmcf_uocbuff_list_n (バッファリスト)

- used_buf_num

使用したバッファ情報の数を設定します。

DCMCF_UOC_MSG_OK_RCV でリターンする場合は、受信バッファリストの used_buf_num を必ず設定してください。

アプリケーション名だけを決定し、メッセージ編集をしない場合や、セグメント数を変更しない場合は、受信バッファリストの buf_num の値を設定してください。

(c) dcmcf_uocbufinf_n (バッファ情報)

- seg_size

バッファの使用長を設定します。

(d) dcmxp_min_ifa (プロトコル個別インタフェース)

- next_map_name

次画面の名称を設定します。

(6) リターン値

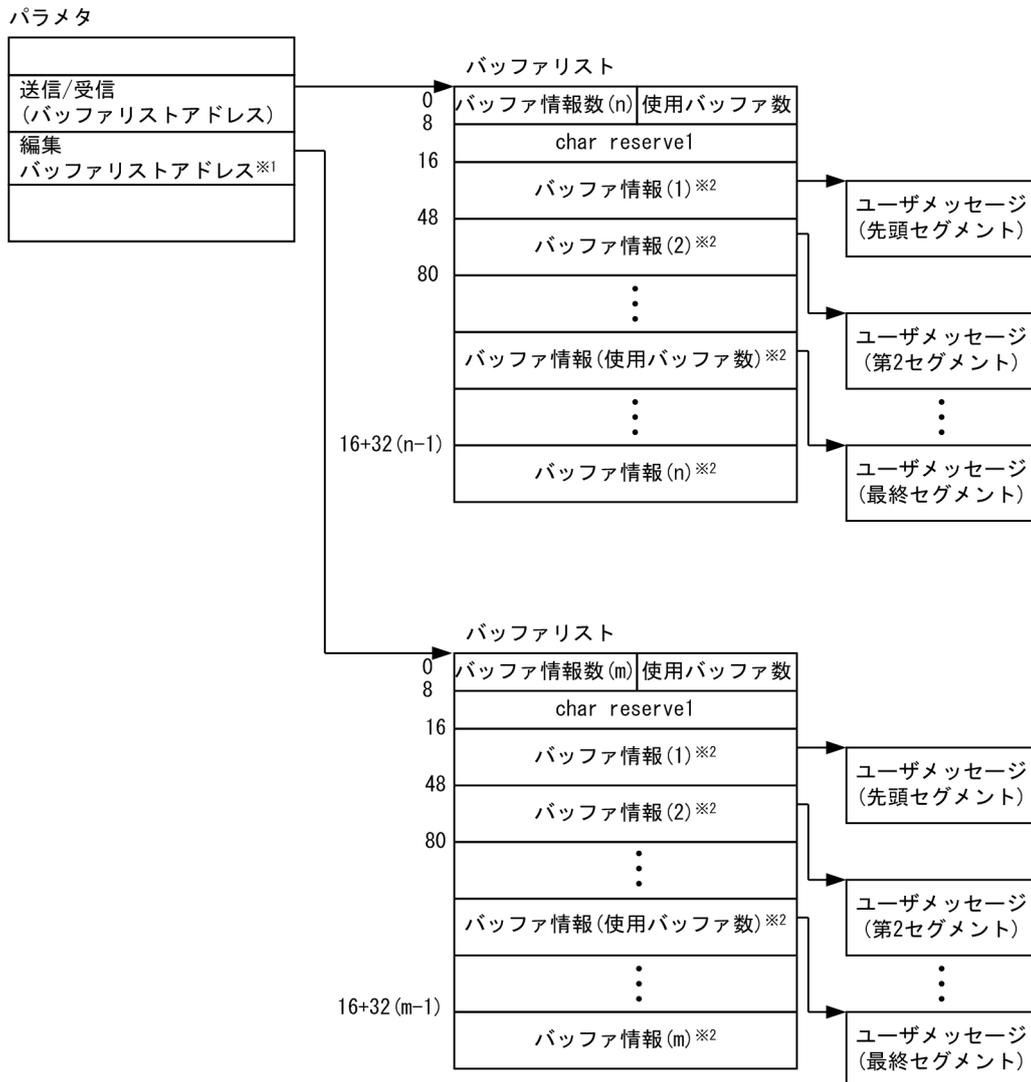
uoc_func()は次のコードでリターンしてください。

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

(7) パラメタとバッファの関係

UOC インタフェース用パラメタとバッファの関係を次の図に示します。

図 5-2 UOC インタフェース用のパラメタとバッファの関係



注※1

コネクション定義 (mcftalccn) の -e オプションを指定しなければ NULL となり、バッファリストとバッファは確保されません。

注※2

バッファ情報は 32 バイトで次の形式をしています。

バッファ情報	
0	バッファアドレス
4	バッファ最大長
8	バッファ使用長
12	予備
16	MCF内部情報
28	予備
32	

5.1.3 出力メッセージの編集

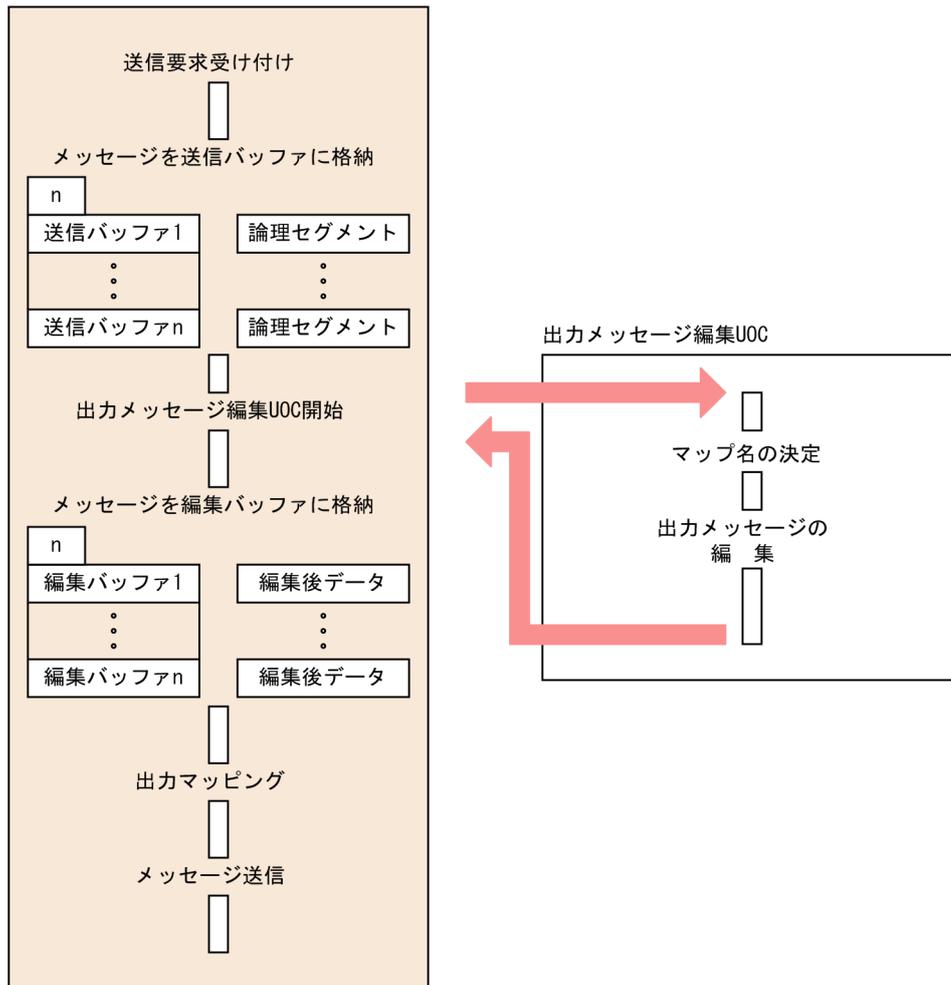
出力メッセージ編集 UOC を作成すると、出力メッセージのメッセージ形式を編集できます。TP1/NET/XMAP3 は、送信するメッセージの最終セグメントを出力キューから取り出すと、出力メッセージ編集 UOC を起動します。

ユーザは、MCF メイン関数で UOC 関数アドレスを設定します。また、コネクション定義 (mcftalccn - e) でメッセージ編集用バッファグループ番号と編集バッファ数を定義しておく必要があります。

出力メッセージ編集 UOC の処理の流れを次の図に示します。

図 5-3 出力メッセージ編集 UOC の処理の流れ

TP1/NET/XMAP3



(1) マップ名の決定

要求マップのパラメタの内容を変更して、実際の表示・印刷に使用するマップ名を独自の方法で決定できます。

マップ名はマップ名称格納領域の先頭から'¥ 0'の手前までの1~6バイトの英数字です。先頭から7バイト目までに'¥ 0'がない場合は、マップ名不正となります。また、マップ名として"THROUGH"、および"SYSMSG"を使用してはいけません。

TP1/NET/XMAP3は、ここで返されたマップ名を表示マップ名として管理します。独自の方法でマップ名を決定しない場合、パラメタの内容をそのままにしておきます。

マップ名不正の場合、MCFはメッセージログを出力し、障害通知イベント（VERREVT）を通知します。また、該当するメッセージが応答メッセージの場合は、破棄します。

(2) 出力メッセージの編集

出力メッセージをユーザ任意のメッセージ形式に編集できます。TP1/NET/XMAP3 は、出力マッピング前の出力メッセージが格納されている送信バッファと、コネクション定義 (mcftalccn -e) で指定した編集バッファを、リスト形式で出力メッセージ編集 UOC に渡します。UOC では、送信バッファと編集バッファを使って、メッセージ形式を変換するなど、出力メッセージを編集します。送信バッファに格納されているメッセージを端末に送信するか、編集バッファに格納したメッセージを端末に送信するかは、UOC からのリターンコードで選択します。

(3) UOC インタフェース用のパラメタとバッファの関係

UOC インタフェース用のパラメタとバッファの関係は、入力メッセージ編集 UOC の場合と同じです。UOC インタフェース用のパラメタとバッファの関係については、[5.1.2(7) パラメタとバッファの関係] を参照してください。

(4) UOC エラーリターン処理

UOC から DCMCF_UOC_MSG_NG で終了した場合、TP1/NET/XMAP3 は該当メッセージを破棄し、メッセージログを出力します。破棄したメッセージが応答メッセージの場合は、オペレータインジケータにその内容を表示します。なお、この場合 MCF イベントは起動しません。

(5) OpenTP1 への組み込み方法

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

5.1.4 出力メッセージ編集 UOC インタフェース

出力メッセージ編集 UOC は、次に示す形式で呼び出します。

(1) 形式

ANSI C, C++の形式

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

K&R 版 C の形式

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

dcmcf_uoc_mout_n *parm;
```

(2) 説明

uoc_func（出力メッセージ編集 UOC）を呼び出す場合、MCF は次に示すパラメタを parm に設定します。

(3) パラメタの内容

(a) dcmcf_uoc_mout_n の内容

```
typedef struct {
    DCLONG pro_kind;           …プロトコル種別
    char le_name[9];          …論理端末名称
    char reserve1[7];         …予備
    dcmcf_uocbuff_list_n *buflist_adr;
                               …送信バッファリストアドレス
    dcmcf_uocbuff_list_n *ebuflist_adr;
                               …編集バッファリストアドレス
    DCLONG output_no;         …メッセージ出力通番
    char msg_type;            …メッセージ種別
    char outputno_flag;       …メッセージ出力通番有効フラグ
    char resend_flag;         …再送メッセージフラグ
    char reserve2[1];         …予備
    char *pro_indv_ifa;
                               …プロトコル個別インタフェース領域
    DCLONG rtn_detail;        …詳細リターンコード
    char reserve3[20];        …予備
} dcmcf_uoc_mout_n;
```

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

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

(c) dcmcf_uocbufinf_n (バッファ情報) の内容

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

(d) dcmxp_mout_ifa (プロトコル個別インタフェース) の内容

```
typedef struct {
    char map_name[9];         …要求マップ名
    char reserve1[7];         …予備
    char print_map_name[9];   …表示中マップ名
    char reserve2[7];         …予備
    char device[8];          …デバイス
    char *inf_adr;            …MCF使用領域
    char reserve3[20];        …予備
} dcmxp_mout_ifa;
```

デバイスについては、[5.1.2 入力メッセージ編集 UOC インタフェース] を参照してください。

(4) MCF が設定する項目

(a) dcmcf_uoc_mout_n

- **pro_kind**
プロトコル種別として、次の値が設定されます。
DCMCF_UOC_PRO_XP
TP1/NET/XMAP3 プロトコル
- **le_name**
UAP で指定した送信先の論理端末の名称が設定されます。
- **buflist_adr**
送信用バッファリストのアドレスが設定されます。
- **ebuflist_adr**
編集用バッファリストのアドレスが設定されます。
メッセージ編集バッファが未定義の場合（コネクション定義（mcftalccn）の-e オプションを省略した場合）、ebuflist_adr には NULL が設定されます。
- **output_no**
メッセージ出力通番が設定されます。ただし、outputno_flag が DCMCF_UOC_OUTPUTNO_OK のときだけ有効です。
- **msg_type**
メッセージ種別が設定されます。
'o'
問い合わせ応答メッセージ
'n'
一般一方送信メッセージ
'p'
優先一方送信メッセージ
- **outputno_flag**
メッセージ出力通番の有効フラグとして、次のどちらかの値が設定されます。
DCMCF_UOC_OUTPUTNO_OK
メッセージ出力通番を有効にします。
DCMCF_UOC_OUTPUTNO_NG
メッセージ出力通番を無効にします。
- **resend_flag**
再送フラグとして、次のどちらかの値が設定されます。

'r'

再送メッセージです。

'n'

再送メッセージではありません。

- **pro_indv_ifa**

プロトコル個別インタフェース領域のアドレスが設定されます。

(b) dcmcf_uocbuff_list_n (バッファリスト)

- **buf_num**

バッファ情報の数が設定されます。

(c) dcmcf_uocbufinf_n (バッファ情報)

- **buf_adr**

バッファのアドレスが設定されます。

- **buf_size**

バッファの最大長が設定されます。

- **seg_size**

送信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。

(d) dcmxp_mout_ifa (プロトコル個別インタフェース)

- **map_name**

要求するマップ名称が設定されます。

- **print_map_name**

現在表示中の、または前回表示・印刷したマップ名称が設定されます。

- **device**

機種またはモデルが設定されます。代行中は代行先の機種またはモデルが設定されます。

- **inf_adr**

MCF 使用領域です。

(5) ユーザが値を設定する項目

(a) dcmcf_uoc_mout_n

- **rtn_detail**

詳細リターンコードを設定します。

このコードは、UOC が DCMCF_UOC_MSG_NG でリターンした場合に、MCF がメッセージログファイルに出力します。

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

(b) dcmcf_uocbuff_list_n (バッファリスト)

- used_buf_num

使用したバッファ情報の数を設定します。

DCMCF_UOC_MSG_OK_SND でリターンする場合は、送信バッファリストの used_buf_num を必ず設定してください。セグメント数を変更しない場合は、送信バッファリストの buf_num の値を設定してください。

(c) dcmcf_uocbufinf_n (バッファ情報)

- seg_size

バッファの使用長を設定します。

(d) dcmxp_mout_ifa (プロトコル個別インタフェース)

- map_name

要求するマップ名称を設定します。

(6) リターン値

uoc_func()は次のコードでリターンしてください。

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

5.1.5 送信メッセージの通番編集

送信メッセージ通番編集 UOC を作成すると、受け取った出力通番を基に、ユーザ独自の処理ができます。

(1) 出力通番編集

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

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

(2) OpenTP1 への組み込み方法

UAP のメイン関数の中に、UOC の関数アドレスを登録しておきます。UAP のメイン関数に登録する dc_mcf_register 関数の形式を次に示します。

(a) 形式

ANSI C, C++の場合

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

K&R 版 C の場合

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

(b) ユーザが値を設定する引数

- flags
DCMCF_SEND_UOC を指定します。
- uoc_addr
flags に対応する UOC のアドレスを指定します。

(c) リターン値

dc_mcf_register() のリターン値を次に示します。

リターン値	意味
DC_OK	正常リターン
DCMCFER_INVALID_ARGS	引数の指定誤り
DCMCFER_NOMEM	ローカルメモリ不足

(d) メイン関数への登録例

- MHP の場合

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

dc_rpc_open();
dc_mcf_open();
dc_mcf_register(DCMCF_SEND_UOC, send_uoc);
```

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

- SPP の場合

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

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

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

5.1.6 送信メッセージ通番編集 UOC インタフェース

送信メッセージ通番編集 UOC は、次に示す形式で、send_uoc 関数として作成します。なお、UOC の関数名称はユーザの任意です。

(1) 形式

ANSI C, C++の場合

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

K&R 版 C の場合

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

DCLONG flags;
char *termname;
DCLONG sendno;
DCLONG sendid;
DCLONG dataleng;
char *senddata;
```

(2) MCF が値を設定する項目

- flags

送信メッセージ通番編集 UOC がいつ呼び出されたかが設定されます。

DCMCF_SEND_DML

メッセージを送信する関数または命令文が呼び出されたとき

DCMCF_RESEND_DML

メッセージを再送する関数または命令文が呼び出されたとき

- **termname**

送信先の論理端末名称が設定されます。

- **sendno**

送信メッセージの出力通番が設定されます。

- **sendid**

送信するメッセージ種別が設定されます。

DCMCF_SEND_PRIO

優先の一方送信メッセージ

DCMCF_SEND_NORM

一般の一方送信メッセージ

DCMCF_SEND_IO

問い合わせ応答メッセージ

- **dataleng**

送信メッセージ長が設定されます。

- **senddata**

セグメントの内容が設定されます。

(3) リターン値

send_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
DC_OK	正常リターン

5.1.7 UOC 作成上の注意事項

UOC 作成上の注意事項を次に示します。

(1) UOC の構造

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

(2) UOC で使用できる関数

UOC を作成する場合、UOC では次に示す関数だけを使用できます。ほかの関数を使用した場合、TP1/NET/XMAP3 の動作に影響を与えるおそれがあるため、使用しないでください。

- メモリ操作をする関数
 - データ領域管理関数 (malloc/free)
 - 共有メモリ管理関数/システムコール
 - メモリ操作 (memory)
 - 文字列操作 (string)
- 時間取得関数 (例: time, gettimeofday)

(3) UOC の異常処理

TP1/NET/XMAP3 の UOC で異常を検知した場合、MCF の所定のリターンコードを使用して MCF に異常の発生を通知してください。UOC でプロセス終了となるシグナル、または、abort() を発行すると、MCF が異常終了します。

(4) UOC の実行タイミング

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

5.2 MCF イベントインタフェース

TP1/NET/XMAP3 でメッセージ送受信をすると、OpenTP1 の各種システム情報が MHP に通知されます。これを MCF イベントと呼びます。この節では、TP1/NET/XMAP3 が通知する MCF イベントについて説明します。

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

5.2.1 MCF イベントの種類

TP1/NET/XMAP3 が通知する MCF イベントを次の表に示します。

表 5-2 MCF イベント一覧

MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP での処理の例
不正アプリケーション名検出通知イベント	ERREVT1	メッセージのアプリケーション名が MCF アプリケーション定義にありません。	該当するアプリケーションがないことを報告します。
メッセージ廃棄通知イベント	ERREVT2	次の理由で、受信メッセージを廃棄しました。 <ul style="list-style-type: none">入力キューに障害が発生しました。入力メッセージ最大格納数を超過しました。動的共用メモリが不足しました。キューファイルが満杯になりました。MHP のサービス、サービスグループ、またはアプリケーションが閉塞しています。スケジュール閉塞されているサービスグループの入力キューに未処理受信メッセージが残った状態で、OpenTP1 を正常終了または計画停止 A で終了しました。MHP のサービスグループ、またはアプリケーションがセキュア状態です。MHP で呼び出す receive 関数にセグメントを渡す前に、MHP が異常終了しました。アプリケーション名に対応する MHP のサービスがありません。ユーザサーバ未起動などによって、MHP の起動に失敗しました。DBMS の障害などによって、トランザクションの開始に失敗しました。	メッセージを廃棄したことを報告します。 問い合わせメッセージの場合は、応答メッセージを出力できます。
UAP 異常終了通知イベント	ERREVT3	MHP のセグメント受信関数にセグメントを渡したあとに、MHP の異常終了が発生しました。	UAP 異常終了時の対処障害メッセージを送信します。

MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP での処理の例
UAP 異常終了通知イベント	ERREVT3	MHP のセグメント受信関数にセグメントを渡したあとに、MHP の異常終了が発生しました。	問い合わせメッセージの場合は、応答メッセージを出力できます。
未処理送信メッセージ廃棄通知イベント	ERREVT4	次の理由で、未処理送信メッセージを破棄しました。 <ul style="list-style-type: none"> • MCF の正常終了処理時に、未処理送信メッセージの滞留時間監視の時間切れ（タイムアウト）が発生しました。 • 運用コマンド (mcftdlqle) の入力、または API (dc_mcf_tdlqle 関数もしくは CBLDCMCF('TDLQLE△△')) の発行によって、出力キューが削除されました。 • 閉塞されている論理端末の出力キューに未処理送信メッセージが残った状態で、dcstop コマンドが実行されました。 	未処理送信メッセージを廃棄したことを報告します。
送信完了通知イベント	SCMPEVT	メッセージの送信を正常に完了しました。	送信の完了を確認して、任意の処理ができます。
送信障害通知イベント	SERREVT	メッセージの送信時に障害が発生しました。	送信障害が発生したことを報告して、任意の処理ができます。
障害通知イベント	VERREVT	次の理由で論理端末画面を廃棄しました。 <ul style="list-style-type: none"> • 運用コマンド (mcftdctcn) の入力、または API (dc_mcf_tdctcn 関数) の発行が行われました。 • XMAP3 Server 障害が発生しました。 • 内部イベントの送受信で障害が発生しました。 • 継続問い合わせ応答終了処理で障害が発生しました。 • メッセージ受信で障害が発生しました。 • 問い合わせ応答終了処理で障害が発生しました。 • メッセージ送信で障害が発生しました。 • 入力通番チェック処理で障害が発生しました。 • キーボードロック解除処理で障害が発生しました。 • プリンタ排他解除で障害が発生しました。 • プリンタ排他リトライで障害が発生しました。 	論理端末画面の廃棄を報告します。
状態通知イベント	VOPNEVT	論理端末画面の生成、および初期画面表示が完了しました。	編集モードで論理端末定義 (mcftalcle -p) の initmap オペランドを指

MCF イベント名	MCF イベントコード	発生した原因	MCF イベント処理用 MHP での処理の例
状態通知イベント	VOPNEVT	論理端末画面の生成、および初期画面表示が完了しました。	定していない場合は、必ず初期画面を出力します。
	VCLSEVT	次の理由で論理端末画面を廃棄しました。 <ul style="list-style-type: none"> 運用コマンド (mcftdctle) の入力、または API (dc_mcf_tdctle 関数) の発行が行われました。 XMAP3 Server と XMAP3 の接続が切断されました。 	論理端末画面の廃棄を報告します。

5.2.2 MCF イベント通知時のセグメント構成

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

MCF イベント通知時のセグメント構成を次の図に示します。

図 5-4 MCF イベント通知時のセグメント構成



MCF イベントは、作成した UAP が C 言語の場合と COBOL 言語の場合で、UAP に通知されるデータの形式が異なります。それぞれの言語の場合について示します。

5.2.3 MCF イベント情報の形式 (C 言語)

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

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

MCF イベントとして設定される項目を次に示します。

(1) MCF イベントの共通ヘッダ

(a) 形式

```
struct dc_mcf_evtheader {
    char mcfevt_name[9];           ... MCFイベントコード
    char le_name[16];             ... 入力元論理端末名称
                                  (ERREVT1, ERREVT2, ERREVT3の場合)
                                  出力先論理端末名称
                                  (ERREVT4, SCMPEVT, SERREVTの場合)
                                  障害発生論理端末名称
                                  (VERREVTの場合)
                                  該当事象発生論理端末名称
                                  (VOPNEVT, VCLSEVTの場合)
    char cn_name[9];             ... コネクション名
    unsigned char format_kind;    ... MCF使用領域
    char reserve01;              ... 予備
    DCLONG time;                 ... メッセージ入力時刻
};
```

(b) MCF イベントとして設定される項目

• le_name

メッセージを入力した論理端末名称が設定されます。

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

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

ERREVT4, SCMPEVT では、メッセージを出力する論理端末名称が設定されます。

SERREVT では、送信障害となった出力先論理端末名称が設定されます。

VERREVT では、障害の発生した論理端末名称が設定されます。コネクション障害時は無効です。ヌル文字が設定されている場合は無効と見なします。

VOPNEVT では、論理端末画面を生成した論理端末名称が設定されます。

VCLSEVT では、論理端末画面を廃棄した論理端末名称が設定されます。

• cn_name

コネクション名が設定されます。

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

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

• time

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

(2) ERREVT1

(a) 形式

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

(b) MCF イベントとして設定される項目

- ap_name
次に示すどれかが設定されます。
 - 不正となったアプリケーション名
 - 定義されていないアプリケーション名

(3) ERREVT2

(a) 形式

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

(b) MCF イベントとして設定される項目

- ap_name
エラーになった UAP のアプリケーション名が設定されます。
- reason_code
ERREVT2 の理由コードが設定されます。理由コードの詳細については、「付録」 理由コード一覧」を参照してください。

(4) ERREVT3

(a) 形式

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

(b) MCF イベントとして設定される項目

- map_name

XMAP3 のドロウで指定した次画面名（マップ名）が返されます。

なお、次に示す場合は、ヌル文字が設定されます。

- 次画面名を指定していない場合
- アプリケーション起動機能で起動された MHP で障害が発生した場合
- 上記の障害発生時に、MCF イベントとして起動された MHP によって、さらにアプリケーション起動機能で起動された MHP で、障害が発生した場合

- ap_name

異常が発生した MHP のアプリケーション名が設定されます。

- service_name

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

- serv_grp_name

異常が発生した MHP のサービスグループ名が設定されます。

- bid

トランザクションのブランチ ID が次の形式で設定されます。

OpenTP1システム ノードID (char型配列)	トランザクション ブランチ番号 (unsigned int型)	OpenTP1使用領域	予備領域
← 8バイト →	← 4バイト →	← 4バイト →	← 20バイト →

(5) ERREVT

(a) 形式

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

(b) MCF イベントとして設定される項目

- map_name

出力マップ名が設定されます。send 関数で省略した場合は、ヌル文字が設定されます。

- ap_name

正常終了したメッセージのアプリケーション名が設定されます。

MHP から送信されたメッセージの場合に設定されます。MHP 以外から送信された場合は、ヌル文字が設定されます。

(6) SCMPEVT

(a) 形式

```
struct dc_mcf_scmpevt_type {
    struct dc_mcf_evtheader evtheader ; ... MCFイベント共通ヘッダ
    DCLONG output_no ; ... 出力通番
    char map_name[9] ; ... マップ名
    char msg_type ; ... メッセージ種別
    char reason_code ; ... 起動理由
    char reserve01[5] ; ... 予備
    DCLONG user_leng ; ... 他プロトコルの場合の使用領域
    char pro_indv_inf[16] ; ... プロトコル個別情報
    char user_data[16] ; ... 他プロトコルの場合の使用領域
};
```

プロトコル個別情報の形式を次に示します。

```
struct dcmxpn_sevtpro_type {
    DCLONG serr_reason1 ; ... 理由コード1
    DCLONG serr_reason2 ; ... 理由コード2
};
```

```

DCLONG serr_detail ;          ... 詳細コード
char reserve1[4] ;          ... 予備
};

```

なお、dcmxpn_sevtpro_type は<dcmxpn.h>で定義されています。

(b) MCF イベントとして設定される項目

- output_no
出力通番が設定されます。出力通番がない場合は、-1 が設定されます。
- map_name
出力マップ名が設定されます。send 関数で省略した場合は、ヌル文字が設定されます。
- msg_type
メッセージ種別として、次の値が設定されます。
o：応答送信メッセージ
n：一般一方送信メッセージ
p：優先一方送信メッセージ
- reason_code
起動理由として次の値が設定されます。
△：メッセージが正常に送信されました。
- pro_indv_inf
ヌル文字が設定されます。

(7) SERREVT

(a) 形式

```

struct dc_mcf_serrevert_type {
    struct dc_mcf_evtheader evtheader ; ... MCFイベント共通ヘッダ
    DCLONG output_no ;                ... 出力通番
    char map_name[9] ;                ... マップ名
    char msg_type ;                    ... メッセージ種別
    char send_error_code ;            ... 送信エラーコード
    unsigned char trouble_code ;      ... 障害コード
    unsigned char detail_info ;       ... 詳細情報
    unsigned char error_code ;        ... エラーコード
    unsigned char error_subcode ;     ... エラーサブコード
    char reserve01[1] ;                ... 予備
    DCLONG user_leng ;                ... 他プロトコルの場合の使用領域
    char pro_indv_inf[16] ;           ... プロトコル個別情報
    char user_data[16] ;              ... 他プロトコルの場合の使用領域
};

```

プロトコル個別情報の形式を次に示します。

```

struct dcmxpn_sevtpro_type {
    DCLONG serr_reason1 ;          ... 理由コード1
    DCLONG serr_reason2 ;          ... 理由コード2
    DCLONG serr_detail ;          ... 詳細コード
    char reserve1[4] ;            ... 予備
};

```

なお、dcmxpn_sevtpro_type は<dcmxpn.h>で定義されています。

(b) MCF イベントとして設定される項目

- output_no
出力通番が設定されます。出力通番がない場合は、-1 が設定されます。
- map_name
出力マップ名が設定されます。send 関数で省略した場合は、ヌル文字が設定されます。
- msg_type
メッセージ種別として、次の値が設定されます。
o：応答送信メッセージ
n：一般一方送信メッセージ
p：優先一方送信メッセージ
- send_error_code
送信エラーコードとして、'E'が設定されます。
- trouble_code
障害コードとして、次の値が設定されます。
DCMCF_SERR_SEND：送信エラー
DCMCF_SERR_LOGICAL：論理エラー
- detail_info
詳細情報として、次に示す値が設定されます。
DCMCF_SERR_ERROR：エラーによる送信障害
DCMCF_SERR_DCTCN：運用コマンド (mcftdctn) による送信障害
DCMCF_SERR_DCTLE：運用コマンド (mcftdctle) による送信障害
- error_code
ヌル文字が設定されます。
- error_subcode
メッセージの状態として、次の値が設定されます。
DCMXP_N_SEVT_QUE：メッセージ保持
DCMXP_N_SEVT_DEL：メッセージ消去
- pro_indv_inf

detail_info が DCMCF_SERR_ERROR 以外の場合、ヌル文字が設定されます。

detail_info が DCMCF_SERR_ERROR の場合、次の項目を設定します。

- serr_reason1

理由コード 1 として、次に示す値が設定されます。理由コード 1 の詳細については、「付録 J 理由コード一覧」参照してください。

DCMXP_N_RSN1_MCF : MCF 障害

DCMXP_N_RSN1_XPW : XMAP3 Server 障害

DCMXP_N_RSN1_PRT : プリンタ障害

DCMXP_N_RSN1_INVL : 無効メッセージ

- serr_reason2

理由コード 1 が DCMXP_N_RSN1_INVL 以外の場合、理由コード 2 が設定されます。理由コード 2 の詳細については、「付録 J 理由コード一覧」参照してください。

- serr_detail

マッピングサービスの障害コードが設定されます。プリンタ送信時、ほかのメッセージのマッピング障害によって、送信障害通知されることがありますが、障害コードが設定されるのは、マッピング障害となったメッセージだけです。障害コードについては、「付録 K マッピングサービスの障害コードの詳細」を参照してください。

(8) VERREVT

(a) 形式

```
struct dcmxpn_verrevt_type {
    struct dc_mcf_evtheader header ;    ... MCFイベント共通ヘッダ
    DCLONG le_kind ;                  ... 論理端末属性
    DCLONG err_fact ;                 ... 障害要因コード
    DCLONG err_reason1 ;              ... 理由コード1(4バイト)
    DCLONG err_reason2 ;              ... 理由コード2(4バイト)
    DCLONG err_detail ;               ... 詳細コード(4バイト)
    char reserve1[36] ;               ... 予備
};
```

(b) MCF イベントとして設定される項目

- le_kind

論理端末の属性として、次の値が設定されます。

(00000001)₁₆ : 画面

(00000002)₁₆ : プリンタ

- err_fact

障害の要因として、次の値が設定されます。

DCMXP_N_PLACE_CN : コネクション障害

DCMXPN_PLACE_LE：論理端末障害

- err_reason1

理由コード 1 が設定されます。理由コード 1 の詳細については、「付録 J 理由コード一覧」を参照してください。

- err_reason2

理由コード 2 が設定されます。理由コード 2 の詳細については、「付録 J 理由コード一覧」を参照してください。

- err_detail

マッピング障害の場合、障害コードが設定されます。障害コードの詳細については、「付録 K マッピングサービスの障害コードの詳細」を参照してください。

(9) VOPNEVT, VCLSEVT

(a) 形式

