

OpenTP1 Version 7

分散トランザクション処理機能

OpenTP1 **使用の手引** Windows(R) **編**

解説・手引・文法・操作書

3000-3-D64-20

■対象製品

・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

P-2464-2284 uCosminexus TP1/Server Base 07-50

P-2464-2384 uCosminexus TP1/FS/Direct Access 07-50

P-2464-2484 uCosminexus TP1/FS/Table Access 07-50

P-2464-2554 uCosminexus TP1/Extension 1 07-50

P-2464-3164 uCosminexus TP1/Message Control 07-50

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

P-F2464-31642 uCosminexus TP1/Message Control - Extension 1 07-50

P-F2464-32645 uCosminexus TP1/NET/XMAP3 07-50

P-F2464-3264C uCosminexus TP1/NET/TCP/IP 07-50

P-F2464-3264D uCosminexus TP1/NET/High Availability 07-50

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

P-2964-2224 uCosminexus TP1/Server Base(64) 07-50

P-2964-2324 uCosminexus TP1/FS/Direct Access(64) 07-50

P-2964-2424 uCosminexus TP1/FS/Table Access(64) 07-50

P-2964-2524 uCosminexus TP1/Extension 1(64) 07-50

P-2964-3124 uCosminexus TP1/Message Control(64) 07-50

P-2964-3224 uCosminexus TP1/NET/Library(64) 07-50

P-F2964-3224C uCosminexus TP1/NET/TCP/IP(64) 07-50

P-F2964-3224D uCosminexus TP1/NET/High Availability(64) 07-50

・適用 OS：Windows Server 2008 R2，Windows Server 2012，Windows Server 2012 R2

P-2464-2944 uCosminexus TP1/High Availability 07-50

P-2964-2924 uCosminexus TP1/High Availability(64) 07-50

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

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

適用 OS の詳細については「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，HA モニタ，HiRDB，JP1，Open TP1，OSAS，uCosminexus は、株式会社 日立製作所の商標または登録商標です。

Microsoft および MS-DOS は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft および SQL Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft および Visual Studio は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

ODBC は、米国 Microsoft Corporation が提唱するデータベースアクセス機構です。

Oracle と Java は、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

UNIX は、The Open Group の米国ならびに他の国における登録商標です。

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

Windows は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Vista は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。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。

本書には、X/Open の許諾に基づき X/Open CAE Specification System Interfaces and Headers, Issue4, (C202 ISBN 1-872630-47-2) Copyright (C) July 1992, X/Open Company Limited の内容が含まれています；

なお、その一部は IEEE Std 1003.1-1990, (C) 1990 Institute of Electrical and Electronics Engineers, Inc. 及び IEEE std 1003.2/D12, (C) 1992 Institute of Electrical and Electronics Engineers, Inc. を基にしています。

事前に著作権所有者の許諾を得ずに、本書の該当部分を複製、複写及び転記することは禁じられています。

本書には、X/Open の許諾に基づき X/Open X/Open Preliminary Specification Distributed Transaction Processing : The TxRPC Specification (P305 ISBN 1-85912-000-8) Copyright (C) July 1993, X/Open Company Limited の内容が含まれています；

事前に著作権所有者の許諾を得ずに、本書の該当部分を複製、複写及び転記することは禁じられています。

本書には、Open Software Foundation, Inc. が著作権を有する内容が含まれています。

This document and the software described herein are furnished under a license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the above copyright notice. Title to and ownership of the document and software remain with OSF or its licensors.

■発行

2015 年 11 月 3000-3-D64-20

■著作権

All Rights Reserved. Copyright (C) 2010, 2015, Hitachi, Ltd.

変更内容

変更内容 (3000-3-D64-20) uCosminexus TP1/Server Base 07-50, uCosminexus TP1/Server Base(64) 07-50

追加・変更内容	変更箇所
Net Express に関する記述を削除した。	3.2, 3.3, 3.4(2)
Visual Studio 2012 以前に関する記述を削除した。	3.2.1, 3.3(4)
Visual Studio 2013, Visual Studio 2015 をサポートした。	3.2.2(2), 3.2.2(3), 3.2.2(4)
COBOL2002 開発マネージャの設定についての説明を追加した。	3.2.3(4), 3.2.3(5)
COBOL2002 02-00 以前に関する記述を削除した。	3.2.3(5)
サービス関数動的ローディング機能使用時の手順を Windows Server 2008 R2 の場合を例にした。	3.5(3)(b)
OpenTP1 ファイルシステムの構築手順で、セクタ長に関する記述を削除した。	4.5.2(1)
系切り替え機能使用時の設定に関する記述を削除し、注意事項を追加した。	4.5.2(4)
HiRDB の登録の実行例を HiRDB Version 9 の場合を例にした。	4.6.1(1)
SQL Server との連携時の注意事項から、Windows Server 2003 に関する記述を削除した。	4.6.3(6)
Windows ファイアウォールの設定手順を Windows 7 以降の場合を例にした。	4.7(2)
次のオペランドを追加した。 - システム共通定義 coredump_type prc_terminate_force - ジャーナルサービス定義 jnl_message_id_change_level - チェックポイントダンプサービス定義 cpd_message_id_change_level - 性能検証用トレース定義 prf_buff_size	5.1(4), 5.1(14), 5.1(16), 5.1(34), 5.3

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-D64-10) uCosminexus TP1/Server Base 07-06, uCosminexus TP1/Server Base(64) 07-06

追加・変更内容
TP1/Server Base 07-06 をサポートした。 これに伴い、次の機能をサポートした。 - ノード自動追加機能 - バッファ領域の共用による共用メモリの節約

追加・変更内容

Visual Studio 2012 をサポートした。

次のオペランド・定義コマンドを追加した。

- システム共通定義
 - name_service_mode
 - name_manager_node
 - name_remove_down_node
 - name_node_add_policy
- ネームサービス定義
 - name_start_watch_time
 - name_start_retry_count
 - name_start_retry_interval
 - name_start_error
 - name_sync_ready_time
 - namnlfil
- スケジュールサービス定義
 - scdbufgrp 定義コマンドの -e, -s, -p
- ユーザサービスデフォルト定義
 - scdbufgrp 定義コマンドの -s, -p
- ユーザサービス定義
 - scdbufgrp 定義コマンドの -s, -p

定義コマンド scdbufgrp の -g オプションをサポートした。

次のコマンドを追加した。

- nammstr
- namndopt
- namndrm
- namnlcre
- namnldel
- namnldsp
- dcalzprf

uCosminexus TP1/Server Base 07-05, uCosminexus TP1/Server Base(64) 07-05

追加・変更内容

TP1/Server Base 07-05 をサポートした。

これに伴い、次の機能をサポートした。

- 非常駐 UAP プロセスのリフレッシュ機能

次の関数を追加した。

- dc_trn_rm_select
 - CBLDCTRN('RMSELECT')
-

追加・変更内容

次のオペランド・定義コマンドを追加した。

- システム共通定義
pre_coredump_filter
- rap リスナーサービス定義
scs_prf_trace_level
rap_extend_function
- ユーザサービスデフォルト定義
scd_refresh_process
rap_extend_function
pre_coredump_filter
trnrmid 定義コマンドの -k
- ユーザサービス定義
scd_refresh_process
rap_extend_function
pre_coredump_filter
trnrmid 定義コマンドの -k

uCosminexus TP1/Server Base 07-04, uCosminexus TP1/Server Base(64) 07-04, uCosminexus TP1/Message Control 07-05, uCosminexus TP1/NET/Library 07-05

追加・変更内容

OpenTP1 コンソール出力機能の注意事項を追加した。

TP1/Server Base 07-04, および TP1/Message Control 07-05 をサポートした。

次の関数の UNIX 版 OpenTP1 との差異を追加した。

- dc_adm_call_command
- CBLDCADM('COMMAND')

メッセージログの通知機能のサポート有無を追加した。

マニフェストファイルが作成されない場合の注意事項を追加した。

メイクファイルの記述例を修正した。

OpenTP1 ファイルシステムを構築できるディスクを追加した。

システムサービス構成定義 mrs_conf オペランドを削除した。

次のオペランドの UNIX 版 OpenTP1 との差異を追加した。

- システム環境定義
user_command_online
- システム共通定義
ipc_conn_interval
- ユーザサービスデフォルト定義
ipc_conn_interval
- ユーザサービス定義
ipc_conn_interval

追加・変更内容

次のオペランド・定義コマンドを追加した。

- プロセスサービス定義
prc_take_over_dlpath
- トランザクションサービス定義
trn_rcv_open_close_scope
trnstring 定義コマンドの -s
- チェックポイントダンプサービス定義
jnl_cdskip_limit
jnl_cdskip_msg
- 拡張 RM 登録定義
trnlrkrm 定義コマンドの -C
trnlrkrm 定義コマンドの -B

mcfalcap コマンドのサポート有無を変更した。

DCMCFQUEBAK オペランドを追加した。

次の現象の原因と対策を追加した。

- UAP を開始できない（ユーザサーバが終了状態 0x00000080 または 0xC0000135 でダウンする）
- OpenTP1 を開始できない、または UAP を開始できない（OpenTP1 システムプロセスや UAP プロセスが、KFCA00100-E メッセージを出力し、ダウンする）

次のコマンドを追加した。

- preclpathls
- preclpath

次のコマンドの注意事項を追加した。

- dcrasget

次のコマンドの -C オプションの説明を追加した。

- trnlrkrm
- trnmkobj

次のメッセージを削除した。

- KFCA26564-E

コマンド引数の説明を追加した。

はじめに

このマニュアルは、次に示す Windows 版の uCosminexus OpenTP1 の製品（以降、Windows 版 OpenTP1 と表記します）について、UNIX 版の uCosminexus OpenTP1 の製品（以降、UNIX 版 OpenTP1 と表記します）との機能差異をまとめたものです。

- P-2464-2284 uCosminexus TP1/Server Base
- P-2464-2384 uCosminexus TP1/FS/Direct Access
- P-2464-2484 uCosminexus TP1/FS/Table Access
- P-2464-2554 uCosminexus TP1/Extension 1
- P-2464-2944 uCosminexus TP1/High Availability
- P-2464-3164 uCosminexus TP1/Message Control
- P-2464-3264 uCosminexus TP1/NET/Library
- P-F2464-31642 uCosminexus TP1/Message Control - Extension 1
- P-F2464-32645 uCosminexus TP1/NET/XMAP3
- P-F2464-3264C uCosminexus TP1/NET/TCP/IP
- P-F2464-3264D uCosminexus TP1/NET/High Availability
- P-2964-2224 uCosminexus TP1/Server Base(64)
- P-2964-2324 uCosminexus TP1/FS/Direct Access(64)
- P-2964-2424 uCosminexus TP1/FS/Table Access(64)
- P-2964-2524 uCosminexus TP1/Extension 1(64)
- P-2964-2924 uCosminexus TP1/High Availability(64)
- P-2964-3124 uCosminexus TP1/Message Control(64)
- P-2964-3224 uCosminexus TP1/NET/Library(64)
- P-F2964-3224C uCosminexus TP1/NET/TCP/IP(64)
- P-F2964-3224D uCosminexus TP1/NET/High Availability(64)

TP1/Message Queue は、米国 International Business Machines Corporation とのライセンス契約に基づき、WebSphere MQ の MQI, MQFAP, MQ クラスタの仕様をベースに実装しています。

■対象読者

Windows 版 OpenTP1 のシステム管理者、システム設計者、プログラマ、およびオペレータの方を対象としています。

Windows の基礎的な知識、オンラインシステムの基礎的な知識があることを前提としています。また、このマニュアルの記述は、次の OpenTP1 シリーズのマニュアルの知識があることを前提としていますので、あわせてお読みください。

- 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）

はじめに

- 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 メッセージ (3000-3-D56)

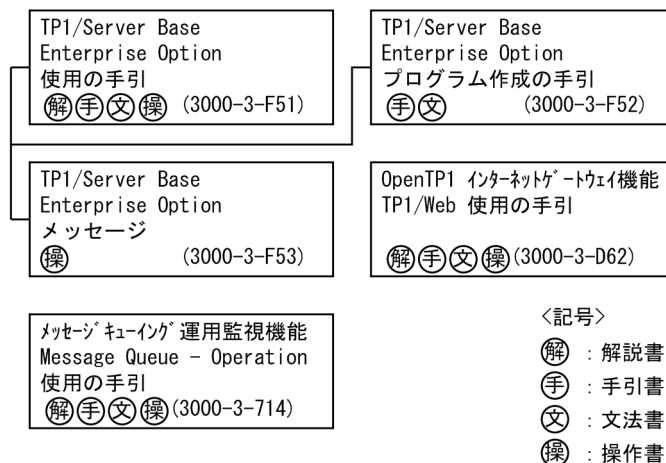
■関連マニュアル

●OpenTP1 Version 7

OpenTP1 解説 (解) (3000-3-D50)	OpenTP1 プログラム作成の手引 (手) (3000-3-D51)	OpenTP1 プログラム作成リファレンス C言語編 (手)(文) (3000-3-D54)
	OpenTP1 システム定義 (文) (3000-3-D52)	OpenTP1 プログラム作成リファレンス COBOL言語編 (手)(文) (3000-3-D55)
	OpenTP1 運用と操作 (操) (3000-3-D53)	OpenTP1 メッセージ (操) (3000-3-D56)
	OpenTP1 テスタ・UAPトレース 使用の手引 (解)(手)(文)(操) (3000-3-D57)	OpenTP1 使用の手引 Windows (R) 編 (解)(手)(文)(操) (3000-3-D64)
	OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P編 (解)(手)(文)(操) (3000-3-D58)	OpenTP1 クライアント使用の手引 TP1/Client/J編 (解)(手)(文)(操) (3000-3-D59)
	TP1/Link 使用の手引 (解)(手)(文)(操) (3000-3-D60)	TP1/Messaging 使用の手引 (解)(手)(操) (3000-3-D61)
	TP1/Extension for .NET Framework 使用の手引 (解)(手)(文)(操) (3000-3-D67)	TP1/Client for .NET Framework 使用の手引 (解)(手)(文)(操) (3000-3-D68)
	TP1/Connector for .NET Framework 使用の手引 (解)(手)(文)(操) (3000-3-D69)	OpenTP1 プロトコル TP1/NET/TCP/IP編 (解)(手)(文)(操) (3000-3-D70)
	OpenTP1 プロトコル TP1/NET/User Agent編 (解)(手)(文)(操) (3000-3-D71)	OpenTP1 プロトコル TP1/NET/HDLC編 (解)(手)(文)(操) (3000-3-D72)

OpenTP1 プロトコル TP1/NET/OSI-TP編 解手文操 (3000-3-D73)	OpenTP1 プロトコル TP1/NET/XMAP3編 解手文操 (3000-3-D74)
OpenTP1 プロトコル TP1/NET/User Datagram Protocol編 解手文操 (3000-3-D75)	OpenTP1 プロトコル TP1/NET/X25編 解手文操 (3000-3-D76)
OpenTP1 プロトコル TP1/NET/HSC編 解手文操 (3000-3-D77)	OpenTP1 プロトコル TP1/NET/NCSB編 解手文操 (3000-3-D78)
OpenTP1 プロトコル TP1/NET/OSAS-NIF編 解手文操 (3000-3-D79)	OpenTP1 プロトコル TP1/NET/Secondary Logical Unit - TypeP2編 解手文操 (3000-3-D80)
OpenTP1 プロトコル TP1/NET/X25-Extended編 解手文操 (3000-3-D82)	OpenTP1 メッセージキューイング機能 TP1/Message Queue 使用の手引 解手文操 (3000-3-D90)
OpenTP1 メッセージキューイング機能 TP1/Message Queue メッセージ 操 (3000-3-D91)	OpenTP1 メッセージキューイング機能 TP1/Message Queue プログラム作成の手引 手 (3000-3-D92)
OpenTP1 メッセージキューイング機能 TP1/Message Queue プログラム作成リファレンス 文 (3000-3-D93)	メッセージキューイングアクセス機能 TP1/Message Queue - Access 使用の手引 解手文操 (3000-3-D94)

●その他のOpenTP1関連



マニュアル「OpenTP1 プロトコル」の各プロトコル編については、ご使用の製品のバージョンに対応するマニュアルの発行時期をご確認ください。

●関連製品

- ・スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (UNIX(R)/Windows(R) 用) (3000-6-276)
- ・Hitachi HA Toolkit (3000-9-115)
- ・スケーラブルデータベースサーバ HiRDB Version 7 システム導入・設計ガイド (Windows(R) 用) (3020-6-272)
- ・スケーラブルデータベースサーバ HiRDB Version 7 UAP 開発ガイド (Windows(R) 用) (3020-6-276)
- ・スケーラブルデータベースサーバ HiRDB Version 8 システム導入・設計ガイド (Windows(R) 用) (3020-6-352)
- ・スケーラブルデータベースサーバ HiRDB Version 8 UAP 開発ガイド (3020-6-356)
- ・HiRDB Version 9 システム導入・設計ガイド (Windows (R) 用) (3020-6-452)
- ・HiRDB Version 9 UAP 開発ガイド (3020-6-456)

このマニュアルでは、次のマニュアルについて、名称を省略して表記しています。マニュアルの正式名称とこのマニュアルでの表記を次に示します。

正式名称	このマニュアルでの表記
OpenTP1 プログラム作成リファレンス C 言語編	OpenTP1 プログラム作成リファレンス
OpenTP1 プログラム作成リファレンス COBOL 言語編	
OpenTP1 メッセージキューイング機能 TP1/Message Queue 使用の手引	TP1/Message Queue 使用の手引
HiRDB Version 9 システム導入・設計ガイド (Windows(R) 用)	スケーラブルデータベースサーバ HiRDB システム導入・設計ガイド

正式名称	このマニュアルでの表記
HiRDB Version 9 UAP 開発ガイド	スケーラブルデータベースサーバ HiRDB UAP 開発ガイド

■文法の記号

(1) 文法記述記号

文法の記述記号について説明する記号です。

文法記述記号	意味
{ } 波括弧	この記号で囲まれている複数の項目のうちから一つを選択できることを示します。 (例) <code>abcdefg {-h -i}</code> -h, -i の二つのオプションのうち、どちらかを指定することを示します。
[] きっ甲	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) <code>abcdefg [-h] [-i]</code> <code>abcdefg</code> , <code>abcdefg -h</code> , または <code>abcdefg -i</code> と指定することを示します。
 ストローク	この記号で区切られた項目は選択できることを示します。 (例) <code>abcdefg -h iii jjj</code> -h オプションに <code>iii</code> , または <code>jjj</code> のどちらかを指定することを示します。
<u> </u> 下線	この記号で示す項目は、該当オペランドまたはコマンド引数を省略した場合の仮定値を示します。 (例) <code>abcdefg [-h {<u>iii</u> <u>jjj</u>}]</code> オプションの指定を省略した場合、-h オプションの指定値に <code>iii</code> を仮定することを示します。
… 点線	この記号で示す直前の項目を繰り返し指定できることを示します。 (例) <code>abcdefg -h ユーザサーバ名 [, ユーザサーバ名] …</code> -h オプションのユーザサーバ名を繰り返し指定できることを示します。
△ 白三角	半角スペースを示します。 (例) 論理ファイル名△物理ファイル名 論理ファイル名と物理ファイル名の間に一つ空白を空けることを示します。

(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。
《 》	ユーザ指定値の省略値を示します。
< >	ユーザ指定値の構文要素記号を示します。
(())	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), および _ (アンダスコア) の文字
英字記号	アルファベット (A～Z, a～z), #, @, および ¥
英数字	英字と数字 (0～9)
英数字記号	英字記号と数字 (0～9)
特殊文字	*
符号なし整数	数字 (0～9)
16 進数	数字 (0～9), A～F, および a～f (ただし, 数字 (0～9), a～f と記述している場合は, A～F は含まない 16 進数)
識別子	先頭がアルファベット (A～Z, a～z) で始まる英数字列
記号名称	先頭が英字記号で始まる英数字記号列
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)
OpenTP1 ファイル名	アルファベット (A～Z, a～z), 数字 (0～9), . (ピリオド), _ (アンダスコア), および @ で構成される文字列 (最大 14 文字)

■このマニュアルでの表記

(1) 製品名

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

製品名称	略称		
COBOL2002 Net Client Suite	COBOL2002		
COBOL2002 Net Client Runtime			
COBOL2002 Net Developer			
COBOL2002 Net Server Suite			
COBOL2002 Net Server Runtime			
HiRDB Server Version 9	HiRDB		
Java™	Java		
JP1/Automatic Job Management System 3 - Agent	JP1/AJS - Agent	JP1/AJS	JP1
JP1/Automatic Job Management System 3 - Manager	JP1/AJS - Manager		
JP1/Automatic Job Management System 2 - Scenario Operation Manager	JP1/AJS2 - Scenario Operation		
JP1/Base			

製品名称	略称	
JP1/Cm2/Operations Assist Agent	JP1/Cm2	
JP1/NETM/Audit - Manager	JP1/NETM/Audit	
JP1/NETM/DM Client	JP1/NETM/DM	
JP1/NETM/DM Manager		
JP1/NETM/DM SubManager		
Microsoft ^(R) MS-DOS ^(R)	MS-DOS	
Oracle Database 12c	Oracle	
Microsoft ^(R) SQL Server 2012	SQL Server	
Microsoft ^(R) SQL Server 2014		
uCosminexus TP1/Client/J	TP1/Client/J	TP1/ Client
uCosminexus TP1/Client/P	TP1/Client/P	
uCosminexus TP1/Client/P(64)		
uCosminexus TP1/Client/W	TP1/Client/W	
uCosminexus TP1/Client/W(64)		
uCosminexus TP1/Client for .NET Framework	TP1/Client for .NET Framework	
uCosminexus TP1/Connector for .NET Framework	TP1/Connector for .NET Framework	
uCosminexus TP1/Server Base Enterprise Option	TP1/EE	
uCosminexus TP1/Server Base Enterprise Option(64)		
uCosminexus TP1/Extension 1	TP1/Extension 1	
uCosminexus TP1/Extension 1(64)		
uCosminexus TP1/FS/Direct Access	TP1/FS/Direct Access	
uCosminexus TP1/FS/Direct Access(64)		
uCosminexus TP1/FS/Table Access	TP1/FS/Table Access	
uCosminexus TP1/FS/Table Access(64)		
uCosminexus TP1/High Availability	TP1/High Availability	
uCosminexus TP1/High Availability(64)		
uCosminexus TP1/LiNK	TP1/LiNK	
uCosminexus TP1/Message Control	TP1/Message Control	
uCosminexus TP1/Message Control(64)		
uCosminexus TP1/Message Control - Extension 1	TP1/Message Control - Extension 1	
uCosminexus TP1/Message Control/Tester	TP1/Message Control/Tester	

製品名称	略称
uCosminexus TP1/Message Queue	TP1/Message Queue
uCosminexus TP1/Message Queue(64)	
uCosminexus TP1/Message Queue - Access	TP1/Message Queue - Access
uCosminexus TP1/Message Queue - Access(64)	
uCosminexus TP1/Messaging	TP1/Messaging
uCosminexus TP1/Multi	TP1/Multi
uCosminexus TP1/NET/HDLC	TP1/NET/HDLC
uCosminexus TP1/NET/High Availability	TP1/NET/High Availability
uCosminexus TP1/NET/High Availability(64)	
uCosminexus TP1/NET/HSC	TP1/NET/HSC
uCosminexus TP1/NET/Library	TP1/NET/Library
uCosminexus TP1/NET/Library(64)	
uCosminexus TP1/NET/NCBS	TP1/NET/NCBS
uCosminexus TP1/NET/OSAS-NIF	TP1/NET/OSAS-NIF
uCosminexus TP1/NET/OSI-TP	TP1/NET/OSI-TP
uCosminexus TP1/NET/Secondary Logical Unit - TypeP2	TP1/NET/Secondary Logical Unit - TypeP2
	TP1/NET/SLU - TypeP2
uCosminexus TP1/NET/TCP/IP	TP1/NET/TCP/IP
uCosminexus TP1/NET/TCP/IP(64)	
uCosminexus TP1/NET/User Datagram Protocol	TP1/NET/UDP
uCosminexus TP1/NET/User Agent	TP1/NET/User Agent
uCosminexus TP1/NET/X25	TP1/NET/X25
uCosminexus TP1/NET/X25-Extended	TP1/NET/X25-Extended
uCosminexus TP1/NET/XMAP3	TP1/NET/XMAP3
uCosminexus TP1/Offline Tester	TP1/Offline Tester
uCosminexus TP1/Online Tester	TP1/Online Tester
uCosminexus TP1/Resource Manager Monitor	TP1/Resource Manager Monitor
uCosminexus TP1/Server Base	TP1/Server Base
uCosminexus TP1/Server Base(64)	
uCosminexus TP1/Shared Table Access	TP1/Shared Table Access
uCosminexus TP1/Web	TP1/Web
UNIX ^(R)	UNIX
Microsoft ^(R) Visual Studio ^(R) Professional 2013	Visual Studio

はじめに

製品名称	略称		
Microsoft ^(R) Visual Studio ^(R) Premium 2013			
Microsoft ^(R) Visual Studio ^(R) Ultimate 2013			
Microsoft ^(R) Visual Studio ^(R) Professional 2015			
Microsoft ^(R) Visual Studio ^(R) Enterprise 2015			
Microsoft ^(R) Visual Studio ^(R) Community 2015			
Microsoft ^(R) Windows ^(R) 7 Enterprise	Windows 7	Windows 7	Windows
Microsoft ^(R) Windows ^(R) 7 Professional			
Microsoft ^(R) Windows ^(R) 7 Ultimate			
Microsoft ^(R) Windows ^(R) 7 Enterprise(x64)	Windows 7 x64 Edition		
Microsoft ^(R) Windows ^(R) 7 Professional(x64)			
Microsoft ^(R) Windows ^(R) 7 Ultimate(x64)			
Windows ^(R) 8 Enterprise	Windows 8	Windows 8	
Windows ^(R) 8 Pro			
Windows ^(R) 8 Enterprise(x64)			
Windows ^(R) 8 Pro(x64)	Windows 8 x64 Edition		
Windows ^(R) 8.1 Enterprise			
Windows ^(R) 8.1 Pro			
Windows ^(R) 8.1 Enterprise(x64)	Windows 8.1 x64 Edition		
Windows ^(R) 8.1 Pro (x64)			
Windows ^(R) 10 Pro	Windows 10	Windows 10	
Windows ^(R) 10 Enterprise			
Windows ^(R) 10 Pro (x64)	Windows 10 x64 Edition		
Windows ^(R) 10 Enterprise(x64)			
Microsoft ^(R) Windows Server ^(R) 2008 R2, Datacenter Edition	Windows Server 2008 R2	Windows Server 2008	
Microsoft ^(R) Windows Server ^(R) 2008 R2, Enterprise Edition			
Microsoft ^(R) Windows Server ^(R) 2008 R2, Standard Edition			

製品名称	略称
Microsoft ^(R) Windows Server ^(R) 2012 Datacenter	Windows Server 2012
Microsoft ^(R) Windows Server ^(R) 2012 Standard	
Microsoft ^(R) Windows Server ^(R) 2012 R2 Datacenter	Windows Server 2012 R2
Microsoft ^(R) Windows Server ^(R) 2012 R2 Standard	

(2) 環境変数の表記について

Windows 環境での環境変数の表記は「%DCDIR%」です。環境変数の値を展開するには、「\$DCDIR」ではなく、「%DCDIR%」になることに注意してください。

(3) 大文字、小文字の違いについて

OpenTP1 では文字の大文字、小文字を区別しています。コマンドのオプションや、定義ファイルに記述した文字列をコマンド引数で使用する場合には、大文字、小文字の扱いに注意してください。例えば、-a オプションと -A オプションでは異なるオプションを表します。

(4) パス名の指定について

パス名を絶対パスで指定する場合は、必ずドライブ文字を記述してください。なお、Windows 環境でのコロン (:) は、ドライブ名とフォルダ名との区切り文字です。ドライブ文字以外にコロン (:) を含んだパス名を指定しないでください。

複数のパス名の区切り文字には、セミコロン (;) を指定してください。

(5) 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。

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

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

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

■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
API	<u>A</u> pplication <u>P</u> rogramming <u>I</u> nterface
CPU	<u>C</u> entral <u>P</u> rocessing <u>U</u> nit
CUP	<u>C</u> lient <u>U</u> ser <u>P</u> rogram
DAM	<u>D</u> irect <u>A</u> ccess <u>M</u> ethod
DB	<u>D</u> atab <u>a</u> se
DBA	<u>D</u> atab <u>a</u> se <u>A</u> dministrator
DBMS	<u>D</u> atab <u>a</u> se <u>M</u> anagement <u>S</u> ystem
EOF	<u>E</u> nd of <u>F</u> ile
GUI	<u>G</u> raphical <u>U</u> ser <u>I</u> nterface
HA	<u>H</u> igh <u>A</u> vailability
I/O	<u>I</u> nput/ <u>O</u> utput
ID	<u>I</u> dentifier
IP	<u>I</u> nternet <u>P</u> rotocol
ISAM	<u>I</u> ndexed <u>S</u> equential <u>A</u> ccess <u>M</u> ethod
IST	<u>I</u> nternode <u>S</u> hared <u>T</u> able
J2EE	<u>J</u> ava <u>2</u> <u>E</u> nterprise <u>E</u> dition
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
MQI	<u>M</u> essage <u>Q</u> ueue <u>I</u> nterface
MSCS	<u>M</u> icrosoft <u>C</u> luster <u>S</u> ervice
MSDTC	<u>M</u> icrosoft <u>D</u> istributed <u>T</u> ransaction <u>C</u> oordinator
NIC	<u>N</u> etwork <u>I</u> nterface <u>C</u> ard
NTFS	<u>N</u> ew <u>T</u> echnology <u>F</u> ile <u>S</u> ystem
ODBC	<u>O</u> pen <u>D</u> ata <u>B</u> ase <u>C</u> onnectivity
OS	<u>O</u> perating <u>S</u> ystem
OSI	<u>O</u> pen <u>S</u> ystems <u>I</u> nterconnection
OSI TP	<u>O</u> pen <u>S</u> ystems <u>I</u> nterconnection <u>T</u> ransaction <u>P</u> rocessing
PRF	<u>P</u> erformance
RM	<u>R</u> esource <u>M</u> anager
RPC	<u>R</u> emote <u>P</u> rocedure <u>C</u> all
RTS	<u>R</u> eal <u>T</u> ime <u>S</u> tatic
SPP	<u>S</u> ervice <u>P</u> roviding <u>P</u> rogram
SUP	<u>S</u> ervice <u>U</u> sing <u>P</u> rogram

英略語	英字での表記
TAM	<u>T</u> able <u>A</u> ccess <u>M</u> ethod
TCP/IP	<u>T</u> ransmission <u>C</u> ontrol <u>P</u> rotocol/ <u>I</u> nternet <u>P</u> rotocol
TP	<u>T</u> ransaction <u>P</u> rocessing
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
WER	<u>W</u> indows <u>E</u> rror <u>R</u> eporting
XA	<u>E</u> xtended <u>A</u> rchitecture
XAR	<u>E</u> xtended <u>A</u> rchitecture Resource

■ KB（キロバイト）などの単位表記について

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

目次

1	このマニュアルの位置づけ	1
1.1	ほかの OpenTP1 シリーズのマニュアルとの関係	2
1.2	このマニュアルの読み方	3
2	機能解説	5
2.1	機能の一覧	6
2.2	標準出力・標準エラー出力を確認するための機能	11
2.2.1	標準出力リダイレクト機能	11
2.2.2	標準出力リダイレクトファイルのコンソール出力機能	13
2.2.3	OpenTP1 コンソール出力機能	14
2.2.4	システム定義の関係	15
2.3	特権制御機能	16
3	プログラムの作成	17
3.1	関数の一覧	18
3.2	UAP の翻訳と結合	31
3.2.1	C 言語での UAP の作成	31
3.2.2	Visual Studio のプロジェクトでの UAP の作成	33
3.2.3	COBOL2002 での UAP の作成	36
3.3	プログラム作成時の Windows 版 OpenTP1 固有の注意事項	40
3.4	サンプルプログラム	42
3.5	サービス関数動的ローディング機能使用時の UAP 共用ライブラリ	43
4	環境設定	47
4.1	環境設定の手順	48
4.2	OpenTP1 管理者の登録	50
4.3	OpenTP1 のシステム定義情報の作成	51
4.3.1	定義ファイルの作成	51
4.3.2	定義の変更	51
4.3.3	ランタイムライブラリのパスの設定	51
4.4	環境変数の設定	53

4.5	OpenTP1 ファイルシステムの構築	55
4.5.1	OpenTP1 ファイルシステム	55
4.5.2	raw I/O 機能を適用した OpenTP1 ファイルシステムの構築	58
4.5.3	サポートディスク形式	59
4.6	データベース連携時の設定	61
4.6.1	HiRDB との連携	61
4.6.2	Oracle との連携	66
4.6.3	SQL Server との連携	73
4.7	OS の設定	80
4.8	マルチ OpenTP1 の設定	82

5	システム定義	85
5.1	システム定義の一覧	86
5.2	システム定義の説明形式	133
	定義名 (定義コマンド)	133
5.3	システム定義の詳細	134
	システム環境定義	135
	システム共通定義	139
	プロセスサービス定義 (prcsvpath)	142
	ジャーナルサービス定義	143
	クライアントサービス定義	144
	rap リスナーサービス定義	145
	ユーザサービスデフォルト定義	147
	ユーザサービス定義	150
	システムサービス情報定義	154

6	運用と操作	155
6.1	Windows 版 OpenTP1 での運用と操作の概要	156
6.2	OpenTP1 の開始	157
6.2.1	Windows の [サービス] ダイアログによる OpenTP1 の開始	157
6.2.2	コマンドによる OpenTP1 の開始	158
6.2.3	GUI による OpenTP1 の開始	159
6.2.4	開始形態の決定	159
6.3	OpenTP1 の停止	161
6.3.1	Windows の [サービス] ダイアログによる OpenTP1 の停止	161

6.3.2	コマンドによる OpenTP1 の停止	161
6.3.3	GUI による OpenTP1 の停止	162
6.3.4	OpenTP1 の停止時の注意事項	162
6.4	マルチ OpenTP1 の運用	163
6.5	シナリオテンプレートを利用したシステムの運用	164
6.5.1	シナリオの登録	164
6.5.2	スケールアウトの運用	164
6.5.3	サンプルシナリオテンプレートの利用	166
6.5.4	シナリオテンプレートの詳細	169
6.6	障害対策	173
6.6.1	障害発生時の現象と原因	173
6.6.2	アプリケーション例外発生時の対策	175
6.6.3	ポート数不足への対策（TCP/IP ポート数のチューニング）	175
6.6.4	障害発生時の取得情報の参照	176
6.7	Windows 版 OpenTP1 の GUI	178
6.7.1	GUI による環境設定	178
6.7.2	GUI による OpenTP1 の起動と停止	187

7

運用コマンド	193
運用コマンドの一覧	195
コマンドの説明形式	212
UNIX 版 OpenTP1 と共通のコマンドの詳細	213
dcpplist（製品情報の表示）	214
dcrasget（保守資料の取得）	215
fills（OpenTP1 ファイルシステムの内容表示）	216
jnlcolc（ファイル回復用ジャーナルの集積）	217
jnlcopy（アンロードジャーナルファイルの複写）	218
jnledit（アンロードジャーナルファイル、または グローバルアーカイブアンロードジャーナルファイルの編集出力）	219
jnlmcst（MCF 稼働統計情報の出力）	220
jnlrput（アンロードジャーナルファイル、または グローバルアーカイブアンロードジャーナルファイルのレコード出力）	221
jnlstts（稼働統計情報の出力）	223
jnlunlfg（ジャーナル関係のファイルのアンロード）	224

rpcdump (RPC トレースの出力)	225
rpcmrg (RPC トレースのマージ)	226
trnlnkrm (リソースマネージャの登録)	227
trnmkobj (トランザクション制御用オブジェクトファイルの作成)	229
Windows 版 OpenTP1 固有のコマンドの詳細	230
dcdls (マルチ OpenTP1 のセットアップ状態の確認)	231
dcmakecon (マルチ OpenTP1 操作コンソールの作成)	233
dcsetupml (マルチ OpenTP1 のセットアップとアンセットアップ)	235
njsmkdll (トランザクション制御用 DLL の作成)	237
ntbstart (OpenTP1 の起動)	239
ntbtail (指定ファイル内容のコンソール出力)	241
tp1console (標準出力リダイレクトファイルのコンソール出力)	243

付録	245
付録 A OpenTP1 のプロセスの一覧	246

索引	251
-----------	-----

1

このマニュアルの位置づけ

このマニュアルは、マニュアルの利用目的に応じて、ほかの OpenTP1 シリーズのマニュアルと併読する必要があります。この章では、このマニュアルとほかの OpenTP1 シリーズのマニュアルとの関係、およびこのマニュアルの読み方について説明します。

1.1 ほかの OpenTP1 シリーズのマニュアルとの関係

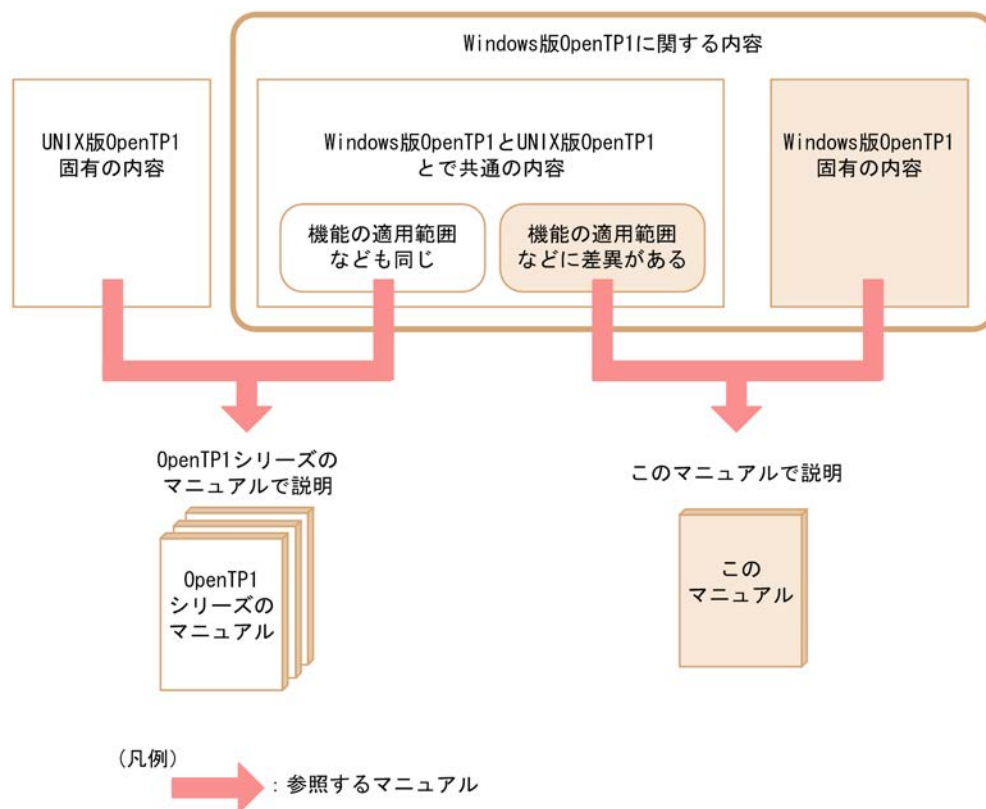
1.2 このマニュアルの読み方

1.1 ほかの OpenTP1 シリーズのマニュアルとの関係

このマニュアルでは、Windows 版 OpenTP1 の機能やプログラムの作成方法などについて、Windows 版 OpenTP1 固有の内容、および UNIX 版 OpenTP1 との差異だけを説明しています。UNIX 版 OpenTP1 と共通の内容については、このマニュアルでは説明を省略しています。このため、Windows 版 OpenTP1 を使用する場合は、このマニュアルと OpenTP1 シリーズのマニュアルを併読してください。

このマニュアルと OpenTP1 シリーズのマニュアルとの関係を次に示します。

図 1-1 このマニュアルと OpenTP1 シリーズのマニュアルとの関係



1.2 このマニュアルの読み方

Windows 版 OpenTP1 は、使用できる機能、プログラムの作成方法、環境設定方法や、指定できるシステム定義のオペランドなどが、UNIX 版 OpenTP1 と異なります。このため、このマニュアルと OpenTP1 シリーズのマニュアルは、次のように参照してください。

1. Windows 版 OpenTP1 の情報について、UNIX 版 OpenTP1 との差異があるかどうかを、このマニュアルで確認します。

このマニュアルの 2 章以降に記載されている機能や関数などの一覧を参照して、UNIX 版 OpenTP1 との差異があるかどうかを確認してください。

2. 確認した結果に応じて、このマニュアルまたは OpenTP1 シリーズのマニュアルを参照します。

- このマニュアル：Windows 版 OpenTP1 固有の内容、および UNIX 版 OpenTP1 との差異の詳細について知りたい場合に参照してください。
- OpenTP1 シリーズのマニュアル：UNIX 版 OpenTP1 と共通の内容について知りたい場合に参照してください。

Windows 版 OpenTP1 についてどんな情報を知りたいかによって、参照先のマニュアルが異なります。Windows 版 OpenTP1 に関する情報と、その情報の参照先を次に示します。

1. このマニュアルの位置づけ

図 1-2 Windows 版 OpenTP1 に関する情報とその参照先

知りたい情報	このマニュアルの参照先	参照するOpenTP1シリーズのマニュアル
OpenTP1の概要	—	「OpenTP1 解説」
機能	2章	「OpenTP1 解説」
プログラムの作成	3章	「OpenTP1 プログラム作成の手引」
関数	3章	「OpenTP1 プログラム作成リファレンス※」
環境設定	4章	「OpenTP1 運用と操作」
システム定義	5章	「OpenTP1 システム定義」
運用と操作	6章	「OpenTP1 運用と操作」
運用コマンド	7章	「OpenTP1 運用と操作」
メッセージ	—	「OpenTP1 メッセージ」

(凡例)

— : 該当しません。

注※ 次の2冊のマニュアルを示します。
「OpenTP1 プログラム作成リファレンス C言語編」
「OpenTP1 プログラム作成リファレンス COBOL言語編」

図中の参照先を基に、このマニュアルの該当する章、および OpenTP1 シリーズのマニュアルを参照してください。

例えば、OpenTP1 の機能について知りたい場合には、このマニュアルの「2. 機能解説」と、マニュアル「OpenTP1 解説」を参照します。「2.1 機能の一覧」で Windows 版 OpenTP1 と UNIX 版 OpenTP1 との差異があるかどうかを確認した上で、Windows 版 OpenTP1 固有の機能の詳細については 2 章の該当箇所を、UNIX 版 OpenTP1 と共通の内容についてはマニュアル「OpenTP1 解説」を参照してください。

また、このマニュアルでは、メッセージについて説明していません。メッセージについては、マニュアル「OpenTP1 メッセージ」を参照してください。

2

機能解説

Windows 版 OpenTP1 は，UNIX 版 OpenTP1 と比べて，使用できる OpenTP1 の機能が異なります。この章では，機能の一覧で，Windows 版 OpenTP1 と UNIX 版 OpenTP1 の機能の差異について説明します。また，Windows 版 OpenTP1 固有の機能の詳細について説明します。

UNIX 版 OpenTP1 と共通の内容については，OpenTP1 シリーズのマニュアルを参照してください。

2.1 機能の一覧

2.2 標準出力・標準エラー出力を確認するための機能

2.3 特権制御機能

2.1 機能の一覧

Windows 版 OpenTP1 は、UNIX 版 OpenTP1 と比べて、使用できる OpenTP1 の機能が異なります。Windows 版 OpenTP1 を初めて使用する場合は、ここで示す一覧表で UNIX 版 OpenTP1 との差異があるかどうかを確認してください。

この一覧表の内容を確認した上で、UNIX 版 OpenTP1 と共通の機能、または差異がある機能については、マニュアル「OpenTP1 解説」を参照してください。Windows 版 OpenTP1 固有の機能については、この章の該当箇所を参照してください。

一覧表で使用している記号の凡例を次に示します。

(凡例)

- ◎：UNIX 版 OpenTP1 と同じ機能をサポートしています。
- ：Windows 版 OpenTP1 固有の機能です。
- △：UNIX 版 OpenTP1 との機能差異や Windows 版 OpenTP1 での制限事項があります。
- ×：サポートしていません。
- －：該当しません。

Windows 版 OpenTP1 と UNIX 版 OpenTP1 の機能の差異を、次に示します。

表 2-1 Windows 版 OpenTP1 と UNIX 版 OpenTP1 の機能の差異

機能名		Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
統合システム運用管理機能 JP1		△	JP1 製品を利用したシステム運用機能のうち、次の機能はサポートしていません。 - JP1/NETM/DM による配布管理・資産管理 - JP1/Cm2 によるネットワーク管理 - JP1/Base の JP1 イベントサービス機能へのイベント登録
UAP のテスト、デバッグ機能		×	オンラインテスタ機能／オフラインテスタ機能／MCF オンラインテスタ機能は、サポートしていません。
トランザクション制御	分散トランザクション	◎	－
	グローバルトランザクション	◎	－
	トランザクションのコミットとロールバック	◎	－
	2 相コミット	◎	－

機能名		Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
	TX インタフェースによるトランザクション制御	◎	—
	XA リソースサービスによるトランザクション制御	◎	—
クライアント / サーバ形態の通信	OpenTP1 のリモート プロシジャコール通信	△	dc_rpc_call 関数の flags の DCRPC_DOMAIN をサポートしていないため、ドメイン修飾をしたサービス 要求は、サポートしていません。
	サービス情報検索の 付加機能 • グローバル検索機能 • サービス情報優先 度指定機能	◎	—
	OpenTP1 のノード管理 • 起動通知機能 • ノード監視機能 • ノード自動追加機能	◎	—
	XATMI インタフェースの通信	×	XATMI インタフェースの通信は、サポートしていません。
	TxRPC インタフェースの通信	×	TxRPC インタフェースの通信は、サポートしていません。
	メッセージ制御	△	次の機能はサポートしていません。 • タイマ起動引き継ぎ決定 UOC • MCF オンラインテスタ • SPP の履歴情報 OJ 取得要否の指定 • 入力キューにメモリキュー使用時の MCF 開始処理高速化 • MCF オンラインコマンドサービス
アプリケーション プログラムの スケジュール	SPP のスケジュール	◎	—
	MHP のスケジュール	◎	—

2. 機能解説

機能名		Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
	プロセスの制御 - マルチサーバ負荷 バランス機能 - スケジュールの優 先度 - 非常駐 UAP プロ セスのリフレッ シュ機能 - ノード間負荷バラ ンス機能 - マルチスケジュー ラ機能	◎	—
	バッファ領域の共用 による共用メモリの 節約	◎	—
OpenTP1 クライ アント機能 (TP1/Client)		◎	—
OSI TP を使った クライアント / サーバ形態の通 信		×	OSI TP を使ったクライアント / サーバ 形態の通信では、XATMI インタフェー スの関数を使用します。このため、 XATMI インタフェースをサポートして いない Windows 版 OpenTP1 では、こ の機能はサポートしていません。
リモート API 機 能		◎	—
サービス関数動 的ローディング 機能		◎	—
OpenTP1 の運用 を補助する機能	資源の排他制御	◎	—
	ユーザジャーナルの 取得	◎	—
	ジャーナル維持機能	△	コマンドの仕様が異なります。
	メッセージログの操 作	△	logcat コマンドをサポートしているた め、標準出力はできます。ただし、 syslog への出力は、サポートしていま せん。
	メッセージログの通 知	◎	—
	OpenTP1 提供以外の リソースマネージャの 制御 (TP1/Resource Manager Monitor)	×	リソースマネージャの制御は、サポート していません。

機能名		Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
	稼働統計情報	△	稼働統計情報は、コマンドの仕様が異なります。 また、MCF 稼働統計情報に関する機能（稼働統計情報取得機能、稼働統計情報出力機能、および稼働統計情報編集機能）は、サポートしていません（取得そのものを行っていません）。
	リアルタイム統計情報サービス	◎	—
シナリオテンプレートを利用したシステムの運用		△	シナリオテンプレートの利用方法が異なります。
監査ログによるシステムの監視		◎	—
OpenTP1 ファイルシステム		△	OpenTP1 ファイルシステムのアクセス権に関する操作（OpenTP1 ファイルの保護）は、サポートしていません。
複数の OpenTP1 を使用する場合の機能	系切り替え機能	△	Windows 版 OpenTP1 では、HA モニタではなく Hitachi HA Toolkit Extension が必要です。Hitachi HA Toolkit Extension については、マニュアル「Hitachi HA Toolkit」を参照してください。 また、次の差異があります。 - システム構成 - システム環境定義の mode_conf オペランドで AUTO しか指定できない - dcstart コマンドをサポートしていない
	マルチノード機能 (TP1/Multi)	×	マルチノード機能は、サポートしていません。
	マルチ OpenTP1	△	運用方法（運用コマンド）が異なります。
OpenTP1 のプロセス構造		△	Windows 版 OpenTP1 固有のプロセスがある、プロセスの稼働数が異なるなどの差異があります。詳細については、「付録 A OpenTP1 のプロセスの一覧」を参照してください。
Windows 版 OpenTP1 固有の機能	標準出力リダイレクト機能	○	Windows 版 OpenTP1 固有の機能です。詳細については、「2.2.1 標準出力リダイレクト機能」を参照してください。

2. 機能解説

機能名		Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
	標準出力リダイレクトファイルのコンソール出力機能	○	Windows 版 OpenTP1 固有の機能です。詳細については、「2.2.2 標準出力リダイレクトファイルのコンソール出力機能」を参照してください。
	OpenTP1 コンソール出力機能	○	Windows 版 OpenTP1 固有の機能です。詳細については、「2.2.3 OpenTP1 コンソール出力機能」を参照してください。
	特権制御機能	○	Windows 版 OpenTP1 固有の機能です。詳細については、「2.3 特権制御機能」を参照してください。

注

UNIX 版 OpenTP1 と共通の機能、または差異がある機能については、この表の機能名を基に、マニュアル「OpenTP1 解説」を参照してください。

2.2 標準出力・標準エラー出力を確認するための機能

Windows 版 OpenTP1 では、システムサーバおよびユーザサーバが標準出力・標準エラー出力に出力した内容（KFCA-xxxxx メッセージや、ユーザが標準出力に指定した内容すべて）を、ファイルやコンソールに出力して確認できます。

標準出力・標準エラー出力を確認するために提供している機能を次に示します。

● 標準出力リダイレクト機能

標準出力・標準エラー出力の内容を、ファイルに出力する機能です。標準出力・標準エラー出力の内容をファイルに出力できるため、履歴を残せます。出力されたファイルの内容を見たいときは、毎回ファイルを開く必要があります。

● 標準出力リダイレクトファイルのコンソール出力機能

標準出力リダイレクト機能で出力したファイルの内容を、コンソールでリアルタイムに確認できる機能です。

● OpenTP1 コンソール出力機能

標準出力・標準エラー出力の内容を、直接コンソールに出力する機能です。出力内容は、ファイルに保存されません。

標準出力リダイレクト機能または OpenTP1 コンソール出力機能のどちらも使用しない場合、UAP から `printf()` 関数などのストリームを使用した出力内容はどこにも表示されません。ストリームを使用した出力内容を確認したい場合は、標準出力リダイレクト機能または OpenTP1 コンソール出力機能のどちらかを使用してください。

以降、それぞれの機能について説明します。

2.2.1 標準出力リダイレクト機能

Windows 版 OpenTP1 では、標準出力・標準エラー出力の内容をファイルに直接出力できます。この機能を、**標準出力リダイレクト機能**といいます。この機能を使用すれば、システムサーバおよびユーザサーバが標準出力・標準エラー出力する内容（KFCA-xxxxx メッセージや、ユーザが標準出力に指定した内容すべて）を、ファイルに出力できます。出力されたファイルの内容を見たいときは、毎回ファイルを開く必要があります。

ここでは、標準出力リダイレクト機能を使用するための環境設定、および注意事項を説明します。

参考

標準出力リダイレクトファイルのコンソール出力機能を使用すれば、出力内容をコンソールでリアルタイムに確認できます。標準出力リダイレクトファイルのコンソール出力機能については、「2.2.2 標準出力リダイレクトファイルのコンソール出力機能」を参照してください。

(1) 環境設定

標準出力リダイレクト機能を使用するためには、次のシステム定義を指定します。

システム環境定義

- `redirect_file` オペランド
標準出力・標準エラー出力の内容をファイルに出力するかどうかを指定します。
- `redirect_file_name` オペランド
標準出力・標準エラー出力の内容を出力するファイル名を指定します。
- `redirect_file_size` オペランド
標準出力・標準エラー出力の内容を出力するファイルの最大サイズを指定します。

それぞれのオペランドの詳細については、「5.3 システム定義の詳細」の「システム環境定義」を参照してください。

なお、この機能は、システム共通定義の `prc_port` オペランド、および `rpc_port_base` オペランドを指定している場合は使用できません。これらのオペランドを指定し、かつ `redirect_file` オペランドに Y を指定した場合、KFCA26520-W メッセージを出力し、`redirect_file` オペランドの指定を無視して OpenTP1 の起動を続行します。KFCA26520-W メッセージが出力された場合は、`prc_port` オペランドおよび `rpc_port_base` オペランド、または `redirect_file` オペランドのどちらかを削除してください。

(2) 注意事項

- この機能の初期化処理に失敗した場合、イベントビューアに KFCA00715-E メッセージが出力され、要因コードとして「200」が出力されます。このメッセージが出力された場合は、OpenTP1 サービスを停止し、OpenTP1 を再起動してください。
- 高負荷での書き込みが多発すると、メッセージが破棄されることがあります。
- OpenTP1 配下のプロセスの標準出力・標準エラー出力メッセージは、`prctee` プロセスを経由して指定されたファイルに出力されます。そのため、オンライン中に `prctee` プロセスがダウンした場合、OpenTP1 配下のプロセスの標準出力・標準エラー出力メッセージは、ファイルに出力されなくなります。この場合は、OpenTP1 サービスを停止し、OpenTP1 を再起動してください。
- `redirect_file_name` オペランドを指定していない場合、またはフォルダが存在しないなどの理由で `redirect_file_name` オペランドに指定したファイルのオープンに失敗し

た場合、OpenTP1 配下のプロセスの標準出力・標準エラー出力先ファイルとして、次のファイルが作成されます。

- %DCDIR%\¥spool¥prclog1
- %DCDIR%\¥spool¥prclog2

- この機能を使用すると、システム環境定義の `user_command_online` オペランドに指定するバッチファイル、および実行形式ファイル内で、標準出力・標準エラー出力にメッセージを出力できません。標準出力・標準エラー出力を使用するバッチファイル、および実行形式ファイルを使用した場合、KFCA01819-E メッセージの出力後に、OpenTP1 がダウンします。標準出力・標準エラー出力を使用するプログラムを指定する場合は、バッチファイル内で `@echo off` を実行し、標準出力・標準エラー出力を無効にしてから使用してください。
- UAP から `printf()` 関数などストリームを使用した出力を行う場合、出力後にストリームをフラッシュしてください。これらの対処をしないと、出力内容がファイルに反映されないことがあります。なお、COBOL-UAP の `DISPLAY` 文についてはリアルタイム出力されるため、ストリームを使用した出力でも影響はありません。

2.2.2 標準出力リダイレクトファイルのコンソール出力機能

標準出力リダイレクト機能で作成されたファイルの内容を確認するためには、毎回ファイルを開く必要があります。このような場合、**標準出力リダイレクトファイルのコンソール出力機能**を使用すれば、標準出力リダイレクト機能を使用して出力したファイルの内容を、コンソールでリアルタイムに確認できます。標準出力リダイレクト機能とこの機能を併用すれば、出力内容をファイルに保存し、かつリアルタイムで出力内容を確認できるようになります。

この機能は、標準出力リダイレクト機能を使用していることが前提です。標準出力リダイレクト機能の環境設定については、「2.2.1(1) 環境設定」を参照してください。

コンソールは、コマンドプロンプトまたは Windows のスタートメニューから起動します。起動方法をそれぞれ次に示します。

- コマンドプロンプトから起動
 コマンドプロンプトを起動したあと、`tp1console` コマンドまたは `ntbtail -t p` コマンドを実行して起動します。
 それぞれのコマンドについては、「7. 運用コマンド」の「`tp1console`（標準出力リダイレクトファイルのコンソール出力）」または「`ntbtail`（指定ファイル内容のコンソール出力）」を参照してください。
 なお、マルチ OpenTP1 環境の場合は、`dcmakecon` コマンドで作成したコンソールから、これらのコマンドを実行してください。`dcmakecon` コマンドについては、「7. 運用コマンド」の「`dcmakecon`（マルチ OpenTP1 操作コンソールの作成）」を参照してください。
- Windows のスタートメニューから起動

オリジナルの OpenTP1（インストーラでインストールした OpenTP1）の場合は、Windows のスタートメニューから起動できます。[スタート] — [プログラム] — [OpenTP1] から「OpenTP1 コンソール」を選択します。

2.2.3 OpenTP1 コンソール出力機能

システムサーバ、ユーザサーバが標準出力・標準エラー出力の内容をコンソールに出力できます。これを、OpenTP1 **コンソール出力機能**といいます。コンソールに出力できる内容は、標準出力リダイレクト機能の出力ファイルの内容と同じです。標準出力リダイレクト機能とは異なり、この機能では、標準出力・標準エラー出力の内容を、ファイルではなくコンソールに直接出力します。このため、出力内容は残りません。

なお、OpenTP1 サービスをユーザアカウントで起動する場合、またはセッション 0 の分離機能をサポートしている Windows（Windows Server 2008, Windows Vista, Windows 7 など）を使用する場合、この機能は使用できません。また、セッション 0 の分離機能をサポートしている Windows を使用する場合、この機能に関する環境設定をしてもエラーメッセージは出力されませんので注意してください。

ここでは、OpenTP1 コンソール出力機能を使用するための環境設定、および注意事項を説明します。

(1) 環境設定

OpenTP1 コンソール出力機能を使用するためには、次のシステム定義を指定します。

システム環境定義

- console_output オペランド

標準出力・標準エラー出力の内容をコンソールに出力するかどうかを指定します。

オペランドの詳細については、「5.3 システム定義の詳細」の「システム環境定義」を参照してください。

なお、OpenTP1 コンソール出力機能は、システム共通定義の prc_port オペランド、および rpc_port_base オペランドを指定している場合でも使用できます。

(2) 注意事項

- 標準出力・標準エラー出力を出力するコンソールは、絶対に閉じないでください。閉じた場合、OpenTP1 がシステムダウンしてしまいます。このとき、ユーザサーバやシステムサーバの一部は、KFCA01820-E メッセージを出力し、終了状態 0xC000013A で異常終了することがあります。
- 標準出力・標準エラー出力を出力するコンソール上では、テキストの選択や、マウスのクリックなどの操作をしないでください。操作した場合、出力内容の表示が止まり、メッセージ出力が遅延したり、OpenTP1 がシステムダウンしたりするおそれがあります。

次に示す方法で選択状態にできないように設定してください。

1. OpenTP1 コンソールに対して、[Alt] キー＋スペースキーで「プロパティ」を選択します。
2. [オプション] タブを選択します。
3. 「編集オプション」の「簡易編集モード」のチェックを外します。
4. [OK] ボタンをクリックします。
5. [プロパティの適用] ダイアログボックスが表示された場合は、「現在のウィンドウだけに適用する」を選択し、[OK] ボタンをクリックします。

- システム環境定義の定義解析でエラーとなった場合、標準出力・標準エラー出力はイベントビューアに出力されます。

2.2.4 システム定義の関係

標準出力・標準エラー出力を確認するための機能を使用する場合に指定するオペランドの対応を、次に示します。

表 2-2 標準出力・標準エラー出力を確認するための機能を使用する場合に指定するオペランドの対応

console_output オペランドの指定値	prc_port オペランド または rpc_port_base オペランドの指定	redirect_file オペランドの指定値	
		Y	N
Y	○	OpenTP1 コンソール出力機能が有効になります。	OpenTP1 コンソール出力機能が有効になります。
	×	標準出力リダイレクト機能が有効になります。	
N	○	出力しません。	出力しません。
	×	標準出力リダイレクト機能が有効になります。	

(凡例)

- ：指定します。
 - ×
- ×：指定しません。

2.3 特権制御機能

Windows では、デフォルトで多くの特権が Administrators グループの権限を持つユーザに与えられています。しかし、必要以上の特権が与えられていると、ユーザサーバのセキュリティが低くなるおそれがあります。

