

OpenTP1 Version 7
分散トランザクション処理機能
OpenTP1 システム定義

文法書

3000-3-D52-60

前書き

■ 対象製品

マニュアル「OpenTP1 解説」を参照してください。

■ 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

HP-UX は、Hewlett-Packard Development Company, L.P.のオペレーティングシステムの名称です。IBM, AIX は、世界の多くの国で登録された International Business Machines Corporation の商標です。

IBM, AIX 5L は、世界の多くの国で登録された International Business Machines Corporation の商標です。

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

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。

OSF は、Open Software Foundation, Inc.の商標です。

PA-RISC は、Hewlett-Packard Development Company, L.P.の商標です。

Red Hat は、米国およびその他の国で Red Hat, Inc. の登録商標もしくは商標です。

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。

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。

X/Open は、The Open Group の英国ならびに他の国における登録商標です。

XATMI は、X/Open Company Limited が開発したアプリケーションインタフェースの名称です。

その他記載の会社名、製品名は、それぞれの会社の商標もしくは登録商標です。

プログラムプロダクト [P-9D64-3F31, P-9D64-8531, P-9D64-8931, R-19451-216, R-19452-216, R-19453-216, R-19454-216, R-19455-216, R-19456-216, R-19459-216, R-1945A-216, R-1945C-216, R-1945D-216, R-1945E-216, R-F19456-2156, R-F19456-21C6] には、Oracle Corporation またはその子会社、関連会社が著作権を有している部分が含まれています。

プログラムプロダクト [P-9D64-3F31, P-9D64-8531, P-9D64-8931, R-19451-216, R-19452-216, R-19453-216, R-19454-216, R-19455-216, R-19456-216, R-19459-216, R-1945A-216, R-1945C-216, R-1945D-216, R-1945E-216, R-F19456-2156, R-F19456-21C6] には、UNIX System Laboratories, Inc.が著作権を有している部分が含まれています。

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 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 4 月 3000-3-D52-60

■ 著作権

All Rights Reserved. Copyright (C) 2006, 2015, Hitachi, Ltd.

変更内容

変更内容 (3000-3-D52-60) uCosminexus TP1/Server Base 07-50, uCosminexus TP1/Server Base(64) 07-50, uCosminexus TP1/Message Control 07-50, uCosminexus TP1/Message Control(64) 07-50, uCosminexus TP1/NET/Library 07-50, uCosminexus TP1/NET/Library(64) 07-50

追加・変更内容	変更箇所
次のオペランドおよびオプションの指定値の上限を変更した。 - システム環境定義 server_count - スケジューラサービス定義 scd_server_count - システムサービス共通情報定義 max_socket_descriptors max_open_fds - MCF マネージャ定義の mcfmexp 定義コマンドの-g オプション - MCF マネージャ定義の mcfmexp 定義コマンドの-l オプション - MCF マネージャ定義の mcfmsts 定義コマンドの-g オプション - MCF 通信構成定義の mcftsts 定義コマンドの-a オプション - MCF 通信構成定義の mcftsts 定義コマンドの-l オプション	2.2.1, 2.2.9, 3. システム環境定義 server_count, 3. スケジューラサービス定義 scd_server_count, 4.2.1, 4.2.2, 4.2.6, 5. システムサービス共通情報定義 max_socket_descriptors, max_open_fds, 5. MCF マネージャ定義 mcfmexp, mcfmsts, 5. MCF 通信構成定義 mcftsts
システム定義のデフォルト値を選択できるようにした。 これに伴い、次のオペランドを追加した。 システム環境定義 default_value_option	2.2.1, 3. システム環境定義 default_value_option, 付録 D
次のオペランドおよびオプションのデフォルト値の説明を変更した。 - システム共通定義 trn_prf_trace_level ipc_listen_sockbufset fil_prf_trace_delay_time - ロックサービス定義 lck_deadlock_info - スケジューラサービス定義 ipc_tcpnodelay - ログサービス定義 log_filesize - 性能検証用トレース定義 prf_file_size	2.2.4, 2.2.5, 2.2.9, 2.2.17, 2.2.34, 2.2.41, 3. システム共通定義 trn_prf_trace_level, ipc_listen_sockbufset, fil_prf_trace_delay_time, 3. ロックサービス定義 lck_deadlock_info, 3. スケジューラサービス定義 ipc_tcpnodelay, 3. ログサービス定義 log_filesize,

追加・変更内容	変更箇所
• ユーザサービスデフォルト定義 ipc_tcpnodelay scd_refresh_process • MCF マネージャ定義の mcfmcomn 定義コマンドの-i オプション • MCF マネージャ定義の mcfmuap 定義コマンドの-e オプション segsz • MCF マネージャ定義の mcfmuap 定義コマンドの-c オプション order noansreply • システムサービス共通情報定義 mcf_prf_trace_level	3. 性能検証用トレース定義 prf_file_size, 3. ユーザサービスデフォルト定義 ipc_tcpnodelay, scd_refresh_process, 4.2.1, 4.2.6, 5. MCF マネージャ定義 mcfmcomn, mcfmuap, 5. システムサービス共通情報定義 mcf_prf_trace_level, 付録 D
スケジューラサービスプロセスで使用するファイルおよびパイプの最大数を指定できるようにした。 これに伴い、次のオペランドを追加した。 • スケジューラサービス定義 max_open_fds また、次のオペランドに説明を追加した。 • ユーザサービス定義 max_open_fds	2.2.9, 3. スケジューラサービス定義 max_open_fds, 3. ユーザサービス定義 max_open_fds
ジャーナルサービス、チェックポイントダンプサービス、およびグローバルアーカイブジャーナルサービスが出力するメッセージの種別を変更できるようにした。 これに伴い、次のオペランドを追加した。 • 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 • グローバルアーカイブジャーナルサービス定義 jnl_message_id_change_level	2.2.14, 2.2.16, 2.2.20, 3. ジャーナルサービス定義 jnl_message_id_change_level, 3. チェックポイントダンプサービス定義 cpd_message_id_change_level, 3. グローバルアーカイブジャーナルサービス定義 jnl_message_id_change_level, 7.2
性能検証用トレースバッファを拡張できるようにした。 これに伴い、次のオペランドを追加した。 • 性能検証用トレース定義 prf_buff_size また、バッファの拡張に伴い、使用する共用メモリが増加するため、静的共用メモリの算出式を変更した。	2.2.34, 3. 性能検証用トレース定義 prf_buff_size, 付録 B.1(1)(n), 付録 B.1(2)(f)
次のオペランドで指定する値が、同一 OpenTP1 ノード内での値であることを明確にした。	3. システム環境定義 server_count,

追加・変更内容	変更箇所
- システム環境定義 server_count - プロセスサービス定義 prc_process_count	3. プロセスサービス定義 prc_process_count
システムサービス構成定義の clt_conf オペランドに、CUP からクライアント拡張サービスを使用する場合の説明を追加した。	3. システムサービス構成定義 clt_conf
dcsvstart コマンドの説明に、ユーザサーバの起動順序の説明を追加した。	3. ユーザサービス構成定義 dcsvstart
システム共通定義の name_notify オペランドの注意事項に、起動通知機能を使用できない理由の説明を追加した。	3. システム共通定義 name_notify
システムサービス共通情報定義でも thdlock_sleep_time オペランドが指定できることの説明を追加した。	3. システム共通定義 thdlock_sleep_time
システム共通定義の prc_current_work_path オペランドに、tmp ディレクトリを削除する場合の説明を追加した。	3. システム共通定義 prc_current_work_path
ネームサービス定義の name_audit_conf オペランドの注意事項に、ノード監視機能を使用できない理由の説明を追加した。	3. ネームサービス定義 name_audit_conf
ステータスデーモンへの RPC 要求電文の拡張に伴い、スケジュールサービス定義の scd_hold_recovery_count オペランドの説明を変更した。	3. スケジュールサービス定義 scd_hold_recovery_count
TCP_NODELAY オプションを使用することで、ネットワーク帯域の負荷が大きくなる場合の対処の説明を変更した。	3. スケジュールサービス定義 ipc_tcpnodelay 3. ユーザサービス定義 ipc_tcpnodelay
次のオプションの説明に、指定可能な文字の説明を追加した。 - trnstring 定義コマンドの-o - trnstring 定義コマンドの-c - trnstring 定義コマンドの-O - trnstring 定義コマンドの-C	3. トランザクションサービス定義 trnstring
チェックポイントダンプサービス定義の jnl_reduced_mode オペランドに、縮退運転時は assurance_count オペランドの指定が無視される説明を追加した。	3. チェックポイントダンプサービス定義 jnl_reduced_mode
次のオペランドで指定する値について説明を追加した。 TAM サービス定義 tam_max_tblnum tam_max_filesize	3. TAM サービス定義 tam_max_tblnum, tam_max_filesize
rap リスナーサービス定義の注意事項の説明を変更した。	3. rap リスナーサービス定義

追加・変更内容	変更箇所
ユーザサービス定義の term_watch_time オペランドについて、hold オペランド、term_watch_time オペランドの指定値と、サーバの閉塞の関係に関する説明を追加した。	3. ユーザサービス定義 term_watch_time
ユーザサービス定義の term_watch_time オペランドについて、連続異常終了限界経過時間が解除される契機の説明を追加した。	3. ユーザサービス定義 term_watch_time
ユーザサービス定義の auto_restart オペランドの説明を変更した。	3. ユーザサービス定義 auto_restart
次の定義コマンドの指定数の上限を拡張した。 • mcfmsvg • mcfaalcap	4.1.6
出力キューの入力保留中に UAP がメッセージを送信したときの動作を指定できるようにした。 これに伴い、次のオペランドを追加した。 • mcfmuap 定義コマンドの -c オプション otqinhold	4.2.1, 5. MCF マネージャ定義 mcfmuap
OpenTP1 再開始時のエラーイベントを引き継ぐかどうかを指定できるようにした。 これに伴い、次のオペランドを追加した。 • mcfmuap 定義コマンドの -c オプション errevt_recovery	4.2.1, 5. MCF マネージャ定義 mcfmuap
OpenTP1 の正常開始または再開始で、MCF のプロセス間で互いの状態を監視する際の時間間隔を指定できるようにした。 これに伴い、次のオペランドを追加した。 • システムサービス共通情報定義 mcf_start_watch_interval	4.2.6, 5. システムサービス共通情報定義 mcf_start_watch_interval, 7.2
OpenTP1 の内部で使用するスレッドスタック領域のサイズを指定できるようにした。 これに伴い、次のオペランドを追加した。 • システムサービス共通情報定義 thread_stack_size mcf_stack_size	4.2.6, 5. システムサービス共通情報定義 thread_stack_size, mcf_stack_size, 7.2
定義オブジェクトファイルの出力先にファイルがすでに存在していたとしても、定義オブジェクトファイルを上書きできるようにした。 これに伴い、次のオプションを追加した。 • mcfmnggr の -r • mcfcomn の -r • mcfpsvr の -r • mcfapli の -r • mcflink の -r	4.3.1, 4.3.2