```
struct dcmxpn_statevt_type {
    struct dc_mcf_evtheader header ;    ... MCFイベント共通ヘッダ
    DCLONG le_kind ;                  ... 論理端末属性
    char reserve1[52] ;                ... 予備
};
```

(b) MCF イベントとして設定される項目

- le_kind

論理端末の属性として、次の値が設定されます。

(00000001)₁₆：画面

(00000002)₁₆：プリンタ

5.2.4 MCF イベント情報の形式 (COBOL 言語)

COBOL 言語の場合はセグメントの並びとして渡されます。

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

表 5-3 COBOL 言語の MCF イベント情報の内容 (ERREVT1)

項目	位置 (バイト)	長さ (バイト)	属性	内容
エラーイベントコード	0	3	英数字	'ERR'が設定されます。
	3	3	—	—

項目	位置 (バイト)	長さ (バイト)	属性	内容
エラーイベントコード	6	2	英数字	ERREVT1 を示す'1△'が設定されます。
入力元論理端末名称	8	8	英数字	メッセージを入力した論理端末名称が設定されます。
予備	16	20	—	—
アプリケーション名	36	8	英数字	次に示すどれかが設定されます。 <ul style="list-style-type: none"> 形式不正のアプリケーション名 定義されていないアプリケーション名
予備	44	8	—	—
予備	52	8	—	—
予備	60	8	—	—
コネクション名	68	8	英数字	コネクション名が設定されます。
予備	76	16	—	—
メッセージが入力された日付	92	8	外部 10 進数字	端末入力メッセージを入力した日付が設定されます。 「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	100	8	外部 10 進数字	端末入力メッセージを入力した時刻が設定されます。 「hhmmss00」の形式です。 hh：時 mm：分 ss：秒 00 は固定です。
予備	108	16	—	—

(凡例)

—：該当しません。または、使用されません。

表 5-4 COBOL 言語の MCF イベント情報の内容 (ERREVT2)

項目	位置 (バイト)	長さ (バイト)	属性	内容
エラーイベントコード	0	3	英数字	'ERR'が設定されます。
	3	3	—	—

項目	位置 (バイト)	長さ (バイト)	属性	内容
エラーイベントコード	6	2	英数字	ERREVT2 を示す'2△'が設定されます。
入力元論理端末名称	8	8	英数字	メッセージを入力した論理端末名称が設定されます。 次に示す場合は、'*'が設定されます。 <ul style="list-style-type: none"> SPP からアプリケーション起動機能で起動した MHP で障害が発生した場合 上記の障害が発生したあとに、MCF イベントとして起動した MHP からさらにアプリケーション起動機能で起動した MHP で、障害が発生した場合
予備	16	20	—	—
アプリケーション名	36	8	英数字	エラーになった UAP のアプリケーション名が設定されます。
予備	44	8	—	—
予備	52	8	—	—
予備	60	8	—	—
コネクション名	68	8	英数字	コネクション名が設定されます。 次に示す場合は、'*'が設定されます。 <ul style="list-style-type: none"> SPP からアプリケーション起動機能で起動した MHP で障害が発生した場合 上記の障害が発生したあとに、MCF イベントとして起動した MHP からさらにアプリケーション起動機能で起動した MHP で、障害が発生した場合
予備	76	16	—	—
メッセージが入力された日付	92	8	外部 10 進数字	端末入力メッセージを入力した日付が設定されます。 「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	100	8	外部 10 進数字	端末入力メッセージを入力した時刻が設定されます。 「hhmmss00」の形式です。 hh：時

項目	位置 (バイト)	長さ (バイト)	属性	内容
メッセージが入力された時刻	100	8	外部 10 進数字	mm：分 ss：秒 00 は固定です。
理由コード※	108	4	外部 10 進数字	理由コードが設定されます。
予備	12	12	—	—

(凡例)

—：該当しません。または、使用されません。

注※

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

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

項目	位置 (バイト)	長さ (バイト)	属性	内容
エラーイベントコード	0	3	英数字	'ERR'が設定されます。
	3	3	—	—
	6	2	英数字	ERREVT3 を示す'3△'が設定されます。
入力元論理端末名称	8	8	英数字	メッセージを入力した論理端末名称が設定されます。 次に示す場合は、'*'が設定されます。 <ul style="list-style-type: none"> SPP からアプリケーション起動機能で起動した MHP で障害が発生した場合 上記の障害が発生したあとに、MCF イベントとして起動した MHP からさらにアプリケーション起動機能で起動した MHP で、障害が発生した場合
予備	16	20	—	—
予備	36	8	—	—
マップ名	44	8	—	XMAP3 のドローで指定した次画面名 (マップ名) が設定されます。 なお、次に示す場合は、空白が設定されます。 <ul style="list-style-type: none"> 次画面名を指定していない場合 アプリケーション起動機能で起動された MHP で障害が発生した場合 上記の障害発生時に、MCF イベントとして起動された MHP によっ

項目	位置 (バイト)	長さ (バイト)	属性	内容
マップ名	44	8	—	て、さらにアプリケーション起動機能で起動された MHP で、障害が発生した場合
アプリケーション名	52	8	英数字	異常が発生したメッセージのアプリケーション名が設定されます。
予備	60	8	—	—
コネクション名	68	8	英数字	コネクション名が設定されます。 次に示す場合は、'*'が設定されます。 <ul style="list-style-type: none"> SPP からアプリケーション起動機能で起動した MHP で障害が発生した場合 上記の障害が発生したあとに、MCF イベントとして起動した MHP からさらにアプリケーション起動機能で起動した MHP で、障害が発生した場合
予備	76	16	—	—
メッセージが入力された日付	92	8	外部 10 進数字	端末入力メッセージを入力した日付が設定されます。 「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	100	8	外部 10 進数字	端末入力メッセージを入力した時刻が設定されます。 「hhmmss00」の形式です。 hh：時 mm：分 ss：秒 00 は固定です。
予備	108	16	—	—
サービス名	124	31	英数字	異常が発生した MHP のアプリケーション名に対応するサービス名が設定されます。
予備	155	1	—	—
サービスグループ名	156	31	英数字	異常が発生した MHP のサービスグループ名が設定されます。
予備	187	1	—	—

項目	位置 (バイト)	長さ (バイト)	属性	内容
トランザクションブランチ ID (BID)	188	36	英数字	異常が発生したトランザクションブランチ ID が設定されます。 トランザクションブランチ ID の形式については、表 5-6 を参照してください。
予備	224	28	—	—

(凡例)

—：該当しません。または、使用されません。

表 5-6 トランザクションブランチ ID の形式

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

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

項目	位置 (バイト)	長さ (バイト)	属性	内容
エラーイベントコード	0	3	英数字	'ERR'が設定されます。
	3	3	—	—
	6	2	英数字	ERREVTA を示す'A△'が設定されます。
出力先論理端末名称	8	8	英数字	メッセージを出力する論理端末名称が設定されます。
予備	16	20	—	—
予備	36	8	—	—
マップ名	44	8	—	出力マップ名が設定されます。 send 関数で省略した場合は、(0000000000000000) ₁₆ が設定されます。
アプリケーション名	52	8	英数字	正常終了したメッセージのアプリケーション名が設定されます。MHP から送信されたメッセージの場合に設定さ

項目	位置 (バイト)	長さ (バイト)	属性	内容
アプリケーション名	52	8	英数字	れます。MHP 以外から送信された場合は、空白が設定されます。
予備	60	8	—	—
コネクション名	68	8	英数字	コネクション名が設定されます。
予備	76	16	—	—
メッセージが入力された日付	92	8	外部 10 進数字	端末入力メッセージを入力した日付が設定されます。 「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	100	8	外部 10 進数字	端末入力メッセージを入力した時刻が設定されます。 「hhmmss00」の形式です。 hh：時 mm：分 ss：秒 00 は固定です。
予備	108	16	—	—
予備	124	31	—	—
予備	155	1	—	—
予備	156	31	—	—
予備	187	1	—	—
予備	188	36	—	—
予備	224	28	—	—

(凡例)

—：該当しません。または、使用されません。

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

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	イベントコード「SCMPEVT」が設定されます。
出力先論理端末名称	8	8	英数字	出力先論理端末名称が設定されます。
予備	16	8	—	—

項目	位置 (バイト)	長さ (バイト)	属性	内容
コネクション名	24	8	英数字	コネクション名が設定されます。
メッセージが入力された日付	32	8	外部 10 進数字	SCMPEVT が通知された日付が設定されます。 「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	40	8	外部 10 進数字	SCMPEVT が通知された時刻が設定されます。 「hhmmss00」の形式です。 hh：時 mm：分 ss：秒 00 は固定です。
出力通番	48	4	2 進数字	出力通番が設定されます。 出力通番がない場合は、 (FFFFFFFF) ₁₆ が設定されます。
マップ名	52	8	英数字	出力マップ名が設定されます。 send 関数で省略した場合は、 (0000000000000000) ₁₆ が設定されます。
メッセージ種別	60	1	英数字	メッセージ種別として、次の値が設定されます。 o：応答送信メッセージ n：一般一方送信メッセージ p：優先一方送信メッセージ
理由コード	61	1	英数字	理由コードが設定されます。 △：送信が正常に終了しました。
予備	62	9	—	—
プロトコル個別情報	71	16	英数字	MCF が使用します。
予備	87	19	—	—

(凡例)

—：該当しません。または、使用されません。

表 5-9 COBOL 言語の MCF イベント情報の内容 (SERREVT)

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	イベントコード「SERREVT」が設定されます。
出力先論理端末名称	8	8	英数字	送信障害となった出力先論理端末名称が設定されます。
予備	16	8	—	—
コネクション名	24	8	英数字	コネクション名が設定されます。
メッセージが入力された日付	32	8	外部 10 進数字	SERREVT が通知された日付が設定されます。 「yyyymmdd」の形式です。 yyyy：西暦の年 mm：月 dd：日
メッセージが入力された時刻	40	8	外部 10 進数字	SERREVT が通知された時刻が設定されます。 「hhmmss00」の形式です。 hh：時 mm：分 ss：秒 00 は固定です。
出力通番	48	4	2 進数字	出力通番が設定されます。 出力通番がない場合は、 (FFFFFFFF) ₁₆ が設定されます。
マップ名	52	8	英数字	出力マップ名が設定されます。 send 関数で省略した場合は、 (0000000000000000) ₁₆ が設定されます。
メッセージ種別	60	1	英数字	メッセージ種別として、次の値が設定されます。 o：応答送信メッセージ n：一般一方送信メッセージ p：優先一方送信メッセージ
送信エラーコード	61	1	英数字	送信エラーコードとして、'E'が設定されます。
障害コード	62	1	2 進数字	障害コードとして、次の値が設定されます。 (01) ₁₆ ：送信エラー (02) ₁₆ ：論理エラー

項目	位置 (バイト)	長さ (バイト)	属性	内容
詳細情報	63	1	2進数字	詳細情報が設定されます。 (01) ₁₆ : エラーによる送信障害 (02) ₁₆ : 運用コマンド (mcftdctcn) による送信障害 (03) ₁₆ : 運用コマンド (mcftdctle) による送信障害
エラーコード	64	1	2進数字	MCF が使用します。
エラーサブコード	65	1	2進数字	メッセージの状態として、次の値が設定されます。 (00) ₁₆ : メッセージ保持 (01) ₁₆ : メッセージ消去
予備	66	5	—	—
プロトコル個別情報	71	16	英数字	詳細情報が(01) ₁₆ の場合は、表 5-10 を参照してください。 それ以外の場合は、MCF が使用します。
予備	87	19	—	—

(凡例)

—: 該当しません。または、使用されません。

表 5-10 プロトコル個別情報の内容

項目	位置 (バイト)	長さ (バイト)	属性	内容
理由コード 1	0	4	2進数字	理由コード 1 が設定されます。 (00000001) ₁₆ : MCF 障害 (00000002) ₁₆ : XMAP3 Server 障害 (00000003) ₁₆ : プリンタ障害 (00000005) ₁₆ : 無効メッセージ
理由コード 2 ^{*1}	4	4	2進数字	理由コード 1 が(00000001) ₁₆ , (00000002) ₁₆ , (00000003) ₁₆ の場合に、理由コード 2 が設定されます。
詳細コード ^{*2}	8	4	2進数字	マッピングサービスの障害コードが設定されます。
予備	12	4	—	—

(凡例)

—: 該当しません。または、使用されません。

注※1

理由コード 2 の内容については、「付録 J 理由コード一覧」を参照してください。

注※2

プリンタ送信時、ほかのメッセージのマッピング障害によって、送信障害通知されることがありますが、障害コードが設定されるのは、マッピング障害となったメッセージだけです。

表 5-11 COBOL 言語の MCF イベント情報の内容 (VERREVT)

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	イベントコード「VERREVT」が設定されます。
入力元論理端末名称	8	8	英数字	障害の発生した論理端末名称が設定されます。接続障害時は無効です。ヌル文字が設定されている場合は無効と見なします。
予備	16	8	—	—
入力元接続名	24	8	英数字	接続名が設定されます。
メッセージ入力日付	32	8	外部 10 進数字	VERREVT を起動した日付が設定されます。
メッセージ入力時刻	40	8	外部 10 進数字	VERREVT を起動した時刻が設定されます。
論理端末属性	48	4	2 進数字	論理端末の属性として、次の値が設定されます。 (00000001) ₁₆ : 画面 (00000002) ₁₆ : プリンタ
障害要因コード	52	4	2 進数字	障害の要因として、次の値が設定されます。 (00000030) ₁₆ : 接続障害 (00000031) ₁₆ : 論理端末障害
理由コード 1*	56	4	2 進数字	理由コード 1 が設定されます。
理由コード 2*	60	4	2 進数字	理由コード 2 が設定されます。
詳細コード	64	4	2 進数字	詳細コードが設定されます。
予備	68	36	—	—

(凡例)

— : 該当しません。または、使用されません。

注※

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

表 5-12 COBOL 言語の MCF イベント情報の内容 (VOPNEVT, VCLSEVT)

項目	位置 (バイト)	長さ (バイト)	属性	内容
イベントコード	0	8	英数字	イベントコード「VOPNEVT」, または「VCLSEVT」が設定されます。
入力元論理端末名称	8	8	英数字	VOPNEVT では, 論理端末画面を生成した論理端末名称が設定されます。 VCLSEVT では, 論理端末画面を廃棄した論理端末名称が設定されます。
予備	16	8	—	—
入力元コネクション名	24	8	英数字	コネクション名が設定されます。
メッセージ入力日付	32	8	外部 10 進数字	VOPNEVT, VCLSEVT を入力した日付が設定されます。
メッセージ入力時刻	40	8	外部 10 進数字	VOPNEVT, VCLSEVT を入力した時刻が設定されます。
論理端末属性	48	4	2 進数字	論理端末の属性として, 次の値が設定されます。 (00000001) ₁₆ : 画面 (00000002) ₁₆ : プリンタ
予備	52	52	—	—

(凡例)

—: 該当しません。または, 使用されません。

6

OpenTP1 マッピングサービス機能

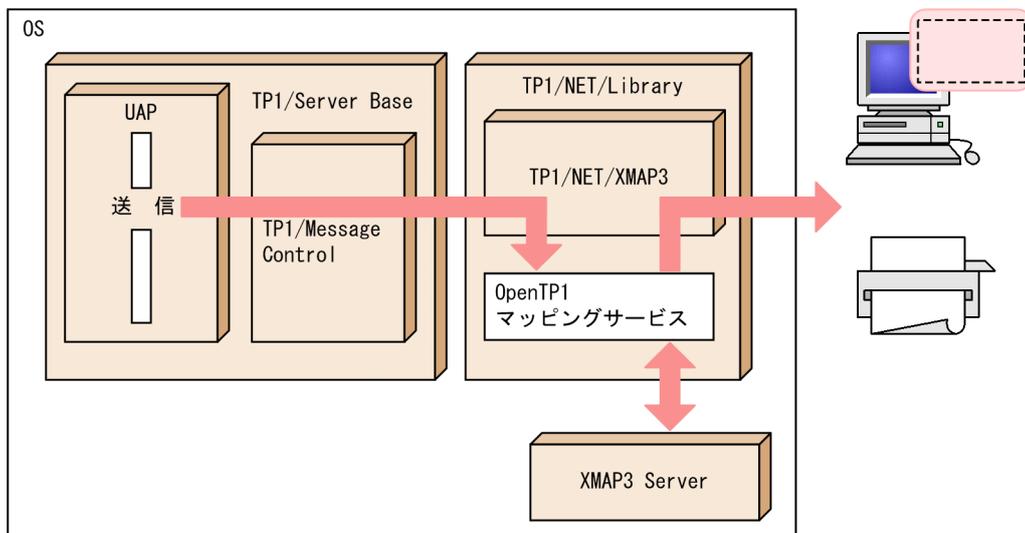
TP1/NET/XMAP3 では、OpenTP1 マッピングサービスを使ってマッピングしています。

この章では、OpenTP1 マッピングサービス機能について説明します。

OpenTP1 マッピングサービス機能の利用形態

OpenTP1 マッピングサービス機能の利用形態を、次の図に示します。

図 6-1 TP1/NET/XMAP3 が OpenTP1 マッピングサービスを利用する形態



OpenTP1 マッピングサービスは、XMAP3 Server のライブラリを使用してマッピングを処理します。

以降、OpenTP1 マッピングサービスのことを、マッピングサービスと呼びます。

OpenTP1 マッピングサービス機能の詳細

マッピングサービス機能には、次の三つがあります。

- 予約項目
- マッピングサービスプロセス
- マッピングサービスコマンド

この節では、予約項目とマッピングプロセスについて説明します。

マッピングサービスコマンドの詳細については、この章の「マッピングサービスコマンド」で説明します。

予約項目

画面・帳票定義で予約項目を指定した場合、マッピングサービスで情報を設定するので UAP で値を設定する必要はありません。

マッピングサービス機能で提供する予約項目を次の表に示します。

表 6-1 予約項目

予約項目	項目名称	フォーマット	備考
日付 1*	DATE1	yy-mm-dd	yy：西暦年の下 2 けた mm：月 dd：日 ddd：通算日
日付 2*	DATE2	yy/mm/dd	
日付 3*	DATE3	dd/mm/yy	
日付 4*	DATE4	mm/dd/yy	
日付 5*	DATE5	yy・ddd	
時刻*	TIME	hh:mm:ss	時：分：秒
論理端末名称	VTNAME	半角 8 文字	—
再送 ID	RESEND	半角 8 文字	再送メッセージ以外は表示されません。
出力通番	VTSEQ	mnnnnn	m：メッセージ種別 • O：問い合わせ応答 • N：一般分岐 • P：優先分岐 n：出力通番
出力通番 1	VTSEQ1	mnnnn	

(凡例)

—：該当しません。

注※

マッピングサービスに対してマッピング要求をしたときの日付、時刻を環境変数 TZ に従って設定します。

マッピングサービスプロセス

マルチスレッド環境では、マップのローディングが要求されると、マップをローディングしている間、要求元スレッドのプロセス全体がブロックされます。プロセス全体がブロックされることを避けるため、マップロード用のプロセスを用意します。これをマッピングサービスプロセスと呼びます。

マッピングサービスプロセスの起動、終了、およびマッピングサービスプロセスの機能について次に示します。

(1) マッピングサービスプロセスの起動

OpenTP1 システムが起動されると、自動的にマッピングサービスプロセスが起動されます。あらかじめ、MCF マネージャ共通定義 (mcfmcomn -m) の mapsvname オペランドで、_mp で始まるマッピングサービス定義ファイル名を指定しておきます。

マッピングサービスプロセスは、マッピングサービス定義^{※1}、およびマッピングサービス属性定義^{※2}の定義に従って、マッピングサービスを開始します。

注※1

ファイル名=\$DCCONFPATH/マッピングサービス名

マッピングサービス定義については、この章の「[OpenTP1 マッピングサービスに関する定義](#)」を参照してください。

注※2

ファイル名=\$DCCONFPATH/MAPSattr/dcmap.def (ユーザで任意に指定できます)

マッピングサービス属性定義については、この章の「[OpenTP1 マッピングサービスに関する定義](#)」を参照してください。

(2) マッピングサービスプロセスの終了

OpenTP1 システムが終了すると、自動的にマッピングサービスプロセスも終了します。

(3) マッピングサービスプロセスの機能

ここでは、マッピングサービスプロセスの機能について説明します。

(a) 常駐マップの管理

常駐マップとは、マッピングサービスの開始から終了まで、常にメモリ上にローディングされている物理マップのことです。常駐マップは、マッピングサービス属性定義の MAPNAME オペランドで指定します。マッピングサービス開始時、OpenTP1 システムは、すべての常駐マップをローディングします。

マッピングサービスは、物理マップを標準読み込みパスから常駐マップローディング用バッファにローディングして管理します。頻繁に使用するマップをバッファに常駐させることで、入出力に掛かる時間を短縮できます。

(b) 非常駐マップの管理

マッピングサービス属性定義の MAPNAME オペランドで指定されていない、常駐マップ以外のマップを**非常駐マップ**と呼びます。メッセージの送受信時、常駐マップローディングバッファに必要な物理マップがない場合は、非常駐マップをローディングします。非常駐マップは LRU 管理マップローディング用バッファに格納されます。

マッピングサービスは、まず、必要な物理マップが LRU 管理マップローディング用バッファ内にある場合には、マップのローディングはしないで、そのマップを使用してマッピング処理をします。

LRU 管理マップローディング用バッファ内に必要な物理マップがない場合には、LRU 管理マップローディング用バッファにマップをローディングします。新しい非常駐マップをローディングするため、ほかの非常駐マップを削除する必要がある場合、最も古い時点に使用されたマップを削除します。最も古い時点に使用されたマップから削除していく物理マップの管理を**マップ LRU 管理**と呼びます。この場合、LRU 管理されるマップの数は、最大でマッピングサービス属性定義の MAPCNT オペランドで指定した数になります。

(c) 物理マップ読み込みパス

マッピングサービスは、通常、マッピングサービス属性定義の MAPPATH オペランドで指定した標準用物理マップ読み込みパスから物理マップをローディングします。マッピングサービスは、標準用物理マップ読み込みパスからの非常駐マップのローディングに失敗すると、交代用の読み込みパスから物理マップをローディングします。マッピングサービス属性定義の ALTPATH オペランドで、交代用物理マップ読み込みパス名を指定しておいてください。なお、交代用物理マップ読み込みパスには標準用物理マップ読み込みパスと同じ物理マップを用意しておいてください。

(d) マップファイルの作成

物理マップを格納するファイルを**マップファイル**と呼びます。

マップファイルは PC で XMAP3 のドロー、またはマップ生成機能でターゲットに合った物理マップと論理マップを作成して使用します。

UNIX の場合は、作成した物理マップを UNIX 環境に転送します。UNIX 環境にファイル転送する際には、転送モードに注意してください。また、UNIX 環境に物理マップを転送後、UNIX 版 XMAP3 Server Runtime が提供する cmapcp コマンドを実行してください。Windows の場合は、作成した物理マップをそのまま使用します。

詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 開発ガイド」、またはマニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」を参照してください。

OpenTP1 マッピングサービスに関する定義

複数の MCF 通信プロセスがマッピングサービスを使用する場合、マッピング処理の性能確保のために、または通信プロセスごとに異なったマッピング環境を使用するために、複数のマッピングサービスを生成できます。

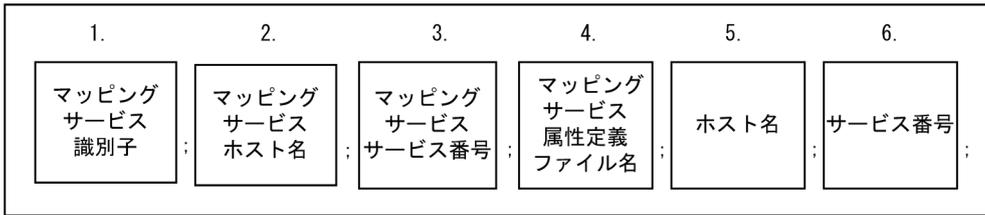
マッピングサービス定義では、マッピングサービスを識別するための識別子および動作条件（サービス番号、属性定義ファイル名）を指定します。

マッピングサービス属性定義では、マッピングサービス識別子ごとに、そのマッピング属性を指定します。

マッピングサービス定義とマッピングサービス属性定義の詳細は以降で説明します。

マッピングサービス定義

形式



この形式で、マッピングサービス識別子ごとに1行（255バイト以内）で記述してください。先頭から15行までを有効とします。行の継続はできません。

複数のマッピングサービスを使用する場合の制約の詳細については、「7. システム定義」の「各種定義数に関する制約事項」を参照してください。

行の先頭に#があった場合、その行をコメントと見なします。

行の途中に#があった場合、その行の#以降をコメントと見なします。

定義文の解析でエラーが発生した場合は、その行をスキップして無効にします。

定義例

適用 OS が UNIX の場合

```
mapserv1;h3050A;8500;dcmap.def;h3050A;0;
```

適用 OS が Windows の場合

```
mapserv1;h3050A;8500;dcmap.def;windows;0;
```

複数のマッピングサービスを使用する場合（適用 OS が Windows の場合の例）

```
mapserv1;h3050A;8500;dcmap.def;windows;0;  
mapserv2;h3050A;8501;dcmap.def;windows;0;
```

機能

マッピングサービス識別子ごとにマッピングサービス定義情報を指定します。

オペランド

1. マッピングサービス識別子 ～ 〈1～8文字の識別子〉

マッピングサービスを識別するための識別子を指定します。

2. マッピングサービスホスト名 ～〈1～32文字の識別子〉

自ホスト名を指定します。

ホスト名は、hosts ファイル、DNS など、IP アドレスとのマッピングができる必要があります。

3. マッピングサービス番号 ～〈符号なし整数〉((1024～65535))

サーバを意識するための番号です。サービス番号がほかのサーバと重複しないように指定してください。

4. マッピングサービス属性定義ファイル名 ～〈1～12文字の識別子〉

マッピングサービスの属性を定義したファイル名を指定します。

5. ホスト名 ～〈1～32文字の識別子〉

- OS が UNIX の場合は、 '*' または自ホスト名を指定します。
- OS が Windows の場合は、 windows を指定します。

6. サービス番号

0 を指定します。

マッピングサービス属性定義

形式

```
MAPDEF
  SYSTNAM = ( XPRES )
  MAPPATH = 標準用物理マップ読み込みパス名
  [ALTPATH = 交代用物理マップ読み込みパス名]
  [MAPNAME = ( 物理マップ名 [, 物理マップ名] ...)]
  [MAPCNT = LRU管理をする物理マップ数]
  [POOLSIZ = 物理マップをロードする領域サイズ]
  [NULLCD = ヌルクリアコード]
  [DATA CD = データ有無コード]
  [PADCHR = 初期設定値]
  [RESEND = 再送ID]
  [RETRYSEC = 再試行間隔]
  [RETRYCNT = 再試行回数]
  [PATHSTS = YES | NO]
  [ERRLOG4 = YES | NO]
END
```

注

各オペランドは形式に示すと通りの順序で記述してください。

1行は、255バイト以内で記述してください。

MAPNAME オペランドの () 内でだけ、行を継続できます。継続文字は¥です。その他のオペランドでの行の継続はできません。

行の先頭に#があった場合、その行をコメントと見なします。

行の途中に#があった場合、その行の#以降をコメントと見なします。

機能

マッピングサービスの属性を定義します。

マッピングサービス定義でマッピングサービス識別子ごとに、マッピングサービス属性定義ファイルを指定します。この場合、複数のマッピングサービス識別子が、同じマッピングサービス属性定義ファイルを指定してもかまいません。

オペランド

●MAPDEF

マッピングサービス属性定義の開始を宣言します。この定義文以前にはほかの定義文を記述することはできません。

●SYSTNAM = (XPRES)

マッピングをする端末システム名を XPRES と指定します。

●MAPPATH = 標準用物理マップ読み込みパス名 ~ 〈1~255 文字のパス名〉

メッセージ編集時に使用する標準用マップライブラリを格納しているパス名を指定します。マッピングサービスは、ここで指定したパス名から物理マップを読み込みます。

●ALTPATH = 交代用物理マップ読み込みパス名 ~ 〈1~255 文字のパス名〉

標準用マップライブラリの物理マップ読み込みに失敗した場合に使用する、交代用マップライブラリのパス名を指定します。

●MAPNAME = (物理マップ名 [, 物理マップ名] ...) ~ 〈3~8 文字の英数字〉

メモリに常駐する物理マップの名称を指定します。指定できる物理マップ数は最大 25 個です。

●MAPCNT = LRU 管理をする物理マップ数 ~ 〈符号なし整数〉 ((1~32767)) 〈10〉

LRU 管理の対象となる物理マップの数を指定します。必ず該当するマッピングサービスを使用する MCF 通信プロセスの数以上を指定してください。

●POOLSIZ = 物理マップをロードする領域サイズ ~ 〈符号なし整数〉 ((1~32767)) 〈32〉

LRU 管理をする物理マップを読み込む領域の大きさをキロバイト単位で指定します。メモリに常駐する物理マップ (常駐マップ) は対象にしません。

領域のサイズを求める計算式を次に示します。

LRU管理をする物理マップのファイルサイズの最大値×MAPCNTの指定値

●NULLCD = nulクリアコード ~ 〈2 けたの 16 進数字〉 ((00~FF))

nulクリアコードを指定します。指定しない場合は、nulクリアコードは無効になります。nulクリアコードの先頭には 16 進数字であることを示す「0x」を指定できます。

出力論理フィールドの先頭 1 バイトにnulクリアコードを格納すると画面上の対応フィールドをnul (00)₁₆ クリアします。

●DATACD = データ有無コード ~ 〈2 けたの 16 進数字〉 ((00~FF))

画面・帳票定義のデータ有無コードを変更する場合に指定します。指定しない場合は、データ有無コードは無効になります。データ有無コードの先頭には 16 進数字であることを示す「0x」を指定できます。

●PADCHR = 初期設定値 ~ 〈2 けたの 16 進数字〉 ((00~FF))

入力編集時に入力論理マップの各データ項目をあらかじめクリアしておく初期設定値を指定します。ユーザが何も入力しなかったデータ項目には、この値が設定されます。初期設定値の先頭には 16 進数字であることを示す「0x」を指定できます。

入力編集時の初期設定値の優先度は次のとおりです。

1. XMAP3 のドロー、ドローセットアップ (ドローセットアップの論理マップ属性ダイアログで指定する「初期クリア文字」など)

2. PADCHR オペランド

XMAP3 のドロー、ドローセットアップ、および PADCHR オペランドで初期設定値を指定しない場合、初期設定値は無効になります。

●RESEND = 再送 ID ~ 〈1~8 文字の英数字〉

画面・帳票定義の予約項目の項目名称を RESEND と定義したフィールドにセットする再送 ID を指定します。

●RETRYSEC = 再試行間隔 ~ 〈符号なし整数〉 ((1~60)) 〈3〉

マッピングサービスプロセスとのデータ送受信に失敗したときの再試行間隔を、秒単位で指定します。

●RETRYCNT = 再試行回数 ~ 〈符号なし整数〉 ((0~255)) 〈10〉

マッピングサービスプロセスとのデータ送受信に失敗したときの再試行回数を指定します。0 を指定した場合は再試行しません。

●PATHSTS={YES | NO}

dcmapchg コマンドで変更したパス情報のリラン引き継ぎをするかどうかを指定します。

YES :

パス情報のリラン引き継ぎをします。

NO :

パス情報のリラン引き継ぎをしません。

オペランドを省略すると NO を指定したとみなされます。

●ERRLOG4={YES | NO}

処理中にマッピングエラーが発生した場合、エラー情報を取得するかどうかを指定します。エラー情報は、\$DCDIR/spool/save の下に dcmap_errxxxxx (xxxxx はプロセス ID) というファイルを作成してその中に出力されます。

YES :

エラー情報を取得します。

NO :

処理中にエラーコード 4 (軽いエラー) のマッピングエラーが発生した場合、エラー情報を取得しません。

オペランドを省略すると NO を指定したとみなされます。

●END

マッピングサービス属性定義の終わりを宣言します。

OpenTP1 マッピングサービス機能の定義例

マッピングサービス機能のシステム構成例を次の図に示します。

図 6-2 マッピングサービス機能のシステム構成例 (UNIX の場合)

ホスト名称 : h3050A

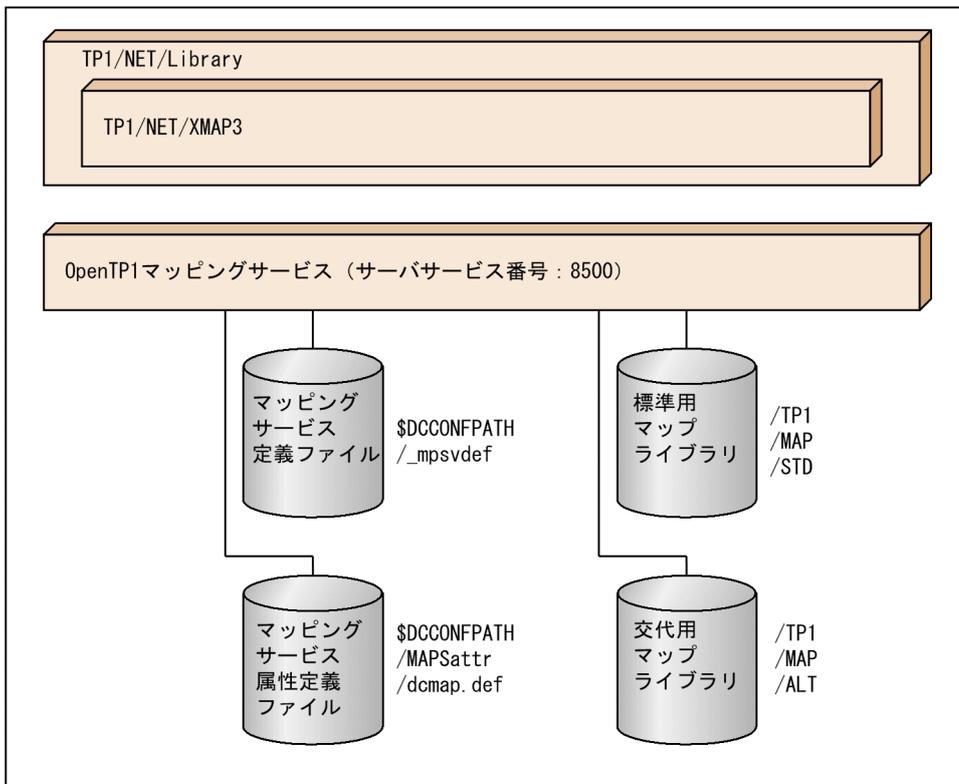
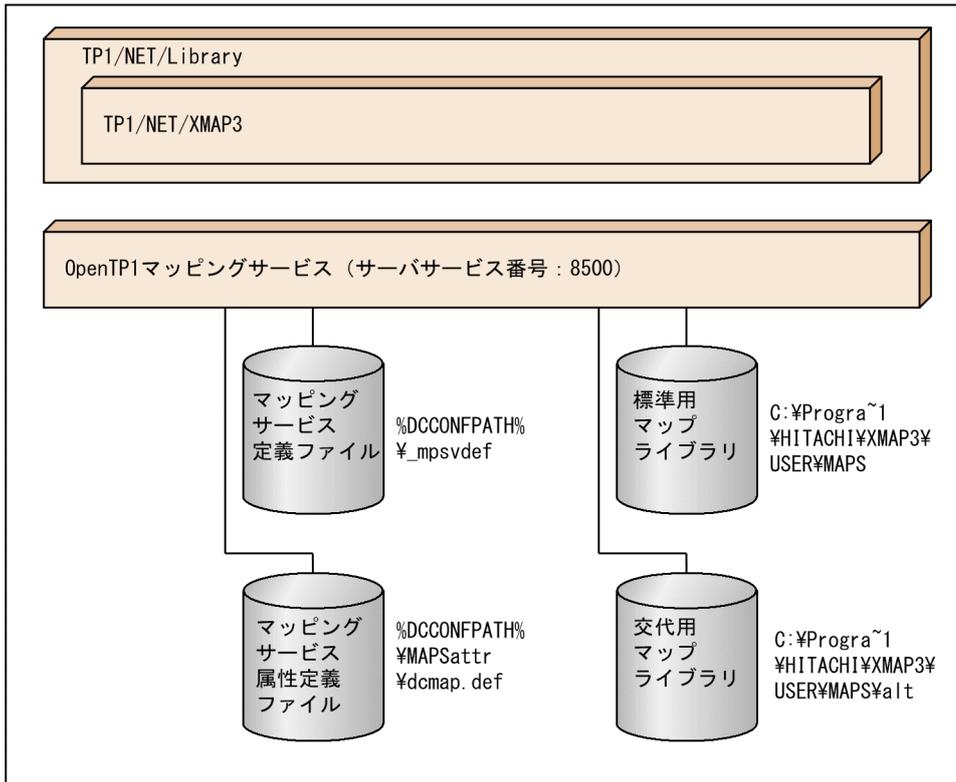


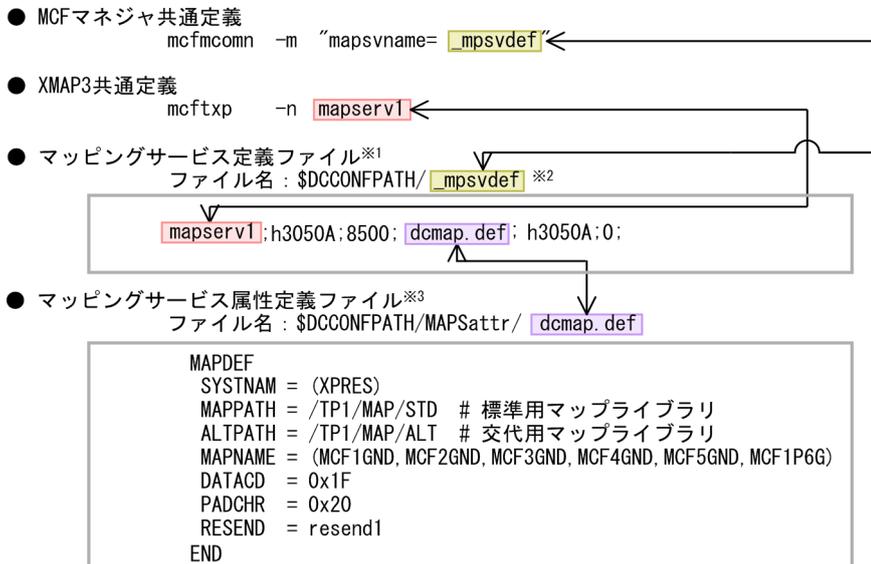
図 6-3 マッピングサービス機能のシステム構成例 (Windows の場合)

ホスト名称 : h3050A



マッピングサービス機能の定義例を次の図に示します。

図 6-4 マッピングサービス機能の定義例



注※1

適用 OS が UNIX の場合の例です。

マッピングサービス定義例は、次の場所で提供しています。

- 適用 OS が UNIX の場合 : /BeTRAN/examples/mcf/XMAP3/conf/_mpsvdef

- 適用 OS が Windows の場合：`%DCDIR%\examples\mcf\XMAP3\conf*_mpsvdef`

注※2

定義文に誤りを検出した場合には、該当するマッピングサービス識別子のサービスを開始しません。

注※3

適用 OS が UNIX の場合の例です。

マッピングサービス属性定義例は、次の場所で提供しています。

- 適用 OS が UNIX の場合：`/BeTRAN/examples/mcf/XMAP3/conf/dcmap.def`
- 適用 OS が Windows の場合：`%DCDIR%\examples\mcf\XMAP3\conf*MAPSattr\dcmap.def`

マッピングサービスコマンド

パス名の変更

マッピングサービス属性定義の MAPPATH で指定した標準用物理マップ読み込みパス名を変更できます。また、ALTPATH で指定した交代用物理マップ読み込みパス名も変更できます。

OpenTP1 システムは、標準用物理マップ読み込みパス名を変更するコマンドが入力されると、常駐マップローディング用バッファと LRU 管理マップローディング用バッファの内容を、いったん消去します。その後、コマンドで指定した読み込みパスから、常駐マップをバッファにローディングします。

交代用物理マップ読み込みパス名を変更するコマンドが入力されても、常駐マップローディング用バッファと LRU 管理マップローディング用バッファの内容は消去されません。また、常駐マップのバッファへのローディングも起きません。

マッピングサービスコマンドを次の表に示します。

表 6-2 マッピングサービスコマンド

コマンド名	機能
dcmaphg	物理マップファイルの読み込みパス名を変更します。
dcmaphs	ロード済みの物理マップを表示します。

dcmapchg (読み込みパス名の変更)

形式

```
dcmapchg {{-mM | -aM} マッピングサービス識別子 パス名 | -h}
```

機能

物理マップファイルの読み込みパス名を変更します。

オプション

●-mM

標準用物理マップ読み込みパス名を変更する場合に指定します。

●-aM

交代用物理マップ読み込みパス名を変更する場合に指定します。

●マッピングサービス識別子 ~ 〈1~8文字の英数字〉

マッピングサービス定義で指定したマッピングサービス識別子を指定します。

●パス名 ~ 〈1~255文字のパス名〉

変更後のパス名を指定します。

●-h

コマンドのヘルプメッセージを表示します。

記述例

- 標準用物理マップ読み込みパス名を/tmpに変更する場合

```
dcmapchg -mM mapserv1 /tmp
```

出力メッセージ

出力メッセージID	内容	出力先
KFCA10865-I	ヘルプメッセージ	標準出力
KFCA10870-E	オプション<%s>の指定に誤りがあります。 再度指定してください。	標準出力
KFCA10871-E	マッピングサービス識別子<%s>の指定に誤りがあります。 再度指定してください。	標準出力
KFCA10872-E	dcmapchg コマンド入力に誤りがあります。	標準出力

出力メッセージ ID	内容	出力先
KFCA10872-E	再度指定してください。	標準出力
KFCA10875-E	マッピングサービスプロセスとの通信に異常を検知しました。	標準出力
KFCA10876-E	パス名の変更が失敗しました。	標準出力
KFCA10877-E	マッピングサービス識別子<%s>はマッピングサービス定義ファイルに定義されていません。	標準出力
KFCA10878-I	パス名を変更しました。 {mappath altpath}= <パス名>	標準出力
KFCA10879-E	マッピングサービスが起動されていません。	標準出力
KFCA10880-E	パス名<%s>の指定に誤りがあります。 再度指定してください。	標準出力
KFCA10886-E	コマンド処理中にエラーが発生しました。	標準出力

dcmapls (ロード済み資源の表示)

形式