Windows 版 OpenTP1 では、ユーザサーバが持つ Windows の特権を制御できます。この機能を、**特権制御機能**といいます。この機能を使用することで、ユーザサーバのセキュリティを高めることができます。

Windows の特権を必要とする操作をユーザサーバが行っている場合は、個別に特権を設定することもできます。例えば、ユーザサーバ内で `dc_adm_call_command` 関数を使用する場合、実行するプロセスが特権を必要とする動作を行うためには、そのユーザサーバに対して個別に特権を与える必要があります。

特権制御機能を使用する場合に必要な環境設定について次に説明します。

(1) 環境設定

特権制御機能を使用するためには、次のシステム定義を指定します。

ユーザサービス定義

- `process_privilege_restrict` オペランド
ユーザサーバに対して Windows の特権を制限するかどうかを指定します。
- `process_privilege_name` オペランド
ユーザサーバに付与する Windows の特権名を指定します。

それぞれのオペランドの詳細については、「5.3 システム定義の詳細」の「ユーザサービス定義」を参照してください。

3

プログラムの作成

Windows 版 OpenTP1 は，UNIX 版 OpenTP1 と比べて，使用できる関数，およびプログラムの作成方法が異なります。この章では，関数の一覧で，Windows 版 OpenTP1 と UNIX 版 OpenTP1 の関数の差異について説明します。また，Windows 版 OpenTP1 固有のプログラムの作成方法について説明します。

UNIX 版 OpenTP1 と共通の内容については，OpenTP1 シリーズのマニュアルを参照してください。

3.1 関数の一覧

3.2 UAP の翻訳と結合

3.3 プログラム作成時の Windows 版 OpenTP1 固有の注意事項

3.4 サンプルプログラム

3.5 サービス関数動的ローディング機能使用時の UAP 共用ライブラリ

3.1 関数の一覧

Windows 版 OpenTP1 は、UNIX 版 OpenTP1 と比べて、使用できる OpenTP1 の関数が異なります。また、一部の関数では、引数や設定値をサポートしていないなどの制限事項があります。Windows 版 OpenTP1 を初めて使用する場合は、ここで示す一覧表で UNIX 版 OpenTP1 との差異があるかどうかを確認してください。

この一覧表の内容を確認した上で、関数の詳細については、マニュアル「OpenTP1 プログラム作成リファレンス」の該当する言語編を参照してください。

！ 注意事項

この一覧表では、メッセージキューイング機能（TP1/Message Queue）の関数について説明していません。マニュアル「TP1/Message Queue 使用の手引」を参照してください。

一覧表で使用している記号の凡例を次に示します。

（凡例）

- ◎：UNIX 版 OpenTP1 と同じ機能をサポートしています。
- △：UNIX 版 OpenTP1 との機能差異や Windows 版 OpenTP1 での制限事項があります。
- ×：サポートしていません。
- －：該当しません。

Windows 版 OpenTP1 と UNIX 版 OpenTP1 の関数の差異を、次に示します。

表 3-1 Windows 版 OpenTP1 と UNIX 版 OpenTP1 の関数の差異

機能		関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
リモート プロシ ャ コール	アプリケーション プログラムの開始	dc_rpc_open CBLDCRPC('OPEN')	◎	－
	SPP のサービス開 始	dc_rpc_mainloop CBLDCRSV('MAINLOOP')	◎	－
	遠隔サービスの要 求	dc_rpc_call CBLDCRPC('CALL')	△	flags の DCRPC_DOMAIN の設定はサポート していません。
	通信先を指定した 遠隔サービスの呼 び出し	dc_rpc_call_to －	◎	－

機能		関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
	処理結果の非同期受信	dc_rpc_poll_any_replies CBLDCRPC('POLLANYR')	◎	—
	エラーが発生した非同期 RPC 要求の記述子の取得	dc_rpc_get_error_descriptor CBLDCRPC('GETERDES')	◎	—
	処理結果の受信の拒否	dc_rpc_discard_further_replies CBLDCRPC('DISCARDF')	◎	—
	非同期応答型 RPC の特定応答電文廃棄	dc_rpc_discard_specific_reply CBLDCRPC('DISCARDS')	◎	—
	サービス関数のリトライ	dc_rpc_service_retry CBLDCRPC('SVRETRY')	◎	—
	サービス要求のスケジュールプライオリティの設定	dc_rpc_set_service_priority CBLDCRPC('SETSVPRI')	◎	—
	サービス要求のスケジュールプライオリティの参照	dc_rpc_get_service_priority CBLDCRPC('GETSVPRI')	◎	—
	サービスの応答待ち時間の参照	dc_rpc_get_watch_time CBLDCRPC('GETWATCH')	◎	—
	サービスの応答待ち時間の更新	dc_rpc_set_watch_time CBLDCRPC('SETWATCH')	◎	—
	クライアント UAP のノードアドレスの取得	dc_rpc_get_callers_address CBLDCRPC('GETCLADR')	◎	—
	ゲートウェイのノードアドレスの取得	dc_rpc_get_gateway_address CBLDCRPC('GETGWADR')	◎	—
	CUP への一方通知	dc_rpc_cltsend CBLDCRPC('CLTSEND')	◎	—
	アプリケーションプログラムの終了	dc_rpc_close CBLDCRPC('CLOSE')	◎	—
リモート API 機能	rap リスナーとのコネクション確立	dc_rap_connect CBLDCRAP('CONNECT') CBLDCRAP('CONNECTX')	◎	—
	rap リスナーとのコネクション解放	dc_rap_disconnect CBLDCRAP('DISCNCT')	◎	—

3. プログラムの作成

機能		関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
トランザクション制御	トランザクションの開始	dc_trn_begin CBLDCTRN('BEGIN ')	◎	—
	連鎖モードのコミット	dc_trn_chained_commit CBLDCTRN('C-COMMIT')	◎	—
	連鎖モードのロールバック	dc_trn_chained_rollback CBLDCTRN('C-ROLL ')	◎	—
	非連鎖モードのコミット	dc_trn_unchained_commit CBLDCTRN('U-COMMIT')	◎	—
	非連鎖モードのロールバック	dc_trn_unchained_rollback CBLDCTRN('U-ROLL ')	◎	—
	現在のトランザクションに関する情報の報告	dc_trn_info CBLDCTRN('INFO ')	◎	—
	リソースマネージャ接続先選択	dc_trn_rm_select CBLDCTRN('RMSELECT')	◎	—

機能		関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
システム運用 の管理	運用コマンドの実 行	dc_adm_call_command CBLDCADM('COMMAND ')	△	• flags の DCADM_DELAY の設定はサポートしていません。 • 指定した文字列中に '!', または '\$' を含む場合、指定した文字列に対して OpenTP1 内部で独自の文字列変換処理を実行します。OpenTP1 が行う文字列変換処理を無効にして、指定したとおりにコマンドを実行したい場合は、ユーザサービス定義、またはユーザサービスデフォルト定義に次の環境変数を設定してください。 putenv DCADMCONV SEP 1 • 文字列の末尾に '&' を指定した場合のコマンドの非同期実行は、サポートしていません。

3. プログラムの作成

機能		関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
				• dc_adm_call_com mand 関数の第 1 引数 com に指 定できる文字列 の長さは 500 バ イト以下です。 501 バイト以上 指定すると dc_adm_call_co mmand 関数は DCADMER_PA RAM をリター ンします。 • CBLDCADM(CO MMAND) の データ名 G に指 定できる文字列 の長さは 500 バ イト以下です。 501 バイト以上 指定すると CBLDCADM(C OMMAND) は ステータスコー ドに 01802 を設 定します。 • DML インタ フェースの SEND (運用コ マンドの実行) の一意名 1 の データ名 4 に指 定できる文字列 の長さは 500 バ イト以下です。 501 バイト以上 指定すると SEND はステー タスコードに 01802 を設定し ます。
	ユーザサーバの開 始処理完了の報告	dc_adm_complete CBLDCADM('COMPLETE')	◎	—
	ユーザサーバの状 態の報告	dc_adm_status CBLDCADM('STATUS ')	◎	—
監査ロ グの出 力	監査ログの出力	dc_log_audit_print CBLDCADT('PRINT ')	◎	—

機能		関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
メッセージ ログの 出力	メッセージログの 出力	dc_logprint CBLDCLOG('PRINT ')	△	color の設定はサ ポートしていません。
メッ セー ジ ログの 通知	メッセージログ通 知の受信の終了	dc_log_notify_close	◎	—
	メッセージログ通 知の受信の開始	dc_log_notify_open	◎	—
	メッセージログ通 知の受信	dc_log_notify_receiv e	◎	—
	ユーザ作成メッ セージログの送信	dc_log_notify_send	◎	—
ユーザ ジャー ナルの 取得	ユーザジャーナル の取得	dc_jnl_ujput CBLDCJNL('UJPUT ')	◎	—
ジャー ナル デー タの 編集	jnlrput 出力ファイ ルのクローズ	— CBLDCJUP('CLOSERPT')	◎	—
	jnlrput 出力ファイ ルのオープン	— CBLDCJUP('OPENRPT ')	◎	—
	jnlrput 出力ファイ ルからジャーナル データの入力	— CBLDCJUP('RDGETRPT')	◎	—
メッ セー ジ 送受信	MCF 環境のオープ ン	dc_mcf_open CBLDCMCF('OPEN ')	◎	—
	MHP のサービス 開始	dc_mcf_mainloop CBLDCMCF('MAINLOOP')	◎	—
	メッセージの受信	dc_mcf_receive CBLDCMCF('RECEIVE ')	◎	—
	応答メッセージの 送信	dc_mcf_reply CBLDCMCF('REPLY ')	◎	—
	メッセージの送信	dc_mcf_send CBLDCMCF('SEND ')	◎	—
	メッセージの再送	dc_mcf_resend CBLDCMCF('RESEND ')	◎	—
	同期型のメッセー ジ受信	dc_mcf_recvsync CBLDCMCF('RECVSYNC')	◎	—

3. プログラムの作成

機能	関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
同期型のメッセージ送信	dc_mcf_sendsync CBLDCMCF('SENDSYNC')	◎	—
同期型のメッセージ送受信	dc_mcf_sendrecv CBLDCMCF('SENDRECV')	◎	—
一時記憶データの受け取り	dc_mcf_tempget CBLDCMCF('TEMPGET')	◎	—
一時記憶データの更新	dc_mcf_tempput CBLDCMCF('TEMPPUT')	◎	—
継続問い合わせ応答の終了	dc_mcf_contend CBLDCMCF('CONTEND')	◎	—
アプリケーションプログラムの起動	dc_mcf_execap CBLDCMCF('EXECAP')	◎	—
アプリケーション情報通知	dc_mcf_ap_info CBLDCMCF('APINFO')	◎	—
UOC へのアプリケーション情報通知	dc_mcf_ap_info_uoc —	◎	—
ユーザタイム監視の設定	dc_mcf_timer_set CBLDCMCF('TIMERSET')	◎	—
ユーザタイム監視の取り消し	dc_mcf_timer_cancel CBLDCMCF('TIMERCAN')	◎	—
MHP のコミット	dc_mcf_commit CBLDCMCF('COMMIT')	◎	—
MHP のロールバック	dc_mcf_rollback CBLDCMCF('ROLLBACK')	◎	—
MCF 環境のクローズ	dc_mcf_close CBLDCMCF('CLOSE')	◎	—
MCF 通信サービスの状態取得	dc_mcf_tlscom CBLDCMCF('TLSCOM')	◎	—
コネクションの状態取得	dc_mcf_tlscn CBLDCMCF('TLSCN')	◎	—
コネクションの確立	dc_mcf_tactcn CBLDCMCF('TACTCN')	◎	—
コネクションの解放	dc_mcf_tdctcn CBLDCMCF('TDCTCN')	◎	—
サーバ型コネクションの確立要求の受付状態取得	dc_mcf_tlsln CBLDCMCF('TLSLN')	◎	—

機能	関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
サーバ型コネクションの確立要求の受付開始	dc_mcf_tonln CBLDCMCF('TONLN ')	◎	—
サーバ型コネクションの確立要求の受付終了	dc_mcf_tofln CBLDCMCF('TOFLN ')	◎	—
アプリケーションに関するタイマ起動要求の削除	dc_mcf_adltap CBLDCMCF('ADLTAP ')	◎	—
論理端末の状態取得	dc_mcf_tlsle CBLDCMCF('TLSLE ')	◎	—
論理端末の閉塞	dc_mcf_tdctle CBLDCMCF('TDCTLE ')	◎	—
論理端末の閉塞解除	dc_mcf_tactle CBLDCMCF('TACTLE ')	◎	—
論理端末の出力キュー削除	dc_mcf_tdlqle CBLDCMCF('TDLQLE ')	◎	—
DAM ファイル サービス	論理ファイルのオープン	◎	—
	論理ファイルからブロックの入力	◎	—
	論理ファイルのブロックの更新	◎	—
	論理ファイルヘブブロックの出力	◎	—
	論理ファイルのクローズ	◎	—
	論理ファイルの閉塞	◎	—
	論理ファイルの閉塞の解除	◎	—
	論理ファイルの状態の参照	◎	—

3. プログラムの作成

機能	関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
回復対象外 DAM ファイル使用の開始	dc_dam_start CBLDCDAM('DCDAMSV', 'STRT')	◎	—
回復対象外 DAM ファイル使用の終了	dc_dam_end CBLDCDAM('DCDAMSV', 'END')	◎	—
物理ファイルの割り当て	dc_dam_create CBLDCDAM('DCDAMINT', 'CRAT')	◎	—
物理ファイルのオープン	dc_dam_iopen CBLDCDAM('DCDAMINT', 'OPEN')	◎	—
物理ファイルから ブロックの入力	dc_dam_get CBLDCDAM('DCDAMINT', 'GET')	◎	—
物理ファイルへブ ロックの出力	dc_dam_put CBLDCDAM('DCDAMINT', 'PUT')	◎	—
物理ファイルのブ ロックの検索	dc_dam_bseek CBLDCDAM('DCDAMINT', 'BSEK')	◎	—
物理ファイルから ブロックの直接入 力	dc_dam_dget CBLDCDAM('DCDAMINT', 'DGET')	◎	—
物理ファイルへブ ロックの直接出力	dc_dam_dput CBLDCDAM('DCDAMINT', 'DPUT')	◎	—
物理ファイルのク ローズ	dc_dam_iclose CBLDCDAM('DCDAMINT', 'CLOS')	◎	—
TAM ファイル サービス	TAM テーブルの オープン	◎	—
	TAM テーブルから レコードの入力	◎	—
	TAM テーブルのレ コード入力を前提 の更新	◎	—
	TAM テーブルのレ コードの更新 / 追 加	◎	—

機能		関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
	TAM テーブルのレコードの削除	dc_tam_delete CBLDCTAM('ERS ')('ERSR')	◎	－
	TAM テーブルのレコードの入力取り消し	dc_tam_read_cancel －	◎	－
	TAM テーブルの状態の取得	dc_tam_get_inf CBLDCTAM('GST ')	◎	－
	TAM テーブルの情報の取得	dc_tam_status CBLDCTAM(' INFO')	◎	－
	TAM テーブルのクローズ	dc_tam_close －	◎	－
IST サービス	IST テーブルのオープン	dc_ist_open CBLDCIST('DCISTSVC', 'OPEN')	×	IST サービスはサ ポートしていま せん。
	IST テーブルからレコードの入力	dc_ist_read CBLDCIST('DCISTSVC', 'READ')	×	
	IST テーブルヘレコードの出力	dc_ist_write CBLDCIST('DCISTSVC', 'WRIT')	×	
	IST テーブルのクローズ	dc_ist_close CBLDCIST('DCISTSVC', 'CLOS')	×	
資源の 排他制 御	資源の排他	dc_lck_get CBLDCLCK('GET ')	◎	－
	全資源の排他の解除	dc_lck_release_all CBLDCLCK('RELALL ')	◎	－
	資源名称を指定した排他の解除	dc_lck_release_byname CBLDCLCK('RELNAME ')	◎	－
XATMI インタ フェース	リクエスト / レス ポンス型サービスの呼び出しと応答の受信	tpcall() TPCALL	×	XATMI インタ フェースはサポ ートしていません。
	リクエスト / レス ポンス型サービスの呼び出し	tpacall() TPACALL	×	
	リクエスト / レス ポンス型サービスからの非同期応答の受信	tpgetrply() TPGETRPLY	×	

3. プログラムの作成

機能	関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
リクエスト / レスポンス型サービスのキャンセル	tpcancel() TPCANCEL	×	
会話型サービスとの接続の確立	tpconnect() TPCONNECT	×	
会話型サービスとの接続の切断	tpdiscon() TPDISCON	×	
会話型サービスからのメッセージの受信	tprecv() TPRECV	×	
会話型サービスへのメッセージの送信	tpsend() TPSEND	×	
型付きバッファの割り当て	tpalloc() —	×	
型付きバッファの解放	tpfree() —	×	
型付きバッファのサイズの変更	tprealloc() —	×	
型付きバッファの情報の取得	tpypes() —	×	
サービス名の広告	tpadvertise() TPADVERTISE	×	
サービス名の広告の取り消し	tpunadvertise() TPUNADVERTISE	×	
サービス関数のテンプレート	tpservice() TPSVCSTART	×	
サービス関数からのリターン	tpreturn() TPRETURN	×	
TX インタフェース	トランザクションの開始	◎	—
	トランザクションのコミット	◎	—
	現在のトランザクションに関する情報の返却	◎	—
	リソースマネージャ集合のオープン	◎	—

機能	関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
トランザクション のロールバック	tx_rollback() TXROLLBACK	◎	—
リソースマネージャ 集合のクローズ	tx_close() TXCLOSE	◎	—
commit_return 特 性の設定	tx_set_commit_return () TXSETCOMMITRET	◎	—
transaction_contr ol 特性の設定	tx_set_transaction_c ontrol() TXSETTRANCTL	◎	—
transaction_timeo ut 特性の設定	tx_set_transaction_t imeout() TXSETTIMEOUT	◎	—
マルチ ノード 機能	OpenTP1 ノードの ステータス取得の 開始	×	マルチノード機能 (TP1/Multi) はサ ポートしていません。
	OpenTP1 ノードの ステータスの取得	×	
	指定した OpenTP1 ノードのステータ スの取得	×	
	OpenTP1 ノードの ステータス取得の 終了	×	
	ノード識別子の取 得の開始	×	
	ノード識別子の取 得	×	
	ノード識別子の取 得の終了	×	
	指定したノード識 別子の取得	×	
	ユーザサーバのス テータス取得の開 始	×	
	ユーザサーバのス テータスの取得	×	

3. プログラムの作成

機能		関数名※	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
	指定したユーザ サーバのステータ スの取得	dc_adm_get_sv_status —	×	
	ユーザサーバのス テータス取得の終 了	dc_adm_get_sv_status _done —	×	
オンラ インテ スタの 管理	ユーザサーバのテ スト状態の報告	dc_uto_test_status CBLDCUTO('T-STATUS')	×	オンラインテスタ 機能 (TP1/Online Tester) はサポー トしていません。
性能検 証用ト レース	ユーザ固有の性能 検証用トレースの 取得	dc_prf_ustrace_put CBLDCPRF('PRFPUT ')	◎	—
	性能検証用トレー ス取得通番の通知	dc_prf_get_trace_num CBLDCPRF('PRFGETN ')	◎	—
リアル タイム 統計情 報サー ビス	任意区間でのリアル タイム統計情報 の取得	dc_rts_ustrace_put CBLDCRTS('RTSPUT ')	◎	—

注※

上段が C 言語の関数、下段が COBOL-UAP 作成用プログラムです。

3.2 UAP の翻訳と結合

ここでは、UAP の翻訳（コンパイル）と結合（リンケージ）をするために必要なファイルやメイクファイルの例などについて、開発環境別に説明します。

UAP のコンパイルとリンケージの方法については、マニュアル「OpenTP1 プログラム作成リファレンス」の該当する言語編を参照してください。

なお、Windows 版 OpenTP1 固有の、プログラム作成時の注意事項があります。事前に、「3.3 プログラム作成時の Windows 版 OpenTP1 固有の注意事項」の内容を確認してから、UAP のコンパイルとリンケージを実施してください。

3.2.1 C 言語での UAP の作成

ここでは、UAP を C 言語で作成する場合について説明します。

(1) C ランタイムライブラリのインストール

TP1/Server Base は Visual Studio で作成されています。このため、TP1/Server Base を動作させるためには、C ランタイムライブラリをインストールする必要があります。

インストールが必要な C ランタイムライブラリについては、リリースノートを参照してください。

(2) メイクファイルの例（C 言語：SPP の場合）

SPP を作成する場合のメイクファイルの例を示します。

(a) ソースファイル名

メイクファイルの例中で使用するソースファイル名を次に示します。

● C 言語で作成した UAP のソースプログラム名：

- exmain.c（メイン関数）
- exsv1.c（サービス関数 1）
- exsv2.c（サービス関数 2）

● RPC インタフェース定義ファイル名：ex.def

(b) メイクファイルの例

メイクファイルの例を次に示します。

3. プログラムの作成

```
NODEBUG = 1
!include <ntwin32.mak>
# TP1TEST Make File
SYSLIB = $(conlibsdl)
TP1LIB = [libdam.lib]**1 [libtam.lib]**2 libbetran.lib

all:TP1TEST.exe
#コンパイル
exmain.obj:exmain.c
$(cc) $(cflags) $(cvarsdll) *.c
exsv1.obj:exsv1.c
$(cc) $(cflags) $(cvarsdll) *.c
exsv2.obj:exsv2.c
$(cc) $(cflags) $(cvarsdll) *.c
ex_sstb.obj:ex_sstb.c
$(cc) $(cflags) $(cvarsdll) *.c
#スタブソースファイルの作成
ex_sstb.c:ex.def
$(DCDIR)¥bin¥stbmake $?
#リンケージ
TP1TEST.exe: exmain.obj exsv1.obj exsv2.obj ex_sstb.obj
$(link) $(conflags) -out:$.exe $** ¥
[$(DCDIR)¥spool¥trnrmcmd¥userobj¥dctrninf.obj ¥]**3
$(TP1LIB) $(SYSLIB)
[mt -manifest $@.manifest -outputresource:$@;1]**4
```

注※ 1

libdam.lib は、TP1/FS/Direct Access を使用する場合だけ指定してください。

注※ 2

libtam.lib は、TP1/FS/Table Access を使用する場合だけ指定してください。

注※ 3

\$(DCDIR)¥spool¥trnrmcmd¥userobj¥dctrninf.obj は、トランザクション機能を使用する場合だけ指定してください。なお、dctrninf.obj には、trnmkobj コマンドで作成したトランザクション制御用オブジェクトファイル名を指定します。trnmkobj コマンドについては、「7. 運用コマンド」を参照してください。

注※ 4

マニフェストファイルを埋め込まない場合は、この行の内容を削除してください。

(3) メイクファイルの例 (C 言語: MHP の場合)

MHP を作成する場合のメイクファイルの例を次に示します。

```

NODEBUG = 1
!include <ntwin32.mak>
# makefile for making UAP(C Language)

# command
STBMAKE           =          $(DCDIR)¥bin¥stbmake.exe

# stub name
STBMAKE_MHP_SRC   =          mhp_sstb

# user server object
MHP_SRC           =          mhpmain.obj mhp_sv.obj

# include file
INCDIR            =          -I.                                     ¥
                    -I $(DCDIR)¥include

# common object
SH_OBJ            =          $(DCDIR)¥spool¥trnmcmd¥userobj¥mcf_sw.obj

# common library
LIBNAME           =          $(DCDIR)¥lib¥libmcf.lib ¥
                    $(DCDIR)¥lib¥libmnet.lib ¥
                    $(DCDIR)¥lib¥libbetran.lib

# compiler options
CC                =          $(cc)
CCOPT             =          -nologo $(cflags) $(cvarsdll)
CONFIG            =          -c
CFLAGS            =          $(CCOPT) $(CONFIG) $(INCDIR)

# linker options
LK               =          $(link)
LNKFLG           =          $(conflags) /NODEFAULTLIB
LINKAGE          =          $(conlibsdl) $(LIBNAME) msvcrt.lib

.c.obj:
    $(CC) *.c $(CFLAGS)

all : $(SH_OBJ) tcpmhp.exe

$(STBMAKE_MHP_SRC).c :
    $(STBMAKE) -s $(STBMAKE_MHP_SRC).c $(STBMAKE_MHP_SRC).def

tcpmhp.exe : $(STBMAKE_MHP_SRC).obj $(MHP_SRC)
    $(LK) $(LNKFLG) $** $(LINKAGE) $(SH_OBJ) -out:$@
    !if exist $@.manifest mt.exe -nologo -manifest $@.manifest
    -outputresource:$@;1
    !if exist $@.manifest del $@.manifest

$(SH_OBJ) :
    $(DCDIR)¥bin¥trnmkobj -o mcf_sw -R OpenTP1_MCF

```

3.2.2 Visual Studio のプロジェクトでの UAP の作成

Visual Studio のプロジェクトで UAP を作成する場合について説明します。ここでは Visual Studio 2013 の場合を例に説明します。

(1) プロジェクトの種類

指定するプロジェクトの種類は、「Win32 プロジェクト」です。

(2) プロジェクトを構成するファイル

指定するファイルは、プロジェクトを構成するソースファイルとして作成したアプリ

3. プログラムの作成

ケーションのソース、スタブソースファイル※、および必要に応じてトランザクション制御用オブジェクトファイルです。

注※

スタブソースファイルをプロジェクトに追加する場合は、あらかじめ stbmake コマンドを実行し、スタブソースファイルを生成しておく必要があります。スタブソースファイルおよび stbmake コマンドについては、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。

(3) プロジェクトの設定

Visual Studio のプロジェクトの設定内容を次に示します。

表 3-2 Visual Studio のプロジェクトの設定内容

カテゴリ	項目		設定値
C/C++	全般	追加のインクルードディレクトリ	OpenTP1 インストールフォルダ ¥include
	コード生成	ランタイムライブラリ	マルチスレッド DLL (/MD)
	プリコンパイル済みヘッダー	プリコンパイル済みヘッダー	プリコンパイル済みヘッダーを使用しない
リンカー	全般	追加のライブラリディレクトリ	OpenTP1 インストールフォルダ ¥lib
	入力	追加の依存ファイル	• msvert.lib • libbetran.lib ※
		すべての既定のライブラリの無視	はい (/NODEFAULTLIB)
	システム	サブシステム	コンソール (/SUBSYSTEM:CONSOLE)

! 注意事項

Visual Studio 2013 以降を使用してプロジェクトから UAP を作成する場合は、「リンカー」－「システム」－「最低限必要なバージョン」に、UAP を実行する OS の種類に応じた適切な値を指定してください。指定値の詳細については、Microsoft サポートページをご確認ください。

注※

必要に応じて次のファイルも設定してください。

- TP1/FS/Direct Access を使用する場合：libdam.lib
- TP1/FS/Table Access を使用する場合：libtam.lib
- TP1/Message Control を使用する場合：libmcf.lib, libmnet.lib

(4) Visual Studio のプロジェクトでサンプルプログラムを作成する

ここでは、TP1/Server Base C 言語サンプルの SPP をプロジェクトで作成する場合について記載します。

次のソースファイルをプロジェクトに追加してください。

- spp.c
- sv.c
- custom.h
- sv_sstb.c[※]

注※

あらかじめ stbmake コマンドで sv_sstb.c を作成する必要があります。

サービス関数動的ローディング機能の SPP を作成する場合は、sv_sstb.c は不要です。

「(3) プロジェクトの設定」で示した設定に加えて、次の設定をする必要があります。

表 3-3 サンプルプログラムを作成する場合に必要な設定

カテゴリ	項目		設定値
C/C++	プリプロセッサ	プリプロセッサの定義	_CRT_SECURE_NO_DEPRECATED
ライブラリアン	全般	出力ファイル	UAP 共用ライブラリを作成する場合 libsv.lib
		関数のエクスポート	/export:refer /export:update
	コマンドライン	追加のオプション	UAP 共用ライブラリを作成する場合 /def
リンカー	全般	出力ファイル	basespp.exe サービス関数動的ローディング機能の場合 basespp2.exe UAP 共用ライブラリを作成する場合 libsv.dll
	入力	追加の依存ファイル	UAP 共用ライブラリを作成する場合 libsv.exp

！ 注意事項

- ・ サービス関数動的ローディング機能では、UAP 共用ライブラリを作成する必要があります。
 - ・ サンプルプログラム ".exe" を作成するプロジェクトの場合、アプリケーションの種類は Windows アプリケーションを選択してください。
 - ・ UAP 共用ライブラリ (libsv.lib) を作成するプロジェクトの場合、アプリケーションの種類はスタティックライブラリを選択してください。
 - ・ UAP 共用ライブラリ (libsv.dll) を作成するプロジェクトの場合、アプリケーションの種類は DLL を選択してください。
-

3.2.3 COBOL2002 での UAP の作成

ここでは、UAP を COBOL2002 で作成する場合について説明します。

(1) コマンドの指定 (COBOL2002)

次に示す場合に指定するコマンドについて説明します。

(a) trnmkobj コマンドでトランザクション制御用オブジェクトを作成する場合

trnmkobj コマンドでトランザクション制御用オブジェクトを作成する場合には、必ず -C "/ZI" オプションを指定してください。

(b) 標準トランザクション制御用オブジェクトファイル (dc_trn_allrm.obj) を使用する場合

trnlnkrm コマンドで作成される標準トランザクション制御用オブジェクトファイル (dc_trn_allrm.obj) を使用する場合、trnlnkrm コマンドを -C "/ZI" オプションを指定して実行し、オブジェクトを再作成してください。

(c) スタブファイルをコンパイルする場合

stbmake で作成された C 言語のソースファイルをコンパイルする場合、必ずコンパイルオプション "/ZI" を指定してください。

(2) メイクファイルの例 (COBOL2002：SPP の場合)

SPP を作成する場合のメイクファイルの例を示します。

(a) ソースファイル名

メイクファイルの例中で使用するソースファイル名を次に示します。

- COBOL2002 で作成した UAP のメインプログラム名：main.cbl
- COBOL2002 で作成した UAP のサービスプログラム名 1：sev1.cbl
- COBOL2002 で作成した UAP のサービスプログラム名 2：sev2.cbl

● RPC インタフェース定義ファイル名：ex.def

(b) メイクファイルの例

メイクファイルの例を次に示します。

```
!include <ntwin32.mak>
# TP1TEST Make File
CBLFG1 = -Compile,NoLink -Comp5 -Lib,CUI -Main,System
CBLFG2 = -Compile,NoLink -Comp5 -Lib,CUI
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]*1 [libtam.lib]*2 libbetran.lib
CCBL = ccbl2002
LK = ccbl2002 -Lib,CUI

all:TP1TEST.exe
#コンパイル
main.obj:main.cbl
    $(CCBL) $(CBLFG1) main.cbl
sev1.obj:sev1.cbl
    $(CCBL) $(CBLFG2) sev1.cbl
sev2.obj:sev2.cbl
    $(CCBL) $(CBLFG2) sev2.cbl
ex_sstb.obj:ex_sstb.c
    [$(cc) $(cflags) /Zl $(cvarsdll) *.c]*3
#スタブソースファイルの作成
ex_sstb.c:ex.def
    $(DCDIR)¥bin¥stbmake $?
#リンケージ
TP1TEST.exe: main.obj sev1.obj sev2.obj ex_sstb.obj
    $(LK) -OutputFile *.exe $** ¥
    [$(DCDIR)¥spool¥trnmcmd¥userobj¥dctrninf.obj ¥]*4
    $(TP1LIB) $(SYSLIB)
```

注※ 1

libdam.lib は、TP1/FS/Direct Access を使用する場合だけ指定してください。

注※ 2

libtam.lib は、TP1/FS/Table Access を使用する場合だけ指定してください。

注※ 3

ex_sstb.obj は、コンパイルオプション /Zl を指定してください。

注※ 4

\$(DCDIR)¥spool¥trnmcmd¥userobj¥dctrninf.obj は、トランザクション機能を使用する場合だけ指定してください。なお、dctrninf.obj には、trnmkobj コマンドで作成したトランザクション制御用オブジェクトファイル名を指定します。また、trnmkobj コマンドを実行する場合、-C "/Zl" オプションを指定して、オブジェクトにマニフェストファイルの情報が埋め込まれないようにしてください。trnmkobj コマンドについては、「7. 運用コマンド」を参照してください。

(3) メイクファイルの例（COBOL2002：MHP の場合）

MHP を作成する場合のメイクファイルの例を示します。

3. プログラムの作成

```
!include <ntwin32.mak>
# makefile for making UAP(COBOL Language)

# COBOL2002 install directory
COBOL2002      = C:¥Progra~1¥HITACHI¥COBOL2002

# command
STBMAKE        = $(DCDIR)¥bin¥stbmake.exe

# user server object
MHP_SRC        = CBLMHPM.obj TCPMHPSV.obj

# include file
INCDIR          =      -I.                ¥
                 -I $(DCDIR)¥include

# common object
SH_OBJ          =      $(DCDIR)¥spool¥trnrmcmd¥userobj¥mcf_sw.obj

# common library
LIBNAME         =      $(DCDIR)¥lib¥libmcf.lib ¥
                       $(DCDIR)¥lib¥libmnet.lib ¥
                       $(DCDIR)¥lib¥libbetran.lib

# COBOL2002 library
CBLLIBNAME      =      $(COBOL2002)¥lib¥cbl2k_32.lib ¥
                       $(COBOL2002)¥lib¥cbl2k1c.lib ¥
                       $(COBOL2002)¥lib¥cbl2kdl.lib

# compiler options
CC              =      cl
CCBL            =      $(COBOL2002)¥bin¥ccbl2002
CCOPT           =      -nologo $(cflags) $(cvarsdll)
CONFIG          =      -c -Zl
CFLAGS          =      $(CCOPT) $(CONFIG) $(INCDIR)
CBLFG1          =      -Compile,Nolink -Comp5 -Lib,CUI -OpenTP1 -Main,System
CBLFG2          =      -Compile,Nolink -Comp5 -Lib,CUI -OpenTP1

# linker options
LK              =      ccbl2002 -Lib,CUI
LNKFLG          =      $(conflags) /NODEFAULTLIB
LINKAGE         =      $(conlibsdl) $(LIBNAME) $(CBLLIBNAME) msvcr.lib

all : $(SH_OBJ) tcpmhp.exe

.SUFFIXES : .cbl

# compile
.c.obj :
                $(CC) $(CFLAGS) *.c

.cbl.obj :
                $(CCBL) $(CBLFG2) *.cbl

TCPMHPSV.obj :
                $(CCBL) $(CBLFG2) TCPMHPSV.cbl

CBLMHPM.obj :
                $(CCBL) $(CBLFG1) cblmhp.cbl
```



```
# stub, creating source file
cblmhp_sstb.c :
$(STBMAKE) cblmhp.def

# link
tcpmhp.exe : $(MHP_SRC) cblmhp_sstb.obj
$(LK) -OutputFile $*.exe $** $(LINKAGE) $(SH_OBJ)
!if exist $@.manifest mt.exe -nologo -manifest $@.manifest
-outputresource:$@;1
!if exist $@.manifest del $@.manifest

$(SH_OBJ) :
$(DCDIR)¥bin¥trnmkobj -o mcf_sw -R OpenTP1_MCF -C "/Zl"
```

(4) COBOL2002 開発マネージャの設定

COBOL2002 開発マネージャは、次のように設定してください。

● コンパイラオプションの設定

「CUI : CUI 用の実行時ライブラリを使用する」を指定してください。

また、MHP の場合は、「-OpenTP1:OpenTP1 を使用した DC 機能を使用する」を指定してください。

● リンカオプションの設定

インポートライブラリ／ユーザ指定ライブラリとして libbetran.lib ※, スタブオブジェクトファイル, および必要に応じてトランザクション制御用オブジェクトファイルを指定してください。

注※

必要に応じて次のファイルも指定してください。

- TP1/FS/Direct Access を使用する場合 : libdam.lib
- TP1/FS/Table Access を使用する場合 : libtam.lib
- TP1/Message Control を使用する場合 : libmcf.lib, libmnet.lib

(5) Visual Studio と併用する際の環境設定

COBOL2002 と Visual Studio 併用時の注意事項については、COBOL2002 のリリースノートの注意事項をご確認ください。

3.3 プログラム作成時の Windows 版 OpenTP1 固有の注意事項

ここでは、プログラムの作成前に知っておきたい、Windows 版 OpenTP1 固有の注意事項について説明します。

(1) スタブの作成時の注意事項

stbmake コマンドで指定するスタブソースファイル名には、RPC インタフェース定義ファイルを示すサフィックス ".def" を必ず付与してください。

また、RPC インタフェース定義ファイルの文字コードは、シフト JIS で記述してください。

(2) シングルスレッド環境での UAP 実行時の注意事項

OpenTP1 の UAP は、メインスレッド以外にスレッドを作成しないで、シングルスレッド環境で実行してください。メインスレッド以外のスレッドを作成して実行すると（マルチスレッド環境）、UAP が誤動作して異常終了するおそれがあります。ただし、OpenTP1 下の UAP は、DLL を使用したマルチスレッド環境で動作するため、UAP 作成時に指定するリンケージランタイムライブラリには、msvcrt.lib を指定してください。

なお、JNI（Java Native Interface）などのスレッドが生成されるコーディングをした場合、動作は保証できません。

(3) UAP の終了時の注意事項

サービス関数（SPP）およびサービスメソッド（SPP.NET）内で、プロセスおよびスレッドを停止させる関数またはメソッドを発行しないでください。発行した場合、次のことが発生するおそれがあります。

- サービス関数（SPP）およびサービスメソッド（SPP.NET）が予期しない動作をする。
- サービスグループに対するサービス要求が失敗する。
- UAP を終了できなくなる。
- OpenTP1 が正常終了できなくなる。

(4) Visual Studio 使用時の注意事項

Visual Studio のデバッグバージョンで作成したオブジェクトファイルは、Visual Studio がインストールされていない環境では動作できないおそれがあります。Visual Studio がインストールされていない環境でデバッグする場合、リンケージされた DLL ファイルおよびマニフェストファイルの内容を確認し、必要に応じて適切なマニフェストファイルを作成してください。マニフェストファイルの詳細については、Visual Studio のマニユ

アルのマニフェストファイルに関する記述を参照してください。

(5) Visual Studio と COBOL 言語を使用する環境が共存する場合の注意事項

Visual Studio と COBOL 言語を使用する環境が共存する場合に各言語をコンパイルするときは、コンパイルする各言語の環境変数がほかの言語より先に設定されていることを確認してください。ほかの言語の環境変数が先に設定されていると、意図しないリンケージが行われることがあります。

3.4 サンプルプログラム

ここでは、Windows 版 OpenTP1 のサンプルプログラムについて、UNIX 版 OpenTP1 との差異を開発言語別に説明します。サンプルプログラムそのものについては、OpenTP1 シリーズのマニュアルを参照してください。

(1) C 言語の場合

C 言語のサンプルプログラムには、UNIX 版 OpenTP1 との次のような差異があります。

- %DCDIR%\examples\base\conf にある各サンプルのユーザサービス定義ファイルの名前は、bsespp_c (SPP)、bsesup_c (SUP) です。各サンプルを起動するときの dcsvstart コマンドの引数にはこのファイル名を指定してください。
- UNIX 版 OpenTP1 で提供している chconf (定義ファイル修正コマンド)、bkconf (chconf で変更した定義ファイルを元に戻すコマンド) は、Windows 版 OpenTP1 では提供していません。
- Visual Studio を使用してサンプルプログラムを作成する場合、次のメッセージが出力されることがあります。
「LINK : warning LNK4098: defaultlib 'LIBCMT' は他のライブラリの使用と競合しています。/NODEFAULTLIB:library を使用してください。」
このメッセージが出力された場合、/NODEFAULTLIB オプションを使用し、必ずランタイムライブラリ msvcrt.lib だけがリンケージされるようにしてください。複数種類のランタイムライブラリがリンケージされたプログラムは、予期しない動作を起こすことがあります。

(2) COBOL 言語の場合

COBOL 言語のサンプルプログラムには、UNIX 版 OpenTP1 との次のような差異があります。

- %DCDIR%\examples\base\conf にある各サンプルのユーザサービス定義ファイルの名前は、bsespp_b (SPP)、bsesup_b (SUP) です。各サンプルを起動するときの dcsvstart コマンドの引数にはこのファイル名を指定してください。
- UNIX 版 OpenTP1 で提供している chconf (定義ファイル修正コマンド)、bkconf (chconf で変更した定義ファイルを元に戻すコマンド) は、Windows 版 OpenTP1 では提供していません。
- COBOL2002 を使用して UAP を作成する場合には、nmake コマンドを実行してください。
- COBOL 言語のサンプルプログラムの引数の数値データ形式には、COMP ではなく COMP-5 を使用しています。COBOL2002 で COMP-5 を使用する場合には、-Comp5 オプションを指定してコンパイルする必要があります。

3.5 サービス関数動的ローディング機能使用時の UAP 共用ライブラリ

ここでは、サービス関数動的ローディング機能使用時の UAP 共用ライブラリについて説明します。サービス関数動的ローディング機能の概要については OpenTP1 シリーズのマニュアルを参照してください。

なお、サービス関数動的ローディング機能は、次の場合は使用できません。

- SPP.NET サーバ
- マネージドコードの UAP 共用ライブラリ
マネージドコードの UAP 共用ライブラリに含まれるサービス関数は、この機能で呼び出せません。

(1) UAP 共用ライブラリのエクスポート

サービス関数が UAP 共用ライブラリに存在する場合、ユーザサーバ起動時の UAP 共用ライブラリに含まれる、ユーザサービス定義で指定するサービス関数は、必ず UAP 共用ライブラリからエクスポートしてください。

エクスポートされていない場合（非常駐サーバの場合はプロセス起動時）は、次のように動作します。

SPP の場合

KFCA00344-E メッセージを出力後、`dc_rpc_mainloop` 関数が
`DCRPCER_FATAL(-303)` でエラーリターンします。

MHP の場合

KFCA00344-E メッセージを出力後、`dc_mcf_mainloop` 関数が
`DCMCFER_FATAL(-11902)` でエラーリターンします。

ユーザサービス定義で指定するサービス関数以外の関数のエクスポートは必須ではないため、処理形態に応じてエクスポートしてください。

UAP 共用ライブラリから関数をエクスポートするには、`lib` コマンドや、`link` コマンドを用いる方法があります。次のどれかの方法でサービス関数をエクスポートしてください。

- `lib` コマンドまたは `link` コマンドで指定するオブジェクトファイル内で、`__declspec (dllexport)` でサービス関数名を定義する。
- `lib` コマンドで、`/EXPORT:` サービス関数名を指定する。
- `lib` コマンドで、`/DEF:` モジュール定義ファイルを指定し、モジュール定義ファイル中に `EXPORTS` 文形式でサービス関数を定義する。

作成例については、次のサンプルソース（メイクファイル）を参照してください。

3. プログラムの作成

SPP 開発言語と開発環境		メイクファイル名称
C 言語		%DCDIR%\examples\base\aplib%c\make_svd1
COBOL 言語	COBOL2002	%DCDIR%\examples\base\aplib%cobol\make_svd1
	Net Express	%DCDIR%\examples\base\aplib%cobol\makemf_svd1

(2) ライブラリサーチパスの処理

UAP 共用ライブラリ名に UAP 共用ライブラリ名称だけを指定する場合、OpenTP1 は指定された UAP 共用ライブラリを、OS のライブラリサーチパスを基に検索して使用します。Windows 上では、次の検索パスで、指定 UAP 共用ライブラリのサーチ処理が行われます。なお、次の検索パスは、優先度の高い順に並んでいます。

- UAP が存在するフォルダ
- UAP のカレントフォルダ (%DCDIR%\tmp\home\ 各サーバのフォルダ) ※1
- Windows のシステムフォルダ※2
- Windows フォルダ※2
- プロセスサービス定義の prcsvpath 定義コマンドに指定したフォルダ
フォルダが複数指定されている場合、最初に検索されたフォルダ下の UAP 共用ライブラリを使用します。

注※1

OpenTP1 の管理する一時フォルダです。OpenTP1 システムや UAP の起動時に削除されますので、このフォルダ下には作成した UAP 共用ライブラリを格納しないでください。

注※2

Windows システムフォルダや Windows フォルダに、作成した UAP 共用ライブラリを格納することは推奨していません。格納しないでください。

次の例では、(b) のフォルダ下にある UAP 共用ライブラリが UAP 上にローディングされます。

```
UAP の存在するパス名：c:\user\aplib\user.exe
DLL の存在するパス名：c:\opentp1\aplib\libusr.dll
                        c:\user\aplib\libusr.dll
< プロセスサービス定義 >
prcsvpath c:\opentp1\aplib; c:\user\aplib
          (a)              (b)
```

(3) ユーザサーバプロセスのサービス関数の入れ替え

サービス関数動的ローディング機能の使用時は、オンラインを停止しなくてもユーザサーバのサービス関数を入れ替えられます。ここでは、サービス関数の入れ替え手順、

および入れ替え後の確認手順について説明します。

(a) サービス関数の入れ替え

ユーザサービス定義を変更する場合と、UAP 共用ライブラリサーチパスを変更する場合に分けて、サービス関数の入れ替え方法を説明します。

ユーザサービス定義を変更する場合

1. 稼働中の該当するユーザサーバを `dcsvstop` コマンドで終了します。
2. 該当するユーザサーバのユーザサービス定義の `service` オペランドを変更します。
3. `dcsvstart` コマンドで該当するユーザサーバを開始します。

UAP 共用ライブラリサーチパスを変更する場合

1. 稼働中の該当するユーザサーバを、`dcsvstop` コマンドで終了します。
2. 新しい UAP 共用ライブラリ格納フォルダに、UAP 共用ライブラリを配置します。
3. `prepath` コマンドで、UAP 共用ライブラリサーチパスを、新しい UAP 共用ライブラリ格納フォルダに変更します。
4. `dcsvstart` コマンドで該当するユーザサーバを開始します。

(b) サービス関数の入れ替え後の確認

新しい実行形式ファイルの起動時刻を確認し、`dcsvstart` コマンドまたは `scdrspre` コマンドの実行時刻と比較してください。

新しい実行形式ファイルの起動時刻は、次の手順で確認します。ここでは、Windows Server 2008 R2 の場合を例に記載しています。

1. **【スタート】－【ファイル名を指定して実行】** を選択します。
2. 「名前」に「`Msinfo32.exe`」を入力して実行すると、**【システム情報】ダイアログボックス**が開きます。
3. **【システム情報】ダイアログボックス**の左ペインにある**【ソフトウェア環境】－【実行中のタスク】**をクリックします。
4. 右ペインに表示されるファイルから**実行形式ファイル**を探し、**開始時刻**を確認します。

(4) UAP 共用ライブラリ名の指定

ユーザサービス定義の `service` オペランドには、絶対パスと、ライブラリ名称の指定を併記できます。

ただし、絶対パスで指定した UAP 共用ライブラリ名とライブラリ名称で指定した UAP 共用ライブラリ名が同一の場合、記述する順序によってローディングされる UAP 共用ライブラリパスが異なります。

先に指定する名称別に、次に説明します。

3. プログラムの作成

ライブラリ名称で指定した UAP 共用ライブラリ名を先に記述する場合

ライブラリ名称で指定した UAP 共用ライブラリに対し、ライブラリ名称で指定したパスの検索順序に従ってライブラリパスが検索されます。この結果、最初に検索されたフォルダ下の UAP 共用ライブラリがローディングされます。

次に、絶対パスで指定した UAP 共用ライブラリがローディングされます。

この指定順序の場合、2 種の異なる UAP 共用ライブラリが UAP 上にローディングされます。

絶対パスで指定した UAP 共用ライブラリ格納フォルダが最初に検索された場合は、同一の UAP 共用ライブラリが 2 回ローディングされます。

絶対パスで指定した UAP 共用ライブラリ名を先に記述する場合

まず、絶対パスで指定した UAP 共用ライブラリがローディングされます。次に、ライブラリ名称で指定した UAP 共用ライブラリをローディングしようと試みます。しかし、先に絶対パスで同一名称の UAP 共用ライブラリがローディングされているため、絶対パスで指定した UAP 共用ライブラリが再度ローディングされます。この指定順序の場合、絶対パスで指定した UAP 共用ライブラリが、UAP 上に 2 回ローディングされます。

(5) サービス関数動的ローディング機能使用時の UAP 共用ライブラリの注意事項

ユーザーサービス定義の service オペランドで指定した UAP 共用ライブラリ内で、TLS（スレッドローカルストレージ）を使用した変数宣言をしないでください。次のようなコーディングを含む UAP オブジェクトが UAP 共用ライブラリ内に存在する場合、該当する変数を参照するときに、アクセス例外が発生します。

```
__declspec(Thread) int sample;
```


4

環境設定

Windows 版 OpenTP1 は，UNIX 版 OpenTP1 と比べて，環境設定の手順が異なります。この章では，Windows 版 OpenTP1 固有の環境設定について説明します。

UNIX 版 OpenTP1 と共通の内容については，OpenTP1 シリーズのマニュアルを参照してください。

4.1 環境設定の手順

4.2 OpenTP1 管理者の登録

4.3 OpenTP1 のシステム定義情報の作成

4.4 環境変数の設定

4.5 OpenTP1 ファイルシステムの構築

4.6 データベース連携時の設定

4.7 OS の設定

4.8 マルチ OpenTP1 の設定

4.1 環境設定の手順

ここでは、OpenTP1 開始直前までの環境設定の手順を説明します。Windows 版 OpenTP1 の環境設定の手順を次に示します。

図 4-1 Windows 版 OpenTP1 の環境設定の手順

環境設定の手順	このマニュアル の参照先	参照するOpenTP1シリーズ のマニュアル
OpenTP1管理者の登録	4.2	「OpenTP1 運用と操作」
OpenTP1のインストール	—※1	—※1
システム定義の作成	4.3	「OpenTP1 システム定義」 「OpenTP1 運用と操作」
環境変数の設定	4.4	—
OpenTP1ファイルシステム領域の作成	4.5	「OpenTP1 解説」 「OpenTP1 運用と操作」
OpenTP1ファイルシステムの初期設定	4.5	「OpenTP1 運用と操作」
OpenTP1ファイルの作成	—	「OpenTP1 解説」 「OpenTP1 運用と操作」
OpenTP1ファイル以外のファイルの作成	—	「OpenTP1 解説」 「OpenTP1 運用と操作」
リソースマネージャの登録	4.6※2	「OpenTP1 運用と操作」
トランザクション制御用オブジェクト ファイルの作成	4.6※2	「OpenTP1 運用と操作」
OSの設定	4.7	—

(凡例)

-  : 必須の設定
 : 任意の設定
 — : 該当しません。

注※1 インストールについては、製品のリリースノートを参照してください。

注※2 ここでは、Windows版OpenTP1で各データベースと連携するための設定について説明します。このほかのリソースマネージャの設定については、マニュアル「OpenTP1 運用と操作」を参照してください。

Windows 版 OpenTP1 では、OpenTP1 管理者が環境を設定します。図中の参照先を基に、この章の該当箇所、および OpenTP1 シリーズのマニュアルを参照して、環境を設定してください。

なお、マルチ OpenTP1 を使用する場合は設定については、「4.8 マルチ OpenTP1 の設定」を参照してください。

注意事項

TP1/Message Control や TP1/Message Queue などの製品が提供する機能を使う場合には、図中に示した手順に加えて、各製品を使用するための専用の手順が必要になります。設定が必要な専用の手順は、UNIX 版 OpenTP1 と共通です。詳細については、マニュアル「OpenTP1 運用と操作」の環境設定手順の概要についての説明を参照してください。

参考

Windows 版 OpenTP1 では、GUI を使用して、OpenTP1 を使用するための最低限の環境設定（システム定義の作成など）をすることもできます。GUI による OpenTP1 の環境設定については、「6.7 Windows 版 OpenTP1 の GUI」を参照してください。

4.2 OpenTP1 管理者の登録

Windows 版 OpenTP1 では、Administrators グループの権限を持つユーザが OpenTP1 管理者になります。なお、OpenTP1 管理者の名称に制限はありません。

このほかの内容については、UNIX 版 OpenTP1 と共通です。UNIX 版 OpenTP1 と共通の部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

4.3 OpenTP1 のシステム定義情報の作成

ここでは、Windows 版 OpenTP1 での OpenTP1 のシステム定義情報の作成について、UNIX 版 OpenTP1 との差異を説明します。UNIX 版 OpenTP1 と共通の部分については、マニュアル「OpenTP1 システム定義」、およびマニュアル「OpenTP1 運用と操作」を参照してください。

4.3.1 定義ファイルの作成

OpenTP1 のシステム定義の定義ファイルは、テキストエディタを使用して作成します。定義ファイルを作成するときの注意事項を次に示します。

- 定義ファイルの文字コードは、シフト JIS で記述してください。
- 定義ファイルの行の最後に、必ず「改行」を入れてください。テキストエディタのバージョンによっては、ファイルの終端に EOF 制御文字が自動的に入らないため、定義ファイルの最後の行に定義が記述されていても、この定義を認識できない場合があります。

作成したシステム定義は、`dcdefchk` コマンドでチェックできます。

システム定義および定義チェックの詳細については、マニュアル「OpenTP1 システム定義」を参照してください。

4.3.2 定義の変更

Windows 版 OpenTP1 では、UNIX 版 OpenTP1 と比べて、システム定義の変更の手順が異なります。Windows 版 OpenTP1 では、次に示す手順でシステム定義を変更してください。

1. OpenTP1 を正常終了させます。
2. システム定義の定義内容を変更します。
3. OpenTP1 を正常開始させます。

このほかの内容については、UNIX 版 OpenTP1 と共通です。定義の変更手順については、マニュアル「OpenTP1 システム定義」を参照してください。

4.3.3 ランタイムライブラリのパスの設定

COBOL-UAP を使用する場合やデータベースと連携する場合など、OpenTP1, Windows, または Microsoft Visual Studio が提供するランタイムライブラリ以外のランタイムライブラリを使用する際には、ランタイムライブラリのパスを設定してください。

4. 環境設定

(1) パスの設定

プロセスサービス定義の `prcsvpath` 定義コマンド、または `prepath` コマンドで、ランタイムライブラリの格納フォルダを指定してください。

プロセスサービス定義の `prcsvpath` 定義コマンドについては、「5. システム定義」を参照してください。

`prepath` コマンドについては、「7. 運用コマンド」を参照してください。

(2) ユーザサーバが起動直後にダウンする場合の対処

ユーザサーバが起動直後に終了状態 `0x00000080` または `0xC0000135` でダウンする場合、ユーザサーバが起動時にロードするランタイムライブラリへのパスが設定されていないことが考えられます。ユーザサーバ作成時にリンケージしたランタイムライブラリを確認し、必要なランタイムライブラリの格納フォルダのパスをプロセスサービス定義の `prcsvpath` 定義コマンドで指定してください。

なお、ユーザサーバの格納フォルダをエクスプローラで開き、ダウンする EXE ファイルをダブルクリックすると、エラーダイアログが表示されて、ロードできないランタイムライブラリ名がわかることがあります。ただし、環境変数 `PATH` にランタイムライブラリの格納フォルダが記述されている場合は、ランタイムライブラリ名は表示されません。この場合、Visual Studio の `dumpbin` コマンドを使用すれば、EXE ファイルがロードするランタイムライブラリ名を調べられます。ランタイムライブラリを調べるときの `dumpbin` コマンドの実行例を次に示します。

```
dumpbin /DEPENDENTS %DCDIR%\%aplib%\basespp.exe
```

4.4 環境変数の設定

Windows 版 OpenTP1 では、OpenTP1 のコマンドを実行するために、ログイン環境に環境変数を設定します。Windows 版 OpenTP1 の環境変数を次に示します。

表 4-1 Windows 版 OpenTP1 の環境変数

環境変数名	設定内容
DCCONFPATH ※	システム定義の定義ファイルを格納するフォルダの絶対パス名を 246 バイト以下で指定します。空白文字を含むパス名は指定できません。 なお、環境変数 DCCONFPATH は、同一マシン内ですべて同じ文字列となるように設定してください。また、大文字と小文字の区別に注意してください。
DCDIR	環境変数 DCDIR は、ユーザによる設定は必要ありません。OpenTP1 をセットアップすると、環境変数 DCDIR に OpenTP1 フォルダのパス名が自動的に設定されます。 環境変数 DCDIR は、定義ファイルやパッチファイルに設定しないでください。システムの運用上、設定が必要な場合は、大文字と小文字の区別に注意して、環境変数 DCDIR と同一の文字列を指定してください。
DCUAPCONFPATH ※	ユーザーサービス定義の定義ファイルを環境変数 DCCONFPATH で指定したフォルダとは別のフォルダに格納したい場合、そのフォルダの絶対パス名を 246 バイト以下で指定します。空白文字を含むパス名は指定できません。
INCLUDE	ユーザーサーバのプログラムを作成する場合に、%DCDIR%\include を指定します。
LIB	trnlkrm コマンドを実行する場合、またはユーザーサーバのプログラムを作成する場合に、%DCDIR%\lib を指定します。
PATH	OpenTP1 をセットアップすると、環境変数 PATH に OpenTP1 の実行形式ファイル格納パス (%DCDIR%\bin) が自動的に追加されます。 %DCDIR%\bin の指定は削除しないでください。削除した場合、OpenTP1 は正常に動作しません。 なお、環境変数 PATH が設定済みの場合、%DCDIR%\bin は最後に追加されます。このとき、%DCDIR%\bin よりも前にネットワーク接続されたドライブのパスが指定されていると、OpenTP1 起動時にアプリケーションエラーが発生し、OpenTP1 が起動できなくなります。ネットワーク接続されたドライブのパスは、%DCDIR%\bin よりも後ろに指定してください。
TZ	タイムゾーンを指定します。 「コントロールパネル」の「日付と時刻」にある時間帯の欄と値の意味を合わせてください。例えば、時間帯に「(GMT+9:00) 大阪, 札幌, 東京」, 「(GMT+9:00) ソウル」, および「(GMT+9:00) ヤクーツク」が指定されている場合、環境変数 TZ の値は「JST+9」になります。環境変数 TZ が設定されていない場合や、システム内で時刻が統一されていない場合、OpenTP1 内部で正しい時間を取得できないことがあります。

注※

OpenTP1 は、システム開始時に環境変数 DCCONFPATH と環境変数 DCUAPCONFPATH で指定したフォルダにあるファイルを定義ファイルとして解析します。指定したフォルダ内に、OpenTP1 で使用する定義ファイル以外のファイルが存在すると、OpenTP1 の起動に時間が掛

4. 環境設定

かる場合があります。このため、環境変数 `DCCONFPATH` と環境変数 `DCUAPCONFPATH` で指定したフォルダには、`OpenTP1` で使用する定義ファイル以外は格納しないでください。

4.5 OpenTP1 ファイルシステムの構築

ここでは、Windows 版 OpenTP1 での OpenTP1 ファイルシステムの構築について、UNIX 版 OpenTP1 との差異を説明します。UNIX 版 OpenTP1 と共通の部分については、マニュアル「OpenTP1 解説」、およびマニュアル「OpenTP1 運用と操作」を参照してください。

4.5.1 OpenTP1 ファイルシステム

OpenTP1 ファイルシステムは、OS が提供するファイルシステムとは独立した、OpenTP1 専用のファイルシステムです。Windows 版 OpenTP1 では、通常の Windows のファイル上、または Windows のダイレクトディスクアクセス (raw I/O) の機能 (raw I/O 機能) を適用した領域に OpenTP1 ファイルシステムを構築します。

ここでは、OpenTP1 ファイルシステムの特性、および OpenTP1 ファイルシステム選択時の留意点について説明します。また、Windows 版 OpenTP1 の OpenTP1 ファイルシステムの注意事項についても説明します。OpenTP1 ファイルシステムの特性を考慮した上で、通常の Windows のファイルと raw I/O 機能を適用した領域のどちらに OpenTP1 ファイルシステムを構築するかを選択してください。

通常の Windows のファイルでの OpenTP1 ファイルシステムの構築については、マニュアル「OpenTP1 解説」、およびマニュアル「OpenTP1 運用と操作」を参照してください。raw I/O 機能を適用した OpenTP1 ファイルシステムの構築については、「4.5.2 raw I/O 機能を適用した OpenTP1 ファイルシステムの構築」を参照してください。