追加・変更内容	変更箇所
MCF 定義オブジェクト解析コマンドの解析結果での省略値の扱いについて、説明を追加した。	4.3.3(5)
MCF マネージャ定義の mcfmexp 定義コマンドの -g オプションについて、指定値の説明を変更した。また、同コマンドの -l オプションについても、指定値と注意事項の説明を変更した。	5. MCF マネージャ定義 mcfmexp
閉塞されている論理端末のメッセージが未処理送信メッセージ滞留時間の対象とならないことを追加した。	5. MCF 通信構成定義 mcfttim
ユーザサーバの多重度を変更する場合に見直しが必要な定義の一覧に、rap リスナーサービス定義の rap_parallel_server オペランドを追加した。	7.3.2
チェックポイントダンプの有効化チェック間隔の変更に伴い、チェックポイントダンプの静的共用メモリの算出式を変更した。	付録 B.1(1)(h)
TP1/Server Base の共用メモリの拡張に伴い、静的共用メモリおよび動的共用メモリの算出式を変更した。	付録 B.1(1)(m), 付録 B.1(1)(q), 付録 B.1(3)(b), 付録 B.1(3)(c), 付録 B.1(3)(d), 付録 B.1(5)
MCF サービス用の共用メモリの見積もり式を変更した。	付録 B.2(1)(a), 付録 B.2(1)(b), 付録 B.2(1)(d), 付録 B.2(2)(a)

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-D52-50) uCosminexus TP1/Server Base 07-06, uCosminexus TP1/Server Base(64) 07-06

追加・変更内容
メッセージキューサービス定義の定義ファイル数を 0~1 に変更した。
ノード構成の変更（ノードの追加や削除）に自動的に対応できるようにした（ノード自動追加機能）。これに伴い、次のオペランド，または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 また、次のオペランドに説明を追加した。

追加・変更内容

- システム共通定義
all_node
name_notify
all_node_ex
- ネームサービス定義
name_total_size
name_cache_size
name_audit_conf
name_audit_interval
name_audit_watch_time
name_cache_validity_time

TCP/IP の送信バッファサイズ、および受信バッファサイズに、OS の送信バッファサイズを適用できるようにした。
これに伴い、次のオペランドで 0 を指定できるようにした。

- システム共通定義
ipc_rcvbuf_size
ipc_sndbuf_size
- ユーザサービスデフォルト定義
ipc_rcvbuf_size
ipc_sndbuf_size
- ユーザサービス定義
ipc_rcvbuf_size
ipc_sndbuf_size

また、定義チェック時に出力される問題識別コードに「RPC-0027」を追加した。

ユーザサーバごとに、共有化したバッファの使用サイズを制限できるようにした。
これに伴い、次のオプションを追加した。

- 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

また、次のオプションの最大値を変更した。

- スケジュールサービス定義
scdbufgrp 定義コマンドの -n

次のオプション、およびオペランドの説明を変更した。

- スケジュールサービス定義
scdbufgrp 定義コマンドの -l
- ユーザサービス定義
message_store_buflen

追加・変更内容
定義変更をシステムに反映するため dcreset コマンドの実行が必要となる説明を追加した。
DCUAPCONFPATH 環境変数にユーザサービスデフォルト定義ファイルが設定できる説明を追加した。
システム環境定義の user_command オペランド、および user_command_online オペランドの、実行タイミング、および標準出力について説明を変更した。
システム環境定義の server_count オペランドに指定するユーザサーバ数の説明を追加した。
システム共通定義の max_socket_descriptors オペランドについて説明を追加した。
自 OpenTP1 内のシステムサービスプロセスの数の具体的な求め方を max_socket_descriptors オペランドの説明に追加した。
システム共通定義の my_host オペランドについて、hostname コマンドが返す名称が IP アドレスとマッピングできる環境設定が必要である説明を追加した。
性能検証用トレースの情報を CSV 形式で出力し、トレース解析できるようにした。
メッセージ格納バッファプール長 (message_store_buflen オペランドの指定値) の計算について、説明を追加した。
select システムコールの待ち時間の一覧表からバージョン情報を削除した。
ネームサービス定義の max_socket_descriptors オペランドの計算式を変更した。
非常駐プロセスを指定しているユーザサーバや scdrsprc コマンドを使用している場合に、一時的にプロセス数が増加するタイミングについて説明を追加した。
ログサービス定義の TZ オペランドを変更する際に、dcsetup の実行が必要である説明を追加した。
DAM サービス定義の dam_update_block オペランドの指定値の算出例を追加した。
DAM サービス定義の dam_update_block_over オペランドの flush の説明を追加した。
TAM サービス定義の tam_max_filesize オペランドの説明を変更した。
uap_trace_max オペランドの説明に UAP トレースの説明を追加した。
leveldown_queue_count オペランドのリンク誤りを修正した。
プロセス強制停止で、設定時間と所要時間に誤差が発生することをオペランドの説明に追加した。
使用する OS が Windows の場合は、次のオプションを指定できない旨を追加した。 • mcfmcomn 定義コマンドの -w • mcftenv 定義コマンドの -q
mcftpsvr 定義コマンドの -o オプションの説明を変更した。

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

追加・変更内容
core ファイルに共有メモリを含めるかどうかを設定できるようにした。

追加・変更内容

これに伴い、次のオペランドを追加した。

- システム共通定義
prc_coredump_filter
- ユーザサービスデフォルト定義
prc_coredump_filter
- ユーザサービス定義
prc_coredump_filter

リモート API 機能における性能検証用トレースの取得レベル、およびリモート API 機能の拡張レベルを指定できるようにした。
これに伴い、次のオペランドを追加した。

- 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
rap_extend_function
- ユーザサービス定義
rap_extend_function

一つのサービス要求ごとに実行するプロセスを起動し直せるようにした（非常駐 UAP プロセスのリフレッシュ機能）。
これに伴い、次のオペランドを追加した。

- ユーザサービスデフォルト定義
scd_refresh_process
- ユーザサービス定義
scd_refresh_process

一つのリソースマネージャを複数の制御単位に分け、接続するユーザ名称などを変更してリソースマネージャに接続できるようにした（リソースマネージャ接続先選択機能）。

これに伴い、次のオプションを追加した。

- ユーザサービスデフォルト定義
trnrmid 定義コマンドの-k
- ユーザサービス定義
trnrmid 定義コマンドの-k

次のオペランドの指定値について、HP-UX 版の 32 ビット版の OpenTP1 を使用する場合は、注意事項を追加した。

- システム環境定義
static_shmpool_size
dynamic_shmpool_size

jnl_max_datasize オペランドの指定値を求める計算式で、MCF サービスを使用する場合の計算式を変更した。

ログサービスにおける、メッセージ量に関する注意事項を追加した。

max_open_fds オペランドに指定する、UAP プロセスでオープンするファイルディスクリプタの最大数に関する説明を追加した。

追加・変更内容

mcfmcomn 定義コマンドの-p オプションの指定値の上限を変更した。

次のオプションの指定値を求める計算式を変更し、かつ 64 ビット版の説明を追加した。

- mcfmcomn 定義コマンドの-j
- mcfmuap 定義コマンドの-j
- mcftcomn 定義コマンドの-j

mcfmexp 定義コマンドの-l オプションについて、指定する論理端末数に関する説明を追加した。

次のオプションの rescheduleint オペランドについて、異常終了した MHP を自動的に再スケジュールするまでの所要量に関する説明を追加した。

- mcfmuap 定義コマンドの-r
- mcfaalcap 定義コマンドの-d

MCF トレースファイルの大きさと、MCF トレースバッファの大きさを求める計算式を追加した。

uCosminexus TP1/Server Base 07-02, uCosminexus TP1/Message Control 07-01

追加・変更内容

前回のオンライン停止時に残っていた未処理受信メッセージや未送信メッセージを引き継ぎ、TP1/Message Control の構成を変更できるようにした (MCF 構成変更再開機能)。

これに伴い、システムサービス共通情報定義の putenv 形式に DCMCFQUEBAK オペランドを追加した。

また、MCF マネジャプロセスで使用する共用メモリの算出式を変更した。

uCosminexus TP1/Message Control 07-00, uCosminexus TP1/Message Control(64) 07-00

追加・変更内容

リモート MCF サービスに関する記述を削除した。

これに伴い、次のオペランドを削除した。

- システムサービス構成定義
mrs_conf
- MCF アプリケーション定義
mcfaalcap 定義コマンドの type

また次のコマンドのオプションを削除した。

- MCF マネジャ定義
mcfmcomn 定義コマンドの-r, -c

変更内容 (3000-3-D52-40) uCosminexus TP1/Server Base 07-04, uCosminexus TP1/Server Base(64) 07-04, uCosminexus TP1/Message Control 07-05, uCosminexus

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

追加・変更内容
トランザクションの回復処理で、XA 関数の発行順序を変更できるようにした。 これに伴い、次のオペランドとオプションを追加した。 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取得契機のスキップ回数を監視できるようにした。 これに伴い、次のオペランドを追加した。 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 登録定義で、コンパイルオプションとリンケージオプションを指定できるようにした。 これに伴い、trnlnkrm 定義コマンドに次のオプションを追加した。 -C -B
ユーザサービス構成定義の dcsvstart 定義コマンドの指定例を追加した。
システム共通定義の all_node オペランドに無効 IP アドレスを指定した場合の注意事項を追加した。
システム共通定義の name_notify オペランドの注意書きを変更した。
システム共通定義の rpc_port_base オペランドの説明で、使用するポート番号の数は、プロセスサービス定義の prc_process_count オペランドで指定した値（最大同時起動サーバプロセス数）に 128 を加えた値である旨に変更した。
オペランドの指定値の優先順位の説明を追加した。
プロセスサービス定義の prc_process_count オペランドで指定するプロセス数に含まれるコマンドについて、説明を追加した。
サービス関数動的ローディング機能で使用する、UAP 共用ライブラリのサーチパスをオンライン中に変更できるようにした。 これに伴い、次のオペランドを追加した。 - プロセスサービス定義 prc_take_over_dlpath また、次のオペランドの説明を変更した。 - ユーザサービス定義 service
指定値を論理和で指定する場合の説明を追加した。
MQA サービスを使用する場合の、ジャーナルに取得するレコードデータの最大長を求める計算式を変更した。
syslog に出力できるメッセージ長に関する注意事項を変更した。
英語環境での syslog へのメッセージ出力可否を追加した。

追加・変更内容
rap リスナーサービス定義の max_socket_descriptors オペランドの、最大数の計算式の説明を追加した。
異常終了した MHP を、自動的に再スケジュールできるようにした。 これに伴い、次のオプションとオペランドを追加した。 - MCF マネージャ定義の mcfmuap 定義コマンドの -r オプション reschedulecnt rescheduleint reschedulelog - MCF アプリケーション定義の mcfaalcap 定義コマンドの -d オプション reschedulecnt rescheduleint reschedulelog
非応答型の MHP からの問い合わせ応答をできるようにした。 これに伴い、次のオペランドを追加した。 mcfmuap 定義コマンドの -c オプション noansreply
データ操作言語で、MHP をコミットできるようにした。 これに伴い、次のオペランドを追加した。 mcfmuap 定義コマンドの -c オプション commitdml
mcftlsmn 定義コマンドの -o オプションは、OS が Windows の場合は指定できない旨を追加した。
mcfaalcap 定義コマンドの次のオペランドの説明に、dc_mcf_rollback 関数に DCMCFRTRY を指定（ROLLBACK 文に RTRY を指定）した場合の説明を追加した。 -n オプションの aplihold -g オプションの srvghold -v オプションの servhold -d オプションの holdlimit
mcftlsbuf コマンドで、送受信バッファの最大使用量を表示できるようにした。
プロセスサービス定義の prc_take_over_svpath オペランドを、再開始時に変更できるシステムサービス定義の一覧に追加した。
XATMI 通信サービスの静的共用メモリの所要量の説明を追加した。
32 ビットでトランザクションアクセス種別がグローバルアクセスの場合の DAM の動的共用メモリの算出式を変更した。
mcftlsle コマンドで、最大未送信メッセージ数を表示できるようにした。
MCF サービス用の動的共用メモリの見積もり式を変更した。
トランザクションサービス定義の trn_optimum_item オペランドに対応するバージョン 5 以前のオペランドの説明を追加した。