```
dcmapls {{-s | -d | -p} マッピングサービス識別子 | -h}
```

機能

ロード済みの物理マップを表示します。LRU 管理されている物理マップは、その時点で最も新しく使用された物理マップから順に表示されます。常駐指定した物理マップでもマッピングサービス開始時に入出力エラーなどで常駐化できなかったものは表示されません。

オプション

●-s

標準形式で表示する場合に指定します。

●-d

詳細形式で表示する場合に指定します。

●-p

読み込みパス名だけ表示する場合に指定します。

●マッピングサービス識別子 ~ 〈1~8文字の英数字〉

マッピングサービス定義で指定したマッピングサービス識別子を指定します。

●-h

コマンドのヘルプメッセージを表示します。

記述例

- マッピングサービス識別子 mapserv1 のロード済み資源情報を標準形式で表示する場合

```
dcmapls -s mapserv1
```

出力形式

```
iiiiiii tttttt pppppp  
mmmmmm ssss※ hhhhhh※ llllll※ oooooo※
```

注※

オプション-d を指定した場合に出力されます。

- iiiiiii : マッピングサービス識別子

- tttttttt：パス表示種別
MAPNAME…常駐する物理マップの読み込みパス
MAPPATH…標準物理マップ読み込みパス
ALTPATH…交代物理マップ読み込みパス
- pppppppp：パス名
- mmmmmmmm：物理マップ名
オプション-p を指定した場合、またはマッピングサービス属性定義ファイルでパス名称が省略されている場合は出力されません。
- sssss：物理マップサイズ
- hhhhhhhh：保守情報 1
- llllllll：保守情報 2
- oooooooo：保守情報 3

出力メッセージ

メッセージ ID	内容	出力先
KFCA10866-I	ヘルプメッセージ	標準出力
KFCA10870-E	オプション<%s>の指定に誤りがあります。 再度指定してください。	標準出力
KFCA10871-E	マッピングサービス識別子<%s>の指定に誤りがあります。 再度指定してください。	標準出力
KFCA10872-E	dcmaplс コマンド入力に誤りがあります。 再度指定してください。	標準出力
KFCA10875-E	マッピングサービスプロセスとの通信に異常を検知しました。	標準出力
KFCA10877-E	マッピングサービス識別子<%s>はマッピングサービス定義ファイルに定義されていません。	標準出力
KFCA10879-E	マッピングサービスが起動されていません。	標準出力
KFCA10881-E	マップ名称一覧表示に失敗しました。	標準出力
KFCA10886-E	コマンド処理中にエラーが発生しました。	標準出力
KFCA10888-I	MPS の状態表示を開始します。	標準出力
KFCA10889-I	MPS の状態表示を終了します。	標準出力
KFCA10890-I	表示情報 (マップ識別子, パス表示種別, パス名称)	標準出力
KFCA10891-I	表示情報 (物理マップ名)	標準出力
KFCA10892-I	表示情報 (名称, サイズ, 保守情報)	標準出力

注意事項

TP1/NET/XMAP3 でマッピングサービス機能を使用する場合の注意事項を次に示します。

(1)定義できないマップ名

マップ定義する場合のマップ名には、"THROUGH", "OPEMSG1", "OPEMSG2", "OPEMSG3", "OPEMSG4", "TESTPRT", および"SYSMSG"を使用できません。

(2)マッピングエラー情報

マッピング処理中にマッピングエラーが発生した場合、エラー情報を次のファイルに採取します。

ファイル名称：\$DCDIR/spool/save/dcmap_err×××

×××は、プロセス番号に置き換えられます。

7

システム定義

この章では、TP1/NET/XMAP3 を使用するために必要な、OpenTP1 のシステム定義の中での TP1/NET/XMAP3 固有のシステム定義、および定義例について説明します。

TP1/NET/XMAP3 の定義の概要

TP1/NET/XMAP3 のシステム定義は、OpenTP1 のネットワークコミュニケーション定義の中で定義します。

OpenTP1 のネットワークコミュニケーション定義の中での定義

OpenTP1 のネットワークコミュニケーション定義のうち、TP1/NET/XMAP3 に固有の定義について説明します。

使用する定義ファイル

MCF および TP1/NET/XMAP3 を起動するには、定義ファイルに環境情報を設定する必要があります。MCF で使用する定義ファイルを次の表に示します。

表 7-1 MCF で使用する定義ファイル

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

定義のソースファイルは、定義コマンド、オプション、およびオペランドを指定して作成します。それらの中には、プロトコルで共通のものと、プロトコルに固有のものがあります。次に示す定義には、TP1/NET/XMAP3 に固有の定義があります。

- MCF マネージャ定義
- MCF 通信構成定義

この章では、TP1/NET/XMAP3 に固有の定義コマンド、オプション、およびオペランドについて説明します。プロトコルで共通の定義については、マニュアル「OpenTP1 システム定義」を参照してください。ただし、バッファグループ定義 (mcftbuf -g) の length, count オペランドの指定値については、この章の「[mcftalccn \(コネクション定義の開始\)](#)」の注意事項に記載しています。

TP1/NET/XMAP3 の組み込み時に必要なファイル

次に示すファイルは、TP1/NET/XMAP3 を OpenTP1 システムに組み込む場合に必要なファイルです。

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

この章では、システムサービス情報定義ファイルとシステムサービス共通情報定義ファイルの記述内容、および MCF 定義オブジェクトファイルを生成するユーティリティの起動コマンドについて説明します。TP1/NET/XMAP3 を組み込む方法については、「[10. 組み込み方法](#)」を参照してください。

TP1/NET/XMAP3 固有のシステム定義の種類

OpenTP1 のネットワークコミュニケーション定義のうち、TP1/NET/XMAP3 に固有の定義の一覧を次の表に示します。

表 7-2 TP1/NET/XMAP3 に固有の定義の一覧

定義名	コマンド	オプション・オペランド		定義内容	指定値((値範囲))《省略時解釈値》		
MCF マネージャ定義	mcfmcomn (MCF マネージャ 共通定義) ※1	-m	mapsvname	マッピングサービス名	1~8 文字の英数字		
		-o	cmdsname	オンラインコマンドサービス名	1~8 文字の英数字		
MCF 通信構成定義	共通定義	プロトコル共通のコマンドだけで指定できます。共通のコマンドについては、マニュアル「OpenTP1 システム定義」を参照してください。					
	プロトコル固有定義	mcftxp (XMAP3 共通定義) 指定数: 1	-m	msgkey	次メッセージ要求キー名	((PF1~PF24, PA1~PA3, BREAK))《PF2》	
				timeoutlog	XMAP3 Server のタイムアウト発生を通知するかどうかを指定	《yes》 no	
			-o	opcolor	オペレータインジケータのユーザ表示フィールドの表示色	符号なし整数 ((1~15))《2》	
				opreverse	オペレータインジケータのユーザ表示フィールドを反転表示するかどうかを指定	yes 《no》	
			-n	—	マッピングサービス識別子	1~8 文字の識別子	
			-g	closebutton	論理端末画面の閉じるボタンを有効にするかどうかを指定	on 《off》	
			-c	actnaction	コネクションの確立の動作モード	《serial》 parallel	
			mcftalccn (コネクション定義の開始) 指定数: 1~512 ※2	-c	—	コネクション ID	1~8 文字の識別子
				-p	—	プロトコルの種別	xp
				-g	sndbuf	メッセージ送信用バッファグループ番号	符号なし整数 ((1~512))
	rcvbuf	メッセージ受信用バッファグループ番号			符号なし整数 ((1~512))		
	-e	msgbuf		メッセージ編集用バッファグループ番号	符号なし整数 ((1~512))		

定義名		コマンド	オプション・オペランド		定義内容	指定値((値範囲))《省略時解釈値》	
MCF 通信構成定義	プロトコル固有定義	mcftalccn (コネクション定義の開始) 指定数：1~512 ※2	-e	count	メッセージ編集用バッファ数	符号なし整数 ((1~131070))	
			-i	—	コネクションの確立方法	auto 《manual》	
			-b	bretry	コネクション確立障害時の確立再試行するかどうかを指定	《yes》 no	
				bretrycnt	コネクション確立障害時の確立再試行回数	符号なし整数 ((0~65535)) 《3》(単位：回)	
				bretryint	コネクション確立障害時の確立再試行間隔	符号なし整数 ((0~2550)) 《60》(単位：秒)	
			-S	svname	接続対象の表示印刷サービス名	1~14 文字の識別子	
				type	接続対象の表示印刷サービスのサービスタイプ	DSP PRT	
			mcftalcle (論理端末定義) 指定数：1~2048	-l	—	論理端末名称	1~8 文字の識別子
				-t	—	論理端末の端末タイプ	any
				-m	mmsgcnt	メモリ出力メッセージ最大格納数	符号なし整数 ((0~65535)) 《0》
		dmsgcnt			ディスク出力メッセージ最大格納数	符号なし整数 ((0~65535)) 《0》	
		-k		quekind	出力メッセージの割り当て先	《memory》 disk	
				quegrpид	キューグループ ID	1~8 文字の識別子	
		-G		groupsend	メッセージグループ送信をするかどうかを指定	yes 《no》	
		-o		aj	メッセージ送信完了ジャーナルを取得するかどうかを指定	《yes》 no	
		-p		dflmap	デフォルトマップ名	1~6 文字の記号名称	
				initmap	イニシャライズマップ名	1~6 文字の記号名称	
		-i	—	論理端末の自動閉塞解除の要否	《auto》 manual		
		-a	msgadv	メッセージの送信スケジュール	《auto》 manual		
		-r	rretry	プリンタ排他再試行をするかどうかを指定	《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》
MCF 通信構成定義	プロトコル固有定義	mcftalcle (論理端末定義) 指定数：1～2048	-r	retrycnt	プリンタ排他再試行回数	符号なし整数 ((0～65535))《0》
				retryint	プリンタ排他再試行間隔	符号なし整数 ((0～255))《5》 (単位：秒)
			-g	title	論理端末画面のタイトルバーを表示するかどうかを指定	on off 《XMAP3 Server の仮定値》
				titlename	論理端末画面のタイトル名	1～80 文字の文字列 《XMAP3 Server の仮定値》
				maxbutton	論理端末画面の最大化ボタンを表示するかどうかを指定	on off 《XMAP3 Server の仮定値》
				minbutton	論理端末画面の最小化ボタンを表示するかどうかを指定	on off 《XMAP3 Server の仮定値》
				resize	論理端末画面の表示の大きさを変更するかどうかを指定	yes no 《XMAP3 Server の仮定値》
				icon	論理端末画面のアイコンを表示するかどうかを指定	yes no 《XMAP3 Server の仮定値》
				iconname	論理端末画面のアイコン名	1～16 文字の文字列 《XMAP3 Server の仮定値》
				pheight	論理端末画面の縦サイズ	符号なし整数 ((24～127))《XMAP3 Server の仮定値》
				pwidth	論理端末画面の横サイズ	符号なし整数 ((80～255))《XMAP3 Server の仮定値》
				wheight	論理端末画面視野枠の縦サイズ	符号なし整数 ((24～48))《XMAP3 Server の仮定値》
			wwidth	論理端末画面視野枠の横サイズ	符号なし整数 ((4～160))《XMAP3 Server の仮定値》	
			-z	device	機種・モデル	6 文字の記号名称

定義名		コマンド	オプション・オペランド		定義内容	指定値((値範囲))《省略時解釈値》
MCF 通信構成定義	プロトコル固有定義	mcftalcle (論理端末定義) 指定数: 1~2048	-y	—	送信完了確認メッセージ数	符号なし整数 ((1~256))《10》
			-x	excnt	プリンタ連続使用メッセージ数	符号なし整数 ((0, 5~256))《0》
			-K	keyboard	論理端末画面のキーボード状態	《aptype》 aplock
				unlock	論理端末画面のキーボードロック解除のタイミング	《auto》 send
		mcftalced (コネクション定義の終了) 指定数: mcftalccn と同数	—	—	コネクション定義の終了	—
MCF アプリケーション定義	mcfaalcap (アプリケーション属性定義) ※1 指定数: 1~8192	-n※3	replychk	応答送信チェックをするかどうかを指定	《yes》 no	
			type	アプリケーションの型	ans 《noans》 cont	
			tempsize	継続問い合わせ応答用一時記憶データ格納用領域サイズ	符号なし整数 ((1~32000))《4096》(単位: バイト)	

(凡例)

—: 該当しません。

注※1

このコマンドのほかのオプションについては、マニュアル「OpenTP1 システム定義」を参照してください。

注※2

実際に接続できるコネクション数は、XMAP3 サーバの接続可能表示印刷サービス数に依存します。

また、1 台の端末が無応答になると、一定時間内は該当する通信プロセス配下のほかの端末に対応したコネクションも使用できなくなることがあります。そのため、多数の端末を接続する場合は、通信プロセスを複数にして、1 通信プロセス当たりのコネクション数を減らし、影響範囲を限定するよう設計してください。

なお、一つの表示印刷サービスに対して同時に接続できるコネクション数、および一つのコネクション配下で同時に閉塞解除できる画面属性の論理端末数はそれぞれ 15 までです。

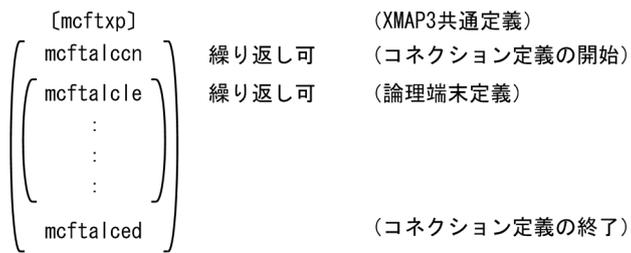
注※3

このオプションのほかのオペランドについては、マニュアル「OpenTP1 システム定義」を参照してください。

定義の指定順序

TP1/NET/XMAP3 のプロトコル固有定義コマンドの指定順序を、次の図に示します。TP1/NET/XMAP3 定義コマンドを指定する場合は、必ずこの順序に従ってください。

図 7-1 TP1/NET/XMAP3 のプロトコル固有定義コマンドの指定順序



注

接続対象が表示サービスの場合、MCF 通信構成定義の mcftalccn 定義コマンドと mcftalcle 定義コマンドとの関係は、1 対 n です。印刷サービスの場合は、1 対 1 です。

mcfaalcap (アプリケーション属性定義)

形式

```
mcfaalcap -n ”
                :
      [replychk=yes | no]
                :
      [type=ans | noans | cont]
                :
      [tempsize=継続問い合わせ応答用一時記憶データ格納用領域サイズ] ”
                :
```

機能

アプリケーションに関する属性を定義します。

オプション

この定義コマンドには、ほかにもオプションがあります。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

●-n

このオプションには、ほかにもオペランドがあります。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

(オペランド)

replychk=yes | no ~ 《yes》

応答型 MHP が応答メッセージを送信したかどうか、MCF にチェックさせるかどうかを指定します。

yes

応答型 MHP が応答メッセージを送信していない場合、またはほかの応答型 MHP を起動していない場合に、トランザクションをロールバックし ERREVT3 を通知します。

no

応答型 MHP が応答メッセージを送信したかどうかをチェックしません。

replychk オペランドを指定した場合は、type オペランドに ans を、kind オペランドに user を指定する必要があります。

replychk オペランドの指定値と、MHP の処理、MCF の処理との関係を次に示します。

replychk オペランドの指定値	MHP の処理	MCF の処理
yes	入力元端末に応答メッセージを送信	端末に応答メッセージを送信し、端末の応答待ち状態を解除する (MHP は正常終了)。

replychk オペランドの指定値	MHP の処理	MCF の処理
yes	ほかの問い合わせ応答型 MHP を起動	指定された MHP を起動する。端末の応答待ち状態は解除しない* (MHP は正常終了)。
	上記以外	エラーイベントの起動によって端末の応答待ち状態を解除し、ロールバック処理する (MHP は異常終了)。
no	入力元端末に回答メッセージを送信	端末に回答メッセージを送信し、端末の応答待ち状態を解除する (MHP は正常終了)。
	ほかの問い合わせ応答型 MHP を起動	指定された MHP を起動する。端末の応答待ち状態は解除しない* (MHP は正常終了)。
	上記以外	端末の応答待ち状態を解除する (MHP は正常終了)。

注※

端末の応答待ち状態を解除しないのは、起動された MHP に応答権が引き継がれるためです。

type=ans | noans | cont ~ 《noans》

アプリケーションの型を指定します。

ans

応答型

noans

非応答型

cont

継続問い合わせ応答型

name オペランドで MCF イベント名を指定した場合、この type オペランドの指定は無効になります。MCF は MCF イベントに対して次のように自動的に設定します。

- 次の MCF イベントの場合、要求元となった論理端末の端末タイプに応じて設定します。
 - 不正アプリケーション名検出通知イベント (ERREVT1)
- 次の MCF イベントの場合、それを起動する要因となったアプリケーションの型をそのまま引き継ぎます。
 - メッセージ廃棄通知イベント (ERREVT2)
 - UAP 異常終了通知イベント (ERREVT3)
- 上記以外の MCF イベントの場合は、noans を設定します。

なお、ERREVT1、ERREVT2、または ERREVT3 を起動する場合、アプリケーション起動プロセスが必要です。アプリケーション起動プロセスを起動するために、MCF 通信構成定義を作成してください。異常が発生した場合、起動元通信プロセスに対応する MCF イベントが起動されます。アプリケーション起動機能を使用して複数の MCF アプリケーション定義を経由した場合も、最初の起動元通信プロセスに対応する MCF イベントが起動されます。

SPP からアプリケーションを起動し、起動元プロセスがない場合は、アプリケーション起動プロセスに対応する MCF イベントが起動されます。

tempsize=継続問い合わせ応答用一時記憶データ格納用領域サイズ ~ 〈符号なし整数〉 ((1~32000))
《4096》 (単位：バイト)

継続問い合わせ応答で使用する一時記憶データ格納用領域サイズを指定します。ただし、type オペランドで cont を指定した場合だけ有効です。

mcfmcomn (MCF マネジャ共通定義)

形式

```
mcfmcomn          :  
  [-m " [mapsvname=マッピングサービス名] "]  
  [-o " [cmdsvname=オンラインコマンドサービス名] "]  
                  :
```

機能

各 MCF 間で共用するリソースの実行環境を定義します。

オプション

この定義コマンドには、ほかにもオプションがあります。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

●-m

(オペランド)

mapsvname=マッピングサービス名 ~ 〈1~8 文字の英数字〉

マッピングサービスプロセスのサービス名を指定します。先頭が `_mp` で始まる 8 文字以内の英数字で指定してください。このオペランドの指定値にはマッピングサービス定義ファイル名を指定してください。

マッピングサービスの定義の詳細については、「6. OpenTP1 マッピングサービス機能」の「OpenTP1 マッピングサービスに関する定義」を参照してください。

●-o

(オペランド)

cmdsvname=オンラインコマンドサービス名 ~ 〈1~8 文字の英数字〉

ACT ALTSSEND コマンド、DEACT ALTSSEND コマンドを使用する場合、MCF オンラインコマンドサービスのサービス名を指定します。先頭が `_mcs` で始まる 8 文字以内の英数字で指定してください。

注意事項

-o オプションは、OS が Windows の場合、指定できません。

mcftalccn (コネクション定義の開始)

形式

```
mcftalccn  -c コネクションID
           -p xp
           -g "sndbuf=メッセージ送信用バッファグループ番号
              rcvbuf=メッセージ受信用バッファグループ番号"
           [-e "msgbuf=メッセージ編集用バッファグループ番号
              count=メッセージ編集用バッファ数"]
           [-i auto | manual]
           [-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] "]
           -S "svname=接続対象の表示印刷サービス名
              type=DSP | PRT"
```

機能

コネクションに関する環境を定義します。

オプション

●-c コネクション ID ～ 〈1～8 文字の識別子〉

OpenTP1 システム内で、一意となるコネクション ID を指定します。

●-p xp

プロトコルの種別を指定します。

xp

XMAP3 プロトコル

●-g

(オペランド)

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

メッセージ送信用バッファグループ番号を指定します。

バッファグループ定義 (mcftbuf -g) の groupno オペランドに指定されているバッファグループ番号を指定してください。

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

メッセージ受信用バッファグループ番号を指定します。

バッファグループ定義 (mcftbuf -g) の groupno オペランドに指定されているバッファグループ番号を指定してください。

●-e

(オペランド)

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

入力、出力メッセージ編集 UOC を使用する場合、メッセージ編集用として使用するバッファグループ番号を指定します。このオペランドを省略した場合は、メッセージ編集用バッファは確保されません。バッファグループ定義 (mcftbuf -g) の groupno オペランドに指定されているバッファグループ番号を指定してください。

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

入力、出力メッセージ編集 UOC を使用する場合、メッセージ編集用として使用するバッファの数を指定します。

ここで指定したバッファ数が、UOC に編集バッファとして渡されるので、最大セグメント分割数を指定してください。

TP1/NET/XMAP3 では、ここに指定した数を入力、出力メッセージ編集 UOC に同時に割り当てることがあるので、ここで指定したバッファグループを定義したバッファグループ定義 (mcftbuf -g) の count、および extend オペランドでの指定値の和は、メッセージ編集用バッファ数の 2 倍を確保してください。複数接続でバッファグループを共用する場合は接続ごとの和を指定してください。

msgbuf オペランドを省略した場合は、このオペランドの指定は無効です。

●-i auto | manual ~ 〈manual〉

OpenTP1 システム開始時および再開始時の、接続の確立方法を指定します。

auto

OpenTP1 システム開始時および再開始時に接続を自動的に確立します。

manual

MCF 起動後、接続を確立します。接続の確立は、運用コマンド (mcftactcn) の入力、または API (dc_mcf_tactcn 関数) の発行で行います。

●-b

(オペランド)

bretry=yes | no ~ 〈yes〉

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

yes

接続の確立再試行をします。

no

接続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)) 《3》 (単位 : 回)

コネクション確立障害時の確立再試行回数を指定します。

このオペランドを省略した場合は、3 が設定されます。

0 を指定した場合、無限に確立再試行を繰り返します。

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

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)

コネクション確立障害時の確立再試行間隔を指定します。

0 を指定した場合、障害が発生するたびにコネクションの確立再試行をします。

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

注意事項

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

●-S

(オペランド)

svname=接続対象の表示印刷サービス名 ~ 〈1~14 文字の識別子〉

接続対象の表示印刷サービス名として XMAP3 Server サービス名ファイルで指定したサービス名を指定します。

type=DSP | PRT

接続対象の表示印刷サービスのサービスタイプを指定します。

DSP

表示サービス

PRT

印刷サービス

PRT を指定した場合、MCF 通信構成定義の mcftalccn 定義コマンドと mcftalcle 定義コマンドとの関係は、1 対 1 です。

注意事項

-g オプション、および-e オプションで指定するバッファグループ番号は、バッファグループ定義 (mcftbuf) で指定したバッファグループ番号に対応させてください。バッファグループ定義 (mcftbuf) でコネクションごとに割り当てる資源の量を次の表に示します。バッファグループ定義 (mcftbuf) の詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

表 7-3 バッファグループ定義 (mcftbuf) でコネクションごとに割り当てる資源の量

オプション・オペランド		バッファグループ定義 (mcftbuf) -g オプション	
		length ^{※1}	count + extend ^{※2}
-g	sndbuf	次に示すどちらか大きい方の値 <ul style="list-style-type: none"> • (物理マップのファイルサイズ+メッセージ長^{※3}) のコネクション間での最大値 • 出力論理メッセージ長^{※4} のコネクション間での最大値 	コネクション数×最大セグメント分割数+ 1
	rcvbuf	次に示すどちらか大きい方の値 <ul style="list-style-type: none"> • 受信物理メッセージ長^{※5} のコネクション間での最大値 • 入力論理メッセージ長^{※6} のコネクション間での最大値 	(コネクション数+ 1) ×最大セグメント分割数
-e	msgbuf	次に示すうちの最大値 <ul style="list-style-type: none"> • (物理マップのファイルサイズ+メッセージ長^{※3}) のコネクション間での最大値 • 出力論理メッセージ長^{※4} のコネクション間での最大値 • 受信物理メッセージ長^{※5} のコネクション間での最大値 • 入力論理メッセージ長^{※6} のコネクション間での最大値 	編集用バッファ数×2

注※1

複数コネクションでバッファグループを共用する場合、コネクション間での最大の値を指定します。

注※2

複数コネクションでバッファグループを共用する場合、コネクションごとの和を指定します。

注※3

XMAP3 のドローでマップ生成時に出力される結果リスト中の「メッセージ長」です。

注※4

XMAP3 のドローでマップ生成時に出力される結果リスト中の「出力マップ長」です。

注※5

次の算出式で求められる値になります。

$256 + (\text{入力マップ長} \times 3) + \text{入力論理マップのデータ項目数}$

ただし、値が 32768 を超える場合は、32768 となります。

注※6

XMAP3 のドローでマップ生成時に出力される結果リスト中の「入力マップ長」です。

mcftalced (コネクション定義の終了)

形式

```
mcftalced
```

機能

コネクションの定義の終了を示します。

オプション

ありません。

mcftalcle (論理端末定義)

形式

```
mcftalcle  -l 論理端末名称
           -t any
           [-m " [mmsgcnt=メモリ出力メッセージ最大格納数]
             [dmsgcnt=ディスク出力メッセージ最大格納数] "]
           [-k " [quekind=memory | disk]
             [quegrpid=キューグループID] "]
           [-G " [groupsend=yes | no] "]
           [-o " [aj=yes | no] "]
           [-p " [dflmap='デフォルトマップ名']
             [initmap='イニシャルサイズマップ名'] "]
           [-i auto | manual]
           [-a " [msgadv=auto | manual] "]
           [-r " [rretry=yes | no]
             [rretrycnt=プリンタ排他再試行回数]
             [rretryint=プリンタ排他再試行間隔] "]
           [-g " [title=on | off]
             [titlname='論理端末画面のタイトル名']
             [maxbutton=on | off]
             [minbutton=on | off]
             [resize=yes | no]
             [icon=yes | no]
             [iconname='論理端末画面のアイコン名']
             [pheight=論理端末画面の縦サイズ]
             [pwidth=論理端末画面の横サイズ]
             [wheight=論理端末画面視野枠の縦サイズ]
             [wwidth=論理端末画面視野枠の横サイズ] "]
           -z "device='機種・モデル'"
           [-y 送信完了確認メッセージ数]
           [-x "excnt=プリンタ連続使用メッセージ数"]
           [-K " [keyboard=atype | aplock]
             [unlock=auto | send] "]
```

機能

論理端末に関する環境を定義します。

オプション

●-l 論理端末名称 ~ <1~8文字の識別子>

OpenTP1 システム内で、一意となる論理端末名称を指定します。

●-t any

この論理端末の端末タイプを指定します。

any

任意型論理端末

●-m

(オペランド)

mmsgcnt=メモリ出力メッセージ最大格納数 ~ 〈符号なし整数〉 ((0~65535)) 《0》

メモリキューで待ち合わせをする出力メッセージの最大格納数を指定します。

出力メッセージの待ち合わせ数が指定した最大数になると、それ以後、UAPからの送信要求 (SEND) はエラーリターンとなります。

0を指定した場合、または省略した場合、メモリキューで待ち合わせをする出力メッセージの数は指定可能な最大数 (65535) になります。ただし、実際に待ち合わせができる出力メッセージ数は動的共用メモリの容量に依存します。

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

ディスクキューで待ち合わせをする出力メッセージの最大格納数を指定します。

出力メッセージの待ち合わせ数が指定した最大数になると、それ以後、UAPからの送信要求 (SEND) はエラーリターンとなります。

0を指定した場合、または省略した場合、ディスクキューで待ち合わせをする出力メッセージの数は指定可能な最大数 (65535) になります。ただし、実際に待ち合わせができる出力メッセージ数はメッセージキューファイルの容量に依存します。

●-k

(オペランド)

quekind=memory | disk ~ 《memory》

送受信メッセージの割り当て先 (メモリキュー、またはディスクキュー) を指定します。

memory

メモリキューだけに割り当てます。

disk

ディスクキュー、およびメモリキューに割り当てます。

quegrpID=キューグループ ID ~ 〈1~8文字の識別子〉

ディスクで待ち合わせる送受信メッセージに使用するキューグループ ID を指定します。入出力キュー定義 (mcfmqgid) で指定するキューグループ ID (キューグループ種別は otq) のどれかを指定してください。

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

●-G

(オペランド)

groupsend=yes | no ~ 《no》

メッセージをグループ送信するかどうかを指定します。

yes

グループ送信します。

no

グループ送信しません。

●-o

(オペランド)

aj=yes | no ~ 《yes》

送信完了の情報の取得の要否を指定します。

yes

取得します。

no

取得しません。

●-p

(オペランド)

dflmap='デフォルトマップ名' ~ 〈1~6文字の記号名称〉

UAPからのメッセージ送信時、マップ名を指定しない場合に使用するデフォルトマップの名称を指定します。

initmap='イニシャライズマップ名' ~ 〈1~6文字の記号名称〉

論理端末画面生成時に表示する初期画面用のマップの名称を指定します。

●-i auto | manual ~ 《auto》

コネクション確立時の、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、システムが自動的に論理端末を閉塞解除します。

manual

運用コマンド (mcftactle) の入力、または API (dc_mcf_tactle 関数) の発行で論理端末を閉塞解除します。

●-a

(オペランド)

msgadv=auto | manual ~ 《auto》

メッセージの送信スケジュールを指定します。

auto

現在出力しているメッセージに続いて、次のメッセージを送信します。

manual

XMAP3 共通定義 (mcftxp -m) の msgkey オペランドで指定した次メッセージ要求キーを押したあとに、次のメッセージを送信します。

このオペランドは、コネクション定義 (mcftalccn -S) の type オペランドで PRT を指定したコネクションに属する論理端末に対する論理端末定義 (mcftalcle) では指定できません。

●-r

(オペランド)

rretry=yes | no ~ 《yes》

プリンタ排他再試行をするかどうかを指定します。

yes

プリンタ排他再試行をします。

no

プリンタ排他再試行をしません。

rretrycnt=プリンタ排他再試行回数 ~ 〈符号なし整数〉 ((0~65535)) 《0》

プリンタ排他再試行回数を指定します。

0 を指定した場合、プリンタ排他再試行回数は無制限になります。

rretry オペランドの指定値が no の場合、このオペランドの指定は無効です。

rretryint=プリンタ排他再試行間隔 ~ 〈符号なし整数〉 ((0~255)) 《5》 (単位: 秒)

プリンタ排他再試行間隔を指定します。

0 を指定した場合、障害発生直後にプリンタ排他再試行をします。

rretry オペランドの指定値が no の場合、このオペランドの指定は無効です。

注意事項

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

●-g

(オペランド)

title=on | off

論理端末画面のタイトルバーを表示するかどうかを指定します。省略時は XMAP3 Server の仮定値に従います。

on

タイトルバー表示あり

off

タイトルバー表示なし

titlename='論理端末画面のタイトル名' ~ 〈1~80文字の文字列〉

論理端末画面のタイトル名を指定します。

title オペランドの指定値が off の場合、このオペランドは指定できません。

省略時は XMAP3 Server の仮定値に従います。

maxbutton=on | off

論理端末画面の最大化ボタンを表示するかどうかを指定します。

title オペランドの指定値が off の場合、このオペランドは指定できません。

省略時は XMAP3 Server の仮定値に従います。

on

最大化ボタン表示あり

off

最大化ボタン表示なし

minbutton=on | off

論理端末画面の最小化ボタンを表示するかどうかを指定します。

title オペランドの指定値が off の場合、このオペランドは指定できません。

省略時は XMAP3 Server の仮定値に従います。

on

最小化ボタン表示あり

off

最小化ボタン表示なし

resize=yes | no

論理端末画面の表示の大きさを変更できるようにするかどうかを指定します。

省略時は XMAP3 Server の仮定値に従います。

yes

論理端末画面の表示の大きさを変更できる

no

論理端末画面の表示の大きさを変更できない

icon=yes | no

論理端末画面をアイコン表示できるようにするかどうかを指定します。

省略時は XMAP3 Server の仮定値に従います。

yes

論理端末画面のアイコン表示ができる

no

論理端末画面のアイコン表示ができない

iconname='論理端末画面のアイコン名' ~ 〈1~16 文字の文字列〉

論理端末画面のアイコン名を指定します。

省略時は XMAP3 Server の仮定値に従います。

pheight=論理端末画面の縦サイズ ~ 〈符号なし整数〉 ((24~127))

論理端末画面の縦サイズを指定します。

省略時は XMAP3 Server の仮定値に従います。

pwidth=論理端末画面の横サイズ ~ 〈符号なし整数〉 ((80~255))

論理端末画面の横サイズを指定します。

省略時は XMAP3 Server の仮定値に従います。

wheight=論理端末画面視野枠の縦サイズ ~ 〈符号なし整数〉 ((24~48))

論理端末画面視野枠の縦サイズを指定します。

省略時は XMAP3 Server の仮定値に従います。

wwidth=論理端末画面視野枠の横サイズ ~ 〈符号なし整数〉 ((4~160))

論理端末画面視野枠の横サイズを指定します。

省略時は XMAP3 Server の仮定値に従います。

●-z

(オペランド)

device='機種・モデル' ~ 〈6 文字の記号名称〉

表示・印刷する画面・帳票に対応する 6 文字の文字列を指定します。

XDSPS1

CUI 画面

XDSPM1

GUI 画面

XPRTL1

けい線帳票

XPRTL3

プレプリント帳票

XP RTP1

網掛け帳票

XP RTP3

グラフィック帳票

●-y 送信完了確認メッセージ数 ~ 〈符号なし整数〉 ((1~256)) 〈10〉

非同期送信をする場合の送信完了確認メッセージ数を指定します。

TP1/NET/XMAP3 は、このオプションで指定されたメッセージ数のメッセージを送信すると送信完了処理します。

しかし、障害などの発生時、または代行送信の終了時に、メッセージが二重出力されることがあります。二重出力を避けたい場合は、送信完了確認メッセージ数に 1 を指定してください。

コネクション定義 (mcftalccn -S) の type オペランドの指定が DSP の場合、このオプションは指定できません。また、送信完了処理を行うまでは、すでに送信したメッセージがあっても、それらは出力キューに残しておきます。このため、`quels` コマンド等で出力キューの状態を参照すると、仕掛りメッセージとして残っていますが、異常ではありません。

●-X

(オペランド)

`excnt=プリンタ連続使用メッセージ数` ~ 〈符号なし整数〉 ((0, 5~256)) 《0》

プリンタの排他を解除するタイミングを、送信メッセージの数で指定します。プリンタの排他を続けることでプリンタを継続して使用できます。TP1/NET/XMAP3 は、指定数に達したときに送信しているグループの、最終のメッセージを送信し終わると送信完了処理し、プリンタの排他を解除します。0 を指定した場合、グループごとに送信完了処理し、プリンタの排他を解除します。

両面印刷する場合に 0 以外の値を指定していると、前のグループの裏面に印刷されることがあります。グループの先頭メッセージに適切なページ制御情報を付けてください。

コネクション定義 (mcftalccn -S) の type オペランドの指定が DSP の場合、このオプションは指定できません。また、-G オプションの `groupsend` オペランドの指定値が `no` の場合も、このオプションを指定できません。

●-K

(オペランド)

`keyboard=atype | aplock` ~ 《atype》

論理端末から非応答型 MHP にメッセージを送信したときの、論理端末画面のキーボード状態を指定します。

`atype`

キーボード状態をロック解除状態にします。

`aplock`

キーボード状態をロック状態にします。

非応答型 MHP 以外のアプリケーションにメッセージを送信した場合のキーボード状態については、[\[2.4.10\(3\) キーボード状態の変更\]](#) を参照してください。

`unlock=auto | send` ~ 《auto》

`keyboard` オペランドで `aplock` を指定した場合に、キーボードロックを解除するタイミングを指定します。`keyboard` オペランドに `aplock` を指定している場合だけ、`unlock` オペランドを指定できます。

auto

非応答型 MHP が終了したときに、キーボードロックを解除します。

send

論理端末に一方送信メッセージを送信したときに、キーボードロックを解除します。

注意事項

GUI 画面の場合、-g オプションの次のオペランドが無効になります。

- maxbutton
- resize
- pheight
- pwidth
- wheight
- wwidth

mcftxp (XMAP3 共通定義)

形式