(1) OpenTP1 ファイルシステムの特性

OpenTP1 のオンライン時の OpenTP1 ファイルシステムに対する書き込み処理、および読み込み処理の性能について説明します。

● OpenTP1 ファイルシステムへの書き込み処理の性能※

- 通常の Windows のファイルの場合

Windows の制御上での遅延書き込みによるデータ紛失防止のため、書き込みごとにディスクに対してデータをフラッシュしています。ディスクに対するフラッシュ処理は、書き込みだけを行う場合に比べ、性能コストが掛かります。

- raw I/O 機能を適用した場合

書き込みごとのフラッシュ処理が不要となるため、通常の Windows のファイルへの書き込みに比べ、フラッシュ処理分の性能コストを削減できます。

注※

OpenTP1 ファイルシステム以外のファイル (dcllog ファイルなど) への書き込み処理では、ディスクに対するフラッシュ処理は行っていません。

● OpenTP1 ファイルシステムからの読み込み処理の性能

- 通常の Windows のファイルの場合
Windows のシステムキャッシュを介して、読み込み処理が行われます。
- raw I/O 機能を適用した場合
読み込みごとに、ディスクからの読み込み処理が行われます。

読み込み処理では、通常の Windows のファイルからの読み込みの方が、raw I/O 機能を適用した領域からの読み込みよりも性能が優れています。

(2) OpenTP1 ファイルシステム選択時の留意点

OpenTP1 ファイルシステムを選択する際は、次のことに留意してください。

● アンロードやバックアップの処理

ジャーナルファイルのアンロードや OpenTP1 ファイルのバックアップでは、読み込み処理が多く発生します。また、ジャーナルファイルやバックアップ対象ファイルに対する処理では、raw I/O 機能を適用した場合と比べて、通常の Windows のファイルの場合の方がコマンド実行時間を短縮できます。
アンロードやバックアップの処理性能を重視する場合は、通常の Windows のファイル上に OpenTP1 ファイルシステムを構築することをお勧めします。

● OpenTP1 がオンラインの場合の I/O 処理

OpenTP1 がオンラインの場合、次に示すファイルシステムへの I/O 処理は、読み込み処理よりも書き込み処理の方が多く発生します。

- システムジャーナルファイル
- チェックポイントダンプファイル
- ステータスファイル
- サーバリカバリジャーナルファイル
- XAR ファイル

オンライン性能を重視する場合は、これらの OpenTP1 ファイルシステムを raw I/O 機能を適用した領域上に構築することをお勧めします。

● OpenTP1 の再開処理

OpenTP1 の再開処理では、ジャーナルなどのデータ読み込み処理が発生するため、通常の Windows のファイルの場合の方が再開処理時間を短縮できることがあります。

● キャッシュブロックでの処理

次の処理では、通常の Windows のファイルの場合の方が、raw I/O 機能を適用した場合よりも性能が優れています。

- キャッシュブロックに確保されていないブロックの読み込み
- キャッシュブロックの再利用と解放の頻度が高い処理

次の処理では、通常の Windows のファイルの場合と raw I/O 機能を適用した場合で、性能に差はありません。

- キャッシュブロックに確保されているブロックの読み込み

● DAM サービスでの I/O 処理

dc_dam_rewrite 関数の発行が多く発生する処理では、raw I/O 機能を適用した場合の方が、通常の Windows のファイルの場合よりも性能が優れています。

DAM サービス定義の dam_update_block_over オペランドに flush を指定した場合、dc_dam_write 関数では読み込み処理が発生します。dc_dam_write 関数で指定されたブロックがキャッシュブロックに確保されている場合は、ディスクからの読み込みは発生しません。

キャッシュブロックの再利用と解放の頻度が高い処理では、raw I/O 機能を適用した場合の書き込み性能が優れています。このため、raw I/O 機能を適用した場合の方が、通常の Windows のファイルの場合よりも dc_dam_write 関数の性能が優れています。ただし、dam_update_block_over オペランドに flush を指定しない場合と比べると、通常の Windows のファイルに対する性能は劣ります。

dc_dam_write 関数と dc_dam_rewrite 関数の性能を重視する場合は、DAM ファイルを raw I/O 機能を適用した領域上に構築し、dc_dam_read 関数の性能を重視する場合は、DAM ファイルを通常の Windows のファイル上に構築することをお勧めします。

● TAM サービスでの I/O 処理

TAM サービスでの I/O 処理は、主に共用メモリ上の TAM テーブルで行われます。

TAM ファイルへの実 I/O 処理が行われる場合を次に示します。

- TAM テーブルのローディング時
- TAM テーブルのアンロード時
- 定間隔での実更新
- チェックポイントダンプ取得タイミングでの実更新

どの場合でも、I/O 処理は、ユーザサーバの処理とは非同期に行われ、ユーザサーバの TAM アクセスが I/O 処理を待ち合わせることはありません。したがって、dc_tam_read 関数、dc_tam_rewrite 関数、および dc_tam_write 関数の処理については、通常の Windows のファイルの場合と raw I/O 機能を適用した場合で、性能に差はありません。

! 注意事項

通常の Windows のファイルの場合と raw I/O 機能を適用した場合の性能差は、使用するディスク装置によって異なります。ディスク装置によっては、性能差がほとんどない場合もありますので、注意してください。

(3) OpenTP1 ファイルシステムの注意事項

Windows 版 OpenTP1 の OpenTP1 ファイルシステムについての注意事項を次に示します。

4. 環境設定

● OpenTP1 ファイルシステムの構築場所

OpenTP1 ファイルシステムは、ネットワークドライブ上に構築しないでください。ディスクへの書き込みを保証するため、ローカルディスク、または共有ディスク上に構築してください。

● OpenTP1 ファイルの保護

Windows 版 OpenTP1 では、OpenTP1 ファイルの保護はサポートしていません。

4.5.2 raw I/O 機能を適用した OpenTP1 ファイルシステムの構築

ここでは、raw I/O 機能を適用した OpenTP1 ファイルシステムの構築方法について説明します。

Windows 版 OpenTP1 では、Windows のダイレクトディスクアクセス (raw I/O) を適用して OpenTP1 ファイルシステム領域を作成できます。この機能を **raw I/O 機能** といいます。raw I/O 機能を適用すれば、パーティションまたは論理ドライブに対してファイルと同様にアクセスできます。

raw I/O 機能は、次の用途で使用できます。

- OpenTP1 ファイルシステムの構築
- OpenTP1 ファイルシステムのリストア

(1) OpenTP1 ファイルシステムの構築手順

raw I/O 機能を適用した領域に OpenTP1 ファイルシステムを構築する手順を次に示します。

1. 未フォーマット状態のパーティションを用意します。

Windows の [ディスクの管理] でパーティションを作成します。パーティションの作成方法については、Windows の [ディスクの管理] のヘルプを参照してください。

2. filmkfs コマンドを実行して、用意したパーティションまたは論理ドライブを、OpenTP1 ファイルシステムとして使用できる状態に初期設定します。

filmkfs コマンドの実行例を次に示します。

```
filmkfs -s 512 -n 7 -l 100 X:
```

この例では、X ドライブを raw I/O 機能を適用したパーティション (raw I/O パーティション) としています。

filmkfs コマンドについては、「7. 運用コマンド」を参照してください。

raw I/O 機能を適用した領域に OpenTP1 ファイルシステムを構築する場合の注意事項を次に示します。

パーティションを用意するときの注意事項

パーティションには、ドライブ文字を割り当ててください。

filmkfs コマンドを実行するときの注意事項

raw I/O 機能を適用する場合、filmkfs コマンドで必ず `-s` オプションを指定してください。

(2) OpenTP1 ファイルの指定

raw I/O パーティションとして割り当てた OpenTP1 ファイルシステム領域は、「ドライブ文字:」の形式で表されます。OpenTP1 ファイルを指定する場合は、通常の Windows のファイルと同様に絶対パスで指定してください。

OpenTP1 ファイルの指定例を次に示します。この例では、X ドライブを raw I/O パーティションに割り当てています。

- システムジャーナルファイルの作成例

```
jnlinit -j jnl -f X:¥jnlf01 -n 1024
```

- システムジャーナルファイルの指定例

```
jnladdpf -g jnlgrp01 -a X:¥jnlf01
```

(3) OpenTP1 ファイルシステムの削除

raw I/O パーティションとして構築した OpenTP1 ファイルシステムを完全に削除する場合は、Windows の「ディスクの管理」を使用します。パーティションの削除方法については、Windows の「ディスクの管理」のヘルプを参照してください。

(4) 系切り替え機能使用時の注意事項

raw I/O パーティションとして構築した OpenTP1 ファイルシステムを系切り替え機能で使用できません。

4.5.3 サポートディスク形式

OpenTP1 ファイルシステムを構築できるディスクを次の表に示します。

表にないディスクについては非対応です。

表 4-2 OpenTP1 ファイルシステムを構築できるディスク

パーティションスタイル	ディスクタイプ	フォーマット	サポート状況
MBR	ベーシック	NTFS	○
		FAT	○

4. 環境設定

パーティションスタイル	ディスクタイプ	フォーマット	サポート状況
		FAT32	○
		未フォーマット (rawI/O)	○
	ダイナミック	NTFS	○
		FAT	○
		FAT32	○
		未フォーマット (rawI/O)	○
GPT	ベーシック	NTFS	○
		未フォーマット (rawI/O)	○
	ダイナミック	NTFS	○
		未フォーマット (rawI/O)	×

(凡例)

- ：サポートしています。
- ×：サポートしていません。

4.6 データベース連携時の設定

Windows 版 OpenTP1 では、次のデータベースと連携できます。

- HiRDB
- Oracle
- SQL Server

ここでは、Windows 版 OpenTP1 で各データベースと連携するための設定について説明します。

4.6.1 HiRDB との連携

ここでは、HiRDB と連携するために必要な設定について説明します。

(1) HiRDB の登録

OpenTP1 システムに HiRDB を登録します。OpenTP1 システムへの HiRDB の登録には、trnlnkrm コマンドを使用します。

HiRDB Version 9 の場合の trnlnkrm コマンドの実行例を次に示します。

```
trnlnkrm -a HiRDB_DB_SERVER -s pdtxa_switch -o pdcltx32.lib
```

trnlnkrm コマンドについては、「7. 運用コマンド」を参照してください。また、trnlnkrm コマンドに指定する「RM 名」、「RM スイッチ名」、「RM 関連オブジェクト名」などについては、マニュアル「スケーラブルデータベースサーバ HiRDB システム導入・設計ガイド」を参照してください。

trnlnkrm コマンドを実行する場合の注意事項を次に示します。

● trnlnkrm コマンドの -o オプションの指定について

事前に、環境変数 LIB に関連オブジェクトのパスを設定しておいてください。また、関連オブジェクト名称をフォルダ付きの絶対パスで指定する場合は、フォルダの区切り文字は「¥¥」としてください。

● prcsvpath 定義コマンドの指定について

OpenTP1 システムに HiRDB を登録すると、トランザクションサービスプロセスが HiRDB のランタイムライブラリを使用します。このため、プロセスサービス定義の prcsvpath 定義コマンドで、HiRDB のランタイムライブラリの格納フォルダを指定してください。prcsvpath 定義コマンドの定義例を次に示します。

```
prcsvpath c:¥hirdb_s¥client¥utl
```

4. 環境設定

(2) 環境変数の設定

HiRDB にアクセスするために、HiRDB の環境変数に値を設定します。環境変数は、次に示す OpenTP1 のシステム定義に `putenv` 形式で設定します。

- トランザクションサービス定義
- ユーザサービス定義
- ユーザサービスデフォルト定義

トランザクションサービス定義の定義例を次に示します。

```
putenv PDHOST host1
putenv PDNAMEPORT 22200
putenv PDTMID smpl
putenv PDSWAITTIME 6000
```

ユーザサービス定義またはユーザサービスデフォルト定義の定義例を次に示します。

```
putenv PDHOST host1
putenv PDNAMEPORT 22200
putenv PDTMID smpl
putenv PDSWAITTIME 6000
putenv PDUSER '"root"/"root"'
putenv PDXAMODE 1
```

! 注意事項

X/Open の XA インタフェースを使用して OpenTP1 からデータベースにアクセスする場合に発生する障害の多くは、環境変数（PDHOST や PDNAMEPORT など）の設定の誤りによって発生します。KFCA00901-W メッセージが出力された場合には、環境変数の設定を見直してください。

OpenTP1 のシステム定義については、「5. システム定義」を参照してください。設定が必要な環境変数名や設定値などについては、HiRDB のインストール時の設定値や、マニュアル「スケーラブルデータベースサーバ HiRDB UAP 開発ガイド」を参照してください。

(3) トランザクションサービス定義の設定

トランザクションサービス定義の `trnstring` 定義コマンドで、リソースマネージャにアクセスするための情報を指定します。HiRDB と連携する場合は、`trnstring` 定義コマンドの `-n` オプション、および必要に応じて `-d` オプションを指定してください。

なお、`xa_open` 関数用文字列、および `xa_close` 関数用文字列（`trnstring` 定義コマンドの `-o` オプション、`-O` オプション、`-c` オプション、および `-C` オプションで指定）については、指定する必要はありません。

トランザクションサービス定義の定義例を次に示します。


```
trnstring -n HiRDB_DB_SERVER
```

(4) UAP の翻訳と結合

HiRDB にアクセスする UAP を翻訳（コンパイル）および結合（リンケージ）する場合、trnmkobj コマンドで %DCDIR%\¥spool¥trnrmcmd¥userobj 下に作成したトランザクション制御用オブジェクトをリンケージする必要があります。

コンパイルとリンケージの手順については、「3.2 UAP の翻訳と結合」、マニュアル「OpenTP1 プログラム作成リファレンス」の該当する言語編、およびマニュアル「スケーラブルデータベースサーバ HiRDB UAP 開発ガイド」を参照してください。trnmkobj コマンドについては、「7. 運用コマンド」を参照してください。

HiRDB にアクセスする UAP 用トランザクション制御用オブジェクトの作成例を、C 言語の場合と COBOL 言語の場合に分けて次に示します。

C 言語の場合

```
trnmkobj -o rm_obj -r HiRDB_DB_SERVER
```

COBOL 言語の場合

```
trnmkobj -o rm_obj -C "/Z1" -r HiRDB_DB_SERVER
```

(5) メイクファイルの例（C 言語の場合）

C 言語で SPP を作成する場合のメイクファイルの例を示します。

(a) ソースファイル名

メイクファイルの例中で使用するソースファイル名を次に示します。

● C 言語で作成した UAP のソースプログラム名：

- exmain.c（メイン関数）
- exsv1.c（サービス関数 1）
- exsv2.c（サービス関数 2）
- exproc.ec（ec ファイル）

● RPC インタフェース定義ファイル名：ex.def

(b) メイクファイルの例

メイクファイルの例を次に示します。

4. 環境設定

```
NODEBUG = 1
!include <ntwin32.mak>
#TP1TEST Make File
SYSLIB = $(conlibsdl)
TP1LIB = [libdam.lib]※1 [libtam.lib]※2 libbetran.lib
HIRDBLIB = pdcltx32.lib※4

all:TP1TEST.exe
#コンパイル
exmain.obj:exmain.c
$(cc) $(cflags) $(cvarsdll) $*.c
exsv1.obj:exsv1.c
$(cc) $(cflags) $(cvarsdll) $*.c
exsv2.obj:exsv2.c
$(cc) $(cflags) $(cvarsdll) $*.c
ex_sstb.obj:ex_sstb.c
$(cc) $(cflags) $(cvarsdll) $*.c
exproc.obj:exproc.c
$(cc) $(cflags) $(cvarsdll) $*.c
#スタブソースファイルの作成
ex_sstb.c:ex.def
$(DCDIR)¥bin¥stbmake $?
#プリコンパイルの実行
exproc.c:exproc.ec
$(PDDIR)¥client¥utl¥pdcp /XA $?※5
#リンケージ
TP1TEST.exe: exmain.obj exsv1.obj exsv2.obj ex_sstb.obj exproc.obj
$(link) $(conflags) -out:$*.exe $** ¥
[ $(DCDIR)¥spool¥trnmcmd¥userobj¥rm_obj.obj ¥]※3
$(TP1LIB) $(SYSLIB) $(HIRDBLIB)
```

注※ 1

libdam.lib は、TP1/FS/Direct Access を使用する場合だけ指定してください。

注※ 2

libtam.lib は、TP1/FS/Table Access を使用する場合だけ指定してください。

注※ 3

TP1/FS/Direct Access または TP1/FS/Table Access を使用する場合には、trnmkobj コマンドでトランザクション制御用オブジェクトを作成するときに、「OpenTP1_DAM」または「OpenTP1_TAM」を追加してください。

注※ 4

TP1/Server Base(64) を使用する場合、ライブラリ名称は pdcltx64.lib になります。

注※ 5

TP1/Server Base(64) を使用する場合、pdcp コマンドのオプションに「/h64」を追加してください。

(6) メイクファイルの例 (COBOL 言語の場合)

COBOL 言語で SPP を作成する場合のメイクファイルの例を示します。

(a) ソースファイル名

メイクファイルの例中で使用するソースファイル名を次に示します。

- COBOL2002 で作成した UAP のソースプログラム名：main.cbl

- COBOL2002 で作成した UAP のサービスプログラム名：sev1.cbl
- COBOL2002 で作成した UAP のサービスプログラム名：sev2.cbl
- ecb ファイル名 1：exprocb1.ecb
- ecb ファイル名 2：exprocb2.ecb
- RPC インタフェース定義ファイル名：ex.def

(b) メイクファイルの例

メイクファイルの例を次に示します。

```
!include <ntwin32.mak>
#TP1TEST Make File
CBLFG1 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC -Main,System
CBLFG2 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC
SYSLIB = $(conlibsdl)
TP1LIB = [libdam.lib]※1 [libtam.lib]※2 libbetran.lib
HIRDBLIB = pdcltx32.lib※4
CCBL = ccbl2002
LK = ccbl2002 -Lib,CUI

all:TP1TEST.exe
#コンパイル
main.obj:main.cbl
    $(CCBL) $(CBLFG1) main.cbl
sev1.obj:sev1.cbl
    $(CCBL) $(CBLFG2) sev1.cbl
sev2.obj:sev2.cbl
    $(CCBL) $(CBLFG2) sev2.cbl
exprocb1.obj:exprocb1.cbl
    $(CCBL) $(CBLFG2) exprocb1.cbl
exprocb2.obj:exprocb2.cbl
    $(CCBL) $(CBLFG2) exprocb2.cbl
ex_sstb.obj:ex_sstb.c
    $(cc) $(cflags) /Zl $(cvarsdll) *.c
#スタブソースファイルの作成
ex_sstb.c:ex.def
    $(DCDIR)¥bin¥stbmake $?
#プリコンパイルの実行
exprocb1.cbl:exprocb1.ecb
    $(PDDIR)¥client¥utl¥pdcb1 /XA $?※5
exprocb2.cbl:exprocb2.ecb
    $(PDDIR)¥client¥utl¥pdcb1 /XA $?※5
#リンケージ
TP1TEST.exe: main.obj sev1.obj sev2.obj ex_sstb.obj ¥
    exprocb1.obj exprocb2.obj
    $(LK) -OutputFile *.exe $** ¥
    [$(DCDIR)¥spool¥trnmcmd¥userobj¥rm_obj.obj ¥]※3
    $(TP1LIB) $(SYSLIB) $(HIRDBLIB)
```

注※ 1

libdam.lib は、TP1/FS/Direct Access を使用する場合だけ指定してください。

注※ 2

libtam.lib は、TP1/FS/Table Access を使用する場合だけ指定してください。

注※ 3

TP1/FS/Direct Access または TP1/FS/Table Access を使用する場合には、trnmkobj コマンドでトランザクション制御用オブジェクトを作成するときに、「OpenTP1_DAM」または

4. 環境設定

「OpenTP1_TAM」を追加してください。

注※ 4

TP1/Server Base(64)を使用する場合、ライブラリ名称は pdcltx64.lib になります。

注※ 5

TP1/Server Base(64)を使用する場合、pdcpp コマンドのオプションに「/h64」を追加してください。

(7) HiRDB との連携時の注意事項

- OpenTP1 と HiRDB を X/Open の XA インタフェースで連携する場合、HiRDB の関連サービスがすべて起動されていることを確認してください。
- OpenTP1 と HiRDB を X/Open の XA インタフェースで連携する場合、トランザクションの開始、および同期点取得は、次に示す OpenTP1 の関数を使用してください。
dc_trn_begin, tx_begin, dc_trn_unchained_commit, dc_trn_chained_commit, tx_commit, dc_trn_unchained_rollback, dc_trn_chained_rollback, tx_rollback
HiRDB の SQL 言語などでコミットやロールバックをした場合、または HiRDB が XA インタフェースで連携する場合に使用できない機能を使用すると、トランザクション処理が正常に実行されないで、リソース間の不整合が発生したり、OpenTP1 システムがダウンしたりすることがあります。

4.6.2 Oracle との連携

ここでは、Oracle と連携するために必要な設定について説明します。

(1) Oracle の登録

OpenTP1 システムに Oracle を登録します。OpenTP1 システムへの Oracle の登録には、trnlnkrm コマンドを使用します。

Oracle Database 12c の場合の trnlnkrm コマンドの実行例を次に示します。

```
trnlnkrm -a Oracle_XA -s xaosw -o oraxa12.lib
```

trnlnkrm コマンドについては、「7. 運用コマンド」を参照してください。また、trnlnkrm コマンドに指定する「RM 名」、「RM スイッチ名」、「RM 関連オブジェクト名」などについては、Oracle のマニュアルを参照してください。

trnlnkrm コマンドを実行する場合の注意事項を次に示します。

- trnlnkrm コマンドの -o オプションの指定について
事前に、環境変数 LIB に関連オブジェクトのパスを設定しておいてください。また、関連オブジェクト名称をフォルダ付きの絶対パスで指定する場合は、フォルダの区切り文字は「¥¥」としてください。

● prcsvpath 定義コマンドの指定について

OpenTP1 システムに Oracle を登録すると、トランザクションサービスプロセスが Oracle のランタイムライブラリを使用します。このため、プロセスサービス定義の prcsvpath 定義コマンドで、Oracle のランタイムライブラリの格納フォルダを指定してください。prcsvpath 定義コマンドの定義例を次に示します。

```
prcsvpath c:¥orant¥bin
```

(2) 環境変数の設定

Oracle にアクセスするために、Oracle の環境変数に値を設定します。環境変数は、次に示す OpenTP1 のシステム定義に putenv 形式で設定します。

- トランザクションサービス定義
- ユーザサービス定義
- ユーザサービスデフォルト定義

これらの定義例を次に示します。

```
putenv ORACLE_HOME c:¥orant
putenv ORACLE_SID ORCL
```

OpenTP1 のシステム定義については、「5. システム定義」を参照してください。設定が必要な環境変数名や設定値などについては、Oracle のインストール時の設定値や、Oracle のマニュアルを参照してください。

(3) トランザクションサービス定義の設定

トランザクションサービス定義の trnstring 定義コマンドで、リソースマネージャにアクセスするための情報を指定します。Oracle と連携する場合は、trnstring 定義コマンドの -n オプション、-o オプション、-O オプション、および必要に応じて -d オプションを指定してください。

-o オプション、および -O オプションには、リソースマネージャへのアクセスに使用する xa_open 関数用文字列を指定します。なお、xa_close 関数用文字列（trnstring 定義コマンドの -c オプション、および -C オプションで指定）については、指定する必要はありません。

-o オプションで指定するトランザクションサービス用 xa_open 関数用文字列の uid には、Oracle の DB 管理者を指定してください。指定した uid が Oracle の DB 管理者でなかった場合、または DBA 権限を持っていない場合には、UAP の部分回復、および OpenTP1 システム再開始後のトランザクション回復処理が正常に行われなかったことがあります。Oracle の xa_open 関数用文字列として指定する内容の詳細については、Oracle のマニュアルを参照してください。

トランザクションサービス定義の定義例を次に示します。

4. 環境設定

```
trnstring -n Oracle_XA ¥  
-o "Oracle_XA+Acc=P/sys/change_on_install+SesTm=60" ¥  
-o "Oracle_XA+Acc=P/scott/tiger+SesTm=60"
```

注

""で囲んだ文字列を継続行マーク（¥）で区切って改行する場合は、次の行は必ず第1カラム目から記述し、スペースを開けないで文字列を続けてください。

！ 注意事項

X/Open の XA インタフェースを使用して OpenTP1 から Oracle にアクセスする場合に発生する障害の多くは、xa_open 関数用文字列の定義誤りによって発生します。KFCA00901-W メッセージが出力された場合には、xa_open 関数用文字列の定義を見直してください。

(4) 複数の Oracle にアクセスする場合の設定

OpenTP1 から複数の Oracle にアクセスする場合の設定について説明します。なお、複数の Oracle の uid で一つの Oracle にアクセスする場合も、複数の Oracle にアクセスする場合と同様の設定をしてください。

● SQL*Net のセットアップ

OpenTP1 から複数の Oracle にアクセスする場合には、Oracle のオープン、クローズ文字列指定の仕様によって、SQL*Net が必要となります。Oracle のマニュアルを参照して、SQL*Net のセットアップを行ってください。

● xa_open 関数用文字列の定義

OpenTP1 から複数の Oracle にアクセスする場合には、リソースマネージャへのアクセスに使用する xa_open 関数用文字列の定義方法が異なります。定義方法を次に示します。

1. トランザクションサービス定義の trnstring 定義コマンドの -i オプションを定義して、「リソースマネージャ名称+リソースマネージャ拡張子」によって、複数の Oracle を識別できるようにします。
2. 1. で定義したリソースマネージャ拡張子を、ユーザーサービス定義、またはユーザーサービスデフォルト定義の trnrmid 定義コマンドの -i オプションで定義します。
トランザクションサービス定義の定義例を次に示します。

```
trnstring -n Oracle_XA -i s1 ¥  
-o "Oracle_XA+Acc=P/sys/change_on_install+SesTm=60" ¥  
-o "Oracle_XA+Acc=P/scott/tiger+SesTm=60"  
trnstring -n Oracle_XA -i s2 ¥  
-o "Oracle_XA+SqlNet=host01+Acc=P/sys/change_on_install+SesTm=60"¥  
-o "Oracle_XA+SqlNet=host01+Acc=P/scott/tiger+SesTm=60"
```

ユーザーサービスデフォルト定義の定義例を次に示します。

```
trnrmid -n Oracle_XA -i s1
```

ユーザサービス定義の定義例を次に示します。

```
trnrmid -n Oracle_XA -i s2
```

注

trnrmid 定義コマンドは、リソースマネージャ拡張子ごとに独立して定義が有効になります。したがって、ユーザサービスデフォルト定義に s1 を、ユーザサービス定義に s2 を指定した場合、ユーザサービス定義では s1 と s2 の両方が有効になっています。

(5) UAP の翻訳と結合

Oracle にアクセスする UAP を翻訳（コンパイル）および結合（リンケージ）する場合、trnmkobj コマンドで %DCDIR%\\$pool¥trnrmcmd¥userobj 下に作成したトランザクション制御用オブジェクトをリンケージする必要があります。

コンパイルとリンケージの手順については、「3.2 UAP の翻訳と結合」、マニュアル「OpenTP1 プログラム作成リファレンス」の該当する言語編、および Oracle のマニュアルを参照してください。trnmkobj コマンドについては、「7. 運用コマンド」を参照してください。

Oracle にアクセスする UAP 用トランザクション制御用オブジェクトの作成例を、C 言語の場合と COBOL 言語の場合に分けて次に示します。

C 言語の場合

```
trnmkobj -o rm_obj -r Oracle_XA
```

COBOL 言語の場合

```
trnmkobj -o rm_obj -C "/Z1" -r Oracle_XA
```

(6) メイクファイルの例（C 言語の場合）

C 言語で SPP を作成する場合のメイクファイルの例を示します。

(a) ソースファイル名

メイクファイルの例中で使用するソースファイル名を次に示します。

● C 言語で作成した UAP のソースプログラム名：

- exmain.c（メイン関数）
- exsv1.c（サービス関数 1）
- exsv2.c（サービス関数 2）
- exproc.pc（pc ファイル）

● RPC インタフェース定義ファイル名：ex.def

4. 環境設定

(b) メイクファイルの例

メイクファイルの例を次に示します。

```
NODEBUG = 1
!include <ntwin32.mak>
#TP1TEST Make File
SYSLIB = $(conlibsdl)
TP1LIB = [libdam.lib]※1 [libtam.lib]※2 libbetran.lib
ORACLELIB = oraxa12.lib

all:TP1TEST.exe
#コンパイル
exmain.obj:exmain.c
$(cc) $(cflags) $(cvarsdll) $*.c
exsv1.obj:exsv1.c
$(cc) $(cflags) $(cvarsdll) $*.c
exsv2.obj:exsv2.c
$(cc) $(cflags) $(cvarsdll) $*.c
ex_sstb.obj:ex_sstb.c
$(cc) $(cflags) $(cvarsdll) $*.c
exproc.obj:exproc.c
$(cc) $(cflags) $(cvarsdll) $*.c
#スタブソースファイルの作成
ex_sstb.c:ex.def
$(DCDIR)¥bin¥stbmake $?
#ブリコンパイルの実行
exproc.c:exproc.pc
$(ORACLE_HOME)¥bin¥proc release_cursor=yes $?
#リンケージ
TP1TEST.exe: exmain.obj exsv1.obj exsv2.obj ex_sstb.obj exproc.obj
$(link) $(conflags) -out:$*.exe $** ¥
[ $(DCDIR)¥spool¥trnmcmd¥userobj¥rm_obj.obj ¥]※3
$(TP1LIB) $(SYSLIB) $(ORACLELIB)
```

注※ 1

libdam.lib は、TP1/FS/Direct Access を使用する場合だけ指定してください。

注※ 2

libtam.lib は、TP1/FS/Table Access を使用する場合だけ指定してください。

注※ 3

TP1/FS/Direct Access または TP1/FS/Table Access を使用する場合には、trnmkobj コマンドでトランザクション制御用オブジェクトを作成するときに、「OpenTP1_DAM」または「OpenTP1_TAM」を追加してください。

(7) メイクファイルの例（COBOL 言語の場合）

COBOL 言語で SPP を作成する場合のメイクファイルの例を示します。

(a) ソースファイル名

メイクファイルの例中で使用するソースファイル名を次に示します。

- COBOL2002 で作成した UAP のソースプログラム名：main.cbl
- COBOL2002 で作成した UAP のサービスプログラム名：sev1.cbl
- COBOL2002 で作成した UAP のサービスプログラム名：sev2.cbl

- pco ファイル名 1 : exprocb1.pco
- pco ファイル名 2 : exprocb2.pco
- RPC インタフェース定義ファイル名 : ex.def

(b) メイクファイルの例

メイクファイルの例を次に示します。

```
!include <ntwin32.mak>
#TP1TEST Make File
CBLFG1 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC -Main,System
CBLFG2 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]※1 [libtam.lib]※2 libbetran.lib
ORACLELIB = oraxa12.lib
CCBL = ccbl2002
LK = ccbl2002 -Lib,CUI

all:TP1TEST.exe
#コンパイル
main.obj:main.cbl
    $(CCBL) $(CBLFG1) main.cbl
sev1.obj:sev1.cbl
    $(CCBL) $(CBLFG2) sev1.cbl
sev2.obj:sev2.cbl
    $(CCBL) $(CBLFG2) sev2.cbl
exprocb1.obj:exprocb1.cbl
    $(CCBL) $(CBLFG2) exprocb1.cbl
exprocb2.obj:exprocb2.cbl
    $(CCBL) $(CBLFG2) exprocb2.cbl
ex_sstb.obj:ex_sstb.c
    $(cc) $(cflags) /Zl $(cvarsdll) *.c
#スタブソースファイルの作成
ex_sstb.c:ex.def
    $(DCDIR)¥bin¥stbmake $?
#プリコンパイルの実行
exprocb1.cbl:exprocb1.pco
    $(ORACLE_HOME)¥bin¥procob release_cursor=yes $?
exprocb2.cbl:exprocb2.pco
    $(ORACLE_HOME)¥bin¥procob release_cursor=yes $?
#リンケージ
TP1TEST.exe: main.obj sev1.obj sev2.obj ex_sstb.obj ¥
    exprocb1.obj exprocb2.obj
    $(LK) -OutputFile *.exe *** ¥
    [$(DCDIR)¥spool¥trnmcmd¥userobj¥rm_obj.obj ¥]※3
    $(TP1LIB) $(SYSLIB) $(ORACLELIB)
```

注※ 1

libdam.lib は、TP1/FS/Direct Access を使用する場合だけ指定してください。

注※ 2

libtam.lib は、TP1/FS/Table Access を使用する場合だけ指定してください。

注※ 3

TP1/FS/Direct Access または TP1/FS/Table Access を使用する場合には、trnmkobj コマンドでトランザクション制御用オブジェクトを作成するときに、「OpenTP1_DAM」または「OpenTP1_TAM」を追加してください。

(8) Oracle との連携時の注意事項

- OpenTP1 と Oracle を X/Open の XA インタフェースで連携する場合、Oracle の関連サービスがすべて起動されていることを確認してください。
- OpenTP1 と Oracle を X/Open の XA インタフェースで連携する場合、トランザクションの開始、および同期点取得は、次に示す OpenTP1 の関数を使用してください。
dc_trn_begin, tx_begin, dc_trn_unchained_commit, dc_trn_chained_commit, tx_commit, dc_trn_unchained_rollback, dc_trn_chained_rollback, tx_rollback
Oracle の SQL 言語などでコミットやロールバックをした場合、または Oracle が XA インタフェースで連携する場合に使用できない機能を使用すると、トランザクション処理が正常に実行されないで、リソース間の不整合が発生したり、OpenTP1 システムがダウンしたりすることがあります。
- Oracle のトラブルシュート情報格納フォルダには、OpenTP1 のサービスログオンアカウントのユーザが書き込みできるアクセス権を設定してください。書き込み権限がない場合、OpenTP1 のサーバプロセスで発行した XA 関数のトラブルシュート情報を格納できないことがあります。
- OpenTP1 と Oracle を X/Open の XA インタフェースで連携する場合、次のメッセージが出力されるときは、Oracle や通信プロセスなどが起動されているかを確認してください。
 - ・ トランザクション中に Oracle の通信プロセス (SQL*Net 機能の一部) が未起動の場合や障害が発生した場合に、KFCA00901-W メッセージが出力される。
 - ・ SQL のアクセスで Oracle の ORA-1012 メッセージが出力される。
- OpenTP1 と Oracle を X/Open の XA インタフェースで連携する場合、Oracle を XA インタフェースでアクセスするプロセス数が、Oracle の "SESSIONS" 定義の指定値を超えたときにも、KFCA00901-W メッセージが出力されます。その場合は、次に示す計算式を参考にして、Oracle の "SESSIONS" 定義を見直してください。

"SESSIONS"定義の最小値 $\geq$ Oracleアクセス基礎値 $\times n^{※1}$

Oracleアクセス基礎値

$\geq (\text{Oracleをアクセスするユーザサーバプロセスの総数} \\ + \text{並行回復プロセス数}^{※2} + 1^{※3}) \\ \times \text{Oracleの数}$

注※ 1

n は、「xa_open → xa_start → UOC(SQL) → xa_end → xa_close」までの処理に必要なセッション数 (通常は 1 ～ 5 程度) です。詳細については、Oracle の設定を確認してください。

注※ 2

並行回復プロセス数は、トランザクションサービス定義の trn_recovery_process_count オペランドの指定値です。

注※ 3

1 は、OpenTP1 のリソースマネージャ監視プロセスの数です。

- OpenTP1 と Oracle を X/Open の XA インタフェースで連携する場合、Oracle のオープン文字列に指定する「SesTm」は、Oracle のグローバルトランザクション監視時間です。「SesTm」に小さい値を指定した場合、指定した時間が経過すると、Oracle が OpenTP1 の指示を待たずにトランザクションを決着させることがあるため、ユーザサーバや OpenTP1 システムがダウンすることがあります。「SesTm」には、余裕を持たせた値を指定してください。
- OpenTP1 と Oracle を X/Open の XA インタフェースで連携する場合、トランザクション実行中に Oracle がダウンして再起動したときに、Oracle のダウン前にアクセスしていたプロセスからの要求を、Oracle が受け付けなくなることがあります。その場合は、KFCA00901-W、KFCA00905-E、または KFCA00906-E メッセージが出力されるため、該当するサーバを終了させて、再起動してください。該当するサーバが「_trnrcv」の場合は、OpenTP1 システムを終了させて、再起動してください。

4.6.3 SQL Server との連携

ここでは、SQL Server と連携するために必要な設定について説明します。

(1) SQL Server の登録

OpenTP1 システムに SQL Server を登録します。OpenTP1 システムへの SQL Server の登録には、trnlnkrm コマンドを使用します。

trnlnkrm コマンドの実行例を次に示します。

```
trnlnkrm -a MS_SQL_Server -s msqslrvxa1 -o xaswitch.lib
```

trnlnkrm コマンドについては、「7. 運用コマンド」を参照してください。また、trnlnkrm コマンドに指定する「RM 名」、「RM スイッチ名」、「RM 関連オブジェクト名」などについては、SQL Server のマニュアルを参照してください。

trnlnkrm コマンドを実行する場合の注意事項を次に示します。

- trnlnkrm コマンドの -o オプションの指定について
事前に、環境変数 LIB に関連オブジェクトのパスを設定しておいてください。また、関連オブジェクト名称をフォルダ付きの絶対パスで指定する場合は、フォルダの区切り文字は「¥¥」としてください。
- prcsvpath 定義コマンドの指定について
OpenTP1 システムに SQL Server を登録すると、トランザクションサービスプロセスが SQL Server のランタイムライブラリを使用します。このため、プロセスサービス定義の prcsvpath 定義コマンドで、SQL Server のランタイムライブラリの格納フォルダを指定してください。prcsvpath 定義コマンドの定義例を次に示します。

```
prcsvpath c:%mssql%binn
```

(2) トランザクションサービス定義の設定

トランザクションサービス定義の `trnstring` 定義コマンドで、リソースマネージャにアクセスするための情報を指定します。SQL Server と連携する場合は、`-n` オプション、`-o` オプション、`-O` オプション、`-c` オプション、`-C` オプション、`-` および必要に応じて `-d` オプションを指定します。

`-o` オプション、`-O` オプション、`-c` オプション、および `-C` オプションには、リソースマネージャへのアクセスに使用する `xa_open` 関数用文字列、および `xa_close` 関数用文字列を指定します。`xa_open` 関数用文字列、および `xa_close` 関数用文字列は次のように指定してください。

- `-o` オプション、`-O` オプション、`-c` オプション、および `-C` オプションには、すべて同じ値を指定します。
- `RmRecoveryGuid` には、グローバルで一意的な値を指定します。Windows の `guidgen` ユーティリティプログラムの `Registry Format` で取得した値（括弧は省く）を指定します。
- `Timeout` には、XA トランザクションのタイムアウト値をミリ秒で指定します。指定したタイムアウト時間が経過すると、SQL Server（Windows の MSDTC）は XA トランザクションを自動的にロールバックします。

SQL Server の `xa_open` 関数用文字列、および `xa_close` 関数用文字列として指定する内容の詳細については、SQL Server のマニュアルを参照してください。

トランザクションサービス定義の定義例を次に示します。

```
set trn_extend_function=00000001      #add when SQL Server is used.

trnstring -n MS_SQL_Server ¥
-o "Tm=OpenTP1,¥
RmRecoveryGuid=986D495E-F4D4-4e80-8B70-9377FFAF33E7,Timeout=180000" ¥
-O "Tm=OpenTP1,¥
RmRecoveryGuid=986D495E-F4D4-4e80-8B70-9377FFAF33E7,Timeout=180000" ¥
-c "Tm=OpenTP1,¥
RmRecoveryGuid=986D495E-F4D4-4e80-8B70-9377FFAF33E7,Timeout=180000" ¥
-C "Tm=OpenTP1,¥
RmRecoveryGuid=986D495E-F4D4-4e80-8B70-9377FFAF33E7,Timeout=180000"
```

注

"" で囲んだ文字列を継続行記号（¥）で区切って改行する場合は、次の行は必ず第 1 カラム目から記述し、スペースを開けないで文字列を続けてください。

！ 注意事項

X/Open の XA インタフェースを使用して OpenTP1 から SQL Server にアクセスする場合に発生する障害の多くは、xa_open 関数用文字列、および xa_close 関数用文字列の定義誤りによって発生します。KFCA00901-W メッセージが出力された場合には、xa_open 関数用文字列、および xa_close 関数用文字列の定義を見直してください。

(3) UAP の翻訳と結合

SQL Server にアクセスする UAP を翻訳（コンパイル）および結合（リンケージ）する場合、trnmkobj コマンドで %DCDIR%\¥spool¥trnrnmcmd¥userobj 下に作成したトランザクション制御用オブジェクトをリンケージする必要があります。

コンパイルとリンケージの手順については、「3.2 UAP の翻訳と結合」、マニュアル「OpenTP1 プログラム作成リファレンス」の該当する言語編、および SQL Server のマニュアルを参照してください。trnmkobj コマンドについては、「7. 運用コマンド」を参照してください。

SQL Server にアクセスする UAP 用トランザクション制御用オブジェクトの作成例を、C 言語の場合と COBOL 言語の場合に分けて次に示します。

C 言語の場合

```
trnmkobj -o rm_obj -r MS_SQL_Server
```

COBOL 言語の場合

```
trnmkobj -o rm_obj -C "/Z1" -r MS_SQL_Server
```

(4) メイクファイルの例（C 言語の場合）

C 言語で SPP を作成する場合のメイクファイルの例を示します。

(a) ソースファイル名

メイクファイルの例中で使用するソースファイル名を次に示します。

● C 言語で作成した UAP のソースプログラム名：

- exmain.c（メイン関数）
- exsv1.c（サービス関数 1）
- exsv2.c（サービス関数 2）
- exsql.c（SQL）

● RPC インタフェース定義ファイル名：ex.def

4. 環境設定

(b) メイクファイルの例

メイクファイルの例を次に示します。

```
NODEBUG = 1
!include <ntwin32.mak>
#TP1TEST Make File
SYSLIB = $(conlibsdl)
TP1LIB = [libdam.lib]*1 [libtam.lib]*2 libbetran.lib
SQLSRVLIB = xaswitch.lib odbc32.lib

all:TP1TEST.exe
#コンパイル
exmain.obj:exmain.c
$(cc) $(cflags) $(cvarsdll) *.c
exsv1.obj:exsv1.c
$(cc) $(cflags) $(cvarsdll) *.c
exsv2.obj:exsv2.c
$(cc) $(cflags) $(cvarsdll) *.c
ex_sstb.obj:ex_sstb.c
$(cc) $(cflags) $(cvarsdll) *.c
exsql.obj:exsql.c
$(cc) $(cflags) $(cvarsdll) *.c
#スタブソースファイルの作成
ex_sstb.c:ex.def
$(DCDIR)¥bin¥stbmake $?
#リンケージ
TP1TEST.exe: exmain.obj exsv1.obj exsv2.obj ex_sstb.obj exsql.obj
$(link) $(conflags) -out:*.exe *** ¥
[ $(DCDIR)¥spool¥trnrmcmd¥userobj¥rm_obj.obj ¥]*3
$(TP1LIB) $(SYSLIB) $(SQLSRVLIB)
```

注※ 1

libdam.lib は、TP1/FS/Direct Access を使用する場合だけ指定してください。

注※ 2

libtam.lib は、TP1/FS/Table Access を使用する場合だけ指定してください。

注※ 3

TP1/FS/Direct Access または TP1/FS/Table Access を使用する場合には、trnmkobj コマンドで
トランザクション制御用オブジェクトを作成するときに、「OpenTP1_DAM」または
「OpenTP1_TAM」を追加してください。

(5) メイクファイルの例（COBOL 言語の場合）

COBOL 言語で SPP を作成する場合のメイクファイルの例を示します。

(a) ソースファイル名

メイクファイルの例中で使用するソースファイル名を次に示します。

- COBOL2002 で作成した UAP のソースプログラム名：main.cbl
- COBOL2002 で作成した UAP のサービスプログラム名：sev1.cbl
- COBOL2002 で作成した UAP のサービスプログラム名：sev2.cbl
- COBOL2002 で作成した SQL プログラム名 1：exsql1.cbl

● COBOL2002 で作成した SQL プログラム名 2：exsql2.cbl

● RPC インタフェース定義ファイル名：ex.def

(b) メイクファイルの例

メイクファイルの例を次に示します。

```
!include <ntwin32.mak>
# TP1TEST Make File
CBLFG1 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC -Main,System
CBLFG2 = -Compile,NoLink -Comp5 -Lib,CUI -SQL,ODBC
SYSLIB = $(conlibsdll)
TP1LIB = [libdam.lib]※1 [libtam.lib]※2 libbetran.lib
SQLSRV0BJ = xaswitch.lib
CCBL = ccbl2002
LK = ccbl2002 -Lib,CUI

all:TP1TEST.exe
# コンパイル
main.obj:main.cbl
    $(CCBL) $(CBLFG1) main.cbl
sev1.obj:sev1.cbl
    $(CCBL) $(CBLFG2) sev1.cbl
sev2.obj:sev2.cbl
    $(CCBL) $(CBLFG2) sev2.cbl
exsql1.obj:exsql1.cbl
    $(CCBL) $(CBLFG2) exsql1.cbl
exsql2.obj:exsql2.cbl
    $(CCBL) $(CBLFG2) exsql2.cbl
ex_sstb.obj:ex_sstb.c
    $(cc) $(cflags) /Zl $(cvarsdll) $*.c
# スタブソースファイルの作成
ex_sstb.c:ex.def
    $(DCDIR)¥bin¥stbmake $?
# リンケージ
TP1TEST.exe: main.obj sev1.obj sev2.obj ex_sstb.obj ¥
    exsql1.obj exsql2.obj
    $(LK) -OutputFile $*.exe $** ¥
    [$(DCDIR)¥spool¥trnrmcmd¥userobj¥rm_obj.obj ¥]※3
    $(TP1LIB) $(SYSLIB) $(SQLSRV0BJ)
```

注※ 1

libdam.lib は、TP1/FS/Direct Access を使用する場合だけ指定してください。

注※ 2

libtam.lib は、TP1/FS/Table Access を使用する場合だけ指定してください。

注※ 3

TP1/FS/Direct Access または TP1/FS/Table Access を使用する場合には、trnmkobj コマンドでトランザクション制御用オブジェクトを作成するときに、「OpenTP1_DAM」または「OpenTP1_TAM」を追加してください。

(6) SQL Server との連携時の注意事項

● OpenTP1 と SQL Server を X/Open の XA インタフェースで連携する場合、SQL Server の関連サービスがすべて起動されていることを確認してください。

● OpenTP1 と SQL Server を X/Open の XA インタフェースで連携する場合、トランザクションの開始、および同期点取得は、次に示す OpenTP1 の関数を使用してください

4. 環境設定

い。

dc_trn_begin, tx_begin, dc_trn_unchained_commit, dc_trn_chained_commit, tx_commit, dc_trn_unchained_rollback, dc_trn_chained_rollback, tx_rollback
SQL Server の SQL 言語などでコミットやロールバックをした場合、または SQL Server が XA インタフェースで連携する場合に使用できない機能を使用すると、トランザクション処理が正常に実行されないで、リソース間の不整合が発生したり、OpenTP1 システムがダウンしたりすることがあります。

- OpenTP1 と SQL Server を X/Open の XA インタフェースで連携する場合の UAP 作成時の注意事項を次に示します。

- DB-Library によって XA インタフェースでの連携をする場合（UAP を C 言語または C++ 言語で作成）

OpenTP1 のトランザクションと SQL Server との接続を XA インタフェースで関連づけるために、各トランザクションブランチの先頭で関連づけを行う関数（dbenlistxatrans）を発行する必要があります。この関数が発行されていないと、OpenTP1 で管理しているトランザクションによる決着ができません。詳細については、SQL Server のマニュアルを参照してください。

- ODBC 接続によって XA インタフェースでの連携をする場合（UAP を C 言語または C++ 言語で作成）

OpenTP1 のトランザクションと SQL Server との接続を XA インタフェースで関連づけるために、各トランザクションブランチの先頭で関連づけを行う関数（SQLSetconnectOption）を発行する必要があります。この関数が発行されていないと、OpenTP1 で管理しているトランザクションによる決着ができません。詳細については、SQL Server のマニュアルを参照してください。

- ODBC 接続によって XA インタフェースでの連携をする場合（UAP を COBOL2002 で作成）

OpenTP1 のトランザクションと SQL Server との接続を XA インタフェースで関連づけるために、各トランザクションブランチの先頭で関連づけを行うサービスルーチン（COBOL2002 が提供する CBLSQLSETOPT）を発行する必要があります。このサービスルーチンが発行されていないと、OpenTP1 で管理しているトランザクションによる決着ができません。また、次の環境変数を SQL Server と XA インタフェースで連携するユーザサーバのユーザサービス定義に指定してください。

```
putenv CBLSQLCOMMOD AUTO
```

- OpenTP1 と SQL Server を X/Open の XA インタフェースで連携する場合、連鎖型 RPC の使用には注意してください。OpenTP1 のトランザクションと SQL Server との接続を XA インタフェースで関連づける三つの関数（dbenlistxatrans, SQLSetconnectOption, CBLSQLSETOPT）を 2 回目以降の RPC で発行しないようにする必要があります。

- OpenTP1 と SQL Server を X/Open の XA インタフェースで連携する場合、XA イン

タフェースのオープン文字列、クローズ文字列に Timeout の設定を行ってください。Timeout の設定を行わないと、UAP が異常終了した場合に SQL Server (Windows の MSDTC) 内で「SPID=-2」という状態のトランザクションが残り続けることがあります。

- OpenTP1 と SQL Server を X/Open の XA インタフェースで連携する場合、トランザクションブランチに関連するすべての OpenTP1 ノードで、トランザクションサービス定義に次の設定をしてください。

```
set trn_extend_function=00000001
```

なお、次に示す条件が重なった場合、OpenTP1 のトランザクション決着関数は、「DC_OK(0)」ではなく「DCTRNER_HAZARD(-904)」または「TX_HAZARD(-4)」を返すようになります。

- リソースマネージャへのアクセスが SQL Server しか存在しない (1 相コミットで決着を行う)
- Timeout の指定によって SQL Server がトランザクションを自動的にロールバックする

4.7 OS の設定

ここでは、Windows 版 OpenTP1 を使用するために必要な OS の設定について説明します。

(1) TCP/IP の設定

複数のネットワークインタフェースカード（NIC）を使用している環境では、TCP/IP プロトコルのバインド順に IP アドレスのリストが返却されます。OpenTP1 で使用する NIC が最優先となるように、バインド順を変更してください。

バインド順は、「コントロールパネル」の「ローカルエリア接続のバインド」で変更します。

(2) Windows ファイアウォールの設定

Windows ファイアウォールの設定を行う場合は、%DCDIR%\¥bin フォルダ、%DCDIR%\¥lib¥servers フォルダ、%DCDIR%\¥aplib フォルダに配置しているプログラムを例外リストに登録してください。

コマンドプロンプトから Windows 7 以降の netsh コマンドを使用した場合の実行例を次に示します。

```
for %p in ("%DCDIR%\¥bin¥*.exe") do (netsh advfirewall firewall add rule name="OpenTP1" dir=in action=allow program="%p" enable=yes profile=public)
for %p in ("%DCDIR%\¥lib¥servers¥*.exe") do (netsh advfirewall firewall add rule name="OpenTP1" dir=in action=allow program="%p" enable=yes profile=public)
for %p in ("%DCDIR%\¥aplib¥*.exe") do (netsh advfirewall firewall add rule name="OpenTP1 UAP" dir=in action=allow program="%p" enable=yes profile=public)
```

OS の違いで netsh コマンドの実行方法が異なる場合がありますので、netsh コマンドの詳細については、Windows のコマンドヘルプを参照してください。

Windows ファイアウォールの設定時の注意事項を次に示します。

- %DCDIR%\¥aplib 以外のフォルダに配置しているユーザサーバは、上記コマンドでは登録されないため、手作業で例外リストに登録する必要があります。
- trnlnkrm コマンド、rapsetup コマンド、および dcsetupml コマンドを実行した場合、プログラムが未登録状態になることがあります。これらのコマンドの実行後は、上記コマンドを再度実行してください。dcsetupml コマンドでマルチ OpenTP1 をセットアップした場合は、各マルチ OpenTP1 のコマンド実行用コンソールから上記コマンドを実行してください。
- 新たにユーザサーバを配置する場合、手作業で例外リストに登録する必要があります。
- 次のどちらかの方法で、例外リストにプログラムを登録してください。
 - ・「コントロールパネル」の「Windows ファイアウォール」の「Windows ファイア

ウォールを介したプログラムまたは機能を許可する」

- Windows の netsh コマンド

Windows 7 以降の netsh コマンドの実行例を次に示します。

```
netsh advfirewall firewall add rule name="OpenTP1 UAP" dir=in  
action=allowprogram=c:¥OpenTP1¥aplib¥sample.exe enable=yes profile=public
```

netsh コマンドの詳細については、Windows のコマンドヘルプを参照してください。

(3) コンピュータ名の設定

OpenTP1 が動作するコンピュータのフルコンピュータ名に「Administrators」という名称を設定しないでください。

4.8 マルチ OpenTP1 の設定

ここでは、Windows 版 OpenTP1 でのマルチ OpenTP1 の設定について、UNIX 版 OpenTP1 との差異を説明します。UNIX 版 OpenTP1 と共通の部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

! 注意事項

Windows 版 OpenTP1 では、マルチ OpenTP1 が示す範囲に、オリジナルの OpenTP1（インストーラでインストールした OpenTP1）を含みません。

(1) マルチ OpenTP1 環境の設定

マルチ OpenTP1 の設定方法を次に示します。

● セットアップ

dcsetupml コマンドを使用して、すべてのマルチ OpenTP1 環境をセットアップします。マルチ OpenTP1 をセットアップするためには、コマンドの引数に識別子を指定します。引数に指定する識別子は、レジストリ登録、サービス登録、およびイベントビューアへの登録で使います。識別子は半角英数字で 4 文字以内です。

dcsetupml コマンドの詳細については、「7. 運用コマンド」の「dcsetupml（マルチ OpenTP1 のセットアップとアンセットアップ）」を参照してください。

なお、マルチ OpenTP1 のセットアップ数の上限は、3 個までです。4 個以上のマルチ OpenTP1 をセットアップした場合の動作は保証しません。

● サービス登録

マルチ OpenTP1 では、dcsetupml コマンドでセットアップするときに指定した識別子をサービス名の末尾に付加します。例えば、識別子を「M001」とした場合、サービス名は「OpenTP1_M001」として登録されます。この場合、OpenTP1 コンソールのタイトルには「OpenTP1_M001 - OpenTP1 for Windows」と表示されます。

● イベントビューアへのメッセージ出力

マルチ OpenTP1 環境でイベントビューアにメッセージを出力する場合のソース名は、サービス名と同様に、dcsetupml コマンドでセットアップするときに指定した識別子を末尾に付加した文字列です。例えば、識別子を「M001」と指定した場合、ソース名は「OpenTP1_M001」となります。これによって、メッセージを出力した OpenTP1 を識別できます。

(2) マルチ OpenTP1 環境の再セットアップの手順

マルチ OpenTP1 環境の再セットアップの手順を次に示します。

1. すべてのマルチ OpenTP1 環境で、ファイルのバックアップを取得します。

バックアップを取得するファイルを次に示します。

- システムサービス情報定義ファイル (%DCDIR%\¥lib¥sysconf¥mcfu*)
- システムサービス共通情報定義ファイル (%DCDIR%\¥lib¥sysconf¥mcf)
- 通信サービス実行形式プログラム (%DCDIR%\¥lib¥servers¥mcfu*)
- マルチ OpenTP1 環境のセットアップフォルダ下で、ユーザが独自に変更または追加しているファイル

2. dcsetupml -d コマンドで、すべてのマルチ OpenTP1 環境を削除します。
3. 入れ替え対象のプログラムをインストールします。
4. dcsetupml コマンドで、すべてのマルチ OpenTP1 環境をセットアップします。
5. セットアップしたすべてのマルチ OpenTP1 環境で、1. でバックアップしたファイルを上書きコピーします。
6. インストール環境と連携するリソースマネージャ[※]が異なる場合、それぞれの環境に合わせて trnlnkrm コマンドを実行します。

注※

TP1/Message Control, TP1/FS/Direct Access, TP1/FS/Table Access, HiRDB, Oracleなどを指します。

5

システム定義

Windows 版 OpenTP1 は，UNIX 版 OpenTP1 と比べて，指定できる OpenTP1 のシステム定義が異なります。この章では，システム定義の一覧で，Windows 版 OpenTP1 と UNIX 版 OpenTP1 のシステム定義の差異について説明します。また，一部のシステム定義は，詳細についても説明します。UNIX 版 OpenTP1 と共通の内容については，OpenTP1 シリーズのマニュアルを参照してください。

5.1 システム定義の一覧

5.2 システム定義の説明形式

5.3 システム定義の詳細

5.1 システム定義の一覧

Windows 版 OpenTP1 は、UNIX 版 OpenTP1 と比べて、指定できるシステム定義が異なります。また、一部の定義では、引数や指定値をサポートしていないなどの制限事項があります。Windows 版 OpenTP1 を初めて使用する場合は、ここで示す一覧表で UNIX 版 OpenTP1 との差異があるかどうかを確認してください。

この一覧表の内容を確認した上で、UNIX 版 OpenTP1 と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。UNIX 版 OpenTP1 と差異のある定義については、必要に応じて、「5.3 システム定義の詳細」の該当箇所を参照してください。Windows 版 OpenTP1 固有の定義については、「5.3 システム定義の詳細」の該当箇所を参照してください。

！ 注意事項

この一覧表では、次の定義について説明していません。

- **メッセージキューイング機能 (TP1/Message Queue) の定義**
定義の詳細については、マニュアル「TP1/Message Queue 使用の手引」を参照してください。
- **TP1/NET/TCP/IP の定義**
定義の詳細については、マニュアル「OpenTP1 プロトコル TP1/NET/TCP/IP 編」を参照してください。
- **TP1/Extension for .NET Framework の定義**
定義の詳細については、マニュアル「TP1/Extension for .NET Framework 使用の手引」を参照してください。

一覧表で使用している記号の凡例を次に示します。

(凡例)

- ◎：UNIX 版 OpenTP1 と同じ機能をサポートしています。
- ：Windows 版 OpenTP1 固有のオペランドです。
- △：UNIX 版 OpenTP1 との機能差異や Windows 版 OpenTP1 での制限事項があります。
- ×：サポートしていません。
- －：該当しません。

(1) システム環境定義の内容

システム環境定義の内容を、次に示します。

表 5-1 システム環境定義の内容

形式	オペランド／パス	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
set	mode_conf	△	指定値は、AUTO だけをサポートしています。 詳細については、「5.3 システム定義の詳細」 の「システム環境定義」を参照してください。
	static_shmpool_size	◎	—
	dynamic_shmpool_size	◎	—
	shmpool_attribute	△	指定値は、free だけをサポートしています。 詳細については、「5.3 システム定義の詳細」 の「システム環境定義」を参照してください。
	user_command	△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「システム環境定 義」を参照してください。
	server_count	◎	—
	user_server_ha	◎	—
	system_terminate_ watch_time	◎	—
	start_scheduling_ti ming	◎	—
	system_init_watch_ time	◎	—
	user_command_onli ne	△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「システム環境定 義」を参照してください。
	preend_warning_w atch_time	◎	—
	user_command_onli ne_tpmngr_id	×	ユーザ ID の指定は、サポートしていません。
	redirect_file	○	Windows 版 OpenTP1 固有のオペランドです。 詳細については、「5.3 システム定義の詳細」 の「システム環境定義」を参照してください。
	redirect_file_name	○	
	redirect_file_size	○	
	console_output	○	
putenv	DCCONFPATH	◎	—
	DCADMDEBUG	△	デフォルト値が UNIX 版 OpenTP1 とは異な ります。詳細については、「5.3 システム定 義の詳細」の「システム環境定義」を参照し てください。

5. システム定義

形式	オペランド／パス	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	DCUAPCONFPATH	◎	—
deput env	DCCONFPATH	△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「システム環境定 義」を参照してください。
	DCUAPCONFPATH	△	