変更内容 (3000-3-D52-30) uCosminexus TP1/Server Base 07-03, uCosminexus TP1/Server Base(64) 07-03, uCosminexus TP1/Message Control 07-03, uCosminexus TP1/Message Control(64) 07-03, uCosminexus TP1/NET/Library 07-04, uCosminexus TP1/NET/Library(64) 07-04

追加・変更内容
ジャーナルサービスで性能検証用トレース (JNL 性能検証用トレース) を出力できるようにした。 これに伴い、次の定義およびオペランドを追加した。 - システム共通定義 jnl_prf_event_trace_level - JNL 性能検証用トレース定義 また、次のオペランドの説明を変更した。 - 性能検証用トレース定義 prf_trace_backup
ロックサービスを使用した排他制御の各種イベントの性能検証用トレース (LCK 性能検証用トレース) を出力できるようにした。 これに伴い、次の定義およびオペランドを追加した。 - ロックサービス定義 lck_prf_trace_level - LCK 性能検証用トレース定義 また、次のオペランドの説明を変更した。 - 性能検証用トレース定義 prf_trace_backup
定義作成時の注意事項を追加した。
定義の 1 行の最大長の説明を変更した。
プロセスをアボートしなくても UAP トレース (UAP トレースデータファイル) を取得できるようにした。 これに伴い、次のオペランドを追加した。 - システム共通定義 uap_trace_file_put - ユーザサービスデフォルト定義 uap_trace_file_put - ユーザサービス定義 uap_trace_file_put
特定のノードのサービス情報を優先的に使用する機能 (サービス情報優先度指定機能) を追加した。 これに伴い、次のオペランドの説明を変更した。 システム共通定義 all_node name_domain_file_use また、次のオペランドを追加した。

追加・変更内容
ネームサービス定義 name_cache_validity_time
次のオペランドで、RPC の処理のリトライを行うタイミングの説明を追加した。 システム共通定義 rpc_retry rpc_retry_count rpc_retry_interval
OpenTP1 ファイルへのアクセス要求で、イベントトレース（FIL イベントトレース）を出力できるようにした。 これに伴い、次のオペランドを追加した。 - システム共通定義 fil_prf_trace_option fil_prf_trace_delay_time また、次のオペランドの説明を変更した。 - 性能検証用トレース定義 prf_trace_backup
次のオペランドのデフォルト値の説明を変更した。 - トランザクションサービス定義 thread_stack_size - TAM サービス定義 tam_pool_attri
次のオペランドの説明を追加した。 - システム共通定義 ipc_sockctl_highwater ipc_sockctl_watchtime - rap リスナーサービス定義 ipc_sockctl_highwater ipc_sockctl_watchtime
次のオペランドの fixed を指定する場合の説明を変更した。 - システム環境定義 shmpool_attribute - DAM サービス定義 dam_cache_attribute - TAM サービス定義 tam_pool_attri
次のオペランドの、グローバルドメインの説明を変更した。 システム共通定義 rpc_multi_tp1_in_same_host

追加・変更内容

次のオペランドの説明に、MQA サービスを使用する場合のジャーナルに取得するレコードデータの最大長の計算式を追加した。

- システムジャーナルサービス定義
jnl_max_datasize

次のオペランドで Y を指定した場合の説明を追加した。

- システムジャーナルサービス定義
jnl_unload_check
- アーカイブジャーナルサービス定義
jnl_unload_check
- ユーザサービス定義
service_hold
service_priority_control

次のオペランドの、syslog 出力の説明を変更した。

- ログサービス定義
log_syslog_out
DCSYSLOGOUT

次のオペランドの、バッファ領域長の仮定値計算式を変更した。

- DAM サービス定義
dam_cache_size
dam_cache_size_fix

rap リスナーサービス定義で、注意事項を追加、および変更した。

rtspout 定義コマンドのオプションと指定できる取得対象オブジェクトの説明を変更した。

次のオペランドの指定を省略した場合の説明を追加した。

- ユーザサービスデフォルト定義
rpc_trace
rpc_trace_name
rpc_trace_size
- ユーザサービス定義
rpc_trace

trnrmid 定義コマンドで -i オプションを指定する場合の注意事項を追加した。

次のオペランドの説明に、status_change_when_termining オペランドを指定する場合の説明を追加した。

- ユーザサービス定義
node_down_restart

次の定義コマンドで、-j オプションの指定値の上限を拡張した。

- mcfmcomn
- mcfmuap
- mcftcomn

追加・変更内容
mcfttim 定義コマンドの-t オプションに、タイマ起動要求メッセージについての説明を追加した。
mcftalcle 定義コマンドで、-m オプションに 0 を指定、または指定を省略した場合の出力メッセージ数を変更した。
mcfaalcap 定義コマンドの-n オプションに、アプリケーション起動機能使用時の定義方法の説明を追加した。
mcfaalcap 定義コマンドの-g オプション、および-v オプションに、指定時の注意と定義例を追加した。
定義の変更手順、注意事項の説明を変更した。
OpenTP1 システムの変更に影響する定義として、状態引き継ぎ定義に次の定義を追加した。 • mcfmsts 定義コマンドの-g オプション • mcfmsts 定義コマンドの-v オプション
OpenTP1 システムの変更に影響する定義に、scd の scd_hold_recovery_count を追加した。
ホスト名または IP アドレスの変更手順を追加した。
TP1/Server Base 用の静的共用メモリの算出式を変更した。
共用メモリプールのサイズの算出式を追加した。
MCF サービス用の静的共用メモリの算出式と、定義指定値の関係についての説明を追加した。
MCF サービス用の静的共用メモリの算出式を変更した。
監査イベントに「OpenTP1 サービス開始」、および「OpenTP1 サービス停止」を追加した。
定義チェック時に出力されるメッセージ、および問題識別コードを追加、変更した。
バージョン 5 以前からバージョン 7 に移行する場合の注意事項についての説明を追加した。

uCosminexus TP1/Message Control 07-02, uCosminexus TP1/NET/Library 07-03

追加・変更内容
MHP でサービス関数動的ローディング機能を使用できるようにした。 これに伴い、次のオペランドの説明を変更した。 • ユーザサービス定義 service
MCF 静的共用メモリが不足した場合、未使用領域から自動的に追加確保できるようにした。 これに伴い、次のオプションを追加した。 • mcfmcomn 定義コマンドの-i オプション

uCosminexus TP1/Message Control 07-01, uCosminexus TP1/NET/Library 07-01

追加・変更内容
メッセージ送受信での主なイベントで、性能検証用トレース（MCF 性能検証用トレース）を出力できるようにした。 これに伴い、次の定義およびオペランドを追加した。 • ユーザサービスデフォルト定義

追加・変更内容

mcf_prf_trace

- ユーザサービス定義

mcf_prf_trace

- MCF 性能検証用トレース定義
- システムサービス情報定義

mcf_prf_trace

- システムサービス共通情報定義

mcf_prf_trace_level

また、次のオペランドの説明を変更した。

- 性能検証用トレース定義

prf_trace_backup

リアルタイム統計情報の取得項目として、MCF の情報も取得できるようにした。

これに伴い、次の定義およびオペランドを追加した。

- リアルタイム取得項目定義

rts_mcf_ap_scd_stay

rts_mcf_ap_usr_srvc

rts_mcf_in_msg_scd_wait

rts_mcf_out_msg_sync_scd_wait

rts_mcf_out_msg_resp_scd_wait

rts_mcf_out_msg_prio_scd_wait

rts_mcf_out_msg_norm_scd_wait

rts_mcf_que_scd_wait_num

変更内容 (3000-3-D52-20) uCosminexus TP1/Server Base 07-02, uCosminexus TP1/Message Control 07-01, uCosminexus TP1/NET/Library 07-01

追加・変更内容

XA リソースサービスで性能検証用トレース (prf トレース) を出力できるようにした。

これに伴い、次の定義およびオペランドを追加した。

- XAR 性能検証用トレース定義
- XAR リソースサービス定義

xar_prf_trace_level オペランド

定義の 1 行の最大長の説明を変更した。

システムジャーナルファイルを使用しないでシステムを運用する機能 (ジャーナルファイルレス機能) を追加した。

これに伴い、次の内容を追加した。

- システム共通定義の jnl_fileless_option オペランド
- ジャーナルサービス定義, およびシステムジャーナルサービス定義の注意事項
- システムサービス定義を登録する定義ファイル数
- ユーザサービス定義の atomic_update オペランドの注意事項

追加・変更内容

- ジャーナルファイルレス機能を使用するノードでの共用メモリの見積もり式

また、ジャーナルファイルレス機能が有効になっている場合に、XA リソースサービスは使用できないことを追加した。

スレッド間で排他が競合した場合のスレッドの待ち時間を設定する機能を拡張した。

これに伴い、次の定義の `thdlock_sleep_time` オペランドの指定範囲を変更した。

- システム共通定義
- システムサービス共通情報定義

また、指定時の注意事項を追加した。

さらに、ユーザサービス定義、およびユーザサービスデフォルト定義の `thdlock_sleep_time` オペランドを削除した。

Linux および AIX でユーザ認証機能を使用できるようにした。

これに伴い、システム共通定義の `client_uid_check` オペランドの省略値を変更した。

ネームサービスでイベントトレースを出力できるようにした。

これに伴い、システム共通定義に `nam_prf_trace_level` オペランドを追加した。

プロセスサービスでイベントトレースを出力できるようにした。

これに伴い、プロセスサービス定義の `prc_prf_trace` オペランドの説明を変更した。

トランザクションサービス定義の `trn_tran_process_count` オペランドに、MCF サービスを使用する場合の指定範囲を追加した。

システムジャーナルファイルの並列アクセス機能を追加した。

これに伴い、システムジャーナルサービス定義に次のオペランドおよびオプションを追加した。

- `jnl_max_file_dispersion` オペランド
- `jnl_min_file_dispersion` オペランド
- `jnladdpf` 定義コマンドの `-e` オプション

また、アーカイブジャーナルサービス定義の次のオペランドおよびオプションの説明を変更した。

- `jnl_max_file_dispersion` オペランド
- `jnl_min_file_dispersion` オペランド
- `jnladdpf` 定義コマンドの `-e` オプション