```
mcftxp [-m " [msgkey='次メッセージ要求キー名']
        [timeoutlog=yes | no] "]
        [-o " [opcolor=色番号]
        [opreverse=yes | no] "]
        -n マッピングサービス識別子
        [-g " [closebutton=on | off] "]
        [-c " [actnaction=serial | parallel] "]
```

機能

TP1/NET/XMAP3 に関する環境を定義します。

オプション

●-m

(オペランド)

msgkey='次メッセージ要求キー名' ~ 〈1~8文字の記号名称〉 ((PF1~PF24, PA1~PA3, BREAK))
《PF2》

次メッセージ要求キー名を指定します。ここで指定したキーを押すことで、次に表示するメッセージを送信します。

PF1~PF24, PA1~PA3

それぞれのキーが押されたときに、次に表示するメッセージを送信します。

BREAK

中断キー、または Ctrl+S キーが押されたときに、次に表示するメッセージを送信します。

timeoutlog=yes | no ~ 《yes》

XMAP3 Server のタイムアウト発生を通知するかどうかを指定します*。

yes

タイムアウト発生を通知するログメッセージを出力します。

メッセージ KFCA13662-E (エラーコード= 10 (TX_CECDOWN)) で、詳細エラーコード= 0x03020202, 0x03020303, または 0x03020203) と、直前にエラー要因を示すメッセージを出力します。

no

タイムアウト発生を通知するログメッセージを出力しません。

注※

システムサービス情報定義に指定する XMAP3 Server の環境変数 XRESPONSETIME または環境変数 XPCLOSEWAITTIME によってタイムアウトが発生した場合に、通知するかどうかを指定します。

●-o

(オペランド)

opcolor=色番号 ~ 〈符号なし整数〉 ((1~15)) 《2》

オペレータインジケータのユーザ表示フィールドの表示色を色番号で指定します。

色番号に対応する表示色は、XMAP3 サーバの定義で決定します。詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

opreverse=yes | no ~ 《no》

オペレータインジケータのユーザ表示フィールドの反転表示の有無を指定します。

yes

反転表示あり

no

反転表示なし

●-n マッピングサービス識別子 ~ 〈1~8 文字の識別子〉

入出力マッピングで使用するマッピングサービスの識別子を指定します。

●-g

(オペランド)

closebutton=on | off ~ 《off》

論理端末画面の閉じるボタンを有効にするかどうかを指定します。論理端末画面の閉じるボタンについては、「2.3.5 論理端末画面の閉じるボタンの有効化」を参照してください。

on

閉じるボタンが有効になります。閉じるボタンを押すと、アプリケーション決定後に TP1/NET/XMAP3 が UAP を起動します。

off

閉じるボタンが無効になります。閉じるボタンを押すと、XMAP3 クライアント端末側で無視します (入力エラーとしてブザーを鳴らします)。

●-c

(オペランド)

actcnaction=serial | parallel ~ 《serial》

接続の確立の動作モードを指定します。

serial

コネクション確立要求を逐次処理します。

コネクション確立要求を行う運用コマンドまたは API がリターンしたとき、MCF 通信プロセスで正常に受け付けられていた場合、コネクションは確立状態となっています。

運用コマンドを多数入力する場合には、順番待ちが発生して、入力数が増えるほど、待ち時間が拡大します。

parallel

コネクション確立要求を並行処理します。

コネクション確立要求を行う運用コマンドまたは API がリターンしたとき、MCF 通信プロセスで正常に受け付けられていたとしても、コネクションの状態が確立状態とは限りません。コネクション確立要求を行う運用コマンドまたは API がリターンしたあとに、コネクションが確立していることを前提とした処理がある場合は、コネクションの状態確認を行う運用コマンドまたは API で、コネクションの状態を確認してください。

また、最大処理多重度定義の最大処理多重度 (mcfttred -m) には、負荷試験を実施したうえで最適な値を指定するようにしてください。多重処理状況は運用コマンド (mcftlstrd) で確認できます。運用コマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

システムサービス情報定義

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

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

システムサービス情報定義は、テキストエディタを使用して作成します。

システムサービス情報定義の完全パス名を次に示します。

`$DCDIR/lib/sysconf/定義ファイル名`

定義ファイル名には、システムサービス情報定義の `module` オペランドで指定する実行形式プログラム名を指定します。この定義ファイル名を通信サービス定義 (`mcfmcname`) に指定してください。

形式

set 形式

```
set module="TP1/NET/XMAP3の実行形式プログラム名"  
[set mcf_prf_trace=Y|N]
```

putenv 形式

```
[putenv TZ 時間帯情報]  
[putenv XRESPONSETIME 監視時間]  
[putenv XPCLOSEWAITTIME XMAP3印刷サービス監視時間]  
[putenv XPW_DAEMON_PORT_NO サービス名]
```

機能

プロセスサービスが MCF 通信サービスを起動するための環境を定義します。

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

説明

set 形式で指定するオペランドと、putenv 形式で指定するオペランドがあります。

ここでは、それぞれのオペランドについて、分けて説明します。

set 形式のオペランド

set 形式で指定するオペランドについて、説明します。

●module="TP1/NET/XMAP3 の実行形式プログラム名" ~ 〈1~8 文字の識別子〉

MCF 通信サービスを起動するための実行形式プログラム名を指定します。

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

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

TP1/NET/XMAP3 の実行形式プログラム名には、先頭 4 文字が mcfu で始まる最大 8 文字の名称を指定してください。

●mcf_prf_trace=Y|N ~ 〈Y〉

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

Y

MCF 性能検証用トレース情報を取得します。

N

MCF 性能検証用トレース情報を取得しません。

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

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

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

putenv 形式のオペランド

putenv 形式で指定するオペランドについて、説明します。

●TZ 時間帯情報 (タイムゾーン)

MCF 通信サービスが動作するプロセスの環境変数 TZ を指定します。省略した場合、プロセスサービスのプロセスの環境変数 TZ を引き継ぎます。

日本標準時の場合、JST-9 を指定してください。

マッピングサービス機能は、この環境変数に従って予約項目の日付、および時刻を設定します。

●XRESPONSETIME 監視時間 ~((0, 25~300)) <25> (単位：秒)

クライアント PC/WS の応答時間を監視する場合に指定します。ただし、XMAP3 印刷サービスからの応答待ち時間は、環境変数 XPCLOSEWAITTIME に指定します。

監視時間には 0 または 25~300 の値を指定してください。0 を指定した場合、応答を監視しません。301 以上の値が指定された場合、または値を省略した場合は、25 を省略時解釈値とし、応答を監視します。TP1/NET/XMAP3 を使用した対話オンライン処理システムでは、このオペランドを設定して、クライアント PC/WS の応答時間を監視することを推奨します。

ただし、タイマ値の設定によっては、運用コマンドのタイムアウトが頻発する場合がありますので注意してください。

詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

●XPCLOSEWAITTIME XMAP3 印刷サービス監視時間 ~((0, 25~86400)) <0> (単位：秒)

XMAP3 印刷サービスの応答時間を監視する場合に指定します。監視時間には 0 または 25~86400 の値を指定してください。0 を指定した場合、応答を監視しません。1~24、および 86401 以上の値が指定された場合、または値を省略した場合は、0 を省略時解釈値とし、応答を監視しません。帳票を印刷するシステムでは、このオペランドを設定して、XMAP3 印刷サービスの応答時間を監視することを推奨します。

指定値は、システム環境定義の system_terminate_watch_time オペランドの指定値を超えないように設定してください。

詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

●XPW_DAEMON_PORT_NO サービス名

1 台のホスト上で複数の XMAP3 サーバを起動する場合、対応する XMAP3 サーバの services ファイルに定義したサービス名を指定します。

詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

システムサービス共通情報定義

TP1/NET/XMAP3 で定義したシステム構成の内容（使用するコネクション数など）によっては、OpenTP1 のシステムサービス共通情報定義を指定する必要があります。

システムサービス共通情報定義の完全パス名を次に示します（適用 OS が Windows の場合を除く）。

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

形式

set 形式

```
set max_socket_descriptors=ソケット用ファイル記述子の最大数
set max_open_fds=MCF通信プロセスでアクセスするファイルの最大数
[set mcf_prf_trace_level=MCF性能検証用トレース情報の取得レベル]
```

機能

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

説明

set 形式のオペランド

この定義には、ほかにもオペランドがあります。詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

●max_socket_descriptors=ソケット用ファイル記述子の最大数 ～〈符号なし整数〉((64~3596))

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

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

各 MCF 通信プロセスまたはアプリケーション起動プロセスが使用するソケット用ファイル記述子の最大数を求める計算式を次に示します。

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

(凡例)

↑↑：小数点以下を切り上げます。

注※1

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

注※2

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

自 OpenTP1 内の各 MCF 通信プロセスおよびアプリケーション起動プロセスごとに計算し、その結果の中で最大値が 64 より大きい場合は、その値を指定します。64 以下の場合は、64 を指定します。

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

●max_open_fds=MCF 通信プロセスでアクセスするファイルの最大数 ~ 〈符号なし整数〉 ((500~4032))

各 MCF 通信プロセスでアクセスするファイル数の中の最大値を指定します。

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

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

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

注※1

TP1/NET/XMAP3 の場合、コネクションの総数を 2 倍した値になります。

注※2

MCF 通信プロセスが扱う定義ファイルなどの数の最大値です。

自 OpenTP1 内の MCF 通信プロセスごとに計算し、その結果の中で最大値が 500 より大きい場合は、その値を指定します。500 以下の場合は、500 を指定します。指定値を超えてファイルのアクセスが発生した場合は、その超過分はソケット用ファイル記述子使用数として扱われます。この場合、「max_socket_descriptors オペランドの指定値-max_open_fds オペランドの指定値の超過分」が、実際のソケット用ファイル記述子の最大数になるため、注意してください。

max_socket_descriptors オペランドと max_open_fds オペランドには次の条件を満たす値を指定してください。

(「max_socket_descriptorsオペランドの指定値」
+「max_open_fdsオペランドの指定値」)
≤4096

ただし、TP1/NET/XMAP3 の MCF 通信プロセスで使用できるファイル記述子の最大数は、適用 OS によって次のように異なります。

OS	1 プロセスで使用できるファイル記述子の最大数
AIX, HP-UX, Windows	2048
Solaris	1024

TP1/NET/XMAP3 の MCF 通信プロセスで、max_socket_descriptors オペランドと max_open_fds オペランドの和が 1 プロセスで使用できるファイル記述子の最大数を超過している場合、TP1/NET/XMAP3 の MCF 通信プロセスで使用できるファイル記述子数は、1 プロセスで使用できるファイル記述子の最大数に強制的に補正されます。

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

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

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

00000000

MCF 性能検証用トレース情報を取得しません。

00000001

MCF 性能検証用トレース情報（イベント ID：0xa000~0xa0ff）を取得します。イベント ID の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。また、TP1/NET/XMAP3 固有の出力情報や取得タイミングについては、「付録 D MCF 性能検証用トレースの取得」を参照してください。

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

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

注意事項

- max_socket_descriptors オペランドの指定値と max_open_fds オペランドの指定値の合計は、OS のシステムパラメタで指定する「1 プロセスでオープンできるファイル数」を超えないようにする必要があります。超える場合は、OS のシステムパラメタの指定を変更してください。

- Windows 版のバージョン 5 ではシステムサービス共通情報定義の `max_socket_descriptors` オペランドにプロトコル制御で使用するファイル記述子数（コネクション数×2）を追加する必要がありましたが、バージョン 7 では UNIX 版と同様に `max_open_fds` オペランドにプロトコル制御で使用するファイル記述子を追加してください。

MCF 定義オブジェクトの生成

MCF 定義オブジェクト生成ユーティリティでは、MCF の定義ファイルの構文のチェックと定義オブジェクトファイルへの変換をします。ここでは、MCF 定義オブジェクト生成ユーティリティの起動コマンドについて説明します。

形式

```
mcfxp  -i [パス名] 入力ファイル名
        -o [パス名] 出力オブジェクトファイル名
        [-r {no | rep} ]
```

機能

MCF 通信構成定義の TP1/NET/XMAP3 のプロトコル固有定義ファイルの構文をチェックし、定義オブジェクトファイルを作成します。

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

TP1/NET/XMAP3 のプロトコル固有定義オブジェクトファイル以外の生成ユーティリティについては、マニュアル「OpenTP1 システム定義」を参照してください。

オプション

●-i [パス名] 入力ファイル名 ~ <パス名> <1~8 文字の識別子>

定義ソースが格納されているファイル名を指定します。

●-o [パス名] 出力オブジェクトファイル名 ~ <パス名> <1~8 文字の英数字>

定義オブジェクトを格納するファイル名を指定します。

次に示す条件を満たした名称を指定してください。

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

●-r {no | rep} ~ <no>

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

no

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

rep

定義オブジェクトファイルを上書きします。

MCF 定義オブジェクトの解析

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

形式

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

機能

TP1/NET/XMAP3 のプロトコル固有定義オブジェクトファイル（または、これと共通定義を結合した MCF 通信構成定義オブジェクトファイル）を解析し、定義ソースの形式で標準出力します。

オプション

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

定義オブジェクトが格納されているファイル名を指定します。

出力例

```
#####
MCF communication configuration definition
XMAP3 definition
#####
OBJECT FILE NAME : xxxxxxxx
VV-RR             : vv-rr
DATE              : yyyy-mm-dd hh:mm:ss
#####
mcfxpr
-m msgkey        = 'PF11'
-m timeoutlog    = yes
-o opcolor       = 2
-o opreverse     = no
*-o opfield1     = 0
*-o opfield2     = 0
*-o opfield3     = 0
*-o opfield4     = 0
-n               = mapserv1

.
.
.

mcftalccn
-c               = XPNCN1
-p               = xp
-g sndbuf        = 1
-g rcvbuf        = 2
-e msgbuf        = 3
-e count         = 5
```

```

-i          = manual
-b bretry  = no
-b bretrycnt = 3
-b bretryint = 60
*-b dretry = no
-S svname  = xppDSP01
-S type    = DSP
*-w ercvtim = 60
mcfatalcle
-l         = XPNLE101
-t         = any
-m mmsgcnt = 0
-m dmsgcnt = 0
-k quekind = memory
-o aj      = yes
-G groupsend = no

.
.
.

mcfatalcle
-l         = XPNLE102

.
.
.

mcfatalced

.
.
.

##### End Of File #####

```

解析結果

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

表 7-4 定義ソースと定義オブジェクト解析結果の差異

項目	定義ソース	定義オブジェクト解析結果
注釈文	書き込みできる。	出力しない。
省略値の扱い	省略できる。	限定公開部分も含めて、省略値を出力する。
限定公開部分の表記方法	一般公開部分と差異なし。	バージョン 7 での限定公開機能の行の先頭に、"*"を付与する。
定義コマンド名とオプションの表記方法	1 行に表記できる。	定義コマンド名を表記後、改行する。また、オプションに"="を付記する。

項目	定義ソース	定義オブジェクト解析結果
定義コマンド名とオプションの表記方法	(例) mcftalccn -c XPNCN1	(例) mcftalccn -c = XPNCN1
1 定義コマンドが複数の行にわたる場合	継続記号"¥"を付与する。 (例) mcftalccn -c XPNCN1 ¥ -p xp	継続記号は出力しない。 (例) mcftalccn -c = XPNCN1 -p = xp
1 定義オプションに複数のオペランドを指定する場合	複数のオペランドをまとめて二重引用符(")で囲む。 (例) mcftalccn -g "sndbuf=1 rcvbuf=2"	個々のオペランドに対してオプションを付記する。 (例) mcftalccn -g sndbuf = 1 -g rcvbuf = 2
その他	なし	<ul style="list-style-type: none"> ファイル名などを記したタイトルが出力される。 定義オブジェクト生成時の補正によって、実際の指定値とは異なる内容が出力される場合がある。 定義ソースと該当コマンドのバージョンの差異によって、解析結果にサポート内容の過不足がある場合がある。

注意事項

解析対象が不正であった場合は、正常に動作しないことがあります。

MCF トレースファイルの見積もり式

ここでは、MCF トレースファイルのトレース情報量の見積もり式、トレース情報が失われる経過時間の見積もり式、および具体的な見積もりの例について説明します。

トレース情報量の見積もり式

1 秒あたりに取得する MCF トレースファイルの、トレース情報量の見積もり式を次に示します。

$$\begin{aligned} & \text{1秒あたりのトレース情報量 (単位: バイト)} \\ & = (A \times B) + (C \times D) + (E \times F) \end{aligned}$$

- A: コネクションの確立と解放時に取得するトレース情報量 (単位: バイト)
 - 画面属性のコネクションの場合: $6800 + 9300 \times \text{コネクション配下の論理端末数}$
 - プリンタ属性のコネクションの場合: 13200
- B: 1 秒あたりのコネクション確立および解放の回数^{※1}
- C: メッセージ受信時に取得するトレース情報量 (単位: バイト)
 - 応答を送信する場合: $4500 + \text{送信する物理マップファイルのサイズ}$
 - 画面に対して一方送信を送信する場合: $4100 + \text{送信する物理マップファイルのサイズ}$
 - プリンタに対して一方送信を送信する場合: $7300 + \text{送信する物理マップファイルのサイズ}$
- D: 1 秒あたりのメッセージ送信回数^{※1}
- E: メッセージ受信時に取得するトレース情報量 (単位: バイト)
 $7100 + \text{入力論理マップの論理マップ長}^{\text{※2}}$
- F: 1 秒あたりのメッセージ受信回数^{※1}

注※1

TP1/NET/XMAP3 の MCF 通信プロセスが複数存在する場合は、MCF 通信プロセス単位で回数を算出してください。

注※2

入力論理マップの論理マップ長については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」を参照してください。

トレース情報が失われる経過時間の見積もり式

MCF トレースファイルから、トレース情報が失われる経過時間の算出式を次に示します。

なお、算出式中の、「1 秒あたりのトレース情報量」とは、トレース情報量の見積もり式で算出した値です。

$$\begin{aligned} & \text{経過時間 (単位: 秒)} \\ & = G \times H \times I / \text{1秒あたりのトレース情報量 (単位: バイト)} \end{aligned}$$

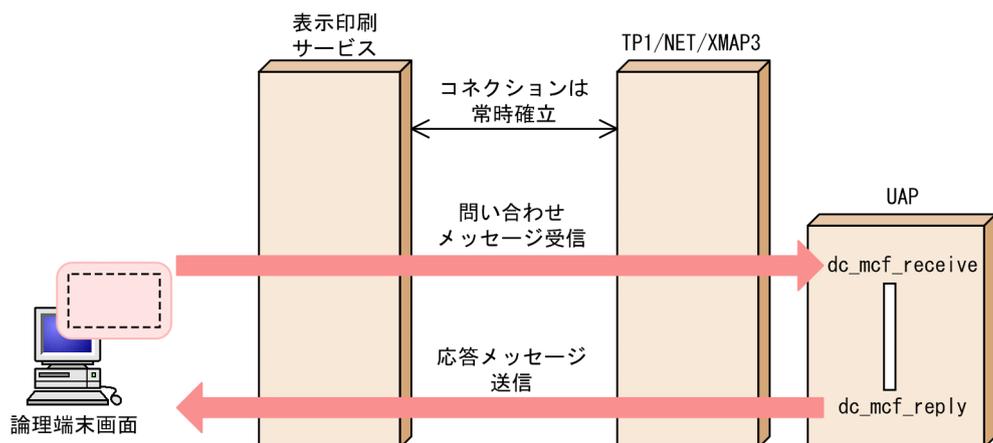
- G：トレースバッファの大きさ (mcftrc -t size)
- H：トレースバッファの数 (mcftrc -t bufcnt)
- I：MCF トレースファイルの数 (mcftrc -t trcnt)

見積もり式の算出例

トレース情報量の見積もり式，およびトレース情報が失われる経過時間の見積もり式の具体的な算出例を示します。

ここでは，常時接続のサーバ型コネクションで，問い合わせ応答をする場合を例に説明します。

図 7-2 常時接続のサーバ型コネクションで，問い合わせ応答をする場合の例



この例では，次の値が想定されています。

項目	想定値
画面属性の論理端末数	1
入力論理マップの論理マップ長	200 バイト
送信する物理マップファイルのサイズ	8000 バイト
1分 (60秒) 当たりのメッセージの受信から送信までの回数	60 回
トレース環境定義 (mcftrc -t) のオペランドの指定値	<ul style="list-style-type: none"> • size = 204800 • bufcnt = 100 • trcnt = 3

この例の場合の，計算例を次に示します。

トレース情報量の見積もり

$$((6800 + 9300 \times 1) \times 0^{\ast}) + ((4500 + 8000) \times (60 / 60)) + ((7100 + 200) \times (60 / 60)) = 19800$$

1 秒当たりのトレース情報量は，19800 バイトとなります。

注※

コネクションが常時接続であるため、1秒当たりのコネクション確立および解放の回数は0として計算してください。

トレース情報が失われる経過時間の見積もり

$$204800 \times 100 \times 3 / 19800 \doteq 3103.0$$

トレース情報が失われる経過時間は、3103.0秒（約51分）となります。

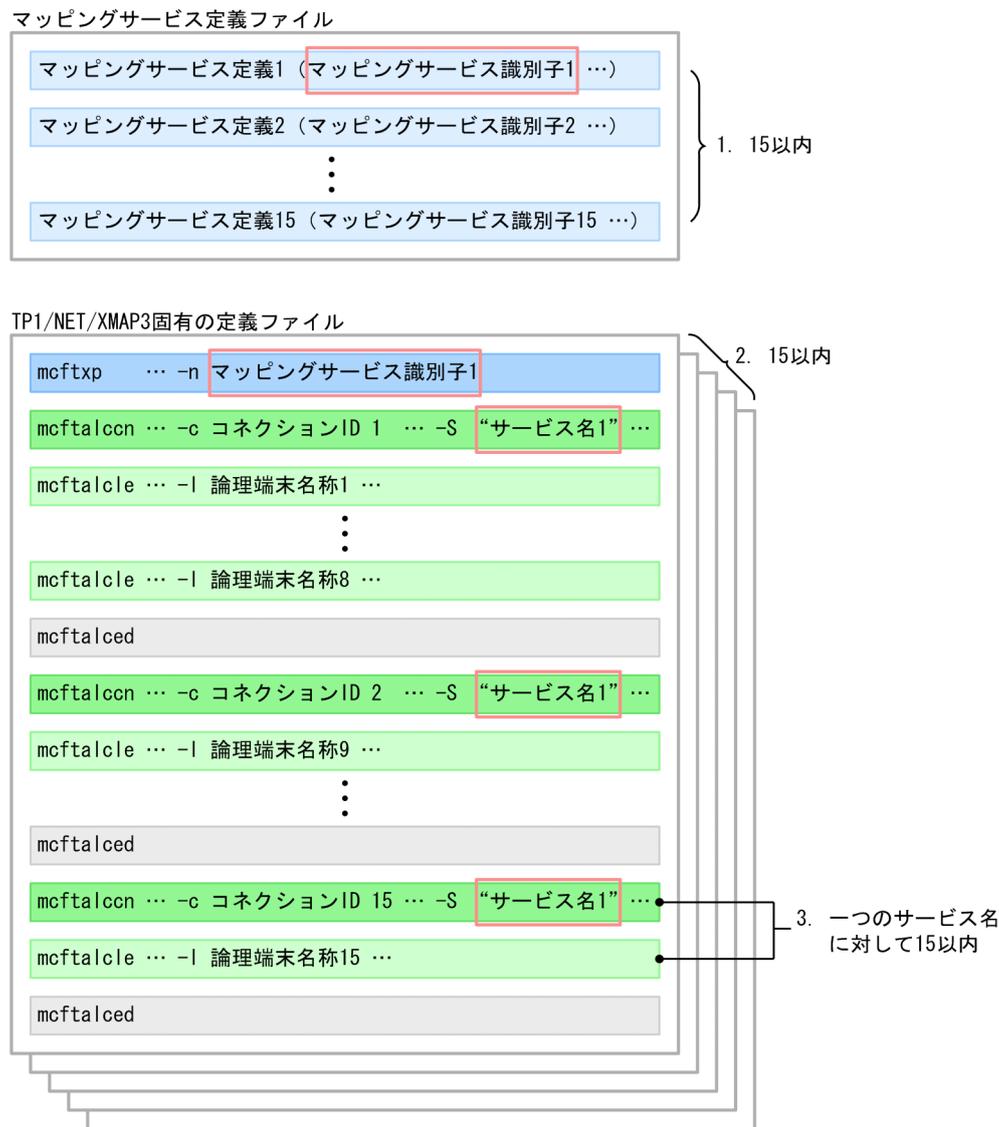
各種定義数に関する制約事項

TP1/NET/XMAP3 では、各種定義の指定数に制約があります。マッピングサービス定義と、TP1/NET/XMAP3 固有の定義では、次の数がそれぞれ 15 以内となっています。

1. マッピングサービス定義の定義数
2. 一つのマッピングサービス識別子を指定できる通信プロセス
3. 一つの表示印刷サービスに対して同時に接続できるコネクション，論理端末の数

これらの定義の関係を、次の図に示します。なお、図中の番号と前述の個条書きの番号は対応しています。

図 7-3 マッピングサービス定義と TP1/NET/XMAP3 固有の定義の関係



OpenTP1 システムの変更に影響する定義

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

ホスト名または IP アドレスの変更

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

ホスト名または IP アドレスを変更する場合に見直しが必要な定義

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

表 7-5 ホスト名または IP アドレスを変更する場合に見直しが必要な定義の一覧

定義ファイル名	定義	見直しが必要な条件
マッピングサービス定義	マッピングサービスホスト名	無条件に見直しが必要
XMAP3 Server サービス名ファイル	ホスト名*	無条件に見直しが必要

注※

詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

ホスト名または IP アドレスの変更手順

ホスト名または IP アドレスは、次の手順で変更してください。

1. OpenTP1 を正常停止します。
2. XMAP3 Server を停止します。
3. マッピングサービス定義および XMAP3 Server サービス名ファイルについて、変更前のホスト名または IP アドレスを検索します。
OS が UNIX の場合は grep コマンド、Windows の場合は findstr コマンドを使用して検索します。
4. 検索の結果、変更前のホスト名または IP アドレスが見つかった場合には、変更します。

コネクション（論理端末）の追加

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

表 7-6 コネクション（論理端末）を追加する場合に見直しが必要な定義の一覧

定義ファイル名	定義	再見積もりが発生する条件
システム環境定義	static_shmpool_size* ¹	無条件に再見積もりが必要

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

注※1

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

注※2

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

注※3

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

注※4

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

注※5

詳細については、「システムサービス共通情報定義」を参照してください。

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

表 7-7 コネクション (論理端末) を追加する場合に見直しが必要な OpenTP1 ファイルの一覧

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

また、TP1/NET/XMAP3 のリリースノートを参照し、MCF 通信プロセスが使用するローカルメモリのメモリ所要量も見直してください。

定義例

TP1/NET/XMAP3 を使用したシステム定義の例を示します。

適用 OS が UNIX の場合のコーディング例は、次のファイルで提供しています。

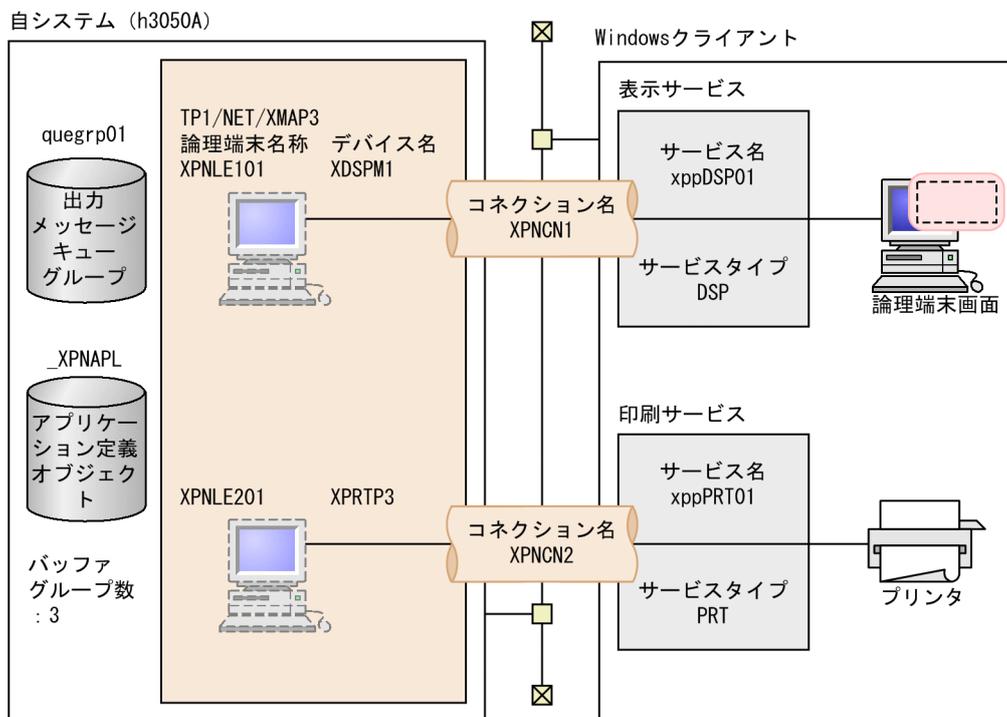
- /BeTRAN/examples/mcf/XMAP3/conf/com_c1
- /BeTRAN/examples/mcf/XMAP3/conf/com_d2

適用 OS が Windows の場合のコーディング例は、次のファイルで提供しています。

- %DCDIR%\examples\mcf\xmap3\conf\source\com_c1
- %DCDIR%\examples\mcf\xmap3\conf\source\com_d2

TP1/NET/XMAP3 システム構成例を次の図に示し、その構成に沿った定義例をそのあとに示します。

図 7-4 TP1/NET/XMAP3 システム構成例



コーディング例 (TP1/NET/XMAP3 の定義)

```
#####
#                MCF通信構成定義(TP1/NET/XMAP3)                #
#####

# ----- MCF環境定義 ----- #
mcfteiv  -s 66                ¥
         -a _XPNAPL

# ----- MCF共通定義 ----- #
```

```

mcftcomn

# ----- トレース環境定義 ----- #
mcfttrc

# ----- 送受信バッファ定義 ----- #
### (送信用)
mcftbuf -g "groupno = 1          ¥
          length = 131072       ¥
          count  = 20"

### (受信用)
mcftbuf -g "groupno = 2          ¥
          length = 4096         ¥
          count  = 10"

### (編集用)
mcftbuf -g "groupno = 3          ¥
          length = 131072       ¥
          count  = 30"

# ----- XMAP3共通定義 ----- #
mcftxp  -m "msgkey = 'PF11'"     ¥
        -n mapserv1

# ----- コネクション定義 ----- #
mcftalccn -c XPNCN1             ¥
          -p xp                  ¥
          -g "sndbuf = 1         ¥
              rcvbuf = 2"       ¥
          -e "msgbuf = 3         ¥
              count  = 1"       ¥
          -i auto                 ¥
          -S "svname = xppDSP01  ¥
              type   = DSP"

# ----- 論理端末定義 ----- #
mcftalcle -l XPNLE101           ¥
          -t any                  ¥
          -p "dflmap = 'MCF1G'   ¥
              initmap = 'MCF1G'" ¥
          -a "msgadv = manual"    ¥
          -g "title = on         ¥
              titlname = 'XPNLE101'" ¥
          -z "device = 'XDSPM1'"

# ----- コネクション定義の終了 ----- #
mcftalced

# ----- コネクション定義 ----- #
mcftalccn -c XPNCN2             ¥
          -p xp                  ¥
          -g "sndbuf = 1         ¥
              rcvbuf = 2"       ¥
          -e "msgbuf = 3         ¥
              count  = 1"       ¥
          -i auto                 ¥
          -S "svname = xppPRT01  ¥

```

```
type = PRT"
```

```
# ----- 論理端末定義 ----- #
```

```
mcftalcle -l XPNLE201 ¥  
          -t any ¥  
          -k "quekind = disk ¥  
            quegrp01" ¥  
          -G "groupsend = yes" ¥  
          -p "dflmap = 'MCF1P'" ¥  
          -z "device = 'XP RTP3'"
```

```
# ----- コネクション定義の終了 ----- #
```

```
mcftalced
```

```
#### 終わり ####
```

8

運用コマンド

この章では、TP1/NET/XMAP3 で使用する運用コマンドについて説明します。

TP1/NET/XMAP3 の運用コマンド

TP1/NET/XMAP3 では、オンライン中に OpenTP1 システム側と端末側からコマンドを入力できます。OpenTP1 システム側から入力するコマンドを運用コマンドと呼びます。

ここでは、TP1/NET/XMAP3 に関係のあるオプションについてだけ説明しています。ほかのオプションおよびその他の運用コマンドについては、マニュアル「OpenTP1 運用と操作」を参照してください。また、コマンド実行時のエラーの対処方法については、マニュアル「OpenTP1 メッセージ」を参照してください。

TP1/NET/XMAP3 で使用する運用コマンドの一覧を次の表に示します。

表 8-1 TP1/NET/XMAP3 で使用する運用コマンドの一覧

機能		コマンド名称	定義中組み込み	オフライン中に実行	オンライン中に実行	UAP から実行
コネクション管理	コネクションの確立	mcftactcn	×	×	○	○
	コネクションの強制解放	mcftdctcn	×	×	○	○
	コネクションの状態表示	mcftlscn	×	×	○	○
論理端末管理	論理端末の閉塞解除	mcftactle	×	×	○	○
	論理端末の閉塞	mcftdctle	×	×	○	○
	論理端末の状態表示	mcftlsle	×	×	○	○
代行送信	代行送信の開始	mcftstalt	×	×	○	○
	代行送信の終了	mcftedalt	×	×	○	○
継続問い合わせ応答処理	継続問い合わせ応答処理の強制終了	mcftendct	×	×	○	○

(凡例)

- ：組み込み、または実行ができます。
- ×

注

各運用コマンドは、入力された単位で順番に処理されます。運用コマンドを多数入力する場合には、順番待ちが発生して、入力数が増えるほど、待ち時間が拡大します。したがって、運用コマンドの入力を業務運用の一部に組み込む場合は、必要最小限の入力数となるように運用設計を行ってください。

mcftactcn (コネクションの確立)

形式

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

機能

コネクションを確立します。

オプション

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

処理対象のコネクションを制御する MCF 通信サービスの MCF 通信プロセス識別子を指定します。MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信サービスに対して、mcftactcn コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

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

●-c コネクション ID ~ 〈1~8 文字の識別子〉

確立するコネクションのコネクション ID を指定します。

コネクション ID は、1 回につき 8 個まで指定できます。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

複数指定する場合は、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。同一コネクション ID は、重複して指定できません。

また、コネクション ID は、*を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定の場合も、引用符 (") で囲んで指定します。

*

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 Δcnn2 Δcnn3"
```

〈一括指定の例〉 cnn で始まるすべての接続を指定する場合

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

出力メッセージ

メッセージ ID	内容	出力先
KFCA10350-I	mcftactcn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	メッセージログファイル, または標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	メッセージログファイル, または標準エラー出力
KFCA10359-W	mcftactcn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftactcn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftactcn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定した接続は登録されていません。	標準エラー出力
KFCA10391-E	mcftactcn コマンドはサポートされていません。	標準エラー出力
KFCA10500-I	ヘルプメッセージ	標準出力
KFCA13604-E	接続確立処理中に障害が発生しました。	標準エラー出力
KFCA13610-E	接続解放処理中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13612-E	接続確立済みのため、運用コマンドは受け付けられません。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

注意事項

mcftactcn コマンドが正常に受け付けられたかどうかは、コマンドのリターン値しないでください。コマンドが出力したメッセージの内容で判断してください。

mcftactle (論理端末の閉塞解除)

形式

```
mcftactle [-s MCF通信プロセス識別子] [-c コネクションID]
          [-l 論理端末名称] [-t 閉塞解除種別]
```

機能

論理端末の閉塞を解除します。

オプション

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

処理対象の論理端末を制御する MCF 通信サービスの MCF 通信プロセス識別子を指定します。MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信サービスに対して、mcftactle コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

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

●-c コネクション ID ~ 〈1~8 文字の識別子〉

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

コネクション ID は複数指定できません。また、一括指定もできません。

●-l 論理端末名称 ~ 〈1~8 文字の識別子〉

閉塞解除する論理端末の名称を指定します。

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

論理端末名称は、1 回につき 8 個まで指定できます。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

複数指定する場合は、引用符 (") で囲んで、論理端末名称と論理端末名称との間を空白で区切ります。同一論理端末名称は、重複して指定できません。

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

*

すべての論理端末の閉塞を解除します。

先行文字列*

先行文字列で始まるすべての論理端末の閉塞を解除します。

〈複数指定の例〉 len1, len2, len3 を指定する場合

```
-l "len1△len2△len3"
```

〈一括指定の例〉 len で始まるすべての論理端末を指定する場合

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

●-t 閉塞解除種別

閉塞解除の種別を指定します。

all : 論理端末の端末状態、およびキュー状態の閉塞を解除します。

term : 論理端末の端末状態の閉塞を解除します。

queue : 論理端末のキュー状態の閉塞を解除します。

このオプションの指定を省略すると、all が設定されます。

出力メッセージ

メッセージID	内容	出力先
KFCA10350-I	mcftactle コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	メッセージログファイル、または標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	メッセージログファイル、または標準エラー出力
KFCA10359-W	mcftactle コマンド入力元への応答を失敗しました。	メッセージログファイル、または標準エラー出力
KFCA10371-I	mcftactle コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftactle コマンドが異常終了しました。	標準エラー出力

メッセージID	内容	出力先
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定されたコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10391-E	mcftactle コマンドはサポートされていません。	標準エラー出力
KFCA10395-E	指定したコネクションには未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10503-I	ヘルプメッセージ	標準出力
KFCA13602-E	論理端末閉塞解除処理で障害が発生しました。	標準エラー出力
KFCA13609-E	論理端末が閉塞解除済みのため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13611-E	コネクションが未確立のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13613-E	論理端末の端末状態が閉塞解除済みのため運用コマンドは受け付けられません。	標準エラー出力
KFCA13615-E	論理端末のキュー状態が閉塞解除済みのため運用コマンドは受け付けられません。	標準エラー出力
KFCA13665-W	コネクション未確立のため論理端末の端末状態は閉塞解除できませんでした。	標準エラー出力
KFCA13694-E	論理端末の端末状態の閉塞解除処理中に障害が発生しました。	標準エラー出力
KFCA13696-E	論理端末のキー状態の閉塞解除処理中に障害が発生しました。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

注意事項

mcftactle コマンドが正常に受け付けられたかどうかは、コマンドのリターン値で判断しないでください。コマンドが出力したメッセージの内容で判断してください。

mcftdctcn (コネクシオンの強制解放)

形式

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

機能

コネクシオンを強制解放します。

なお、継続問い合わせ応答処理中に mcftdctcn コマンドを入力した場合、処理中の MHP が終了したあと、受け付けられます。

オプション

●-f

該当するコネクシオンを無条件に解放します。

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

処理対象のコネクシオンを制御する MCF 通信サービスの MCF 通信プロセス識別子を指定します。MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信サービスに対して、mcftdctcn コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

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

●-c コネクション ID ~ 〈1~8 文字の識別子〉

解放するコネクシオンのコネクション ID を指定します。

コネクション ID は、1 回につき 8 個まで指定できます。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

複数指定する場合は、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。同一コネクション ID は、重複して指定できません。

また、コネクション ID は、* を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定の場合も、引用符 (") で囲んで指定します。