(2) システムサービス構成定義の内容

システムサービス構成定義の内容を、次に示します。

表 5-2 システムサービス構成定義の内容

形式	オペランド／ コマンド	オブ ション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
set	uap_conf		◎	—
	dam_conf		◎	—
	tam_conf		◎	—
	que_conf		◎	—
	ha_conf		◎	—
	jar_conf		×	グローバルアーカイブジャーナル機能は、 サポートしていません。
	clt_conf		◎	—
	ist_conf		×	IST サービスは、サポートしていません。
	rmm_conf		×	RMM サービスは、サポートしていま せん。
	xat_conf		×	XATMI 通信サービスは、サポートして いません。
	mqa_conf		◎	—
コマン ド	dcsvstart	-m	◎	—

(3) ユーザサービス構成定義の内容

ユーザサービス構成定義の内容を、次に示します。

表 5-3 ユーザサービス構成定義の内容

形式	コマンド	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
コマンド	dcsvstart	-u	◎	—

(4) システム共通定義の内容

システム共通定義の内容を、次に示します。

表 5-4 システム共通定義の内容

形式	オペランド／コマンド／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
set	rpc_trace		◎	—
	rpc_trace_name		◎	—
	rpc_trace_size		◎	—
	name_port		◎	—
	system_id		◎	—
	all_node		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「システム共通定義」を参照してください。
	node_id		◎	—
	rpc_retry		◎	—
	rpc_retry_count		◎	—
	rpc_retry_interval		◎	—
	multi_node_option		×	マルチノード機能（TP1/Multi）は、サポートしていません。
	prc_port		×	マルチノード機能（TP1/Multi）をサポートしていないため、指定する必要はありません。
	rpc_delay_statistics		◎	—
	my_host		◎	—
	jp1_use		×	JP1/Base の JP1 イベントサービス機能へのイベント登録は、サポートしていません。
	rpc_message_level		◎	—
	rpc_multi_tp1_in_same_host		×	同一ホスト内の複数の OpenTP1 を同一グローバルドメインとする運用は、サポートしていません。
	max_socket_descriptors		◎	—

5. システム定義

形式	オペランド／コマンド／パス	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	rpc_datacomp		◎	—
	domain_masters_addr		×	ドメインを指定した RPC は、サポートしていません。
	domain_masters_port		×	
	domain_use_dns		×	
	client_uid_check		◎	—
	rpc_port_base		◎	—
	prf_trace		◎	—
	trn_prf_trace_level		◎	—
	core_suppress_watch_time		◎	—
	rpc_netmask		◎	—
	ipc_socketl_highwater		◎	—
	ipc_socketl_watchtime		◎	—
	ipc_conn_interval		◎	指定値が UNIX 版 OpenTP1 とは異なります。詳細については、「5.3 システム定義の詳細」の「システム共通定義」を参照してください。
	ipc_send_interval		◎	—
	ipc_send_count		◎	—
	ipc_header_rcv_time		◎	—
	name_notify		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「システム共通定義」を参照してください。
	all_node_ex		◎	—
	rpc_server_busy_count		◎	—
	rpc_send_retry_count		◎	—
	rpc_send_retry_interval		◎	—
	thdlock_sleep_time		◎	—
	ipc_rcvbuf_size		◎	—
	ipc_sendbuf_size		◎	—

形式	オペランド／コマンド／パス	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	ipc_listen_sockbufs		◎	—
	rpc_router_retry_count		◎	—
	rpc_router_retry_interval		◎	—
	ipc_backlog_count		◎	—
	statistics		◎	—
	name_domain_file_use		◎	—
	all_node_extend_number		◎	—
	all_node_ex_extend_number		◎	—
	prc_current_work_path		◎	—
	rpc_max_message_size		◎	—
	uap_trace_file_put		×	デフォルトで UAP トレースファイルが出力されます。UAP トレースファイルの出力場所とファイル名については、マニュアル「OpenTP1 運用と操作」の OpenTP1 が出力するファイル一覧についての説明を参照してください。
	dcstart_wakeup_retry_count		×	サポートしていません。
	dcstart_wakeup_retry_interval		×	
	nam_prf_trace_level		◎	—
	fil_prf_trace_option		◎	—
	fil_prf_trace_delay_time		◎	—
	jnl_prf_event_trace_level		◎	—
	jnl_fileless_option		×	ジャーナルファイルレス機能は、サポートしていません。
	prc_coredump_filter		×	core ファイルに共有メモリを含めません。
	coredump_type		○	Windows 版 OpenTP1 固有のオペランドです。詳細については、「5.3 システム定義の詳細」の「システム共通定義」を参照してください。

5. システム定義

形式	オペランド／コマンド／パス	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	prc_terminate_force		○	Windows 版 OpenTP1 固有のオペランドです。詳細については、「5.3 システム定義の詳細」の「システム共通定義」を参照してください。
	name_service_mode		◎	—
	name_manager_node		◎	—
	name_remove_down_node		◎	—
	name_node_add_policy		◎	—
	watch_time		◎	—
コマンド	dcbindht	-h	◎	—
		-n	◎	—
putenv	LANG		◎	—

(5) ロックサービス定義の内容

ロックサービス定義の内容を、次に示します。

表 5-5 ロックサービス定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	lck_limit_foruser	◎	—
	lck_limit_fordam	◎	—
	lck_limit_fortam	◎	—
	lck_limit_formqa	◎	—
	lck_wait_timeout	◎	—
	lck_deadlock_info	◎	—
	lck_deadlock_info_remove	◎	—
	lck_deadlock_info_remove_level	◎	—
	lck_release_detect	◎	—
	lck_release_detect_interval	◎	—
	lck_prf_trace_level	◎	—

(6) タイマサービス定義の内容

タイマサービス定義の内容を、次に示します。

表 5-6 タイマサービス定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
set	tim_watch_count	◎	—

(7) ネームサービス定義の内容

ネームサービス定義の内容を、次に示します。

表 5-7 ネームサービス定義の内容

形式	オペランド ／コマンド	オ プ シ ョ ン	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
set	name_total_size		◎	—
	name_cache_size		◎	—
	max_socket_descri ptors		◎	—
	name_global_looku p		◎	—
	name_service_exte nd		◎	—
	name_audit_conf		◎	—
	name_audit_interv al		◎	—
	name_audit_watch _time		◎	—
	name_rpc_control_l ist		◎	—
	name_nodeid_chec k_message		◎	—
	name_cache_validit y_time		◎	—
	watch_time		◎	—
	name_start_watch _time		◎	—
	name_start_retry_c ount		◎	—

5. システム定義

形式	オペランド ／コマンド	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	name_start_retry_i nterval		◎	—
	name_start_error		◎	—
	name_sync_ready_t ime		◎	—
	namnlfil	·r	◎	—
コマ ンド		·n	◎	—

(8) プロセスサービス定義の内容

プロセスサービス定義の内容を、次に示します。

表 5-8 プロセスサービス定義の内容

形式	オペランド／コマ ンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	prc_process_count	◎	—
	prc_recovery_residen t	◎	—
	prc_take_over_svpat h	◎	—
	prc_take_over_dlpat h	×	サポートしていません。
	term_watch_time	◎	—
	max_socket_descript ors	◎	—
	term_watch_count	◎	—
	prc_prf_trace	◎	—
	prc_coresave_path	◎	—
	prc_corecompress	◎	—
	watch_time	◎	—
コマ ンド	prcsvpath	△	ユーザサーバのパス名を変更した場合でも、必ず先頭に %DCDIR%\¥bin が付与されます。詳細については、「5.3 システム定義の詳細」の「プロセスサービス定義 (prcsvpath)」を参照してください。

(9) スケジュールサービス定義の内容

スケジュールサービス定義の内容を、次に示します。

表 5-9 スケジュールサービス定義の内容

形式	オペランド／コマンド	オプション	Windows 版 OpenTP1 でのサポ ートの有無	UNIX 版 OpenTP1 との差異
set	scd_server_count		○	—
	scd_hold_recovery		○	—
	scd_hold_recovery_co unt		○	—
	scd_port		○	—
	scd_this_node_first		○	—
	scd_announce_server _status		○	—
	max_socket_descript ors		○	—
	schedule_rate		○	—
	scd_retry_of_comm_e rror		○	—
	scd_advertise_control		○	—
	scd_message_level		○	—
	ipc_tcpnodelay		○	—
	watch_time		○	—
コマ ンド	scdbufgr p	-g	○	—
		-e	○	—
		-n	○	—
		-l	○	—
		-s	○	—
		-p	○	—
	scdmulti	-m	○	—
		-p	○	—
		-g	○	—
		-t	○	—

(10) トランザクションサービス定義の内容

トランザクションサービス定義の内容を、次に示します。

表 5-10 トランザクションサービス定義の内容

形式	オペランド／ コマンド	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
set	trn_tran_process_count		◎	—
	trn_recovery_process_count		◎	—
	trn_expiration_time		◎	—
	trn_expiration_time_suspen d		◎	—
	trn_tran_statistics		◎	—
	trn_tran_recovery_list		◎	—
	trn_cpu_time		×	CPU 監視時間の指定は、サポートしていません。このオペランドには 0（CPU 時間を監視しない）が指定されます。
	trn_statistics_item		△	指定値として、cputime はサポートしていません。
	trn_max_subordinate_count		◎	—
	trn_rm_open_close_scope		◎	—
	trn_optimum_item		◎	—
	trn_processing_in_rm_error		◎	—
	trn_recovery_list_remove		◎	—
	trn_recovery_list_remove_le vel		◎	—
	trn_crm_use		×	通信リソースマネージャは、サポートしていません。
	trn_max_crm_subordinate_ count		×	
	trn_watch_time		◎	—
	trn_rollback_information_p ut		◎	—
	trn_limit_time		◎	—
	trn_rollback_response_recei ve		◎	—
	trn_partial_recovery_type		◎	—
	max_socket_descriptors		◎	—
	trn_recovery_failmsg_interv al		◎	—
	trn_wait_rm_open		◎	—
	trn_retry_interval_rm_open		◎	—
	trn_retry_count_rm_open		◎	—

形式	オペランド／ コマンド	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	thread_stack_size		◎	—
	polling_control_data		◎	—
	thread_yield_interval		◎	—
	groups		×	サービスグループのグループアクセス リストの設定は、サポートしていません。
	trn_xar_use		◎	—
	trn_start_recovery_mode		◎	—
	trn_start_recovery_watch_time		◎	—
	trn_start_recovery_interval		◎	—
	trn_xa_commit_error		◎	—
	trn_prf_event_trace_level		◎	—
	trn_prf_event_trace_condition		◎	—
	trn_completion_limit_time		◎	—
	trn_extend_function		◎	—
	watch_time		◎	—
	trn_rcv_open_close_scope		◎	—
コマンド	trnstring	-n	◎	—
		-i	◎	—
		-o	◎	—
		-c	◎	—
		-O	◎	—
		-C	◎	—
		-d	◎	—
		-e	◎	—
		-m	◎	—
		-r	◎	—
		-s	◎	—
putenv	任意		◎	—
dputenv	任意		◎	—

(11) XA リソースサービス定義の内容

XA リソースサービス定義の内容を、次に示します。

表 5-11 XA リソースサービス定義の内容

形式	オペランド／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	xar_eventtrace_level		◎	—
	xar_eventtrace_record		◎	—
	xar_session_time		◎	—
	xar_msdtc_use		◎	—
	xar_prf_trace_level		◎	—
コマンド	xarfile	-t	◎	—
		-a	◎	—

(12) インタバルサービス定義の内容

インタバルサービス定義の内容を、次に示します。

表 5-12 インタバルサービス定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	watch_time	◎	—

(13) ステータスサービス定義の内容

ステータスサービス定義の内容を、次に示します。

表 5-13 ステータスサービス定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	sts_file_name_1	◎	—
	sts_file_name_2	◎	—
	sts_file_name_3	◎	—
	sts_file_name_4	◎	—
	sts_file_name_5	◎	—
	sts_file_name_6	◎	—
	sts_file_name_7	◎	—
	sts_initial_error_switch	◎	—

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	sts_single_operation_switch	◎	—
	sts_last_active_file	◎	—
	sts_last_active_side	◎	—
	watch_time	◎	—

(14) ジャーナルサービス定義の内容

ジャーナルサービス定義の内容を、次に示します。

表 5-14 ジャーナルサービス定義の内容

形式	オペランド／ コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	jnl_tran_optimum_level		◎	—
	jnl_arc_terminate_timeout		×	グローバルアーカイブジャーナル機能は、サポートしていません。
	max_socket_descriptors		◎	—
	jnl_arc_ipc_buff_size		×	グローバルアーカイブジャーナル機能は、サポートしていません。
	jnl_watch_time		△	有効範囲に、ジャーナルファイルの I/O 処理完了受信応答待ち時間も含まれます。詳細については、「5.3 システム定義の詳細」の「ジャーナルサービス定義」を参照してください。
	watch_time		◎	—
	jnl_message_id_change_level		◎	—
コマンド	jnldfsv	-r	◎	—
		-c	◎	—

(15) システムジャーナルサービス定義の内容

システムジャーナルサービス定義の内容を、次に示します。

表 5-15 システムジャーナルサービス定義の内容

形式	オペランド／ コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	jnl_max_datasize		◎	—
	jnl_edinterval		◎	—
	jnl_rerun_swap		◎	—

5. システム定義

形式	オペランド／ コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	jnl_dual		◎	—
	jnl_singleoperation		◎	—
	jnl_rerun_reserved_file_ope n		◎	—
	jnl_arc_name		×	グローバルアーカイブジャーナル 機能は、サポートしていません。
	jnl_arc_buff_size		×	
	jnl_arc_max_datasize		×	
	jnl_arc_terminate_check		×	
	jnl_arc_rec_kind		×	
	jnl_arc_uj_code		×	
	jnl_arc_check_level		×	
	jnl_arc_trn_stat		×	
	jnl_unload_check		◎	—
	jnl_auto_unload		◎	—
	jnl_auto_unload_path		◎	—
	jnl_max_file_dispersion		◎	—
	jnl_min_file_dispersion		◎	—
	watch_time		◎	—
コマン ド	jnladdfg	-g	◎	—
		ONL	◎	—
	jnladdpf	-g	◎	—
		-e	◎	—
		-a	◎	—
		-b	◎	—

(16) チェックポイントダンプサービス定義の内容

チェックポイントダンプサービス定義の内容を、次に示します。

表 5-16 チェックポイントダンプサービス定義の内容

形式	オペランド／ コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	jnl_objservername		◎	—
	jnl_max_datasize		◎	—
	assurance_count		◎	—

形式	オペランド／ コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	jnl_reduced_mode		◎	—
	jnl_reserved_file_auto_open		◎	—
	jnl_dual		◎	—
	jnl_singleoperation		◎	—
	jnl_cdskip_limit		◎	—
	jnl_cdskip_msg		◎	—
	watch_time		◎	—
	cpd_message_id_change_level		◎	—
コマンド	jnladdfg	-j	◎	—
		-g	◎	—
		ONL	◎	—
	jnladdpf	-j	◎	—
		-g	◎	—
		-a	◎	—
		-b	◎	—

(17) ログサービス定義の内容

ログサービス定義の内容を、次に示します。

表 5-17 ログサービス定義の内容

形式	オペランド／パス	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	log_filesize	◎	—
	log_msg_console	◎	—
	log_msg_allno	◎	—
	log_msg_prcid	◎	—
	log_msg_prcno	◎	—
	log_msg_sysid	◎	—
	log_msg_date	◎	—
	log_msg_time	◎	—
	log_msg_hostname	◎	—
	log_msg_pgmid	◎	—
	log_netm_out	×	JP1/Cm2 によるネットワーク管理は、サポートしていません。

5. システム定義

形式	オペランド／パス	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	log_netm_allno	×	
	log_netm_prcid	×	
	log_netm_prcno	×	
	log_netm_sysid	×	
	log_netm_date	×	
	log_netm_time	×	
	log_netm_hostname	×	
	log_netm_pgmid	×	
	log_jp1_allno	×	JP1/Base の JP1 イベントサービス機能へのイベント登録は、サポートしていません。
	log_jp1_prcid	×	
	log_jp1_prcno	×	
	log_jp1_sysid	×	
	log_jp1_date	×	
	log_jp1_time	×	
	log_jp1_hostname	×	
	log_jp1_pgmid	×	
	log_notify_out	◎	—
	log_notify_allno	◎	—
	log_notify_prcid	◎	—
	log_notify_prcno	◎	—
	log_notify_sysid	◎	—
	log_notify_date	◎	—
	log_notify_time	◎	—
	log_notify_hostname	◎	—
	log_notify_pgmid	◎	—
	log_jerr_rint	×	JP1/Base の JP1 イベントサービス機能へのイベント登録は、サポートしていません。
	log_syslog_out	×	syslog への出力機能は、サポートしていません。
	log_syslog_allno	×	
	log_syslog_prcid	×	
	log_syslog_prcno	×	
	log_syslog_sysid	×	
	log_syslog_date	×	

形式	オペランド／パス	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	log_syslog_time	×	
	log_syslog_hostname	×	
	log_syslog_pgmid	×	
	log_syslog_append_nodeid	×	
	log_syslog_elist	×	
	log_syslog_elist_rint	×	
	log_syslog_synchro	×	
	log_audit_out	◎	—
	log_audit_path	◎	—
	log_audit_size	◎	—
	log_audit_count	◎	—
	log_audit_message	◎	—
	watch_time	◎	—
putenv	TZ	◎※	—
	DCSYSLOGOUT	×	syslog への出力機能は、サポートしていません。

注※

定義を指定する前に、「4.4 環境変数の設定」に示す環境変数 TZ の注意事項を参照してください。

(18) マルチノード構成定義の内容

マルチノード構成定義の内容を、次に示します。

表 5-18 マルチノード構成定義の内容

形式	オペランド／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	dcmstart_watch_time		×	マルチノード機能 (TP1/Multi) は、サポートしていません。
	dcmstop_watch_time		×	
	watch_time		×	
コマンド	dcmarea	-m	×	
		-g	×	
		-w	×	

(19) マルチノード物理定義の内容

マルチノード物理定義の内容を、次に示します。

表 5-19 マルチノード物理定義の内容

形式	コマンド	オプション	Windows 版 OpenTP1 でのサポ- ートの有無	UNIX 版 OpenTP1 との差異
コマ ンド	deprepor t	-w	×	マルチノード機能 (TP1/Multi) は、サ ポートしていません。
		-h	×	
		-p	×	

(20) グローバルアーカイブジャーナルサービス定義の内容

グローバルアーカイブジャーナルサービス定義の内容を、次に示します。

表 5-20 グローバルアーカイブジャーナルサービス定義の内容

形式	オペラ ンド/コマ ンド	オプション	Windows 版 OpenTP1 でのサポ- ートの有無	UNIX 版 OpenTP1 との差異
set	jnl_arc_terminate_ti meout		×	グローバルアーカイブジャーナル機能は、 サポートしていません。
	max_socket_descript ors		×	
	jnl_arc_ipc_buff_size		×	
	nl_watch_time		×	
コマ ンド	jnldfs	-a	×	

(21) アーカイブジャーナルサービス定義の内容

アーカイブジャーナルサービス定義の内容を、次に示します。

表 5-21 アーカイブジャーナルサービス定義の内容

形式	オペラ ンド/コマ ンド	オプション	Windows 版 OpenTP1 でのサポ- ートの有無	UNIX 版 OpenTP1 との差異
set	jnl_dual		×	グローバルアーカイブジャーナル機能は、 サポートしていません。
	jnl_singleoperation		×	
	jnl_rerun_swap		×	
	jnl_max_file_dispersi on		×	

形式	オペランド／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
コマンド	jnl_min_file_dispersi on		×	
			×	
			×	
	jnladdfg	-g	×	
		ONL	×	
	jnladdpf	-g	×	
		-e	×	
		-a	×	
		-b	×	

(22) DAM サービス定義の内容

DAM サービス定義の内容を、次に示します。

表 5-22 DAM サービス定義の内容

形式	オペランド／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	dam_update_block		◎	—
	dam_added_file		◎	—
	dam_update_block_o ver		◎	—
	dam_message_level		◎	—
	dam_tran_process_co unt		◎	—
	dam_cache_size		◎	—
	dam_cache_size_fix		◎	—
	dam_cache_attribute		△	指定値は、free だけをサポートしています。
	dam_io_interval		◎	—
	dam_transaction_acc ess		◎	—
	dam_io_error_occur		◎	—
	dam_cache_reuse_fro m		◎	—
	dam_default_cache_n um		◎	—

5. システム定義

形式	オペランド／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	dam_ex_refer_read		◎	—
	dam_max_block_size		◎	—
	dam_kb_size		◎	—
	watch_time		◎	—
コマンド	damcache	引数	◎	—
	damchlimit	引数	◎	—
	damfile	-d	◎	—
		-n	◎	—
		-f	◎	—
		-c	◎	—
		引数	◎	—

(23) TAM サービス定義の内容

TAM サービス定義の内容を、次に示します。

表 5-23 TAM サービス定義の内容

形式	オペランド／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	tam_max_tblnum		◎	—
	tam_max_filesize		◎	—
	tam_max_recsz		◎	—
	tam_jnl_err_flag		◎	—
	tam_pool_attri		△	指定値は、free だけをサポートしています。
	tam_tbl_lock_mode		◎	—
	tam_cbl_level		◎	—
	tam_max_trnnum		◎	—
	tam_max_trnfilnum		◎	—
	watch_time		◎	—
コマンド	tamtable	-o	◎	—
		-a	◎	—
		-i	◎	—

形式	オペランド／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
		-j	◎	—
		引数	◎	—

(24) クライアントサービス定義の内容

クライアントサービス定義の内容を、次に示します。

表 5-24 クライアントサービス定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	parallel_count	◎	—
	balance_count	◎	—
	trn_expiration_time	◎	—
	trn_expiration_time_suspend	◎	—
	trn_cpu_time	×	CPU 監視時間の指定は、サポートしていません。このオペランドには 0 (CPU 時間を監視しない) が指定されます。
	open_rm	◎	—
	clt_inquire_time	◎	—
	clt_port	◎	—
	clt_trn_conf	◎	—
	clt_cup_conf	◎	—
	cup_parallel_count	◎	—
	cup_balance_count	◎	—
	clttrn_port	◎	—
	cltcon_port	◎	—
	trn_statistics_item	△	指定値として、cputime はサポートしていません。
	trn_optimum_item	◎	—
	trn_watch_time	◎	—
	trn_rollback_information_put	◎	—
	trn_limit_time	◎	—
	trn_rollback_response_receive	◎	—
	trn_partial_recovery_type	◎	—

5. システム定義

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	trn_completion_limit_time	◎	—
	message_store_buffer	△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「クライアントサービス定義」を参照してください。
	watch_time	◎	—

(25) IST サービス定義の内容

IST サービス定義の内容を、次に示します。

表 5-25 IST サービス定義の内容

形式	オペランド／コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	ist_node	×	IST サービスは、サポートしていません。
	ist_node_group	×	
コマンド	istdef	×	

(26) RMM サービス定義の内容

RMM サービス定義の内容を、次に示します。

表 5-26 RMM サービス定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	rmm_check_services	×	RMM サービスは、サポートしていません。
	rmm_system_behavior	×	
	rmm_down_with_system	×	
	rmm_sysdown_with_rm	×	

(27) 監視対象 RM 定義の内容

監視対象 RM 定義の内容を、次に示します。

表 5-27 監視対象 RM 定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	rmm_start_command	×	RMM サービスは、サポートしていません。
	rmm_stop_command	×	
	rmm_abort_command	×	
	rmm_get_pid_command	×	
	rmm_command_watch_time	×	
	rmm_command_uid	×	
	rmm_command_gid	×	
	rmm_start_watch_time	×	

(28) 拡張 RM 登録定義の内容

拡張 RM 登録定義の内容を、次に示します。

表 5-28 拡張 RM 登録定義の内容

形式	コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
コマンド	trnlrkrm	-a	×	拡張 RM 登録定義は、サポートしていません。
		-s	×	
		-o	×	
		-C	×	
		-B	×	
		-l	×	
		-f	×	

(29) XATMI 通信サービス定義の内容

XATMI 通信サービス定義の内容を、次に示します。

5. システム定義

表 5-29 XATMI 通信サービス定義の内容

形式	オペランド／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	xatinitapt		×	XATMI 通信サービスは、サポートしていません。
	xatinitaeq		×	
	xat_aso_con_event_svcname		×	
	xat_aso_discon_event_svcname		×	
	xat_aso_failure_event_svcname		×	
	max_open_fds		×	
	max_socket_descriptors		×	
コマンド	xatsrvad	-p	×	
		-q	×	
		-s	×	

(30) メッセージキューサービス定義の内容

メッセージキューサービス定義の内容を、次に示します。

表 5-30 メッセージキューサービス定義の内容

形式	オペランド／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	que_xidnum		◎	—
	que_io_maxrecsize		◎	—
コマンド	quegrp	-g	◎	—
		-f	◎	—
		-n	◎	—
		-m	◎	—
		-w	◎	—
		-c	◎	—

(31) ユーザサービスネットワーク定義の内容

ユーザサービスネットワーク定義の内容を、次に示します。

表 5-31 ユーザーサービスネットワーク定義の内容

形式	コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
コマンド	dcsvgdef	-g	◎	—
		-h	◎	—
		-p	◎	—
		-t	◎	—
		-w	◎	—

(32) rap リスナーサービス定義の内容

rap リスナーサービス定義の内容を、次に示します。

表 5-32 rap リスナーサービス定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	rap_listen_port	◎	—
	rap_parallel_server	△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「rap リスナーサービス定義」を参照してください。
	rap_watch_time	◎	—
	rap_inquire_time	◎	—
	nice	△	指定値の意味が UNIX 版 OpenTP1 とは異なります。詳細については、「5.3 システム定義の詳細」の「rap リスナーサービス定義」を参照してください。
	uap_trace_max	◎	—
	uid	×	ユーザ ID の指定は、サポートしていません。
	rpc_response_statistics	◎	—
	rpc_trace	◎	—
	rpc_trace_name	◎	—
	rpc_trace_size	◎	—
	trn_expiration_time	◎	—
	trn_expiration_time_suspend	◎	—
	trn_cpu_time	×	CPU 監視時間の指定は、サポートしていません。このオペランドには 0 (CPU 時間を監視しない) が指定されます。
	trf_put	◎	—

5. システム定義

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	trn_statistics_item	△	指定値として、cputime はサポートしていません。
	trn_optimum_item	◎	—
	trn_watch_time	◎	—
	trn_rollback_information_put	◎	—
	trn_limit_time	◎	—
	trn_rollback_response_receive	◎	—
	trn_partial_recovery_type	◎	—
	rap_inquire_timeout_message	◎	—
	rap_connection_assignment_type	◎	—
	rap_max_client	◎	—
	rap_notify	◎	—
	rap_client_manager_node	◎	—
	rap_max_buff_size	◎	—
	rap_io_retry_interval	◎	—
	rap_sock_count	◎	—
	rap_sock_interval	◎	—
	rap_connect_retry_count	◎	—
	rap_connect_retry_interval	◎	—
	rap_listen_backlog	△	指定値が UNIX 版 OpenTP1 とは異なります。詳細については、「5.3 システム定義の詳細」の「rap リスナーサービス定義」を参照してください。
	rap_msg_output_interval	◎	—
	rap_recovery_server	◎	—
	rap_connect_interval	◎	—
	rpc_extend_function	◎	—
	max_socket_descriptors	◎	—
	trn_completion_limit_time	◎	—

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	rap_message_id_change_level	◎	—
	rap_term_disconnect_time	◎	—
	rap_stay_watch_time	◎	—
	rap_stay_warning_interval	◎	—
	log_audit_out_suppress	◎	—
	log_audit_message	◎	—
	ipc_socketl_highwater	◎	—
	ipc_socketl_watchtime	◎	—
	scs_prf_trace_level	◎	—
	rap_extend_function	◎	—
	watch_time	◎	—

(33) rap クライアントマネージャサービス定義の内容

rap クライアントマネージャサービス定義の内容を、次に示します。

表 5-33 rap クライアントマネージャサービス定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	rap_client_manager_port	◎	—
	rap_listen_inf	◎	—
	uid	×	ユーザ ID の指定は、サポートしていません。
	log_audit_out_suppress	◎	—
	log_audit_message	◎	—
	rap_watch_time	◎	—

(34) 性能検証用トレース定義の内容

性能検証用トレース定義の内容を、次に示します。

表 5-34 性能検証用トレース定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	prf_file_size	◎	—
	prf_information_level	△	性能検証用トレース関連のメッセージは、イベントビューアに出力されます。なお、システム共通定義、または性能検証用トレース定義の環境変数 DCSYSLOGOUT をサポートしていないため、イベントビューアへのメッセージ出力の抑止はできません。
	prf_file_count	◎	—
	prf_trace_backup	◎	—
	prf_buff_size	◎	—

(35) XAR 性能検証用トレース定義の内容

XAR 性能検証用トレース定義の内容を、次に示します。

表 5-35 XAR 性能検証用トレース定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	prf_file_size	◎	—
	prf_information_level	△	XAR 性能検証用トレース関連のメッセージは、イベントビューアに出力されます。なお、システム共通定義、または性能検証用トレース定義の環境変数 DCSYSLOGOUT をサポートしていないため、イベントビューアへのメッセージ出力の抑止はできません。
	prf_file_count	◎	—

(36) JNL 性能検証用トレース定義の内容

JNL 性能検証用トレース定義の内容を、次に示します。

表 5-36 JNL 性能検証用トレース定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	prf_file_size	◎	—
	prf_file_count	◎	—
	prf_trace_backup	◎	—

(37) LCK 性能検証用トレース定義の内容

LCK 性能検証用トレース定義の内容を、次に示します。

表 5-37 LCK 性能検証用トレース定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	prf_file_size	◎	—
	prf_information_level	△	LCK 性能検証用トレース関連のメッセージは、イベントビューアに出力されます。なお、システム共通定義、または性能検証用トレース定義の環境変数 DCSYSLOGOUT をサポートしていないため、イベントビューアへのメッセージ出力の抑止はできません。
	prf_file_count	◎	—

(38) TRN イベントトレース定義の内容

TRN イベントトレース定義の内容を、次に示します。

表 5-38 TRN イベントトレース定義の内容

形式	オペランド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	prf_file_size	◎	—
	prf_information_level	△	TRN イベントトレース関連のメッセージは、イベントビューアに出力されます。なお、システム共通定義、または性能検証用トレース定義の環境変数 DCSYSLOGOUT をサポートしていないため、イベントビューアへのメッセージ出力の抑止はできません。
	prf_file_count	◎	—

(39) リアルタイム統計情報サービス定義の内容

リアルタイム統計情報サービス定義の内容を、次に示します。

表 5-39 リアルタイム統計情報サービス定義の内容

形式	オペランド ／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	rts_treput_interval		◎	—
	rts_service_max		◎	—
	rts_item_max		◎	—

5. システム定義

形式	オペランド ／コマンド	オプション	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	rts_log_file		◎	—
	rts_log_file_name		◎	—
	rts_log_file_size		◎	—
	rts_log_file_count		◎	—
	rts_log_file_backup		◎	—
	rts_swap_message		◎	—
コマンド	rtsput	-u	◎	—
		-s	◎	—
		-v	◎	—
		-o	◎	—
		-b	◎	—
		-e	◎	—
		-f	◎	—

(40) リアルタイム取得項目定義の内容

リアルタイム取得項目定義の内容を、次に示します。

表 5-40 リアルタイム取得項目定義の内容

形式	オペランド	定義種別	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
set	rts_cpd_collect_cpd	チェックポイント ダンプ情報	◎	—
	rts_cpd_validt_cpd		◎	—
	rts_jnl_buf_full	ジャーナル情報	◎	—
	rts_jnl_wait_buf		◎	—
	rts_jnl_jnl_output		◎	—
	rts_jnl_io_wait		◎	—
	rts_jnl_write		◎	—
	rts_jnl_swap		◎	—
	rts_jnl_jnl_input		◎	—
	rts_jnl_read		◎	—
	rts_lck_lock_acqst	ロック情報	◎	—
	rts_lck_lock_wait		◎	—
	rts_lck_deadlock		◎	—

形式	オペランド	定義種別	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差 異
	rts_nam_global_cache_hit	ネーム情報	○	—
	rts_nam_local_cache_hit		○	—
	rts_nam_lookup		○	—
	rts_nam_node_lookup		○	—
	rts_nam_node_lookup_responce		○	—
	rts_osl_stamem_acq	共用メモリ管理 情報	○	—
	rts_osl_stamem_poll		○	—
	rts_osl_dynmem_acq		○	—
	rts_osl_dynmem_poll		○	—
	rts_prc_prc_genert	プロセス情報	○	—
	rts_prc_uap_abnml		○	—
	rts_prc_sys_abnml		○	—
	rts_prc_prc_term		○	—
	rts_prc_prc_num		○	—
	rts_que_read	メッセージ キュー情報	○	—
	rts_que_write		○	—
	rts_que_read_err		○	—
	rts_que_write_err		○	—
	rts_que_wait_buf		○	—
	rts_que_real_read		○	—
	rts_que_real_write		○	—
	rts_que_delay_wrt		○	—
	rts_que_delay_rec		○	—
	rts_que_delay_msg		○	—
	rts_rpc_rpc_call	RPC 情報	○	—
	rts_rpc_rpc_call_chained		○	—
	rts_rpc_usr_srvc		○	—

5. システム定義

形式	オペランド	定義種別	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差 異
	rts_rpc_rpc_ovrtim		◎	—
	rts_scd_scd_wait	スケジュール情 報	◎	—
	rts_scd_schedule		◎	—
	rts_scd_using_buf		◎	—
	rts_scd_lack_buf		◎	—
	rts_scd_scd_stay		◎	—
	rts_scd_svc_scd_w ait		◎	—
	rts_scd_svc_using_ buf		◎	—
	rts_scd_parallel		◎	—
	rts_trn_commit	トランザクシ ョン情報	◎	—
	rts_trn_rollback		◎	—
	rts_trn_cmt_cmd		◎	—
	rts_trn_rbk_cmd		◎	—
	rts_trn_haz_cmd		◎	—
	rts_trn_mix_cmd		◎	—
	rts_trn_branch		◎	—
	rts_trn_sync_point		◎	—
	rts_dam_read	DAM 情報	◎	—
	rts_dam_read_err		◎	—
	rts_dam_write		◎	—
	rts_dam_write_err		◎	—
	rts_dam_fj		◎	—
	rts_dam_trn_bran ch		◎	—
	rts_dam_cache_blo ck		◎	—
	rts_dam_shm_pool		◎	—
	rts_tam_real_rene w	TAM 情報	◎	—
	rts_tam_real_rene w_time		◎	—
	rts_tam_rec_refer		◎	—
	rts_tam_rec_renew		◎	—

形式	オペランド	定義種別	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差 異
	rts_tam_read		○	—
	rts_tam_read_err		○	—
	rts_tam_write		○	—
	rts_tam_write_err		○	—
	rts_xar_start	XA リソース サービス情報	○	—
	rts_xar_start_err		○	—
	rts_xar_call		○	—
	rts_xar_call_err		○	—
	rts_xar_end		○	—
	rts_xar_end_err		○	—
	rts_xar_prepare		○	—
	rts_xar_prepare_e rr		○	—
	rts_xar_commit		○	—
	rts_xar_commit_er r		○	—
	rts_xar_rollback		○	—
	rts_xar_rollback_e rr		○	—
	rts_xar_recover		○	—
	rts_xar_recover_er r		○	—
	rts_xar_forget		○	—
	rts_xar_forget_err		○	—
	rts_mcf_ap_scd_st ay	MCF 情報	○	—
	rts_mcf_ap_usr_sr vc		○	—
	rts_mcf_in_msg_sc d_wait		○	—
	rts_mcf_out_msg_s ync_scd_wait		○	—
	rts_mcf_out_msg_r esp_scd_wait		○	—
	rts_mcf_out_msg_ prio_scd_wait		○	—
	rts_mcf_out_msg_ norm_scd_wait		○	—

5. システム定義

形式	オペランド	定義種別	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差 異
	rts_mcf_que_scd_ wait_num		◎	—

(41) ユーザサービスデフォルト定義の内容

ユーザサービスデフォルト定義の内容を、次に示します。

表 5-41 ユーザサービスデフォルト定義の内容

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
set	nice		△	指定値の意味が UNIX 版 OpenTP1 とは異なります。詳細については、「5.3 システム定義の詳細」の「ユーザサービスデフォルト定義」を参照してください。
	parallel_count		◎	—
	hold		◎	—
	hold_recovery		◎	—
	deadlock_priority		◎	—
	schedule_priority		◎	—
	message_buflen		◎	—
	message_store_buflen		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「ユーザサービスデフォルト定義」を参照してください。
	trn_expiration_time		◎	—
	trn_expiration_time_suspe nd		◎	—
	watch_next_chain_time		◎	—
	atomic_update		◎	—
	receive_from		◎	—
	uap_trace_max		◎	—
	uap_trace_file_put		×	デフォルトで UAP トレースファイルが出力されます。UAP トレースファイルの出力場所とファイル名については、マニュアル「OpenTP1 運用と操作」の OpenTP1 が出力するファイル一覧についての説明を参照してください。
	term_watch_time		◎	—
	mcf_jnl_buff_size		◎	—
	type		◎	—

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	balance_count		◎	—
	uid		×	ユーザ ID の指定は、サポートしていません。
	auto_restart		◎	—
	critical		◎	—
	lck_wait_priority		◎	—
	mcf_psv_id		◎	—
	trn_cpu_time		×	CPU 監視時間の指定は、サポートしていません。このオペランドには 0（CPU 時間を監視しない）が指定されます。
	service_hold		◎	—
	service_priority_control		◎	—
	message_cell_size		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「ユーザサービスデフォルト定義」を参照してください。
	max_socket_msg		◎	—
	max_socket_msglen		◎	—
	trf_put		◎	—
	mcf_mgrid		◎	—
	mcf_service_max_count		◎	—
	trn_statistics_item		△	指定値として、cputime はサポートしていません。
	node_down_restart		◎	—
	rpc_response_statistics		◎	—
	server_type		△	指定値は、betran だけをサポートしています。
	trn_rm_open_close_scope		◎	—
	trn_optimum_item		◎	—
	purge_msgget		◎	—
	cancel_normal_terminate		◎	—
	prc_abort_signal		△	3, 6, 15 の中から、シグナル番号を指定してください。詳細については、「5.3 システム定義の詳細」の「ユーザサービスデフォルト定義」を参照してください。
	rpc_service_retry_count		◎	—
	rpc_extend_function		◎	—

5. システム定義

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	max_socket_descriptors		◎	—
	max_open_fds		◎	—
	service_term_watch_time		◎	—
	termed_after_service		◎	—
	xat_trn_expiration_time		×	XATMI 通信サービスは、サポートしていません。
	xat_osi_usr		×	
	rpc_trace		◎	—
	rpc_trace_name		◎	—
	rpc_trace_size		◎	—
	trn_rollback_information_put		◎	—
	schedule_method		×	ユーザサーバのスケジューリング方式の指定は、サポートしていません。
	service_wait_time		×	
	mcf_spp_oj		×	SPP の履歴情報 OJ 取得要否の指定は、サポートしていません。
	adm_message_option		◎	—
	trn_watch_time		◎	—
	trn_limit_time		◎	—
	trn_rollback_response_receive		◎	—
	trn_partial_recovery_type		◎	—
	rpc_destination_mode		◎	—
	rpc_rap_auto_connect		◎	—
	rpc_rap_inquire_time		◎	—
	rpc_request_cancel_for_timeout		◎	—
	status_change_when_termining		◎	—
	service_expiration_time		◎	—
	multi_schedule		◎	—
	make_queue_on_starting		◎	—
	loadcheck_interval		◎	—
	levelup_queue_count		◎	—
	leveldown_queue_count		◎	—

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	ipc_socketctl_highwater		◎	—
	ipc_socketctl_wathtime		◎	—
	ipc_conn_interval		◎	指定値が UNIX 版 OpenTP1 とは異なります。詳細については、「5.3 システム定義の詳細」の「システム共通定義」を参照してください。
	ipc_send_interval		◎	—
	ipc_send_count		◎	—
	ipc_header_recv_time		◎	—
	rpc_send_retry_count		◎	—
	rpc_send_retry_interval		◎	—
	ipc_recvbuf_size		◎	—
	ipc_sendbuf_size		◎	—
	ipc_listen_sockbufset		◎	—
	polling_control_data		◎	—
	thread_yield_interval		◎	—
	groups		×	サービスグループのグループアクセスリストの設定は、サポートしていません。
	loadlevel_message		◎	—
	ipc_backlog_count		◎	—
	rpc_buffer_pool_max		◎	—
	schedule_delay_limit		◎	—
	schedule_delay_abort		◎	—
	rap_autoconnect_con_error_msg		◎	—
	core_shm_suppress		×	コアファイルに共用メモリダンプは出力されません。
	xat_connect_resp_time		×	XATMI 通信サービスは、サポートしていません。
	scd_poolfull_check_interval		◎	—
	scd_poolfull_check_count		◎	—
	scd_pool_warning_use_rate		◎	—
	scd_pool_warning_interval		◎	—
	ipc_tcpnodelay		◎	—
	stay_watch_queue_count		◎	—
	stay_watch_check_rate		◎	—

5. システム定義

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	stay_watch_abort		◎	—
	stay_watch_start_interval		◎	—
	stay_watch_check_interval		◎	—
	trn_completion_limit_time		◎	—
	rap_message_id_change_level		◎	—
	log_audit_out_suppress		◎	—
	log_audit_message		◎	—
	mcf_prf_trace		◎	—
	scd_refresh_process		◎	—
	rap_extend_function		◎	—
	prc_coredump_filter		×	core ファイルに共有メモリを含めません。
	watch_time		◎	—
	process_privilege_restrict		○	Windows 版 OpenTP1 固有のオペランドです。詳細については、「5.3 システム定義の詳細」の「ユーザサービスデフォルト定義」を参照してください。
	process_privilege_name		○	
コマ ンド	trnrmid	-n	◎	—
		-i	◎	—
		-k	◎	—
	scdbufgrp	-g	◎	—
		-s	◎	—
		-p	◎	—
	scdmulti	-g	◎	—
	scdsvdef	-c	◎	—
		-p	◎	—
		-n	◎	—
		-l	◎	—
puten v	任意		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「ユーザサービスデフォルト定義」を参照してください。
	XAT_CONNECT_RESP_TIME		×	XATMI 通信サービスは、サポートしていません。

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
dcp ut env	任意		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「ユーザサービスデフォルト定義」を参照してください。

(42) ユーザサービス定義の内容

ユーザサービス定義の内容を、次に示します。

表 5-42 ユーザサービス定義の内容

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
set	service_group		◎	—
	module		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「ユーザサービス定義」を参照してください。
	service		◎	—
	nice		△	指定値の意味が UNIX 版 OpenTP1 とは異なります。詳細については、「5.3 システム定義の詳細」の「ユーザサービス定義」を参照してください。
	parallel_count		◎	—
	hold		◎	—
	hold_recovery		◎	—
	deadlock_priority		◎	—
	schedule_priority		◎	—
	message_buflen		◎	—
	message_store_buflen		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「ユーザサービス定義」を参照してください。
	trn_expiration_time		◎	—
	trn_expiration_time_suspend		◎	—
	watch_next_chain_time		◎	—
	atomic_update		◎	—
	receive_from		◎	—
	uap_trace_max		◎	—

5. システム定義

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	uap_trace_file_put		×	デフォルトで UAP トレースファイルが出力されます。UAP トレースファイルの出力場所とファイル名については、マニュアル「OpenTP1 運用と操作」の OpenTP1 が出力するファイル一覧についての説明を参照してください。
	term_watch_time		◎	—
	mcf_jnl_buff_size		◎	—
	type		◎	—
	balance_count		◎	—
	uid		×	ユーザ ID の指定は、サポートしていません。
	auto_restart		◎	—
	critical		◎	—
	lck_wait_priority		◎	—
	mcf_psv_id		◎	—
	trn_cpu_time		×	CPU 監視時間の指定は、サポートしていません。このオペランドには 0（CPU 時間を監視しない）が指定されます。
	service_hold		◎	—
	service_priority_control		◎	—
	message_cell_size		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「ユーザサービス定義」を参照してください。
	max_socket_msg		◎	—
	max_socket_msglen		◎	—
	trf_put		◎	—
	mcf_mgrid		◎	—
	mcf_service_max_count		◎	—
	trn_statistics_item		△	指定値として、cputime はサポートしていません。
	node_down_restart		◎	—
	rpc_response_statistics		◎	—
	server_type		△	指定値は、betran だけをサポートしています。
	trn_rm_open_close_scope		◎	—
	trn_optimum_item		◎	—

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	purge_msgget		◎	—
	cancel_normal_terminate		◎	—
	prc_abort_signal		△	3, 6, 15 の中から、シグナル番号を指定してください。詳細については、「5.3 システム定義の詳細」の「ユーザサービス定義」を参照してください。
	rpc_service_retry_count		◎	—
	rpc_extend_function		◎	—
	max_socket_descriptors		◎	—
	max_open_fds		◎	—
	service_term_watch_time		◎	—
	termed_after_service		◎	—
	xat_trn_expiration_time		×	XATMI 通信サービスは、サポートしていません。
	xat_osi_usr		×	
	rpc_trace		◎	—
	rpc_trace_name		◎	—
	rpc_trace_size		◎	—
	trn_rollback_information_put		◎	—
	schedule_method		×	ユーザサーバのスケジューリング方式の指定は、サポートしていません。
	service_wait_time		×	
	mcf_spp_oj		×	SPP の履歴情報 OJ 取得要否の指定は、サポートしていません。
	adm_message_option		◎	—
	trn_watch_time		◎	—
	trn_limit_time		◎	—
	trn_rollback_response_receive		◎	—
	trn_partial_recovery_type		◎	—
	rpc_destination_mode		◎	—
	rpc_rap_auto_connect		◎	—
	rpc_rap_inquire_time		◎	—
	rpc_request_cancel_for_timeout		◎	—

5. システム定義

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	status_change_when_termining		◎	—
	service_expiration_time		◎	—
	multi_schedule		◎	—
	make_queue_on_starting		◎	—
	loadcheck_interval		◎	—
	levelup_queue_count		◎	—
	leveldown_queue_count		◎	—
	ipc_sockctl_highwater		◎	—
	ipc_sockctl_watchtime		◎	—
	ipc_conn_interval		◎	指定値が UNIX 版 OpenTP1 とは異なります。詳細については、「5.3 システム定義の詳細」の「システム共通定義」を参照してください。
	ipc_send_interval		◎	—
	ipc_send_count		◎	—
	ipc_header_recv_time		◎	—
	rpc_send_retry_count		◎	—
	rpc_send_retry_interval		◎	—
	ipc_recvbuf_size		◎	—
	ipc_sendbuf_size		◎	—
	ipc_listen_sockbufset		◎	—
	polling_control_data		◎	—
	thread_yield_interval		◎	—
	groups		×	サービスグループのグループアクセスリストの設定は、サポートしていません。
	loadlevel_message		◎	—
	ipc_backlog_count		◎	—
	rpc_buffer_pool_max		◎	—
	schedule_delay_limit		◎	—
	schedule_delay_abort		◎	—
	rap_autoconnect_con_error_msg		◎	—
	core_shm_suppress		×	コアファイルに共用メモリダンプは出力されません。

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
	xat_connect_resp_time		×	XATMI 通信サービスは、サポートしていません。
	scd_poolfull_check_interval		◎	—
	scd_poolfull_check_count		◎	—
	scd_pool_warning_use_rate		◎	—
	scd_pool_warning_interval		◎	—
	ipc_tcpnodelay		◎	—
	stay_watch_queue_count		◎	—
	stay_watch_check_rate		◎	—
	stay_watch_abort		◎	—
	stay_watch_start_interval		◎	—
	stay_watch_check_interval		◎	—
	trn_completion_limit_time		◎	—
	rap_message_id_change_level		◎	—
	log_audit_out_suppress		◎	—
	log_audit_message		◎	—
	mcf_prf_trace		◎	—
	scd_refresh_process		◎	—
	rap_extend_function		◎	—
	prc_coredump_filter		×	core ファイルに共有メモリを含めません。
	watch_time		◎	—
	process_privilege_restrict		○	Windows 版 OpenTP1 固有のオペランドです。詳細については、「5.3 システム定義の詳細」の「ユーザサービス定義」を参照してください。
	process_privilege_name		○	
コマンド	trnrmid	-n	◎	—
		-i	◎	—
		-k	◎	—
	scdbufgrp	-g	◎	—
		-s	◎	—
		-p	◎	—
	scdmulti	-g	◎	—
	scdsvcddef	-c	◎	—

5. システム定義

形式	オペランド ／コマンド ／パス	オプション	Windows 版 OpenTP1 でのサ ポートの有無	UNIX 版 OpenTP1 との差異
		-p	◎	—
		-n	◎	—
		-l	◎	—
putenv	任意		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「ユーザーサービス定義」を参照してください。
	DCFPL_CONNECT_RETR Y_COUNT		◎	—
	DCFPL_CONNECT_RETR Y_INTERVAL		◎	—
	XAT_CONNECT_RESP_TIME		×	XATMI 通信サービスは、サポートしていません。
dcputenv	任意		△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「ユーザーサービス定義」を参照してください。

(43) MCF マネージャ定義の内容

MCF マネージャ定義の内容を、次に示します。

表 5-43 MCF マネージャ定義の内容

形式	コマンド	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
コマンド	mcfmenv	◎	—
	mcfmcomn	△	-o オプションの cmdsvname オペランド (MCF オンラインコマンドサービス), および -w オプションの stats オペランド (MCF 稼働統計情報) は、サポートしていません。
	mcfmcname	◎	—
	mcfmuap	◎	—
	mcfmqgid	◎	—
	mcfmexp	◎	—
	mcfmsts	◎	—
	mcfmsmsg	◎	—
	mcfmsvg	◎	—

(44) MCF 通信構成定義の内容

MCF 通信構成定義の内容を、次に示します。

表 5-44 MCF 通信構成定義の内容

形式	コマンド	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
コマンド	mcftenv	△	-q オプションの diskitq オペランド（入力キューにメモリキュー使用時の MCF 開始処理高速化）は、サポートしていません。
	mcftcomn	◎	—
	mcfttred	◎	—
	mcfttim	◎	—
	mcfttrc	◎	—
	mcftsts	◎	—
	mcftbuf	◎	—
	mcftpsvr	◎	—
	mcftalcle	◎	—
	mcftped	◎	—

(45) MCF アプリケーション定義の内容

MCF アプリケーション定義の内容を、次に示します。

表 5-45 MCF アプリケーション定義の内容

形式	コマンド	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
コマンド	mcfenv	◎	—
	mcfalcap	◎	—

(46) MCF 性能検証用トレース定義の内容

MCF 性能検証用トレース定義の内容を、次に示します。

表 5-46 MCF 性能検証用トレース定義の内容

形式	オペランド	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
set	prf_file_size	◎	—
	prf_file_count	◎	—

(47) システムサービス情報定義の内容

システムサービス情報定義の内容を、次に示します。

表 5-47 システムサービス情報定義の内容

形式	オペランド	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
set	module	△	注意事項があります。詳細については、「5.3 システム定義の詳細」の「システムサービス情報定義」を参照してください。
	mcf_prf_trace	◎	—

(48) システムサービス共通情報定義の内容

システムサービス共通情報定義の内容を、次に示します。

表 5-48 システムサービス共通情報定義の内容

形式	オペランド／パス	Windows 版 OpenTP1 での サポートの有無	UNIX 版 OpenTP1 との差異
set	max_socket_descriptors	◎	—
	max_open_fds	◎	—
	thdlock_sleep_time	◎	—
	mcf_prf_trace_level	◎	—
putenv	DCMCFQUEBAK	◎	—

5.2 システム定義の説明形式

各定義（定義コマンド）で説明する項目を次に示します。

定義名（定義コマンド）

形式

Windows 版 OpenTP1 固有のオプション・オペランドや、UNIX 版 OpenTP1 と差異があるオプション・オペランド定義の形式だけを説明しています。

説明、またはオプション

set 形式のオペランド

定義のオペランドについて説明しています。

コマンド形式

コマンド形式で指定する定義（定義コマンド）について説明しています。

putenv 形式のオペランド

ユーザの環境変数を指定する putenv 形式について説明しています。

dcputenv 形式のオペランド

ユーザの環境変数を指定する dcputenv 形式について説明しています。

注意事項

コマンドの注意事項について説明しています。

5.3 システム定義の詳細

ここでは、UNIX 版 OpenTP1 との定義の差異の詳細について説明します。ここで説明していない定義（オペランド）については、「5.1 システム定義の一覧」に示す一覧を参照してください。

システム環境定義

ここで説明していないオペランドについては、「5.1(1) システム環境定義の内容」に示す一覧を参照してください。

形式

set 形式

```
[set mode_conf=AUTO]
[set shmpool_attribute=free]
[set user_command=ユーザ環境設定コマンド]
[set user_command_online=システム開始完了コマンド]
[set redirect_file=Y|N]
[set redirect_file_name=出力先ファイル名]
[set redirect_file_size=出力先ファイルの最大サイズ]
[set console_output=Y|N]
```

putenv 形式

```
[putenv DCADMDEBUG 0|1]
```

dcputenv 形式

```
[dcputenv DCCONFPATH 定義ファイルの格納フォルダ]
[dcputenv DCUAPCONFPATH ユーザサービス定義ファイルの格納フォルダ]
```

説明

set 形式のオペランド

● mode_conf=AUTO

～《AUTO》

システムの開始方法には、AUTO だけを指定できます。また、前回の終了モードによる開始形態の違いについては、「6.2.4 開始形態の決定」を参照してください。

これ以外に差異はありません。

● shmpool_attribute=free

～《free》

システムサービス用共用メモリプールをメモリ上に固定できないため、fixed は指定できません。これ以外に差異はありません。

● user_command= ユーザ環境設定コマンド

～〈パス名〉

5. システム定義

OpenTP1 起動前に、実行するユーザ環境設定コマンドを完全パス名（1 ～ 260 バイト）で指定します。

標準入力待ち状態を発生させるコマンドおよびバッチファイルを指定しないでください。

指定したパス名の中に '!', または '\$' を含む場合、指定した文字列に対して OpenTP1 内部で独自の文字列変換処理を実行します。

これ以外に差異はありません。

● user_command_online= システム開始完了コマンド

～ 〈パス名〉

OpenTP1 が起動を完了するときに実行するシステム開始完了コマンドを完全パス名（1 ～ 260 バイト）で指定します。

コマンド文字列の末尾に '&' を指定した場合のコマンドの非同期実行は、サポートしていません。

ユーザ ID の指定（user_command_online_tp1mgr_id オペランド）は、サポートしていません。

標準入力待ち状態を発生させるコマンドおよびバッチファイルを指定しないでください。

標準出力リダイレクト機能を使用すると、このオペランドに指定するバッチファイル、および実行形式ファイル内で、標準出力・標準エラー出力にメッセージを出力できません。

標準出力・標準エラー出力を使用するバッチファイル、および実行形式ファイルを使用した場合、KFCA01819-E メッセージの出力後に、OpenTP1 がダウンします。標準出力・標準エラー出力を使用するプログラムを指定する場合は、バッチファイル内で @echo off を実行し、標準出力・標準エラー出力を無効にしてから使用してください。

指定したパス名中に '!', または '\$' を含む場合、指定した文字列に対して OpenTP1 内部で独自の文字列変換処理を実行します。

これ以外に差異はありません。

● redirect_file=Y|N

～ 〈N〉

Windows 版 OpenTP1 固有のオペランドです。OpenTP1 配下のプロセスの標準出力・標準エラー出力の内容をファイルに出力するかどうか（標準出力リダイレクト機能を使用するかどうか）を指定します。

Y

標準出力・標準エラー出力の内容をファイルに出力します。OpenTP1 のコンソールウィンドウは表示されません。

N

標準出力・標準エラー出力の内容をファイルに出力しません。

システム共通定義の `pre_port` オペランド、または `rpc_port_base` オペランドを指定した場合、標準出力リダイレクト機能は使用できません。詳細については、「2.2.4 システム定義の関係」を参照してください。

`pre_port` オペランド、または `rpc_port_base` オペランドを指定し、かつこのオペランドに Y を指定した場合、KFCA26520-W メッセージを出力したあと、このオペランドの指定を無視して OpenTP1 の起動を続行します。KFCA26520-W メッセージが出力される場合は、`pre_port` オペランド、または `rpc_port_base` オペランドの指定を削除するか、`redirect_file` オペランドの指定を削除してください。

● `redirect_file_name=` 出力先ファイル名

～〈パス名〉

Windows 版 OpenTP1 固有のオペランドです。標準出力・標準エラー出力の内容を出力するファイル名を絶対パス名で指定します。

世代管理する場合は、ファイル名のあとに "1" または "2" が付加されます。ファイル名を省略した場合は、`%DCDIR%\¥spool¥prelog1`、`%DCDIR%\¥spool¥prelog2` の二つのファイルが作成されます。

● `redirect_file_size=` 出力先ファイルの最大サイズ

～〈符号なし整数〉((0 ～ 65535))《1024》(単位：キロバイト)

Windows 版 OpenTP1 固有のオペランドです。標準出力・標準エラー出力の内容を出力するファイルの最大サイズを、キロバイト単位で指定します。

出力先ファイルは、指定した出力サイズを基に 2 世代管理されます。メッセージの出力内容によっては、出力ファイルのサイズが指定値を超える場合があります。

0 を指定した場合、2 世代管理をしないため、単調増加ファイルになります。

標準出力リダイレクト機能を使用し、同時に多量のメッセージを出力する場合、同時に出力先ファイルを更新してしまい、次回 OpenTP1 の開始時に最新世代のファイルを現用に割り当てることがあります。そのため、このオペランドの指定値は、1 世代を使い切る時間が 1 秒以上掛かるサイズを目安として指定してください。

● `console_output=Y|N`

～《N》

Windows 版 OpenTP1 固有のオペランドです。標準出力・標準エラー出力の内容をコンソールに出力するかどうか (OpenTP1 コンソール出力機能を使用するかどうか) を指定します。

5. システム定義

Y

標準出力・標準エラー出力の内容をコンソールに出力します。

N

標準出力・標準エラー出力の内容をコンソールに出力しません。

このオペランドに Y を指定した場合でも、そのほかのオペランドの指定値によっては、標準出力・標準エラー出力の内容がコンソールに出力されないことがあります。詳細については、「2.2.4 システム定義の関係」を参照してください。

putenv 形式のオペランド

● DCADMDEBUG 0|1

～《1》

デフォルト値が "1" になります。これ以外に差異はありません。

dcputenv 形式のオペランド

● DCCONFPATH 定義ファイルの格納フォルダ

～〈パス名〉

環境変数 DCCONFPATH の指定値は、大文字、小文字が区別されます。必ず、マシン内で同じ文字列となるように設定してください。

環境変数を指定する場合は、\$ではなく、%で環境変数を囲んでください。これ以外に差異はありません。

● DCUAPCONFPATH ユーザサービス定義ファイルの格納フォルダ

～〈パス名〉

環境変数を指定する場合は、\$ではなく、%で環境変数を囲んでください。これ以外に差異はありません。

システム共通定義

ここで説明していないオペランドについては、「5.1(4) システム共通定義の内容」に示す一覧を参照してください。

形式

set 形式

```
[set all_node="ノード名 [:ポート番号] [:high] "
               [, "ノード名 [:ポート番号] [:high] "...] ]
[set name_notify=Y|N]
[set ipc_conn_interval=コネクション確立監視時間]
[set coredump_type=ADD|ONLY]
[set prc_terminate_force=Y|N]
```

説明

set 形式のオペランド

● all_node=" ノード名 [: ポート番号] [:high] " [, " ノード名 [: ポート番号] [:high] " ...]