監査ログを出力する機能を追加した。

これに伴い、次のオペランドを追加した。

ログサービス定義

- `log_audit_out`
- `log_audit_path`
- `log_audit_size`
- `log_audit_count`
- `log_audit_message`

rap リスナーサービス定義

- `log_audit_out_suppress`
- `log_audit_message`

追加・変更内容

rap クライアントマネージャサービス定義

- log_audit_out_suppress
- log_audit_message

ユーザサービスデフォルト定義

- log_audit_out_suppress
- log_audit_message

ユーザサービス定義

- log_audit_out_suppress
- log_audit_message

リモート API 機能に関する次の説明を追加した。

rap リスナーサービス定義

- rap_term_disconnect_time
- rap_stay_watch_time
- rap_stay_warning_interval
- watch_time

再開始時に変更できるシステムサービス定義

- rap リスナーサービス定義
- rap クライアントマネージャサービス定義

また、次の説明を変更した。

rap リスナーサービス定義

- rap_connect_interval
- rap_recovery_server

rap リスナーサービス定義の注意事項

PRF トレースファイルのバックアップ抑止機能を追加した。

これに伴い、性能検証用トレース定義に prf_trace_backup オペランドを追加した。

リアルタイム統計情報サービスで、RTS ログファイルをバックアップする機能を追加した。

これに伴い、リアルタイム統計情報サービス定義に rts_log_file_backup オペランドを追加した。

リアルタイム統計情報サービスで取得できる項目を追加した。

これに伴い、リアルタイム取得項目定義に次のオペランドを追加した。

- rts_scd_svc_scd_wait オペランド
- rts_scd_svc_using_buf オペランド
- rts_scd_parallel オペランド

スケジュールサービスの動作をサービス単位で指定できるようになった。

これに伴い、次の定義に scdsvcddef 定義コマンドを追加した。

- ユーザサービス定義
- ユーザサービスデフォルト定義

また、ユーザサービス定義の次のオペランドに scdsvcddef 定義コマンドを指定した場合の留意点を追加した。

追加・変更内容

- message_store_bufllen オペランド
- message_cell_size オペランド

さらに、TP1/Server Base 用の静的共用メモリの算出式で、スケジューラのメモリ算出式を変更した。

サービス関数を動的にローディングできる機能を追加した。

これに伴い、ユーザサービス定義の service オペランドの指定値に、「UAP 共用ライブラリ名」を追加した。

また、リアルタイム統計情報サービス定義の rtspat 定義コマンドの注意事項を追加した。

Linux 版で、syslog メッセージの出力を失敗した場合に、syslog メッセージの出力をリトライできるようにした。

これに伴い、次に示すオペランドの説明を変更した。

- log_syslog_elist
- log_syslog_elist_rint

タイムゾーンの指定についての注意事項を追加した。

リアルタイム統計情報の取得項目の定義例を追加した。

type オペランドで指定できるサービスグループの種別から RAP を削除した。

入力キューに滞留するメッセージキューを監視する機能を追加した。

これに伴い、MCF マネージャ定義にサービスグループ属性定義を追加した。

mcfmsmsg 定義コマンドの指定範囲を変更した。

アプリケーション属性定義を指定するとき、モデルとする定義の指定内容を流用できる機能を追加した。

これに伴い、MCF アプリケーション定義の mcfaalcap 定義コマンドに -N オプションを追加した。

MCF 定義オブジェクト生成ユーティリティ起動コマンドの -o オプションの説明を変更した。

次に示すオプションを指定した場合の引き継ぐ対象の説明を変更した。

- mcfmsts 定義コマンドの -g オプション
- mcfmsts 定義コマンドの -v オプション
- mcftsts 定義コマンドの -a オプション

OpenTP1 システムの変更に伴って見直す必要のある定義の一覧を追加した。

システム共通定義の prf_trace オペランドを指定した場合の共用メモリの所要量を追加した。

TP1/Server Base 用の動的共用メモリの算出式で、メモリキューだけの構成の場合の算出方法を追加した。

定義チェック時に出力されるメッセージおよび問題識別コードを追加、変更した。

はじめに

このマニュアルは、分散トランザクション処理機能 OpenTP1 の定義方法について説明したものです。

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

■ 対象読者

システム管理者、およびシステム設計者を対象としています。

このマニュアルは、マニュアル「OpenTP1 解説」を前提として書かれていますので、あらかじめお読みいただくことをお勧めします。

■ その他の前提条件

このマニュアルをお読みになる際のその他の前提情報については、マニュアル「OpenTP1 解説」を参照してください。

目次

前書き	2
変更内容	4
はじめに	23

1	概要	30
1.1	定義の体系	31
1.2	定義の構成	37
1.2.1	システムサービス定義の構成	38
1.2.2	ネットワークコミュニケーション定義の構成	40
1.2.3	メッセージキュー定義の構成	41
1.2.4	ネットワークライブラリ定義の構成	41
1.3	定義の作成手順	42
1.4	定義の規則	44
1.4.1	基本事項	44
1.4.2	記述形式	45
1.4.3	記号の説明	48

2	システムサービス定義の概要	51
2.1	概要	52
2.1.1	システムサービス定義とファイル名	52
2.2	定義の種類	56
2.2.1	システム環境定義の内容	56
2.2.2	システムサービス構成定義の内容	57
2.2.3	ユーザサービス構成定義の内容	58
2.2.4	システム共通定義の内容	58
2.2.5	ロックサービス定義の内容	63
2.2.6	タイマサービス定義の内容	64
2.2.7	ネームサービス定義の内容	65
2.2.8	プロセスサービス定義の内容	66
2.2.9	スケジュールサービス定義の内容	67
2.2.10	トランザクションサービス定義の内容	69
2.2.11	XA リソースサービス定義の内容	73
2.2.12	インタバルサービス定義の内容	74
2.2.13	ステータスサービス定義の内容	74
2.2.14	ジャーナルサービス定義の内容	75

2.2.15	システムジャーナルサービス定義の内容	76
2.2.16	チェックポイントダンプサービス定義の内容	78
2.2.17	ログサービス定義の内容	79
2.2.18	マルチノード構成定義の内容	83
2.2.19	マルチノード物理定義の内容	83
2.2.20	グローバルアーカイブジャーナルサービス定義の内容	84
2.2.21	アーカイブジャーナルサービス定義の内容	84
2.2.22	DAM サービス定義の内容	85
2.2.23	TAM サービス定義の内容	87
2.2.24	クライアントサービス定義の内容	88
2.2.25	IST サービス定義の内容	90
2.2.26	RMM サービス定義の内容	91
2.2.27	監視対象 RM 定義の内容	91
2.2.28	拡張 RM 登録定義の内容	92
2.2.29	XATMI 通信サービス定義の内容	92
2.2.30	メッセージキューサービス定義の内容	93
2.2.31	ユーザサービスネットワーク定義の内容	94
2.2.32	rap リスナーサービス定義の内容	94
2.2.33	rap クライアントマネージャサービス定義の内容	98
2.2.34	性能検証用トレース定義の内容	99
2.2.35	XAR 性能検証用トレース定義の内容	100
2.2.36	JNL 性能検証用トレース定義の内容	100
2.2.37	LCK 性能検証用トレース定義の内容	100
2.2.38	TRN イベントトレース定義の内容	101
2.2.39	リアルタイム統計情報サービス定義の内容	101
2.2.40	リアルタイム取得項目定義の内容	102
2.2.41	ユーザサービスデフォルト定義の内容	109
2.2.42	ユーザサービス定義の内容	117

3 システムサービス定義の詳細 127

システム環境定義	128
システムサービス構成定義	140
dcsvstart	144
ユーザサービス構成定義	145
dcsvstart	146
システム共通定義	148
dcbindht	191
ロックサービス定義	195
タイマサービス定義	200
ネームサービス定義	202
namnlfil	217

プロセスサービス定義 219
prcsvpath 227
スケジュールサービス定義 228
scdbufgrp 236
scdmulti 239
トランザクションサービス定義 243
trnstring 270
XA リソースサービス定義 275
xarfile 278
インタバルサービス定義 279
ステータスサービス定義 280
ジャーナルサービス定義 286
jnldfsv 291
システムジャーナルサービス定義 292
jnladdfg 303
jnladdpf 304
チェックポイントダンプサービス定義 306
jnladdfg 311
jnladdpf 313
ログサービス定義 315
マルチノード構成定義 335
dcmarea 337
マルチノード物理定義 339
dcprcport 340
グローバルアーカイブジャーナルサービス定義 342
jnldfsv 346
アーカイブジャーナルサービス定義 347
jnladdfg 350
jnladdpf 351
DAM サービス定義 353
damcache 367
damchlmnt 369
damfile 370
TAM サービス定義 372
tamtable 377
クライアントサービス定義 379
IST サービス定義 394
istdef 396
RMM サービス定義 397
監視対象 RM 定義 399
拡張 RM 登録定義 402
trnlnkrm 403
XATMI 通信サービス定義 406
xatsrvadd 409
メッセージキューサービス定義 410

quegrp 412
ユーザサービスネットワーク定義 415
dcsvgdef 418
rap リスナーサービス定義 427
rap クライアントマネージャサービス定義 455
性能検証用トレース定義 458
XAR 性能検証用トレース定義 462
JNL 性能検証用トレース定義 464
LCK 性能検証用トレース定義 467
TRN イベントトレース定義 469
リアルタイム統計情報サービス定義 471
rtspat 475
リアルタイム取得項目定義 479
ユーザサービスデフォルト定義 506
trnrmid 525
scdbufgrp 526
scdmulti 527
scdsvcdef 528
ユーザサービス定義 529
trnrmid 593
scdbufgrp 596
scdmulti 599
scdsvcdef 600
ユーザサービス定義の指定値と UAP との関係 602

4 ネットワークコミュニケーション定義の概要 608

- 4.1 概要 609
 - 4.1.1 ネットワークコミュニケーション定義とファイル名 609
 - 4.1.2 MCF サービスと定義との関係 610
 - 4.1.3 MCF の運用形態と定義との関係 611
 - 4.1.4 ネットワークコミュニケーション定義とシステムサービス定義との関連 612
 - 4.1.5 アプリケーション名とサービス名との関係 613
 - 4.1.6 ネットワークコミュニケーション定義のコマンド 613
- 4.2 定義の種類 615
 - 4.2.1 MCF マネージャ定義の内容 615
 - 4.2.2 MCF 通信構成定義の内容 618
 - 4.2.3 MCF アプリケーション定義の内容 620
 - 4.2.4 MCF 性能検証用トレース定義の内容 622
 - 4.2.5 システムサービス情報定義の内容 622
 - 4.2.6 システムサービス共通情報定義の内容 623
- 4.3 定義オブジェクトファイルの生成 624
 - 4.3.1 mcf×××× (MCF 定義オブジェクト生成ユーティリティ起動コマンド) 625
 - 4.3.2 mcflink (MCF 定義結合ユーティリティ起動コマンド) 626