*

すべてのコネクシオンを解放します。

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*"
```

出力メッセージ

メッセージ ID	内容	出力先
KFCA10350-I	mcftdctcn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	メッセージログファイル、または標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	メッセージログファイル、または標準エラー出力
KFCA10359-W	mcftdctcn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftdctcn コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftdctcn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10391-E	mcftdctcn コマンドはサポートされていません。	標準エラー出力
KFCA10501-I	ヘルプメッセージ	標準出力
KFCA13603-E	コネクション解放処理中に障害が発生しました。	標準エラー出力
KFCA13610-E	コネクション解放処理中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13611-E	コネクションが確立されていません。	標準エラー出力
KFCA13686-W	-f オプションなしはサポートしていません。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

注意事項

mcftdctcn コマンドが正常に受け付けられたかどうかは、コマンドのリターン値で判断しないでください。コマンドが出力したメッセージの内容で判断してください。

mcftdctl (論理端末の閉塞)

形式

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

機能

論理端末を閉塞します。ただし、継続問い合わせ応答処理中の論理端末を閉塞することはできません。

オプション

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

処理対象の論理端末を制御する MCF 通信サービスの MCF 通信プロセス識別子を指定します。MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信サービスに対して、mcftdctl コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

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

●-c コネクション ID ~ 〈1~8 文字の識別子〉

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

コネクション ID は複数指定できません。また、一括指定もできません。

●-l 論理端末名称 ~ 〈1~8 文字の識別子〉

閉塞する論理端末の名称を指定します。

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

論理端末名称は、1 回につき 8 個まで指定できます。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

複数指定する場合は、引用符 (") で囲んで、論理端末名称と論理端末名称との間を空白で区切ります。同一論理端末名称は、重複して指定できません。

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

*

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*"
```

●-t 閉塞種別

閉塞の種別を指定します。

all：論理端末の端末状態、およびキュー状態を閉塞します。

term：論理端末の端末状態を閉塞します。

queue：論理端末のキュー状態を閉塞します。

このオプションの指定を省略すると、all が設定されます。

出力メッセージ

メッセージID	内容	出力先
KFCA10350-I	mcftdctl コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	メッセージログファイル、または標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	メッセージログファイル、または標準エラー出力
KFCA10359-W	mcftdctl コマンド入力元への応答を失敗しました。	メッセージログファイル、または標準エラー出力
KFCA10371-I	mcftdctl コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftdctl コマンドが異常終了しました。	標準エラー出力

メッセージ ID	内容	出力先
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定したコネクション ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10391-E	mcftdctl コマンドはサポートされていません。	標準エラー出力
KFCA10395-E	指定したコネクションには未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10504-I	ヘルプメッセージ	標準出力
KFCA13601-E	論理端末閉塞処理で障害が発生しました。	標準エラー出力
KFCA13607-E	論理端末が継続問い合わせ応答中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13608-E	論理端末が閉塞解除されていないため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13614-E	論理端末の端末状態が閉塞解除されていないため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13616-E	論理端末のキー状態が閉塞解除されていないため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13695-E	論理端末の端末状態の閉塞処理中に障害が発生しました。	標準エラー出力
KFCA13697-E	論理端末のキュー状態の閉塞処理中に障害が発生しました。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

注意事項

- 受信仕掛り中に運用コマンド (mcftdctl) を入力した場合は、受信仕掛り中のメッセージは破棄されます。以降の受信メッセージは入力キューに格納されません。
- 送信仕掛り中に運用コマンド (mcftdctl) を入力した場合の処理は、通信形態によって異なります。分岐メッセージのときは送信処理が中断されます。以降の UAP からの送信メッセージは、出力キュー上に格納されます。
応答メッセージのときは、次に論理端末が閉塞解除状態になった時に破棄されます。
- mcftdctl コマンドが正常に受け付けられたかどうかは、コマンドのリターン値で判断しないでください。コマンドが出力したメッセージの内容で判断してください。

mcftedalt (代行送信の終了)

形式

```
mcftedalt [-s MCF通信プロセス識別子] -l 論理端末名称
```

機能

代行送信を終了します。

オプション

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

処理対象の論理端末を制御する MCF 通信サービスの MCF 通信プロセス識別子を指定します。MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信サービスに対して、mcftedalt コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

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

●-l 論理端末名称 ~ (1~8 文字の識別子)

代行を終了する代行送信元の論理端末の名称を指定します。mcftstalt コマンドの-f オプションで指定した論理端末名称を指定してください。

論理端末名称は複数指定できません。また、一括指定もできません。

出力メッセージ

メッセージID	内容	出力先
KFCA10117-I	メッセージの代行送信を終了します。	メッセージログファイル
KFCA10350-I	mcftedalt コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	メッセージログファイル、または標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力

メッセージID	内容	出力先
KFCA10358-E	内部関数のエラーが発生しました。	メッセージログファイル, または標準エラー出力
KFCA10359-W	mcftedalt コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcftedalt コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftedalt コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10391-E	mcftedalt コマンドはサポートされていません。	標準エラー出力
KFCA10542-I	ヘルプメッセージ	標準出力
KFCA13689-E	代行送信終了処理で障害が発生しました。	メッセージログファイル
KFCA13699-E	代行送信終了処理中に障害が発生しました。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

注意事項

mcftedalt コマンドが正常に受け付けられたかどうかは、コマンドのリターン値で判断しないでください。コマンドが出力したメッセージの内容で判断してください。

mcftendct (継続問い合わせ応答処理の強制終了)

形式

```
mcftendct [-f]
           [-s MCF通信プロセス識別子]
           [-l 論理端末名称]
```

機能

継続問い合わせ応答処理を強制終了します。

オプション

●-f

継続問い合わせ応答処理を即時強制終了する場合に指定します。

このオプションを指定すると、直ちに継続問い合わせ用一時記憶領域を解放し、該当する論理端末からの問い合わせを処理している MHP (エラーイベントを含む) は異常終了します。このときエラーイベントは起動しません。

-f オプションを指定した mcftendct コマンドの実行後に MHP が次に示す処理をしたとき、MHP が異常終了します。

- TP1/Message Control 以外のリソースマネージャにもアクセスする MHP のサービス終了時
- 継続問い合わせ用一時記憶領域アクセス時 (dc_mcf_tempget 関数, dc_mcf_tempput 関数発行時)
- ロールバック要求時 (dc_mcf_rollback 関数発行時 (action : DCMCFRTRY, または DCMCFNRTN の場合))

このオプションの指定を省略すると、MHP が該当する論理端末からの問い合わせを処理中の場合、エラーメッセージが出力されます。

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

処理対象の論理端末を制御する MCF 通信サービスの MCF 通信プロセス識別子を指定します。MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信サービスに対して、mcftendct コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

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

●-l 論理端末名称 ~ 〈1~8 文字の識別子〉

継続問い合わせ応答処理中の論理端末名称を指定します。

論理端末名称は、1回につき8個まで指定できます。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

複数指定する場合は、引用符 (") で囲んで、論理端末名称と論理端末名称との間を空白で区切ります。同一論理端末名称は、重複して指定できません。

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

*

すべての論理端末を対象とします。

先行文字列*

先行文字列で始まるすべての論理端末を対象とします。

〈複数指定の例〉

```
-l "論理端末名称1△論理端末名称2△論理端末名称3"
```

〈一括指定の例〉 len1, len2, len3 を一括して指定する場合

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

出力メッセージ

メッセージID	内容	出力先
KFCA10350-I	mcftendct コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	メッセージログファイル、または標準エラー出力
KFCA10355-W	引数の指定に誤りがあります。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	メッセージログファイル、または標準エラー出力
KFCA10359-W	mcftendct コマンド入力元への応答を失敗しました。	メッセージログファイル、または標準エラー出力
KFCA10371-I	mcftendct コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcftendct コマンドが異常終了しました。	標準エラー出力

メッセージID	内容	出力先
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10391-E	mcftendct コマンドはサポートされていません。	標準エラー出力
KFCA10521-I	ヘルプメッセージ	標準出力
KFCA13600-E	継続問い合わせ応答終了処理で障害が発生しました。	標準エラー出力
KFCA13605-E	UAP が継続問い合わせ応答処理中のため、運用コマンドは受け付けられません。	標準エラー出力
KFCA13606-E	論理端末は継続問い合わせ中ではありません。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

注意事項

mcftendct コマンドが正常に受け付けられたかどうかは、コマンドのリターン値で判断しないでください。コマンドが出力したメッセージの内容で判断してください。

mcftlscn (コネクションの状態表示)

形式

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

機能

コネクションの状態を標準出力に出力します。

オプション

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

処理対象のコネクションを制御する MCF 通信サービスの MCF 通信プロセス識別子を指定します。MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信サービスに対して、mcftlscn コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

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

●-c コネクション ID ~ 〈1~8 文字の識別子〉

状態を表示するコネクションのコネクション ID を指定します。

コネクション ID は、1 回につき 8 個まで指定できます。多数入力する運用を行う場合は、次に示す複数指定または一括指定を使用して、一つの運用コマンドで行う並列処理数を増やし、運用コマンド入力数を減らすように運用設計を行ってください。

複数指定する場合は、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。同一コネクション ID は、重複して指定できません。

また、コネクション ID は、*を使って一括指定ができます。一括指定は一つだけ指定できます。一括指定と一括指定以外のコネクション ID を混在して指定できません。一括指定の場合も引用符 (") で囲んで指定します。

*

すべてのコネクションの状態を表示します。

先行文字列*

先行文字列で始まるすべてのコネクションの状態を表示します。

〈複数指定の例〉 cnn1, cnn2, cnn3 を指定する場合

```
-c "cnn1 Δcnn2 Δcnn3"
```

〈一括指定の例〉 cnn で始まるすべてのコネクションを指定する場合

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

●-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。また現在接続中または接続リトライ中の表示印刷サービス名も表示します。

このオプションの指定を省略すると、コネクションの状態だけを表示します。

出力形式

```
mmm ccccccc ppp sssss dddd .....-dオプションを指定しないと、  
lllllllll ttt uuu          この行だけ出力されます。  
xxxx...x
```

- mmm : MCF 識別子
- ccccccc : コネクション ID
- ppp : プロトコル種別
XP...TP1/NET/XMAP3
- sssss : コネクションの状態
ACT...確立状態
ACT/B...確立処理中状態
DCT...解放状態
DCT/B...解放処理中状態
- dddd : 詳細ステータス (保守情報)
WCL...コネクション解放準備完了待ち状態
WCO...継続問い合わせ仕掛け完了待ち状態
上記以外のステータスが出力されることがありますが、内部情報のため無視してください。
- lllllll : 論理端末名称
- ttt : 論理端末タイプ
ANY...any
- uuu : 論理端末の端末状態
ACT...閉塞解除状態
DCT...閉塞状態
- xxxx...x : 接続中または接続リトライ中の表示印刷サービス名

出力メッセージ

メッセージID	内容	出力先
KFCA10350-I	mcftlscn コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	メッセージログファイル, または 標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	メッセージログファイル, または 標準エラー出力
KFCA10359-W	mcftlscn コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10360-I	状態表示を開始します。	標準出力
KFCA10361-I	標準情報を示します。	標準出力
KFCA10362-I	詳細情報を示します。	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10373-E	mcftlscn コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定したコネクションは登録されていません。	標準エラー出力
KFCA10391-E	mcftlscn コマンドはサポートされていません。	標準エラー出力
KFCA10502-I	ヘルプメッセージ	標準出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力
KFCA16429-I	付加情報を示します。	標準出力

mcftlsle (論理端末の状態表示)

形式

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

機能

論理端末のキュー状態を標準出力に出力します。

論理端末の端末状態を表示する場合は、mcftlscn コマンドを-d オプション付きで使用します。

オプション

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

処理対象の論理端末を制御する MCF 通信サービスの MCF 通信プロセス識別子を指定します。アプリケーション起動サービスのアプリケーション起動プロセス識別子は指定できません。MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信サービスに対して、mcftlsle コマンドを実行します。したがって、MCF 通信サービスを検索するオーバヘッドが、運用コマンドの処理に加わります。

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

●-c コネクション ID ~ 〈1~8 文字の識別子〉

状態を表示したい論理端末に対応するコネクションのコネクション ID を指定します。

コネクション ID は複数指定できません。また、一括指定もできません。

●-l 論理端末名称 ~ 〈1~8 文字の識別子〉

状態を表示する論理端末の名称を指定します。

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

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

複数の論理端末名称を指定する場合は、引用符 (") で囲んで、論理端末名称と論理端末名称との間を空白で区切ります。同一論理端末名称は、重複して指定できません。

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

*

すべての論理端末の状態を表示します。

先行文字列*

先行文字列で始まるすべての論理端末の状態を表示します。

〈複数指定の例〉 len1, len2, len3 を指定する場合

```
-l "len1△len2△len3"
```

〈一括指定の例〉 len で始まるすべての論理端末を指定する場合

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

●-q

指定した論理端末に対応する出力キューの保留状態を表示します。

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

出力形式

```
mmm llllllll sss [tttt]
aaaaa bbbbbbbb
SYNC xxxxxxxxxxxx yyyyyyyyyy zzzzzzzzzz
  IO   :       :
  PRIO :       :
  NORM :       :
  iii  ooo
```

- mmm : MCF 識別子
- llllllll : 論理端末名称
- sss : 論理端末のキュー状態
ACT…閉塞解除状態
DCT…閉塞状態
- tttt : 論理端末のテストモード状態 (TP1/Message Control/Tester 使用時だけ表示)
TEST…テストモード
空白…非テストモード
- aaaaa : 代行種別
ALT_F…代行元
ALT_T…代行先
- bbbbbbbb : 論理端末名称

状態を表示している論理端末が代行元論理端末の場合、対応する代行先論理端末名称を表示します。この場合 aaaaa は ALT_F (代行先) を表示します。

状態を表示している論理端末が代行先論理端末の場合、対応する代行元論理端末名称を表示します。このとき aaaaa は ALT_T (代行元) を表示します。

- SYNC：同期型メッセージ

未処理送信メッセージを破棄した場合 (ERREVTa を起動する場合)、メッセージの破棄が完了するまでいったん同期型メッセージの出力キューに格納するため、未送信メッセージ数が一時的に増加します。

- IO：非同期型問い合わせ応答メッセージ

次の場合、未送信メッセージ数が一時的に増加します。

- 端末から非応答型 MHP にメッセージを送信したときの、論理端末画面のキーボード状態をロック状態にするように論理端末定義 (mcftalcle -K keyboard=aplock) で指定している場合
- ans 型または cont 型のアプリケーションを起動するメッセージを、運用コマンド (mcftdlqsg) で入力キューから削除した場合
- ans 型または cont 型のアプリケーションの動作 (異常終了など) によって、本来ならば起動される ERREVT3 が何らかの要因で起動できなかった場合
- PRIO：非同期型一方送信メッセージ (優先)
- NORM：非同期型一方送信メッセージ (一般)
- xxxxxxxxxx：未送信メッセージ数 (unsigned short の上限値まで表示可能)
- yyyyyyyyyy：未送信メッセージの先頭の出力通番 (int の上限値まで表示可能)
- zzzzzzzzzz：未送信メッセージの最後の出力通番 (int の上限値まで表示可能)
- iii：出力キューの入力の保留状態 (-q オプション指定時だけ表示)
NOH…保留解除
HLD…保留
- ooo：出力キューの出力の保留状態 (-q オプション指定時だけ表示)
NOH…保留解除
HLD…保留

出力メッセージ

メッセージ ID	内容	出力先
KFCA10350-I	mcftlsle コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	メッセージログファイル, または標準エラー出力

メッセージID	内容	出力先
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	メッセージログファイル, または 標準エラー出力
KFCA10359-W	mcftlsle コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10360-I	状態表示を開始します。	標準出力
KFCA10364-I	表示情報	標準出力
KFCA10365-I	表示情報	標準出力
KFCA10369-I	状態表示を終了します。	標準出力
KFCA10373-E	mcftlsle コマンドが異常終了しました。	標準エラー出力
KFCA10378-I	上記の出力形式を参照してください。	標準出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10381-E	指定した接続は登録されていません。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10390-E	指定した接続 ID と論理端末名称の対応が正しくありません。	標準エラー出力
KFCA10391-E	mcftlsle コマンドはサポートされていません。	標準エラー出力
KFCA10395-E	指定した接続には未接続の論理端末名称が指定されています。	標準エラー出力
KFCA10399-I	上記の出力形式を参照してください。	標準出力
KFCA10505-I	ヘルプメッセージ	標準出力
KFCF16402-E	RPC 障害が発生しました。	標準エラー出力

mcftstalt (代行送信の開始)

形式

```
mcftstalt [-s MCF通信プロセス識別子] -f 代行元論理端末名称  
          -t 代行先論理端末名称
```

機能

送信先のプリンタに障害が発生していて送信ができないメッセージ（一方送信メッセージ）を、ほかのプリンタに送信します。メッセージグループ送信機能、またはページ制御機能を使用している場合、TP1/NET/XMAP3 では代行送信によるプリンタ出力の結果を保証しません。また、代行先論理端末のキュー状態が閉塞中の場合は代行送信されません。論理端末のキュー状態を代行先、代行元ともに閉塞解除状態にしてください。

オプション

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

処理対象の論理端末を制御する MCF 通信サービスの MCF 通信プロセス識別子を指定します。代行元と代行先の端末が同じプロセスに対応していなければなりません。MCF 通信プロセス識別子は複数指定できません。

このオプションの指定を省略すると、すべての MCF 通信サービスに対して、mcftstalt コマンドを実行します。したがって、MCF 通信サービスを検索するオーバーヘッドが、運用コマンドの処理に加わります。

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

●-f 代行元論理端末名称 ~ 〈1~8 文字の識別子〉

代行送信元の論理端末の名称を指定します。

指定する論理端末は、端末状態が閉塞中、またはプリンタが障害中でなければなりません。

代行元論理端末名称は複数指定できません。また、一括指定もできません。

●-t 代行先論理端末名称 ~ 〈1~8 文字の識別子〉

代行送信先の論理端末の名称を指定します。すでに代行先として指定されている論理端末名称は指定できません。

指定する論理端末の端末状態が閉塞している場合は、指定できません。

指定する論理端末プリンタが障害中の場合は、指定できません。

代行先論理端末名称は複数指定できません。また、一括指定もできません。

出力メッセージ

メッセージID	内容	出力先
KFCA10116-I	メッセージの代行送信を開始します。	メッセージログファイル
KFCA10350-I	mcfstalt コマンドが入力されました。	標準出力
KFCA10351-E	MCF 開始処理中です。	標準エラー出力
KFCA10352-E	MCF 終了処理中です。	標準エラー出力
KFCA10353-W	入力形式が誤っています。	標準エラー出力
KFCA10354-E	メモリ不足です。	メッセージログファイル, または標準エラー出力
KFCA10355-W	引数の指定が誤っています。	標準エラー出力
KFCA10356-E	プロセス間でタイムアウトが発生しました。	標準エラー出力
KFCA10357-E	MCF 内でタイムアウトが発生しました。	標準エラー出力
KFCA10358-E	内部関数のエラーが発生しました。	メッセージログファイル, または標準エラー出力
KFCA10359-W	mcfstalt コマンド入力元への応答を失敗しました。	メッセージログファイル
KFCA10371-I	mcfstalt コマンドを正常に受け付けました。	標準出力
KFCA10373-E	mcfstalt コマンドが異常終了しました。	標準エラー出力
KFCA10380-E	相手プロセスの検索に失敗しました。	標準エラー出力
KFCA10382-E	指定した論理端末は登録されていません。	標準エラー出力
KFCA10391-E	mcfstalt コマンドはサポートされていません。	標準エラー出力
KFCA10541-I	ヘルプメッセージ	標準出力
KFCA13687-W	表示サーバ下の論理端末の代行送信は、サポートしていません。	標準エラー出力
KFCA13688-E	代行送信開始処理で障害が発生しました。	メッセージログファイル
KFCA13698-E	代行送信開始処理中に障害が発生しました。	標準エラー出力
KFCA16402-E	RPC 障害が発生しました。	標準エラー出力

注意事項

mcfstalt コマンドが正常に受け付けられたかどうかは、コマンドのリターン値で判断しないでください。コマンドが出力したメッセージの内容で判断してください。

9

オンラインコマンド

この章では、TP1/NET/XMAP3 のオンラインコマンドについて説明します。

TP1/NET/XMAP3 のオンラインコマンド

TP1/NET/XMAP3 では、オンライン中に OpenTP1 システム側と端末側からコマンドを入力できます。端末側から入力するコマンドをオンラインコマンドと呼びます。

オンラインコマンドは、受信メッセージの先頭セグメントだけ有効です。

オンラインコマンドの一覧

TP1/NET/XMAP3 のオンラインコマンドの一覧を次の表に示します。

表 9-1 TP1/NET/XMAP3 のオンラインコマンドの一覧

コマンド	機能
ACT ALTSEND	代行送信を開始します。
DEACT ALTSEND	代行送信を終了します。
init	指定されたマップで初期画面を表示します。

オンラインコマンド使用時の注意事項

- init コマンド以外のオンラインコマンドを使用する場合、オンラインコマンドサービスを起動する必要があります。オンラインコマンドサービスを起動するには、MCF マネージャ共通定義 (mcfmcomn -o) の cmdsvname オペランドで、_mcs で始まる MCF オンラインコマンドサービス名を指定します。
- 適用 OS が Windows の場合は、init コマンドだけ使用できます。

オンラインコマンドの省略形

オンラインコマンドには、コマンド名とオプションに省略形があります。オンラインコマンドの省略形式を次の表に示します。

表 9-2 オンラインコマンドの省略形式

本来形式	省略形式
ACT ALTSEND FROM=論理端末名称 1, TO=論理端末名称 2	A ALTSEND FROM=論理端末名称 1, TO=論理端末名称 2
DEACT ALTSEND VTERM=論理端末名称	DEA ALTSEND VT=論理端末名称
init MAP=マップ名	なし

オンラインコマンドの入力形式

次のどちらかの形式で指定します。

C△オンラインコマンド
c△オンラインコマンド

入力例

```
c△init△MAP=MAP001
```

ACT ALTSEND (代行送信の開始)

形式

```
ACT ALTSEND FROM=論理端末名称1, TO=論理端末名称2
```

機能

送信先のプリンタに障害が発生していて送信ができないメッセージ（一方送信メッセージ）をほかのプリンタに送信します。メッセージグループ送信機能，またはページ制御機能を使用している場合，TP1/NET/XMAP3 では代行送信によるプリンタ出力の結果を保証しません。

オプション

●FROM=論理端末名称 1 ～ 〈1～8 文字の識別子〉

代行送信元の論理端末名称を指定します。

●TO=論理端末名称 2 ～ 〈1～8 文字の識別子〉

代行送信先の論理端末名称を指定します。

DEACT ALTSEND (代行送信の終了)

形式

```
DEACT ALTSEND VTERM=論理端末名称
```

機能

代行送信を終了します。

オプション

●VTERM=論理端末名称 ~ 〈1~8文字の識別子〉

代行送信を終了する代行送信元の論理端末名称 (ACT ALTSEND コマンドの FROM オプションで指定したもの) を指定します。

init (初期画面の表示)

形式

```
init [MAP=マップ名]
```

機能

初期画面を表示します。

オプション

●MAP=マップ名 ~ 〈1~6文字の記号名称〉

表示する初期画面のマップ名を指定します。

このオプションを省略すると、論理端末定義 (mcftalcle) で指定したイニシャライズマップを表示します。

10

組み込み方法

この章では、TP1/NET/XMAP3 を OpenTP1 システムに組み込む方法について説明します。

10.1 TP1/NET/XMAP3 の組み込みの流れ

TP1/NET/XMAP3 を OpenTP1 システムに組み込む場合の作業の流れを説明します。

10.1.1 MCF メイン関数の作成

TP1/NET/XMAP3 を起動するためには、MCF メイン関数をコーディングし、コンパイル、およびリンケージしておく必要があります。詳細は、「[10.2 MCF メイン関数の作成](#)」を参照してください。

10.1.2 MCF サービス名の登録

TP1/NET/XMAP3 を実行するために、MCF サービス名をシステムサービス構成定義で定義しておく必要があります。

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

詳細は、マニュアル「OpenTP1 システム定義」を参照してください。

10.1.3 システムサービス情報定義ファイルの作成

システムサービス情報定義ファイルをテキストエディタで作成します。作成するファイルのパス名は、「\$DCDIR/lib/sysconf/システムサービス情報定義ファイル名」としてください。

ファイルの定義形式については、「[7. システム定義](#)」の「[システムサービス情報定義](#)」を参照してください。

10.1.4 定義オブジェクトファイルの作成

OpenTP1 のネットワークコミュニケーション定義の各ソースファイルから定義オブジェクトファイルを生成します。詳細は、「[10.3 定義オブジェクトファイルの生成](#)」を参照してください。

10.2 MCF メイン関数の作成

TP1/NET/XMAP3 は、OpenTP1 プロセスサービスによって起動されます。

TP1/NET/XMAP3 を起動するためには、ユーザが MCF メイン関数をコーディングし、コンパイル、およびリンケージを行って TP1/NET/XMAP3 の実行形式プログラムを作成する必要があります。実行形式プログラムのリンケージには、mcfplxp コマンドを使用します。

MCF メイン関数では、スタート関数 (dc_mcf_svstart) を呼び出します。UOC を使用する場合は、MCF メイン関数で UOC の関数アドレスを指定してください。UOC は、MCF メイン関数と同じ言語 (ANSI C, C++, または K&R 版 C) で作成してください。MCF メイン関数のコーディング概要、および MCF メイン関数のディレクトリへの組み込み方法を、以降の図に示します。

なお、これらのコーディング例を次のファイルで提供しています。

適用 OS が UNIX の場合

- /BeTRAN/examples/mcf/XMAP3/cmlib/ansi/com.c
- /BeTRAN/examples/mcf/XMAP3/cmlib/c/com.c

適用 OS が Windows の場合

- %DCDIR%\examples\mcf\XMAP3\cmlib\c\com.c

図 10-1 MCF メイン関数のコーディング概要 (ANSI C, C++の場合)

```
#include <dcmxp.h> /*TP1/NET/XMAP3用ヘッダファイル */ 1.
extern DCLONG msgrcv01(dcmcf_uoc_min_n *); /*入力メッセージ編集UOC関数extern宣言 */ 2.
extern DCLONG msgsend01(dcmcf_uoc_mout_n *); /*出力メッセージ編集UOC関数extern宣言 */ 2.
extern dcmcf_uoc_t dcmcf_uoctbl; /*UOCテーブルextern宣言 */ 3.
int main()
{
    dcmcf_uoctbl.msgrcv = (dcmcf_uocfunc)msgrcv01;
    dcmcf_uoctbl.msgsend = (dcmcf_uocfunc)msgsend01; /*入力メッセージ編集UOCアドレス設定 */ 4.
    /*出力メッセージ編集UOCアドレス設定 */ 4.
    dc_mcf_svstart(); /*スタート関数発行 */ 5.
    return 0;
}
```

1. TP1/NET/XMAP3 で提供するヘッダファイルを取り込みます。
2. 使用する UOC 関数を extern 宣言します。UOC のリターン値は DCLONG 型にしてください。
これらの UOC を使用する場合だけ、コーディングしてください。
3. UOC テーブルを extern 宣言します。UOC を使用する場合、必ずこのとおりにコーディングしてください。
4. 各 UOC 関数のアドレスを、次に示すシステム提供変数に設定します。

```
dcmcf_uoctbl.msgrcv /*入力メッセージ編集UOCアドレス*/
dcmcf_uoctbl.msgsend /*出力メッセージ編集UOCアドレス*/
```

これらの UOC を使用する場合だけ、コーディングしてください。

5. スタート関数を呼び出します。MCF メイン関数には必ずコーディングしてください。

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

図 10-2 MCF メイン関数のコーディング概要 (K&R 版 C の場合)

```
#include <dcmxp.h> /*TP1/NET/XMAP3用ヘッダファイル */ 1.
extern DCLONG msgrcv01(); /*入力メッセージ編集UOC関数extern宣言 */ 2.
extern DCLONG msgsend01(); /*出力メッセージ編集UOC関数extern宣言 */ 2.
extern dcmcf_uoc_t dcmcf_uoctbl; /*UOCテーブルextern宣言 */ 3.
main()
{
    dcmcf_uoctbl.msgrcv = msgrcv01; /*入力メッセージ編集UOCアドレス設定 */ 4.
    dcmcf_uoctbl.msgsend = msgsend01; /*出力メッセージ編集UOCアドレス設定 */ 4.
    dc_mcf_svstart(); /*スタート関数発行 */ 5.
}
```

1. TP1/NET/XMAP3 で提供するヘッダファイルを取り込みます。

2. 使用する UOC 関数を extern 宣言します。UOC のリターン値は DCLONG 型にしてください。
これらの UOC を使用する場合だけ、コーディングしてください。

3. UOC テーブルを extern 宣言します。UOC を使用する場合、必ずこのとおりにコーディングしてください。

4. 各 UOC 関数のアドレスを、次に示すシステム提供変数に設定します。

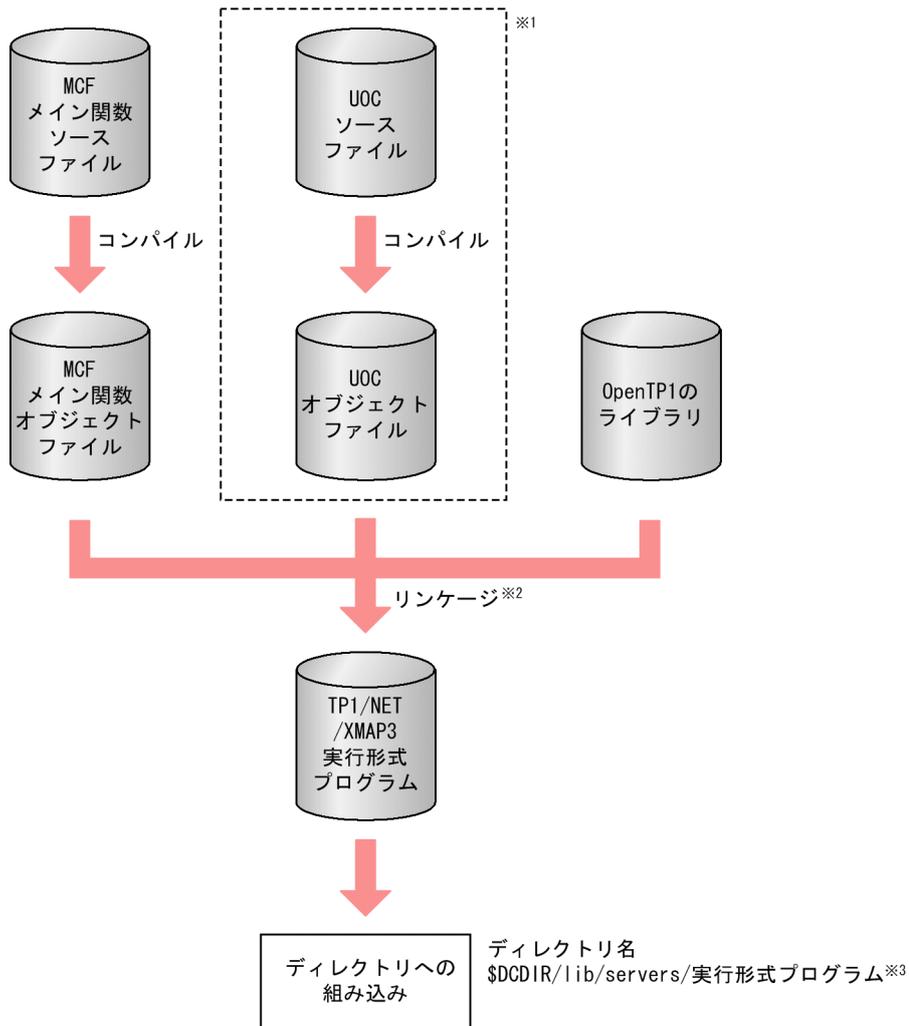
```
dcmcf_uoctbl.msgrcv /*入力メッセージ編集UOCアドレス*/
dcmcf_uoctbl.msgsend /*出力メッセージ編集UOCアドレス*/
```

これらの UOC を使用する場合だけ、コーディングしてください。

5. スタート関数を呼び出します。MCF メイン関数には必ずコーディングしてください。

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

図 10-3 MCF メイン関数のディレクトリへの組み込み方法



注※1

ユーザ作成の UOC を使用しない場合は、必要ありません。

注※2

mcfplxp コマンドでリンケージします。

mcfplxp コマンドの詳細については、TP1/NET/XMAP3 の「リリースノート」を参照してください。

注※3

TP1/NET/XMAP3 の実行形式プログラム名は、先頭が mcfu で始まる 8 文字以内の名称です。

10.3 定義オブジェクトファイルの生成

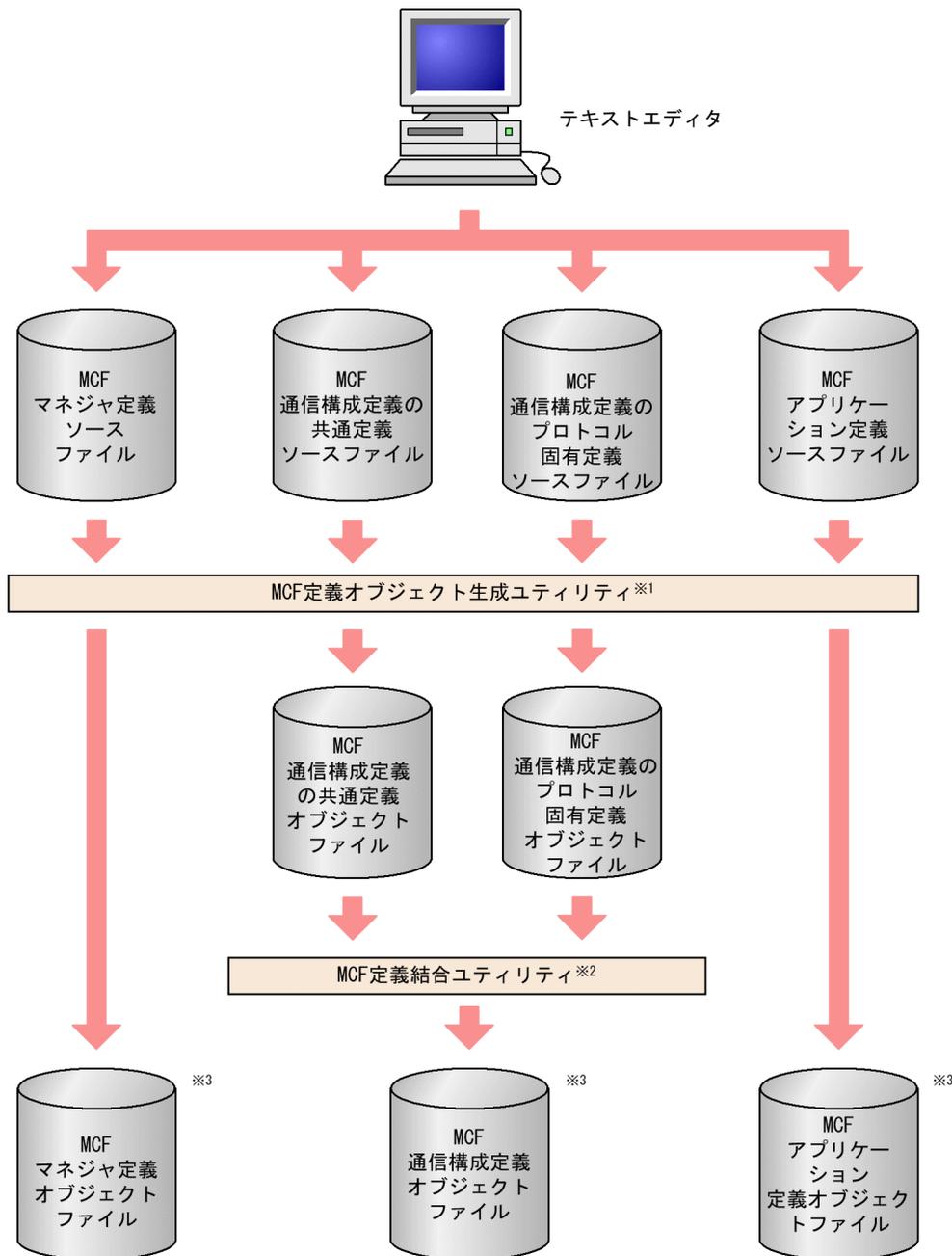
定義オブジェクトファイルを次の手順で生成します。

ただし、開始から再開始の間に定義オブジェクトファイルを変更してはいけません。変更した場合、再開始の動作は保証できません。

1. OS のテキストエディタを使用して、MCF の定義ファイルから、次に示す定義ソースファイルを作成します。
 - MCF マネージャ定義ソースファイル
 - MCF 通信構成定義の共通定義ソースファイル
 - MCF 通信構成定義の TP1/NET/XMAP3 のプロトコル固有定義ソースファイル
 - MCF アプリケーション定義ソースファイル
2. MCF 定義オブジェクト生成ユーティリティを使用して、定義ソースファイルから、次に示すオブジェクトファイルを作成します。
 - MCF マネージャ定義オブジェクトファイル
 - MCF 通信構成定義の共通定義オブジェクトファイル
 - MCF 通信構成定義の TP1/NET/XMAP3 のプロトコル固有定義オブジェクトファイル
 - MCF アプリケーション定義オブジェクトファイル
3. MCF 定義結合ユーティリティを使用して、MCF 通信構成定義の共通定義とプロトコル固有定義のオブジェクトファイルを結合します。

定義オブジェクトファイルの作成方法の概要を次の図に示します。

図 10-4 定義オブジェクトファイルの作成方法の概要



注※1

次に示すコマンドで生成します。

```

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

mcf××××は、ソースファイルごとに異なります。

- mcfmgr : MCF マネージャ定義ソースファイル
- mcfcomn : MCF 通信構成定義の共通定義ソースファイル
- mcfxp : MCF 通信構成定義のプロトコル (TP1/NET/XMAP3) 固有定義ソースファイル

- mcfapli : MCF アプリケーション定義ソースファイル

MCF 定義オブジェクト生成ユーティリティの mcfxp コマンドについては「7. システム定義」の「MCF 定義オブジェクトの生成」を、その他のコマンドについてはマニュアル「OpenTP1 システム定義」を参照してください。

注※2

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