このオペランドに指定するノード名は、

%SystemRoot%\¥system32¥drivers¥etc¥HOSTS に、IP アドレスと共にあらかじめ指定しておくか、WINS サーバを使用してください。これ以外に差異はありません。

● name_notify=Y|N

～《N》

このオペランドに Y を指定した場合、all_node オペランドに指定したノードの数だけ起動通知処理を行います。

このオペランドに Y を指定し、all_node オペランドの指定値に OpenTP1 未起動のノードが含まれる場合、未起動状態の OpenTP1 がオンライン状態になるまでに、およそ次の時間が必要となるため注意してください。

connect() がタイムアウトするノードが含まれている場合

タイムアウトするノード数×システム共通定義のipc_conn_intervalオペランドの指定値×2（単位：秒）

OpenTP1 が通知できないノードが含まれている場合

通知できないノード数×TCP/IPの再転送動作時間※×2（単位：秒）

注※

"TcpMaxConnectRetransmissions" などのレジストリパラメータによって決定される TCP/IP の再転送動作に必要な時間です。デフォルトの設定では、およそ 1 秒となります。

5. システム定義

● ipc_conn_interval= コネクション確立監視時間

～ 〈符号なし整数〉 ((8 ～ 65535)) 《12》 (単位：秒)

デフォルト値が "12" になります。これ以外に差異はありません。

● coredump_type=ADD|ONLY

～ 《ADD》

システムサーバがダウンした場合に取得する保守資料を強化するかどうかを指定します。

ADD

システムサーバおよびコマンドがダウンした場合に、従来の退避コアファイルに加えて Windows のプロセスダンプファイルを取得し、保守資料を強化します。障害発生時の早期原因究明のため、デフォルト値の適用を推奨します。ただし、このオペランドに ONLY を指定した場合に比べて、次に示す時間が十数秒長くなることがあります。

- dcstop -fd コマンドの実行時間
- システム環境定義の system_terminate_watch_time オペランド指定値が満了した場合の OpenTP1 強制停止の実行時間

ONLY

従来の退避コアファイルだけを取得します。

なお、このオペランドの指定値に関わらず、ユーザサーバがダウンした場合には、従来の退避コアファイルだけを取得します。

● prc_terminate_force=Y|N

～ 《Y》

OpenTP1 強制停止の際に、既定時間内[※]に終了しないプロセスを Win32API の TerminateProcess 関数で強制終了させます。

Y

OpenTP1 強制停止の際に、既定時間内[※]に終了しないプロセスを Win32API の TerminateProcess 関数で強制終了させます。

プロセスの強制終了を実施してもプロセスが残存した場合、KFCA00736-E メッセージを出力します。

N

OpenTP1 強制停止の際に、既定時間内に終了しないプロセスが存在した場合、KFCA00736-E メッセージを出力します。

注※

既定時間とは、KFCA01840-I メッセージ（終了形態：FORCE）を出力してからの

監視時間です。OpenTP1 の停止方法によって、既定時間は次のように異なります。
なお、マシン高負荷の場合は次に示す既定時間を過ぎることがあります。

- `dcstop -f` を実行した場合：約 30 秒
- `dcstop -fd` を実行した場合：約 210 秒
- 開始パラメータに `forcedterminate` を指定して開始した OpenTP1 サービスに対して、`net stop` コマンドを実行した場合：約 30 秒
- システム環境定義の `system_terminate_watch_time` が満了し、OpenTP1 強制停止に切り替わった場合：約 210 秒

注意事項

プロセスの強制終了では、Win32API の `TerminateProcess` 関数を使用します。
`TerminateProcess` 関数の詳細については、Microsoft サポートページをご確認ください。

プロセスサービス定義 (prcsvpath)

ここで説明していないオペランドについては、「5.1(8) プロセスサービス定義の内容」に示す一覧を参照してください。

形式

[prcsvpath パス名]

オプション

●パス名

～〈パス名〉《%DCDIR%¥aplib;%DCDIR%¥bin》

UNIX 版 OpenTP1 との差異は次のとおりです。これ以外に差異はありません。

- ユーザサーバのパス名を変更した場合でも、必ず先頭に %DCDIR%¥bin が付与されます。
- 空白文字を含むパス名は指定できません。空白文字を含むパス名を指定する場合は、8.3 形式の短い名前 (MS-DOS 形式) のパス名に変換してから指定してください。
- XMAP3 Server Runtime Version 5 を使用する場合、"XMAP3 Server Runtime インストールフォルダ ¥bin" を設定してください。

ランタイムライブラリのパス名の指定については、「4.3.3 ランタイムライブラリのパスの設定」を参照してください。

ジャーナルサービス定義

ここで説明していないオペランドについては、「5.1(14) ジャーナルサービス定義の内容」に示す一覧を参照してください。

形式

set 形式

[set jnl_watch_time=ジャーナルサービスの通信応答待ち時間]

説明

set 形式のオペランド

● jnl_watch_time= ジャーナルサービスの通信応答待ち時間

～〈符号なし整数〉((0 ～ 65535))《180》(単位：秒)

このオペランドが有効になる範囲を次に示します。

- jnlopnfg コマンド
- jnlclsfg コマンド
- jnlchgfg コマンド
- jnlunlfg コマンド
- jnlswpfg コマンド※
- ジャーナルファイル障害、満杯時のスワップ要求通信応答待ち時間
- ジャーナルファイルの I/O 処理完了受信応答待ち時間

注※

jnlswpfg コマンドの通信応答待ち時間は、このオペランドに指定した値の 2 倍になります。

これ以外に差異はありません。

クライアントサービス定義

ここで説明していないオペランドについては、「5.1(24) クライアントサービス定義の内容」に示す一覧を参照してください。

形式

set 形式

[set message_store_buflen=メッセージ格納バッファプール長]

説明

set 形式のオペランド

● message_store_buflen= メッセージ格納バッファプール長

～〈符号なし整数〉((1024 ～ 31457280))《8196》(単位：バイト)

OpenTP1 がオンラインの状態でこのオペランドの指定値を変更した場合、KFC A00850-E メッセージが出力されることがあります。

これは、OpenTP1 起動時に、このオペランドの指定値を基にあらかじめ確保している内部リソースが、指定値の変更によって不足したためです。一度 OpenTP1 を停止させ、再起動することでこの現象を回避できます。

なお、このオペランドの指定値の変更によって、必要となる静的共用メモリのサイズが変化するため、注意してください。メモリサイズの算出式については、マニュアル「OpenTP1 システム定義」を参照してください。

rap リスナーサービス定義

ここで説明していないオペランドについては、「5.1(32) rap リスナーサービス定義の内容」に示す一覧を参照してください。

形式

set 形式

```
[set rap_parallel_server=rapサーバ数]
[set nice=プロセスの優先順位の変更]
[set rap_listen_backlog=接続待ちキューに保留できる最大要求数]
```

説明

set 形式のオペランド

● rap_parallel_server=rap サーバ数

～〈符号なし整数〉((1 ～ 1024))《1》

OpenTP1 がオンラインの状態でのオペランドの指定値を変更した場合、KFCA00850-E メッセージが出力されることがあります。

これは、OpenTP1 起動時に、このオペランドの指定値を基にあらかじめ確保している内部リソースが、指定値の変更によって不足したためです。一度 OpenTP1 を停止させ、再起動することでこの現象を回避できます。

なお、このオペランドの指定値を変更すると、必要となる静的共用メモリのサイズが変化します。メモリサイズの算出式については、マニュアル「OpenTP1 システム定義」を参照してください。

● nice= プロセスの優先順位の変更

～〈符号なし整数〉((0 ～ 39))《0》

指定値の意味が異なります。Windows 版 OpenTP1 では、このオペランドの指定値は次のとおりに解釈されます。

nice オペランドの指定値	適用される優先順位
0 ～ 9	THREAD_PRIORITY_NORMAL
10 ～ 19	THREAD_PRIORITY_BELOW_NORMAL
20 ～ 29	THREAD_PRIORITY_LOWEST
30 ～ 39	THREAD_PRIORITY_IDLE

5. システム定義

● rap_listen_backlog= 接続待ちキューに保留できる最大要求数

～〈符号なし整数〉(5 ～ OS のデフォルト値)《OS のデフォルト値》

指定範囲，デフォルト値が異なります。OS のデフォルト値は OS によって異なります。OS のデフォルト値（実数値）については、「リリースノート」を参照してください。これ以外に差異はありません。

ユーザサービスデフォルト定義

ここで説明していないオペランドについては、「5.1(41) ユーザサービスデフォルト定義の内容」に示す一覧を参照してください。

形式

set 形式

```
[set nice=プロセスの優先順位の変更]
[set message_store_buflen=メッセージ格納バッファプール長]
[set message_cell_size=スケジュールメッセージ格納セル長]
[set prc_abort_signal=アボート用シグナル番号]
[set process_privilege_restrict=Y|N]
[set process_privilege_name="特権名" [, "特権名"...] ]
[set ipc_conn_interval=コネクション確立監視時間]
```

putenv 形式

```
{{ [putenv 環境変数名 環境変数値] }}
```

dcputenv 形式

```
{{ [dcputenv 環境変数名 環境変数値] }}
```

説明

set 形式のオペランド

● nice= プロセスの優先順位の変更

～〈符号なし整数〉((0 ～ 39))《0》

指定値の意味が異なります。Windows 版 OpenTP1 では、このオペランドの指定値は次のとおりに解釈されます。

nice オペランドの指定値	適用される優先順位
0 ～ 9	THREAD_PRIORITY_NORMAL
10 ～ 19	THREAD_PRIORITY_BELOW_NORMAL
20 ～ 29	THREAD_PRIORITY_LOWEST
30 ～ 39	THREAD_PRIORITY_IDLE

● message_store_buflen= メッセージ格納バッファプール長

～〈符号なし整数〉((1024 ～ 31457280))《4096》(単位: バイト)

5. システム定義

OpenTP1 がオンラインの状態でのオペランドの指定値を変更した場合、KFCA00850-E メッセージが出力されることがあります。

これは、OpenTP1 起動時に、このオペランドの指定値を基にあらかじめ確保している内部リソースが、指定値の変更によって不足したためです。一度 OpenTP1 を停止させ、再起動することでこの現象を回避できます。

なお、このオペランドの指定値の変更によって、必要となる静的共用メモリのサイズが変化するため、注意してください。メモリサイズの算出式については、マニュアル「OpenTP1 システム定義」を参照してください。

● message_cell_size= スケジュールメッセージ格納セル長

～ 〈符号なし整数〉 ((512 ～ 31457280)) 《512》 (単位：バイト)

OpenTP1 がオンラインの状態でのオペランドの指定値を変更した場合、KFCA00850-E メッセージが出力されることがあります。

これは、OpenTP1 起動時に、このオペランドの指定値を基にあらかじめ確保している内部リソースが、指定値の変更によって不足したためです。一度 OpenTP1 を停止させ、再起動することでこの現象を回避できます。

なお、このオペランドの指定値の変更によって、必要となる静的共用メモリのサイズが変化するため、注意してください。メモリサイズの算出式については、マニュアル「OpenTP1 システム定義」を参照してください。

● prc_abort_signal= アボート用シグナル番号

～ 〈符号なし整数〉 ((3, 6, 15)) 《3》

3, 6, 15 の中から、シグナル番号を指定してください。これ以外に差異はありません。

● process_privilege_restrict=Y|N

～ 《N》

ユーザサービス定義の process_privilege_restrict オペランドの省略時解釈値を指定します。

● process_privilege_name=" 特権名 " [, " 特権名 "...]

～ 〈文字列〉

ユーザサービス定義の process_privilege_name オペランドの省略時解釈値を指定します。

● ipc_conn_interval= コネクション確立監視時間

～ 〈符号なし整数〉 ((8 ～ 65535)) 《12》 (単位：秒)

デフォルト値が "12" になります。これ以外に差異はありません。

putenv 形式のオペランド

●環境変数名 環境変数値

～〈文字列〉

UNIX 版 OpenTP1 との差異は次のとおりです。これ以外に差異はありません。

- 環境変数 LANG を putenv 形式でユーザサービス定義またはユーザサービスデフォルト定義に指定する場合は、システム共通定義と同じ内容を指定してください。システム共通定義に定義していない場合は、C 以外を指定しないでください。
- 環境変数 PATH を putenv 形式でユーザサービス定義またはユーザサービスデフォルト定義に指定する場合は、OpenTP1 のコマンド、UAP のロードモジュール、および UAP が使用する DLL の場所を含めて指定してください。

dcputenv 形式のオペランド

●環境変数名 環境変数値

～〈文字列〉

環境変数を指定する場合は、\$ ではなく、% で環境変数を囲んでください。これ以外に差異はありません。

ユーザサービス定義

ここで説明していないオペランドについては、「5.1(42) ユーザサービス定義の内容」に示す一覧を参照してください。

形式

set 形式

```
set module="実行形式プログラム名"  
[set nice=プロセスの優先順位の変更]  
[set message_store_bufllen=メッセージ格納バッファプール長]  
[set message_cell_size=スケジュールメッセージ格納セル長]  
[set prc_abort_signal=アボート用シグナル番号]  
[set process_privilege_restrict=Y|N]  
[set process_privilege_name="特権名" [, "特権名"...] ]  
[set ipc_conn_interval=コネクション確立監視時間]
```

putenv 形式

```
{ { [putenv 環境変数名 環境変数値] } }
```

dcputenv 形式

```
{ { [dcputenv 環境変数名 環境変数値] } }
```

説明

set 形式のオペランド

● module=" 実行形式プログラム名 "

～ 〈1 ～ 14 文字の識別子〉

実行形式プログラム名には、「.exe」を指定しないでください。例えば、実行形式プログラム名が spp.exe の場合は、set module ="spp" と指定してください。これ以外に差異はありません。

● nice= プロセスの優先順位の変更

～ 〈符号なし整数〉 ((0 ～ 39))

指定値の意味が異なります。Windows 版 OpenTP1 では、このオペランドの指定値は次のとおりに解釈されます。

nice オペランドの指定値	適用される優先順位
0 ～ 9	THREAD_PRIORITY_NORMAL

nice オペランドの指定値	適用される優先順位
10 ～ 19	THREAD_PRIORITY_BELOW_NORMAL
20 ～ 29	THREAD_PRIORITY_LOWEST
30 ～ 39	THREAD_PRIORITY_IDLE

● message_store_buflen= **メッセージ格納バッファプール長**

～ 〈符号なし整数〉 ((1024 ～ 31457280)) (単位：バイト)

OpenTP1 がオンラインの状態でのオペランドの指定値を変更した場合、KFCA00850-E メッセージが出力されることがあります。

これは、OpenTP1 起動時に、このオペランドの指定値を基にあらかじめ確保している内部リソースが、指定値の変更によって不足したためです。一度 OpenTP1 を停止させ、再起動することでこの現象を回避できます。

なお、このオペランドの指定値の変更によって、必要となる静的共用メモリのサイズが変化するため、注意してください。メモリサイズの算出式については、マニュアル「OpenTP1 システム定義」を参照してください。

● message_cell_size= **スケジュールメッセージ格納セル長**

～ 〈符号なし整数〉 ((512 ～ 31457280)) (単位：バイト)

OpenTP1 がオンラインの状態でのオペランドの指定値を変更した場合、KFCA00850-E メッセージが出力されることがあります。

これは、OpenTP1 起動時に、このオペランドの指定値を基にあらかじめ確保している内部リソースが、指定値の変更によって不足したためです。一度 OpenTP1 を停止させ、再起動することでこの現象を回避できます。

なお、このオペランドの指定値の変更によって、必要となる静的共用メモリのサイズが変化するため、注意してください。メモリサイズの算出式については、マニュアル「OpenTP1 システム定義」を参照してください。

● prc_abort_signal= **アボート用シグナル番号**

～ 〈符号なし整数〉 ((3, 6, 15))

3, 6, 15 の中から、シグナル番号を指定してください。これ以外に差異はありません。

● process_privilege_restrict=Y|N

～ 〈N〉

ユーザーサーバに対して Windows の特権を制限するかどうかを指定します。

Y

5. システム定義

ユーザサーバは、SeChangeNotifyPrivilege 以外の特権が制限されます。

Y を指定した場合は、process_privilege_name オペランドで指定した特権をユーザサーバに付与できます。

N

ユーザサーバは、サービスログオンアカウントと同じ特権を持ちます。

● process_privilege_name= **特権名** [" **特権名** "...]

～〈文字列〉

ユーザサーバに付与する Windows の特権名（例：SeSecurityPrivilege）を指定します。

このオペランドは、process_privilege_restrict オペランドに Y を指定した場合に有効です。

このオペランドに指定した特権をサービスログオンアカウントが所持していない場合は、その特権を付与できません。特権を付与する場合は、必要な特権かどうかを十分に検討してから付与してください。

● ipc_conn_interval= **コネクション確立監視時間**

～〈符号なし整数〉((8 ～ 65535))《12》(単位：秒)

デフォルト値が "12" になります。これ以外に差異はありません。

putenv 形式

●環境変数名 環境変数値

～〈文字列〉

UNIX 版 OpenTP1 との差異は次のとおりです。これ以外に差異はありません。

- 環境変数 LANG を putenv 形式でユーザサービス定義またはユーザサービスデフォルト定義に指定する場合は、システム共通定義と同じ内容を指定してください。システム共通定義に定義していない場合は、C 以外を指定しないでください。
- 環境変数 PATH を putenv 形式でユーザサービス定義またはユーザサービスデフォルト定義に指定する場合は、OpenTP1 のコマンド、UAP のロードモジュール、および UAP が使用する DLL の場所を含めて指定してください。

dcputenv 形式

●環境変数名 環境変数値

～〈文字列〉

環境変数を指定する場合は、\$ ではなく、% で環境変数を囲んでください。これ以外に差異はありません。

注意事項

OpenTP1 がオンラインの状態ユーザサービス定義を新規に追加した場合、KFCA00850-E メッセージが出力されることがあります。

これは、OpenTP1 起動時に、この定義内容に基づきあらかじめ確保している内部リソースが、定義の新規追加によって不足したためです。一度 OpenTP1 を停止させ、再起動することでこの現象を回避できます。

なお、ユーザサービス定義の追加によって、必要となる静的共用メモリのサイズが変化します。メモリサイズの算出式については、マニュアル「OpenTP1 システム定義」を参照してください。

システムサービス情報定義

ここで説明していないオペランドについては、「5.1(47) システムサービス情報定義の内容」に示す一覧を参照してください。

形式

set 形式

```
set module="MCF実行形式プログラム名"
```

説明

set 形式のオペランド

● module="MCF 実行形式プログラム名 "

～ 〈1 ～ 8 文字の識別子〉

実行形式プログラム名には、「.exe」を指定しないでください。例えば、実行形式プログラム名が mcfu01.exe の場合は、set module ="mcfu01" と指定してください。これ以外に差異はありません。

6

運用と操作

Windows 版 OpenTP1 は，UNIX 版 OpenTP1 と比べて，システムの運用方法や，操作方法が異なります。また，Windows 版 OpenTP1 では，一部の操作を GUI でも実施できます。この章では，Windows 版 OpenTP1 固有の運用と操作，および Windows 版 OpenTP1 の GUI について説明します。UNIX 版 OpenTP1 と共通の内容については，OpenTP1 シリーズのマニュアルを参照してください。

6.1 Windows 版 OpenTP1 での運用と操作の概要

6.2 OpenTP1 の開始

6.3 OpenTP1 の停止

6.4 マルチ OpenTP1 の運用

6.5 シナリオテンプレートを利用したシステムの運用

6.6 障害対策

6.7 Windows 版 OpenTP1 の GUI

6.1 Windows 版 OpenTP1 での運用と操作の概要

Windows 版 OpenTP1 は、Windows のサービスとして動作している点が UNIX 版 OpenTP1 と大きく異なります。Windows へのサービスの登録・削除はすべてインストーラが行います（マルチ OpenTP1 の場合は `dcsetupml` コマンドで行います）。サービス名称は「OpenTP1」です（マルチ OpenTP1 の場合は「OpenTP1_XXXX」（XXXX はユーザが指定した識別子））。

このほかに、Windows 版 OpenTP1 は、UNIX 版 OpenTP1 と比べて、システムの運用方法や操作方法が異なります。この章では、Windows 版 OpenTP1 固有のシステムの運用方法や操作方法について説明します。UNIX 版 OpenTP1 と共通の内容については、マニュアル「OpenTP1 運用と操作」を参照してください。

この章で説明する Windows 版 OpenTP1 での運用と操作の概要を、次に示します。

表 6-1 Windows 版 OpenTP1 での運用と操作の概要

運用と操作	概要	参照先
OpenTP1 の開始	OpenTP1 の開始方法について説明しています。 Windows 版 OpenTP1 では、OpenTP1 の開始方法を次の三つから選択できます。 • Windows の [サービス] ダイアログ • コマンド • GUI	6.2
OpenTP1 の停止	OpenTP1 の停止方法について説明しています。 Windows 版 OpenTP1 では、OpenTP1 の停止方法を次の三つから選択できます。 • Windows の [サービス] ダイアログ • コマンド • GUI	6.3
マルチ OpenTP1 の運用	Windows 版 OpenTP1 でのマルチ OpenTP1 の運用について説明しています。Windows 版 OpenTP1 では、コマンドの入力環境と運用するためのコマンドが UNIX 版 OpenTP1 と異なります。	6.4
シナリオテンプレートを利用したシステムの運用	Windows 版 OpenTP1 でのシナリオテンプレートを利用した運用について説明しています。	6.5
障害対策	Windows 版 OpenTP1 での障害対策について説明しています。	6.6
Windows 版 OpenTP1 の GUI	Windows 版 OpenTP1 で提供している GUI について説明しています。	6.7

6.2 OpenTP1 の開始

Windows 版 OpenTP1 を開始するには、次の方法があります。

- Windows の [サービス] ダイアログ
- コマンド
- GUI

ここでは、それぞれの開始方法について説明します。また、Windows 版 OpenTP1 での開始形態の決定についても説明します。

6.2.1 Windows の [サービス] ダイアログによる OpenTP1 の開始

Windows の [サービス] ダイアログでは、OpenTP1 は次の方法で開始できます。

- 手動開始
- 自動開始
- 強制正常開始

ここでは、それぞれの開始方法について説明します。

参考

Windows の [サービス] ダイアログは、[コントロールパネル] から開きます。

(1) OpenTP1 の手動開始

OpenTP1 を手動で開始させる手順を次に示します。

1. Windows の [サービス] ダイアログを開きます。
2. 表示されたサービスの一覧から、「OpenTP1」（マルチ OpenTP1 の場合は「OpenTP1_XXXX」（XXXX はユーザが指定した識別子））を選択します。
3. [開始] ボタンをクリックします。
[サービスコントロール] ダイアログボックスが表示され、OpenTP1 が開始します。
4. [OK] ボタンをクリックします。

(2) OpenTP1 の自動開始

Windows 起動時には、OpenTP1 を自動開始させます。OpenTP1 を自動で開始させる手順を次に示します。

1. Windows の [サービス] ダイアログを開きます。

2. 表示されたサービスの一覧から、「OpenTP1」（マルチ OpenTP1 の場合は「OpenTP1_XXXX」（XXXX はユーザが指定した識別子））を選択し、ダブルクリックします。
3. 「スタートアップの種類」を「自動」に変更します。

(3) OpenTP1 の強制正常開始

OpenTP1 を強制正常開始させる手順を次に示します。

1. Windows の [サービス] ダイアログを開きます。
2. 表示されたサービスの一覧から、「OpenTP1」（マルチ OpenTP1 の場合は「OpenTP1_XXXX」（XXXX はユーザが指定した識別子））を選択し、ダブルクリックします。
3. 「開始パラメタ」テキストボックスに、"normal" を入力します。
4. [開始] ボタンをクリックします。

(4) 注意事項

- 「デスクトップとの対話をサービスに許可」チェックボックスを OFF にした状態、または OpenTP1 のサービスアカウントをローカルシステムアカウント以外に変更した状態で OpenTP1 を開始した場合、デスクトップヒープの不足によって、システムダウンするおそれがあります。そのため、必要に応じてレジストリ操作でデスクトップヒープを拡張してください。設定方法については、Microsoft サポートページをご確認ください。なお、「デスクトップとの対話をサービスに許可」チェックボックスは、Windows の [サービス] ダイアログで OpenTP1 のサービス名を右クリックし、プロパティを選択して表示される [OpenTP1 のプロパティ] ダイアログボックスの [ログオン] タブにあります。
- 「デスクトップとの対話をサービスに許可」チェックボックスを OFF にした状態で OpenTP1 を開始すると、メッセージ KFCA26524-W がイベントビューアに出力されます。このメッセージが出力されたあとも OpenTP1 は通常どおり起動しますが、いったん OpenTP1 を終了させたあと、「デスクトップとの対話をサービスに許可」チェックボックスを ON にして再起動してください。
- Version 6 以降の OpenTP1 では、デフォルトではコンソールが表示されません。コンソールを表示するには、OpenTP1 コンソール出力機能を使用してください。OpenTP1 コンソール出力機能については、「2.2.3 OpenTP1 コンソール出力機能」を参照してください。

6.2.2 コマンドによる OpenTP1 の開始

コマンドで OpenTP1 を開始する場合、コマンドプロンプトから次のどちらかのコマンドを実行します。

● net start

引数には「OpenTP1」を指定します。マルチ OpenTP1 の場合は、引数に「OpenTP1_XXXX」（XXXX はユーザが指定した識別子）を指定してください。

● ntbstart

マルチ OpenTP1 の場合は「dcmakecon XXXX」（XXXX はユーザが指定した識別子）で作成したコンソールのコマンドプロンプトから実行します。

Windows 版 OpenTP1 では、dcstart コマンドによる OpenTP1 の起動はできません。

また、コマンドプロンプトから OpenTP1 を強制的に正常開始させるには、ntbstart -n コマンドを実行してください。マルチ OpenTP1 の場合は「dcmakecon XXXX」（XXXX はユーザが指定した識別子）で作成したコンソールのコマンドプロンプトから実行します。

ntbstart コマンドについては「7. 運用コマンド」の「ntbstart (OpenTP1 の起動)」を、dcmakecon コマンドについては、「7. 運用コマンド」の「dcmakecon (マルチ OpenTP1 操作コンソールの作成)」を参照してください。net start コマンドについては、Windows のマニュアルを参照してください。

6.2.3 GUI による OpenTP1 の開始

Windows 版 OpenTP1 で提供している GUI を使用して、OpenTP1 を開始できます。

Windows 版 OpenTP1 の GUI の詳細については、「6.7 Windows 版 OpenTP1 の GUI」を参照してください。

6.2.4 開始形態の決定

開始方法と開始モードの組み合わせを**開始形態**といいます。

開始形態は、前回の終了モードと、システム環境定義の mode_conf オペランドの指定値によって決定されます。Windows 版 OpenTP1 は、システム環境定義の mode_conf オペランドで AUTO だけを指定できます（MANUAL1、および MANUAL2 は指定できません）。このため、開始形態の決定条件が UNIX 版 OpenTP1 と異なります。

Windows 版 OpenTP1 の開始形態を、次に示します。

表 6-2 Windows 版 OpenTP1 の開始形態

開始形態決定の条件		開始形態	
前回の終了モード	mode_conf の指定値	開始方法	開始モード
正常終了	AUTO	手動※1	正常開始
強制正常終了	AUTO	手動※1	正常開始

6. 運用と操作

開始形態決定の条件		開始形態	
前回の終了モード	mode_conf の指定値	開始方法	開始モード
計画停止 A	AUTO	手動※ 1	再開始※ 2
計画停止 B	AUTO	手動※ 1	再開始※ 2
強制停止	AUTO	手動※ 1	再開始※ 2
異常終了	AUTO	自動	再開始

注※ 1

Windows 起動時は自動開始となります。

注※ 2

Windows の [サービス] ダイアログ、コマンド、および GUI を使用して、強制的に正常開始させることもできます。

なお、Windows 版 OpenTP1 では、dcstart コマンドはサポートしていません。このため、dcstart -n コマンドで強制的に正常開始させることはできません。

6.3 OpenTP1 の停止

Windows 版 OpenTP1 を停止するには、次の方法があります。

- Windows の [サービス] ダイアログ
- コマンド
- GUI

ここでは、それぞれの停止方法について説明します。

6.3.1 Windows の [サービス] ダイアログによる OpenTP1 の停止

Windows の [サービス] ダイアログで OpenTP1 を停止させる手順を次に示します。

1. Windows の [サービス] ダイアログを開きます。
2. 表示されたサービスの一覧から、「OpenTP1」（マルチ OpenTP1 の場合は「OpenTP1_XXXX」（XXXX はユーザが指定した識別子））を選択します。
3. [停止] ボタンをクリックします。
OpenTP1 は、正常終了として停止します。

6.3.2 コマンドによる OpenTP1 の停止

コマンドで OpenTP1 を停止する場合、コマンドプロンプトから次のどちらかのコマンドを実行します。OpenTP1 が正常終了で停止します。

● net stop

引数には「OpenTP1」を指定します。マルチ OpenTP1 の場合は、引数に「OpenTP1_XXXX」（XXXX はユーザが指定した識別子）を指定してください。

このコマンドを実行しても何らかの原因で正常終了できない場合、OpenTP1 は強制正常終了で停止されます。強制正常終了もできない場合は、強制停止で停止されます。強制停止した場合、次に OpenTP1 を開始するときには再開始（リラン）となるため、注意してください。

● dctestop

マルチ OpenTP1 の場合は「dctestop XXXX」（XXXX はユーザが指定した識別子）で作成したコンソールのコマンドプロンプトから実行します。

dctestop コマンドについては「7. 運用コマンド」の「運用コマンドの一覧」を、dctestop コマンドについては、「7. 運用コマンド」の「dctestop（マルチ OpenTP1 操作コンソールの作成）」を参照してください。net stop コマンドについては、Windows のマニュアルを参照してください。

6.3.3 GUI による OpenTP1 の停止

Windows 版 OpenTP1 で提供している GUI を使用して、OpenTP1 を停止できます。

Windows 版 OpenTP1 の GUI の詳細については、「6.7 Windows 版 OpenTP1 の GUI」を参照してください。

6.3.4 OpenTP1 の停止時の注意事項

OpenTP1 にシステム障害が発生し、システム再開始処理中に次の方法で OpenTP1 を停止すると、OpenTP1 のサービスのステータスが「停止処理中」のままになる場合があります。

- Windows の [サービス] ダイアログによる OpenTP1 の停止
- コマンドによる OpenTP1 の停止

この場合、障害要因の確認および対策後に、`dcstop` コマンドを実行し、OpenTP1 を停止させてください。マルチ OpenTP1 の場合は、`dcmakecon` コマンドで作成したコンソールから `dcstop` コマンドを実行してください。`dcstop` コマンドについては「7. 運用コマンド」の「運用コマンドの一覧」を、`dcmakecon` コマンドについては、「7. 運用コマンド」の「`dcmakecon` (マルチ OpenTP1 操作コンソールの作成)」を参照してください。

6.4 マルチ OpenTP1 の運用

ここでは、マルチ OpenTP1 の運用について、UNIX 版 OpenTP1 との差異を説明します。Windows 版 OpenTP1 では、コマンドの入力環境と、運用するためのコマンドに差異があります。

UNIX 版 OpenTP1 と共通の部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

(1) コマンドの入力環境

Windows 版 OpenTP1 と UNIX 版 OpenTP1 では、コマンドの入力方法が環境別に次のように異なります。

- オリジナルの OpenTP1 環境の場合
コマンドプロンプトからコマンドを入力し、OpenTP1 のコマンドを動作させます。オリジナルの OpenTP1 を操作する場合は、コマンドプロンプトから OpenTP1 コマンドを実行することも、dcmakecon コマンドで作成したコンソールから OpenTP1 コマンドを実行することもできます。
- マルチ OpenTP1 環境の場合
dcmakecon コマンドを実行して、特定の OpenTP1 を動作させるためのコンソールを作成します。マルチ OpenTP1 環境でコマンドを実行するときは、このコンソールから実行します。
マルチ OpenTP1 を操作するには、dcmakecon コマンドで作成したコンソール内でコマンドを実行します。ただし、「net start OpenTP1_XXXX」や「net stop OpenTP1_XXXX」（XXXX はユーザが指定した識別子）は、通常のコマンドプロンプトからも実行できます。

OpenTP1 の起動方法については「6.2 OpenTP1 の開始」を、停止方法については「6.3 OpenTP1 の停止」を参照してください。

(2) マルチ OpenTP1 を運用するためのコマンド

マルチ OpenTP1 を運用するために、次のコマンドを示します。

- dcdls：マルチ OpenTP1 のセットアップ状態を確認します。
- dcmakecon：マルチ OpenTP1 操作コンソールを作成します。
- dcsetupml：マルチ OpenTP1 のセットアップ／アンセットアップを実行します。

それぞれのコマンドの詳細については、「7. 運用コマンド」の「Windows 版 OpenTP1 固有のコマンドの詳細」を参照してください。

6.5 シナリオテンプレートを利用したシステムの運用

ここでは、シナリオテンプレートを利用したシステムの運用について、UNIX 版 OpenTP1 との差異を説明します。UNIX 版 OpenTP1 と共通の部分については、マニュアル「OpenTP1 運用と操作」の JP1 連携時の運用についての説明を参照してください。

6.5.1 シナリオの登録

Windows 版 OpenTP1 では、シナリオの登録のうち、シナリオテンプレート定義ファイルの格納場所が UNIX 版 OpenTP1 と異なります。UNIX 版 OpenTP1 と共通の部分については、マニュアル「OpenTP1 運用と操作」のシナリオの登録についての説明を参照してください。

シナリオテンプレート定義ファイル（TP1_ServerBase.sjis.xml）は、次のフォルダに格納されています。

OpenTP1のインストールフォルダ¥jp1_template¥ScenarioTemplate¥

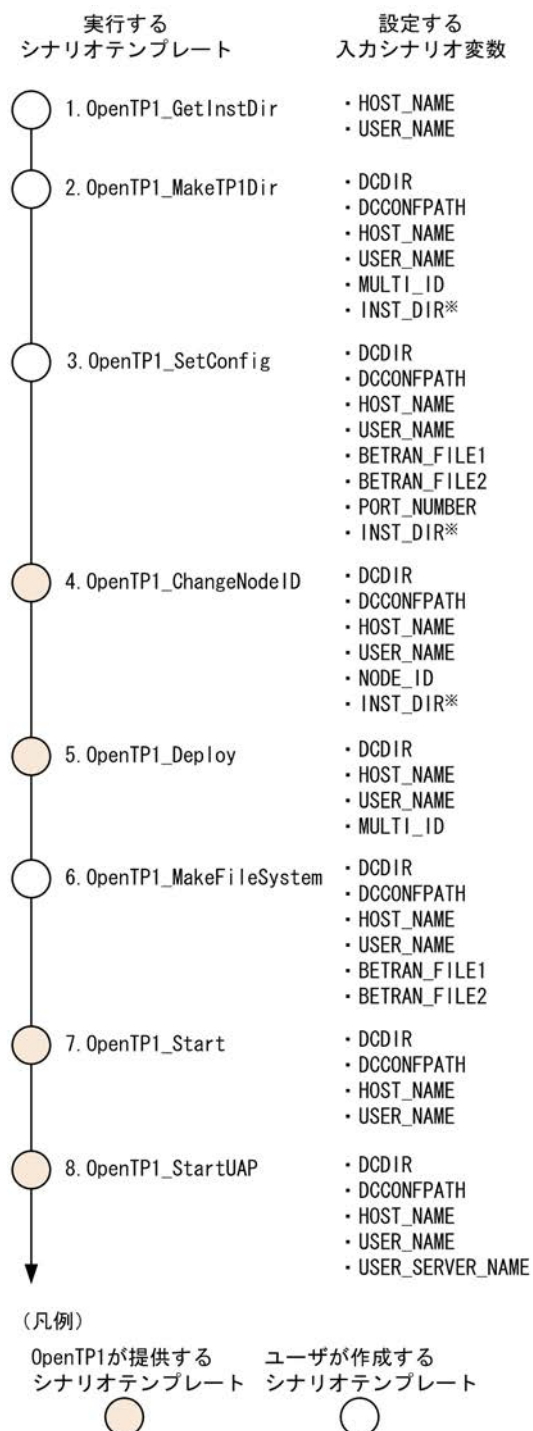
6.5.2 スケールアウトの運用

Windows 版 OpenTP1 では、スケールアウトの運用のスケールアウトのシナリオ構成のうち、新しい OpenTP1 ノードを追加するシナリオが UNIX 版 OpenTP1 と異なります。UNIX 版 OpenTP1 と共通の部分については、マニュアル「OpenTP1 運用と操作」のスケールアウトの運用についての説明を参照してください。

(1) 新しい OpenTP1 ノードを追加するシナリオ

新しい OpenTP1 ノードを追加するシナリオの実行順序を、次に示します。

図 6-1 新しい OpenTP1 ノードを追加するシナリオの実行順序



注※

6. 運用と操作

INST_DIR は、OpenTP1_GetInstDir から引き継がれる変数です。

各シナリオテンプレートの説明を次の表に示します。図中の番号と表中の番号は対応しています。

表 6-3 シナリオテンプレートの名称と説明

図中の番号	シナリオテンプレートの名称	説明
1	OpenTP1_GetInstDir ※ 1	OpenTP1 インストールフォルダの取得
2	OpenTP1_MakeTP1Dir ※ 1	OpenTP1 フォルダの作成
3	OpenTP1_SetConfig ※ 1	OpenTP1 の定義設定
4	OpenTP1_ChangeNodeID ※ 2	ノード ID の設定
5	OpenTP1_Deploy ※ 2	OpenTP1 の登録
6	OpenTP1_MakeFileSystem ※ 1	OpenTP1 ファイルシステムの作成
7	OpenTP1_Start ※ 2	OpenTP1 の起動
8	OpenTP1_StartUAP ※ 2	UAP の起動

注※ 1

ユーザが作成します。OpenTP1 が提供するサンプルについては、マニュアル「OpenTP1 運用と操作」のサンプルシナリオテンプレートの利用についての説明を参照してください。

注※ 2

OpenTP1 が提供します。詳細については、マニュアル「OpenTP1 運用と操作」のシナリオテンプレートの詳細についての説明を参照してください。また、OpenTP1_Deploy については、「6.5.4(1) OpenTP1_Deploy」を参照してください。

6.5.3 サンプルシナリオテンプレートの利用

OpenTP1 では、スケールアウトのシナリオテンプレートのうち、新しいノードを追加するシナリオのサンプルシナリオテンプレートを提供しています。

Windows 版 OpenTP1 では、マルチ OpenTP1 に関するサンプルシナリオテンプレートの一部が UNIX 版 OpenTP1 と異なります。UNIX 版 OpenTP1 と共通する部分については、マニュアル「OpenTP1 運用と操作」のサンプルシナリオテンプレートの利用についての説明を参照してください。

Windows 版 OpenTP1 のサンプルシナリオテンプレート (OpenTP1_ScenarioScaleout) では、マルチ OpenTP1 環境またはインストーラでインストールしたオリジナルの OpenTP1 環境を構築します。

(1) サンプルシナリオテンプレートの名称と処理

サンプルシナリオテンプレートの名称と処理を、次に示します。

表 6-4 サンプルシナリオテンプレートの名称と処理

項番	サンプルシナリオテンプレートの名称	処理	参照する入力シナリオ変数
1	OpenTP1_GetInstDir	OpenTP1 インストールフォルダの取得	HOST_NAME USER_NAME
2	OpenTP1_MakeTP1Dir	OpenTP1 フォルダの作成	DCDIR DCCONFPATH HOST_NAME USER_NAME MULTI_ID INST_DIR ※
3	OpenTP1_SetConfig	OpenTP1 の定義設定	DCDIR DCCONFPATH HOST_NAME USER_NAME BETRAN_FILE1 BETRAN_FILE2 PORT_NUMBER INST_DIR ※
4	OpenTP1_ChangeNodeID	ノード ID の設定	DCDIR DCCONFPATH HOST_NAME USER_NAME NODE_ID INST_DIR ※
5	OpenTP1_Deploy	OpenTP1 の登録	DCDIR HOST_NAME USER_NAME MULTI_ID
6	OpenTP1_MakeFileSystem	OpenTP1 ファイルシステムの作成	DCDIR DCCONFPATH HOST_NAME USER_NAME BETRAN_FILE1 BETRAN_FILE2
7	OpenTP1_Start	OpenTP1 の起動	DCDIR DCCONFPATH HOST_NAME USER_NAME
8	OpenTP1_StartUAP	UAP（サンプルプログラム）の起動	DCDIR DCCONFPATH HOST_NAME USER_NAME USER_SERVER_NAME

注※

入力シナリオ変数 INST_DIR は、OpenTP1_GetInstDir によって取得され、その後の処理で使用するシナリオテンプレートに引き継がれるため、ユーザが入力する必要はありません。

サンプルシナリオテンプレートの詳細についてそれぞれ説明します。以降の番号は、表

6. 運用と操作

中の項番と対応しています。

1. OpenTP1_GetInstDir (OpenTP1 インストールフォルダの取得)

環境変数 DCDIR を、OpenTP1 のインストールフォルダとして取得します。また、取得した OpenTP1 インストールフォルダを、出力シナリオ変数 OUT_INST_DIR に引き継ぎます。

2. OpenTP1_MakeTP1Dir (OpenTP1 フォルダの作成)

OpenTP1 フォルダ作成バッチファイル (OpenTP1 インストールフォルダ ¥jp1_template¥examples¥tools¥dcjmk_dcdir.bat) を使用して、入力シナリオ変数 DCDIR で設定した OpenTP1 フォルダを作成します。ただし、オリジナルの OpenTP1 の場合は、フォルダは作成済みであるため、バッチファイルで処理しません。

なお、Windows 版 OpenTP1 では、ユーザ、グループ、およびモードの設定は行いません。

3. OpenTP1_SetConfig (OpenTP1 の定義設定)

OpenTP1 環境設定バッチファイル (OpenTP1 インストールフォルダ ¥jp1_template¥examples¥tools¥dcjset_conf.bat) を使用して、OpenTP1 のシステム定義およびロードモジュールを、次に示す順序で設定します。

- 次のフォルダを作成します。
マルチ OpenTP1 の場合：%DCDIR%¥aplib, %DCDIR%¥jp1_template¥tools, および %DCCONFPATH%
オリジナルの OpenTP1 の場合：%DCDIR%¥jp1_template¥tools
- 作成したフォルダ下に、OpenTP1 インストールフォルダ ¥jp1_template¥examples フォルダ下の ap lib, conf, および tools をコピーします。
- システム定義のノード固有情報（ファイルシステムのパス名など）を設定します。

4. OpenTP1_ChangeNodeID (ノード ID の設定)

OpenTP1 のノード ID を入力シナリオ変数 NODE_ID の設定値で設定します。

5. OpenTP1_Deploy (OpenTP1 の登録)

マルチ OpenTP1 の場合は、dcsetupml コマンドを実行してマルチ OpenTP1 をセットアップします。

オリジナルの OpenTP1 の場合は、すでに登録されているため処理しません。

6. OpenTP1_MakeFileSystem (OpenTP1 ファイルシステムの作成)

OpenTP1 ファイル作成バッチファイル

(%DCDIR%¥jp1_template¥examples¥tools¥dcj_mkfs.bat) を使用して、OpenTP1 ファイルシステムおよび OpenTP1 ファイルを作成します。OpenTP1 ファイルシステム名は、入力シナリオ変数 BETRAN_FILE1, および BETRAN_FILE2 の設定値になります。

7. OpenTP1_Start (OpenTP1 の起動)

OpenTP1 を起動します。

8. OpenTP1_StartUAP (UAP (サンプルプログラム) の起動)

入力シナリオ変数 USER_SERVER_NAME に設定されたユーザサーバの UAP を起

動します。

(2) サンプルシナリオテンプレートの登録

サンプルシナリオテンプレートを実行するには、サンプルシナリオテンプレートを、JP1/AJS2 - Scenario Operation View を使用して JP1/AJS - Manager に登録します。

サンプルシナリオテンプレートは、次の場所に格納されています。

/ScenarioLibrary/OpenTP1/TP1_ServerBase/OpenTP1_ScenarioScaleout

サンプルシナリオテンプレートを登録するには、入力シナリオ変数を設定する必要があります。サンプルシナリオテンプレートへの設定情報を、次に示します。

表 6-5 サンプルシナリオテンプレートへの設定情報

入力シナリオ変数	内容
DCDIR	OpenTP1 のホームフォルダ
DCCONFPATH	OpenTP1 定義ファイル格納フォルダ
HOST_NAME ※ 1	実行する OpenTP1 ノードのホスト名
USER_NAME ※ 1	OpenTP1 のユーザ名
BETRAN_FILE1 ※ 2	OpenTP1 ファイルシステムのパス
BETRAN_FILE2 ※ 2	OpenTP1 ファイルシステムのパス
USER_SERVER_NAME	ユーザサーバ名
NODE_ID	OpenTP1 のノード識別子（ノードごとに一意の 4 文字の文字列）
PORT_NUMBER	OpenTP1 のネームポート番号
MULTI_ID ※ 3	マルチ OpenTP1 の識別子（ノード内で一意の 4 文字以内の文字列）

注※ 1

設定を省略できます。

注※ 2

キャラクタ型スペシャルファイルを設定する場合は、シナリオを実行する前にパーティションを割り当ててください。

注※ 3

オリジナルの OpenTP1 の場合は設定を省略できます。

6.5.4 シナリオテンプレートの詳細

Windows 版 OpenTP1 では、次のシナリオテンプレートの仕様が UNIX 版 OpenTP1 と異なります。

● OpenTP1_Deploy ※

- OpenTP1_Undeploy ※
- OpenTP1_ScenarioScaleout

注※

オリジナルの OpenTP1 の場合は実行しません。

ここでは、これらのシナリオテンプレートの詳細について説明します。これ以外のシナリオテンプレートについては、マニュアル「OpenTP1 運用と操作」のシナリオテンプレートの詳細についての説明を参照してください。

(1) OpenTP1_Deploy

指定した OpenTP1 フォルダに、マルチ OpenTP1 をセットアップするテンプレートです。ただし、指定した OpenTP1 フォルダがオリジナルの OpenTP1 の場合は、実行しません。

入力シナリオ変数

- DCDIR ～ 〈1 ～ 50 文字の文字列〉
 セットアップする OpenTP1 フォルダ名を指定します。
 OpenTP1 システムで一意になるように指定してください。
- HOST_NAME
 シナリオジョブを実行する OpenTP1 ノードのホスト名を指定します。
 この入力シナリオ変数は省略できます。省略した場合、シナリオの実行先ホストでシナリオジョブを実行します。
- USER_NAME
 シナリオジョブを実行する OpenTP1 管理者のユーザ名を指定します。
 この入力シナリオ変数は省略できます。省略した場合、JP1/Base で定義したユーザマッピングのプライマリユーザがシナリオジョブを実行します。
- MULTI_ID ～ 〈1 ～ 4 文字の文字列〉
 マルチ OpenTP1 の識別子を指定します。
 入力シナリオ変数 DCDIR で指定した OpenTP1 フォルダがオリジナルの OpenTP1 の場合には、省略できます。

実行ユーザ

入力シナリオ変数 USER_NAME で指定した OpenTP1 管理者です。

(2) OpenTP1_Undeploy

指定した OpenTP1 フォルダがマルチ OpenTP1 の場合、マルチ OpenTP1 を削除します。指定した OpenTP1 フォルダがオリジナルの OpenTP1 の場合は、実行しません。

入力シナリオ変数

- DCDIR ～ 〈1 ～ 50 文字の文字列〉

削除する OpenTP1 フォルダ名を指定します。

● **HOST_NAME**

シナリオジョブを実行する OpenTP1 ノードのホスト名を指定します。

この入力シナリオ変数は省略できます。省略した場合、シナリオの実行先ホストでシナリオジョブを実行します。

● **USER_NAME**

シナリオジョブを実行する OpenTP1 管理者のユーザ名を指定します。

この入力シナリオ変数は省略できます。省略した場合、JP1/Base で定義したユーザマッピングのプライマリユーザがシナリオジョブを実行します。

実行ユーザ

入力シナリオ変数 USER_NAME で指定した OpenTP1 管理者です。

(3) OpenTP1_ScenarioScaleout

OpenTP1 の環境設定をして、OpenTP1 およびサンプル SPP を起動します。

入力シナリオ変数

● **DCDIR** ～ 〈1 ～ 50 文字の文字列〉

セットアップする OpenTP1 フォルダ名を指定します。

OpenTP1 システムで一意になるように指定してください。

● **DCCONFPATH**

OpenTP1 定義ファイル格納フォルダを指定します。

● **HOST_NAME**

シナリオジョブを実行する OpenTP1 ノードのホスト名を指定します。

この入力シナリオ変数は省略できます。省略した場合、シナリオの実行先ホストでシナリオジョブを実行します。

● **USER_NAME**

シナリオジョブを実行する OpenTP1 管理者のユーザ名を指定します。

この入力シナリオ変数は省略できます。省略した場合、JP1/Base で定義したユーザマッピングのプライマリユーザがシナリオジョブを実行します。

● **BETRAN_FILE1**

A 系の OpenTP1 ファイルシステムのパスを指定します。

キャラクタ型スペシャルファイルを指定する場合は、シナリオテンプレートを実行する前に、パーティションを割り当ててください。

● **BETRAN_FILE2**

B 系の OpenTP1 ファイルシステムのパスを指定します。

キャラクタ型スペシャルファイルを指定する場合は、シナリオテンプレートを実行する前に、パーティションを割り当ててください。

● **USER_SERVER_NAME** ～ 〈1 ～ 8 文字の文字列〉

6. 運用と操作

起動するユーザサーバ名を指定します。

サンプルシナリオテンプレートを使用する場合は、basespp を指定してください。

- **NODE_ID** ～ 〈4 文字の文字列〉

設定する OpenTP1 のノード ID を指定します。

OpenTP1 システムで一意になるように指定してください。

- **PORT_NUMBER** ～ ((5001 ～ 65535))

OpenTP1 システムに新しく追加する OpenTP1 ノードのネームサーバが使用するポート番号を指定します。

- **MULTI_ID** ～ 〈1 ～ 4 文字の文字列〉

マルチ OpenTP1 の識別子を指定します。

入力シナリオ変数 DCDIR で指定した OpenTP1 フォルダがオリジナルの OpenTP1 の場合には、省略できます。

実行ユーザ

入力シナリオ変数 USER_NAME で指定した OpenTP1 管理者です。

6.6 障害対策

ここでは、マニュアル「OpenTP1 運用と操作」に記載されている障害対策のうち、Windows 版 OpenTP1 固有の部分についてだけ説明します。

6.6.1 障害発生時の現象と原因

Windows 版 OpenTP1 固有の障害発生時の現象、推定できる原因、および対策を次に示します。この表以外の障害発生時の現象、推定できる原因、および対策については、マニュアル「OpenTP1 運用と操作」を参照してください。

表 6-6 障害発生時の現象、推定できる原因、および対策

現象	原因	対策
UAP を開始できない	プロセス固有メモリが不足しています。	不要なプロセスが存在している場合は、そのプロセスを停止させてください。不要なプロセスがない場合は、物理メモリを追加し、Windows の仮想メモリページファイルを拡張してください。
UAP を開始できない（ユーザサーバが終了状態 0x00000080 または 0xC0000135 でダウンする）	UAP の実行に必要なランタイムライブラリのパスが設定されていません。	UAP の実行に必要なランタイムライブラリのパスを設定してください。詳細については、「4.3.3 ランタイムライブラリのパスの設定」を参照してください。
	マニフェストファイルが存在しないか、またはマニフェストファイルに記載されているライブラリを参照できません。	マニフェストファイル、またはライブラリを確認してください。詳細については、「3.3(4) Visual Studio 使用時の注意事項」を参照してください。
	プロセスの起動とサーバマシンのログオフ処理が重なると、プロセスの起動に失敗することがあります。	プロセスが頻繁に生成される OpenTP1 起動時、および UAP プロセスの起動と停止が頻繁に発生するシステムのオンライン業務中は、当該サーバマシンでのログオフ操作を控えてください。
	デスクトップヒープの不足が原因で、子プロセスの生成に失敗することがあります。	不要なプロセスが存在している場合は、そのプロセスを停止させてください。不要なプロセスがない場合は、OS のデスクトップヒープ領域を拡張してください。
UAP を終了できない	アプリケーション例外が発生しています。	対策については、「6.6.2 アプリケーション例外発生時の対策」を参照してください。

6. 運用と操作

現象	原因	対策
OpenTP1 を開始できない	メモリ、またはディスク容量が不足しています。	OpenTP1 の定義内容を見直してください。見直しても開始できない場合は、次に示す要因ごとにそれぞれ見直してください。 • プロセス固有メモリが不足している場合 物理メモリのサイズ、および Windows の仮想メモリページファイルの容量を見直してください。 • 共用メモリが不足している場合 不要なファイルを削除してください。OpenTP1 をインストールしたディスクパーティションの残容量が、OpenTP1 が確保する共用メモリ容量より大きくなるようにします。
OpenTP1 を開始できない、または UAP を開始できない (OpenTP1 システムプロセスや UAP プロセスが、KFCA00100-E メッセージを出力し、ダウンする)	連続した空き仮想アドレス空間領域が不足しています。	共用メモリサイズが大き過ぎないか、OpenTP1 の定義内容を見直してください。見直しても開始できない場合は、次に示す要因ごとに対策してください。 • マッピングサービスプロセス (mapsmgrd.exe) が、メモリ不足でダウンする場合 マッピングサービスプロセスが使用する XMAP3 の DLL を入れ替える※ことで、仮想アドレス空間上の空き連続領域を増やせることがあります。XMAP3 の提供する DLL の入れ替え変更手順については、XMAP3 のリリースノートを参照してください。 • OpenTP1 システムプロセス (trnrmd.exe, trnrvd.exe, clttrnd.exe)、または UAP プロセスが、メモリ不足でダウンする場合 保守員へ連絡してください。

現象	原因	対策
通信性能やディスク I/O 性能が劣化する	OpenTP1 が動作するコンピュータで、実行プログラムをリアルタイムに検疫する機能を持つウィルス駆除プログラムが起動している可能性があります。 ウィルス駆除プログラムの動作によっては、OpenTP1 が動作に必要なファイルに一時的にアクセスできなくなり、OpenTP1 の動作に影響を及ぼすおそれがあります。	次のパスをウィルス駆除プログラムの検疫対象から除外することをお勧めします。 • %DCDIR% 下のフォルダのパス • 定義ファイルや TP1 ファイルシステムなどを %DCDIR% 下以外に格納している場合は、そのフォルダのパス
ポートが不足している	高トラフィック状態になっています。	対策については、「6.6.3 ポート数不足への対策（TCP/IP ポート数のチューニング）」を参照してください。

注※

XMAP3 のバージョンによっては、対応していない場合があります。

6.6.2 アプリケーション例外発生時の対策

OpenTP1 のオンライン中に OpenTP1 配下のプロセスからアプリケーション例外が発生した場合には、Windows のデバッグハンドラがコールされ、レジストリに設定されたデバッグプログラムが起動されます。起動されるデバッグプログラムが VC++ などの対話形式プログラムの場合、例外が発生したプログラムが終了されないため、OpenTP1 システムが再開始（リラン）できないおそれがあります。システムの本番環境など、システムの回復処理が必須である環境では、Windows エラー報告（WER）を使用して、クラッシュダンプを取得する設定をしてください。

6.6.3 ポート数不足への対策（TCP/IP ポート数のチューニング）

OpenTP1 を使用して大規模なシステムを構築する場合、TCP/IP が管理するポートが不足するおそれがあります。このような場合は、一時クローズ処理を実施してポート不足を解消します。

しかし、高トラフィックな状態では、一時クローズ処理を実施してもポート不足になるおそれがあります。このような場合は、Windows の設定で自動割り当てポートを拡張すれば、該当するプラットフォーム上で使用ポートに関するチューニングができます。

Windows のレジストリ・コマンド操作で、次のパラメータを設定してください。

- MaxUserPort
- TcpTimedWaitDelay

6. 運用と操作

- ReservedPorts

設定方法については、Windows のマニュアルを参照してください。

また、一時クローズ処理については、マニュアル「OpenTP1 解説」のポート数の制限方法についての説明を参照してください。

6.6.4 障害発生時の取得情報の参照

障害要因を調査するための、取得情報の参照方法について説明します。

障害時に取得する情報については、マニュアル「OpenTP1 運用と操作」の OpenTP1 が出力するファイル一覧についての説明を参照してください。

(1) 標準出力・標準エラー出力の参照

Windows 版 OpenTP1 では、システムサーバおよびユーザサーバが標準出力・標準エラー出力に出力した内容（KFCA-xxxxx メッセージや、ユーザが標準出力に指定した内容すべて）を、ファイルやコンソールに出力して確認できます。

標準出力・標準エラー出力を確認するための機能については、「2.2 標準出力・標準エラー出力を確認するための機能」を参照してください。

(2) Windows の [イベントビューア] でのイベントログの参照

次の情報が、Windows の [イベントビューア] に出力されます。

- インストール・セットアッププログラム、運用コマンドの出力メッセージを除く OpenTP1 のメッセージ
- マニュアル「OpenTP1 メッセージ」で、出力先が標準出力、または標準エラー出力と記載されているメッセージ

OpenTP1 のイベントログを参照する手順を次に示します。

1. Windows の [イベントビューア] を表示します。

2. 「アプリケーション」を選択します。

アプリケーションログが表示されます。「ソース」に「OpenTP1」と表示されているイベントが、OpenTP1 が出力したメッセージです。

また、dc_logprint 関数または CBLDCLOG('PRINT')で指定したメッセージも Windows の [イベントビューア] に出力されます。dc_logprint 関数または CBLDCLOG('PRINT')を使用する場合は、次の点に注意してください。

- CBLDCLOG('PRINT')の場合、メッセージ表示色の指定は無視されます。
- 引数に不正な値が設定されたときは、イベントログに次の形式でエラーメッセージが出力されます。

メッセージ不正のため、(xxx...xx) は出力できません

(凡例)

xxx...xx: メッセージ ID

- 「KFCAn1n2n3n4n5-x」で表されるメッセージ形式の「-x」には、-I, -R, -W, -E 以外は指定できません。これ以外の値を指定した場合は、引数の不正として、前述のエラーメッセージが出力されます。また、「-x」は、Windows の「イベントビューア」の「種類」に表示される内容と対応しています。「-x」と表示内容の対応を次に示します。

- -I, または -R: 情報
- -W: 警告
- -E: エラー

6.7 Windows 版 OpenTP1 の GUI

Windows 版 OpenTP1 では、GUI を使用して OpenTP1 の環境設定や、OpenTP1 の起動と停止ができます。これらの操作はそれぞれ独立しているため、例えば、GUI で環境設定をして、コマンドプロンプトから OpenTP1 を起動するという操作ができます。

なお、Windows 版 OpenTP1 の GUI を使用する前に、次のことを確認してください。

- .NET Framework のバージョン

GUI を使用するためには、.NET Framework 4 以降が必要です。.NET Framework 4 以降がインストールされていない場合は、インストールしてください。

- ユーザーアカウント制御の設定

Windows 7 以降、および Windows Server 2008 R2 以降で GUI を使用する場合、GUI の起動前に、[ユーザーアカウント制御] ダイアログボックスが表示されます。[許可 (A)] ボタンをクリックしてください。

- セキュリティポリシーの設定

次に示すファイルに完全信頼を指定してください。完全信頼を指定しないと、GUI を使用できません。セキュリティポリシーの設定方法については、.NET Framework のドキュメントを参照してください。

- %DCDIR%\¥tp1_tools¥Control.exe
- %DCDIR%\¥tp1_tools¥Environment.exe

- コマンドログ、および監査ログの出力先

GUI を使用すると、コマンドログ、および監査ログが出力されることがあります。ただし、GUI の操作対象に関係なく、各ログは、マルチ OpenTP1 には出力されません。

6.7.1 GUI による環境設定

ここでは、GUI による OpenTP1 の環境設定について説明します。GUI では、OpenTP1 を使用するために必要な最低限のシステム定義をテンプレートとして提供しています。テンプレートには、システム構成に応じた値（パラメータ）が設定された複数のモデルが用意されています。各モデルに設定されているパラメータの値は、必要に応じて、GUI で変更できます。なお、GUI で設定したパラメータの値は、対応するシステム定義のオペランドの指定値として扱われます。