4.3.3 mcf×××× (MCF 定義オブジェクト解析コマンド) 627

5 ネットワークコミュニケーション定義の詳細 631

MCF マネージャ定義 632

mcfmenv 633

mcfmcomn 634

mcfmcname 638

mcfmuap 639

mcfmqgid 648

mcfmexp 650

mcfmsts 652

mcfmsmsg 653

mcfmsvg 654

MCF 通信構成定義 656

mcftenv 658

mcftcomn 660

mcfttred 662

mcfttim 663

mcfttrc 665

mcftsts 668

mcftbuf 669

mcftpsvr 670

mcftalcle 672

mcftped 674

MCF アプリケーション定義 675

mcfaenv 677

mcfaalcap 678

MCF 性能検証用トレース定義 689

システムサービス情報定義 691

システムサービス共通情報定義 693

6 マルチ OpenTP1 システムの定義 698

6.1 定義のしかた 699

7 OpenTP1 システムの定義の変更 701

7.1 定義の変更手順 702

7.2 再開始時に変更できる定義 703

7.3 OpenTP1 システムの変更に影響する定義 713

7.3.1 ユーザサーバの追加 713

7.3.2 ユーザサーバの多重度の変更 715

7.3.3 ノードの追加 717

7.3.4 ジャーナルファイルグループの追加 718

7.3.5 ホスト名または IP アドレスの変更 719

8 定義例 721

- 8.1 OpenTP1 システムを構成する定義例 722
- 8.2 リアルタイム統計情報の取得項目の定義例 737
 - 8.2.1 リアルタイム統計情報サービス定義の定義例 737
 - 8.2.2 オペランドの指定値による取得情報の違い 739

付録 742

- 付録 A アプリケーションプログラムとシステム環境設定の関連 743
 - 付録 A.1 アプリケーションプログラムに関連する OpenTP1 のシステム定義 743
 - 付録 A.2 トランザクションの時間監視 745
 - 付録 A.3 サービス応答の待ち時間 753
 - 付録 A.4 常設コネクション問い合わせ間隔最大時間の適用範囲 754
 - 付録 A.5 アプリケーションプログラムが閉塞する場合 755
- 付録 B 共用メモリの見積もり式 758
 - 付録 B.1 TP1/Server Base 用の共用メモリの見積もり式 758
 - 付録 B.2 MCF サービス用の共用メモリの見積もり式 774
- 付録 C 監査イベントを取得する定義 779
- 付録 D 定義チェックの詳細 782
- 付録 E 旧製品からの移行に関する注意事項 899
 - 付録 E.1 バージョン 5 以前からの移行 899

索引 901

1

概要

OpenTP1 のシステム定義の体系，定義の構成，定義の作成手順，および定義の規則について説明します。

1.1 定義の体系

OpenTP1 の定義には、次の四つがあります。

1. システムサービス定義

TP1/Server Base に関する情報を定義します。

システムで動作するサービス群の構成や、システム共通の実行環境の定義です。

2. ネットワークコミュニケーション定義

TP1/Message Control に関する情報を定義します。

TP1/Message Control のメッセージ制御機能を使って他システムと通信する場合の、システム論理構成や、使用する業務プログラム、サービスなどの実行環境の定義です。

プロトコル部分の詳細については、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

なお、このマニュアルでは、以後メッセージ制御機能を **MCF** (Message Control Facility) と呼びます。

3. メッセージキュー定義

TP1/Message Queue に関する情報を定義します。

TP1/Message Queue のメッセージキューイング機能を使って、他システムと通信する場合の実行環境の定義です。

なお、このマニュアルでは、メッセージキュー定義の詳細 (MQT 通信構成定義) は説明しません。

MQT 通信構成定義については、マニュアル「TP1/Message Queue 使用の手引」を参照してください。

4. ネットワークライブラリ定義

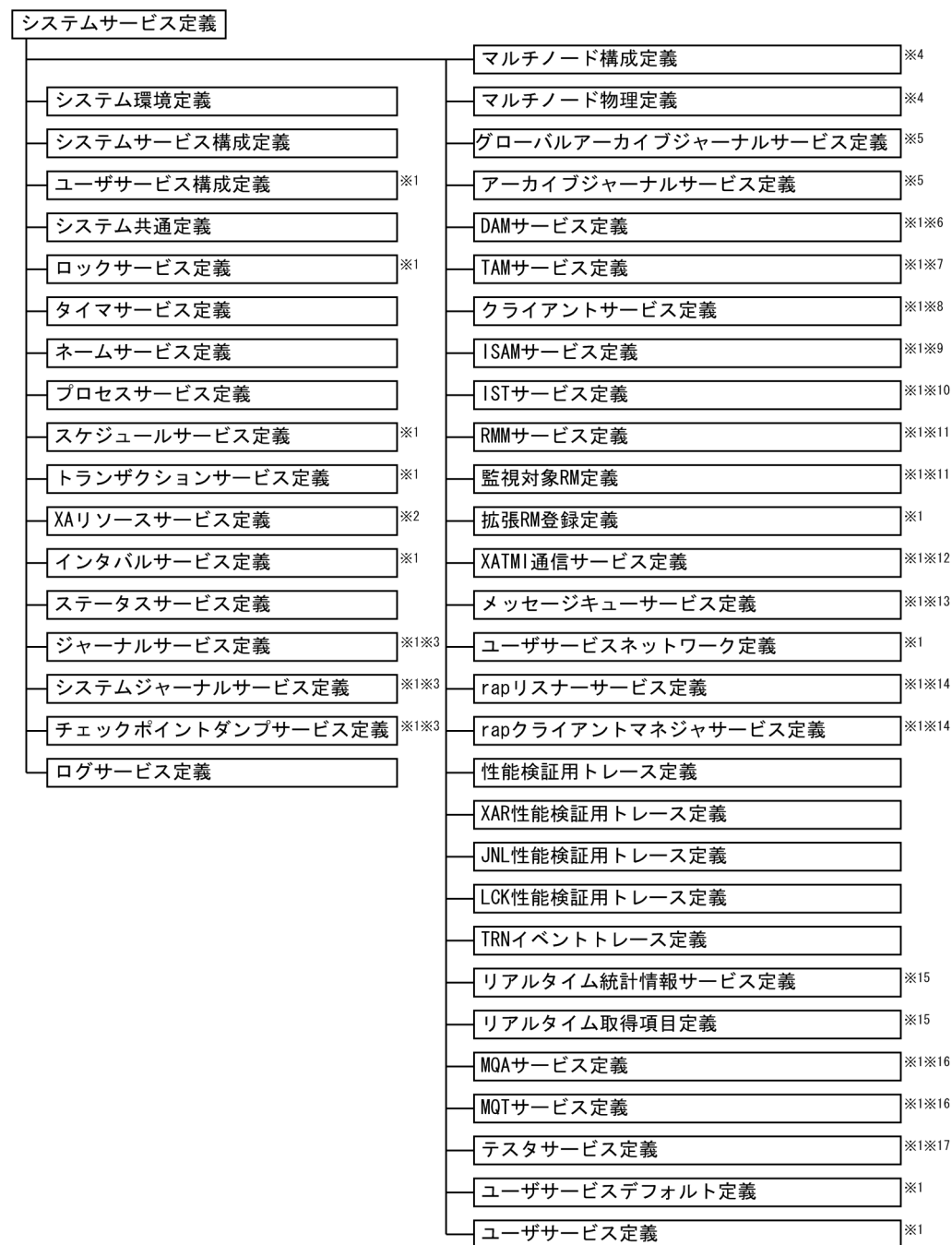
TP1/NET/Library に関する情報を定義します。

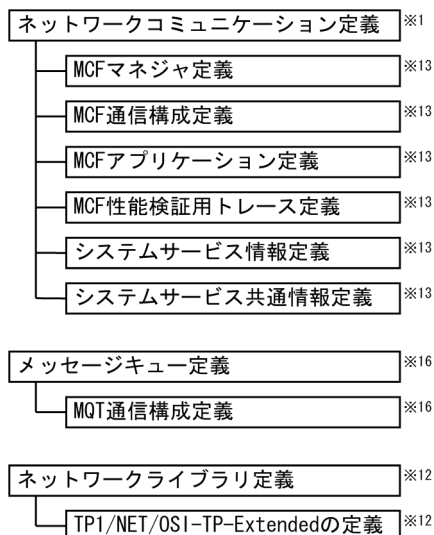
TP1/NET/OSI-TP-Extended を使って XATMI インタフェースで OSI TP 通信をする場合の実行環境の定義です。

なお、このマニュアルでは、ネットワークライブラリ定義の詳細 (TP1/NET/OSI-TP-Extended の定義) は説明しません。TP1/NET/OSI-TP-Extended の定義については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended 編」を参照してください。

OpenTP1 のシステム定義の体系を、次の図に示します。また、各定義の概要を以降の表に示します。

図 1-1 OpenTP1 システム定義の体系





- 注※1 グローバルアーカイブジャーナルサービスを使うOpenTP1ノードでは、定義しません。
- 注※2 XAリソースサービス機能を使う場合に定義します。
- 注※3 ジャーナルファイルレス機能を使用するOpenTP1ノードでは、定義しません。
- 注※4 TP1/Multiを使う場合に定義します。
- 注※5 グローバルアーカイブジャーナルサービスを使う場合に定義します。
- 注※6 TP1/FS/Direct Accessを使う場合に定義します。
- 注※7 TP1/FS/Table Accessを使う場合に定義します。
- 注※8 TP1/Client/W, またはTP1/Client/Pを使う場合に定義します。
- 注※9 ISAMファイルをトランザクション処理する機能 (ISAM/B) を使う場合に定義します。
定義の内容については、マニュアル「索引順編成ファイル管理 ISAM」を参照してください。
- 注※10 TP1/Shared Table Accessを使う場合に定義します。
- 注※11 TP1/Resource Manager Monitorを使う場合に定義します。
- 注※12 TP1/NET/OSI-TP-Extendedを使ったOSI TP通信をする場合に定義します。
定義の内容については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended編」を参照してください。
- 注※13 メッセージ送受信機能 (TP1/Message Control) を使う場合に定義します。
- 注※14 リモートAPI機能を使う場合に定義します。
- 注※15 リアルタイム統計情報サービスを使う場合に定義します。
- 注※16 メッセージキューイング機能 (TP1/Message Queue) を使う場合に定義します。
定義の内容については、マニュアル「TP1/Message Queue 使用の手引」を参照してください。
- 注※17 TP1/Online Testerを使う場合に定義します。定義の内容については、マニュアル「OpenTP1 テスタ・UAPトレース使用の手引」を参照してください。

表 1-1 システムサービス定義の概要

項番	定義名	内容
1	システム環境定義	OpenTP1 システムの実行環境を定義します。
2	システムサービス構成定義	同一マシンで一つの OpenTP1 システムとしてシステム起動時に一緒に開始するサービス群の構成を定義します。
3	ユーザサービス構成定義	同一マシンで一つの OpenTP1 システムとしてシステム起動時に一緒に開始するユーザサーバ群の構成を定義します。
4	システム共通定義	OpenTP1 システム共通の実行環境を定義します。
5	ロックサービス定義	OpenTP1 で排他制御機能を使用するための実行環境を定義します。
6	タイマサービス定義	時間監視サービスを行うための実行環境を定義します。
7	ネームサービス定義	RPC を可能にするサービス名とアドレスの対を管理するための実行環境を定義します。