```
mcf link  $\Delta$ -i  $\Delta$ MCF通信構成定義の共通定義オブジェクトファイル名  $\Delta$ 
          TP1/NET/XMAP3固有定義オブジェクトファイル名
           $\Delta$ -o  $\Delta$ MCF通信構成定義のオブジェクトファイル名
```

注※3

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

11

障害対策

この章では、TP1/NET/XMAP3 の障害対策について説明します。

11.1 障害の種類と対応処理

TP1/NET/XMAP3 は、障害の種類によってそれぞれに対応する処理をします。障害内容によっては、TP1/NET/XMAP3 はオペレータインジケータにメッセージを表示します。オペレータインジケータに表示されるメッセージについては、「付録I オペレータインジケータ表示メッセージ一覧」を参照してください。

TP1/NET/XMAP3 の障害発生時の処理について、次に示す障害の種類ごとに説明します。

11.1.1 コネクション障害

表 11-1 コネクション障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
XMAP3 Server ・通信路切断 ・サーバ未登録 ・サーバ利用不可 など	<ol style="list-style-type: none">1. 障害発生時の処理に対応したメッセージログを出力します。2. コネクション内の全論理端末を閉塞します。3. コネクション定義 (mcftalccn) で、再試行を指定すればコネクション確立再試行をします。再試行を指定しなければコネクションを閉塞します。	コネクションが閉塞した場合、運用コマンド (mcftactcn, mcftactle) の入力、または API (dc_mcf_tactcn 関数, dc_mcf_tactle 関数) の発行で、コネクションを確立し、論理端末の閉塞を解除します。
・作業領域確保失敗 など	<ol style="list-style-type: none">1. 障害発生時の処理に対応したメッセージログを出力します。2. コネクション内の全論理端末を閉塞します。3. コネクションを閉塞します。	運用コマンド (mcftactcn, mcftactle) の入力、または API (dc_mcf_tactcn 関数, dc_mcf_tactle 関数) の発行で、コネクションを確立し、論理端末の閉塞を解除します。

11.1.2 論理端末障害

表 11-2 論理端末障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
XMAP3 Server ・作業領域確保失敗 ・論理端末画面生成数超過 など	<ol style="list-style-type: none">1. 障害発生時の処理に対応したメッセージログを出力します。2. VERREVT を起動します。3. 論理端末画面を破棄します (画面の場合)。4. 論理端末を閉塞します。	運用コマンド (mcftactle) の入力、または API (dc_mcf_tactle 関数) の発行で、論理端末の閉塞を解除します。

11.1.3 メッセージ障害

表 11-3 メッセージ障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
XMAP3 Server ・メッセージ不正	<ol style="list-style-type: none"> 1. 障害発生時の処理に対応したメッセージログを出力します。 2. 送信メッセージを破棄します。応答メッセージの場合は、オペレータインジケータに表示します。 3. 処理を続行します。 	ありません。

11.1.4 マッピング障害

表 11-4 マッピング障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
<ul style="list-style-type: none"> ・マッピングサービス障害 ・XMAP3 マッピング機能の障害 	<ol style="list-style-type: none"> 1. 障害発生時の処理に対応したメッセージログを出力します。 2. メッセージの種別によって次の処理をします。 	<ul style="list-style-type: none"> ・マッピング障害の原因を取り除きます。 ・保留された送信メッセージが不要な場合は、運用コマンド (mcftdlqle) の入力、または API (dc_mcf_tdlqle 関数) の発行で、出力メッセージを破棄します。 ・論理端末が閉塞した場合、運用コマンド (mcftactle) の入力、または API (dc_mcf_tactle 関数) の発行で論理端末の閉塞を解除します。
	応答メッセージ <ul style="list-style-type: none"> ・送信メッセージを破棄します。 ・論理端末を閉塞します。 	
	一方送信メッセージ <ul style="list-style-type: none"> ・送信メッセージを保留します。 ・論理端末を閉塞します。 	
	問い合わせメッセージ <ul style="list-style-type: none"> ・受信メッセージを破棄します。 ・論理端末を閉塞します。 	

11.1.5 プリンタ障害

表 11-5 プリンタ障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
(排他時) XMAP3 Server <ul style="list-style-type: none"> ・準備完 OFF ・印刷用紙なし ・電源 OFF ・ハード障害など 	<ol style="list-style-type: none"> 1. 障害発生時の処理に対応したメッセージログを出力します。 2. 送信メッセージを保留します。 3. 論理端末定義 (mcftalcle) で、再試行を指定すればプリンタ排他再試行をします。再試行を指定しなければ論理端末を閉塞します。 	<ul style="list-style-type: none"> ・プリンタ障害の原因を取り除きます。 ・保留された送信メッセージが不要な場合は、運用コマンド (mcftdlqle) の入力、または API (dc_mcf_tdlqle 関数) の発行で、出力メッセージを破棄します。 ・論理端末が閉塞した場合、運用コマンド (mcftactle) の入力、または API

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
(排他時) XMAP3 Server ・準備完 OFF ・印刷用紙なし ・電源 OFF ・ハード障害 など	<ol style="list-style-type: none"> 1. 障害発生時の処理に対応したメッセージログを出力します。 2. 送信メッセージを保留します。 3. 論理端末定義 (mcftalcle) で、再試行を指定すればプリンタ排他再試行をします。再試行を指定しなければ論理端末を閉塞します。 	(dc_mcf_tactle 関数) の発行で論理端末の閉塞を解除します。
(送信完待ち時) XMAP3 Server ・準備完 OFF ・印刷用紙なし ・電源 OFF など	<ol style="list-style-type: none"> 1. 障害発生時の処理に対応したメッセージログを出力します。 2. 障害回復を待ちます。 	プリンタ障害の原因を取り除きます。
(送信完待ち時) ・ハード障害 など	<ol style="list-style-type: none"> 1. 障害発生時の処理に対応したメッセージログを出力します。 2. 送信メッセージを保留します。 3. 論理端末を閉塞します。 	<ul style="list-style-type: none"> ・プリンタ障害の原因を取り除きます。 ・保留された送信メッセージが不要な場合は、運用コマンド (mcftdlqle) の入力、または API (dc_mcf_tdlqle 関数) の発行で、出力メッセージを破棄します。 ・運用コマンド (mcftactle) の入力、または API (dc_mcf_tactle 関数) の発行で論理端末の閉塞を解除します。

11.1.6 入力メッセージ編集 UOC の障害

表 11-6 入力メッセージ編集 UOC の障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
UOC エラーリターン	<ol style="list-style-type: none"> 1. UOC エラーリターンを通知するメッセージログ (KFCA10611-E)、およびアプリケーション名取得失敗を通知するメッセージログ (KFCA10610-E) を出力します。 2. 障害発生時の処理に対応したメッセージログを出力します。 3. 受信メッセージを破棄します。 4. オペレータインジケータを表示します。 5. 処理を続行します。 	ありません。

11.1.7 入力キュー、スケジュールサービス、ネームサービス、RPC の障害

表 11-7 入力キュー、スケジュールサービス、ネームサービス、RPC の障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
<ul style="list-style-type: none"> ・アプリケーション閉塞 ・サービス閉塞 ・アプリケーション名不正 ・入力キュー書き込み障害 (メモリ含む) など	<ol style="list-style-type: none"> 1. 入力キュー障害を通知するメッセージログ (KFCA10604-E) を出力します。 2. 障害発生時の処理に対応したメッセージログを出力します。 3. メッセージの種類によって次の処理をします。 	ありません。
	画面入力メッセージ <ul style="list-style-type: none"> ・ ERREVT1 または ERREVT2 を起動します。 	
	ERREVT1, ERREVT2, ERREVT3, ERREVT4 <ul style="list-style-type: none"> ・ メッセージを破棄します。 	
	VERREVT, VOPNEVT, VCLSEVT <ul style="list-style-type: none"> ・ イベントを破棄します。 	

11.1.8 出力キュー障害

表 11-8 出力メッセージ編集 UOC を含む出力キュー障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
<ul style="list-style-type: none"> ・出力キュー読み込み障害 (メモリ含む) ・UOC エラーリターン 	<ol style="list-style-type: none"> 1. UOC エラーリターンを通知するメッセージログ (KFCA10611-E), および出力キュー障害を通知するメッセージログ (KFCA10605-E) を出力します。 2. 障害発生時の処理に対応したメッセージログを出力します。 3. メッセージの種類によって次の処理をします。 	<ul style="list-style-type: none"> ・ 保留された送信メッセージが不要な場合は、運用コマンド (mcftdlqle) の入力、または API (dc_mcf_tdlqle 関数) の発行で、出力メッセージを破棄します。 ・ 論理端末が閉塞した場合、運用コマンド (mcftactle) の入力、または API (dc_mcf_tactle 関数) の発行で論理端末の閉塞を解除します。
	応答メッセージ <ul style="list-style-type: none"> ・ 送信メッセージを破棄します。 ・ オペレータインジケータを表示します。 ・ 処理を続行します。 	
	一方送信メッセージ <ul style="list-style-type: none"> ・ 送信メッセージを保留します。 ・ 論理端末を閉塞します。 	
<ul style="list-style-type: none"> ・メッセージ送信済み障害 ・メッセージリセット障害 	<ol style="list-style-type: none"> 1. 送信完了処理障害を通知するメッセージログ (KFCA10617-E) を出力します。 2. 障害発生時の処理に対応したメッセージログを出力します。 3. メッセージの種類によって次の処理をします。 	<ul style="list-style-type: none"> ・ 保留された送信メッセージが不要な場合は、運用コマンド (mcftdlqle) の入力、または API (dc_mcf_tdlqle 関数) の発行で、出力メッセージを破棄します。 ・ 論理端末が閉塞した場合、運用コマンド (mcftactle) の入力、または API

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
<ul style="list-style-type: none"> ・メッセージ送信済み障害 ・メッセージリセット障害 	応答メッセージ <ul style="list-style-type: none"> ・送信メッセージを破棄します。 ・オペレータインジケータを表示します。 ・処理を続行します。 	(dc_mcf_tactle 関数) の発行で論理端末の閉塞を解除します。
	一方送信メッセージ <ul style="list-style-type: none"> ・送信メッセージを保留します。 ・論理端末を閉塞します。 	

11.1.9 ジャーナル障害

表 11-9 ジャーナル障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
<ul style="list-style-type: none"> ・ジャーナル取得障害 (aj, ij) 	<ol style="list-style-type: none"> 1. ジャーナル障害を通知するメッセージログ (KFCA10609-E) を出力します。 2. 処理を続行します。 	ありません。

11.1.10 バッファ障害

表 11-10 送信バッファ, 受信バッファ, または編集バッファ障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
<ul style="list-style-type: none"> ・サイズ不足 ・テーブル数不足 	<ol style="list-style-type: none"> 1. 障害発生時の処理に対応したメッセージログを出力します。 2. 送信メッセージの場合はメッセージを保留します。 3. 論理端末を閉塞します。 	保留された送信メッセージが不要な場合は、運用コマンド (mcfhdlqle) の入力, または API (dc_mcf_tdlqle 関数) の発行で、出力メッセージを破棄します。

11.1.11 UAP 障害

表 11-11 UAP 障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
<ul style="list-style-type: none"> ・UAP 異常終了 ・rollback 発行 など	ERREVT2 または ERREVT3 を起動します。	ありません。

11.1.12 MCF の障害

表 11-12 MCF の障害と対応処理

障害内容	TP1/NET/XMAP3 の処理	ユーザの処理
・内部論理矛盾 など	1. 障害発生時の処理に対応したメッセージログを出力します。 2. メモリダンプを出力します。 3. 必要に応じ、MCF のプロセスを異常終了します（続行可能な場合は異常終了しません）。	保守情報（\$DCDIR/spool ディレクトリ以下）を退避してください。
・MCF プログラムエラー ・UOC プログラムエラー	OS によってコアダンプを出力します。	保守情報（\$DCDIR/spool ディレクトリ以下）を退避してください。

11.2 通信形態と障害の処理

11.2.1 非問い合わせ応答形態の障害

TP1/NET/XMAP3 が非問い合わせ応答形態のメッセージ処理中に、障害の発生する個所の区分を次の図に示します。また、障害発生個所ごとにする TP1/NET/XMAP3 の障害処理について、次の表に示します。

図 11-1 非問い合わせ応答形態の障害

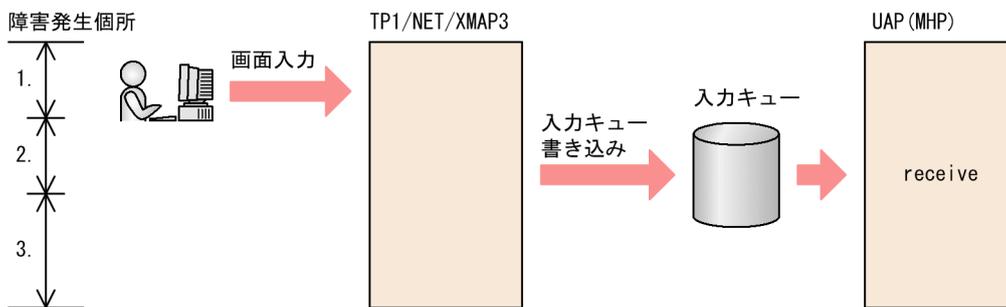


表 11-13 非問い合わせ応答形態の障害の処理

障害発生個所	内容	TP1/NET/XMAP3 の処理	送信メッセージの扱い	受信メッセージの扱い	UAP でできる処理
1.	論理端末障害 入力マッピング 障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します。 論理端末を閉塞します。	該当しません。	破棄します。	VERREVT 処理
	入力編集 UOC 障害	メッセージログを出力します。 オペレータインジケータを表示 します。 論理端末は閉塞しません。	該当しません。	破棄します。	ありません。
2.	入力キュー障害	メッセージログを出力します。 ERREVT1 または ERREVT2 を起動します。起動できない場 合はオペレータインジケータを 表示し、受信メッセージは破棄 します。 論理端末は閉塞しません。	該当しません。	ERREVT1, ま たは ERREVT2 で UAP に渡し ます。	ERREVT1 または ERREVT2 処理 (noans 型で起動) send による画面への メッセージ送信
3.	UAP 異常終了	ERREVT2 または ERREVT3 を起動します。 論理端末は閉塞しません。	該当しません。	ERREVT2, ま たは ERREVT3 で UAP に渡し ます。	ERREVT2 または ERREVT3 処理 send による画面への メッセージ送信

11.2.2 分岐送信形態の障害

TP1/NET/XMAP3 が分岐送信形態のメッセージ処理中に、障害の発生する個所の区分を次の図に示します。また、障害発生個所ごとにする TP1/NET/XMAP3 の障害処理について、次の表に示します。

図 11-2 分岐送信形態の障害

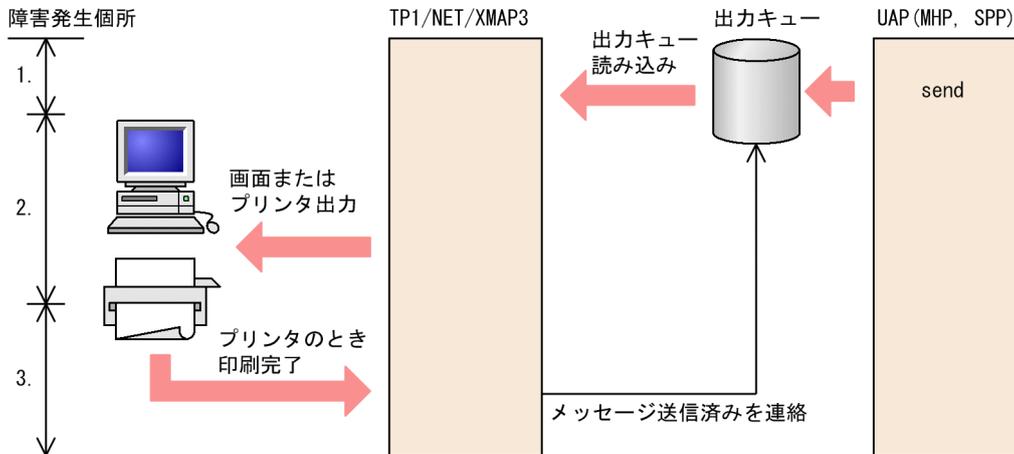


表 11-14 分岐送信形態の障害の処理

障害発生個所	内容	TP1/NET/XMAP3 の処理	送信メッセージの扱い	受信メッセージの扱い	UAP のできる処理
1.	出力キュー障害 出力編集 UOC 障害 出力マッピング障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します (画面の場合)。 論理端末を閉塞します。	保留します。	該当しません。	VERREVT 処理
2.	論理端末障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します (画面の場合)。 論理端末を閉塞します。	保留します。	該当しません。	VERREVT 処理
3.	メッセージ送信済み障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します (画面の場合)。 論理端末を閉塞します。	保留します。	該当しません。	VERREVT 処理

11.2.3 問い合わせ応答形態の障害

TP1/NET/XMAP3 が問い合わせ応答形態のメッセージ処理中に、障害の発生する個所の区分を次の図に示します。また、障害発生個所ごとにする TP1/NET/XMAP3 の障害処理について、次の表に示します。

図 11-3 問い合わせ応答形態の障害

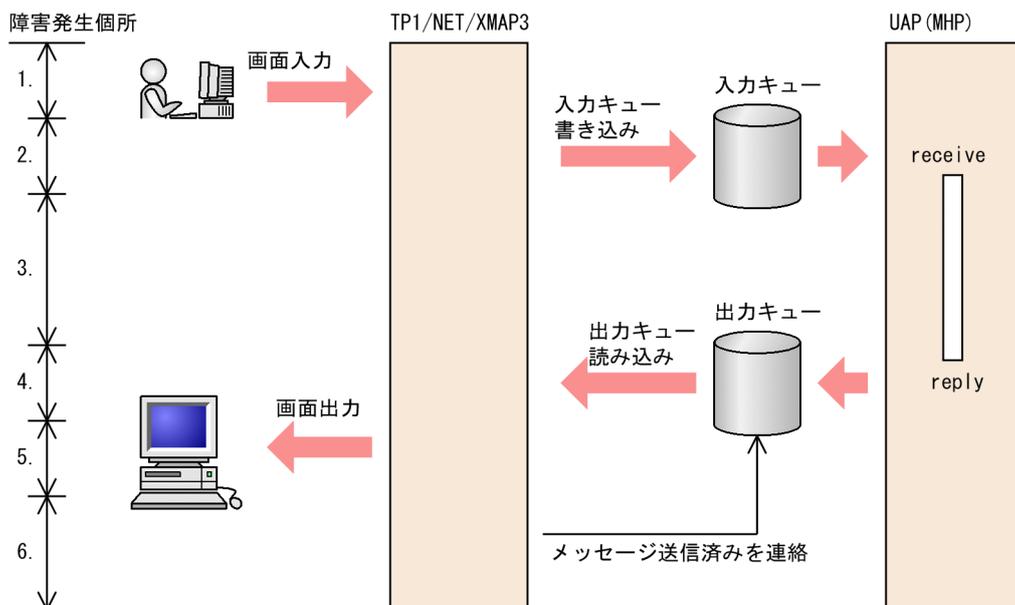


表 11-15 問い合わせ応答形態の障害の処理

障害発生個所	内容	TP1/NET/XMAP3 の処理	送信メッセージの扱い	受信メッセージの扱い	UAP のできる処理
1.	論理端末障害 入力マッピング 障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します。 論理端末を閉塞します。	該当しません。	破棄します。	VERREVT 処理
	入力編集 UOC 障害	メッセージログを出力します。 オペレータインジケータを表示 します。 論理端末は閉塞しません。	該当しません。	破棄します。	ありません。
2.	入力キュー障害	メッセージログを出力します。 ERREVT1 または ERREVT2 を起動します。起動できない場 合はオペレータインジケータを 表示し、受信メッセージは破棄 します。 論理端末は閉塞しません。	該当しません。	ERREVT1, ま たは ERREVT2 で UAP に渡し ます。	<ul style="list-style-type: none"> ERREVT1 処理 (noans 型で起動) send による画面への メッセージ送信 ERREVT2 処理 (ans 型で起動) reply による画面へ のメッセージ送信
3.	UAP 異常終了	ERREVT2 または ERREVT3 を起動します。 論理端末は閉塞しません。	異常終了前の 応答送信は破 棄します。	ERREVT2, ま たは ERREVT3 で UAP に渡し ます。	ERREVT2 または ERREVT3 処理 reply による画面への メッセージ送信
4.	出力キュー障害 出力マッピング 障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します。	破棄します。	該当しません。	VERREVT 処理

障害発生個所	内容	TP1/NET/XMAP3 の処理	送信メッセージの扱い	受信メッセージの扱い	UAP ができる処理
4.	出力キュー障害 出力マッピング 障害	論理端末を閉塞します。	破棄します。	該当しません。	VERREVT 処理
	出力編集 UOC 障害	メッセージログを出力します。 オペレータインジケータを表示 します。 論理端末は閉塞しません。	破棄します。	該当しません。	ありません。
5.	論理端末障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します。 論理端末を閉塞します。	破棄します。	該当しません。	VERREVT 処理
6.	メッセージ送信 済み障害	メッセージログを出力します。 オペレータインジケータを表示 します。 論理端末は閉塞しません。	破棄します。	該当しません。	ありません。

11.2.4 継続問い合わせ応答形態の障害

TP1/NET/XMAP3 が継続問い合わせ応答形態のメッセージ処理中に、障害の発生する個所の区分を次の図に示します。また、障害発生個所ごとにする TP1/NET/XMAP3 の障害処理について、次の表に示します。

なお、継続問い合わせ応答形態でメッセージ処理中に障害が発生した場合、一時記憶データは消去されます。

障害発生個所	内容	TP1/NET/XMAP3 の処理	送信メッセージの扱い	受信メッセージの扱い	UAP ができる処理
2.	入力キュー障害	表示し、受信メッセージは破棄します。 論理端末は閉塞しません。	該当しません。	ERREVT1、 または ERREVT2 で UAP に渡します。	(cont 型で起動) 1.reply (次アプリケーション名なし) による画面への応答メッセージ送信 2.contend による継続問い合わせ応答終了 3.reply (次アプリケーション名あり) による継続問い合わせ応答の続行 4.execap による継続問い合わせ応答の続行
3.	UAP 異常終了	ERREVT2 または ERREVT3 を起動します。 論理端末は閉塞しません。	異常終了前の応答送信は破棄します。	ERREVT2、 または ERREVT3 で UAP に渡します。	ERREVT2 または ERREVT3 処理 発生個所 2. の ERREVT2 処理と同じ
4.	出力キュー障害 出力マッピング障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します。 論理端末を閉塞します。	破棄します。	該当しません。	VERREVT 処理
	出力編集 UOC 障害	メッセージログを出力します。 オペレータインジケータを表示します。 論理端末は閉塞しません。 継続問い合わせ応答は終了しません。	破棄します。	該当しません。	ありません。
5.	論理端末障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します。 論理端末を閉塞します。	破棄します。	該当しません。	VERREVT 処理
6.	メッセージ送信済み障害	メッセージログを出力します。 オペレータインジケータを表示します。 論理端末は閉塞しません。 継続問い合わせ応答を終了します。	破棄します。	該当しません。	ありません。
7.	論理端末障害 入力マッピング障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します。	該当しません。	破棄します。	VERREVT 処理

障害発生個所	内容	TP1/NET/XMAP3 の処理	送信メッセージの扱い	受信メッセージの扱い	UAP でできる処理
7.	論理端末障害 入力マッピング障害	論理端末を閉塞します。	該当しません。	破棄します。	VERREVT 処理
	入力編集 UOC 障害	メッセージログ出力 オペレータインジケータを表示 します。 論理端末は閉塞しません。 継続問い合わせ応答は終了し ません。	該当しません。	破棄します。	ありません。
8.	入力キュー障害	メッセージログを出力します。 ERREVT1 または ERREVT2 を起動します。起動できない場 合はオペレータインジケータを 表示し、受信メッセージは破棄 します。 論理端末は閉塞しません。	該当しません。	ERREVT1, または ERREVT2 で UAP に渡しま す。	発生個所 2.と同じ
9.	UAP 異常終了	ERREVT2 または ERREVT3 を起動します。 論理端末は閉塞しません。	異常終了前の応 答送信は破棄し ます。	ERREVT2, または ERREVT3 で UAP に渡しま す。	発生個所 2.と同じ
10.	出力キュー障害 出力マッピング 障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します。 論理端末を閉塞します。	破棄します。	該当しません。	VERREVT 処理
	出力編集 UOC 障害	メッセージログを出力します。 オペレータインジケータを表示 します。 論理端末は閉塞しません。	破棄します。	該当しません。	ありません。
11.	論理端末障害	メッセージログを出力します。 VERREVT を起動します。 論理端末画面を破棄します。 論理端末を閉塞します。	該当しません。	破棄します。	VERREVT 処理
12.	メッセージ送信 済み障害	メッセージログを出力します。 オペレータインジケータを表示 します。 論理端末は閉塞しません。 継続問い合わせ応答を終了しま す。	破棄します。	該当しません。	ありません。

注※

1.の処理時に 2.の処理をしない場合、MCF が強制的に継続問い合わせ応答を終了します。

付録

付録 A バージョンアップ時の変更点

各バージョンでの変更点を次に示す分類ごとに示します。

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

付録 A.1 07-50 での変更点

TP1/NET/XMAP3 07-50 での関数、定義およびコマンドの追加・変更・削除を次の表に示します。

表 A-1 TP1/NET/XMAP3 07-50 での関数、定義およびコマンドの追加・変更・削除

種別	分類	内容
追加	関数	なし
	定義	XMAP3 共通定義 (mcftxp) <ul style="list-style-type: none"> • -c オプションの actcnaction オペランド
	コマンド	mcfxp コマンド <ul style="list-style-type: none"> • -r オプション
変更		なし
削除	関数	CBLDCMCF('SEND△△△△')
		SEND - メッセージの送信 <ul style="list-style-type: none"> • PAGE NAME 句 • PAGE CONTROL 句
	定義	マッピングサービス定義 <ul style="list-style-type: none"> • XMAPC サーバホスト名 • XMAPC サーバサービス番号
		マッピングサービス属性定義 <ul style="list-style-type: none"> • PGCSTDPN オペランド • PGCALTPN オペランド • PGCMNAME オペランド • PGCCNT オペランド • PGCPOLSZ オペランド
		コネクション定義 (mcftalccn) <ul style="list-style-type: none"> • -p オプションの dflpagec オペランド • -z オプションの remote オペランド
コマンド	dcmaphg コマンド	

種別	分類	内容
削除	コマンド	<ul style="list-style-type: none"> • -mP • -aP
		mcftactcn コマンド <ul style="list-style-type: none"> • -S

TP1/NET/XMAP3 07-50 での動作の変更はありません。

TP1/NET/XMAP3 07-50 でのデフォルト値の変更はありません。

付録 A.2 07-01 での変更点

TP1/NET/XMAP3 07-01 での関数，定義およびコマンドの追加・変更・削除を次の表に示します。

表 A-2 TP1/NET/XMAP3 07-01 での関数，定義およびコマンドの追加・変更・削除

種別	分類	内容
追加	関数	dc_mcf_tactcn
		dc_mcf_tactle
		dc_mcf_tdctcn
		dc_mcf_tdctle
		dc_mcf_tlscn
		dc_mcf_tlsle
		CBLDCMCF('TACTCN△△')
		CBLDCMCF('TACTLE△△')
		CBLDCMCF('TDCTCN△△')
		CBLDCMCF('TDCTLE△△')
		CBLDCMCF('TLSCN△△△')
		CBLDCMCF('TLSLE△△△')
		送信メッセージ通番編集 UOC
	定義	XMAP3 共通定義 (mcftxp) <ul style="list-style-type: none"> • -g オプション, closebutton オペランド
システムサービス情報定義 <ul style="list-style-type: none"> • 環境変数 XPCLOSEWAITTIME 		
変更	コマンド	mcfxpr コマンド
変更		なし

種別	分類	内容
削除		なし

TP1/NET/XMAP3 07-01 での動作の変更点を次に示します。

表 A-3 TP1/NET/XMAP3 07-01 での動作の変更点

分類	内容
定義	定義オブジェクトファイル生成時に、印刷サービス用のコネクション配下に二つ以上の論理端末が定義されている場合（定義不正）、定義オブジェクトファイルの生成を中止し、KFCA11557-E メッセージを出力するように変更
その他	MCF 性能検証用トレースに、TP1/NET/XMAP3 のメッセージ送受信イベントを取得するように変更

TP1/NET/XMAP3 07-01 でのデフォルト値の変更はありません。

付録 A.3 07-00 での変更点

TP1/NET/XMAP3 07-00 での関数、定義およびコマンドの追加・変更・削除はありません。

TP1/NET/XMAP3 07-00 での動作の変更はありません。

TP1/NET/XMAP3 07-00 でのデフォルト値の変更点を次に示します。

表 A-4 TP1/NET/XMAP3 07-00 でのデフォルト値の変更点

分類	内容
定義	マッピングサービス属性定義 <ul style="list-style-type: none"> RETRYCNT オペランドのデフォルト値を 0 から 10 に変更 RETRYSEC オペランドのデフォルト値を 0 から 3 に変更 ERRLOG4 オペランドのデフォルト値を YES から NO に変更
	XMAP3 共通定義 (mcftxp) <ul style="list-style-type: none"> -m オプション timeoutlog オペランドのデフォルト値を no から yes に変更
	コネクション定義 (mcftalccn) <ul style="list-style-type: none"> -b オプション bretrycnt オペランドのデフォルト値を 0 から 3 に変更
	論理端末定義 (mcftalcle) <ul style="list-style-type: none"> -z オプション remote オペランドのデフォルト値を no から yes に変更
コマンド	mcftlsle コマンドの -t オプションを削除

付録 B 旧製品からの移行に関する注意事項

ここでは、旧製品からの移行に関する注意事項について説明します。

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

表 B-1 バージョン 6 以前で使用していたソースファイルの互換性

ソースファイルの種類	ソースファイルを作成した言語	互換性
UAP	C 言語	バージョン 7 では、メッセージ送受信インタフェース、および MCF イベントインタフェースのそれぞれの引数ならびにパラメタの型が変更されていますが、この変更による UAP の処理への影響はなく、ソースファイルを変更しないで使用できます。 詳細については、「付録 C インタフェースの変更一覧 (バージョン 6 以前から移行する場合)」を参照してください。
	COBOL 言語	ソースファイルを変更しないで使用できます。
UOC	C 言語	バージョン 7 では、UOC インタフェースのそれぞれのパラメタの型およびリターン値の型が変更されていますが、この変更による UOC の処理への影響はなく、ソースファイルを変更しないで使用できます。 詳細については、「付録 C インタフェースの変更一覧 (バージョン 6 以前から移行する場合)」を参照してください。
マッピングサービス定義	—	ソースファイルを変更しないで使用できます。
マッピングサービス属性定義 MCF 通信構成定義 (プロトコル固有の定義)	—	バージョン 7 で省略時解釈値を変更しているため、必要に応じて定義ソースファイルを見直してください。 詳細は、表 B-2 を参照してください。
システムサービス共通情報定義	—	Windows 版のバージョン 5 以前では、max_socket_descriptors オペランドにプロトコル制御で使用するファイル記述子数 (コネクション数×2) を追加する必要がありましたが、バージョン 7 では不要です。ただし、バージョン 7 では max_open_fds オペランドにプロトコル制御で使用するファイル記述子数を追加してください。

(凡例)

— : 該当する内容がないことを表します。

表 B-2 バージョン 6 以前からバージョン 7 へ移行する場合の定義ファイルの互換性

定義の種類	オペランド	省略時解釈値		説明	
		バージョン 6 以前	バージョン 7		
マッピングサービス属性定義	RETRYCNT	0	10	マッピングサービスプロセスとのデータ送受信に失敗したときの再試行回数と再試行間隔です。 バージョン 6 以前でこのオペランドの指定を省略すると、再試行を実行しません。バージョン 7 では再試行回数が 10 回、再試行間隔が 3 秒となります。	
	RETRYSEC	0	3		
	ERRLOG4	YES	NO		
MCF 通信構成定義	mcftxp (XMAP3 共通定義)	-m timeoutlog	no	yes	XMAP3 Server でタイムアウトが発生した場合の通知の有無です。 バージョン 6 以前でこのオペランドの指定を省略すると、ログメッセージを出力しません。バージョン 7 では、ログメッセージを出力します。
	mcftalccn (コネクション定義の開始)	-b bretrycnt	0	3	コネクション確立時に障害が発生した場合に MCF が行う確立再試行回数です。 バージョン 6 以前でこのオペランドの指定を省略すると、無限に確立再試行を繰り返します。バージョン 7 では、3 回まで再試行を繰り返します。
	mcftalcle (論理端末定義)	-z remote	no	yes	プリンタ種別です。 バージョン 6 以前でこのオペランドの指定を省略すると、高機能ページプリンタとなります。バージョン 7 では、LAN 直結プリンタとなります。

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

バージョン 6 以前のインタフェースの変更一覧を示します。

ここで説明するインタフェースを次に示します。

表 C-1 インタフェースの変更一覧

変更されたインタフェース	バージョン 7 のマニュアルの該当箇所	
メッセージ送受信インタフェース	dc_mcf_contend	3. dc_mcf_contend – 継続問い合わせ応答の終了 (C 言語)
	dc_mcf_receive	3. dc_mcf_receive – メッセージの受信 (C 言語)
	dc_mcf_reply	3. dc_mcf_reply – 応答メッセージの送信 (C 言語)
	dc_mcf_resend	3. dc_mcf_resend – メッセージの再送 (C 言語)
	dc_mcf_send	3. dc_mcf_send – メッセージの送信 (C 言語)
	dc_mcf_tempget	3. dc_mcf_tempget – 一時記憶データの受け取り (C 言語)
	dc_mcf_tempput	3. dc_mcf_tempput – 一時記憶データの更新 (C 言語)
ユーザOWNコーディング	入力メッセージ編集 UOC	5.1.2 入力メッセージ編集 UOC インタフェース
	出力メッセージ編集 UOC	5.1.4 出力メッセージ編集 UOC インタフェース
MCF イベントインタフェース	5.2.3 MCF イベント情報の形式 (C 言語)	
MCF メイン関数のコーディング概要	10.2 MCF メイン関数の作成	

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

付録 C.1 メッセージ送受信インタフェース

(1) dc_mcf_contend – 継続問い合わせ応答の終了

(a) ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcpcf.h> int dc_mcf_contend(<u>long</u> action, char *resv01)</pre>	<pre>#include <dcpcf.h> int dc_mcf_contend(<u>DCLONG</u> action, char *resv01)</pre>

(b) K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcpcf.h> int dc_mcf_contend(action, resv01) long action; char *resv01;</pre>	<pre>#include <dcpcf.h> int dc_mcf_contend(action, resv01) DCLONG action; char *resv01;</pre>

(2) dc_mcf_receive – メッセージの受信

(a) ANSI C, C++の形式

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

(b) K&R 版 C の形式

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

(3) dc_mcf_reply – 応答メッセージの送信

(a) ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcpcf.h> int dc_mcf_reply(long action,</pre>	<pre>#include <dcpcf.h> int dc_mcf_reply(DCLONG action,</pre>

バージョン 6 以前	バージョン 7
<pre> long commform, char *resv01, char *mapname, char *senddata, long sdataleng, char *nextap, long opcd) </pre>	<pre> DCLONG commform, char *resv01, char *mapname, char *senddata, DCLONG sdataleng, char *nextap, DCLONG opcd) </pre>

(b) K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre> #include <dcpcf.h> int dc_mcf_reply(action, commform, resv01, mapname, senddata, sdataleng, nextap, opcd) long action; long commform; char *resv01; char *mapname; char *senddata; long sdataleng; char *nextap; long opcd; </pre>	<pre> #include <dcpcf.h> int dc_mcf_reply(action, commform, resv01, mapname, senddata, sdataleng, nextap, opcd) DCLONG action; DCLONG commform; char *resv01; char *mapname; char *senddata; DCLONG sdataleng; char *nextap; DCLONG opcd; </pre>

(4) dc_mcf_resend – メッセージの再送

(a) ANSI C, C++ の形式

バージョン 6 以前	バージョン 7
<pre> #include <dcpcf.h> int dc_mcf_resend(long action, long commform, char *rtermnam, char *resv01, long oseqid, long orgseq, char *otermnam, char *resv02, char *resv03, char *resv04, long opcd) </pre>	<pre> #include <dcpcf.h> int dc_mcf_resend(DCLONG action, DCLONG commform, char *rtermnam, char *resv01, DCLONG oseqid, DCLONG orgseq, char *otermnam, char *resv02, char *resv03, char *resv04, DCLONG opcd) </pre>

(b) K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre> #include <dcpcf.h> int dc_mcf_resend(action, commform, rtermnam, </pre>	<pre> #include <dcpcf.h> int dc_mcf_resend(action, commform, rtermnam, </pre>

バージョン 6 以前	バージョン 7
<pre> resv01, oseqid, orgseq, otermnam, resv02, resv03, resv04, opcd) long action; long commform; char *rtermnam; char *resv01; long oseqid; long orgseq; char *otermnam; char *resv02; char *resv03; char *resv04; long opcd; </pre>	<pre> resv01, oseqid, orgseq, otermnam, resv02, resv03, resv04, opcd) DCLONG action; DCLONG commform; char *rtermnam; char *resv01; DCLONG oseqid; DCLONG orgseq; char *otermnam; char *resv02; char *resv03; char *resv04; DCLONG opcd; </pre>

(5) dc_mcf_send – メッセージの送信

(a) ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre> #include <dcmcf.h> int dc_mcf_send(long action, long commform, char *termnam, char *resv01, char *senddata, long sdataleng, char *resv02, long opcd) </pre>	<pre> #include <dcmcf.h> int dc_mcf_send(DCLONG action, DCLONG commform, char *termnam, char *resv01, char *senddata, DCLONG sdataleng, char *resv02, DCLONG opcd) </pre>

(b) K&R 版 C の形式

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

(6) dc_mcf_tempget – 一時記憶データの受け取り

(a) ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcacf.h> int dc_mcf_tempget(long action, char *getdata, long gtempleng, long *gdataleng, char *resv01)</pre>	<pre>#include <dcacf.h> int dc_mcf_tempget(DCLONG action, char *getdata, DCLONG gtempleng, DCLONG *gdataleng, char *resv01)</pre>

(b) K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcacf.h> int dc_mcf_tempget(action, getdata, gtempleng, gdataleng, resv01) long action; char *getdata; long gtempleng; long *gdataleng; char *resv01;</pre>	<pre>#include <dcacf.h> int dc_mcf_tempget(action, getdata, gtempleng, gdataleng, resv01) DCLONG action; char *getdata; DCLONG gtempleng; DCLONG *gdataleng; char *resv01;</pre>

(7) dc_mcf_tempput – 一時記憶データの更新

(a) ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcacf.h> int dc_mcf_tempput(long action, char *putdata, long pdataleng, char *resv01)</pre>	<pre>#include <dcacf.h> int dc_mcf_tempput(DCLONG action, char *putdata, DCLONG pdataleng, char *resv01)</pre>

(b) K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcacf.h> int dc_mcf_tempput(action, putdata, pdataleng, resv01) long action; char *putdata; long pdataleng; char *resv01;</pre>	<pre>#include <dcacf.h> int dc_mcf_tempput(action, putdata, pdataleng, resv01) DCLONG action; char *putdata; DCLONG pdataleng; char *resv01;</pre>

付録 C.2 ユーザOWNコーディング

(1) 入力メッセージ編集 UOC

(a) 形式

ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcmfuoc.h> #include <dcmxpn.h> long uoc_func(dcmcf_uoc_min_n *parm)</pre>	<pre>#include <dcmfuoc.h> #include <dcmxpn.h> DCLONG uoc_func(dcmcf_uoc_min_n *parm)</pre>

K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcmfuoc.h> #include <dcmxpn.h> long uoc_func(parm) dcmcf_uoc_min_n *parm ;</pre>	<pre>#include <dcmfuoc.h> #include <dcmxpn.h> DCLONG uoc_func(parm) dcmcf_uoc_min_n *parm ;</pre>

(b) パラメタの内容

dcmcf_uoc_min_n の内容

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

dcmcf_uocbuff_list_n (バッファリスト) の内容

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

dcmf_uocbufinf_n (バッファ情報) の内容

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

(2) 出力メッセージ編集 UOC

(a) 形式

ANSI C, C++の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcmfuoc.h> #include <dcmxpn.h> long uoc_func(dcmcf_uoc_mout_n *parm)</pre>	<pre>#include <dcmfuoc.h> #include <dcmxpn.h> DCULONG uoc_func(dcmcf_uoc_mout_n *parm)</pre>

K&R 版 C の形式

バージョン 6 以前	バージョン 7
<pre>#include <dcmfuoc.h> #include <dcmxpn.h> long uoc_func(parm) dcmcf_uoc_mout_n *parm ;</pre>	<pre>#include <dcmfuoc.h> #include <dcmxpn.h> DCULONG uoc_func(parm) dcmcf_uoc_mout_n *parm ;</pre>

(b) パラメタの内容

dcmf_uoc_mout_n の内容

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

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

入力メッセージ編集 UOC のパラメタの内容と同じです。「付録 C.2(1)(b) パラメタの内容」を参照してください。

付録 C.3 MCF イベントインタフェース

(1) MCF イベントの共通ヘッダの形式

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

(2) ERREVT1 の形式

バージョン 6 以前とバージョン 7 で、差異はありません。

(3) ERREVT2 の形式

バージョン 6 以前とバージョン 7 で、差異はありません。

(4) ERREVT3 の形式

バージョン 6 以前とバージョン 7 で、差異はありません。

(5) ERREVTA の形式

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

(6) SCMPEVT の形式

バージョン 6 以前	バージョン 7
<pre>struct dc_mcf_scmpevt_type { struct dc_mcf_evtheader evtheader; long output_no; char map_name[9]; char msg_type; char reason_code; chat reserve01[5]; long user_leng; char pro_indv_inf[16]; char user_data[16]; };</pre>	<pre>struct dc_mcf_scmpevt_type { struct dc_mcf_evtheader evtheader; DCLONG output_no; char map_name[9]; char msg_type; char reason_code; chat reserve01[5]; DCLONG user_leng; char pro_indv_inf[16]; char user_data[16]; };</pre>

(7) SERREVT の形式

バージョン 6 以前	バージョン 7
<pre>struct dc_mcf_serrevt_type { struct dc_mcf_evtheader evtheader; long output_no; char map_name[9]; char msg_type; char send_error_code; unsigned char trouble_code; unsigned char detail_info; unsigned char error_code; unsigned char error_subcode; chat reserve01[1]; long user_leng; char pro_indv_inf[16]; char user_data[16]; }; struct dcmxpn_sevtpro_type { long serr_reason1; long serr_reason2; long serr_detail; char reserve1[4]; };</pre>	<pre>struct dc_mcf_serrevt_type { struct dc_mcf_evtheader evtheader; DCLONG output_no; char map_name[9]; char msg_type; char send_error_code; unsigned char trouble_code; unsigned char detail_info; unsigned char error_code; unsigned char error_subcode; chat reserve01[1]; DCLONG user_leng; char pro_indv_inf[16]; char user_data[16]; }; struct dcmxpn_sevtpro_type { DCLONG serr_reason1; DCLONG serr_reason2; DCLONG serr_detail; char reserve1[4]; };</pre>

(8) VERREVT の形式

バージョン 6 以前	バージョン 7
<pre>struct dcmxpn_verrevt_type { struct dc_mcf_evtheader header; long le_kind; long err_fact; long err_reason1; long err_reason2; long err_detail; char reserve1[36]; };</pre>	<pre>struct dcmxpn_verrevt_type { struct dc_mcf_evtheader header; DCLONG le_kind; DCLONG err_fact; DCLONG err_reason1; DCLONG err_reason2; DCLONG err_detail; char reserve1[36]; };</pre>

(9) VOPNEVT, VCLSEVT の形式

バージョン 6 以前	バージョン 7
<pre>struct dcmxpn_statevt_type { struct dc_mcf_evtheader header; long le_kind; char reserve1[52]; };</pre>	<pre>struct dcmxpn_statevt_type { struct dc_mcf_evtheader header; DCLONG le_kind; char reserve1[52]; };</pre>

付録 C.4 MCF メイン関数のコーディング概要

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

(1) ANSI C, C++ の場合

(a) バージョン 6 以前