注意事項

テンプレートに設定されていない OpenTP1 のシステム定義を追加するなど、テンプレートの内容そのものは変更できません。

GUI による OpenTP1 の環境設定の手順を次に示します。

1. **【スタート】－【プログラム】－【OpenTP1】－【OpenTP1 環境設定】メニューを選択します。**
 【OpenTP1 環境設定】ウィンドウが表示されます。
2. **【OpenTP1 環境設定】ウィンドウで定義種別およびモデルを選択し、【次へ (N)】ボタンをクリックします。**
 【OpenTP1 環境設定】ウィンドウ（パラメータ情報）が表示されます。
3. **【OpenTP1 環境設定】ウィンドウ（パラメータ情報）でパラメータの値を入力します。**
4. **設定した内容を確認する場合は、【OpenTP1 環境設定】ウィンドウ（パラメータ情報）の【詳細 (D)】ボタンをクリックします。**
 【詳細情報】ダイアログボックスが表示されます。
5. **【OpenTP1 環境設定】ウィンドウ（パラメータ情報）で【次へ (N)】ボタンをクリックします。**
 【OpenTP1 環境設定】ウィンドウ（格納先情報）が表示されます。
6. **【OpenTP1 環境設定】ウィンドウ（格納先情報）で環境設定の対象となる OpenTP1 のサービス名を選択し、「バッチファイルを実行する」チェックボックスを ON にします。**
 「バッチファイルを実行する」チェックボックスを ON にすると、【実行 (E)】ボタンをクリックしたときに、定義ファイル、および OpenTP1 ファイルシステムが作成されます。
7. **【OpenTP1 環境設定】ウィンドウ（格納先情報）の【実行 (E)】ボタンをクリックします。**
 定義種別、モデル、およびパラメータの値が反映されたテンプレートが格納されます。また、バッチファイルが実行され、定義ファイル、および OpenTP1 ファイルシステムが作成されます。
8. **バッチファイルの実行結果を確認する場合は、【OpenTP1 環境設定】ウィンドウ（格納先情報）の【ログ確認 (V)】ボタンをクリックします。**
 【ログ確認】ダイアログボックスが表示されます。
9. **【OpenTP1 環境設定】ウィンドウ（格納先情報）の【閉じる (C)】ボタンをクリックします。**
 【OpenTP1 環境設定】ウィンドウ（格納先情報）が閉じ、OpenTP1 の環境設定を終了します。

環境設定で使用するウィンドウおよびダイアログボックスの詳細を説明します。

(1) 【OpenTP1 環境設定】ウィンドウ

【OpenTP1 環境設定】ウィンドウでは、定義種別、およびモデルを設定します。

【OpenTP1 環境設定】ウィンドウは、【スタート】－【プログラム】－【OpenTP1】－

6. 運用と操作

「OpenTP1 環境設定」メニューを選択すると表示されます。

「OpenTP1 環境設定」ウィンドウを次に示します。

図 6-2 「OpenTP1 環境設定」ウィンドウ



メニューバー

「ファイル(F)」メニュー

「ファイル(F)」メニューで「閉じる(C)」を選択すると、「OpenTP1 環境設定」ウィンドウを閉じます。

「ヘルプ(H)」メニュー

「ヘルプ(H)」メニューで「バージョン情報(A)」を選択すると、「バージョン情報(A)」ダイアログボックスが表示されます。このダイアログボックスには、OpenTP1 のインストール時に入力した OpenTP1 管理者の情報が表示されます。

「プラットフォーム」ドロップダウンリスト

「Windows」が表示されます。この項目は、設定する必要はありません。

「定義種別」ドロップダウンリスト

次のどちらかの定義を選択できます。

- TP1_Server_base System Definition

OpenTP1 を開始させる場合に選択します。この定義を選択すると、OpenTP1 の開始に必要なパラメータ（システム共通定義や rap リスナーサービス定義など一部の定義の特定のオペランド）を指定できます。

ユーザサービス定義を指定する場合は、「User Service Definition」を選択します。

- User Service Definition

ユーザサーバを起動させる場合に選択します。この定義を選択すると、ユーザサーバの起動に必要なパラメータ（ユーザサービス定義の一部のオペランド）を指定できます。ただし、「User Service Definition」を選択する前に、

「TP1_Server_base System Definition」を選択して、定義ファイルを作成しておく必要があります。

「モデル」ドロップダウンリスト

テンプレートの定義のモデルを選択できます。「定義種別」ドロップダウンリストで選択する定義によって、選択できるモデルが異なります。選択できるモデルの一覧を次に示します。

表 6-7 選択できるモデルの一覧

選択する定義	選択できるモデル	説明
TP1_Server_base System Definition	Sample Model	OpenTP1 サービスを起動させて、サンプル UAP を使用できるモデルです。
	Sample Model(RAP)	OpenTP1 サービスを起動させて、RAP サーバを使用できるモデルです。
	Large Model	大規模なシステム環境向きのモデルです。
User Service Definition	SPP Sample Model	サンプル SPP 用のモデルです。
	SPP.NET Sample Model	サンプル SPP.NET 用のモデルです。
	SPP.NET Sample Model(.NET Interface)	.NET インタフェースを使用したサンプル SPP.NET 用のモデルです。
	SUP Sample Model	サンプル SUP 用のモデルです。
	SUP.NET Sample Model	サンプル SUP.NET 用のモデルです。
	SUP.NET Sample Model(.NET Interface)	.NET インタフェースを使用したサンプル SUP.NET 用のモデルです。

「定義情報」

「モデル」ドロップダウンリストで選択したモデルの説明が表示されます。

【次へ(N)】ボタン

[OpenTP1 環境設定] ウィンドウ (パラメータ情報) が表示されます。[OpenTP1 環境設定] ウィンドウ (パラメータ情報) については、「(2) [OpenTP1 環境設定] ウィンドウ (パラメータ情報)」を参照してください。

【閉じる(C)】ボタン

設定した内容を保存しないで、GUI による環境設定を終了します。

(2) [OpenTP1 環境設定] ウィンドウ (パラメータ情報)

[OpenTP1 環境設定] ウィンドウ (パラメータ情報) では、パラメータの値を入力します。

[OpenTP1 環境設定] ウィンドウ (パラメータ情報) は、[OpenTP1 環境設定] ウィンドウで [次へ(N)] ボタンをクリックすると表示されます。

[OpenTP1 環境設定] ウィンドウ (パラメータ情報) を次に示します。

図 6-3 [OpenTP1 環境設定] ウィンドウ (パラメータ情報)



「パラメーター一覧」

[OpenTP1 環境設定] ウィンドウで選択した定義種別とモデルに対応したパラメータ名と値が表示されます。OpenTP1 の環境に合わせて、パラメータの値を変更できます。なお、「パラメーター一覧」の「値」以外の項目は、変更できません。パラメータについては、「5. システム定義」、およびマニュアル「OpenTP1 システム定義」の同名のオペランドについての説明を参照してください。

「パラメータ情報」

パラメータの「値」のセルを選択すると、そのパラメータの説明が表示されます。

「詳細 (D)」 ボタン

[詳細情報] ダイアログボックスが表示されます。[詳細情報] ダイアログボックスについては、「(3) [詳細情報] ダイアログボックス」を参照してください。

「戻る (B)」 ボタン

[OpenTP1 環境設定] ウィンドウに戻ります。

「次へ (N)」 ボタン

[OpenTP1 環境設定] ウィンドウ（格納先情報）が表示されます。[OpenTP1 環境設定] ウィンドウ（格納先情報）については、「(4) [OpenTP1 環境設定] ウィンドウ（格納先情報）」を参照してください。

「閉じる (C)」 ボタン

設定した内容を保存しないで、GUI による環境設定を終了します。

(3) [詳細情報] ダイアログボックス

[詳細情報] ダイアログボックスでは、[OpenTP1 環境設定] ウィンドウ、および [OpenTP1 環境設定] ウィンドウ（パラメータ情報）で指定した内容を確認できます。

[詳細情報] ダイアログボックスは、[OpenTP1 環境設定] ウィンドウ（パラメータ情報）で [詳細 (D)] ボタンをクリックすると表示されます。

[詳細情報] ダイアログボックスを次に示します。

図 6-4 「詳細情報」 ダイアログボックス



表示内容

「OpenTP1 環境設定」ウィンドウ, および 「OpenTP1 環境設定」ウィンドウ (パラメータ情報) で設定した内容が表示されます。

「OK(Q)」ボタン

「詳細情報」ダイアログボックスを閉じます。

(4) 「OpenTP1 環境設定」ウィンドウ (格納先情報)

「OpenTP1 環境設定」ウィンドウ (格納先情報) では, 環境設定の対象となる OpenTP1 のサービスを選択し, テンプレートの格納先フォルダを指定します。

「OpenTP1 環境設定」ウィンドウ (格納先情報) は, 「OpenTP1 環境設定」ウィンドウ (パラメータ情報) で 「次へ (N)」 ボタンをクリックすると表示されます。

「OpenTP1 環境設定」ウィンドウ (格納先情報) を次に示します。

図 6-5 「OpenTP1 環境設定」ウィンドウ（格納先情報）



「サービス名」ドロップダウンリスト

テンプレートを格納する OpenTP1 のサービスを選択できます。ドロップダウンリストには、Windows にセットアップされている次の OpenTP1 のサービスが表示されます。

- OpenTP1
OpenTP1 をインストールしたあとに、自動的に作成されるサービスです。
「OpenTP1」を選択すると、テンプレートは %DCDIR%\conf に格納されます。
- OpenTP1_XXXX
マルチ OpenTP1 のサービスです。XXXX は、マルチ OpenTP1 の識別子を示します。「OpenTP1_XXXX」を選択すると、テンプレートはマルチ OpenTP1 のフォルダに格納されます。
- その他
ユーザが、任意にテンプレートの格納先を選べます。

「バッチファイルを実行する」チェックボックス

チェックボックスを ON にすると、テンプレートを格納するときに自動的にバッチファイルを実行して、定義ファイルや OpenTP1 ファイルシステムを作成できます。

「格納先ディレクトリ」

6. 運用と操作

「サービス名」ドロップダウンリストで選択した内容によって、格納先フォルダが異なります。

- 「OpenTP1」または「OpenTP1_XXXX」を選択した場合
選択した OpenTP1 の %DCDIR%\%conf が表示されます。
- 「その他」を選択した場合
テンプレートを格納するフォルダパスを入力します。

格納先フォルダを指定する場合の注意事項

OpenTP1 が提供しているテンプレートには、パラメーター一覧に表示されるパラメータのほかに、環境変数 DCDIR の値も定義されます。環境変数 DCDIR の値は、「サービス名」ドロップダウンリストで「その他」を選択した場合、任意の格納先を入力できるため、一時的に @DCDCDIR@ が設定されます。しかし、@DCDCDIR@ を設定したままでは、環境設定を実行できません。必ず使用する環境の %DCDIR% の値に置き換えてください。

【ログ確認 (V)] ボタン

【ログ確認】ダイアログボックスが表示されます。【ログ確認】ダイアログボックスについては、「(5) 【ログ確認】ダイアログボックス」を参照してください。

【戻る (B)] ボタン

【OpenTP1 環境設定】ウィンドウ（パラメータ情報）に戻ります。

【実行 (E)] ボタン

【格納先ディレクトリ】に表示されているフォルダパスにテンプレートが格納されます。また、「バッチファイルを実行する」チェックボックスを ON にしている場合には、バッチファイルが実行されます。

なお、バッチファイルを実行した場合、処理が完了すると、ダイアログボックスが表示されます。処理に失敗した場合は、ダイアログボックスにエラーメッセージが表示されます。エラーメッセージについては、マニュアル「OpenTP1 メッセージ」を参照してください。

【閉じる (C)] ボタン

GUI による環境設定を終了します。

(5) 【ログ確認】ダイアログボックス

【ログ確認】ダイアログボックスでは、バッチファイルの実行結果を確認できます。

【ログ確認】ダイアログボックスは、【OpenTP1 環境設定】ウィンドウ（格納先情報）で【ログ確認 (V)] ボタンをクリックすると表示されます。

【ログ確認】ダイアログボックスを次に示します。

図 6-6 「ログ確認」 ダイアログボックス

**「ファイル名」**

バッチファイルの実行結果を格納しているログファイル名が表示されます。

「更新日時」

バッチファイルを実行した日時が表示されます。

表示内容

バッチファイルの実行結果が表示されます。

[OK(Q)] ボタン

[ログ確認] ダイアログボックスを閉じます。

6.7.2 GUI による OpenTP1 の起動と停止

GUI を操作して、OpenTP1 の起動と停止ができます。また、起動オプション（開始モード）、および停止オプション（停止モード）を設定してから、OpenTP1 を起動または停止できます。

GUI で環境設定をしなくても、定義ファイルや OpenTP1 ファイルシステムが作成されていれば、GUI で OpenTP1 の起動と停止ができます。

6. 運用と操作

GUI による OpenTP1 の起動の手順を次に示します。

1. **【スタート】－【プログラム】－【OpenTP1】－【OpenTP1 起動 / 停止】メニューを選択します。**
【OpenTP1 起動 / 停止】ウィンドウが表示されます。
2. **【OpenTP1 起動 / 停止】ウィンドウで起動させたい OpenTP1 のサービスを選択します。**
3. **OpenTP1 の起動オプション（開始モード）を指定する場合は、【OpenTP1 起動 / 停止】ウィンドウで【オプション (O)...】ボタンをクリックします。**
【起動 / 停止オプション】ダイアログボックスが表示されます。
4. **【OpenTP1 起動 / 停止】ウィンドウで【起動 (S)】ボタンをクリックします。**
選択した OpenTP1 のサービスが起動されます。
5. **【OpenTP1 起動 / 停止】ウィンドウで【更新 (R)】ボタンをクリックします。**
選択した OpenTP1 のサービスが最新の状態に更新されます。
6. **【OpenTP1 起動 / 停止】ウィンドウで【閉じる (C)】ボタンをクリックします。**
「OpenTP1 起動 / 停止」ウィンドウが閉じ、OpenTP1 の起動が完了します。

GUI による OpenTP1 の停止の手順を次に示します。

1. **【スタート】－【プログラム】－【OpenTP1】－【OpenTP1 起動 / 停止】メニューを選択します。**
【OpenTP1 起動 / 停止】ウィンドウが表示されます。
2. **【OpenTP1 起動 / 停止】ウィンドウで停止させたい OpenTP1 のサービスを選択します。**
3. **OpenTP1 の停止オプション（終了モード）を指定する場合は、【OpenTP1 起動 / 停止】ウィンドウで【オプション (O)...】ボタンをクリックします。**
【起動 / 停止オプション】ダイアログボックスが表示されます。
4. **【OpenTP1 起動 / 停止】ウィンドウで【停止 (I)】ボタンをクリックします。**
選択した OpenTP1 のサービスが停止されます。
5. **【OpenTP1 起動 / 停止】ウィンドウで【更新 (R)】ボタンをクリックします。**
選択した OpenTP1 のサービスが最新の状態に更新されます。
6. **【OpenTP1 起動 / 停止】ウィンドウで【閉じる (C)】ボタンをクリックします。**
「OpenTP1 起動 / 停止」ウィンドウが閉じ、OpenTP1 の停止が完了します。

！ 注意事項

OpenTP1 が KFCA00715-E メッセージを出力した場合、[停止 (T)] ボタンでは停止できないことがあります。その場合は、Windows の [サービス] ダイアログ、または `net stop` コマンドで停止してください。停止方法の詳細については、「6.3 OpenTP1 の停止」を参照してください。

OpenTP1 の起動と停止で使用するウィンドウとダイアログボックスの詳細を説明します。

(1) [OpenTP1 起動 / 停止] ウィンドウ

[OpenTP1 起動 / 停止] ウィンドウでは、選択したサービスを起動または停止できます。

[OpenTP1 起動 / 停止] ウィンドウは、[スタート] — [プログラム] — [OpenTP1] — [OpenTP1 起動 / 停止] メニューを選択すると表示されます。

[OpenTP1 起動 / 停止] ウィンドウを次に示します。

図 6-7 [OpenTP1 起動 / 停止] ウィンドウ



メニューバーについては、「6.7.1(1) [OpenTP1 環境設定] ウィンドウ」を参照してください。

「サービス名」ドロップダウンリスト

OpenTP1 のサービスが表示されます。表示される内容は、[OpenTP1 環境設定] ウィンドウ (格納先情報) の「サービス名」ドロップダウンリストと同じです。

[OpenTP1 環境設定] ウィンドウ（格納先情報）の「サービス名」ドロップダウンリストについては、「6.7.1(4) [OpenTP1 環境設定] ウィンドウ（格納先情報）」を参照してください。

「OpenTP1 の状態」

「サービス名」ドロップダウンリストで選択した OpenTP1 のサービスの状態が表示されます。

OpenTP1 の開始または停止状態の詳細は、メッセージログファイルやイベントビューアで確認できます。

「起動」

[起動 (S)] ボタンをクリックすると、「サービス名」ドロップダウンリストで選択した OpenTP1 のサービスが起動します。「オプション」には、OpenTP1 の起動オプション（開始モード）が表示されます。OpenTP1 の起動オプション（開始モード）の詳細については、「(2) [起動 / 停止オプション] ダイアログボックス」を参照してください。

「停止」

[停止 (T)] ボタンをクリックすると、「サービス名」ドロップダウンリストで選択した OpenTP1 のサービスが停止します。「オプション」には、OpenTP1 の停止オプション（終了モード）が表示されます。OpenTP1 の停止オプション（終了モード）の詳細については、「(2) [起動 / 停止オプション] ダイアログボックス」を参照してください。

「更新 (R)」 ボタン

「サービス名」ドロップダウンリストに表示された OpenTP1 のサービスを最新の状態に更新します。

「オプション (O)...」 ボタン

[起動 / 停止オプション] ダイアログボックスが表示されます。[起動 / 停止オプション] ダイアログボックスについては、「(2) [起動 / 停止オプション] ダイアログボックス」を参照してください。

「閉じる (C)」 ボタン

GUI による OpenTP1 の起動、または停止の設定を終了します。

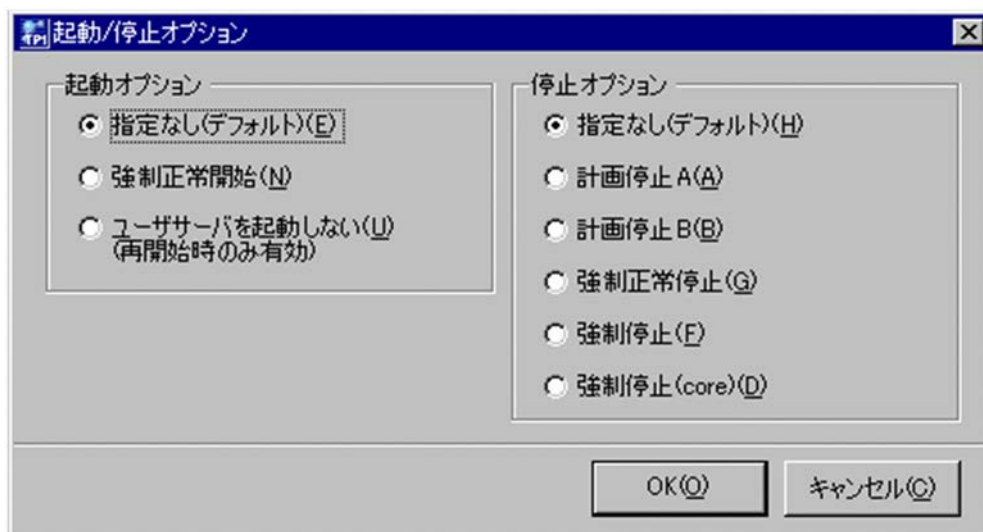
(2) [起動 / 停止オプション] ダイアログボックス

[起動 / 停止オプション] ダイアログボックスでは、OpenTP1 の起動オプション（開始モード）および停止オプション（終了モード）を設定できます。それぞれのモー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。

[起動 / 停止オプション] ダイアログボックスは、[OpenTP1 起動 / 停止] ウィンドウで [オプション (O)...] ボタンをクリックすると表示されます。

「起動 / 停止オプション」ダイアログボックスを次に示します。

図 6-8 「起動 / 停止オプション」ダイアログボックス



「起動オプション」

OpenTP1 の起動オプション（開始モード）を、次のラジオボタンから選択します。

- 「指定なし (デフォルト) (E)」

前回のオンラインが正常に終了して引き継ぐ情報がないとき、または新たに OpenTP1 を開始するときに正常開始します。正常開始ができない場合は、前回の処理内容を引き継ぐために再開します。

- 「強制正常開始 (N)」

前回の停止オプション（終了モード）に関係なく、OpenTP1 を強制的に正常開始します。

- 「ユーザーサーバを起動しない (U)」

再起動するときに、ユーザーサーバを起動しません。

「停止オプション」

OpenTP1 の停止オプション（終了モード）を、次のラジオボタンから選択します。

- 「指定なし (デフォルト) (H)」

OpenTP1 を正常終了します。

- 「計画停止 A (A)」

新しいサービス要求の受け付けを禁止し、スケジュールキュー上のすべてのサービス要求の処理を完了したあと、OpenTP1 を終了します。

- 「計画停止 B (B)」

現在実行しているサービスの完了を待って、OpenTP1 を終了します。

- 「強制正常停止 (G)」

運用中に異常終了したサーバがあっても、正常終了と同じように、新しいサービ

6. 運用と操作

ス要求の受け付けを禁止し、スケジュールキュー上のすべてのサービス要求の処理を完了したあとに、OpenTP1 を終了します。

また、ユーザサーバの状態に関係なく、強制的に正常終了します。

- **「強制停止 (E)」**

現在実行しているサービスの完了を待たないで、すぐに OpenTP1 を強制停止します。

- **「強制停止 (core)(D)」**

稼働するすべてのプロセスのコアファイル、および共用メモリ情報を取得して、OpenTP1 を強制停止します。

[OK(Q)] ボタン

選択したオプションを反映してダイアログボックスを閉じます。

[キャンセル (C)] ボタン

選択したオプションを反映しないでダイアログボックスを閉じます。

7

運用コマンド

Windows 版 OpenTP1 は，UNIX 版 OpenTP1 と比べて，使用できる運用コマンドが異なります。この章では，運用コマンドの一覧で，Windows 版 OpenTP1 と UNIX 版 OpenTP1 の運用コマンドの差異について説明します。また，一部の運用コマンドは，詳細についても説明します。

UNIX 版 OpenTP1 と共通の内容については，OpenTP1 シリーズのマニュアルを参照してください。

運用コマンドの一覧

コマンドの説明形式

UNIX 版 OpenTP1 と共通のコマンドの詳細

dcplist (製品情報の表示)

dcrasget (保守資料の取得)

fills (OpenTP1 ファイルシステムの内容表示)

jnlcolc (ファイル回復用ジャーナルの集積)

jnlcopy (アンロードジャーナルファイルの複写)

jnledit (アンロードジャーナルファイル，またはグローバルアーカイブアンロードジャーナルファイルの編集出力)

jnlmcst (MCF 稼働統計情報の出力)

jnlrput (アンロードジャーナルファイル，またはグローバルアーカイブアンロードジャーナルファイルのレコード出力)

jnlsort (アンロードジャーナルファイル，およびグローバルアーカイブアンロードジャーナルファイルの時系列ソート，およびマージ)

jnlstts (稼働統計情報の出力)

7. 運用コマンド

jnlunlfg (ジャーナル関係のファイルのアンロード)

rpcdump (RPC トレースの出力)

rpcmrg (RPC トレースのマージ)

trnlnkrm (リソースマネージャの登録)

trnmkobj (トランザクション制御用オブジェクトファイルの作成)

Windows 版 OpenTP1 固有のコマンドの詳細

dcdls (マルチ OpenTP1 のセットアップ状態の確認)

dcmakecon (マルチ OpenTP1 操作コンソールの作成)

dcsetupml (マルチ OpenTP1 のセットアップとアンセットアップ)

njsmkdll (トランザクション制御用 DLL の作成)

ntbstart (OpenTP1 の起動)

ntbtail (指定ファイル内容のコンソール出力)

tp1console (標準出力リダイレクトファイルのコンソール出力)

運用コマンドの一覧

Windows 版 OpenTP1 は、UNIX 版 OpenTP1 と比べて、使用できるコマンドが異なります。また、一部のコマンドでは、オプションや設定値をサポートしていないなどの制限事項があります。Windows 版 OpenTP1 を初めて使用する場合は、ここで示す一覧表で UNIX 版 OpenTP1 との差異があるかどうかを確認してください。

この一覧表の内容を確認した上で、UNIX 版 OpenTP1 と共通の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。UNIX 版 OpenTP1 と差異のあるコマンドについては、必要に応じて、この章の「UNIX 版 OpenTP1 と共通のコマンドの詳細」の該当箇所を参照してください。Windows 版 OpenTP1 固有のコマンドについては、この章の「Windows 版 OpenTP1 固有のコマンドの詳細」の該当箇所を参照してください。

！ 注意事項

この一覧表では、次のコマンドについて説明していません。

- **メッセージキューイング機能（TP1/Message Queue）のコマンド**

コマンドの詳細については、マニュアル「TP1/Message Queue 使用の手引」を参照してください。

- **TP1/Extension for .NET Framework のコマンド**

コマンドの詳細については、マニュアル「TP1/Extension for .NET Framework 使用の手引」を参照してください。

一覧表で使用している記号の凡例を次に示します。

（凡例）

◎：UNIX 版 OpenTP1 と同じ機能をサポートしています。

○：Windows 版 OpenTP1 固有のコマンドです。

△：UNIX 版 OpenTP1 との機能差異や Windows 版 OpenTP1 での制限事項があります。

×：サポートしていません。

－：該当しません。

Windows 版 OpenTP1 と UNIX 版 OpenTP1 の運用コマンドの差異を、次に示します。

7. 運用コマンド

運用コマンドの一覧

表 7-1 Windows 版 OpenTP1 と UNIX 版 OpenTP1 の運用コマンドの差異

機能	運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
システム管理	OpenTP1 の OS への登録と削除	×	サポートしていません。 Windows 版 OpenTP1 は、インストーラによって OS に登録されます。そのため、インストール後にこのコマンドを実行すると、OpenTP1 環境が破壊されるおそれがあります。このコマンドを誤って実行した場合には、OpenTP1 を再インストールしてください。
	プロセスサービスの再起動および定義の反映	×	サポートしていません。 システム定義の変更については、「4.3.2 定義の変更」を参照してください。
	OpenTP1 の内部制御用資源の確保と解放	×	サポートしていません。
	OpenTP1 の開始	×	サポートしていません。 Windows 版 OpenTP1 の開始方法については、「6.2 OpenTP1 の開始」を参照してください。
	OpenTP1 の終了	◎	Windows 版 OpenTP1 の停止方法については、「6.3 OpenTP1 の停止」を参照してください。
	システム統計情報の取得開始、終了	△	-k オプションの指定値のうち、ist, xat はサポートしていません。
	マルチノードエリア、サブエリアの開始	×	サポートしていません。
	マルチノードエリア、サブエリアの終了	×	
	シナリオテンプレートからの OpenTP1 コマンドの実行	◎	—
	システム定義のオペランドの指定	◎	—
	ドメイン定義ファイルの更新	◎	—
	OpenTP1 ノードの状態表示	×	サポートしていません。
	共用メモリの状態表示	◎	—

機能	運用コマンド	Windows 版 OpenTP1 で のサポート の有無	UNIX 版 OpenTP1 との差異
一時クローズ処理 の実行状態の表示	rpcstat	◎	—
標準出力、標準エ ラー出力のリダイ レクト	prctee	×	サポートしていません。 システム環境定義の <code>redirect_file</code> オペ ランドで出力形式を指定してください。 <code>redirect_file</code> オペランドの詳細について は、「5.3 システム定義の詳細」の 「システム環境定義」を参照してくださ い。
prctee プロセス の停止と再開始	prctctrl	×	サポートしていません。
保守資料の取得	dcrasget	△	-c オプションは、サポートしていま せん。また、保守資料の格納先について 差異があります。 詳細については、「dcrasget（保守資料 の取得）」を参照してください。
システム統計情報 の標準出力へのリ アルタイム編集出 力	dcreport	◎	—
トラブルシュート 情報の削除	dccspool	◎	—
システム定義の チェック	dcddefchk	△	このコマンドによる論理チェックでは、 NTFS レベルでのアクセス権はチェッ クしていません。そのため、アクセス 権限のないフォルダをシステム定義に 指定している場合、ファイルの作成な どに失敗します。
製品情報の表示	dcplist	△	OpenTP1 ディレクトリに構築した環境 で動作している製品名の出力形式につ いて差異があります。 詳細については、「dcplist（製品情報 の表示）」を参照してください。

7. 運用コマンド

運用コマンドの一覧

機能	運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
サーバ管理	サーバの開始	dcsvstart	△
			次に示す注意事項があります。 dcsvstart コマンドに指定するユーザサーバ名は、大文字、小文字を正確に指定してください（ユーザサービス定義のファイル名と一致させてください）。 Windows 版では、大文字、小文字の違いがあっても同一の定義ファイルと認識します。このため、dcsvstart コマンドに指定したユーザサーバ名と、ユーザサービス定義のファイル名との間で大文字、小文字の違いがあっても、ユーザサーバを起動します。ただし、OpenTP1 内部で管理するユーザサーバ名は大文字、小文字を区別していますので、ユーザサーバ名を引数に指定するコマンド (dcsvstop など) では、dcsvstart コマンドに指定したユーザサーバ名と完全に一致する名称を指定するように注意してください。なお、起動済みのユーザサーバ名を確認したい場合は、prcls コマンドまたは scdls コマンドを使用してください。
	サーバの終了	dcsvstop	◎
	サーバの状態表示	prcls	△
			-l オプションで指定するモジュール名には、ユーザサービス定義の module オペランドで指定した値を指定してください。
	ユーザサーバ、およびユーザサーバから起動されるコマンドのサーチパス名の表示	prepathls	◎
	ユーザサーバ、およびユーザサーバから起動されるコマンドのサーチパスの変更	prepath	△
			ユーザサーバのパスを変更した場合でも、必ず先頭に %DCDIR%\bin が付与されます。
	UAP 共用ライブラリのサーチパス名の表示	predlpathls	×
			サポートしていません。
	UAP 共用ライブラリのサーチパスの変更	predlpath	×
			サポートしていません。

機能		運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	OpenTP1 のプロセスの強制停止	prckill	◎	このコマンドに UNIX 版 OpenTP1 との差異はありませんが、強制停止時のシグナル番号としてユーザサービス定義の <code>prc_abort_signal</code> オペランドに指定できる値は、3, 6, 15 のどれかです。 <code>prc_abort_signal</code> オペランドの詳細については、「5.3 システム定義の詳細」の「ユーザサービス定義」を参照してください。
スケジュー ル管理	スケジュールの状態表示	scdls	△	-b オプションは、サポートしていません。
	スケジュールの閉塞	scdhold	◎	—
	スケジュールの再開始	scdrls	◎	—
	プロセス数の変更	scdchprc	◎	—
	プロセスの停止および再起動	scdrsprc	◎	—
トランザク ション 管理	トランザクションの状態表示	trnls	△	-R オプションに指定するリソースマネージャとして、OpenTP1_ISM は指定できません。
	トランザクションのコミット	trncmt	◎	—
	トランザクションのロールバック	trnrbk	◎	—
	トランザクションの強制終了	trnfgt	◎	—
	トランザクション統計情報の取得開始、終了	trnstics	◎	—
	未決着トランザクション情報ファイルの削除	trndlinf	◎	—
	OSI TP 通信の未決着トランザクション情報の表示	tptrnls	×	サポートしていません。
XA リ ソース 管理	XAR イベントトレース情報の表示	xarevtr	◎	—

7. 運用コマンド

運用コマンドの一覧

機能		運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	XAR ファイルの状態表示	xarfills	◎	—
	XAR トランザクション状態の変更	xarforce	◎	—
	XA リソースサービスの閉塞	xarhold	◎	—
	XAR ファイルの作成	xarinit	◎	—
	XAR トランザクション情報の表示	xarls	◎	—
	XA リソースサービスの閉塞解除	xarrles	◎	—
	XAR ファイルの削除	xarmrm	◎	—
排他管理	排他情報の表示	lckls	◎	—
	排他制御用テーブルのプールの情報の表示	lckpool	◎	—
	デッドロック情報ファイルとタイムアウト情報ファイルの削除	lckrminf	◎	—
ネーム管理	OpenTP1 起動確認, キャッシュ削除	namalivechk	◎	—
	ドメイン代表スケジューラサービスの登録, 削除	namdomainsetup	×	サポートしていません。
	ドメイン構成の変更 (システム共通定義使用)	namndchg	◎	—
	ドメイン構成の変更 (ドメイン定義ファイル使用)	namchgfl	◎	—
	起動通知情報の強制的無効化	namunavl	◎	—
	OpenTP1 のサーバ情報の表示	namsvinf	△	システム共通定義の all_node オペランド, または all_node_ex オペランドに, アドレス解決のできないノード名が指定されていた場合, コマンドの実行に時間が掛かるおそれがあります。

機能	運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
RPC 抑止リストの操作	namblad	◎	—
ノードリスト情報の削除	namndrm	◎	—
マネージャノードの変更	nammstr	◎	—
ノードリストファイルの作成	namnlcre	◎	—
ノードリストファイルの内容表示	namnldsp	◎	—
ノードリストファイルの削除	namnldel	◎	—
ノードのオプション情報の変更	namndopt	◎	—
メッセージログ管理	メッセージログファイルの内容表示	◎	—
	メッセージログのリアルタイム出力機能の切り替え	◎	—
監査ログ管理	監査ログ機能の環境設定	dcauditsetup	—
OpenTP1 ファイル管理	OpenTP1 ファイルシステムの初期設定	filmkfs	—
	OpenTP1 ファイルシステムの状態表示	filstatfs	△ -w オプションは、サポートしていません。
	OpenTP1 ファイルシステムの内容表示	fills	△ -H オプションと -L オプションを同時に指定して実行した場合でも、ロックを掛けているプロセスのプロセス ID は表示できません。 詳細については、「fills (OpenTP1 ファイルシステムの内容表示)」を参照してください。
	OpenTP1 ファイルシステムのバックアップ	filbkup	◎
	OpenTP1 ファイルシステムのリストア	filrstr	◎

7. 運用コマンド

運用コマンドの一覧

機能	運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	OpenTP1 ファイルグループの変更	×	サポートしていません。
	OpenTP1 ファイルのアクセス許可モードの変更	×	
	OpenTP1 ファイル所有者の変更	×	
ステータスファイル管理	ステータスファイルの作成, 初期設定	◎	—
	ステータスファイルの状態表示	◎	—
	ステータスファイルの内容表示	◎	—
	ステータスファイルのオープン	◎	—
	ステータスファイルのクローズ	◎	—
	ステータスファイルの削除	◎	—
	ステータスファイルのスワップ	◎	—
ジャーナル関係のファイル管理	ジャーナル関係のファイルの初期設定	◎	—
	ジャーナル関係のファイル情報の表示	△	次の内容はサポートしていません。 • <code>-j</code> オプションの指定値 <code>jar</code> • <code>-r</code> オプション
	再開開始中読み込み済みジャーナル関係のファイル情報の表示	◎	—
	ジャーナル関係のファイルのオープン	△	次の内容はサポートしていません。 • <code>-j</code> オプションの指定値 <code>jar</code> • <code>-r</code> オプション • <code>-e</code> オプション
	ジャーナル関係のファイルのクローズ	△	

機能	運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
ジャーナル関係の物理ファイルの割り当て	jnladdpf	◎	—
ジャーナル関係の物理ファイルの削除	jnldelpf	◎	—
ジャーナル関係のファイルのスワップ	jnlswpfg	△	次の内容はサポートしていません。 • -j オプションの指定値 jar • -r オプション
ジャーナル関係のファイルの削除	jnlrm	◎	—
ジャーナル関係のファイルのステータス変更	jnlchgfg	△	次の内容はサポートしていません。 • -j オプションの指定値 jar • -S オプション
ジャーナル関係のファイルのアンロード	jnlunlfg	△	-j オプションの指定値 jar はサポートしていません。また、-o オプション（出力先ファイル名の指定）は、省略できません。 詳細については、「jnlunlfg（ジャーナル関係のファイルのアンロード）」を参照してください。
自動アンロード機能の制御	jnlatlunl	◎	—
ジャーナル関係のファイルの回復	jnlmkrf	◎	—
ファイル回復用ジャーナルの集積	jnlcolc	△	Windows 版 OpenTP1 固有の -z オプション（出力ファイル名の指定）は、省略できません。コマンド引数のファイル名の指定は省略できません。また、-k オプションの指定値 i はサポートしていません。 詳細については、「jnlcolc（ファイル回復用ジャーナルの集積）」を参照してください。
アンロードジャーナルファイルの複写	jnlcopy	△	Windows 版 OpenTP1 固有の -z オプション（出力ファイル名の指定）は、省略できません。コマンド引数のアンロードジャーナルファイル名の指定は省略できません。また、-o オプションはサポートしていません。 詳細については、「jnlcopy（アンロードジャーナルファイルの複写）」を参照してください。
アーカイブ状態の表示	jnlarls	×	サポートしていません。

7. 運用コマンド

運用コマンドの一覧

機能	運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異	
アンロードジャーナルファイル、またはグローバルアーカイブアンロードジャーナルファイルの編集出力	jnledit	△	コマンド引数のファイル名の指定は省略できません。また、-o オプションはサポートしていません。 詳細については、「jnledit（アンロードジャーナルファイル、またはグローバルアーカイブアンロードジャーナルファイルの編集出力）」を参照してください。	
アンロードジャーナルファイル、またはグローバルアーカイブアンロードジャーナルファイルのレコード出力	jnlrput	△	Windows 版 OpenTP1 固有の -z オプション（出力ファイル名の指定）は、省略できません。また、コマンド引数のファイル名の指定も省略できません。 -q _trn オプション、および -o オプションはサポートしていません。 詳細については、「jnlrput（アンロードジャーナルファイル、またはグローバルアーカイブアンロードジャーナルファイルのレコード出力）」を参照してください。	
アンロードジャーナルファイル、およびグローバルアーカイブアンロードジャーナルファイルの時系列ソート、およびマージ	jnlsort	△	Windows 版 OpenTP1 固有の -z オプション（出力ファイル名の指定）は、省略できません。コマンド引数のファイル名の指定は省略できません。また、このコマンドは、-g オプションを指定した場合にだけ実行できます。 詳細については、「jnlsort（アンロードジャーナルファイル、およびグローバルアーカイブアンロードジャーナルファイルの時系列ソート、およびマージ）」を参照してください。	
稼働統計情報の出力	jnlstts	△	コマンド引数のファイル名の指定は省略できません。 詳細については、「jnlstts（稼働統計情報の出力）」を参照してください。	
MCF 稼働統計情報の出力	jnlmcst	△	コマンド引数のファイル名の指定は省略できません。また、-o オプションはサポートしていません。 詳細については、「jnlmcst（MCF 稼働統計情報の出力）」を参照してください。	
リソースグループの接続の強制解除	jnlardis	×	サポートしていません。	
DAM ファイル管理	物理ファイルの初期設定	damload	◎	—
	論理ファイルの状態表示	damlsl	◎	—

機能		運用コマンド	Windows 版 OpenTP1 で のサポート の有無	UNIX 版 OpenTP1 との差異
	論理ファイルの追加	damadd	◎	—
	論理ファイルの切り離し	damrm	◎	—
	論理ファイルの論理閉塞	damhold	◎	—
	論理ファイルの閉塞解除	damrles	◎	—
	物理ファイルの削除	damdel	◎	—
	物理ファイルのバックアップ	dambkup	△	-s オプションは、サポートしていません。
	物理ファイルのリストア	damrstr	△	
	論理ファイルの回復	damfre	◎	—
	キャッシュブロック数のしきい値の設定	damchdef	◎	—
	キャッシュブロック数の取得	damchinf	◎	—
TAM ファイル 管理	TAM ファイルの初期設定	tamcre	◎	—
	TAM テーブルの状態表示	tamls	◎	—
	TAM テーブルの追加	tamadd	◎	—
	TAM テーブルの切り離し	tamrm	◎	—
	TAM テーブルの論理閉塞	tamhold	◎	—
	TAM テーブルの閉塞解除	tamrles	◎	—
	TAM テーブルのロード	tamload	◎	—
	TAM テーブルのアンロード	tamunload	◎	—
	TAM ファイルの削除	tamdel	◎	—

7. 運用コマンド

運用コマンドの一覧

機能	運用コマンド	Windows 版 OpenTP1 で のサポート の有無	UNIX 版 OpenTP1 との差異
TAM ファイルの バックアップ	tambkup	△	-s オプションは、サポートしていません。
TAM ファイルの リストア	tamrstr	△	
TAM ファイルの 回復	tamfre	◎	—
TAM 排他資源名 称の変換	tamlckls	◎	—
ハッシュ形式の TAM ファイルお よび TAM テーブ ルのシノニム情報 の表示	tamhsls	◎	—
メッ セージ キュー ファイ ル管理	キューグループの 状態表示	◎	—
	メッセージキュー 用物理ファイルの 割り当て	◎	—
	メッセージキュー 用物理ファイルの 削除	◎	—
リソー スマネ ジャ管 理	リソースマネジャ の情報の表示	◎	—
	リソースマネジャ の登録	△	各オプションで指定するリソースマネ ジャとして、OpenTP1_ISM、および OpenTP1_XATMI は指定できません。 また、-P オプションを指定する必要は ありません。 詳細については、「trnlncrm (リソース マネジャの登録)」を参照してくださ い。
	トランザクション 制御用オブジェク トファイルの作成	△	-R オプションに指定するリソースマネ ジャとして、OpenTP1_ISM は指定で きません。また、-P オプションはサ ポートしていません。 詳細については、「trnmkobj (トランザ クション制御用オブジェクトファイル の作成)」を参照してください。

	機能	運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
トレース管理	UAP トレースの編集出力	uatdump	△	デフォルトで UAP トレースファイルが出力されます。UAP トレースファイルの出力場所とファイル名については、マニュアル「OpenTP1 運用と操作」の OpenTP1 が出力するファイル一覧についての説明を参照してください。
	RPC トレースのマージ	rpcmrg	△	Windows 版 OpenTP1 固有の -z オプション（出力ファイル名の指定）は、省略できません。詳細については、「rpcmrg（RPC トレースのマージ）」を参照してください。
	RPC トレースの出力	rpcdump	△	-r オプションを指定した場合、必ず出力ファイル名を指定してください。詳細については、「rpcdump（RPC トレースの出力）」を参照してください。
	共用メモリダンプの出力	usmdump	◎	—
リモート API 管理	リモート API 機能の実行環境の設定	rapsetup	◎	—
	リモート API 機能に使用する定義の自動生成	rapdfgen	◎	—
	rap リスナーおよび rap サーバの状態表示	rapls	◎	—
性能検証用トレース管理	トレース情報ファイルの編集出力	prfed	◎	—
	トレース情報ファイルの取り出し	prfget	◎	—
リアルタイム統計情報サービス管理	RTS ログファイルの編集出力	rtsedit	△	-v オプションで、サービス以外の処理単位で取得したリアルタイム統計情報だけを出力したい場合に、"*****" は指定できません。
	リアルタイム統計情報の標準出力への出力	rtsls	△	
		リアルタイム統計情報サービスの実行環境の設定	rtsssetup	◎

7. 運用コマンド

運用コマンドの一覧

機能		運用コマンド	Windows 版 OpenTP1 で のサポート の有無	UNIX 版 OpenTP1 との差異
	リアルタイム統計 情報の設定変更	rtsstats	◎	—
OpenT P1 解 析支援	性能検証用トレース 情報解析	dcalzprf	◎	—
コネク ション 管理	コネクションの状態表示	mcfctlscn	◎	—
	コネクションの確立	mcfactctn	◎	—
	コネクションの解放	mcfldctn	◎	—
	コネクションの切り替え	mcfctchn	◎	—
	ネットワークの状態表示	mcfctlsln	◎	—
	サーバ型コネクションの確立要求の受付開始	mcfctonln	◎	—
	サーバ型コネクションの確立要求の受付終了	mcfctofln	◎	—
	メッセージ多重処理状況の表示	mcfctlstrd	◎	—
アプリ ケー ション 管理	アプリケーションの状態表示	mcfalsap	◎	—
	アプリケーションの閉塞	mcfadctap	◎	—
	アプリケーションの閉塞解除	mcfactap	◎	—
	アプリケーション異常終了回数の初期化	mcfacldcap	◎	—
	アプリケーションに関するタイマ起動要求の表示	mcfalstap	◎	—
	アプリケーションに関するタイマ起動要求の削除	mcfadltap	◎	—

機能		運用コマンド	Windows 版 OpenTP1 で のサポート の有無	UNIX 版 OpenTP1 との差異
アプリケーション 運用支援	アプリケーション プログラムの起動	mcftuevt	◎	—
論理端 末管理	論理端末の状態表示	mcftlsle	◎	—
	論理端末の閉塞	mcftdctle	◎	—
	論理端末の閉塞解除	mcftactle	◎	—
	論理端末のメッセージキューの先頭スキップ	mcftspqle	◎	—
	論理端末の出力 キュー処理の保留	mcfthldoq	◎	—
	論理端末の出力 キュー処理の保留 解除	mcftlrsoq	◎	—
	論理端末の出力 キュー削除	mcftdlqle	◎	—
	論理端末に関する メッセージジャー ナルの取得開始	mcftactmj	◎	—
	論理端末に関する メッセージジャー ナルの取得終了	mcftdctmj	◎	—
	論理端末に対する 継続問い合わせ応 答処理の強制終了	mcftendct	◎	—
	代行送信の開始	mcftstalt	◎	—
	代行送信の終了	mcftedalt	◎	—
サービ スグ ループ 管理	サービスグループ の状態表示	mcftlssg	◎	—
	サービスグループ の閉塞	mcftdctsg	◎	—
	サービスグループ の閉塞解除	mcftactsg	◎	—
	サービスグループ の入力キュー処理 の保留	mcfthldiq	◎	—

7. 運用コマンド

運用コマンドの一覧

機能		運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	サービスグループの入力キュー処理の保留解除	mcftlsiq	◎	—
	サービスグループの入力キュー削除	mcftdlqsg	◎	—
サービス管理	サービスの状態表示	mcftlssv	◎	—
	サービスの閉塞	mcftdctsv	◎	—
	サービスの閉塞解除	mcftactsv	◎	—
セッション管理	セッションの開始	mcftactss	◎	—
	セッションの終了	mcftdetss	◎	—
バッファ管理	バッファグループの使用状況表示	mcftlsbuf	◎	—
マップ管理	マップファイルのパス名変更	dcmapchg	◎	—
	マップファイルのロード済み資源の表示	dcmapls	◎	—
キュー管理	入出力キューの内容複写	mcftdmpqu	◎	—
MCF トレース取得 管理	MCF トレースファイルの強制スワップ	mcftswptr	◎	—
	MCF トレース取得の開始	mcftstrtr	◎	—
	MCF トレース取得の終了	mcftstptr	◎	—
MCF 稼働統計情報 管理	MCF 稼働統計情報の編集	mcfreport	×	サポートしていません。
	MCF 稼働統計情報の出力	mcfststats	×	
MCF 通信サービス管理	MCF 通信サービスの部分停止	mcftstop	◎	—

機能		運用コマンド	Windows 版 OpenTP1 でのサポートの有無	UNIX 版 OpenTP1 との差異
	MCF 通信サービスの部分開始	mcftstart	◎	—
	MCF 通信サービスの状態参照	mcftlscom	◎	—
ユーザタイマ管理	ユーザタイマ監視の状態表示	mcftlsutm	◎	—
マルチ OpenTP1 管理	マルチ OpenTP1 のセットアップ状態の確認	dccls	○	Windows 版 OpenTP1 固有のコマンドです。 詳細については、「dccls (マルチ OpenTP1 のセットアップ状態の確認)」を参照してください。
	マルチ OpenTP1 操作コンソールの作成	dcmakecon	○	Windows 版 OpenTP1 固有のコマンドです。 詳細については、「dcmakecon (マルチ OpenTP1 操作コンソールの作成)」を参照してください。
	マルチ OpenTP1 のセットアップとアンセットアップ	dcsetupml	○	Windows 版 OpenTP1 固有のコマンドです。 詳細については、「dcsetupml (マルチ OpenTP1 のセットアップとアンセットアップ)」を参照してください。
リソースマネージャ管理	トランザクション制御用 DLL の作成	njsmkdll	○	Windows 版 OpenTP1 固有のコマンドです。 詳細については、「njsmkdll (トランザクション制御用 DLL の作成)」を参照してください。
システム管理	OpenTP1 の起動	ntbstart	○	Windows 版 OpenTP1 固有のコマンドです。 詳細については、「ntbstart (OpenTP1 の起動)」を参照してください。
	指定ファイル内容のコンソール出力	ntbtail	○	Windows 版 OpenTP1 固有のコマンドです。 詳細については、「ntbtail (指定ファイル内容のコンソール出力)」を参照してください。
	標準出力リダイレクトファイルのコンソール出力	tp1console	○	Windows 版 OpenTP1 固有のコマンドです。 詳細については、「tp1console (標準出力リダイレクトファイルのコンソール出力)」を参照してください。

コマンドの説明形式

各コマンドで説明する項目を次に示します。

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

コマンドの形式を説明しています。Windows 版 OpenTP1 固有のオプションと引数、および UNIX 版 OpenTP1 と差異があるオプションと引数は、**太字**で示しています。

機能

コマンドの機能について説明しています。

オプション

コマンドのオプションについて説明しています。

コマンド引数

コマンドのコマンド引数について説明しています。

出力形式

コマンドの出力形式について説明しています。

出力メッセージ

コマンドの出力メッセージについて説明しています。

注意事項

コマンドの注意事項について説明しています。

UNIX 版 OpenTP1 と共通のコマンドの詳細

UNIX 版 OpenTP1 と共通のコマンドについて、差異の詳細を以降で説明します。

注意事項

Windows 7 以降、または Windows Server 2008 R2 以降で OpenTP1 のコマンドを実行する場合にコマンドプロンプトを起動するときは、「管理者として実行」を選択してください。通常のコマンドプロンプトからコマンドを実行した場合、OpenTP1 のコマンドが正常に動作しないことがあります。

dcpplist (製品情報の表示)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

出力形式

#	Product ID	Version	Product Name
---	-----	-----	-----
aaa	bb...bb	cc...cc	dee...ee

- d: インストールした製品が TP1/Server Base をインストールしたフォルダと同じフォルダにインストールされている場合は '*' を、それ以外の場合は '△' (半角スペース) を表示します。
- ee...ee: 製品名 (50 バイト以内)

これ以外に差異はありません。

注意事項

OpenTP1 関連製品の情報、および OpenTP1 以外の日立オープンミドルウェア製品の情報についても一部表示されます。

dcrasget (保守資料の取得)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

dcrasget [-g] [-l] 取得先ディレクトリ

コマンド引数

●取得先ディレクトリ ~<パス名>

指定したフォルダ下に、次に示すフォルダが生成されて、保守資料が格納されます。

dcrasget.HOSTNAME.mmddHHMMSS.AAA

HOSTNAME：標準ホスト名

mmddHHMMSS：月日時分秒

AAA：取得される情報単位

SAV：%DCDIR%¥spool¥save フォルダ下の情報

SPL：%DCDIR%¥spool フォルダ下の情報（save を除く）

TMP：%DCDIR%¥tmp フォルダ下の情報

CNF：%DCDIR%¥conf フォルダなど定義関係の情報

INF：情報ファイルなどの情報

これ以外に差異はありません。

注意事項

dcrasget コマンド内部では、OpenTP1 がオンライン中にだけ実行できるコマンドも実行するため、イベントビューアにエラーメッセージが出力されることがあります。メッセージについては、マニュアル「OpenTP1 メッセージ」を参照してください。

fills (OpenTP1 ファイルシステムの内容表示)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

出力形式

- -H オプションと -L オプションを指定した場合

MODE	UID	GID	PID	L	TIME	FILE
aabbcc	dd...dd	ee...ee	pppp	q	hh...hh	ii...ii

- pppp: ロックを掛けているプロセスのプロセス ID
Windows 版 OpenTP1 では、常に「0」が表示されます。

これ以外に差異はありません。

jnlcolc (ファイル回復用ジャーナルの集積)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
jnlcolc -k 回復種別 -z 出力ファイル名 [-f] [-l] [-n] [-m]
          [-i オンラインバックアップ情報名] [-c キー]
          ファイル名 [ [△ファイル名] ...]
```

機能

指定したファイルから、DAM ファイル、または TAM ファイルの回復に必要なジャーナルレコードを抽出し、ファイルに出力します。

オプション

● -k 回復種別

回復対象とするファイル種別に、i (ISAM ファイル) は指定できません。これ以外に差異はありません。

● -z 出力ファイル名 ~ 〈パス名〉

実行結果を出力するファイルの名称を指定します。このオプションは省略できません。

コマンド引数

● ファイル名 ~ 〈パス名〉

このコマンド引数は省略できません。これ以外に差異はありません。

jnlcopy (アンロードジャーナルファイルの複写)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
jnlcopy -z 出力ファイル名 [-t [開始] [, 終了] ]  
          [-j レコード種別 [レコード種別] ...]  
          アンロードジャーナルファイル名  
          { [△アンロードジャーナルファイル名] ...}
```

機能

指定したアンロードジャーナルファイル内の情報を、ファイルに出力します。

オプション

● -z 出力ファイル名 ~ 〈パス名〉

実行結果を出力するファイルの名称を指定します。このオプションは省略できません。

コマンド引数

● アンロードジャーナルファイル名 ~ 〈パス名〉

このコマンド引数は省略できません。これ以外に差異はありません。

jnledit (アンロードジャーナルファイル, またはグローバルアーカイブアンロードジャーナルファイルの編集出力)

jnledit (アンロードジャーナルファイル, またはグローバルアーカイブアンロードジャーナルファイルの編集出力)

このマニュアルでは, UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については, マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
jnledit [-e 編集種別] [-t [開始] [, 終了] ]
        [-j レコード種別 [レコード種別] ...] [-s サーバ名]
        [-v サービス名]
        [-u トランザクショングローバル識別子
          [, トランザクションブランチ識別子] ]
        [-m 論理端末名称 [, 論理端末名称] ...]
        [-w けた] [-l 行] [-c]
        ファイル名 [ (△ファイル名) ...]
```

機能

指定したアンロードジャーナルファイル, またはグローバルアーカイブアンロードジャーナルファイル内の情報を編集し, ファイルに出力します。

コマンド引数

●ファイル名 ～〈パス名〉

このコマンド引数は省略できません。これ以外に差異はありません。

7. 運用コマンド

jnlmcst (MCF 稼働統計情報の出力)

jnlmcst (MCF 稼働統計情報の出力)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
jnlmcst [-e 編集種別] [-l 行] [-i 時間間隔]  
        [-t [開始] [, 終了] ] [-m 論理端末名称]  
        [-a アプリケーション名称]  
        ファイル名
```

機能

指定したアンロードジャーナルファイル、またはグローバルアーカイブジャーナルファイルから、MCF 稼働統計情報を収集し、編集後、ファイルに出力します。

出力する情報を次に示します。

- MCF 稼働統計情報のメッセージ受信系、および送信系情報

コマンド引数

●ファイル名 ~ 〈パス名〉

このコマンド引数は省略できません。これ以外に差異はありません。

jnlrput (アンロードジャーナルファイル, またはグローバルアーカイブアンロードジャーナルファイルのレコード出力)

jnlrput (アンロードジャーナルファイル, またはグローバルアーカイブアンロードジャーナルファイルのレコード出力)

このマニュアルでは, UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については, マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
jnlrput -z 出力ファイル名 [-t [開始] [, 終了]]
        [-e] [-f] [-l] [-c キー]
        [-u トランザクショングローバル識別子
          [, トランザクションブランチ識別子]]
        [-j レコード種別 [レコード種別] ...] [-q _rpc]
        [-d 任意抽出条件] [-x] ファイル名 { [Δファイル名] ...}
```

機能

指定したアンロードジャーナルファイル, またはグローバルアーカイブアンロードジャーナルファイル内のユーザジャーナルレコードの情報, トランザクションブランチの CPU 使用時間情報, およびレスポンス統計情報をファイルに出力します。

オプション

● -z 出力ファイル名 ~〈パス名〉

実行結果を出力するファイルの名称を指定します。このオプションは省略できません。

コマンド引数

●ファイル名 ~〈パス名〉

このコマンド引数は省略できません。これ以外に差異はありません。

7. 運用コマンド

jnlsort (アンロードジャーナルファイル, およびグローバルアーカイブアンロードジャーナルファイルの時系列ソート, およびマージ)

jnlsort (アンロードジャーナルファイル, およびグローバルアーカイブアンロードジャーナルファイルの時系列ソート, およびマージ)

このマニュアルでは, UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については, マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
jnlsort -z 出力ファイル名 [-n ノード識別子] -g  
          [-i 〔開始ジャーナルサーバランID〕  
          〔終了ジャーナルサーバランID〕]  
          ファイル名 [△ファイル名] ...
```

機能

指定したアンロードジャーナルファイル, およびグローバルアーカイブアンロードジャーナルファイルの内容をマージします。結果は, グローバルアーカイブアンロードジャーナルファイルの形式で, ファイルに出力します。

このコマンドは, 必ず `-g` オプションを指定して実行してください。 `-g` オプションを指定しない場合の動作は保証できません。

オプション

● `-z` 出力ファイル名 ~ 〈パス名〉

実行結果を出力するファイルの名称を指定します。このオプションは省略できません。

● `-g`

このコマンドを実行する場合は, 必ず指定してください。これ以外に差異はありません。

コマンド引数

● ファイル名 ~ 〈パス名〉

このコマンド引数は省略できません。これ以外に差異はありません。

jnlstts (稼働統計情報の出力)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
jnlstts [-e 編集項目] [-u 編集単位] [-l 行] [-c]
        [-i 時間間隔] [-t [開始] [, 終了]]
        [-s {サーバ名 | サービス名}]
        ファイル名
```

機能

指定したアンロードジャーナルファイル、またはグローバルアーカイブアンロードジャーナルファイルから統計情報を収集し、編集後、ファイルに出力します。

出力する情報を次に示します。

- ・システム統計情報（システム統計レコード内の各コンポーネント情報）
- ・トランザクション統計情報（システム統計レコード内の各トランザクションの情報）
- ・レスポンス統計情報（システム統計レコード内の各レスポンスタイムの情報）
- ・通信遅延時間統計情報（システム統計レコード内の通信遅延に関する情報）

コマンド引数

●ファイル名 ~ 〈パス名〉

このコマンド引数は省略できません。これ以外に差異はありません。

注意事項

RPC 送信元・RPC 送信先の OS の時刻精度によっては、時刻戻しが発生しなくても、通信時間が OS の時刻精度より短い場合、通信遅延時間統計情報の編集結果のデータがマイナス値になることがあります。取得編集時間帯のデータにマイナス値が含まれているときは、編集対象時間帯の最大値、最小値、および平均値に '*' を出力します。

jnlunlfg (ジャーナル関係のファイルのアンロード)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
jnlunlfg -j sys [ [-f] | [-t [開始] [, 終了] ] ]  
          [-r リソースグループ] [-R ランID]  
          [-g ファイルグループ名]  
          -o 出力先ファイル名  
          [-d サービス定義名] [-n]
```

機能

指定したファイルグループを出力先のファイルにアンロードし、ファイルグループをアンロード済みにします。これ以外に差異はありません。

オプション

● -j sys

アンロードするジャーナル関係のファイルとして、jar (アーカイブジャーナルファイル) は指定できません。これ以外に差異はありません。

● -o 出力先ファイル名 ~ 〈パス名〉

このオプションは省略できません。これ以外に差異はありません。

rpcdump (RPC トレースの出力)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
rpcdump [ {-r 出力ファイル名 | -m} ] [ {-c | -f} ] [-d 電文長]
        [-t [開始] [, 終了] ]
        [-s サービスグループ名 [, サービス名] ...]
        [-b ノード識別子 [, ノード識別子] ...]
        [-v サーバ名 [, サーバ名] ...]
        [-p プロセスID [, プロセスID] ...]
        [-x xid [, xid] ...]
        [-n [開始トレース番号] [, 終了トレース番号] ]
        [RPCトレースファイル名]
```

機能

指定した RPC トレースファイルのトレース情報を編集して、ファイルに出力します。

オプション

● -r 出力ファイル名 ~ 〈パス名〉

実行結果を出力するファイルの名称を指定します。このオプションは省略できません。
これ以外に差異はありません。

rpcmrg (RPC トレースのマージ)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

rpcmrg -z 出力ファイル名
RPC トレースファイル名 [[△RPC トレースファイル名] …]

機能

指定した複数の RPC トレースファイルのトレース情報は時系列に並べられ、RPC トレースファイルの形式でファイルに出力されます。

指定した複数の RPC トレースファイルに同じトレース情報がある場合は、重複した情報を削除します。

オプション

● **-z 出力ファイル名** ~ 〈パス名〉

実行結果を出力するファイルの名称を指定します。このオプションは省略できません。

trnlnkrm (リソースマネージャの登録)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
trnlnkrm
{ [-A 追加するOpenTP1提供RM名 [,追加するOpenTP1提供RM名] ...]
  [-D 削除するOpenTP1提供RM名 [,削除するOpenTP1提供RM名] ...]
  [-a 追加するOpenTP1提供以外のRM名
    [,追加するOpenTP1提供以外のRM名] ...]
  [-s RMスイッチ名 [,RMスイッチ名] ...]
  [-o "RM関連オブジェクト名 [△RM関連オブジェクト名] ..."
    [, "RM関連オブジェクト名 [△RM関連オブジェクト名] ..." ] ...]
  [-d 削除するOpenTP1提供以外のRM名
    [,削除するOpenTP1提供以外のRM名] ...]
  | [-n] }
[-C "コンパイルオプション名 [△コンパイルオプション名] ..."]
[-B "リンケージオプション名 [△リンケージオプション名] ..."] [-L] [-f] [-P]
```