項番	定義名	内容
8	プロセスサービス定義	OpenTP1 環境下で実行するサーバプロセスを管理するための実行環境を定義します。
9	スケジュールサービス定義	OpenTP1 環境下で動作するサーバへのサービス要求のスケジューリングをするための実行環境を定義します。
10	トランザクションサービス定義	トランザクション管理をするための実行環境を定義します。
11	XA リソースサービス定義	XA リソースサービスを使用してトランザクション管理をするための実行環境を定義します。
12	インタバルサービス定義	各システムサーバへ提供するインタバルタイマ機能の実行環境を定義します。
13	ステータスサービス定義	OpenTP1 環境下で動作する各システムサービスの状態を管理するための実行環境を定義します。
14	ジャーナルサービス定義	システムジャーナルサービスとチェックポイントダンプサービスを使用するための実行環境を定義します。
15	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にジャーナルを取得するための実行環境を定義します。
16	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にチェックポイントダンプを取得するための実行環境を定義します。
17	ログサービス定義	メッセージログの出力環境を定義します。
18	マルチノード構成定義	マルチノードエリアまたはマルチノードサブエリアの構成を定義します。
19	マルチノード物理定義	OpenTP1 ノードがあるホスト名とプロセスサーバデーモン (prcd) のポート番号を定義します。
20	グローバルアーカイブジャーナルサービス定義	グローバルアーカイブジャーナルサービスを使用するための実行環境を定義します。
21	アーカイブジャーナルサービス定義	アーカイブジャーナルファイルのファイルグループ名と属性を定義します。
22	DAM サービス定義	ファイル回復機能付き直接アクセスファイルを管理するための実行環境を定義します。
23	TAM サービス定義	TAM テーブルを管理するための実行環境を定義します。
24	クライアントサービス定義	OpenTP1 のクライアント機能を支援するためのサーバの実行環境を定義します。
25	IST サービス定義	IST サービスを使用するための実行環境を定義します。
26	RMM サービス定義	RMM サービスを使用するリソースマネージャを定義します。
27	監視対象 RM 定義	RMM サービスのコマンドを定義します。
28	拡張 RM 登録定義	OpenTP1 提供以外のリソースマネージャを登録するための実行環境を定義します。
29	XATMI 通信サービス定義	TP1/NET/OSI-TP-Extended を使った XATMI インタフェースでの OSI TP 通信の実行環境を定義します。

項番	定義名	内容
30	メッセージキューサービス定義	入出力メッセージのキューイングを管理するための実行環境を定義します。
31	ユーザサービスネットワーク定義	通信相手のサービス情報を定義します。
32	rap リスナーサービス定義	リモート API 機能を使用する場合の rap リスナーを定義します。
33	rap クライアントマネージャサービス定義	リモート API クライアントマネージャ機能を使用するための実行環境を定義します。
34	性能検証用トレース定義	性能検証用トレース情報を取得する場合のファイルを定義します。
35	XAR 性能検証用トレース定義	XAR 性能検証用トレース情報を取得する場合の実行環境を定義します。
36	JNL 性能検証用トレース定義	JNL 性能検証用トレース情報を取得する場合の実行環境を定義します。
37	LCK 性能検証用トレース定義	LCK 性能検証用トレース情報を取得する場合の実行環境を定義します。
38	TRN イベントトレース定義	TRN イベントトレース情報を取得場合の実行環境を定義します。
39	リアルタイム統計情報サービス定義	リアルタイム統計情報サービスの実行環境を定義します。
40	リアルタイム取得項目定義	リアルタイム統計情報を取得する項目を定義します。
41	MQA サービス定義	TP1/Message Queue のキューを管理する MQA サーバを定義します。
42	MQT サービス定義	TP1/Message Queue の通信を制御する MQT サーバを定義します。
43	テストサービス定義	オンライン上で UAP をテストするための実行環境を定義します。
44	ユーザサービスデフォルト定義	ユーザサービス定義の省略時解釈値を定義します。
45	ユーザサービス定義	ユーザサーバの実行環境を定義します。

表 1-2 ネットワークコミュニケーション定義の概要

項番	定義名	内容
1	MCF マネージャ定義	各 MCF 間で共用するリソースの実行環境を定義します。
2	MCF 通信構成定義	個々の MCF の接続端末と、送受信メッセージの入出力キューなどの実行環境を定義します。
3	MCF アプリケーション定義	ユーザアプリケーション名および MCF イベント名に対応するサービス名などの実行環境を定義します。
4	MCF 性能検証用トレース定義	MCF 性能検証用トレース情報を取得場合の実行環境を定義します。
5	システムサービス情報定義	MCF 通信サービスを起動するための環境を定義します。
6	システムサービス共通情報定義	複数の MCF 通信サービスに共通する情報を定義します。

表 1-3 メッセージキュー定義の概要

項番	定義名	内容
1	MQT 通信構成定義	TP1/Message Queue のメッセージキューイング形態で通信する MQT サーバを定義します。

表 1-4 ネットワークライブラリ定義の概要

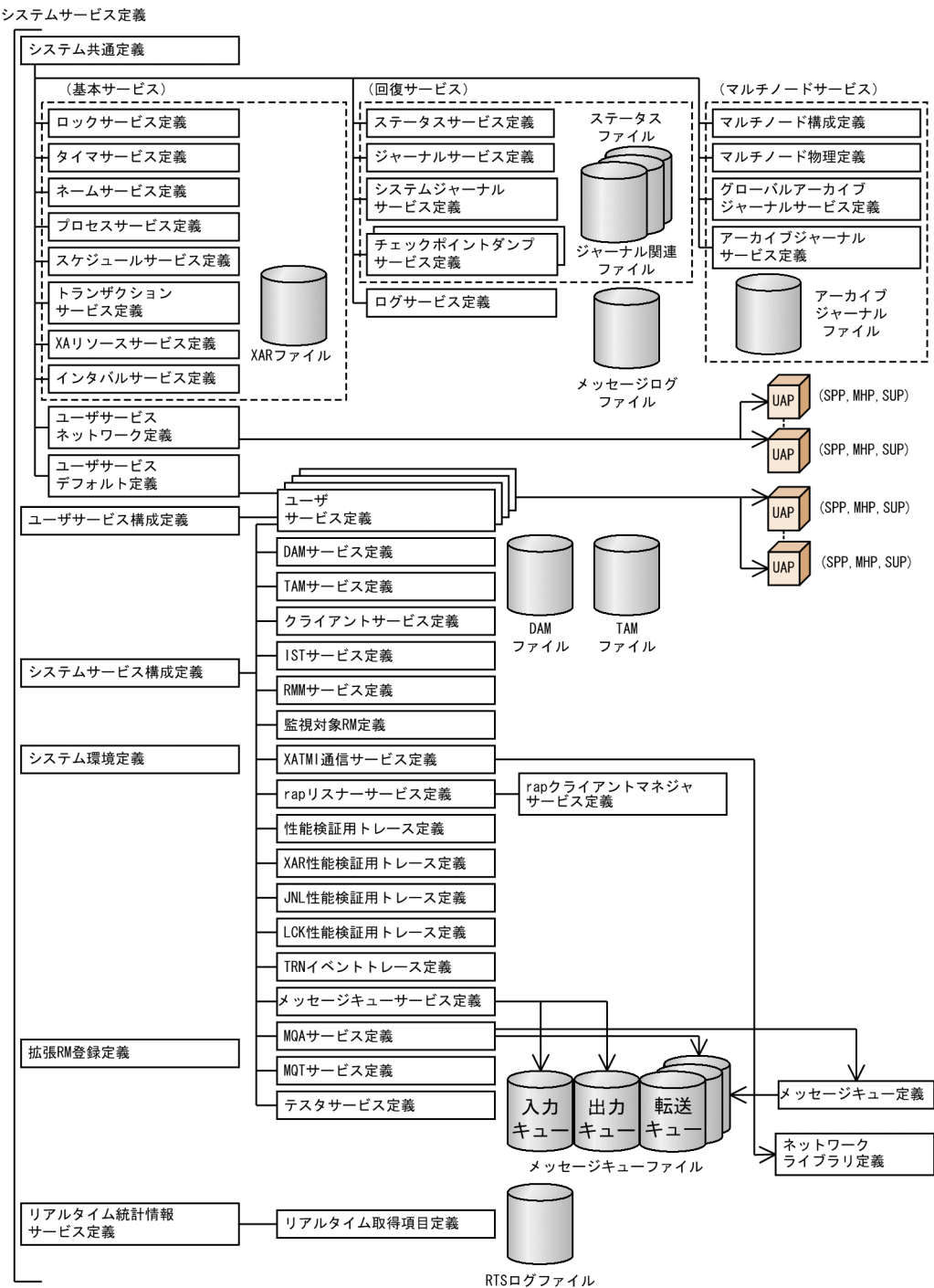
項番	定義名	内容
1	TP1/NET/OSI-TP-Extended の定義	TP1/NET/OSI-TP-Extended の実行環境を定義します。

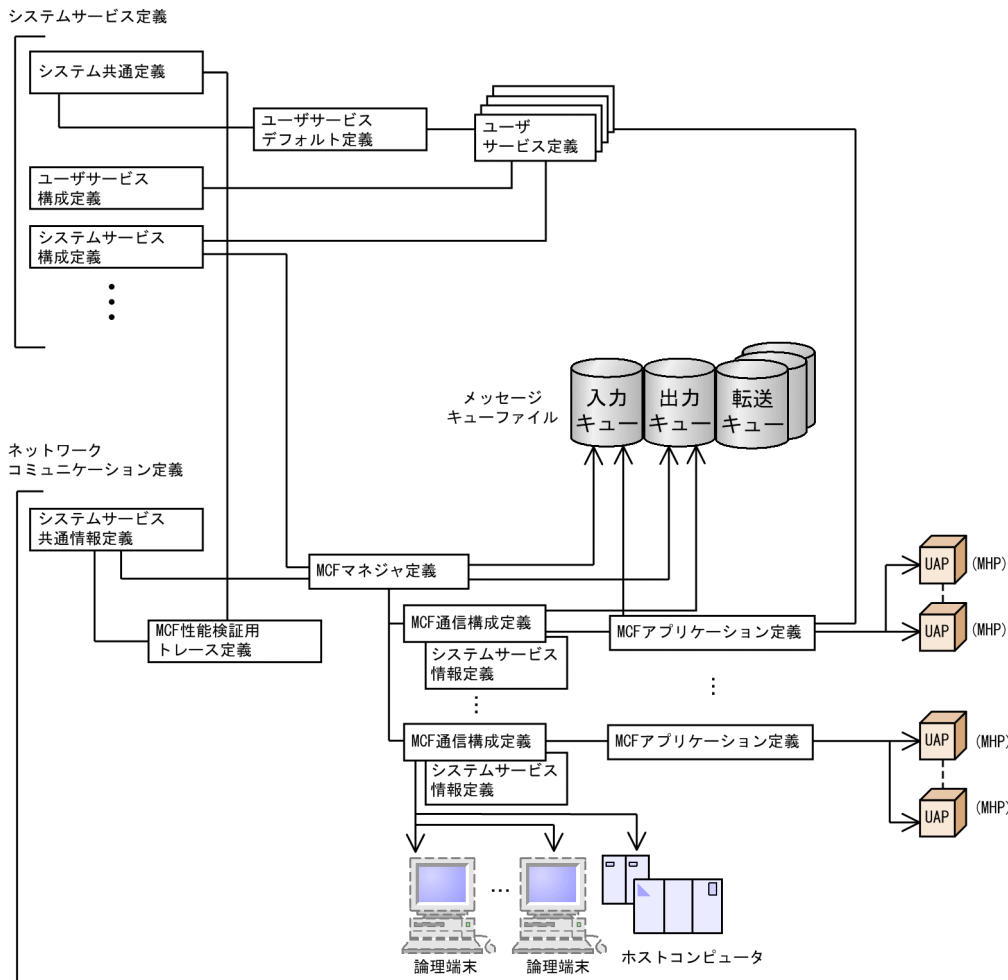
1.2 定義の構成

OpenTP1 システム定義では、OpenTP1 システムの動作する環境をマシンごとに定義します。個々のマシンで、そのマシンの動作環境を定義します。

OpenTP1 システム定義の構成を次の図に示します。

図 1-2 OpenTP1 システム定義の構成





1.2.1 システムサービス定義の構成

システムサービス定義には、OpenTP1 システム全体の構成と実行環境を定義するものとして、システム環境定義、システムサービス構成定義、ユーザサービス構成定義などがあります。