```
#include <dcmxp.h> /*TP1/NET/XMAP3用ヘッダファイル */
extern long msgrcv01(dcmcf_uoc_min_n *); /*入力メッセージ編集UOC関数extern宣言 */
extern long msgsend01(dcmcf_uoc_mout_n *); /*出力メッセージ編集UOC関数extern宣言 */
extern dcmcf_uoc_t dcmcf_uoctbl; /*UOCテーブルextern宣言 */
int main()
{
    dcmcf_uoctbl.msgrcv = (dcmcf_uocfunc)msgrcv01; /*入力メッセージ編集UOCアドレス設定 */
    dcmcf_uoctbl.msgsend = (dcmcf_uocfunc)msgsend01; /*出力メッセージ編集UOCアドレス設定 */
    dc_mcf_svstart(); /*スタート関数発行 */
    return 0;
}
```

(b) バージョン 7

```
#include <dcmxp.h> /*TP1/NET/XMAP3用ヘッダファイル */
extern DCLONG msgrcv01(dcmcf_uoc_min_n *); /*入力メッセージ編集UOC関数extern宣言 */
extern DCLONG msgsend01(dcmcf_uoc_mout_n *); /*出力メッセージ編集UOC関数extern宣言 */
extern dcmcf_uoc_t dcmcf_uoctbl; /*UOCテーブルextern宣言 */
int main()
{
    dcmcf_uoctbl.msgrcv = (dcmcf_uocfunc)msgrcv01; /*入力メッセージ編集UOCアドレス設定 */
    dcmcf_uoctbl.msgsend = (dcmcf_uocfunc)msgsend01; /*出力メッセージ編集UOCアドレス設定 */
    dc_mcf_svstart(); /*スタート関数発行 */
    return 0;
}
```

(2) K&R 版 C の場合

(a) バージョン 6 以前

```
#include <dcmap.h> /*TP1/NET/XMAP3用ヘッダファイル */
extern long msgrcv01(); /*入力メッセージ編集UOC関数extern宣言 */
extern long msgsend01(); /*出力メッセージ編集UOC関数extern宣言 */
extern dcmcf_uoc_t dcmcf_uoctbl; /*UOCテーブルextern宣言 */
main()
{
    dcmcf_uoctbl.msgrcv = msgrcv01; /*入力メッセージ編集UOCアドレス設定 */
    dcmcf_uoctbl.msgsend = msgsend01; /*出力メッセージ編集UOCアドレス設定 */
    dc_mcf_svstart(); /*スタート関数発行 */
}
```

(b) バージョン 7

```
#include <dcmap.h> /*TP1/NET/XMAP3用ヘッダファイル */
extern DCLONG msgrcv01(); /*入力メッセージ編集UOC関数extern宣言 */
extern DCLONG msgsend01(); /*出力メッセージ編集UOC関数extern宣言 */
extern dcmcf_uoc_t dcmcf_uoctbl; /*UOCテーブルextern宣言 */
main()
{
    dcmcf_uoctbl.msgrcv = msgrcv01; /*入力メッセージ編集UOCアドレス設定 */
    dcmcf_uoctbl.msgsend = msgsend01; /*出力メッセージ編集UOCアドレス設定 */
    dc_mcf_svstart(); /*スタート関数発行 */
}
```

付録 D MCF 性能検証用トレースの取得

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

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

付録 D.1 MCF 固有情報の出力情報

ここでは、メッセージ送受信時、および UOC 呼び出し時の MCF 性能検証用トレースのダンプ出力情報について説明します。

(1) メッセージ送受信時

論理端末単位に相手システムと送受信するメッセージの情報を取得します。MCF 固有情報のダンプ出力情報を、次の表に示します。

表 D-1 メッセージ送受信時の MCF 固有情報のダンプ出力情報

イベント ID	オフセット				
	0x0000~ 0x0003	0x0004~ 0x0007	0x0008~0x000f	0x0010~ 0x0017	0x0018~0x001f
0xa000	MCF 識別子	スレッド ID	—	入力元論理端末名	—
0xa001			—	出力元論理端末名	マップ名称

(凡例)

—：情報を取得しません。

(2) UOC 呼び出し時

TP1/NET/XMAP3 で使用する UOC の情報を取得します。MCF 固有情報のダンプ出力情報、および UOC 名称の出力情報を、以降の表に示します。

表 D-2 UOC 呼び出し時の MCF 固有情報のダンプ出力情報

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

(凡例)

－：情報を取得しません。

表 D-3 UOC 名称の出力情報

UOC の種類	UOC 名称出力情報
入力メッセージ編集 UOC	"MSGRCV"
出力メッセージ編集 UOC	"MSGSEND"
送信メッセージ通番編集 UOC	"SEND_UOC"

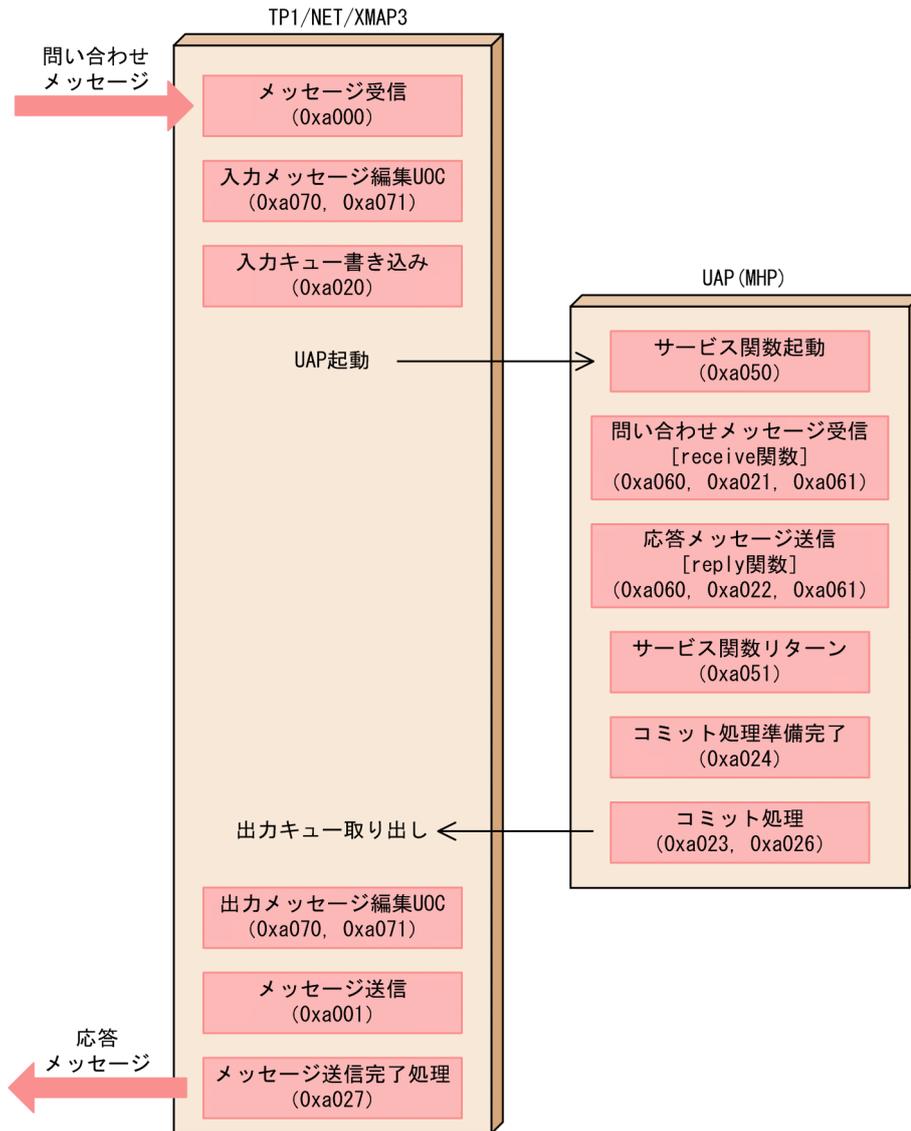
付録 D.2 MCF 性能検証用トレースの取得タイミング

MCF 性能検証用トレースの取得タイミングを、問い合わせメッセージ受信時、および一方送信メッセージ送信時に分けて説明します。

(1) 問い合わせメッセージ受信時

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

図 D-1 問い合わせメッセージ受信時の MCF 性能検証用トレースの取得タイミング



(凡例)

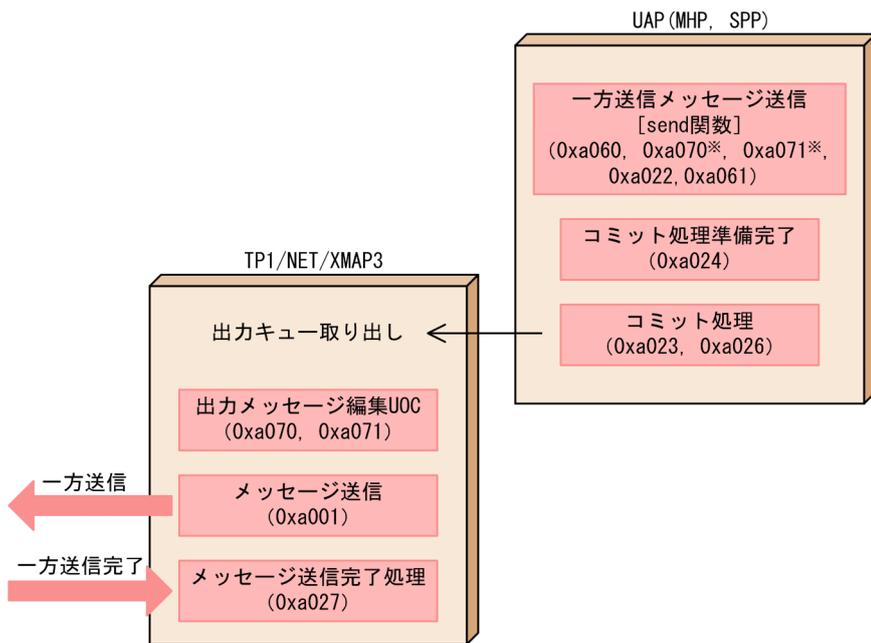
■ : MCF性能検証用トレースの取得タイミング

() : イベントID

(2) 一方送信メッセージ送信時

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

図 D-2 一方送信メッセージ送信時の MCF 性能検証用トレースの取得タイミング



(凡例)

■ : MCF性能検証用トレースの取得タイミング

() : イベントID

注※

送信メッセージ通番編集 UOC 使用時に該当します。

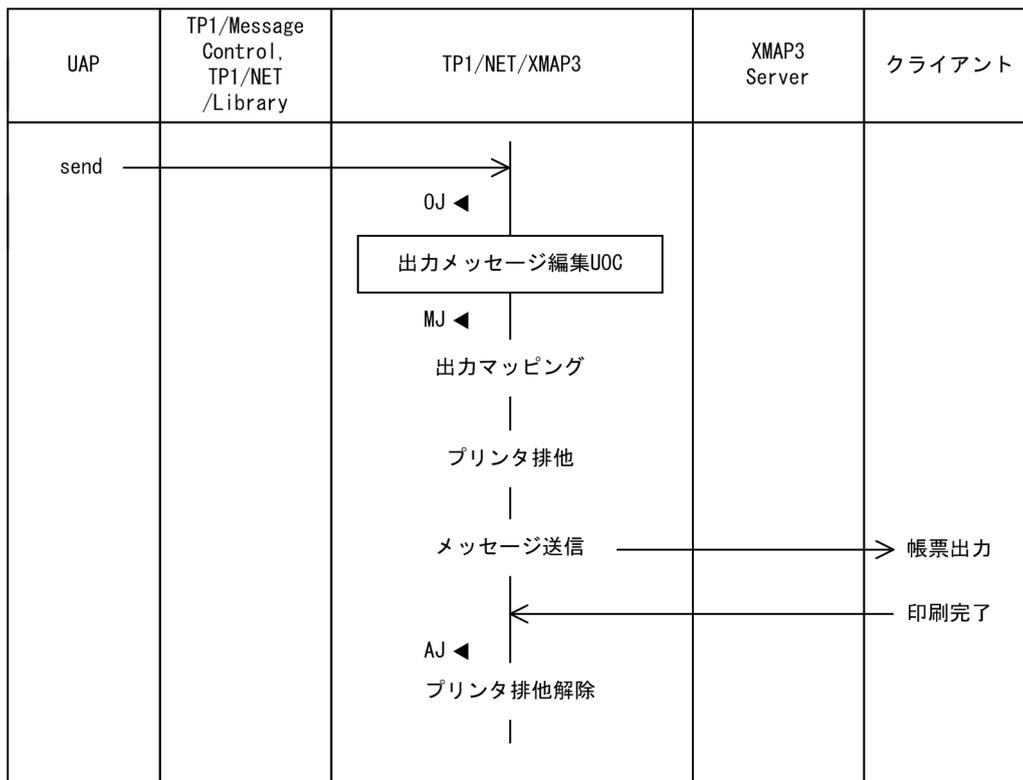
付録 D.3 MCF 性能検証用トレースの取得量

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

表 D-4 MCF 性能検証用トレースの取得量

メッセージの送受信形態	トレース取得量 (単位: キロバイト)
問い合わせメッセージの受信	3.0
一方送信メッセージの送信	1.8

図 E-2 帳票印刷の流れ



- (凡例)
- 0J ◀ : メッセージ出力ジャーナル取得
 - MJ ◀ : メッセージジャーナル取得
 - AJ ◀ : メッセージ送信完了ジャーナル取得

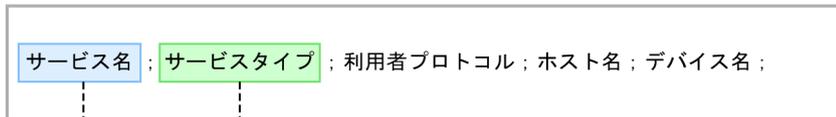
付録 F XMAP3 Server サービス名との関係

TP1/NET/XMAP3 の定義と XMAP3 Server サービス名との関係を次の図に示します。XMAP3 Server サービス名については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

図 F-1 XMAP3 Server サービス名との関係

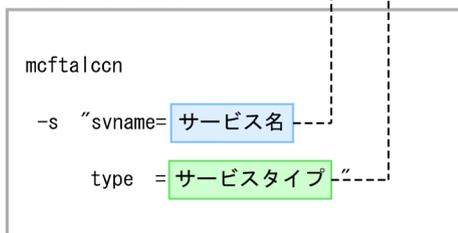
● XMAP3 Server サービス名

XMAP3 Server サービス名ファイル (/etc/opt/HIXMAP/XPWhosts)



● TP1/NET/XMAP3の定義

コネクション定義



また、XMAP3 Server サービス名ファイル、および OS ホストアドレス管理ファイル (hosts ファイル) のコーディング例を次に示します。

• XMAP3 Server サービス名ファイルのコーディング例

```
#
# XMAP3 Server サービス名定義 (/etc/opt/HIXMAP/XPWhosts)
#
XPWDSP1;DSP;TCP;h3050B;h3050B:0.0;
XPWPRT1;PRT;TCP;h3050C;PR1;
XPWDSP2;DSP;TCP;h3020A;h3020A;
```

注

この定義は XMAP3 サーバを起動するホスト内に定義します。

詳細は、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

• OS ホストアドレス管理ファイル (hosts ファイル) のコーディング例

```
192.10.10.1    h3050A
192.10.10.2    h3050B
192.10.10.3    h3050C
192.10.10.4    h3020A
```

注

この定義はネットワーク内の全ホスト内に定義します。

付録 G 表示印刷サービスの起動でコネクション確立をするための環境設定

表示印刷サービスの起動でコネクションと論理端末を確立して初期画面を表示するための環境作成について説明します。

表示印刷サービスを起動すると同時に XMAP3 Server 側で AP を実行できます。この場合は、XMAP3 Server サービス名ファイルの AP パス名称で、実行する AP を定義します。この XMAP3 の機能を利用して、該当する表示印刷サービスに対応するコネクション確立の運用コマンドを発行するようにします。

なお、XMAP3 サーバの環境定義ファイルおよび XMAP3 Server のサービス名ファイルに指定するサービス起動時に受け渡す環境変数名には、表示サービスと印刷サービスで別の環境変数名を指定してください。

付録 G.1 TP1/NET/XMAP3 の運用コマンドを発行するプログラムの作成

XMAP3 Server サービス名ファイルに指定した環境変数から起動されたサービスを求めて、SYSTEM サブルーチンで対応するコネクションに運用コマンド (mcftactcn) を発行するプログラムを作成します。

付録 G.2 XMAP3 Server サービス名ファイルの設定

XMAP3 Server サービス名ファイルの AP ホスト名称、AP パス名称、および環境変数名を、該当するサービスの内容に合わせて設定します。

AP ホスト名称には、OpenTP1 システムのホスト名称を指定します。

AP パス名称には、TP1/NET/XMAP3 の運用コマンドを発行するプログラムの完全パス名称を指定します。

環境変数名には、TP1/NET/XMAP3 の運用コマンドを発行プログラムで起動されたサービスを求めるために指定した環境変数を指定します。

付録 G.3 TP1/NET/XMAP3 プロトコル固有定義の設定

TP1/NET/XMAP3 プロトコル固有定義に必ず設定しなければならない項目を次に示します。

- コネクション定義 (mcftalccn) の -i オプションに manual を設定します。
- 論理端末定義 (mcftalcle) の -i オプションに auto を設定します。

初期画面の表示は、論理端末定義 (mcftalcle -p) の initmap オペランドを指定して初期画面を表示する方式か、VOPNEVT の発生によって起動したアプリケーションで初期画面を表示する方式のどちらかで行ないます。

付録 G.4 TP1/NET/XMAP3 運用コマンドを発行するプログラムと XMAP3 Server サービス名ファイルの例

運用コマンドを発行するプログラムと XMAP3 Server サービス名ファイルの例を示します。

- 運用コマンドを発行するプログラムの例

ファイル名: `/BeTRAN/usr/bin/actcn_dsp` ※1

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
main ()
{
    char dsp[512];
    char onlinecom[512];

    static char svname[6][11] = { "XPPDSP1", "XPNCN5",
                                  /* サービス名 コネクション名 */
                                  /* コネクション名はmcftalccn */
                                  "XPPPRT1", "XPNCN6",
                                  /* 定義コマンドで指定する */
                                  "XPPDSP2", "XPNCN7" };
                                  /* コネクションID */
                                  ※2

    struct mcfcmd {
        char mcfid[32];
        char svid[32];
        char prgid[8];
        char curname[32];
        char cnflg[8];
    } mcfcmd_txt;
    int svno = 6;
    int rtycnt = 6; /* コネクション確立処理のリトライ回数 */
    int tblp = 0;
    int loopc = 0;
    int eflag = 1;
    FILE *fp;

    strcpy(dsp, getenv("XPPDSP"));
    while ((eflag != 0) && (loopc < svno)) {
        tblp = loopc;
        eflag = strcmp(dsp, svname[tblp]);
        loopc = loopc + 2;
    }
    if (eflag == 0) {
        sprintf(onlinecom, "/BeTRAN/bin/mcftdctcn -s 01 -f -c %s",
              svname[loopc]);
        system(onlinecom);
        ※3
    }
    eflag = 1;
    while ((eflag != 0) && (rtycnt > 0)) {
        sprintf(onlinecom, "/BeTRAN/bin/mcftlscn -s 01 -c %s > /tmp/work1",
              svname[tblp]);
        system(onlinecom);
        ※4
        sprintf(onlinecom, "grep \"KFCA10361\" /tmp/work1 > /tmp/work2");
        system(onlinecom);
        ※5
        fp = fopen("/tmp/work2", "r");
    }
}
```

```

fscanf(fp,"%s %s %s %s %s",mfcfcmd_txt.mcfid,mfcfcmd_txt.svid,
      mfcfcmd_txt.cname,mfcfcmd_txt.prgid,mfcfcmd_txt.cnflg);
eflag = strcmp(mfcfcmd_txt.cnflg,"DCT");
      /* コネクション状態の確認 */
if (eflag == 0) {
    sprintf(onlinecom,"/BeTRAN/bin/mcftactcn -s 01 -c %s",
          svname[tblp]);
    system(onlinecom);      ※6
}
sleep(10);
rtycnt--;
}
}

```

- XMAP3 Server サービス名ファイルの例

```

#サービス名称;サービスタイプ;利用者プロトコル;ホスト名称;デバイス名称;APホスト名称;APパス
名称;環境変数名;
XPPDSP1;DSP;TCP;h3020A;XPPDSP1;h3050A;/BeTRAN/usr/bin/actcn_dsp;XPPDSP;      ※1
XPPPRT1;PRT;TCP;FLORA1;XPPPRT1;h3050A;/BeTRAN/usr/bin/actcn_prt;XPPPRT;      ※1
XPPDSP2;DSP;TCP;FLORA2;XPPDSP2;h3050A;/BeTRAN/usr/bin/actcn_dsp;XPPDSP;      ※1

```

注※1, 2, 3

太字の部分は、運用コマンドを発行するプログラムと XMAP3 Server サービス名ファイルとで一致させる必要があります。

注※4

/BeTRAN/bin/mcftdctcn は、OpenTP1 プログラムファイルのパス名です。

注※5

/BeTRAN/bin/mcftlscn は、OpenTP1 プログラムファイルのパス名です。

注※6

/BeTRAN/bin/mcftactcn は、OpenTP1 プログラムファイルのパス名です。

付録 H ユーザアプリケーションプログラムの作成例

ここでは、コーディング例と提供するサンプルコーディングについて説明します。

付録 H.1 コーディング例

使用する言語ごとに UAP のコーディング例を示します。

(1) C 言語 (K&R 版)

C 言語 (K&R 版) を使用した UAP のコーディング例を次に示します。

(a) コーディング例 (C 言語)

```
/*
 * MHP サービス関数
 */
#include <stdio.h>
#include <dcmcf.h>
/*
 * 論理マップの取り込み
 */
#include "LOG0101I.h"          .....1.
#include "LOG02010.h"         .....1.
#include "LOG10010.h"         .....1.

void uap01()
{
    int rtn_cod;

    DCLONG action;
    DCLONG commform;
    char recv_term[9];
    char recv_map[9];
    DCLONG rdataleng;
    DCLONG inbufleng;
    DCLONG time;

    static char send_term[9] = "XPNLE201";
    static char send_map[9] = "MAM10";
    DCLONG sdataleng;
    char *pagecdat = NULL;
    DCLONG opcd;

    static char resv01[9] = "¥0";
    static char rply_map[9] = "MAM02";
    static char nextap[9] = "¥0";

/*
 * メッセージの受信
 */
    action = DCMCFFRST | DCMCFBUF2;
```

```

commform = DCNOFLAGS;
inbufleng = (DCLONG)LOG0101I.LOG0101S;
rtn_cod = dc_mcf_receive(action,
    commform,
    recv_term,
    recv_map,
    &LOG0101I,          .....2.
    &rdatalleng,
    inbufleng,
    &time);

/*
*****
*
*          データの処理
*
*
*****
*/

/*
* 一方送信メッセージの送信
*/
action = DCMCFEMI | DCMCFNORM | DCMCFNSEQ | DCMCFBUF2;
commform = DCMCFOUT;
sdatalleng = (DCLONG)LOG10010.LOG1001L -4;
opcd = DCNOFLAGS;
rtn_cod = dc_mcf_send(action,
    commform,
    send_term,
    send_map,
    &LOG10010,          .....2.
    sdatalleng,
    pagecdat,
    opcd);

/*
* 応答メッセージの送信
*/
action = DCMCFEMI|DCMCFBUF2;
commform = DCNOFLAGS;
sdatalleng = (DCLONG)LOG02010.LOG0201L -4;
opcd = DCNOFLAGS;
rtn_cod = dc_mcf_reply(action,
    commform,
    resv01,
    rply_map,
    &LOG02010,          .....2.
    sdatalleng,
    nextap,
    opcd);
}

```

コーディング例中の 1.および 2.は次の意味になります。

1. XMAP3 で生成した入力・出力論理マップを#include で取り込みます。

2. recvdata・senddata には、XMAP3 で作成した入力・出力論理マップの先頭アドレスを指定します。

(2) COBOL 言語

COBOL 言語を使用した UAP のコーディング例を次に示します。

(a) コーディング例 (COBOL 言語)

```
*
*****
*   MHPサービスプログラム (COBOL)   *
*****
*
IDENTIFICATION DIVISION.

PROGRAM-ID. MHPC.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*
*****
*   ワーク変数                       *
*****
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
*****
*   論理マップコピー                 *
*   LOG0101I : 入力ロジカルマップ   *
*   LOG02010 : 出力ロジカルマップ   *
*   LOG10010 : 出力ロジカルマップ   *
*****
*
COPY LOG0101I.          .....1.
COPY LOG02010.          .....1.
COPY LOG10010.          .....1.
*
*****
*   メッセージの受信データ領域       *
*****
*
01 RECV-PARM1.
   02 RE-NAME           PIC X(8) VALUE 'RECEIVE '.
   02 RE-STATUS        PIC X(5).
   02 FILLER           PIC X(3).
   02 RE-SEG           PIC X(4) VALUE 'FRST'.
   02 RE-RSV1          PIC X(4) VALUE SPACE.
   02 RE-DATE          PIC 9(8).
   02 RE-TIME          PIC 9(8).
   02 RE-LENG          PIC 9(9) COMP VALUE 44.
   02 RE-RSV2          PIC X(4) VALUE SPACE.
   02 RE-RSV3          PIC X(4) VALUE SPACE.
   02 RE-RSV4          PIC X(4) VALUE SPACE.
   02 RE-RSV5          PIC X(4) VALUE SPACE.
   02 RE-RSV6          PIC X(8) VALUE SPACE.
```

```

02 RE-RSV7          PIC X(4) VALUE SPACE.
02 RE-RSV8          PIC X(8) VALUE SPACE.
02 RE-RSV9          PIC X(4) VALUE SPACE.
02 RE-RSV10         PIC 9(9) COMP VALUE ZERO.
02 RE-RSV11         PIC 9(9) COMP VALUE ZERO.
02 RE-RSV12         PIC X(1) VALUE SPACE.
02 RE-BUFTYPE       PIC X(1) VALUE '2'.
02 RE-RSV13         PIC X(14) VALUE LOW-VALUE.
01 RECV-PARM2.
02 RE-RSV14         PIC X(4) VALUE SPACE.
02 RE-TERMNAM       PIC X(8).
02 RE-MAPNAME       PIC X(8).
02 RE-RSV15         PIC X(8) VALUE SPACE.
02 RE-RSV16         PIC X(28) VALUE LOW-VALUE.
*
*****
* 一方送信メッセージの送信データ領域 *
*****
*
01 SEND-PARM1.
02 SE-NAME          PIC X(8) VALUE 'SEND  '.
02 SE-STATUS        PIC X(5).
02 FILLER           PIC X(3).
02 SE-RSV1          PIC X(4) VALUE SPACE.
02 SE-RSV2          PIC X(4) VALUE SPACE.
02 SE-RSV3          PIC 9(8).
02 SE-RSV4          PIC 9(8).
02 SE-RSV5          PIC 9(9) COMP VALUE ZERO.
02 SE-SEG           PIC X(4) VALUE 'EMI '.
02 SE-SYNC          PIC X(4) VALUE SPACE.
02 SE-RSV6          PIC X(4) VALUE 'NORM'.
02 SE-NO            PIC X(4) VALUE 'NSEQ'.
02 SE-RSV7          PIC X(8) VALUE SPACE.
02 SE-RSV8          PIC X(4) VALUE SPACE.
02 SE-RSV9          PIC X(8) VALUE SPACE.
02 SE-RSV10         PIC X(4) VALUE SPACE.
02 SE-RSV11         PIC 9(9) COMP VALUE ZERO.
02 SE-RSV12         PIC 9(9) COMP.
02 SE-MAPMOD        PIC X(1) VALUE '0'.
02 SE-BUFTYPE       PIC X(1) VALUE '2'.
02 SE-PAGEC         PIC X(1) VALUE '0'.
02 SE-RSV13         PIC X(13) VALUE LOW-VALUE.
01 SEND-PARM2.
02 SE-COMF          PIC X(4) VALUE 'OUT '.
02 SE-TERMNAM       PIC X(8) VALUE 'XPNLE201'.
02 SE-MAPNAM        PIC X(8) VALUE 'MAM10 '.
02 SE-RSV14         PIC X(8) VALUE SPACE.
02 SE-PAGECNAM      PIC X(6) VALUE SPACE.
02 SE-RSV15         PIC X(28) VALUE LOW-VALUE.
*
*****
* 応答メッセージの送信データ領域 *
*****
*
01 RPLY-PARM1.
02 RP-NAME          PIC X(8) VALUE 'REPLY  '.
02 RP-STATUS        PIC X(5).
02 FILLER           PIC X(3).

```

```

02 RP-RSV1          PIC X(4) VALUE SPACE.
02 RP-RSV2          PIC X(4) VALUE SPACE.
02 RP-RSV3          PIC 9(8).
02 RP-RSV4          PIC 9(8).
02 RP-RSV5          PIC 9(9) COMP VALUE ZERO.
02 RP-SEG           PIC X(4) VALUE 'EMI '.
02 RP-SYNC          PIC X(4) VALUE SPACE.
02 RP-RSV6          PIC X(4) VALUE SPACE.
02 RP-RSV7          PIC X(4) VALUE SPACE.
02 RP-RSV8          PIC X(8) VALUE SPACE.
02 RP-OPT           PIC X(4) VALUE SPACE.
02 RP-RSV9          PIC X(8) VALUE SPACE.
02 RP-RSV10         PIC X(4) VALUE SPACE.
02 RP-RSV11         PIC 9(9) COMP VALUE ZERO.
02 RP-RSV12         PIC 9(9) COMP VALUE ZERO.
02 RP-RSV13         PIC X(1) VALUE '0'.
02 RP-BUFTYPE       PIC X(1) VALUE '2'.
02 RP-RSV14         PIC X(14) VALUE LOW-VALUE.
01 RPLY-PARM2.
02 RP-RSV15         PIC X(4) VALUE SPACE.
02 RP-RSV16         PIC X(8) VALUE SPACE.
02 RP-MAPNAM        PIC X(8) VALUE 'MAM02 '.
02 RP-NEXTAP        PIC X(8) VALUE SPACE.
02 RP-RSV18         PIC X(28) VALUE LOW-VALUE.
*
PROCEDURE DIVISION.
*
*****
*      メッセージの受信      *
*****
*
CALL 'CBLDCMCF' USING RECV-PARM1 RECV-PARM2 LOG0101I.  ....2.
*
*****
*
*      データの処理      *
*
*****
*
*****
*      一方送信メッセージの送信      *
*****
*
CALL 'CBLDCMCF' USING SEND-PARM1 SEND-PARM2 LOG10010.  ....2.
*
*****
*      応答メッセージの送信      *
*****
*
CALL 'CBLDCMCF' USING RPLY-PARM1 RPLY-PARM2 LOG02010.  ....2.
*
*****
*      終了処理      *
*****

```

```
*****
*
EXIT PROGRAM.
```

コーディング例中の 1.および 2.は次の意味になります。

1. XMAP3 で生成した入力・出力論理マップを COPY 文で取り込みます。
2. 一意名 3 には, XMAP3 で作成した入力・出力論理マップを指定します。

(b) コーディング例 (データ操作言語)

```
*
*****
*   MHPサービスプログラム (DML)   *
*****
*
IDENTIFICATION DIVISION.

PROGRAM-ID. MHPD.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
*
*****
*   ワーク変数   *
*****
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
*****
*   論理マップコピー   *
*   LOG0101I : 入力ロジカルマップ   *
*   LOG02010 : 出力ロジカルマップ   *
*   LOG10010 : 出力ロジカルマップ   *
*****
*
COPY LOG0101I.          .....1.
COPY LOG02010.          .....1.
COPY LOG10010.          .....1.
*
*****
*   通信記述項   *
*****
*
COMMUNICATION SECTION.
*
*****
*   メッセージの受信   *
*****
*
CD RECV-INF
FOR INPUT
STATUS KEY IS          RE-STATUS
SYMBOLIC TERMINAL IS RE-TERMNAM
```

```

MESSAGE DATE IS      RE-DATE
MESSAGE TIME IS     RE-TIME
MAP NAME IS        RE-MAPNAM.
*
*****
* 一方送信メッセージの送信 *
*****
*
CD SEND-INF-OUT
FOR OUTPUT
STATUS KEY IS      SE-STATUS-OUT
SYMBOLIC TERMINAL IS SE-TERMNAM-OUT
MAP NAME IS       SE-MAPNAM-OUT
SWITCHING MODE IS  NORMAL.
*
*****
* 応答メッセージの送信 *
*****
*
CD SEND-INF-IO
FOR I-O
STATUS KEY IS      SE-STATUS-IO
MAP NAME IS       SE-MAPNAM-IO.

PROCEDURE DIVISION.

*
*****
*  メッセージの受信 *
*****
*
RECEIVE RECV-INF
FIRST SEGMENT
INTO LOG0101I.      .....2.

*
*****
*
*
*  データの処理 *
*
*
*****
*
*****
* 一方送信メッセージの送信 *
*****
*
MOVE 'XPNLE201' TO SE-TERMNAM-OUT.
MOVE 'MAM10' TO SE-MAPNAM-OUT.

SEND SEND-INF-OUT
FROM LOG10010      .....2.
WITH EMI.
*
*****

```

```

*      応答メッセージの送信      *
*****
*
MOVE 'MAM02 '      TO SE-MAPNAM-I0.

SEND SEND-INF-I0
FROM LOG02010      .....2.
WITH EMI.

*
*****
*      終了処理      *
*****
*
EXIT PROGRAM.