オプション

● -A 追加する OpenTP1 提供 RM 名 ～〈1 ～ 31 文字の英数字〉

追加する OpenTP1 提供のリソースマネージャの名称として、OpenTP1_ISM, OpenTP1_XATMI は指定できません。これ以外に差異はありません。

● -D 削除する OpenTP1 提供 RM 名 ～〈1 ～ 31 文字の英数字〉

削除する OpenTP1 提供のリソースマネージャの名称として、OpenTP1_ISM, OpenTP1_XATMI は指定できません。これ以外に差異はありません。

● -o RM 関連オブジェクト名 ～〈英数字〉

複数のオブジェクトを指定する場合、ダブルクォーテーション (") で囲ってください。また、RM 関連オブジェクト名称をパスで指定する場合、区切り文字には、'¥¥' を指定してください。これ以外に差異はありません。

● -C コンパイルオプション名 ～〈1 ～ 512 文字の文字列〉

コンパイルオプション名を指定する場合、ダブルクォーテーション (") で囲ってください。COBOL2002 で UAP を作成する場合は、必ず -C"/ZI" オプションを指定してください。これ以外に差異はありません。

● -B リンケージオプション名 ～〈1 ～ 512 文字の文字列〉

ライブラリリンケージ実行時に使用するリンケージオプション名を指定する場合、ダブルクォーテーション (") で囲ってください。これ以外に差異はありません。

7. 運用コマンド

trnlnkrm (リソースマネージャの登録)

● -P

デフォルトで指定されているものと見なされるため、指定する必要はありません。これ以外に差異はありません。

注意事項

- このコマンドの実行には、C コンパイラが必要です。コマンド実行時に、Visual Studio の cl コマンド、link コマンド、および mt コマンドを起動するため、あらかじめ環境変数 PATH にコマンドの格納先フォルダを指定してください。また、コンパイラを正常に動作させるための環境変数 LIB、および環境変数 INCLUDE の指定も必要です。詳細については、リリースノートを参照してください。
- このコマンドで作成される実行形式ファイルには、-B オプションに /MANIFEST を指定した場合、マニフェストファイルが埋め込まれます。マニフェストファイルについては、Windows のマニュアルを参照してください。
- このコマンドを実行する場合、必ず環境変数 PATH、環境変数 LIB、および環境変数 INCLUDE の先頭に Visual Studio のパスを記述してください。記述しない場合、意図しないリンケージが行われるおそれがあります。

trnmkobj (トランザクション制御用オブジェクトファイルの作成)

このマニュアルでは、UNIX 版 OpenTP1 との差異についてだけ説明します。差異がない部分については、マニュアル「OpenTP1 運用と操作」を参照してください。

形式

```
trnmkobj -o トランザクション制御用オブジェクト名  
          [-R OpenTP1提供RM名 [,OpenTP1提供RM名] ...]  
          [-r OpenTP1提供以外のRM名 [,OpenTP1提供以外のRM名] ...]  
          [-C "コンパイルオプション名 [△コンパイルオプション名] ..."]  
          [-l]
```

オプション

● -R OpenTP1 提供 RM 名 ～ 〈1 ～ 31 文字の英数字〉

UAP からアクセスする OpenTP1 提供のリソースマネージャの名称として、OpenTP1_ISM は指定できません。これ以外に差異はありません。

● -C コンパイルオプション名 ～ 〈1 ～ 512 文字の文字列〉

コンパイルオプション名を指定する場合、ダブルクォーテーション (") で囲ってください。COBOL2002 で UAP を作成する場合は、必ず -C"/ZI" オプションを指定してください。これ以外に差異はありません。

Windows 版 OpenTP1 固有のコマンドの詳細

Windows 版 OpenTP1 固有のコマンドの詳細を以降で説明します。

! 注意事項

Windows 7 以降，または Windows Server 2008 R2 以降で OpenTP1 のコマンドを実行する場合にコマンドプロンプトを起動するときは，「管理者として実行」を選択してください。通常のコマンドプロンプトからコマンドを実行した場合，OpenTP1 のコマンドが正常に動作しないことがあります。

dcdls (マルチ OpenTP1 のセットアップ状態の確認)

形式

dcdls [-d OpenTP1ディレクトリ]

機能

現在セットアップされているマルチ OpenTP1 を含む OpenTP1 識別子、およびセットアップフォルダを表示します。OpenTP1 のオンライン、オフラインに関係なく実行できます。

このコマンドは、オリジナルの OpenTP1 のコンソールからだけ実行できます。

オプション

● -d OpenTP1 ディレクトリ

指定したフォルダが、マルチ OpenTP1 のセットアップフォルダである場合、マルチ OpenTP1 の識別子を表示します。

オリジナルの OpenTP1 のインストールフォルダを指定した場合は、「original」と表示されます。

出力形式

● -d オプションを指定しない場合

識別子	ディレクトリ
-----	aa...aa
-----	bb...bb

- aa...aa：オリジナルの OpenTP1 のインストールフォルダ
- bb...bb：マルチ OpenTP1 のセットアップフォルダ

● -d オプションを指定した場合

識別子	ディレクトリ
original	aa...aa
bbbb	cc...cc

- aa...aa：オリジナルの OpenTP1 のインストールフォルダ
- bbbb：マルチ OpenTP1 の識別子（4 文字以内）
- cc...cc：マルチ OpenTP1 のセットアップフォルダ

7. 運用コマンド

dccls (マルチ OpenTP1 のセットアップ状態の確認)

出力メッセージ

メッセージ ID	内容	出力先
KFCA26551-E	コマンドの実行に失敗しました	標準エラー出力
KFCA26552-E	コマンド引数の指定に誤りがあります	標準エラー出力
KFCA26554-I	ヘルプメッセージ	標準エラー出力
KFCA26556-E	指定したフォルダは OpenTP1 フォルダではありません	標準エラー出力

注意事項

このコマンドの実行時に重大な障害が発生した場合、できるだけ速やかに次のデータを退避した上で、保守員に連絡してください。

- %TEMP%\%dcsetupml.log
- %TEMP%\%trnltnkrm.log
- %DCDIR%\%spool%\%errlog%*. * (%DCDIR% は、オリジナルの OpenTP1 のインストールフォルダ)

dcmakecon (マルチ OpenTP1 操作コンソールの作成)

形式

dcmakecon [識別子]

機能

OpenTP1 識別子に対応する OpenTP1 でコマンドを実行するための新しいコンソールを作成します。識別子の指定を省略した場合は、オリジナルの OpenTP1 でコマンドを実行するための新しいコンソールを作成します。OpenTP1 のオンライン、オフラインに関係なく実行できます。

このコマンドは、オリジナルの OpenTP1 のコンソールからだけ実行できます。

マルチ OpenTP1 では、オリジナルの OpenTP1 でこのコマンドを実行して作成したコンソールからコマンドを実行します。ただし、次のコマンドはマルチ OpenTP1 のコンソールからは実行できません。

- dcdls
- dcmakecon
- dcsetupml

このコマンドを実行して新しく作成したコンソールでは、次に示す表のとおり環境変数が設定されます。なお、この設定値は、設定したコンソールの中でだけ有効となります。

表 7-2 コンソール中での環境変数の設定

環境変数	設定
DCCONFPATH	"OpenTP1 識別子に対応する OpenTP1 セットアップフォルダ ¥conf¥env" で定義した環境変数 DCCONFPATH と同じ値が設定されます。
DCDIR	"OpenTP1 識別子に対応する OpenTP1 セットアップフォルダ" が設定されます。
DCUAPCONFPATH	"OpenTP1 識別子に対応する OpenTP1 セットアップフォルダ ¥conf¥env" で定義した環境変数 DCUAPCONFPATH と同じ値が設定されます。
INCLUDE	"OpenTP1 識別子に対応する OpenTP1 セットアップフォルダ ¥include" が先頭に追加されます。
LIB	"OpenTP1 識別子に対応する OpenTP1 セットアップフォルダ ¥lib" が先頭に追加されます。
PATH	対応する OpenTP1 のコマンドを最優先で実行するために、"OpenTP1 識別子に対応する OpenTP1 セットアップフォルダ ¥bin" が先頭に追加されます。

7. 運用コマンド

dcmakecon (マルチ OpenTP1 操作コンソールの作成)

コマンド引数

●識別子 ～ 〈1 ～ 4 文字の半角英数字〉

マルチ OpenTP1 をセットアップしたときに設定した文字列を指定します。

コマンド引数を指定しないで実行した場合は、オリジナルの OpenTP1 を操作するためのコンソールが作成されます。

出力メッセージ

メッセージ ID	内容	出力先
KFCA26551-E	コマンドの実行に失敗しました	標準エラー出力
KFCA26552-E	コマンド引数の指定に誤りがあります	標準エラー出力
KFCA26555-I	ヘルプメッセージ	標準エラー出力

注意事項

- マルチ OpenTP1 を運用する場合は、必ずこのコマンドで作成したコンソールから環境変数を変更してください。それ以外の方法で環境変数を変更した場合の動作は、保証できません。
- dcsetupml コマンドを実行すると、インストールされたすべてのプログラムプロダクトがマルチ OpenTP1 にコピーされます。TP1/Server Base 以外のプログラムプロダクト (TP1/FS/Direct Access, TP1/FS/Table Access, TP1/Message Control など) もこのコマンドで作成したコンソール上で操作してください。
- マルチ OpenTP1 システムで、コントロールパネルの「アプリケーションの追加と削除」によってオリジナルの OpenTP1 をアンインストールした場合、すべてのマルチ OpenTP1 が削除されます。マルチ OpenTP1 のセットアップフォルダ下にあるフォルダ (bin, etc, examples, include, lib, spool, tmp, tools) も削除されるため、これらのフォルダ下にユーザが独自にファイルなどを作成している場合は、アンインストール作業の前にバックアップを取得してください。
- このコマンドの実行時に重大な障害が発生した場合、できるだけ速やかに次のデータを退避した上で、保守員に連絡してください。
 - %TEMP%\¥dcsetupml.log
 - %TEMP%\¥trlnlkrm.log
 - %DCDIR%\¥spool¥errlog¥*. * (%DCDIR% は、オリジナルの OpenTP1 のインストールフォルダ)
- このコマンドでマルチ OpenTP1 用のコンソールを作成したあとに、マルチ OpenTP1 の定義や、環境変数 DCCONFPATH、環境変数 DCUAPCONFPATH を変更する場合、変更後にこのコマンドを実行し、マルチ OpenTP1 用のコンソールを再作成してください。

dcsetupml (マルチ OpenTP1 のセットアップとアンセットアップ)

形式

```
dcsetupml [ {-u | -d} ]
           マルチOpenTP1セットアップディレクトリ 識別子
```

機能

指定したフォルダに、マルチ OpenTP1 をセットアップします。OpenTP1 識別子は、同一ノード内の OpenTP1 を識別するために使用され、半角英数字 1 ～ 4 文字でユーザが任意に指定します。このコマンドは、OpenTP1 がオフラインの場合にだけ実行できます。

このコマンドは、オリジナルの OpenTP1 のコンソールからだけ実行できます。

オプション

● -u

コマンド引数に対応するマルチ OpenTP1 を最新の状態に更新します。

● -d

コマンド引数に対応するマルチ OpenTP1 を削除します。

コマンド引数

●マルチ OpenTP1 セットアップディレクトリ ～〈パス名〉

マルチ OpenTP1 をセットアップするフォルダを 50 文字以内で指定します。

●識別子 ～〈1 ～ 4 文字の半角英数字〉

マルチ OpenTP1 を識別するための文字列を指定します。

出力メッセージ

メッセージ ID	内容	出力先
KFCA26551-E	コマンドの実行に失敗しました	標準エラー出力
KFCA26552-E	コマンド引数の指定に誤りがあります	標準エラー出力
KFCA26553-I	ヘルプメッセージ	標準出力, 標準エラー出力

注意事項

- このコマンドは、オリジナルの OpenTP1 およびセットアップ先のマルチ OpenTP1 が停止している状態で実行してください。

7. 運用コマンド

dcsetupm1 (マルチ OpenTP1 のセットアップとアンセットアップ)

- このコマンドは Administrators グループの権限を持つユーザが実行してください。
- このコマンドは、OpenTP1 インストールフォルダに存在するフォルダ (bin, etc, examples, include, lib, tools) をコピーします。ユーザが OpenTP1 インストールフォルダ下の各フォルダのファイルを変更または追加した場合、そのままコピーされるため注意してください。
- このコマンドを実行すると、インストールされたすべてのプログラムプロダクトがマルチ OpenTP1 にコピーされます。TP1/Server Base 以外のプログラムプロダクト (TP1/FS/Direct Access, TP1/FS/Table Access, TP1/Message Control など) も demakecon コマンドで作成したコンソール上で操作してください。
- このコマンドを -u オプションまたは -d オプションを指定して実行する場合、マルチ OpenTP1 のセットアップフォルダ下にあるフォルダ (bin, etc, examples, include, lib, spool, tmp, tools) に、ユーザが独自にファイルなどを作成している場合は、コマンドを実行する前にバックアップを取得してください。
- マルチ OpenTP1 に対して、個別に他社のリソースマネージャを接続していた場合、TP1/Server Base のアンインストールやマルチ OpenTP1 の更新によってリソースマネージャの情報がリセットされます。マルチ OpenTP1 の再セットアップ時や更新時には、再度 trnlncrm コマンドで他社リソースマネージャを接続してください。
- このコマンドの実行時やアンインストール時に、OpenTP1 のコマンドが実行中であったり、OpenTP1 セットアップフォルダ下でコマンドプロンプトを開いていたりすると、フォルダやファイルが残ったり、コピーに失敗したりするおそれがあります。コマンドの実行やフォルダでの作業が完了してから、このコマンドやアンインストールを実行してください。
- マルチ OpenTP1 システムでは、OpenTP1 ごとにネームサービスのポート番号を変えてください。また、そのほかの明示的に指定するポート番号も重複しないようにしてください。ポート番号が重複した場合、イベントビューアに KFCA00107-E メッセージが出力され、OpenTP1 が異常終了するおそれがあります。
- このコマンドの実行中に、[Ctrl] キーと [Break] キーを同時に入力したり、コンソールを閉じたりしてコマンドを停止させないでください。
- このコマンドの実行時に重大な障害が発生した場合、できるだけ速やかに次のデータを退避した上で、保守員に連絡してください。
 - %TEMP%\dcsetupm1.log
 - %TEMP%\trnlncrm.log
 - %DCDIR%\spool\Errlog*. * (%DCDIR% は、オリジナルの OpenTP1 のインストールフォルダ)

njsmkdll (トランザクション制御用 DLL の作成)

形式

```
njsmkdll -d トランザクション制御用オブジェクト名  
          [-R OpenTP1提供RM名 [,OpenTP1提供RM名] ...]  
          [-r OpenTP1提供以外のRM名 [,OpenTP1提供以外のRM名] ...]  
          [-o RM関連オブジェクト名 [△RM関連オブジェクト名] ...]
```

機能

TP1/Extension for .NET Framework を使用し、リソースマネージャと XA インタフェースで連携する場合、このコマンドを実行してトランザクション制御用 DLL を作成する必要があります。

オプション

● -d トランザクション制御用オブジェクト名 ～〈1～12文字の英数字〉

トランザクション制御用 DLL の名称を指定します。このコマンドを実行すると、トランザクション制御用 DLL 名 .dll ファイルが、%DCDIR%\\$spool¥trnrmcmd¥userdll フォルダ下に作成されます。

● -R OpenTP1 提供 RM 名 ～〈1～32文字の英数字〉

UAP からアクセスする OpenTP1 提供のリソースマネージャの名称を指定します。OpenTP1 に登録されていないリソースマネージャは指定できません。指定できる OpenTP1 提供のリソースマネージャ名を次に示します。

OpenTP1_TAM : TP1/FS/Table Access のリソースマネージャ名

OpenTP1_MCF : TP1/Message Control のリソースマネージャ名

複数の RM 名を指定する場合は、RM 名と RM 名との間をコンマ (,) で区切ります。

● -r OpenTP1 提供以外の RM 名 ～〈1～32文字の英数字〉

UAP からアクセスする OpenTP1 提供以外のリソースマネージャの名称を指定します。OpenTP1 に登録されていないリソースマネージャは指定できません。複数の RM 名を指定する場合は、RM 名と RM 名との間をコンマ (,) で区切ります。

● -o RM 関連オブジェクト名 ～〈絶対パス名〉

リソースマネージャに関連のあるオブジェクトファイル (XA インタフェース用オブジェクトファイル) の名称を絶対パス (空白を含まない MS-DOS 形式ファイル) で指定します。複数の RM 関連オブジェクト名を指定する場合は、RM 関連オブジェクト名と RM 関連オブジェクト名との間を空白で区切ります。

7. 運用コマンド

njsmkdll (トランザクション制御用 DLL の作成)

出力メッセージ

メッセージ ID	内容	出力先
KFCA32298-I	ヘルプメッセージ	標準出力

注意事項

このコマンド内では、trnmkobj コマンドが実行されます。trnmkobj コマンドの詳細については、「trnmkobj (トランザクション制御用オブジェクトファイルの作成)」を参照してください。

ntbstart (OpenTP1 の起動)

形式

ntbstart [-n] [-U]

機能

コマンド実行環境の環境変数 DCDIR に対応した OpenTP1 サービスを起動します。

オプション

● -n

前回の終了モードに関係なく、強制的に正常開始します。このオプションの指定を省略すると、前回の終了モードの指定内容から開始モードを選択します。

前回の終了モードが正常終了の場合：正常開始

前回の終了モードが正常終了以外の場合：再開始

OpenTP1 サービスの正常開始の場合、このオプションの指定は無視されます。

● -U

開始モードが再開始の場合に、ユーザサーバの起動を抑止します。

障害が発生し、OpenTP1 の状態を回復するために OpenTP1 をオンラインにして作業する場合など、ユーザサーバを起動する必要がないときに使用してください。

出力メッセージ

メッセージ ID	内容	出力先
KFCA26560-I	OpenTP1 は開始中です	標準出力
KFCA26561-I	OpenTP1 が開始しました	標準出力
KFCA26562-E	OpenTP1 の開始中に障害が発生しました	標準エラー出力
KFCA26563-E	オプションの指定に誤りがあります	標準エラー出力
KFCA26565-E	コマンド引数の指定に誤りがあります	標準エラー出力
KFCA26567-E	メモリの確保に失敗しました	標準エラー出力
KFCA26569-E	OpenTP1 サービスが登録されていません	標準エラー出力
KFCA26570-E	OpenTP1 サービスを起動する権限がありません	標準エラー出力
KFCA26571-E	サービス制御マネージャへの接続に失敗しました	標準エラー出力
KFCA26572-E	OpenTP1 サービスの状態取得に失敗しました	標準エラー出力
KFCA26573-E	OpenTP1 サービスが停止中ではありません	標準エラー出力
KFCA26574-E	OpenTP1 サービスが無効化されています	標準エラー出力

7. 運用コマンド

ntbstart (OpenTP1 の起動)

メッセージ ID	内容	出力先
KFCA26575-E	レジストリへのアクセスに失敗しました	標準エラー出力
KFCA26578-I	ヘルプメッセージ	標準エラー出力

注意事項

このコマンドに `-U` オプションを指定して実行する場合は、次に示す項目に注意してください。

- ・オンライン開始後にユーザサーバを起動したい場合は、`dcsvstart` コマンドを使用してください。
- ・オンライン開始後に、ユーザサーバの構成を回復したい場合には、正常終了および強制正常終了以外でシステムを停止し、`-U` オプションを指定しないでこのコマンドを実行し、オンラインを開始してください。
- ・前回のオンラインで、リモート API 機能を使用したあとのリラン時に `-U` オプションを指定した場合、リモート API の構成までは回復できません。`-U` オプションを指定したあとにリモート API 機能を使用する場合は、次のどちらかの対処をしてください。
 - ・オンライン開始後に `dcsvstart` コマンドで `rap` リスナーを手動で起動する。
 - ・正常終了および強制正常終了以外でシステムを停止してから、`-U` オプションを指定しないでこのコマンドを実行し、オンラインを開始する。
- ・サーバ閉塞引き継ぎ機能を使用すると、次の問題が発生します。
 - ・スケジュールの閉塞状態は、このコマンドに `-U` オプションを指定して実行したりランによるオンラインでは引き継がれません。
 - ・次の条件を満たす場合、このコマンドに `-U` オプションを指定して実行する以前のオンラインのスケジュール閉塞状態が、SPP に引き継がれます。
 - ・スケジュール閉塞状態の SPP が存在する。
 - ・このコマンドに `-U` オプションを指定して実行したりランによるオンラインを、計画停止または強制停止してから、再度このコマンドを使用して起動している。これを防ぐには、次のような対策が必要です。
 - ・このコマンドに `-U` オプションを指定して実行したりランによるオンラインは、正常停止するか強制正常開始する。
 - ・すべてのユーザサービス定義に `hold_recovery=N` を指定する。
 - ・スケジュールサービス定義に `scd_hold_recovery_count=0` を指定する。
- ・系切り替え機能を使用している場合、待機系の OpenTP1 では `-n` オプションの指定は無視されます。

ntbtail (指定ファイル内容のコンソール出力)

形式

```
ntbtail [-i 更新チェック間隔] -t p  
ntbtail [-i 更新チェック間隔] リダイレクトファイル名
```

機能

ファイル名に指定されたファイルに出力された内容を一定間隔で読み取り、コンソールに出力します。指定されたファイルが世代管理ファイルの場合は、ファイル名の末尾に 1, または 2 の世代通番を持つファイルに出力された内容を一定間隔で読み取り、コンソールに出力します。

このコマンドはシステム環境定義の `redirect_file_name` オペランドに指定したファイルにだけ実行できます。`redirect_file_name` オペランドの詳細については、「5.3 システム定義の詳細」の「システム環境定義」を参照してください。

オプション

● **-i 更新チェック間隔** ～ 〈符号なし整数〉 ((1 ～ 60)) 《1》 (単位: 秒)

ファイルの更新内容をチェックする間隔を指定します。

● **-t p**

標準出力リダイレクトファイルに出力された内容を一定間隔で読み取り、コンソールに出力します。このオプションは、`tp1console` コマンドと同じ機能を提供します。

`tp1console` コマンドについては、「`tp1console` (標準出力リダイレクトファイルのコンソール出力)」を参照してください。

コマンド引数

● **リダイレクトファイル名** ～ 〈1 ～ 260 文字のパス名〉

更新されたファイルの内容を出力するファイル名を指定します。`OpenTP1` が出力するファイルのうち、標準出力リダイレクトファイルだけを指定できます。メッセージログファイルは指定できません。

指定するファイルは次に示す条件に従い世代管理されている必要があります。

- ・ ファイル名に拡張子がない。
- ・ 2 世代管理されていてファイル名の末尾に 1 または 2 の世代通番を持つ。
- ・ 世代スワップ時に現用ファイルおよび待機ファイルが削除されない。

このコマンドで出力対象としている世代管理ファイルは、コマンドを終了させるまで削除できません。

指定するファイル名には、ファイル名末尾の世代通番を除いたファイル名称を指定して

7. 運用コマンド

ntbtail (指定ファイル内容のコンソール出力)

ください。例えば、%DCDIR%\¥spool¥prclog1, %DCDIR%\¥spool¥prclog2 を出力対象にする場合は次のとおりに指定してください。

```
ntbtail %DCDIR%\¥spool¥prclog
```

出力メッセージ

メッセージ ID	内容	出力先
KFCA26526-E	コマンドの実行に失敗しました	標準エラー出力
KFCA26527-I	ヘルプメッセージ	標準出力

注意事項

- OpenTP1 のセットアップ（インストール、アンインストール、マルチ OpenTP1 のセットアップ）を実施する場合は、必ずこのコマンドを終了させてください。
- dc_adm_call_command 関数による UAP からのこのコマンドの実行はできません。また、システム環境定義の user_command オペランド、および user_command_online オペランドにこのコマンドは指定できません。
- このコマンドは、引数に指定されたファイルの更新日付を基に、出力するファイル世代を決定しています。引数に指定した現用ファイルのデータを更新する場合は、_commit() 関数や、FlushFileBuffers() 関数を使用するか、ファイルをクローズするかして、データをディスクに書き込み、ファイルの更新日付を最新にしてください。現用ファイルの更新日付が最新になっていない場合、現用ファイルを正確に選択できないため、コマンドが正常に動作しないおそれがあります。
- このコマンドの実行前に指定したファイルに書き込まれている内容については、コンソールへの出力対象としません。このコマンド実行後に書き込まれた内容を順次コンソールに出力します。
- リダイレクトファイル名に指定したファイルが世代管理ファイルの場合に、ファイル世代が切り替わったときは、世代切り替え後の最新世代のファイルを出力対象ファイルとします。
- マルチ OpenTP1 システムでこのコマンドを実行する場合は、dcmakecon コマンドで作成したコンソールのコマンドプロンプトから実行してください。詳細については、「dcmakecon (マルチ OpenTP1 操作コンソールの作成)」を参照してください。
- このコマンドを停止する場合は、コンソールに対し [Ctrl] キーと [C] キーを同時に入力するか、コンソールのクローズ操作を行ってください。

tp1console (標準出力リダイレクトファイルのコンソール出力)

形式

tp1console

機能

一定間隔で標準出力リダイレクトファイルに出力された内容を読み取り、コンソールに出力します。

このコマンドは、システム環境定義の `redirect_file` オペランドに `Y` を指定した場合にだけ実行できます。`redirect_file` オペランドの詳細については、「5.3 システム定義の詳細」の「システム環境定義」を参照してください。また、オペランドの指定内容による標準出力リダイレクトファイルのコンソール出力の詳細は、「2.2.2 標準出力リダイレクトファイルのコンソール出力機能」を参照してください。

出力メッセージ

メッセージ ID	内容	出力先
KFCA26526-E	コマンドの実行に失敗しました	標準エラー出力
KFCA26527-I	ヘルプメッセージ	標準出力

注意事項

- OpenTP1 のセットアップ（インストール、アンインストール、マルチ OpenTP1 のセットアップ）を実施する場合は、必ずコマンドを終了させてください。
- `dc_adm_call_command` 関数による UAP からのこのコマンドの実行はできません。また、システム環境定義の `user_command` オペランド、および `user_command_online` オペランドにこのコマンドは指定できません。
- このコマンドが出力対象としている標準出力リダイレクトファイルは、このコマンドを終了させるまで削除できません。
- このコマンドの実行前に標準出力リダイレクトファイルに書き込まれた内容は、コンソールへの出力対象としません。このコマンドの実行後に書き込まれた内容だけを順次コンソールに出力します。
- 標準出力リダイレクトファイルが世代管理ファイルの場合に、ファイル世代が切り替わったときは、世代切り替え後の最新世代のファイルを出力対象ファイルとします。
- オリジナルの OpenTP1 の場合は、Windows のスタートメニューからこのコマンドを実行できます。デスクトップからの起動方法は次のとおりです。
[スタート] — [プログラム] — [OpenTP1] — [OpenTP1 コンソール]
- マルチ OpenTP1 システムでこのコマンドを実行する場合は、`dcmakecon` コマンドで作成したコンソールのコマンドプロンプトから実行してください。詳細については、「dcmakecon (マルチ OpenTP1 操作コンソールの作成)」を参照してください。

7. 運用コマンド

tp1console (標準出力リダイレクトファイルのコンソール出力)

- このコマンドを停止する場合は、コンソールに対し [Ctrl] キーと [C] キーを同時に入力するか、コンソールのクローズ操作を行ってください。
- このコマンドの実行時に障害が発生した場合、KFCA26526-E メッセージを出力後に、pause 状態になります。リターンキーの入力、または、コンソールのクローズ操作によって、pause 状態を解除してください。
- このコマンドは、バッチファイルとして提供しています。バッチファイルの内容を書き換えないようにしてください。バッチファイルの内容を書き換えて実行した場合の動作は保証できません。

付録

付録 A OpenTP1 のプロセスの一覧

付録 A OpenTP1 のプロセスの一覧

ここでは、Windows 版 OpenTP1 のシステムサービスプロセスを、製品別に説明します。

注

「稼働数」は、一つのノード内で OpenTP1 が一つ稼働する場合の、一つのノード当たりのシステムサービスプロセス数です。

(1) TP1/Server Base

TP1/Server Base のシステムサービスプロセスを、次に示します。

表 A-1 システムサービスプロセス (TP1/Server Base)

実行形式ファイル名	稼働数	サービス	関連するシステム定義	入出力するファイル
admrsvre	0 以上	部分回復	プロセスサービス定義	なし
cltcond	0 以上	CUP 実行サービス	クライアントサービス定義	なし
cltd	0 または 1	クライアント拡張サービス	クライアントサービス定義	なし
clttrnd	0 以上	クライアント実行プロセス	クライアントサービス定義	なし
cpdd	1 以上	チェックポイントダンプサービス	ジャーナルサービス定義 チェックポイントダンプサービス定義	チェックポイントダンプファイル
dcservice	1	Windows サービスプロセス	なし	なし
itvd	1	インタバルタイマサービス	インタバルサービス定義	なし
jnlld	1	ジャーナル管理サービス	ジャーナルサービス定義	なし
jnl iod	1 ～ 16 ※ 1	ジャーナルファイル入出力サービス	なし	システムジャーナルファイル
jnlswd	1	ジャーナルファイル管理サービス	システムジャーナルサービス定義	なし
jnlutld	0 または 1 ※ 2	ジャーナルユーティリティサービス	システムジャーナルサービス定義	システムジャーナルファイル
logd	1	ログサービス	ログサービス定義	メッセージログファイル
namaudtd	0 または 1	ネームサービス	ネームサービス定義	なし
namd	1 ※ 3	ネームサービス	ネームサービス定義	なし

実行形式ファイル名	稼働数	サービス	関連するシステム定義	入出力するファイル
pred	1	プロセスサービス (スーパーユーザプロセス)	プロセスサービス定義	なし
prctee	0 または 1 ※ 4	標準出力リダイレクトサービス	システム環境定義	標準出力リダイレクトファイル
prfiop ※ 5	0, 8, または 9 ※ 6	性能検証用トレース取得サービス	システム共通定義	トレースファイル
qued	0 または 1	キューサービス	メッセージキューサービス定義	メッセージキューファイル
rapclman	0 または 1	リモート API 機能	rap クライアントマネージャサービス定義	なし
raplisnr	0 ～ 1024	リモート API 機能	rap リスナーサービス定義	トレースファイル
rapservr	0 ～ 1024	リモート API 機能	rap リスナーサービス定義	トレースファイル
rtsspp	0 以上	リアルタイム統計情報サービス	リアルタイム取得項目定義	RTS ログファイル
rtssup	0 以上	リアルタイム統計情報サービス	リアルタイム取得項目定義	RTS ログファイル
scdd	1 ※ 3	スケジューラ	スケジューラサービス定義	なし
scdm1td	0 以上	スケジューラ	スケジューラサービス定義	なし
stsd	1	ステータスサービス	ステータスサービス定義	ステータスファイル
tjld	1	トランザクションジャーナルサービス	なし	トランザクションリカバリジャーナルファイル
trnd	1 ※ 3	トランザクション管理サービス	トランザクションサービス定義	なし
trnrmd	1	リソースマネージャ監視サービス	トランザクションサービス定義	なし
trnrvd	1 以上	トランザクション回復サービス	トランザクションサービス定義	なし

注※ 1

この値は、次に示す計算式で算出されます。

 $a \times b$

a: ジャーナルを二重化しているときは 2, 二重化していないときは 1。

b: システムジャーナルファイルの並列アクセス機能での**最大分散数**。

注※ 2

システムジャーナルサービス定義の jnl_auto_unload オペランドに Y を指定した場合、稼働数が 1 になります。

注※ 3

OpenTP1 プロセスの監視を行う場合は、これらのプロセスのどれかを監視対象プロセスとして設定してください。

注※ 4

システム環境定義の redirect_file オペランドに Y を指定した場合、稼働数が 1 になります。

注※ 5

トレース取得用のプロセスです。このプロセスは prcls コマンドでは表示されません。起動有無を確認する場合は、Windows のタスクマネージャのプロセス一覧で確認してください。

注※ 6

システムの稼働数は、次のとおりです。

- ・システム共通定義の prf_trace オペランドに N を指定した場合
稼働数は 0 になります。
- ・システム共通定義の prf_trace オペランドに Y を指定した場合
MCF を使用しないとき、稼働数は 8 になります。
MCF を使用するとき、稼働数は 9 になります。

(2) TP1/FS/Direct Access

TP1/FS/Direct Access のシステムサービスプロセスを、次に示します。

表 A-2 システムサービスプロセス (TP1/FS/Direct Access)

実行形式ファイル名	稼働数	サービス	関連するシステム定義	入出力するファイル
damd	0 または 1	DAM サービス	DAM サービス定義	DAM ファイル

(3) TP1/FS/Table Access

TP1/FS/Table Access のシステムサービスプロセスを、次に示します。

表 A-3 システムサービスプロセス (TP1/FS/Table Access)

実行形式ファイル名	稼働数	サービス	関連するシステム定義	入出力するファイル
tamd	0 または 1	TAM サービス	TAM サービス定義	TAM ファイル
tamiod	0 または 1	TAM ファイル入出力サービス	TAM サービス定義	TAM ファイル

(4) TP1/Message Control

TP1/Message Control のシステムサービスプロセスを、次に示します。

表 A-4 システムサービスプロセス (TP1/Message Control)

実行形式ファイル名	稼働数	サービス	関連するシステム定義	入出力するファイル
mapsmgrd	0 または 1	マッピングサービス	マッピングサービス定義 マッピングサービス属性定義	なし
mcfmngprd	0 または 1	MCF マネジャ	MCF マネジャ定義	トレースファイル
ユーザ指定※	0 以上	MCF 通信サービス	MCF 通信構成定義 MCF アプリケーション定義	なし

注※

TP1/Messaging を使用した場合は、mcfutcpd および mcfupsvd で固定となります。

(5) TP1/Message Queue

TP1/Message Queue のシステムサービスプロセスを、次に示します。

表 A-5 システムサービスプロセス (TP1/Message Queue)

実行形式ファイル名	稼働数	サービス	関連するシステム定義	入出力するファイル
mqaacmd	0 または 1	メッセージキューイング (TP1/Message Queue) オンラインコマンドサービス	MQA サービス定義	MQA キューファイル
mquad	0 または 1	メッセージキューイング (TP1/Message Queue) MQA サービス	MQA サービス定義	MQA キューファイル MQ 管理情報ファイル
mqaiod	0 または 1 以上	メッセージキューイング (TP1/Message Queue) キューファイル入出力サービス	MQA サービス定義	MQA キューファイル
mqamnd	0 または 1	メッセージキューイング (TP1/Message Queue) 満了メッセージ監視サービス	MQA サービス定義	なし
mqcdtcp	0 または 1	メッセージキューイング (TP1/Message Queue) MQC サービス	MQA サービス定義 MQC サービス定義	トレースファイル

実行形式ファイル名	稼働数	サービス	関連するシステム定義	入出力するファイル
mqcgrp	0 または 1 以上	メッセージキューイング (TP1/Message Queue) MQC ゲートウェイサーバ	ユーザサービス定義	MQA キューファイル トレースファイル
mqrsp	0 または 1 以上	メッセージキューイング (TP1/Message Queue) リポジトリ管理サーバ SPP	ユーザサービス定義 (%DCCONFPATH%\mqrsp)	MQA キューファイル
mqrsp	0 または 1	メッセージキューイング (TP1/Message Queue) リポジトリ管理サーバ SUP	ユーザサービス定義 (%DCCONFPATH%\mqrsp)	MQA キューファイル トレースファイル
mqtctcp	0 または 1 以上	メッセージキューイング (TP1/Message Queue) MQT 通信サービス	MQT 通信構成定義 MQT サービス定義	MQA キューファイル チャネル管理情報格納ファイル トレースファイル ユーザ定義ファイル MQ 管理情報ファイル
mqtmgd	0 または 1	メッセージキューイング (TP1/Message Queue) MQT 通信マネージャサービス	MQA サービス定義	チャネル管理情報格納ファイル

索引

A

adm_message_option [ユーザサービス定義] 127
adm_message_option [ユーザサービスデフォルト定義] 122
all_node [システム共通定義] 89
all_node_ex [システム共通定義] 90
all_node_ex_extend_number [システム共通定義] 91
all_node_extend_number [システム共通定義] 91
assurance_count [チェックポイントダンプサービス定義] 100
atomic_update [ユーザサービス定義] 125
atomic_update [ユーザサービスデフォルト定義] 120
auto_restart [ユーザサービス定義] 126
auto_restart [ユーザサービスデフォルト定義] 121

B

balance_count [クライアントサービス定義] 107
balance_count [ユーザサービス定義] 126
balance_count [ユーザサービスデフォルト定義] 121

C

cancel_normal_terminate [ユーザサービス定義] 127
cancel_normal_terminate [ユーザサービスデフォルト定義] 121
client_uid_check [システム共通定義] 90
clt_conf [システムサービス構成定義] 88
clt_cup_conf [クライアントサービス定義] 107
clt_inquire_time [クライアントサービス定義] 107
clt_port [クライアントサービス定義] 107

clt_trn_conf [クライアントサービス定義] 107
cltcon_port [クライアントサービス定義] 107
clttrn_port [クライアントサービス定義] 107
COBOL2002 開発マネージャの設定 39
COBOL2002 での UAP の作成 36
console_output [システム環境定義] 87
core_shm_suppress [ユーザサービス定義] 128
core_shm_suppress [ユーザサービスデフォルト定義] 123
core_suppress_watch_time [システム共通定義] 90
coredump_type [システム共通定義] 91
cpd_message_id_change_level [チェックポイントダンプサービス定義] 101
critical [ユーザサービス定義] 126
critical [ユーザサービスデフォルト定義] 121
cup_balance_count [クライアントサービス定義] 107
cup_parallel_count [クライアントサービス定義] 107
C 言語での UAP の作成 31
C ランタイムライブラリのインストール 31

D

dam_added_file [DAM サービス定義] 105
dam_cache_attribute [DAM サービス定義] 105
dam_cache_reuse_from [DAM サービス定義] 105
dam_cache_size [DAM サービス定義] 105
dam_cache_size_fix [DAM サービス定義] 105
dam_conf [システムサービス構成定義] 88
dam_default_cache_num [DAM サービス定義] 105

- dam_ex_refer_read [DAM サービス定義] 106
- dam_io_error_occur [DAM サービス定義] 105
- dam_io_interval [DAM サービス定義] 105
- dam_kb_size [DAM サービス定義] 106
- dam_max_block_size [DAM サービス定義] 106
- dam_message_level [DAM サービス定義] 105
- dam_tran_process_count [DAM サービス定義] 105
- dam_transaction_access [DAM サービス定義] 105
- dam_update_block [DAM サービス定義] 105
- dam_update_block_over [DAM サービス定義] 105
- damadd [論理ファイルの追加] 205
- dambkup [物理ファイルのバックアップ] 205
- damcache [DAM サービス定義] 106
- damchdef [キャッシュブロック数のしきい値の設定] 205
- damchinf [キャッシュブロック数の取得] 205
- damchlmnt [DAM サービス定義] 106
- damdel [物理ファイルの削除] 205
- damfile [DAM サービス定義] 106
- damfrc [論理ファイルの回復] 205
- damhold [論理ファイルの論理閉塞] 205
- damload [物理ファイルの初期設定] 204
- damls [論理ファイルの状態表示] 204
- damrles [論理ファイルの閉塞解除] 205
- damrm [論理ファイルの切り離し] 205
- damrstr [物理ファイルのリストア] 205
- DCADMDEBUG [システム環境定義] 87
- dcalzprf [性能検証用トレース情報解析] 208
- dcauditsetup [監査ログ機能の環境設定] 201
- dcbindht [システム共通定義] 92
- DCCONFPATH [システム環境定義] [dcputenv] 88
- DCCONFPATH [システム環境定義] [putenv] 87
- dccspool [トラブルシュート情報の削除] 197
- dcdefchk [システム定義のチェック] 197
- dccls [マルチ OpenTP1 のセットアップ状態の確認] 211
- DCFPL_CONNECT_RETRY_COUNT [ユーザサービス定義] 130
- DCFPL_CONNECT_RETRY_INTERVAL [ユーザサービス定義] 130
- dcjchconf [システム定義のオペランドの指定] 196
- dcjcmdex [シナリオテンプレートからの OpenTP1 コマンドの実行] 196
- dcjnamch [ドメイン定義ファイルの更新] 196
- dcmakecon [マルチ OpenTP1 操作コンソールの作成] 211
- dcmakeup [OpenTP1 の内部制御用資源の確保と解放] 196
- dcmaphcg [マップファイルのパス名変更] 210
- dcmapls [マップファイルのロード済み資源の表示] 210
- dcmarea [マルチノード構成定義] 103
- DCMCFQUEBAK [システムサービス共通定義] 132
- dcmstart [マルチノードエリア, サブエリアの開始] 196
- dcmstart_watch_time [マルチノード構成定義] 103
- dcmstop [マルチノードエリア, サブエリアの終了] 196
- dcmstop_watch_time [マルチノード構成定義] 103
- dcndls [OpenTP1 ノードの状態表示] 196
- dcplist [製品情報の表示] 197
- dcpreport [マルチノード物理定義] 104
- dcrasget [保守資料の取得] 197

dcreport [システム統計情報の標準出力へのリアルタイム編集出力] 197

dcreset [プロセスサービスの再起動および定義の反映] 196

dcsetup [OpenTP1 の OS への登録と削除] 196

dcsetupml [マルチ OpenTP1 のセットアップとアンセットアップ] 211

dcshmls [共用メモリの状態表示] 196

dcstart [OpenTP1 の開始] 196

dcstart_wakeup_retry_count [システム共通定義] 91

dcstart_wakeup_retry_interval [システム共通定義] 91

dcstats [システム統計情報の取得開始, 終了] 196

dcstop [OpenTP1 の終了] 196

dcsvgdef [ユーザサービスネットワーク定義] 111

dcsvstart [サーバの開始] 198

dcsvstart [システムサービス構成定義] 88

dcsvstart [ユーザサービス構成定義] 89

dcsvstop [サーバの終了] 198

DCSYSLOGOUT [ログサービス定義] 103

DCUAPCONFPATH [システム環境定義]
[dcpulenv] 88

DCUAPCONFPATH [システム環境定義]
[putenv] 88

deadlock_priority [ユーザサービス定義] 125

deadlock_priority [ユーザサービスデフォルト定義] 120

domain_masters_addr [システム共通定義] 90

domain_masters_port [システム共通定義] 90

domain_use_dns [システム共通定義] 90

dynamic_shmpool_size [システム環境定義] 87

F

fil_prf_trace_delay_time [システム共通定義] 91

fil_prf_trace_option [システム共通定義] 91

filbkup [OpenTP1 ファイルシステムのバックアップ] 201

filchgrp [OpenTP1 ファイルグループの変更] 202

filchmod [OpenTP1 ファイルのアクセス許可モードの変更] 202

filchown [OpenTP1 ファイル所有者の変更] 202

fills [OpenTP1 ファイルシステムの内容表示] 201

filmkfs [OpenTP1 ファイルシステムの初期設定] 201

filrstr [OpenTP1 ファイルシステムのリスト] 201

filstatfs [OpenTP1 ファイルシステムの状態表示] 201

G

groups [トランザクションサービス定義] 97

groups [ユーザサービス定義] 128

groups [ユーザサービスデフォルト定義] 123

GUI による OpenTP1 の開始 159

GUI による OpenTP1 の起動と停止 187

GUI による OpenTP1 の停止 162

GUI による環境設定 178

H

ha_conf [システムサービス構成定義] 88

HiRDB との連携 61

hold [ユーザサービス定義] 125

hold [ユーザサービスデフォルト定義] 120

hold_recovery [ユーザサービス定義] 125

hold_recovery [ユーザサービスデフォルト定義] 120

I

ipc_backlog_count [システム共通定義] 91

ipc_backlog_count [ユーザサービス定義] 128

- ipc_backlog_count [ユーザサービスデフォルト定義] 123
- ipc_conn_interval [システム共通定義] 90
- ipc_conn_interval [ユーザサービス定義] 128
- ipc_conn_interval [ユーザサービスデフォルト定義] 123
- ipc_header_rcv_time [システム共通定義] 90
- ipc_header_rcv_time [ユーザサービス定義] 128
- ipc_header_rcv_time [ユーザサービスデフォルト定義] 123
- ipc_listen_sockbufset [システム共通定義] 91
- ipc_listen_sockbufset [ユーザサービス定義] 128
- ipc_listen_sockbufset [ユーザサービスデフォルト定義] 123
- ipc_rcvbuf_size [システム共通定義] 90
- ipc_rcvbuf_size [ユーザサービス定義] 128
- ipc_rcvbuf_size [ユーザサービスデフォルト定義] 123
- ipc_send_count [システム共通定義] 90
- ipc_send_count [ユーザサービス定義] 128
- ipc_send_count [ユーザサービスデフォルト定義] 123
- ipc_send_interval [システム共通定義] 90
- ipc_send_interval [ユーザサービス定義] 128
- ipc_send_interval [ユーザサービスデフォルト定義] 123
- ipc_sendbuf_size [システム共通定義] 90
- ipc_sendbuf_size [ユーザサービス定義] 128
- ipc_sendbuf_size [ユーザサービスデフォルト定義] 123
- ipc_sockctl_highwater [rap リスナーサービス定義] 113
- ipc_sockctl_highwater [システム共通定義] 90
- ipc_sockctl_watchtime [ユーザサービス定義] 128
- ipc_sockctl_watchtime [ユーザサービスデフォルト定義] 123
- ipc_tcpnodelay [スケジューラサービス定義] 95
- ipc_tcpnodelay [ユーザサービス定義] 129
- ipc_tcpnodelay [ユーザサービスデフォルト定義] 123
- ist_conf [システムサービス構成定義] 88
- ist_node [IST サービス定義] 108
- ist_node_group [IST サービス定義] 108
- istdef [IST サービス定義] 108

J

- jar_conf [システムサービス構成定義] 88
- jnl_arc_buff_size [システムジャーナルサービス定義] 100
- jnl_arc_check_level [システムジャーナルサービス定義] 100
- jnl_arc_ipc_buff_size [グローバルアーカイブジャーナルサービス定義] 104
- jnl_arc_ipc_buff_size [ジャーナルサービス定義] 99
- jnl_arc_max_datasize [アーカイブジャーナルサービス定義] 105
- jnl_arc_max_datasize [システムジャーナルサービス定義] 100
- jnl_arc_name [システムジャーナルサービス定義] 100
- jnl_arc_rec_kind [システムジャーナルサービス定義] 100
- jnl_arc_terminate_check [システムジャーナルサービス定義] 100
- jnl_arc_terminate_timeout [グローバルアーカイブジャーナルサービス定義] 104

- jnl_arc_terminate_timeout [ジャーナルサービス定義] 99
- jnl_arc_trn_stat [システムジャーナルサービス定義] 100
- jnl_arc_uj_code [システムジャーナルサービス定義] 100
- jnl_auto_unload [システムジャーナルサービス定義] 100
- jnl_auto_unload_path [システムジャーナルサービス定義] 100
- jnl_cdinterval [システムジャーナルサービス定義] 99
- jnl_cdskip_limit [チェックポイントダンプサービス定義] 101
- jnl_cdskip_msg [チェックポイントダンプサービス定義] 101
- jnl_dual [アーカイブジャーナルサービス定義] 104
- jnl_dual [システムジャーナルサービス定義] 100
- jnl_dual [チェックポイントダンプサービス定義] 101
- jnl_fileless_option [システム共通定義] 91
- jnl_max_datasize [システムジャーナルサービス定義] 99
- jnl_max_datasize [チェックポイントダンプサービス定義] 100
- jnl_max_file_dispersion [アーカイブジャーナルサービス定義] 104
- jnl_max_file_dispersion [システムジャーナルサービス定義] 100
- jnl_message_id_change_level [ジャーナルサービス定義] 99
- jnl_min_file_dispersion [アーカイブジャーナルサービス定義] 105
- jnl_min_file_dispersion [システムジャーナルサービス定義] 100
- jnl_objservername [チェックポイントダンプサービス定義] 100
- jnl_prf_event_trace_level [システム共通定義] 91
- jnl_reduced_mode [チェックポイントダンプサービス定義] 101
- jnl_rerun_reserved_file_open [システムジャーナルサービス定義] 100
- jnl_rerun_swap [アーカイブジャーナルサービス定義] 104
- jnl_rerun_swap [システムジャーナルサービス定義] 99
- jnl_reserved_file_auto_open [チェックポイントダンプサービス定義] 101
- jnl_singleoperation [アーカイブジャーナルサービス定義] 104
- jnl_singleoperation [システムジャーナルサービス定義] 100
- jnl_singleoperation [チェックポイントダンプサービス定義] 101
- jnl_tran_optimum_level [ジャーナルサービス定義] 99
- jnl_unload_check [アーカイブジャーナルサービス定義] 105
- jnl_unload_check [システムジャーナルサービス定義] 100
- jnl_watch_time [ジャーナルサービス定義] 99
- jnladdfg [アーカイブジャーナルサービス定義] 105
- jnladdfg [システムジャーナルサービス定義] 100
- jnladdfg [チェックポイントダンプサービス定義] 101
- jnladdpf [アーカイブジャーナルサービス定義] 105
- jnladdpf [システムジャーナルサービス定義] 100
- jnladdpf [ジャーナル関係の物理ファイルの割り当て] 203
- jnladdpf [チェックポイントダンプサービス定義] 101
- jnlardis [リソースグループの接続の強制解除] 204
- jnlarls [アーカイブ状態の表示] 203
- jnlautunl [自動アンロード機能の制御] 203
- jnlchgfg [ジャーナル関係のファイルのステータス変更] 203

- jnlclsfg [ジャーナル関係のファイルのク
ローズ] 202
 - jnlcolc [ファイル回復用ジャーナルの集積]
203
 - jnlcopy [アンロードジャーナルファイルの
複写] 203
 - jnlldelpf [ジャーナル関係の物理ファイルの
削除] 203
 - jnldfs [グローバルアーカイブジャーナル
サービス定義] 104
 - jnldfs [ジャーナルサービス定義] 99
 - jnledit [アンロードジャーナルファイル, ま
たはグローバルアーカイブアンロード
ジャーナルファイルの編集出力] 204
 - jnlinit [ジャーナル関係のファイルの初期設
定] 202
 - jnlis [ジャーナル関係のファイル情報の表
示] 202
 - jnlmst [MCF 稼働統計情報の出力] 204
 - jnlmkrf [ジャーナル関係のファイルの回復]
203
 - jnlpnfg [ジャーナル関係のファイルのオー
プン] 202
 - jnlrinf [再開始中読み込み済みジャーナル関
係のファイル情報の表示] 202
 - jnlrm [ジャーナル関係のファイルの削除]
203
 - jnlrput [アンロードジャーナルファイル, ま
たはグローバルアーカイブアンロード
ジャーナルファイルのレコード出力] 204
 - jnlrsort [アンロードジャーナルファイル, お
よびグローバルアーカイブアンロード
ジャーナルファイルの時系列ソート, およ
びマージ] 204
 - jnlstts [稼働統計情報の出力] 204
 - jnlswpfg [ジャーナル関係のファイルのス
ワップ] 203
 - jnlunlfg [ジャーナル関係のファイルのアン
ロード] 203
 - jp1_use [システム共通定義] 89
- L**
-
- LANG [システム共通定義] 92
 - lck_deadlock_info [ロックサービス定義]
92
 - lck_deadlock_info_remove [ロックサービス
定義] 92
 - lck_deadlock_info_remove_level [ロック
サービス定義] 92
 - lck_limit_fordam [ロックサービス定義] 92
 - lck_limit_formqa [ロックサービス定義] 92
 - lck_limit_fortam [ロックサービス定義] 92
 - lck_limit_foruser [ロックサービス定義] 92
 - lck_prf_trace_level [ロックサービス定義]
92
 - lck_release_detect [ロックサービス定義]
92
 - lck_release_detect_interval [ロックサービ
ス定義] 92
 - lck_wait_priority [ユーザサービス定義]
126
 - lck_wait_priority [ユーザサービスデフォル
ト定義] 121
 - lck_wait_timeout [ロックサービス定義] 92
 - lckls [排他情報の表示] 200
 - lckpool [排他制御用テーブルのプール情報
の表示] 200
 - lckrminf [デッドロック情報ファイルとタイ
ムアウト情報ファイルの削除] 200
 - leveldown_queue_count [ユーザサービス定
義] 128
 - leveldown_queue_count [ユーザサービスデ
フォルト定義] 122
 - levelup_queue_count [ユーザサービス定義]
128
 - levelup_queue_count [ユーザサービスデ
フォルト定義] 122
 - loadcheck_interval [ユーザサービス定義]
128
 - loadcheck_interval [ユーザサービスデフォ
ルト定義] 122
 - loadlevel_message [ユーザサービス定義]
128
 - loadlevel_message [ユーザサービスデフォ
ルト定義] 123
 - log_audit_count [ログサービス定義] 103

- log_audit_message [rap クライアントマネ
ジャサービス定義] 113
 - log_audit_message [rap リスナーサービ
ス定義] 113
 - log_audit_message [ユーザサービス定義]
129
 - log_audit_message [ユーザサービスデフォ
ルト定義] 124
 - log_audit_message [ログサービス定義]
103
 - log_audit_out [ログサービス定義] 103
 - log_audit_out_suppress [rap クライアント
マネジャサービス定義] 113
 - log_audit_out_suppress [rap リスナーサー
ビス定義] 113
 - log_audit_out_suppress [ユーザサービス定
義] 129
 - log_audit_out_suppress [ユーザサービスデ
フォルト定義] 124
 - log_audit_path [ログサービス定義] 103
 - log_audit_size [ログサービス定義] 103
 - log_filesize [ログサービス定義] 101
 - log_jerr_rint [ログサービス定義] 102
 - log_jp1_allno [ログサービス定義] 102
 - log_jp1_date [ログサービス定義] 102
 - log_jp1_hostname [ログサービス定義] 102
 - log_jp1_pgmid [ログサービス定義] 102
 - log_jp1_prcid [ログサービス定義] 102
 - log_jp1_prcno [ログサービス定義] 102
 - log_jp1_sysid [ログサービス定義] 102
 - log_jp1_time [ログサービス定義] 102
 - log_msg_allno [ログサービス定義] 101
 - log_msg_console [ログサービス定義] 101
 - log_msg_date [ログサービス定義] 101
 - log_msg_hostname [ログサービス定義]
101
 - log_msg_pgmid [ログサービス定義] 101
 - log_msg_prcid [ログサービス定義] 101
 - log_msg_prcno [ログサービス定義] 101
 - log_msg_sysid [ログサービス定義] 101
 - log_msg_time [ログサービス定義] 101
 - log_netm_allno [ログサービス定義] 102
 - log_netm_date [ログサービス定義] 102
 - log_netm_hostname [ログサービス定義]
102
 - log_netm_out [ログサービス定義] 101
 - log_netm_pgmid [ログサービス定義] 102
 - log_netm_prcid [ログサービス定義] 102
 - log_netm_prcno [ログサービス定義] 102
 - log_netm_sysid [ログサービス定義] 102
 - log_netm_time [ログサービス定義] 102
 - log_notify_allno [ログサービス定義] 102
 - log_notify_date [ログサービス定義] 102
 - log_notify_hostname [ログサービス定義]
102
 - log_notify_out [ログサービス定義] 102
 - log_notify_pgmid [ログサービス定義] 102
 - log_notify_prcid [ログサービス定義] 102
 - log_notify_prcno [ログサービス定義] 102
 - log_notify_sysid [ログサービス定義] 102
 - log_notify_time [ログサービス定義] 102
 - log_syslog_allno [ログサービス定義] 102
 - log_syslog_append_nodeid [ログサービス定
義] 103
 - log_syslog_date [ログサービス定義] 102
 - log_syslog_elist [ログサービス定義] 103
 - log_syslog_elist_rint [ログサービス定義]
103
 - log_syslog_hostname [ログサービス定義]
103
 - log_syslog_out [ログサービス定義] 102
 - log_syslog_pgmid [ログサービス定義] 103
 - log_syslog_prcid [ログサービス定義] 102
 - log_syslog_prcno [ログサービス定義] 102
 - log_syslog_synchro [ログサービス定義]
103
 - log_syslog_sysid [ログサービス定義] 102
 - log_syslog_time [ログサービス定義] 103
 - logcat [メッセージログファイルの内容表
示] 201
 - logcon [メッセージログのリアルタイム出力
機能の切り替え] 201
- ## M
-
- make_queue_on_starting [ユーザサービス
定義] 128

- make_queue_on_starting [ユーザサービスデフォルト定義] 122
- max_open_fds [XATMI 通信サービス定義] 110
- max_open_fds [システムサービス共通情報定義] 132
- max_open_fds [ユーザサービス定義] 127
- max_open_fds [ユーザサービスデフォルト定義] 122
- max_socket_descriptors [rap リスナーサービス定義] 112
- max_socket_descriptors [XATMI 通信サービス定義] 110
- max_socket_descriptors [グローバルアーカイブジャーナルサービス定義] 104
- max_socket_descriptors [システム共通定義] 89
- max_socket_descriptors [システムサービス共通情報定義] 132
- max_socket_descriptors [ジャーナルサービス定義] 99
- max_socket_descriptors [スケジュールサービス定義] 95
- max_socket_descriptors [トランザクションサービス定義] 96
- max_socket_descriptors [ネームサービス定義] 93
- max_socket_descriptors [プロセスサービス定義] 94
- max_socket_descriptors [ユーザサービス定義] 127
- max_socket_descriptors [ユーザサービスデフォルト定義] 122
- max_socket_msg [ユーザサービス定義] 126
- max_socket_msg [ユーザサービスデフォルト定義] 121
- max_socket_msglen [ユーザサービス定義] 126
- max_socket_msglen [ユーザサービスデフォルト定義] 121
- mcf_jnl_buff_size [ユーザサービス定義] 126
- mcf_jnl_buff_size [ユーザサービスデフォルト定義] 120
- mcf_mgrid [ユーザサービス定義] 126
- mcf_mgrid [ユーザサービスデフォルト定義] 121
- mcf_prf_trace [システムサービス情報定義] 132
- mcf_prf_trace [ユーザサービス定義] 129
- mcf_prf_trace [ユーザサービスデフォルト定義] 124
- mcf_prf_trace_level [システムサービス共通情報定義] 132
- mcf_psv_id [ユーザサービス定義] 126
- mcf_psv_id [ユーザサービスデフォルト定義] 121
- mcf_service_max_count [ユーザサービス定義] 126
- mcf_service_max_count [ユーザサービスデフォルト定義] 121
- mcf_spp_oj [ユーザサービス定義] 127
- mcf_spp_oj [ユーザサービスデフォルト定義] 122
- mcfactap [アプリケーションの閉塞解除] 208
- mcfalcap [MCF アプリケーション定義] 131
- mcfadlcap [アプリケーション異常終了回数
の初期化] 208
- mcfadctap [アプリケーションの閉塞] 208
- mcfadltap [アプリケーションに関するタイ
マ起動要求の削除] 208
- mcfacnv [MCF アプリケーション定義] 131
- mcfalsap [アプリケーションの状態表示] 208
- mcfalstap [アプリケーションに関するタイ
マ起動要求の表示] 208
- mcfmcname [MCF マネージャ定義] 130
- mcfmcomn [MCF マネージャ定義] 130
- mcfmenv [MCF マネージャ定義] 130
- mcfmexp [MCF マネージャ定義] 130
- mcfmqgid [MCF マネージャ定義] 130
- mcfmsmsg [MCF マネージャ定義] 130
- mcfmsts [MCF マネージャ定義] 130