また、OpenTP1 システム全体に共通な項目で、そのノード内のシステムサービス、およびユーザサーバの実行環境などを定義するシステム共通定義とユーザサービスデフォルト定義があります。

そのほかの定義は、個々のシステムサービスごとに、そのシステムサービスが使用するリソースや実行オプションを記述するものです。OpenTP1 システムの各システムサービスは、独立したプロセス（空間）として動作します。ジャーナルサービス、スケジュールサービスなど、OpenTP1 のシステムサービスに対応した実行環境を定義するのが各システムサービス定義です。

マルチノードサービスを使用する場合、各ノードでグローバルアーカイブジャーナルサービスの有無によって、定義する必要があるものとないものがあります。グローバルアーカイブジャーナルサービスと各定義の関係を次の表に示します。

表 1-5 グローバルアーカイブジャーナルサービスと各定義の関係

項番	定義名	グローバルジャーナルサービスの ないノード	グローバルジャーナルサービスの あるノード
1	システム環境定義	○	△
2	システムサービス構成定義	○	○
3	ユーザサービス構成定義	○	×
4	システム共通定義	○	○
5	ロックサービス定義	△	×
6	タイマサービス定義	△	△
7	ネームサービス定義	△	△
8	プロセスサービス定義	○	△
9	スケジュールサービス定義	△	×
10	トランザクションサービス定義	○	×
11	XA リソースサービス定義	△	×
12	インタバルサービス定義	△	×
13	ステータスサービス定義	○	○
14	ジャーナルサービス定義	○	×
15	システムジャーナルサービス定義	○	×
16	チェックポイントダンプサービス定義	○	×
17	ログサービス定義	○	○
18	マルチノード構成定義	○※	○※
19	マルチノード物理定義	○※	○※
20	グローバルアーカイブジャーナルサービス定義	×	○
21	アーカイブジャーナルサービス定義	×	○
22	DAM サービス定義	○※	×
23	TAM サービス定義	○※	×
24	クライアントサービス定義	○※	×
25	IST サービス定義	○※	×
26	RMM サービス定義	○※	×
27	監視対象 RM 定義	○※	×
28	拡張 RM 登録定義	△	×

項番	定義名	グローバルジャーナルサービスのないノード	グローバルジャーナルサービスのあるノード
29	XATMI 通信サービス定義	○※	×
30	メッセージキューサービス定義	△	×
31	ユーザサービスネットワーク定義	△	×
32	rap リスナーサービス定義	○	×
33	rap クライアントマネージャサービス定義	○	×
34	性能検証用トレース定義	△	△
35	XAR 性能検証用トレース定義	△	△
36	JNL 性能検証用トレース定義	△	△
37	LCK 性能検証用トレース定義	△	△
38	TRN イベントトレース定義	△	△
39	リアルタイム統計情報サービス定義	△	×
40	リアルタイム取得項目定義	△	×
41	MQA サービス定義	○※	×
42	MQT サービス定義	○※	×
43	テストサービス定義	○※	×
44	ユーザサービスデフォルト定義	○	×
45	ユーザサービス定義	○	×

(凡例)

○：定義ファイル必要。

△：定義ファイル不要。

・システム内部は省略時解釈値で動作します。

・定義ファイルがある場合は、それに従って動作します。

×：定義ファイル不要。定義ファイルがあっても無視します。

注※

該当するプログラムプロダクトを使用している場合は、必要です。

ユーザサーバでも同様に、ユーザサーバごとに使用するリソースや実行オプションを定義します。MCF を使用する場合は、一つのサービスに一つのアプリケーションが対応します。ユーザサーバは複数のサービスを提供できます。このサービスの集合体をサービスグループといい、一つのサービスグループは一つのユーザサーバに対応します。このサービスグループの実行環境を定義するのがユーザサービス定義です。

1.2.2 ネットワークコミュニケーション定義の構成

ネットワークコミュニケーション定義は六つの定義から構成され、次の用途で作成します。

1. 概要

- MCF マネジャ定義…MCF を管理・運用するための環境を定義します。
- MCF 通信構成定義…個々の通信サービスの環境を定義します。
- MCF アプリケーション定義…アプリケーションの属性を定義します。
- MCF 性能検証用トレース定義…MCF 性能検証用トレースを取得するための環境を定義します。
- システムサービス情報定義…MCF 通信サービスを起動するための環境を定義します。
- システムサービス共通情報定義…複数の MCF 通信サービスに共通する情報を定義します。

プロトコル部分の詳細については、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

1.2.3 メッセージキュー定義の構成

メッセージキュー定義は、MQT 通信構成定義だけで構成されています。MQT 通信構成定義については、マニュアル「TP1/Message Queue 使用の手引」を参照してください。

1.2.4 ネットワークライブラリ定義の構成

ネットワークライブラリ定義は、TP1/NET/OSI-TP-Extended の定義だけで構成されています。TP1/NET/OSI-TP-Extended の定義については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended 編」を参照してください。

1.3 定義の作成手順

システムサービス定義およびネットワークコミュニケーション定義を作成する場合は、テキストエディタを使用して定義ファイルを作成します。

ネットワークコミュニケーション定義の場合は、オンライン開始前にオブジェクトファイルを生成しておく必要があります。オブジェクトファイルは、テキストエディタによって作成したソースファイルを、ユーティリティで変換して生成します。

MCF マネージャ定義、MCF 通信構成定義、および MCF アプリケーション定義は、個別のソースファイルに作成し、個別のオブジェクトファイルを生成します。ただし、MCF 通信構成定義は共通定義とデータコミュニケーション定義から構成されているため、ソースファイルが二つ必要です。

なお、ネットワークコミュニケーション定義で、プロトコル部分の詳細については、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

メッセージキュー定義（MQT 通信構成定義）の作成手順については、マニュアル「TP1/Message Queue 使用の手引」を参照してください。

ネットワークライブラリ定義（TP1/NET/OSI-TP-Extended の定義）の作成手順については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended 編」を参照してください。

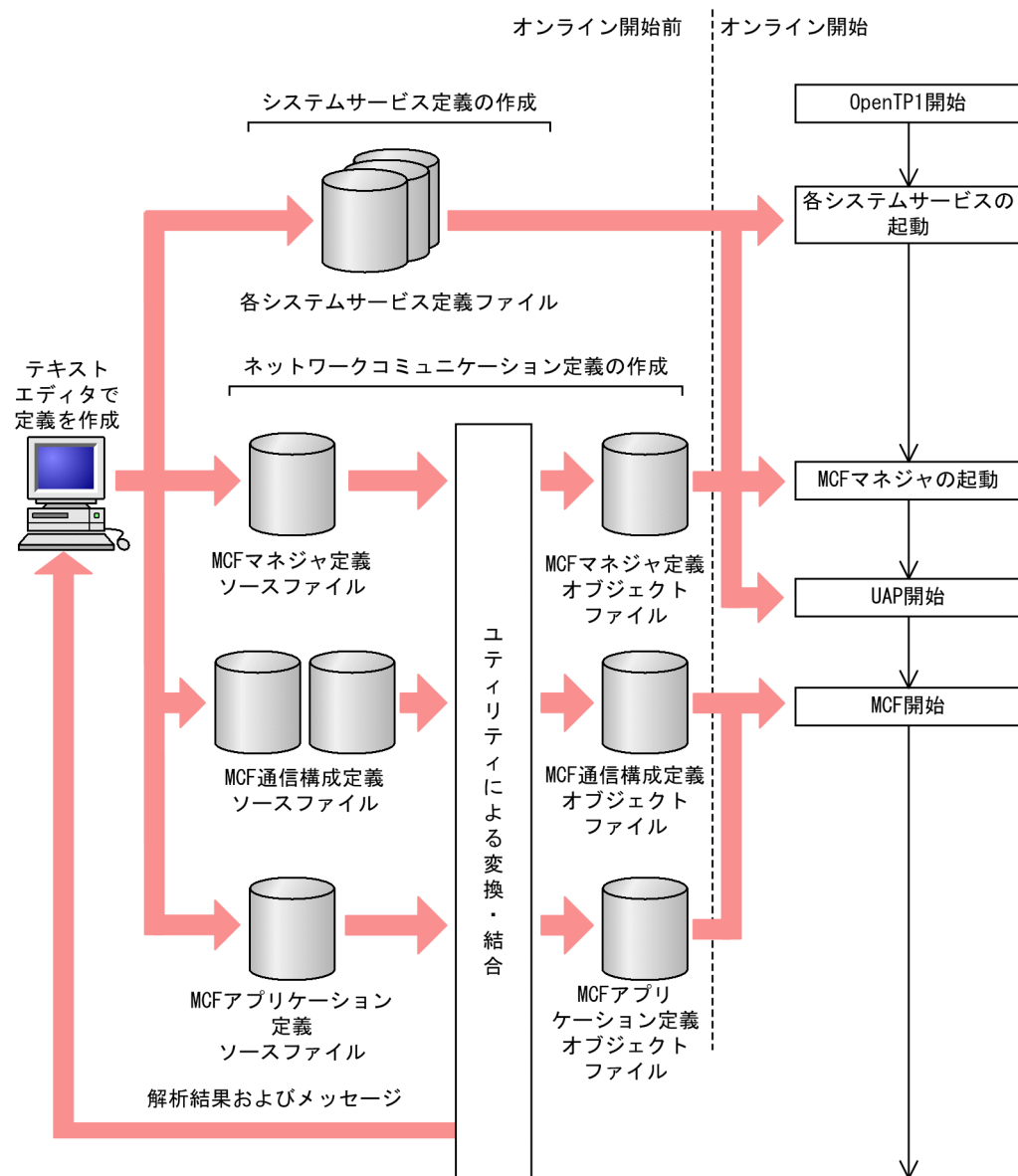
注

\$DCDIR/lib/sysconf 以下の定義ファイルは、OpenTP1 のシステムサービスで使用する定義ファイルです。このため、\$DCDIR/lib/sysconf 以下のファイルをコピーして定義を作成しないでください。コピーして作成した場合、OpenTP1 が正常に動作しなくなるおそれがあります。