```

コーディング例中の 1.および 2.は次の意味になります。

1. XMAP3 で生成した入力・出力論理マップを COPY 文で取り込みます。
2. 一意名 1 には，XMAP3 で作成した入力・出力論理マップを指定します。

付録 H.2 提供するサンプルコーディング

ここでは，TP1/NET/XMAP3 が提供するサンプルコーディングの格納場所について言語ごとに示します。

なお，XMAP3 Developer[※]も TP1/NET/XMAP3 を使用した画面表示と帳票印刷を行うサンプルプログラムを提供しています。XMAP3 Developer が提供するサンプルプログラムの格納場所については，マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 プログラミングガイド」を参照してください。

注※

画面表示や帳票印刷に使用するマップの作成を支援するプログラム

(1) C 言語

C 言語のサンプルコーディングは，次のファイルで提供しています。

適用 OS が UNIX の場合

- /BeTRAN/examples/mcf/XMAP3/aplib/c/ap.c

(2) COBOL 言語

COBOL 言語のサンプルコーディングは，次のファイルで提供しています。

適用 OS が UNIX の場合

その 1

- /BeTRAN/examples/mcf/XMAP3/aplib/cobol/ap.cbl

その 2

- /BeTRAN/examples/mcf/XMAP3/aplib/dml/ap.cbl

適用 OS が Windows の場合

- %DCDIR%\examples\mcf\XMAP3\aplib\cobol\ap.cbl

付録I オペレータインジケータ表示メッセージ一覧

オペレータインジケータに表示されるメッセージを五十音順に示します。(O)に示す内容は、オペレータの対応です。

イニシャライズマップが定義されていません

イニシャライズマップが定義されていない環境で、MAP オペランドを省略した init コマンドが入力されました。

(O)
MAP オペランドを指定した init コマンドを入力します。またはシステム管理者に連絡してください。

【対策】

論理端末定義 (mcftalcle) の-p オプションでイニシャライズマップを定義します。

オペランドの指定が不正です

"C コマンド名 オペランド"の形式で入力されたオペランドが不正です。

(O)
コマンドに対応した正しいオペランドを指定してコマンドを入力してください。

キュー状態が閉塞中です

キュー状態が閉塞中のため、メッセージを入力できません。

(O)
キュー状態を閉塞解除したあと、再度メッセージを入力してください。

継続問い合わせ応答が強制終了しました

運用コマンドによって、継続問い合わせ応答処理が強制終了しました。

コマンドが異常終了しました

入力されたコマンドが異常終了しました。

(O)

コンソールに出力されているログメッセージを参照して対策をしてください。

コマンドが正常に終了しました。

入力されたコマンドが正常に終了しました。

システム終了中です

システム終了中のため、メッセージを入力できません。

投入されたコマンド名が不正です

"C コマンド名 …"の形式で入力されたコマンド名が不正です。

(O)

TP1/NET/XMAP3 でサポートしている正しいコマンドを入力してください。

内部処理でエラーが発生しました

入力メッセージの処理中にエラーが発生しました。

(O)

システム管理者に連絡してください。

【対策】

コンソールに出力されているログメッセージを参照して対策をします。

入力必須項目が入力されていません

マップ定義で指定された入力必須項目に入力されていない項目があります。

(O)

入力必須項目をすべて入力してください。

ハードコピーを取得できません

プリンタ使用中などでハードコピーを取得できません。原因はコンソールに出力されているログメッセージを参照してください。

(O)

原因を取り除いたあと、再度印刷キーを押してください。

【対策】

コンソールに出力されているログメッセージを参照して対策をします。

付録 J 理由コード一覧

ERREVT2 の理由コードを次の表に示します。

表 J-1 ERREVT2 の理由コード一覧

C 言語の理由コード (16 進数字)	COBOL 言語の理由コード (外部 10 進数字)	ERREVT2 の通知理由
DCMCF_NO_SERV (0010)	0010	アプリケーション名に相当する MHP のサービスがありません。
DCMCF_SCD_ERR (0020)	0020	ユーザサーバ未起動などによって MHP の起動に失敗しました。
DCMCF_QUE_BUF_ERR (0030)	0030	動的共用メモリが不足しました。
DCMCF_QUE_FIL_OVER (0031)	0031	キューファイルが満杯になりました。
DCMCF_QUE_LIMIT_OVER (0032)	0032	入力メッセージ最大格納数を超過しました。
DCMCF_QUE_IO_ERR (0033)	0033	入力キューに障害が発生しました。
DCMCF_AP_CLOSE (0040)	0040	MHP のアプリケーションが閉塞しています。
DCMCF_AP_SECURE (0041)	0041	MHP のアプリケーションがセキュア状態です。
DCMCF_SERV_CLOSE (0042)	0042	<ul style="list-style-type: none"> • MHP のサービスまたはサービスグループが閉塞しています。 • スケジュール閉塞されているサービスグループの入力キューに未処理受信メッセージが残った状態で、OpenTP1 を正常終了または計画停止 A で終了しました。
DCMCF_SERV_SECURE (0043)	0043	MHP のサービスグループがセキュア状態です。
DCMCF_ABNORMAL_END (0050)	0050	<ul style="list-style-type: none"> • MHP で呼び出す receive 関数にセグメントを渡す前に、MHP が異常終了しました。 • DBMS の障害などによって、トランザクションの開始に失敗しました。

SERREVT および VERREVT の理由コードを次の表に示します。

表 J-2 SERREVT および VERREVT の理由コード一覧

理由コード 1 (16 進数字)	理由コード 2 (16 進数字)	発生条件	発生個所
DCMXP_N_RSN1_MCF (00000001)	DCMXP_N_RSN2_ITQ (00000001)	メッセージ入力障害	LE
	DCMXP_N_RSN2_APL (00000002) ※1	アプリケーション名取得障害	LE
	DCMXP_N_RSN2_OTGET (00000003) ※1	メッセージ出力障害	LE
	DCMXP_N_RSN2_OTCMP (00000004)	メッセージ送信完了処理障害	LE
	DCMXP_N_RSN2_DCTLE (00000005)	mcftdctle による論理端末の閉塞	LE
	DCMXP_N_RSN2_DCTCN (00000006)	mcftdctcn による論理端末の閉塞	CN
	その他	上記以外の障害 (理由コード 2 は保守情報)	不定
DCMXP_N_RSN1_XPW (00000002)	XMAP3 Server エラーコード	XMAP3 Server 障害	CN, LE
DCMXP_N_RSN1_PRT (00000003)	XMAP3 Server 障害コード※2	回復できないプリンタの障害	LE
その他	不定	上記以外の障害	不定

(凡例)

LE：論理端末

CN：コネクション

注※1

マッピング障害の場合は、詳細コードにマッピングサービスの障害コードを設定します。障害コードの詳細は、「付録 K マッピングサービスの障害コードの詳細」を参照してください。

注※2

プリンタ障害発生イベント情報内の障害内容 (inf) です。

詳細については、マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」を参照してください。

付録K マッピングサービスの障害コードの詳細

マッピング処理で障害が発生した場合、TP1/NET/XMAP3は、メッセージログファイルにメッセージを出力します。その際、理由コードまたは保守コード2として、マッピングに関するリターンコードを出力します。リターンコードは8けたの16進数字です。リターンコードの形式は次のとおりです。

0xaabbcccc

(凡例)

0x：16進数字を示す記号

aa：エラー発生関数

bb：エラー状態

cccc：エラー詳細

リターンコードの先頭2けた、aaの部分にはエラーが発生した関数を示すコードが設定されます。aaの値とエラー発生関数の対応を、次の表に示します。

表 K-1 エラー発生関数

aa	エラー発生関数	ccccの意味（エラー詳細）
10	XMAP3 内部関数	マニュアル「XMAP3 Version 5 画面・帳票サポートシステム XMAP3 実行ガイド」のリターンコードと詳細コードの説明を参照してください。
11		
20		
21		
40		
60		
61		
62		
A0		
F0		
B1	getservbyname	UNIX システムコールの errno です。
B2	gethostbyname	
B3	socket	
B4	connect	
B5	send	
B6	recv	
B7	pthread_select	

aa	エラー発生関数	cccc の意味 (エラー詳細)
B8	pthread_sleep	UNIX システムコールの errno です。
B9	shmat	
BA	shmdt	
BB	shutdown	
BC	setsockopt	
BD	time	
BE	fcntl	
BF	close	
C0	マッピングサービス内部関数	「表 K-3 エラー詳細」を参照してください。
C1		
C2		
C3		
C4		
C5		
C6		
D0	XMAP3/PRINTER UTILITY 内部関数	マニュアル「画面・帳票サポートシステム/プリンタユーティリティ XMAP3/PRINTER UTILITY 解説・手引書」のページ制御エラーのリターンコードに関する説明を参照してください。
D1		
D2		
D3		
D4		
D5		
E0	TP1/Server Base 内部関数	マッピングサービス内保守情報です。このメッセージの直前に出力されているメッセージの処置を実施してください。
E1		
E2		
E3		
E8	TP1/Message Control 内部関数	マッピングサービス内保守情報です。このメッセージの直前に出力されているメッセージの処置を実施してください。
E9		

リターンコードの bb の部分にはエラー状態を示すコードが設定されます。bb の値とエラー状態の対応を、次の表に示します。

表 K-2 エラー状態

bb	エラー状態
00	正常
04	軽いエラーがあるがマッピングの結果は正しい
08	マッピングを実行できなかった
0C	マッピングサービス未起動

リターンコードの先頭 2 けたが C0~C6 の場合、リターンコードの末尾 4 けた、cccc の部分にはエラーの詳細を示すコードが設定されます。cccc の値とエラー詳細の対応を、次の表に示します。

表 K-3 エラー詳細

cccc	内容
0004	マップ名が指定されていません。または、マップ名が不正です。*1 (O) : マップ名を正しく指定してください。
	PAGEC 名が指定されていません。または PAGEC 名が不正です。*2 (O) : PAGEC 名を正しく指定してください。
0008	機種・モデルが指定されていません。または機種・モデルが不正です。 (O) : 論理端末定義 (mcftalcle -z) の device オペランドを正しく指定してください。
000C	機種・モデルが指定されていません。または機種・モデルが不正です。 (O) : 論理端末定義 (mcftalcle -z) の device オペランドを正しく指定してください。
00A5	XMAP3 共通定義 (mcftxp) の -n オプションが正しく設定されていません。 (O) : XMAP3 共通定義 (mcftxp) のマッピングサービス識別子を正しく指定して、OpenTP1 を再起動してください。
0408	論理端末定義 (mcftalcle -z) の device オペランドの機種・モデルが不正です。 (O) : 論理端末定義 (mcftalcle -z) の device オペランドの機種・モデルを正しく指定してください。
0428	サポートされていない端末システム名が指定されました。 (O) : マッピングサービス属性定義の SYSTNAM オペランドを正しく指定してください。
1418	物理マップをロードするメモリが不足しました。*1 (O) : マッピングサービス属性定義の MAPCNT, POOLSIZ オペランドを正しく指定してください。
	PAGEC モジュールをロードするメモリが不足しました。*2 (O) : マッピングサービス属性定義の PGCCNT, PGCPOLSZ オペランドを正しく指定してください。
FF01	一つのマッピングサービス識別子に対して 16 個以上の通信プロセスが接続を要求しました。 (O) : マッピングサービス識別子を追加して通信プロセスを振り分け、一つのマッピングサービス識別子に対して、通信プロセスが 15 個以内になるように定義してください。
FF03	マッピングサービス処理中にローカルメモリ不足が発生しました。 (O) : データ・セグメント領域を拡張してください。
FF04	マッピングサービス処理中に共有メモリ不足が発生しました。

CCCC	内容
FF04	(O)：マッピングサービス属性定義の MAPNAME, MAPCNT, POOLSIZ オペランドを正しく指定してください。
FF09	ページ制御機能を使用できません。 (O)：TP1/NET/XMAP3 の main 関数内のページ制御機能使用宣言を正しく指定してください。
FF0B	ページ制御機能を使用できません。 (O)：マッピングサービス属性定義内のページ制御機能に関連する設定値を正しく指定してください。
FF0F	論理端末閉塞解除中に dcmaphg コマンドが入力されました。 (O)：dcmaphg コマンドは論理端末閉塞中に入力してください。
FF10	マッピングサービス属性定義ファイル内の端末システム名称 (SYSTNAM) と OpenTP1 のプロトコルが一致しません。 (O)：マッピングサービス属性定義ファイル内の端末システム名称を正しく指定して、OpenTP1 を再起動してください。

(凡例)

(O)：オペレータの対応

注※1

aa で表されたエラー発生関数が、XMAP3 内部関数の場合

注※2

aa で表されたエラー発生関数が、XMAP3/PRINTER UTILITY 内部関数の場合

(英字)

AJ (メッセージ送信完了ジャーナル)

MCF が取得する統計用の履歴情報の一つです。メッセージ出力通番，出力先論理端末名などで構成されます。

AJ の取得タイミングは，表示印刷サービスのサービスタイプで異なります。画面表示の場合は，メッセージを XMAP3 Server にすべて渡した直後です。帳票印刷の場合は，メッセージを XMAP3 Server にすべて渡したあと，プリンタから印刷完了を受信した直後です。

GJ (メッセージ受信ジャーナル)

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

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

IJ (メッセージ入力ジャーナル)

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

IJ の取得タイミングは，クライアントから受信したメッセージを入力キューに格納する直前です。

MJ (メッセージジャーナル)

MCF が取得する統計用の履歴情報の一つです。入力論理端末名または，出力論理端末名，メッセージジャーナル種別，入力または出力メッセージ（入力メッセージ編集前のデータ，または出力メッセージ編集後のデータ）などで構成されます。

MJ の取得タイミングは，入力メッセージの場合，入力メッセージ編集 UOC を呼び出す直前です。また，出力メッセージの場合，出力メッセージ編集 UOC を呼び出した直後です。

OJ (メッセージ出力ジャーナル)

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

OJ の取得タイミングは，非同期送信関数を発行して，UAP が送信メッセージを出力キューに格納した直後です。

(カ行)

コネクション

TP1/NET/XMAP3 と表示印刷サービスの間に確立された論理的な通信路です。表示印刷サービスのサービスと 1 対 1 に対応します。

(サ行)

出力キュー

処理の結果として出力メッセージを管理する待ち行列です。

出力マッピング

UAP が論理マップにセットしたデータを、送信ストリームに変換することです。

送信完了

メッセージが送信済みであることを、TP1/NET/XMAP3 が出力キューに連絡することです。

(ナ行)

入力キュー

処理要求として入力メッセージを管理する待ち行列です。

入力マッピング

受信ストリームを UAP が使用できる形式に変換して論理マップにセットすることです。

(ハ行)

物理マップ

ディスプレイ端末、またはプリンタ端末に対する入出力時、UAP で取り扱う論理的なデータ項目を論理端末画面のどの位置へどのように表示するかなどの物理的な情報から構成される制御情報です。

物理メッセージ

XMAP3 が扱う XP データストリーム形式のメッセージです。EBCDIK/KEIS のコード体系を持ちます。

論理項目

可変項目に対応して、論理端末画面から入力される論理データや、論理端末画面に出力する論理データを格納する項目です。

論理セグメント

論理項目の集まりです。

論理端末

TP1/NET/XMAP3 と UAP の間に確立された論理的な端末です。論理端末画面、またはプリンタと 1 対 1 に対応します。

論理端末画面

表示印刷サービスがディスプレイ上に生成する一つのウィンドウです。

論理マップ

論理セグメントの集まりです。

論理メッセージ

データとして意味を持つメッセージの単位です。1 論理メッセージは 1 セグメント、または複数セグメントから構成されています。

索引

A

ACT ALTSEND 346
actcnaction [mcftxp (XMAP3 共通定義)] 294
aj [mcftalcle (論理端末定義)] 287
ALTPATH [マッピングサービス属性定義] 257
ans 42
any (任意型) 42

B

bretrycnt [mcftalccn (コネクション定義の開始)] 282
bretryint [mcftalccn (コネクション定義の開始)] 282
bretry [mcftalccn (コネクション定義の開始)] 281

C

CBLDCMCF('CONTEND ') 133
CBLDCMCF('RECEIVE ') 135
CBLDCMCF('REPLY ') 140
CBLDCMCF('RESEND ') 146
CBLDCMCF('SEND ') 152
CBLDCMCF('TACTCN ') 159
CBLDCMCF('TACTLE ') 163
CBLDCMCF('TDCTCN ') 166
CBLDCMCF('TDCTLE ') 170
CBLDCMCF('TEMPGET ') 173
CBLDCMCF('TEMPPUT ') 177
CBLDCMCF('TLSCN ') 180
CBLDCMCF('TLSLE ') 184
closebutton [mcftxp (XMAP3 共通定義)] 294
cmdsname [mcfmcomn (MCF マネージャ共通定義)] 279
COBOL 言語のプログラムインタフェース 129
COBOL 言語のプログラムインタフェースの一覧 129
COBOL-UAP 作成用プログラムインタフェース 128
COBOL-UAP 作成用プログラムインタフェースの一覧 129

cont 42
count [mcftalccn (コネクション定義の開始)] 281
C 言語のライブラリ関数 77
C 言語のライブラリ関数の一覧 78

D

DATACD [マッピングサービス属性定義] 257
dc_mcf_contend 79
dc_mcf_receive 81
dc_mcf_reply 86
dc_mcf_resend 91
dc_mcf_send 96
dc_mcf_tactcn 100
dc_mcf_tactle 104
dc_mcf_tdctcn 107
dc_mcf_tdctle 111
dc_mcf_tempget 114
dc_mcf_tempput 117
dc_mcf_tlscn 119
dc_mcf_tlsle 124
dcmaphg 263
dcmaps 265
DEACT ALTSEND 347
device [mcftalcle (論理端末定義)] 290
dfmap [mcftalcle (論理端末定義)] 287
DISABLE [継続問い合わせ応答の終了] 187
dmsgcnt [mcftalcle (論理端末定義)] 286

E

EBCDIK/KEIS 63
END [マッピングサービス属性定義] 258
ERREVT1 [MCF イベント情報の形式 (COBOL 言語)] 236
ERREVT1 [MCF イベント情報の形式 (C 言語)] 230
ERREVT1 [MCF イベントの種類] 226
ERREVT2 [MCF イベント情報の形式 (COBOL 言語)] 237

MCF 性能検証用トレース情報の取得レベル 301
MCF 性能検証用トレースの取得タイミング 389
MCF 性能検証用トレースの取得量 391
MCF 通信構成定義 269
MCF 通信プロセスでアクセスするファイルの最大数 300
MCF 定義オブジェクトの解析 304
MCF 定義オブジェクトの生成 303
MCF で使用する定義ファイル 269
MCF トレースファイルの見積もり式 307
MCF マネジャ共通定義 279
MCF マネジャ定義 269
MCF メイン関数のコーディング概要 386
MCF メイン関数の作成 351
MHP 31
minbutton [mcftalcle (論理端末定義)] 289
mmsgcnt [mcftalcle (論理端末定義)] 286
module [システムサービス情報定義] 297
msgadv [mcftalcle (論理端末定義)] 287
msgbuf [mcftalccn (コネクション定義の開始)] 281
msgkey [mcftxp (XMAP3 共通定義)] 293

N

noans 42
NULLCD [マッピングサービス属性定義] 257

O

opcolor [mcftxp (XMAP3 共通定義)] 294
OpenTP1 システムの変更に影響する定義 311
OpenTP1 マッピングサービス機能 248
OpenTP1 マッピングサービス機能の詳細 250
OpenTP1 マッピングサービス機能の定義例 259
OpenTP1 マッピングサービス機能の利用形態 249
OpenTP1 マッピングサービスに関する定義 253
opreverse [mcftxp (XMAP3 共通定義)] 294
OS ホストアドレス管理ファイル (hosts ファイル) のコーディング例 394

P

PADCHR [マッピングサービス属性定義] 257
PATHSTS [マッピングサービス属性定義] 258
pheight [mcftalcle (論理端末定義)] 290
POOLSIZ [マッピングサービス属性定義] 257
pwidth [mcftalcle (論理端末定義)] 290

Q

quegrpid [mcftalcle (論理端末定義)] 286
quekind [mcftalcle (論理端末定義)] 286

R

rcvbuf [mcftalccn (コネクション定義の開始)] 280
RECEIVE [一時記憶データの受け取り] 194
RECEIVE [メッセージの受信] 189
reply, send 呼び出しの可否 43
replychk [mcfaalcap (アプリケーション属性定義)] 276
RESEND [マッピングサービス属性定義] 258
resize [mcftalcle (論理端末定義)] 289
RETRYCNT [マッピングサービス属性定義] 258
RETRYSEC [マッピングサービス属性定義] 258
rretrycnt [mcftalcle (論理端末定義)] 288
rretryint [mcftalcle (論理端末定義)] 288
rretry [mcftalcle (論理端末定義)] 288

S

SCMPEVT [MCF イベント情報の形式 (COBOL 言語)] 242
SCMPEVT [MCF イベント情報の形式 (C 言語)] 232
SCMPEVT [MCF イベントの種類] 227
SEND [一時記憶データの更新] 204
SEND [メッセージの送信] 197
SERREVT [MCF イベント情報の形式 (COBOL 言語)] 244
SERREVT [MCF イベント情報の形式 (C 言語)] 233
SERREVT [MCF イベントの種類] 227
SERREVT および VERREVT の理由コード一覧 412

sndbuf [mcftalccn (コネクション定義の開始)]
280
SPP 31
svname [mcftalccn (コネクション定義の開始)]
282
SYSTNAM [マッピングサービス属性定義] 256

T

tempsize [mcfaalcap (アプリケーション属性定義)] 278
timeoutlog [mcftxp (XMAP3 共通定義)] 293
title [mcftalcle (論理端末定義)] 288
titlename [mcftalcle (論理端末定義)] 289
TP1/Message Control 33
TP1/NET/XMAP3 システム構成例 313
TP1/NET/XMAP3 に固有の定義の一覧 271
TP1/NET/XMAP3 の運用コマンド 317
TP1/NET/XMAP3 の運用コマンドを発行するプログラムの作成 396
TP1/NET/XMAP3 のオンラインコマンド 344
TP1/NET/XMAP3 の組み込みの流れ 350
TP1/NET/XMAP3 の実行形式プログラム名 297
TP1/NET/XMAP3 の定義の概要 269
TP1/NET/XMAP3 のプロトコル固有定義コマンドの指定順序 275
TP1/NET/XMAP3 プロトコル固有定義の設定 396
type [mcfaalcap (アプリケーション属性定義)]
277
type [mcftalccn (コネクション定義の開始)] 282
TZ [システムサービス情報定義] 297

U

UAP 異常終了通知イベント 226
unlock [mcftalcle (論理端末定義)] 291
UOC 207
UOC インタフェース用のパラメータとバッファの関係
217
UOC 作成上の注意事項 224
UOC で使用できる関数 225
UOC の異常処理 225

UOC の実行タイミング 225
UOC 名称の出力情報 389
UOC 呼び出し時の MCF 固有情報のダンプ出力情報
388

V

VCLSEVT [MCF イベント情報の形式 (COBOL 言語)] 247
VCLSEVT [MCF イベント情報の形式 (C 言語)]
236
VCLSEVT [MCF イベントの種類] 228
VERREVT [MCF イベント情報の形式 (COBOL 言語)] 246
VERREVT [MCF イベント情報の形式 (C 言語)]
235
VERREVT [MCF イベントの種類] 227
VOPNEVT [MCF イベント情報の形式 (COBOL 言語)] 247
VOPNEVT [MCF イベント情報の形式 (C 言語)]
236
VOPNEVT [MCF イベントの種類] 227

W

wheight [mcftalcle (論理端末定義)] 290
wwidth [mcftalcle (論理端末定義)] 290

X

XMAP3 28
XMAP3 Server 28
XMAP3 Server サービス名との関係 394
XMAP3 Server サービス名ファイルのコーディング例
394
XMAP3 Server サービス名ファイルの設定 396
XMAP3 Server サービス名ファイルの例 398
XMAP3 印刷サービス監視時間 298
XMAP3 共通定義 293
XMAP3 サーバ 28
XPCLOSEWAITTIME [システムサービス情報定義]
298
XPW_DAEMON_PORT_NO [システムサービス情報定義]
298

XP データストリーム形式 63
XRESPONSE TIME [システムサービス情報定義] 298

あ

アプリケーション決定の流れ 64
アプリケーション属性定義 276
アプリケーションの型 277
アプリケーションの決定 64
アプリケーション名格納領域 209
アプリケーション名の決定 209

い

一時記憶データ 58
一時記憶データの受け取り (COBOL 言語) 173
一時記憶データの受け取り (C 言語) 114
一時記憶データの受け取り (データ操作言語) 194
一時記憶データの更新 (COBOL 言語) 177
一時記憶データの更新 (C 言語) 117
一時記憶データの更新 (データ操作言語) 204
一方送信メッセージ 32
一方送信メッセージ送信時の MCF 性能検証用トレースの取得タイミング 391
イニシャライズマップ名 287
色番号 294
印刷キー処理 73
インタフェースの変更一覧 377

う

運用コマンド 316
運用コマンドを発行するプログラムの例 397

え

エラーイベント 61
エラー詳細 415
エラー状態 415
エラー発生関数 413

お

応答型 42
応答型 MHP が応答メッセージを送信 276

応答メッセージ 32
応答メッセージの送信 (COBOL 言語) 140
応答メッセージの送信 (C 言語) 86
オペレーティングケータのメッセージ表示 75
オペレーティングケータ表示メッセージ一覧 408
オンラインコマンド 343
オンラインコマンドサービス名 279
オンラインの開始形態 35
オンラインの開始と終了 35
オンラインの終了形態 35

か

概要 27
画面属性の端末状態の閉塞解除処理 44
画面属性の端末状態の閉塞処理 46
画面に対する分岐送信形態 51
監視時間 298

き

キー入力サービス 71
キーボード状態 291
キーボード状態の変更 73
機能 34
キューグループ ID 286
キュー状態 43
キュー状態の閉塞と閉塞解除 43
旧製品からの移行に関する注意事項 375

<

組み込み方法 349
クライアント PC/WS 応答時間タイムアウト発生時の障害メッセージ出力 75
グループ送信 287

け

継続問い合わせ応答型 42
継続問い合わせ応答形態 55
継続問い合わせ応答形態の障害 367
継続問い合わせ応答処理の強制終了 [運用コマンド] 331

継続問い合わせ応答の開始 56
継続問い合わせ応答の終了 60
継続問い合わせ応答の終了 (COBOL 言語) 133
継続問い合わせ応答の終了 (C 言語) 79
継続問い合わせ応答の終了 (データ操作言語) 187
継続問い合わせ応答用一時記憶データ 58
継続問い合わせ応答用一時記憶データ格納用領域サイズ 278

こ

交代用物理マップ読み込みパス名 262
コネクション 30
コネクション (論理端末) の追加 311
コネクション ID 280
コネクション確立障害時の確立再試行回数 282
コネクション確立障害時の確立再試行間隔 282
コネクション定義の開始 280
コネクション定義の終了 284
コネクションと論理端末の関係 30
コネクションに関する機能 37
コネクションの解放 40
コネクションの解放 (COBOL 言語) 166
コネクションの解放 (C 言語) 107
コネクションの解放の流れ 40
コネクションの確立 37
コネクションの確立 (COBOL 言語) 159
コネクションの確立 (C 言語) 100
コネクションの確立 [運用コマンド] 318
コネクションの確立再試行 281
コネクションの確立の流れ 37
コネクションの強制解放 [運用コマンド] 323
コネクションの状態取得 (COBOL 言語) 180
コネクションの状態取得 (C 言語) 119
コネクションの状態表示 40
コネクションの状態表示 [運用コマンド] 334
コネクション [用語解説] 418
コネクションを自動的に確立 281

さ

サービス番号 [マッピングサービス定義] 255
サービス名 298

し

次画面名の決定 208
時間帯情報 297
次起動アプリケーションの予約 57
次起動アプリケーション名 65
システムサービス共通情報定義 299
システムサービス情報定義 296
システムサービス情報定義ファイルの作成 350
システム定義 268
システムで使用するアプリケーション名 65
次メッセージ要求キー 73
次メッセージ要求キー名 293
受信バッファ [入力メッセージの編集とアプリケーション名の決定] 208
出力キュー [用語解説] 418
出力マッピング [用語解説] 418
出力メッセージの編集 215
出力メッセージ編集 UOC インタフェース 217
障害対策 357
障害通知イベント 227
障害の種類と対応処理 358
状態通知イベント 227
常駐マップ 251
常駐マップローディング用バッファ 251
初期画面の表示 [オンラインコマンド] 348

せ

接続対象の表示印刷サービスのサービスタイプ 282
接続対象の表示印刷サービス名 282

そ

送受信メッセージの割り当て先 286
送信完了確認メッセージ数 290
送信完了通知イベント 227
送信完了の情報の取得の要否 287

送信完了〔用語解説〕 418
送信障害通知イベント 227
送信バッファ〔出力メッセージの編集〕 217
送信要求発生時の処理 71
ソースファイルの互換性 375
ソケット用ファイル記述子の最大数 299
ソフトウェア構成の例 33

た

代行先論理端末 70
代行送信の開始 70
代行送信の開始〔運用コマンド〕 341
代行送信の開始〔オンラインコマンド〕 346
代行送信の終了 71
代行送信の終了〔運用コマンド〕 329
代行送信の終了〔オンラインコマンド〕 347
代行元論理端末 70
タイムゾーン 297
対話オンライン処理機能 28
対話オンライン処理の概要 28
対話オンライン処理の通信形態 30
端末状態 44
端末タイプ 285

つ

通信形態と障害の処理 364

て

定義オブジェクトファイルの作成 350
定義オブジェクトファイルの生成 354
定義できないマップ名 267
定義の指定順序 274
定義例 313
ディスク出力メッセージ最大格納数 286
データ操作言語のプログラムインタフェース 129
データ操作言語のプログラムインタフェースの一覧 130
デフォルトマップ名 287

と

問い合わせ応答形態 53
問い合わせ応答形態の障害 365
問い合わせメッセージ 32
問い合わせメッセージ受信時の MCF 性能検証用トレースの取得タイミング 390
トランザクションブランチ ID の形式 241
トレース情報が失われる経過時間の見積もり式 307
トレース情報量の見積もり式 307

に

入力キュー〔用語解説〕 418
入力マッピング〔用語解説〕 418
入力メッセージの編集 208
入力メッセージの編集とアプリケーション名の決定 207
入力メッセージ編集 UOC インタフェース 209

ね

ネットワーク構成の例 29

は

パス名の変更 262
バッファグループ定義 (mcftbuf) でコネクションごとに割り当てる資源の量 283

ひ

非応答型 42
非常駐マップ 252
非問い合わせ応答形態 50
非問い合わせ応答形態の障害 364
表示印刷サービス 28
表示印刷サービスの起動でコネクション確立をするための環境設定 396
標準用物理マップ読み込みパス名 262

ふ

複数セグメントのメッセージ関数の使用方法 62
不正アプリケーション名検出通知イベント 226
物理マッピング処理 (表示・印刷) 28

物理マップ〔用語解説〕 418
物理メッセージ 63
物理メッセージ〔用語解説〕 418
プリンタ属性の端末状態の閉塞解除処理 45
プリンタ属性の端末状態の閉塞処理 47
プリンタに対する分岐送信形態 52
プリンタの連続使用 69
プリンタ排他再試行 288
プリンタ排他再試行回数 288
プリンタ排他再試行間隔 288
プリンタ連続使用メッセージ数 291
プロトコルの種別 280
分岐送信形態 50
分岐送信形態の障害 365

へ

編集バッファ〔出力メッセージの編集〕 217
編集バッファ〔入力メッセージの編集とアプリケーション名の決定〕 208

ほ

ホスト名または IP アドレスの変更 311
ホスト名〔マッピングサービス定義〕 255

ま

マッピング 63
マッピングエラー情報 267
マッピングサービス機能のシステム構成例 259
マッピングサービス機能の定義例 260
マッピングサービスコマンド 262
マッピングサービス識別子 294
マッピングサービス識別子〔マッピングサービス定義〕 254
マッピングサービス属性定義 256
マッピングサービス属性定義ファイル名〔マッピングサービス定義〕 255
マッピングサービス定義 254
マッピングサービス定義と TP1/NET/XMAP3 固有の定義の関係 310
マッピングサービスの障害コードの詳細 413

マッピングサービス番号〔マッピングサービス定義〕 255
マッピングサービスプロセス 251
マッピングサービスプロセスの起動 251
マッピングサービスプロセスの機能 251
マッピングサービスプロセスの終了 251
マッピングサービスホスト名〔マッピングサービス定義〕 255
マッピングサービス名 279
マッピングによるメッセージの編集 63
マッピングによるメッセージ編集の流れ 63
マップ LRU 管理 252
マップ名の決定 216

み

未処理送信メッセージ廃棄通知イベント 227

め

メッセージグループ送信機能 65
メッセージグループ送信の流れ 68
メッセージ受信用バッファグループ番号 280
メッセージ制御機能 33
メッセージ送受信インタフェース 377
メッセージ送受信時の MCF 固有情報のダンプ出力情報 388
メッセージ送受信に関する機能 50
メッセージ送受信の形態 31
メッセージ送受信の例 31
メッセージ送信用バッファグループ番号 280
メッセージの再送 (COBOL 言語) 146
メッセージの再送 (C 言語) 91
メッセージの受信 (COBOL 言語) 135
メッセージの受信 (C 言語) 81
メッセージの受信 (データ操作言語) 189
メッセージの送信 (COBOL 言語) 152
メッセージの送信 (C 言語) 96
メッセージの送信 (データ操作言語) 197
メッセージの送信スケジュール 287
メッセージの代行送信 70
メッセージの分割と組み立て 62

メッセージ廃棄通知イベント 226
メッセージ編集用バッファグループ番号 281
メッセージ編集用バッファ数 281
メモリ出力メッセージ最大格納数 286

ゆ

ユーザアプリケーションプログラムの作成例 399
ユーザOWNコーディング 207, 382
ユーザOWNコーディングインタフェース 207
ユーザ任意のアプリケーション名 65

よ

読み込みパス名の変更 [マッピングサービスコマンド] 263
予約項目 250

り

理由コード一覧 411

ろ

ロード済み資源の表示 [マッピングサービスコマンド] 265
論理項目 [用語解説] 419
論理セグメント群形式 62
論理セグメント [用語解説] 419
論理端末 30
論理端末画面 30
論理端末画面視野枠の縦サイズ 290
論理端末画面視野枠の横サイズ 290
論理端末画面のアイコン名 290
論理端末画面のキーボード状態 73
論理端末画面の最小化ボタン 289
論理端末画面の最大化ボタン 289
論理端末画面のタイトルバー 288
論理端末画面のタイトル名 289
論理端末画面の縦サイズ 290
論理端末画面の閉じるボタンの有効化 48
論理端末画面の横サイズ 290
論理端末画面 [用語解説] 419
論理端末定義 285

論理端末とアプリケーションの型の関係 42
論理端末に関する機能 42
論理端末の状態取得 (COBOL 言語) 184
論理端末の状態取得 (C 言語) 124
論理端末の状態表示 48
論理端末の状態表示 [運用コマンド] 337
論理端末の端末状態の閉塞と閉塞解除 44
論理端末の閉塞 (COBOL 言語) 170
論理端末の閉塞 (C 言語) 111
論理端末の閉塞 [運用コマンド] 326
論理端末の閉塞解除 (COBOL 言語) 163
論理端末の閉塞解除 (C 言語) 104
論理端末の閉塞解除 [運用コマンド] 320
論理端末名称 285
論理端末 [用語解説] 419
論理ハードコピー 73
論理マップ [用語解説] 419
論理メッセージ [用語解説] 419