- mcfmsvg [MCF マネジャ定義] 130
- mcfmuap [MCF マネジャ定義] 130
- mcfreport [MCF 稼働統計情報の編集] 210
- mcfstats [MCF 稼働統計情報の出力] 210
- mcftactcn [コネクションの確立] 208
- mcftactle [論理端末の閉塞解除] 209
- mcftactmj [論理端末に関するメッセージ
ジャーナルの取得開始] 209
- mcftactsg [サービスグループの閉塞解除]
209
- mcftactss [セッションの開始] 210
- mcftactsv [サービスの閉塞解除] 210
- mcftalcle [MCF 通信構成定義] 131
- mcftbuf [MCF 通信構成定義] 131
- mcftchcn [コネクションの切り替え] 208
- mcftcomn [MCF 通信構成定義] 131
- mcftdctcn [コネクションの解放] 208
- mcftdctle [論理端末の閉塞] 209
- mcftdctmj [論理端末に関するメッセージ
ジャーナルの取得終了] 209
- mcftdctsg [サービスグループの閉塞] 209
- mcftdctss [セッションの終了] 210
- mcftdctsv [サービスの閉塞] 210
- mcftdlql [論理端末の出力キュー削除] 209
- mcftdlqsg [サービスグループの入力キュー
削除] 210
- mcftdmpqu [入出力キューの内容複写] 210
- mcftedalt [代行送信の終了] 209
- mcftendct [論理端末に対する継続問い合わ
せ応答処理の強制終了] 209
- mcftenv [MCF 通信構成定義] 131
- mcfthldiq [サービスグループの入力キュー
処理の保留] 209
- mcfthldoq [論理端末の出力キュー処理の保
留] 209
- mcftlsbuf [バッファグループの使用状況表
示] 210
- mcftlscn [コネクションの状態表示] 208
- mcftlscom [MCF 通信サービスの状態参照]
211
- mcftlsle [論理端末の状態表示] 209
- mcftlsln [ネットワークの状態表示] 208
- mcftlssg [サービスグループの状態表示]
209
- mcftlssv [サービスの状態表示] 210
- mcftlstnd [メッセージ多重処理状況の表示]
208
- mcftlsutm [ユーザタイム監視の状態表示]
211
- mcftofln [サーバ型コネクションの確立要求
の受付終了] 208
- mcftonln [サーバ型コネクションの確立要求
の受付開始] 208
- mcftped [MCF 通信構成定義] 131
- mcftpsvr [MCF 通信構成定義] 131
- mcftrlsiq [サービスグループの入力キュー処
理の保留解除] 210
- mcftrlsoq [論理端末の出力キュー処理の保
留解除] 209
- mcftspqle [論理端末のメッセージキューの
先頭スキップ] 209
- mcftstalt [代行送信の開始] 209
- mcftstart [MCF 通信サービスの部分開始]
211
- mcftstop [MCF 通信サービスの部分停止]
210
- mcftstptr [MCF トレース取得の終了] 210
- mcftstrtr [MCF トレース取得の開始] 210
- mcftsts [MCF 通信構成定義] 131
- mcftswptr [MCF トレースファイルの強制
スワップ] 210
- mcfttim [MCF 通信構成定義] 131
- mcfttre [MCF 通信構成定義] 131
- mcfttred [MCF 通信構成定義] 131
- mcfuevt [アプリケーションプログラムの起
動] 209
- MCF 稼働統計情報の出力 [jnlmcst] 204
- MCF 稼働統計情報の出力 [mcfstats] 210
- MCF 稼働統計情報の編集 [mcfreport] 210
- MCF 通信サービスの状態参照 [mcftlscom]
211
- MCF 通信サービスの部分開始 [mcftstart]
211
- MCF 通信サービスの部分停止 [mcftstop]
210

MCF トレース取得の開始 [mcftstrtr] 210
 MCF トレース取得の終了 [mcftstptr] 210
 MCF トレースファイルの強制スワップ
 [mcftswptr] 210
 message_bufflen [ユーザサービス定義] 125
 message_bufflen [ユーザサービスデフォ
 ルト定義] 120
 message_cell_size [ユーザサービス定義]
 126
 message_cell_size [ユーザサービスデフォ
 ルト定義] 121
 message_store_bufflen [クライアントサービ
 ス定義] 108
 message_store_bufflen [ユーザサービス定
 義] 125
 message_store_bufflen [ユーザサービスデ
 フォルト定義] 120
 mode_conf [システム環境定義] 87
 module [システムサービス情報定義] 132
 module [ユーザサービス定義] 125
 mqa_conf [システムサービス構成定義] 88
 multi_node_option [システム共通定義] 89
 multi_schedule [ユーザサービス定義] 128
 multi_schedule [ユーザサービスデフォルト
 定義] 122
 my_host [システム共通定義] 89

N

nam_prf_trace_level [システム共通定義]
 91
 namalivechk [OpenTP1 起動確認, キャッ
 シュ削除] 200
 namblad [RPC 抑止リストの操作] 201
 namchgfl [ドメイン構成の変更 (ドメイン
 定義ファイル使用)] 200
 namdomainsetup [ドメイン代表スケジュー
 ルサービスの登録, 削除] 200
 name_audit_conf [ネームサービス定義] 93
 name_audit_interval [ネームサービス定義]
 93
 name_audit_watch_time [ネームサービス
 定義] 93
 name_cache_size [ネームサービス定義] 93

name_cache_validity_time [ネームサービス
 定義] 93
 name_domain_file_use [システム共通定義]
 91
 name_global_lookup [ネームサービス定義]
 93
 name_manager_node [システム共通定義]
 92
 name_node_add_policy [システム共通定義]
 92
 name_nodeid_check_message [ネームサー
 ビス定義] 93
 name_notify [システム共通定義] 90
 name_port [システム共通定義] 89
 name_remove_down_node [システム共通定
 義] 92
 name_rpc_control_list [ネームサービス定
 義] 93
 name_service_extend [ネームサービス定
 義] 93
 name_service_mode [システム共通定義]
 92
 name_start_error [ネームサービス定義] 94
 name_start_retry_count [ネームサービス定
 義] 93
 name_start_retry_interval [ネームサービ
 ス定義] 94
 name_start_watch_time [ネームサービス定
 義] 93
 name_sync_ready_time [ネームサービス定
 義] 94
 name_total_size [ネームサービス定義] 93
 nammstr [マネジャノードの変更] 201
 namndchg [ドメイン構成の変更 (システム
 共通定義使用)] 200
 namndopt [ノードのオプション情報の変更]
 201
 namndrm [ノードリスト情報の削除] 201
 namnlcre [ノードリストファイルの作成]
 201
 namnlldel [ノードリストファイルの削除]
 201

namnldsp [ノードリストファイルの内容表示] 201
 namnlfil [ネームサービス定義] 94
 namsvinf [OpenTP1 のサーバ情報の表示] 200
 namunavl [起動通知情報の強制的無効化] 200
 nice [rap リスナーサービス定義] 111
 nice [ユーザサービス定義] 125
 nice [ユーザサービスデフォルト定義] 120
 njsmkdll [トランザクション制御用 DLL の作成] 211
 nl_watch_time [グローバルアーカイブジャーナルサービス定義] 104
 node_down_restart [ユーザサービス定義] 126
 node_down_restart [ユーザサービスデフォルト定義] 121
 node_id [システム共通定義] 89
 ntbstart [OpenTP1 の起動] 211
 ntbtail [指定ファイル内容のコンソール出力] 211

O

open_rm [クライアントサービス定義] 107
 OpenTP1_Deploy 170
 OpenTP1_ScenarioScaleout 171
 OpenTP1_Undeploy 170
 [OpenTP1 環境設定] ウィンドウ 179
 [OpenTP1 環境設定] ウィンドウ (格納先情報) 184
 [OpenTP1 環境設定] ウィンドウ (パラメータ情報) 182
 OpenTP1 管理者の登録 50
 [OpenTP1 起動/停止] ウィンドウ 189
 OpenTP1 起動確認, キャッシュ削除 [namalivechk] 200
 OpenTP1 コンソール出力機能 14
 OpenTP1 の OS への登録と削除 [dcsetup] 196
 OpenTP1 ノードの状態表示 [dcndls] 196
 OpenTP1 の開始 157
 OpenTP1 の開始 [dcstart] 196

OpenTP1 の起動 [ntbstart] 211
 OpenTP1 のサーバ情報の表示 [namsvinf] 200
 OpenTP1 のシステム定義情報の作成 51
 OpenTP1 の終了 [dcstop] 196
 OpenTP1 の停止 161
 OpenTP1 の停止時の注意事項 162
 OpenTP1 の内部制御用資源の確保と解放 [dcmakeup] 196
 OpenTP1 のプロセスの一覧 246
 OpenTP1 のプロセスの強制停止 [prckill] 199
 OpenTP1 ファイルグループの変更 [filchgrp] 202
 OpenTP1 ファイルシステム 55
 OpenTP1 ファイルシステムの構築 55
 OpenTP1 ファイルシステムの状態表示 [filstatfs] 201
 OpenTP1 ファイルシステムの初期設定 [filmkfs] 201
 OpenTP1 ファイルシステムの内容表示 [fills] 201
 OpenTP1 ファイルシステムのバックアップ [filbkup] 201
 OpenTP1 ファイルシステムのリストア [filrstr] 201
 OpenTP1 ファイル所有者の変更 [filchown] 202
 OpenTP1 ファイルのアクセス許可モードの変更 [filchmod] 202
 Oracle との連携 66
 OSI TP 通信の未決着トランザクション情報の表示 [tptrnls] 199
 OS の設定 80

P

parallel_count [クライアントサービス定義] 107
 parallel_count [ユーザサービス定義] 125
 parallel_count [ユーザサービスデフォルト定義] 120
 polling_control_data [トランザクションサービス定義] 97

- polling_control_data [ユーザサービス定義]
128
- polling_control_data [ユーザサービスデ
フォルト定義] 123
- prc_abort_signal [ユーザサービス定義]
127
- prc_abort_signal [ユーザサービスデフォ
ルト定義] 121
- prc_corecompress [プロセスサービス定義]
94
- prc_coredump_filter [システム共通定義]
91
- prc_coredump_filter [ユーザサービス定義]
129
- prc_coredump_filter [ユーザサービスデ
フォルト定義] 124
- prc_coresave_path [プロセスサービス定義]
94
- prc_current_work_path [システム共通定
義] 91
- prc_port [システム共通定義] 89
- prc_prf_trace [プロセスサービス定義] 94
- prc_process_count [プロセスサービス定義]
94
- prc_recovery_resident [プロセスサービス定
義] 94
- prc_take_over_dlp_path [プロセスサービス定
義] 94
- prc_take_over_svspath [プロセスサービス定
義] 94
- prc_terminate_force [システム共通定義]
92
- predlpath [UAP 共用ライブラリのサーチパ
スの変更] 198
- predlpaths [UAP 共有ライブラリのサー
チパス名の表示] 198
- prckill [OpenTP1 のプロセスの強制停止]
199
- prcls [サーバの状態表示] 198
- prepath [ユーザサーバ, およびユーザサ
ーバから起動されるコマンドのサーチパス
の変更] 198
- prepaths [ユーザサーバ, およびユーザ
サーバから起動されるコマンドのサーチパ
ス名の表示] 198
- prcsvpath 142
- prcsvpath [プロセスサービス定義] 94
- prctctrl [prctee プロセスの停止と再開始]
197
- prctee [標準出力, 標準エラー出力のリダイ
レクト] 197
- prctee プロセスの停止と再開始 [prctctrl]
197
- preend_warning_watch_time [システム環
境定義] 87
- prf_buff_size [性能検証用トレース定義]
114
- prf_file_count [JNL 性能検証用トレース定
義] 114
- prf_file_count [LCK 性能検証用トレース定
義] 115
- prf_file_count [MCF 性能検証用トレース定
義] 131
- prf_file_count [TRN イベントトレース定
義] 115
- prf_file_count [XAR 性能検証用トレース定
義] 114
- prf_file_count [性能検証用トレース定義]
114
- prf_file_size [JNL 性能検証用トレース定
義] 114
- prf_file_size [LCK 性能検証用トレース定
義] 115
- prf_file_size [MCF 性能検証用トレース定
義] 131
- prf_file_size [TRN イベントトレース定義]
115
- prf_file_size [XAR 性能検証用トレース定
義] 114
- prf_file_size [性能検証用トレース定義] 114
- prf_information_level [LCK 性能検証用ト
レース定義] 115
- prf_information_level [TRN イベントト
レース定義] 115

prf_information_level [XAR 性能検証用トレース定義] 114
 prf_information_level [性能検証用トレース定義] 114
 prf_trace [システム共通定義] 90
 prf_trace_backup [JNL 性能検証用トレース定義] 114
 prf_trace_backup [性能検証用トレース定義] 114
 prfcd [トレース情報ファイルの編集出力] 207
 prfget [トレース情報ファイルの取り出し] 207
 process_privilege_name [ユーザサービス定義] 129
 process_privilege_name [ユーザサービスデフォルト定義] 124
 process_privilege_restrict [ユーザサービス定義] 129
 process_privilege_restrict [ユーザサービスデフォルト定義] 124
 purge_msgget [ユーザサービス定義] 127
 purge_msgget [ユーザサービスデフォルト定義] 121

Q

que_conf [システムサービス構成定義] 88
 que_io_maxrecsize [メッセージキューサービス定義] 110
 que_xidnum [メッセージキューサービス定義] 110
 quegrp [メッセージキューサービス定義] 110
 queinit [メッセージキュー用物理ファイルの割り当て] 206
 quels [キューグループの状態表示] 206
 querm [メッセージキュー用物理ファイルの削除] 206

R

rap_autoconnect_con_error_msg [ユーザサービス定義] 128

rap_autoconnect_con_error_msg [ユーザサービスデフォルト定義] 123
 rap_client_manager_node [rap リスナーサービス定義] 112
 rap_client_manager_port [rap クライアントマネージャサービス定義] 113
 rap_connect_interval [rap リスナーサービス定義] 112
 rap_connect_retry_count [rap リスナーサービス定義] 112
 rap_connect_retry_interval [rap リスナーサービス定義] 112
 rap_connection_assign_type [rap リスナーサービス定義] 112
 rap_extend_function [rap リスナーサービス定義] 113
 rap_extend_function [ユーザサービス定義] 129
 rap_extend_function [ユーザサービスデフォルト定義] 124
 rap_inquire_time [rap リスナーサービス定義] 111
 rap_inquire_timeout_message [rap リスナーサービス定義] 112
 rap_io_retry_interval [rap リスナーサービス定義] 112
 rap_listen_backlog [rap リスナーサービス定義] 112
 rap_listen_inf [rap クライアントマネージャサービス定義] 113
 rap_listen_port [rap リスナーサービス定義] 111
 rap_max_buff_size [rap リスナーサービス定義] 112
 rap_max_client [rap リスナーサービス定義] 112
 rap_message_id_change_level [rap リスナーサービス定義] 113
 rap_message_id_change_level [ユーザサービス定義] 129
 rap_message_id_change_level [ユーザサービスデフォルト定義] 124

- rap_msg_output_interval [rap リスナーサービス定義] 112
- rap_notify [rap リスナーサービス定義] 112
- rap_parallel_server [rap リスナーサービス定義] 111
- rap_recovery_server [rap リスナーサービス定義] 112
- rap_sock_count [rap リスナーサービス定義] 112
- rap_sock_interval [rap リスナーサービス定義] 112
- rap_stay_warning_interval [rap リスナーサービス定義] 113
- rap_stay_watch_time [rap リスナーサービス定義] 113
- rap_term_disconnect_time [rap リスナーサービス定義] 113
- rap_watch_time [rap クライアントマネージャサービス定義] 113
- rap_watch_time [rap リスナーサービス定義] 111
- rapdfgen [リモート API 機能に使用する定義の自動生成] 207
- rapls [rap リスナーおよび rap サーバの状態表示] 207
- rapsetup [リモート API 機能の実行環境の設定] 207
- rap リスナーおよび rap サーバの状態表示 [rapls] 207
- rap リスナーサービス定義 145
- raw I/O 機能 58
- raw I/O 機能を適用した OpenTP1 ファイルシステムの構築 58
- receive_from [ユーザサービス定義] 125
- receive_from [ユーザサービスデフォルト定義] 120
- redirect_file [システム環境定義] 87
- redirect_file_name [システム環境定義] 87
- redirect_file_size [システム環境定義] 87
- rmm_abort_command [監視対象 RM 定義] 109
- rmm_check_services [RMM サービス定義] 108
- rmm_command_gid [監視対象 RM 定義] 109
- rmm_command_uid [監視対象 RM 定義] 109
- rmm_command_watch_time [監視対象 RM 定義] 109
- rmm_conf [システムサービス構成定義] 88
- rmm_down_with_system [RMM サービス定義] 108
- rmm_get_pid_command [監視対象 RM 定義] 109
- rmm_start_command [監視対象 RM 定義] 109
- rmm_start_watch_time [監視対象 RM 定義] 109
- rmm_stop_command [監視対象 RM 定義] 109
- rmm_sysdown_with_rm [RMM サービス定義] 108
- rmm_system_behavior [RMM サービス定義] 108
- rpc_buffer_pool_max [ユーザサービス定義] 128
- rpc_buffer_pool_max [ユーザサービスデフォルト定義] 123
- rpc_datacomp [システム共通定義] 90
- rpc_delay_statistics [システム共通定義] 89
- rpc_destination_mode [ユーザサービス定義] 127
- rpc_destination_mode [ユーザサービスデフォルト定義] 122
- rpc_extend_function [rap リスナーサービス定義] 112
- rpc_extend_function [ユーザサービス定義] 127
- rpc_extend_function [ユーザサービスデフォルト定義] 121
- rpc_max_message_size [システム共通定義] 91
- rpc_message_level [システム共通定義] 89
- rpc_multi_tp1_in_same_host [システム共通定義] 89
- rpc_netmask [システム共通定義] 90

- rpc_port_base [システム共通定義] 90
- rpc_rap_auto_connect [ユーザサービス定義] 127
- rpc_rap_auto_connect [ユーザサービスデフォルト定義] 122
- rpc_rap_inquire_time [ユーザサービス定義] 127
- rpc_rap_inquire_time [ユーザサービスデフォルト定義] 122
- rpc_request_cancel_for_timedout [ユーザサービス定義] 127
- rpc_request_cancel_for_timedout [ユーザサービスデフォルト定義] 122
- rpc_response_statistics [rap リスナーサービス定義] 111
- rpc_response_statistics [ユーザサービス定義] 126
- rpc_response_statistics [ユーザサービスデフォルト定義] 121
- rpc_retry [システム共通定義] 89
- rpc_retry_count [システム共通定義] 89
- rpc_retry_interval [システム共通定義] 89
- rpc_router_retry_count [システム共通定義] 91
- rpc_router_retry_interval [システム共通定義] 91
- rpc_send_retry_count [システム共通定義] 90
- rpc_send_retry_count [ユーザサービス定義] 128
- rpc_send_retry_count [ユーザサービスデフォルト定義] 123
- rpc_send_retry_interval [システム共通定義] 90
- rpc_send_retry_interval [ユーザサービス定義] 128
- rpc_send_retry_interval [ユーザサービスデフォルト定義] 123
- rpc_server_busy_count [システム共通定義] 90
- rpc_service_retry_count [ユーザサービス定義] 127
- rpc_service_retry_count [ユーザサービスデフォルト定義] 121
- rpc_trace [rap リスナーサービス定義] 111
- rpc_trace [システム共通定義] 89
- rpc_trace [ユーザサービス定義] 127
- rpc_trace [ユーザサービスデフォルト定義] 122
- rpc_trace_name [rap リスナーサービス定義] 111
- rpc_trace_name [システム共通定義] 89
- rpc_trace_name [ユーザサービス定義] 127
- rpc_trace_name [ユーザサービスデフォルト定義] 122
- rpc_trace_size [rap リスナーサービス定義] 111
- rpc_trace_size [システム共通定義] 89
- rpc_trace_size [ユーザサービス定義] 127
- rpc_trace_size [ユーザサービスデフォルト定義] 122
- rpcdump [RPC トレースの出力] 207
- rpcmrg [RPC トレースのマージ] 207
- rpcstat [一時クローズ処理の実行状態の表示] 197
- RPC トレースの出力 [rpcdump] 207
- RPC トレースのマージ [rpcmrg] 207
- RPC 抑止リストの操作 [namblad] 201
- rts_cpd_collet_cpd [リアルタイム取得項目定義] 116
- rts_cpd_validt_cpd [リアルタイム取得項目定義] 116
- rts_dam_cache_block [リアルタイム取得項目定義] 118
- rts_dam_fj [リアルタイム取得項目定義] 118
- rts_dam_read [リアルタイム取得項目定義] 118
- rts_dam_read_err [リアルタイム取得項目定義] 118
- rts_dam_shm_pool [リアルタイム取得項目定義] 118
- rts_dam_trn_branch [リアルタイム取得項目定義] 118

- rts_dam_write〔リアルタイム取得項目定義〕 118
- rts_dam_write_err〔リアルタイム取得項目定義〕 118
- rts_item_max〔リアルタイム統計情報サービス定義〕 115
- rts_jnl_buf_full〔リアルタイム取得項目定義〕 116
- rts_jnl_io_wait〔リアルタイム取得項目定義〕 116
- rts_jnl_jnl_input〔リアルタイム取得項目定義〕 116
- rts_jnl_jnl_output〔リアルタイム取得項目定義〕 116
- rts_jnl_read〔リアルタイム取得項目定義〕 116
- rts_jnl_swap〔リアルタイム取得項目定義〕 116
- rts_jnl_wait_buf〔リアルタイム取得項目定義〕 116
- rts_jnl_write〔リアルタイム取得項目定義〕 116
- rts_lck_deadlock〔リアルタイム取得項目定義〕 116
- rts_lck_lock_acqst〔リアルタイム取得項目定義〕 116
- rts_lck_lock_wait〔リアルタイム取得項目定義〕 116
- rts_log_file〔リアルタイム統計情報サービス定義〕 116
- rts_log_file_backup〔リアルタイム統計情報サービス定義〕 116
- rts_log_file_count〔リアルタイム統計情報サービス定義〕 116
- rts_log_file_name〔リアルタイム統計情報サービス定義〕 116
- rts_log_file_size〔リアルタイム統計情報サービス定義〕 116
- rts_mcf_ap_scd_stay〔リアルタイム取得項目定義〕 119
- rts_mcf_ap_usr_srvc〔リアルタイム取得項目定義〕 119
- rts_mcf_in_msg_scd_wait〔リアルタイム取得項目定義〕 119
- rts_mcf_out_msg_norm_scd_wait〔リアルタイム取得項目定義〕 119
- rts_mcf_out_msg_prio_scd_wait〔リアルタイム取得項目定義〕 119
- rts_mcf_out_msg_resp_scd_wait〔リアルタイム取得項目定義〕 119
- rts_mcf_out_msg_sync_scd_wait〔リアルタイム取得項目定義〕 119
- rts_mcf_que_scd_wait_num〔リアルタイム取得項目定義〕 120
- rts_nam_global_cache_hit〔リアルタイム取得項目定義〕 117
- rts_nam_local_cache_hit〔リアルタイム取得項目定義〕 117
- rts_nam_lookup〔リアルタイム取得項目定義〕 117
- rts_nam_node_lookup〔リアルタイム取得項目定義〕 117
- rts_nam_node_lookup_responce〔リアルタイム取得項目定義〕 117
- rts_osl_dynmem_acq〔リアルタイム取得項目定義〕 117
- rts_osl_dynmem_pol〔リアルタイム取得項目定義〕 117
- rts_osl_stamem_acq〔リアルタイム取得項目定義〕 117
- rts_osl_stamem_pol〔リアルタイム取得項目定義〕 117
- rts_prc_prc_genert〔リアルタイム取得項目定義〕 117
- rts_prc_prc_num〔リアルタイム取得項目定義〕 117
- rts_prc_prc_term〔リアルタイム取得項目定義〕 117
- rts_prc_sys_abnml〔リアルタイム取得項目定義〕 117
- rts_prc_uap_abnml〔リアルタイム取得項目定義〕 117
- rts_que_delay_msg〔リアルタイム取得項目定義〕 117

- rts_que_delay_rec [リアルタイム取得項目定義] 117
- rts_que_delay_wrt [リアルタイム取得項目定義] 117
- rts_que_read [リアルタイム取得項目定義] 117
- rts_que_read_err [リアルタイム取得項目定義] 117
- rts_que_real_read [リアルタイム取得項目定義] 117
- rts_que_real_write [リアルタイム取得項目定義] 117
- rts_que_wait_buf [リアルタイム取得項目定義] 117
- rts_que_write [リアルタイム取得項目定義] 117
- rts_que_write_err [リアルタイム取得項目定義] 117
- rts_rpc_rpc_call [リアルタイム取得項目定義] 117
- rts_rpc_rpc_call_chained [リアルタイム取得項目定義] 117
- rts_rpc_rpc_ovrtim [リアルタイム取得項目定義] 118
- rts_rpc_usr_srvc [リアルタイム取得項目定義] 117
- rts_scd_lack_buf [リアルタイム取得項目定義] 118
- rts_scd_parallel [リアルタイム取得項目定義] 118
- rts_scd_scd_stay [リアルタイム取得項目定義] 118
- rts_scd_scd_wait [リアルタイム取得項目定義] 118
- rts_scd_schedule [リアルタイム取得項目定義] 118
- rts_scd_svc_scd_wait [リアルタイム取得項目定義] 118
- rts_scd_svc_using_buf [リアルタイム取得項目定義] 118
- rts_scd_using_buf [リアルタイム取得項目定義] 118
- rts_service_max [リアルタイム統計情報サービス定義] 115
- rts_swap_message [リアルタイム統計情報サービス定義] 116
- rts_tam_read [リアルタイム取得項目定義] 119
- rts_tam_read_err [リアルタイム取得項目定義] 119
- rts_tam_real_renew [リアルタイム取得項目定義] 118
- rts_tam_real_renew_time [リアルタイム取得項目定義] 118
- rts_tam_rec_refer [リアルタイム取得項目定義] 118
- rts_tam_rec_renew [リアルタイム取得項目定義] 118
- rts_tam_write [リアルタイム取得項目定義] 119
- rts_tam_write_err [リアルタイム取得項目定義] 119
- rts_trcput_interval [リアルタイム統計情報サービス定義] 115
- rts_trn_branch [リアルタイム取得項目定義] 118
- rts_trn_cmt_cmd [リアルタイム取得項目定義] 118
- rts_trn_commit [リアルタイム取得項目定義] 118
- rts_trn_haz_cmd [リアルタイム取得項目定義] 118
- rts_trn_mix_cmd [リアルタイム取得項目定義] 118
- rts_trn_rbk_cmd [リアルタイム取得項目定義] 118
- rts_trn_rollback [リアルタイム取得項目定義] 118
- rts_trn_sync_point [リアルタイム取得項目定義] 118
- rts_xar_call [リアルタイム取得項目定義] 119
- rts_xar_call_err [リアルタイム取得項目定義] 119

rts_xar_commit [リアルタイム取得項目定義] 119
 rts_xar_commit_err [リアルタイム取得項目定義] 119
 rts_xar_end [リアルタイム取得項目定義] 119
 rts_xar_end_err [リアルタイム取得項目定義] 119
 rts_xar_forget [リアルタイム取得項目定義] 119
 rts_xar_forget_err [リアルタイム取得項目定義] 119
 rts_xar_prepare [リアルタイム取得項目定義] 119
 rts_xar_prepare_err [リアルタイム取得項目定義] 119
 rts_xar_recover [リアルタイム取得項目定義] 119
 rts_xar_recover_err [リアルタイム取得項目定義] 119
 rts_xar_rollback [リアルタイム取得項目定義] 119
 rts_xar_rollback_err [リアルタイム取得項目定義] 119
 rts_xar_start [リアルタイム取得項目定義] 119
 rts_xar_start_err [リアルタイム取得項目定義] 119
 rtsedit [RTS ログファイルの編集出力] 207
 rtsls [リアルタイム統計情報の標準出力への出力] 207
 rtsput [リアルタイム統計情報サービス定義] 116
 rtssetup [リアルタイム統計情報サービスの実行環境の設定] 207
 rtsstats [リアルタイム統計情報の設定変更] 208
 RTS ログファイルの編集出力 [rtsedit] 207

S

scd_advertise_control [スケジューラサービス定義] 95
 scd_announce_server_status [スケジューラサービス定義] 95
 scd_hold_recovery [スケジューラサービス定義] 95
 scd_hold_recovery_count [スケジューラサービス定義] 95
 scd_message_level [スケジューラサービス定義] 95
 scd_pool_warning_interval [ユーザサービス定義] 129
 scd_pool_warning_interval [ユーザサービスデフォルト定義] 123
 scd_pool_warning_use_rate [ユーザサービス定義] 129
 scd_pool_warning_use_rate [ユーザサービスデフォルト定義] 123
 scd_poolfull_check_count [ユーザサービス定義] 129
 scd_poolfull_check_count [ユーザサービスデフォルト定義] 123
 scd_poolfull_check_interval [ユーザサービス定義] 129
 scd_poolfull_check_interval [ユーザサービスデフォルト定義] 123
 scd_port [スケジューラサービス定義] 95
 scd_refresh_process [ユーザサービス定義] 129
 scd_refresh_process [ユーザサービスデフォルト定義] 124
 scd_retry_of_comm_error [スケジューラサービス定義] 95
 scd_server_count [スケジューラサービス定義] 95
 scd_this_node_first [スケジューラサービス定義] 95
 scdbufgrp [スケジューラサービス定義] 95
 scdbufgrp [ユーザサービス定義] 129
 scdbufgrp [ユーザサービスデフォルト定義] 124
 scdchpre [プロセス数の変更] 199
 scdhold [スケジューラの閉塞] 199
 scdls [スケジューラの状態表示] 199
 scdmulti [スケジューラサービス定義] 95

- scdmulti [ユーザサービス定義] 129
- scdmulti [ユーザサービスデフォルト定義] 124
- scdrls [スケジュールの再開] 199
- scdsprc [プロセスの停止および再起動] 199
- scdsvcddef [ユーザサービス定義] 129
- scdsvcddef [ユーザサービスデフォルト定義] 124
- schedule_delay_abort [ユーザサービス定義] 128
- schedule_delay_abort [ユーザサービスデフォルト定義] 123
- schedule_delay_limit [ユーザサービス定義] 128
- schedule_delay_limit [ユーザサービスデフォルト定義] 123
- schedule_method [ユーザサービス定義] 127
- schedule_method [ユーザサービスデフォルト定義] 122
- schedule_priority [ユーザサービス定義] 125
- schedule_priority [ユーザサービスデフォルト定義] 120
- schedule_rate [スケジュールサービス定義] 95
- scs_prf_trace_level [rap リスナーサービス定義] 113
- server_count [システム環境定義] 87
- server_type [ユーザサービス定義] 126
- server_type [ユーザサービスデフォルト定義] 121
- service [ユーザサービス定義] 125
- service_expiration_time [ユーザサービス定義] 128
- service_expiration_time [ユーザサービスデフォルト定義] 122
- service_group [ユーザサービス定義] 125
- service_hold [ユーザサービス定義] 126
- service_hold [ユーザサービスデフォルト定義] 121
- service_priority_control [ユーザサービス定義] 126
- service_priority_control [ユーザサービスデフォルト定義] 121
- service_term_watch_time [ユーザサービス定義] 127
- service_term_watch_time [ユーザサービスデフォルト定義] 122
- service_wait_time [ユーザサービス定義] 127
- service_wait_time [ユーザサービスデフォルト定義] 122
- shmpool_attribute [システム環境定義] 87
- SQL Server との連携 73
- start_scheduling_timing [システム環境定義] 87
- static_shmpool_size [システム環境定義] 87
- statistics [システム共通定義] 91
- status_change_when_termining [ユーザサービス定義] 128
- status_change_when_termining [ユーザサービスデフォルト定義] 122
- stay_watch_abort [ユーザサービス定義] 129
- stay_watch_abort [ユーザサービスデフォルト定義] 124
- stay_watch_check_interval [ユーザサービス定義] 129
- stay_watch_check_interval [ユーザサービスデフォルト定義] 124
- stay_watch_check_rate [ユーザサービス定義] 129
- stay_watch_check_rate [ユーザサービスデフォルト定義] 123
- stay_watch_queue_count [ユーザサービス定義] 129
- stay_watch_queue_count [ユーザサービスデフォルト定義] 123
- stay_watch_start_interval [ユーザサービス定義] 129
- stay_watch_start_interval [ユーザサービスデフォルト定義] 124

sts_file_name_1 [ステータスサービス定義] 98
 sts_file_name_2 [ステータスサービス定義] 98
 sts_file_name_3 [ステータスサービス定義] 98
 sts_file_name_4 [ステータスサービス定義] 98
 sts_file_name_5 [ステータスサービス定義] 98
 sts_file_name_6 [ステータスサービス定義] 98
 sts_file_name_7 [ステータスサービス定義] 98
 sts_initial_error_switch [ステータスサービス定義] 98
 sts_last_active_file [ステータスサービス定義] 99
 sts_last_active_side [ステータスサービス定義] 99
 sts_single_operation_switch [ステータスサービス定義] 99
 stsclose [ステータスファイルのクローズ] 202
 stsfills [ステータスファイルの内容表示] 202
 stsinit [ステータスファイルの作成, 初期設定] 202
 stsls [ステータスファイルの状態表示] 202
 stsopen [ステータスファイルのオープン] 202
 stsrms [ステータスファイルの削除] 202
 stsswap [ステータスファイルのスワップ] 202
 system_id [システム共通定義] 89
 system_init_watch_time [システム環境定義] 87
 system_terminate_watch_time [システム環境定義] 87

T

tam_cbl_level [TAM サービス定義] 106
 tam_conf [システムサービス構成定義] 88
 tam_jnl_err_flag [TAM サービス定義] 106
 tam_max_filesize [TAM サービス定義] 106
 tam_max_recsz [TAM サービス定義] 106
 tam_max_tblnum [TAM サービス定義] 106
 tam_max_trnfilnum [TAM サービス定義] 106
 tam_max_trnnum [TAM サービス定義] 106
 tam_pool_attri [TAM サービス定義] 106
 tam_tbl_lock_mode [TAM サービス定義] 106
 tamadd [TAM テーブルの追加] 205
 tambkup [TAM ファイルのバックアップ] 206
 tamcre [TAM ファイルの初期設定] 205
 tamdel [TAM ファイルの削除] 205
 tamfre [TAM ファイルの回復] 206
 tamhold [TAM テーブルの論理閉塞] 205
 tamhsls [ハッシュ形式の TAM ファイルおよび TAM テーブルのシノニム情報の表示] 206
 tamlckls [TAM 排他資源名称の変換] 206
 tamload [TAM テーブルのロード] 205
 tamls [TAM テーブルの状態表示] 205
 tamrles [TAM テーブルの閉塞解除] 205
 tamrm [TAM テーブルの切り離し] 205
 tamrstr [TAM ファイルのリストア] 206
 tamtable [TAM サービス定義] 106
 tamunload [TAM テーブルのアンロード] 205
 TAM テーブルのアンロード [tamunload] 205
 TAM テーブルの切り離し [tamrm] 205
 TAM テーブルの状態表示 [tamls] 205
 TAM テーブルの追加 [tamadd] 205
 TAM テーブルの閉塞解除 [tamrles] 205
 TAM テーブルのロード [tamload] 205
 TAM テーブルの論理閉塞 [tamhold] 205
 TAM 排他資源名称の変換 [tamlckls] 206
 TAM ファイルの回復 [tamfre] 206
 TAM ファイルの削除 [tamedel] 205
 TAM ファイルの初期設定 [tamcre] 205

- TAM ファイルのバックアップ [tambkup] 206
- TAM ファイルのリストア [tamrstr] 206
- term_watch_count [プロセスサービス定義] 94
- term_watch_time [プロセスサービス定義] 94
- term_watch_time [ユーザサービス定義] 126
- term_watch_time [ユーザサービスデフォルト定義] 120
- termed_after_service [ユーザサービス定義] 127
- termed_after_service [ユーザサービスデフォルト定義] 122
- thdlock_sleep_time [システム共通定義] 90
- thdlock_sleep_time [システムサービス共通情報定義] 132
- thread_stack_size [トランザクションサービス定義] 97
- thread_yield_interval [トランザクションサービス定義] 97
- thread_yield_interval [ユーザサービス定義] 128
- thread_yield_interval [ユーザサービスデフォルト定義] 123
- tim_watch_count [タイマサービス定義] 93
- tp1console [標準出力リダイレクトファイルのコンソール出力] 211
- tptrnls [OSI TP 通信の未決着トランザクション情報の表示] 199
- trf_put [rap リスナーサービス定義] 111
- trf_put [ユーザサービス定義] 126
- trf_put [ユーザサービスデフォルト定義] 121
- trn_completion_limit_time [rap リスナーサービス定義] 112
- trn_completion_limit_time [クライアントサービス定義] 108
- trn_completion_limit_time [トランザクションサービス定義] 97
- trn_completion_limit_time [ユーザサービス定義] 129
- trn_completion_limit_time [ユーザサービスデフォルト定義] 124
- trn_cpu_time [rap リスナーサービス定義] 111
- trn_cpu_time [クライアントサービス定義] 107
- trn_cpu_time [トランザクションサービス定義] 96
- trn_cpu_time [ユーザサービス定義] 126
- trn_cpu_time [ユーザサービスデフォルト定義] 121
- trn_crm_use [トランザクションサービス定義] 96
- trn_expiration_time [rap リスナーサービス定義] 111
- trn_expiration_time [クライアントサービス定義] 107
- trn_expiration_time [トランザクションサービス定義] 96
- trn_expiration_time [ユーザサービス定義] 125
- trn_expiration_time [ユーザサービスデフォルト定義] 120
- trn_expiration_time_suspend [rap リスナーサービス定義] 111
- trn_expiration_time_suspend [クライアントサービス定義] 107
- trn_expiration_time_suspend [トランザクションサービス定義] 96
- trn_expiration_time_suspend [ユーザサービス定義] 125
- trn_expiration_time_suspend [ユーザサービスデフォルト定義] 120
- trn_extend_function [トランザクションサービス定義] 97
- trn_limit_time [rap リスナーサービス定義] 112
- trn_limit_time [クライアントサービス定義] 107
- trn_limit_time [トランザクションサービス定義] 96
- trn_limit_time [ユーザサービス定義] 127

- trn_limit_time [ユーザサービスデフォルト定義] 122
- trn_max_crm_subordinate_count [トランザクションサービス定義] 96
- trn_max_subordinate_count [トランザクションサービス定義] 96
- trn_optimum_item [rap リスナーサービス定義] 112
- trn_optimum_item [クライアントサービス定義] 107
- trn_optimum_item [トランザクションサービス定義] 96
- trn_optimum_item [ユーザサービス定義] 126
- trn_optimum_item [ユーザサービスデフォルト定義] 121
- trn_partial_recovery_type [rap リスナーサービス定義] 112
- trn_partial_recovery_type [クライアントサービス定義] 107
- trn_partial_recovery_type [トランザクションサービス定義] 96
- trn_partial_recovery_type [ユーザサービス定義] 127
- trn_partial_recovery_type [ユーザサービスデフォルト定義] 122
- trn_prf_event_trace_condition [トランザクションサービス定義] 97
- trn_prf_event_trace_level [トランザクションサービス定義] 97
- trn_prf_trace_level [システム共通定義] 90
- trn_processing_in_rm_error [トランザクションサービス定義] 96
- trn_rcv_open_close_scope [トランザクションサービス定義] 97
- trn_recovery_failmsg_interval [トランザクションサービス定義] 96
- trn_recovery_list_remove [トランザクションサービス定義] 96
- trn_recovery_list_remove_level [トランザクションサービス定義] 96
- trn_recovery_process_count [トランザクションサービス定義] 96
- trn_retry_count_rm_open [トランザクションサービス定義] 96
- trn_retry_interval_rm_open [トランザクションサービス定義] 96
- trn_rm_open_close_scope [トランザクションサービス定義] 96
- trn_rm_open_close_scope [ユーザサービス定義] 126
- trn_rm_open_close_scope [ユーザサービスデフォルト定義] 121
- trn_rollback_information_put [rap リスナーサービス定義] 112
- trn_rollback_information_put [クライアントサービス定義] 107
- trn_rollback_information_put [トランザクションサービス定義] 96
- trn_rollback_information_put [ユーザサービス定義] 127
- trn_rollback_information_put [ユーザサービスデフォルト定義] 122
- trn_rollback_response_receive [rap リスナーサービス定義] 112
- trn_rollback_response_receive [クライアントサービス定義] 107
- trn_rollback_response_receive [トランザクションサービス定義] 96
- trn_rollback_response_receive [ユーザサービス定義] 127
- trn_rollback_response_receive [ユーザサービスデフォルト定義] 122
- trn_start_recovery_interval [トランザクションサービス定義] 97
- trn_start_recovery_mode [トランザクションサービス定義] 97
- trn_start_recovery_watch_time [トランザクションサービス定義] 97
- trn_statistics_item [rap リスナーサービス定義] 112
- trn_statistics_item [クライアントサービス定義] 107
- trn_statistics_item [トランザクションサービス定義] 96

- trn_statistics_item [ユーザサービス定義]
126
 - trn_statistics_item [ユーザサービスデフォ
ルト定義] 121
 - trn_tran_process_count [トランザクション
サービス定義] 96
 - trn_tran_recovery_list [トランザクション
サービス定義] 96
 - trn_tran_statistics [トランザクションサー
ビス定義] 96
 - trn_wait_rm_open [トランザクションサー
ビス定義] 96
 - trn_watch_time [rap リスナーサービス定
義] 112
 - trn_watch_time [クライアントサービス定
義] 107
 - trn_watch_time [トランザクションサービ
ス定義] 96
 - trn_watch_time [ユーザサービス定義] 127
 - trn_watch_time [ユーザサービスデフォ
ルト定義] 122
 - trn_xa_commit_error [トランザクション
サービス定義] 97
 - trn_xar_use [トランザクションサービス定
義] 97
 - trncmt [トランザクションのコミット] 199
 - trndlinf [未決着トランザクション情報ファ
イルの削除] 199
 - trnftgt [トランザクションの強制終了] 199
 - trnlnkrm [拡張 RM 登録定義] 109
 - trnlnkrm [リソースマネージャの登録] 206
 - trnls [トランザクションの状態表示] 199
 - trnlsrm [リソースマネージャの情報の表示]
206
 - trnmkobj [トランザクション制御用オブ
ジェクトファイルの作成] 206
 - trnrbk [トランザクションのロールバック]
199
 - trnrmid [ユーザサービス定義] 129
 - trnrmid [ユーザサービスデフォルト定義]
124
 - trnstics [トランザクション統計情報の取得
開始, 終了] 199
 - trnstring [トランザクションサービス定義]
97
 - type [ユーザサービス定義] 126
 - type [ユーザサービスデフォルト定義] 120
 - TZ [ログサービス定義] 103
- ## U
-
- uap_conf [システムサービス構成定義] 88
 - uap_trace_file_put [システム共通定義] 91
 - uap_trace_file_put [ユーザサービス定義]
126
 - uap_trace_file_put [ユーザサービスデフォ
ルト定義] 120
 - uap_trace_max [rap リスナーサービス定
義] 111
 - uap_trace_max [ユーザサービス定義] 125
 - uap_trace_max [ユーザサービスデフォルト
定義] 120
 - UAP 共用ライブラリのサーチパスの変更
[prcdlpath] 198
 - UAP 共用ライブラリのサーチパス名の表示
[prcdlpaths] 198
 - UAP トレースの編集出力 [uatdump] 207
 - UAP の終了時の注意事項 40
 - UAP の翻訳と結合 31
 - uatdump [UAP トレースの編集出力] 207
 - uid [rap クライアントマネージャサービス定
義] 113
 - uid [rap リスナーサービス定義] 111
 - uid [ユーザサービス定義] 126
 - uid [ユーザサービスデフォルト定義] 121
 - UNIX 版 OpenTP1 と共通のコマンドの詳細
213
 - user_command [システム環境定義] 87
 - user_command_online [システム環境定義]
87
 - user_command_online_tplmgr_id [システ
ム環境定義] 87
 - user_server_ha [システム環境定義] 87
 - usmdump [共用メモリダンプの出力] 207

V

Visual Studio 使用時の注意事項 40
 Visual Studio と COBOL 言語を使用する環境が共存する場合の注意事項 41
 Visual Studio と併用する際の環境設定 39
 Visual Studio のプロジェクトでの UAP の作成 33
 Visual Studio のプロジェクトの設定内容 34

W

watch_next_chain_time [ユーザサービス定義] 125
 watch_next_chain_time [ユーザサービスデフォルト定義] 120
 watch_time [DAM サービス定義] 106
 watch_time [rap リスナーサービス定義] 113
 watch_time [TAM サービス定義] 106
 watch_time [インタバルサービス定義] 98
 watch_time [クライアントサービス定義] 108
 watch_time [システム共通定義] 92
 watch_time [システムジャーナルサービス定義] 100
 watch_time [ジャーナルサービス定義] 99
 watch_time [スケジュールサービス定義] 95
 watch_time [ステータスサービス定義] 99
 watch_time [チェックポイントダンプサービス定義] 101
 watch_time [トランザクションサービス定義] 97
 watch_time [ネームサービス定義] 93
 watch_time [プロセスサービス定義] 94
 watch_time [マルチノード構成定義] 103
 watch_time [ユーザサービス定義] 129
 watch_time [ユーザサービスデフォルト定義] 124
 watch_time [ログサービス定義] 103
 Windows の [サービス] ダイアログによる OpenTP1 の開始 157

Windows の [サービス] ダイアログによる OpenTP1 の停止 161
 Windows 版 OpenTP1 固有のコマンドの詳細 230
 Windows 版 OpenTP1 での運用と操作の概要 156
 Windows 版 OpenTP1 と UNIX 版 OpenTP1 の運用コマンドの差異 196
 Windows 版 OpenTP1 と UNIX 版 OpenTP1 の関数の差異 18
 Windows 版 OpenTP1 と UNIX 版 OpenTP1 の機能の差異 6
 Windows 版 OpenTP1 の GUI 178

X

xar_eventtrace_level [XA リソースサービス定義] 98
 xar_eventtrace_record [XA リソースサービス定義] 98
 xar_msdtc_use [XA リソースサービス定義] 98
 xar_prf_trace_level [XA リソースサービス定義] 98
 xar_session_time [XA リソースサービス定義] 98
 xarevtr [XAR イベントトレース情報の表示] 199
 xarfile [XA リソースサービス定義] 98
 xarfills [XAR ファイルの状態表示] 200
 xarforce [XAR トランザクション状態の変更] 200
 xarhold [XA リソースサービスの閉塞] 200
 xarinit [XAR ファイルの作成] 200
 xarls [XAR トランザクション情報の表示] 200
 xarrles [XA リソースサービスの閉塞解除] 200
 xarm [XAR ファイルの削除] 200
 XAR イベントトレース情報の表示 [xarevtr] 199
 XAR トランザクション状態の変更 [xarforce] 200

XAR トランザクション情報の表示 [xarls] 200
 XAR ファイルの削除 [xarrm] 200
 XAR ファイルの作成 [xarinit] 200
 XAR ファイルの状態表示 [xarfills] 200
 xat_aso_con_event_svcname [XATMI 通信サービス定義] 110
 xat_aso_discon_event_svcname [XATMI 通信サービス定義] 110
 xat_aso_failure_event_svcname [XATMI 通信サービス定義] 110
 xat_conf [システムサービス構成定義] 88
 XAT_CONNECT_RESP_TIME [ユーザーサービス定義] 130
 xat_connect_resp_time [ユーザーサービス定義] 129
 XAT_CONNECT_RESP_TIME [ユーザーサービスデフォルト定義] 124
 xat_connect_resp_time [ユーザーサービスデフォルト定義] 123
 xat_osi_usr [ユーザーサービス定義] 127
 xat_osi_usr [ユーザーサービスデフォルト定義] 122
 xat_trn_expiration_time [ユーザーサービス定義] 127
 xat_trn_expiration_time [ユーザーサービスデフォルト定義] 122
 xatinitaeq [XATMI 通信サービス定義] 110
 xatinitapt [XATMI 通信サービス定義] 110
 xatsrvadd [XATMI 通信サービス定義] 110
 XA リソースサービスの閉塞 [xarhold] 200
 XA リソースサービスの閉塞解除 [xarrles] 200

あ

アーカイブ状態の表示 [jnlarls] 203
 アプリケーション異常終了回数の初期化 [mcfacldcap] 208
 アプリケーションに関するタイマ起動要求の削除 [mcfadltap] 208
 アプリケーションに関するタイマ起動要求の表示 [mcfalstap] 208

アプリケーションの状態表示 [mcfalsap] 208
 アプリケーションの閉塞 [mcfadctap] 208
 アプリケーションの閉塞解除 [mcfaactap] 208
 アプリケーションプログラムの起動 [mcfuevt] 209
 アプリケーション例外発生時の対策 175
 アンロードジャーナルファイル, およびグローバルアーカイブアンロードジャーナルファイルの時系列ソート, およびマージ [jnlsort] 204
 アンロードジャーナルファイル, またはグローバルアーカイブアンロードジャーナルファイルの編集出力 [jnledit] 204
 アンロードジャーナルファイル, またはグローバルアーカイブアンロードジャーナルファイルのレコード出力 [jnlrput] 204
 アンロードジャーナルファイルの複写 [jnlcopy] 203

い

一時クローズ処理の実行状態の表示 [rpcstat] 197

う

運用コマンドの一覧 195

か

開始形態の決定 159
 稼働統計情報の出力 [jnlstts] 204
 環境設定の手順 48
 環境変数の設定 53
 監査ログ機能の環境設定 [dcauditsetup] 201
 関数の一覧 18

き

[起動/停止オプション] ダイアログボックス 190

起動通知情報の強制的無効化 [namunavl]
200

機能解説 5

機能の一覧 6

キャッシュブロック数のしきい値の設定

[damchdef] 205

キャッシュブロック数の取得 [damchinf]
205

キューグループの状態表示 [quels] 206

共用メモリダンプの出力 [usmdump] 207

共用メモリの状態表示 [dcshmls] 196

く

クライアントサービス定義 144

こ

コネクションの解放 [mcftdctcn] 208

コネクションの確立 [mcftactcn] 208

コネクションの切り替え [mcftchcn] 208

コネクションの状態表示 [mcftlscn] 208

このマニュアルの読み方 3

コマンドによる OpenTP1 の開始 158

コマンドによる OpenTP1 の停止 161

コマンドの指定 (COBOL2002) 36

さ

サーバ型コネクションの確立要求の受付開始
[mcftonln] 208

サーバ型コネクションの確立要求の受付終了
[mcftofln] 208

サーバの開始 [dcsvstart] 198

サーバの終了 [dcsvstop] 198

サーバの状態表示 [prcls] 198

サービス関数動的ローディング機能使用時の
UAP 共用ライブラリ 43

サービスグループの状態表示 [mcftlssg]
209

サービスグループの入力キュー削除
[mcftdlqsg] 210

サービスグループの入力キュー処理の保留
[mcfthldiq] 209

サービスグループの入力キュー処理の保留解
除 [mcftrlisq] 210

サービスグループの閉塞 [mcftdctsg] 209

サービスグループの閉塞解除 [mcftactsg]
209

サービスの状態表示 [mcftlssv] 210

サービスの閉塞 [mcftdctsv] 210

サービスの閉塞解除 [mcftactsv] 210

再開始中読み込み済みジャーナル関係のファ
イル情報の表示 [jnlrlnf] 202

サポートディスク形式 59

サンプルシナリオテンプレートの登録 169

サンプルシナリオテンプレートの名称と処理
166

サンプルシナリオテンプレートの利用 166

サンプルプログラム 42

サンプルプログラムを作成する場合に必要な
設定 35

し

システム環境定義 135

システム共通定義 139

システムサービス情報定義 154

システム定義の一覧 86

システム定義のオペランドの指定

[dcjchconf] 196

システム定義の説明形式 133

システム定義のチェック [dcdefchk] 197

システム統計情報の取得開始、終了

[dcstats] 196

システム統計情報の標準出力へのリアルタイ
ム編集出力 [dcreport] 197

指定ファイル内容のコンソール出力

[ntbtail] 211

自動アンロード機能の制御 [jnlatusl] 203

シナリオテンプレートからの OpenTP1 コマ
ンドの実行 [dcjcmdux] 196

シナリオテンプレート定義ファイル 164

シナリオテンプレートの詳細 169

シナリオテンプレートを利用したシステムの
運用 164

シナリオの登録 164

ジャーナル関係のファイル情報の表示
[jnlls] 202

ジャーナル関係のファイルのアンロード
[jnlunlfg] 203

ジャーナル関係のファイルのオープン
[jnlopnfg] 202

ジャーナル関係のファイルの回復 [jnlmkrf]
203

ジャーナル関係のファイルのクローズ
[jnlclsfg] 202

ジャーナル関係のファイルの削除 [jnlrm]
203

ジャーナル関係のファイルの初期設定
[jnlinit] 202

ジャーナル関係のファイルのステータス変更
[jnlchgfg] 203

ジャーナル関係のファイルのスワップ
[jnlswpfg] 203

ジャーナル関係の物理ファイルの削除
[jnldeplf] 203

ジャーナル関係の物理ファイルの割り当て
[jnladdpf] 203

ジャーナルサービス定義 143

障害対策 173

障害発生時の現象と原因 173

障害発生時の取得情報の参照 176

[詳細情報] ダイアログボックス 183

シングルスレッド環境での UAP 実行時の注
意事項 40

す

スケールアウトの運用 164

スケジュールの再開始 [scdrls] 199

スケジュールの状態表示 [scdls] 199

スケジュールの閉塞 [scdhold] 199

スタブの作成時の注意事項 40

ステータスファイルのオープン [stsoopen]
202

ステータスファイルのクローズ [stsclose]
202

ステータスファイルの削除 [stsrml] 202

ステータスファイルの作成, 初期設定
[stssinit] 202

ステータスファイルの状態表示 [stsls] 202

ステータスファイルのスワップ [stsswap]
202

ステータスファイルの内容表示 [stsfills]
202

せ

性能検証用トレース情報解析 [dcalzprf]
208

製品情報の表示 [dcpllist] 197

セッションの開始 [mcftactss] 210

セッションの終了 [mcftdctss] 210

た

代行送信の開始 [mcftstalt] 209

代行送信の終了 [mcftedalt] 209

て

定義の変更 51

定義ファイルの作成 51

データベース連携時の設定 61

デッドロック情報ファイルとタイムアウト情
報ファイルの削除 [lckrminf] 200

と

特権制御機能 16

ドメイン構成の変更 (システム共通定義使
用) [namndchg] 200

ドメイン構成の変更 (ドメイン定義ファイル
使用) [namchgfl] 200

ドメイン代表スケジュールサービスの登録,
削除 [namdomainsetup] 200

ドメイン定義ファイルの更新 [dcjnamch]
196

トラブルシュート情報の削除 [dccspool]
197

トランザクション制御用 DLL の作成
[njsmkdll] 211

トランザクション制御用オブジェクトファイ
ルの作成 [trnmkobj] 206

トランザクション統計情報の取得開始、終了
〔trnstics〕 199
トランザクションの強制終了〔trnfgt〕 199
トランザクションのコミット〔trncmt〕 199
トランザクションの状態表示〔trnls〕 199
トランザクションのロールバック〔trnrbk〕
199
トレース情報ファイルの取り出し〔prfget〕
207
トレース情報ファイルの編集出力〔prfed〕
207

に

入出力キューの内容複写〔mcftdmpqu〕 210

ね

ネットワークの状態表示〔mcftlsln〕 208

の

ノードのオプション情報の変更〔namndopt〕
201
ノードリスト情報の削除〔namndrm〕 201
ノードリストファイルの削除〔namnldel〕
201
ノードリストファイルの作成〔namnlcre〕
201
ノードリストファイルの内容表示
〔namnldsp〕 201

は

排他情報の表示〔lckls〕 200
排他制御用テーブルのプール情報の表示
〔lckpool〕 200
ハッシュ形式の TAM ファイルおよび TAM
テーブルのシノニム情報の表示〔tamhsls〕
206
バッファグループの使用状況表示
〔mcftlsbuf〕 210

ひ

標準出力、標準エラー出力のリダイレクト
〔prctee〕 197
標準出力・標準エラー出力を確認するための
機能 11
標準出力・標準エラー出力を確認するための
機能を使用する場合に指定するオペランド
の対応 15
標準出力リダイレクト機能 11
標準出力リダイレクトファイルのコンソール
出力〔tp1console〕 211
標準出力リダイレクトファイルのコンソール
出力機能 13

ふ

ファイル回復用ジャーナルの集積〔jnlcolc〕
203
物理ファイルの削除〔damdel〕 205
物理ファイルの初期設定〔damload〕 204
物理ファイルのバックアップ〔dambkup〕
205
物理ファイルのリストア〔damrstr〕 205
プログラム作成時の Windows 版 OpenTP1
固有の注意事項 40
プロセスサービス定義 142
プロセスサービスの再起動および定義の反映
〔dcreset〕 196
プロセス数の変更〔scdchprc〕 199
プロセスの停止および再起動〔scdrsprc〕
199

ほ

ポート数不足への対策（TCP/IP ポート数の
チューニング） 175
ほかの OpenTP1 シリーズのマニュアルとの
関係 2
保守資料の取得〔dcrasget〕 197

ま

マップファイルのパス名変更〔dcmapchg〕
210

マップファイルのロード済み資源の表示
 [demapls] 210
 マネジャノードの変更 [nammstr] 201
 マルチ OpenTP1 操作コンソールの作成
 [dcmakecon] 211
 マルチ OpenTP1 の運用 163
 マルチ OpenTP1 の設定 82
 マルチ OpenTP1 のセットアップ状態の確認
 [dcdls] 211
 マルチ OpenTP1 のセットアップとアンセッ
 トアップ [dcsetupml] 211
 マルチノードエリア, サブエリアの開始
 [dcmstart] 196
 マルチノードエリア, サブエリアの終了
 [dcmstop] 196

み

未決着トランザクション情報ファイルの削除
 [trndlinf] 199

め

メイクファイルの例 (COBOL2002:MHP の
 場合) 37
 メイクファイルの例 (COBOL2002:SPP の
 場合) 36
 メイクファイルの例 (C 言語:MHP の場合)
 32
 メイクファイルの例 (C 言語:SPP の場合)
 31
 メッセージキュー用物理ファイルの削除
 [querm] 206
 メッセージキュー用物理ファイルの割り当て
 [queinit] 206
 メッセージ多重処理状況の表示 [mcftlstrd]
 208
 メッセージログのリアルタイム出力機能の切
 り替え [logcon] 201
 メッセージログファイルの内容表示
 [logcat] 201

ゆ

ユーザサーバ, およびユーザサーバから起動
 されるコマンドのサーチパスの変更
 [prcpath] 198
 ユーザサーバ, およびユーザサーバから起動
 されるコマンドのサーチパス名の表示
 [prcpathls] 198
 ユーザサーバが起動直後にダウンする場合の
 対処 52
 ユーザサーバプロセスのサービス関数の入れ
 替え 44
 ユーザサービス定義 150
 ユーザサービスデフォルト定義 147
 ユーザタイム監視の状態表示 [mcftlsutm]
 211

ら

ライブラリサーチパス 44
 ランタイムライブラリのパスの設定 51

り

リアルタイム統計情報サービスの実行環境の
 設定 207
 リアルタイム統計情報の設定変更 [rtsstats]
 208
 リアルタイム統計情報の標準出力への出力
 [rtsls] 207
 リソースグループの接続の強制解除
 [jnlardis] 204
 リソースマネジャの情報の表示 [trnlstrm]
 206
 リソースマネジャの登録 [trnlnkrm] 206
 リモート API 機能に使用する定義の自動生
 成 [rapdfgen] 207
 リモート API 機能の実行環境の設定
 [rapsetup] 207

ろ

[ログ確認] ダイアログボックス 186
 論理端末に関するメッセージジャーナルの取
 得開始 [mcftactmj] 209

論理端末に関するメッセージジャーナルの取得終了〔mcftdctmj〕 209
論理端末に対する継続問い合わせ応答処理の強制終了〔mcftendct〕 209
論理端末の出力キュー削除〔mcftdlqle〕 209
論理端末の出力キュー処理の保留〔mcfthldoq〕 209
論理端末の出力キュー処理の保留解除〔mcfttrlsoq〕 209
論理端末の状態表示〔mcftlsle〕 209
論理端末の閉塞〔mcftdctle〕 209
論理端末の閉塞解除〔mcftactle〕 209
論理端末のメッセージキューの先頭スキップ〔mcftspqle〕 209
論理ファイルの回復〔damfrc〕 205
論理ファイルの切り離し〔damrm〕 205
論理ファイルの状態表示〔damls〕 204
論理ファイルの追加〔damadd〕 205
論理ファイルの閉塞解除〔damrles〕 205
論理ファイルの論理閉塞〔damhold〕 205