定義の作成手順とオンライン開始時の処理の流れを次の図に示します。

なお、定義を作成してから、OpenTP1 システムを OS へ登録してください。

図 1-3 定義の作成手順とオンライン開始時の処理の流れ



1.4 定義の規則

定義の基本事項と記述形式について説明します。

1.4.1 基本事項

(1) システムサービス定義について

システムサービス定義の、定義間の記述順序は任意です。

システムサービス定義には set 形式、コマンド形式、putenv 形式、および dcputenv 形式があります。

どの定義形式も同一定義ファイルに格納します。

同一定義内では、set 形式を定義してからコマンド形式、putenv 形式、または dcputenv 形式の順に定義してください。

set 形式では OpenTP1 システム開始時に値が決定します。

コマンド形式では OpenTP1 システム開始時に値が決定しても、オンライン中に運用コマンドを入力すると OpenTP1 システム開始時に決定した値を変更できます。

putenv 形式、dcputenv 形式では環境変数を設定できます。

なお、システムサービス定義のコマンドは定義として記載しています。

このため、実際の運用コマンドとオプションや使用上の制限が異なります。

運用コマンドとして使用する場合は、マニュアル「OpenTP1 運用と操作」で利用できるコマンドかどうかを確認してください。

(2) ネットワークコミュニケーション定義について

ネットワークコミュニケーション定義では、各定義間の記述順序は任意ですが、各定義内での記述順序が決まっています。詳しくは、「[5. ネットワークコミュニケーション定義の詳細](#)」を参照してください。

なお、プロトコル部分の詳細については、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

(3) メッセージキュー定義について

メッセージキュー定義（MQT 通信構成定義）の規則については、マニュアル「TP1/Message Queue 使用の手引」を参照してください。

(4) ネットワークライブラリ定義について

ネットワークライブラリ定義（TP1/NET/OSI-TP-Extended の定義）の規則については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended 編」を参照してください。

(5) 定義情報のメンテナンスについて

定義情報はテキストエディタでメンテナンスできます。

ただし、オンラインで使用中の定義を変更または削除しないでください。

定義を変更または削除した場合、そのオンラインの動作は保証できません。

定義を変更する手順については、「[7.1 定義の変更手順](#)」を参照してください。

(6) 同一のオペランドが複数のシステム定義に存在する場合

システム共通定義、ユーザサービスデフォルト定義、ユーザサービス定義に同一のオペランドを指定できる場合の優先順位を次に示します（1.> 2.> 3.）。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

1.4.2 記述形式

(1) set 形式

set 形式で指定する定義をオペランドと呼びます。set 形式では、オペランドに値を設定します。

形式

set オペランド=値

(2) コマンド形式

コマンド形式で指定する定義を定義コマンドと呼びます。コマンド形式では、次の形式に従ってオプションおよび引数を設定します。

形式

コマンド名 オプション コマンド引数

用語の定義

コマンド名：コマンド名称です。

オプション：マイナス記号で始まる文字列です。

引数を取らない形式 1 と引数を必要とする形式 2 の二つがあります。

形式 1 -オプションフラグ

形式 2 -オプションフラグ フラグ引数

(凡例)

オプションフラグ：マイナス記号に続く 1 文字の英数字です。英大文字と英小文字とは区別します。

フラグ引数：オプションフラグに対する文字列です。

コマンド引数：マイナス記号以外で始まる引数でコマンドの操作対象です。

(3) putenv 形式

putenv 形式ではユーザの環境変数を設定します。環境変数値に環境変数名を設定した場合、その環境変数は展開しません。

形式

putenv 環境変数名 環境変数値

(4) dcputenv 形式

dcputenv 形式ではユーザの環境変数を設定します。環境変数値に環境変数名を設定した場合、その環境変数を展開した値で、ユーザの環境変数を設定します。

形式

dcputenv 環境変数名 環境変数値

(5) コメント

コメントを記入する場合は、コメントの先頭に「#」を記述してください。

行の先頭に「#」を記述すると 1 行全体がコメント扱いになります。

形式 1 定義の記述 #コメント

形式 2 #コメント

(6) 制限事項

複数の項目をコンマで区切る定義があります。このような定義を記述する場合、項目の直後にコンマを記述してください。項目とコンマの間に空白を入れないでください。

定義の 1 行の最大長は次のとおりです。

- ・ システムサービス定義：定義 1 行の長さはコメントを含め最大 80 バイトです。

- ・ネットワークコミュニケーション定義※：定義 1 行の長さはコメントを含め最大 1024 バイトです。

注※

システムサービス情報定義、またはシステムサービス共通情報定義の場合はコメントを含め最大 80 バイトです。

1 行の最大長を超えて定義を指定すると、OpenTP1 の開始に失敗したり、エラーメッセージを出力したりする場合があります。

定義が 1 行の最大長を超える場合は、複数行に分けて記述してください。このとき、行の末尾に継続符号'¥'を記述します。継続符号'¥'の後ろには何も記述しないでください。また、複数の項目をコンマで区切りながら複数行に分けて記述する場合、継続符号'¥'を記述する行にある項目の末尾にコンマを記述してください。

ネットワークコミュニケーション定義で、引用符 (") を使って囲むオペランドを指定するオプションに、次の行にほかのオプションを記述する場合、引用符 (") と継続符号'¥'の間に一つ以上の空白を入れてください。次に指定例を示します (△は半角スペースを示します)。

```
-y "mode=client"△¥  
-r "portno=10002"
```

なお、コメントを記述した行に継続行を続ける場合の制限事項は、次のとおりです。

- ・システムサービス定義の場合
コメントを記述した行に継続行を続けることはできません。'#'を記述すると、その行の終わりまでがコメントと見なされるため、'¥'を記述しても行の継続はできなくなります。
- ・ネットワークコミュニケーション定義の場合
コメントを記述した行に継続行を続ける場合は、コメントを'#'で囲んだあと、その行の末尾に'¥'を記述してください。

(7) 環境変数を設定する場合の注意事項

dcputenv 形式では、環境変数値に設定した環境変数名も getenv 関数によって展開します。展開する環境変数名の先頭には'\$'を記述してください。環境変数名として指定できるのは、英数字、およびアンダスコアです。

環境変数値に環境変数名を指定する場合の注意事項

- ・展開した環境変数値にさらに環境変数名がある場合、その環境変数名については展開しません。
- ・環境変数名を展開しないで文字列と見なす場合は、環境変数値全体をダブルクォーテーションで囲ってください。
- ・環境変数値の途中でダブルクォーテーションを指定した場合、そのダブルクォーテーションは、環境変数値の一部として扱います。

(例) 環境変数\$DCDIR にディレクトリ/OpenTP1 が設定されている場合

```
putenv TMPPATH $DCDIR/tmp          ...1
dcputenv FILEPATH $DCDIR/tmp       ...2
dcputenv TMPDIR $TMPPATH/file      ...3
dcputenv REALPATH $FILEPATH:$DCDIR/file ...4
dcputenv ENVPATH "$TMPPATH/file"   ...5
dcputenv DIRTMP $DCDIR:"$FILEPATH":/tmp ...6
```

- 1. TMPPATH に"\$DCDIR/tmp"という文字列が設定されます (\$DCDIR は展開しません)。
- 2. FILEPATH に"/OpenTP1/tmp"が設定されます (\$DCDIR を展開し, "/OpenTP1"とします)。
- 3. TMPDIR に"\$DCDIR/tmp/file"が設定されます (TMPPATH は'\$DCDIR/tmp'に展開されますが, \$DCDIR は展開されません)。
- 4. REALPATH に"/OpenTP1/tmp:/OpenTP1/file"が設定されます (\$FILEPATH:と指定されている場合, コロンの直前までの文字列が環境変数名として認識されます)。
- 5. ENVPATH に"\$TMPPATH/file"が設定されます (\$TMPPATH は展開されません)。
- 6. DIRTMP に'/OpenTP1:/OpenTP1/tmp':/tmp'が設定されます (ダブルクォーテーションを環境変数値の一部として扱うため, ダブルクォーテーションも環境変数値として設定されます。さらに, 環境変数名を展開しないためのダブルクォーテーションとしては扱わないため, \$FILEPATH を展開します)。

同じ環境変数を putenv 形式と dcputenv 形式の両方で設定している場合は, あとに設定したものが有効になります。

1.4.3 記号の説明

定義の説明に使用する各種の記号を説明します。

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

(1) 文法記述記号

オプションおよびコマンド引数を説明する記号です。

各記号の意味を次の表に示します。

表 1-6 文法記述記号一覧

文法記述記号	意味
{ }	この記号で囲まれている複数の項目のうちから一つを選択することを示します。項目の区切りは で示します。 (例) {-m マルチエリアノード識別子 -g マルチノードサブエリア識別子} これは, -m マルチエリアノード識別子と-g マルチノードサブエリア識別子の二つの項目のうち, どちらかを指定することを示します。

文法記述記号	意味
[]	この記号で囲まれている項目は省略してもよいことを示します。 (例) [prcsvpath パス名] これは prcsvpath パス名と指定するか、何も指定しないことを示します。 (例) [-c cid] これは何も指定しないか、または -c オプションで指定された項目は cid という値を取ることを示します。
{ { } }	この記号で囲まれた複数の項目が一つの繰り返し項目の単位であることを示します。 (例) {{dcsvstart -u サーバ名 [, サーバ名] ...}} これは、次のように指定できることを示します。 dcsvstart -u サーバ名 [, サーバ名] ... dcsvstart -u サーバ名 [, サーバ名] ...
	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) set hold = Y N これは、Y か N を指定できることを示します。
— (下線)	この記号で示す項目は、該当オペランド、オプションまたはコマンド引数を省略した場合の省略時解釈値を示します。 (例) set type = <u>other</u> MHP type オペランドを省略した場合、other を省略時解釈値とすることを示します。
...	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) dcsvstart -u サーバ名 [, サーバ名] ... これは、dcsvstart で指定するサービスグループ名を複数個指定できることを示します。 (例) -l "le1 le2...len" これは、-l オプションの中の指定を空白で区切った文字列を繰り返すことを示します。

(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。

各記号の意味を次の表に示します。

表 1-7 属性表示記号一覧

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

(3) 構文要素記号

ユーザ指定値の内容を説明する記号です。

各記号の意味を次の表に示します。

表 1-8 構文要素記号一覧

構文要素記号	意味
英字	アルファベット (A～Z, a～z) と_ (アンダスコア)
英字記号	英字と#, @, ¥, \$, %
英数字	英字と数字 (0～9)
英数字記号	英字記号と数字 (0～9)
符号なし整数	数字列 (0～9)
16 進数	数字 (0～9) と (A～F, 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の英数字列
記号名称	先頭が英字記号の英数字記号列
文字列	英数字記号の配列
パス名	/ と記号名称の並び

使用上の注意

- すべて半角文字を使用してください。
- パス名は使用している OS に依存します。

2

システムサービス定義の概要

システムサービス定義の概要と定義の種類について説明します。

2.1 概要

2.1.1 システムサービス定義とファイル名

システムサービス定義を登録するファイル名とファイル数を、次の表に示します。

なお、次の表で示すノードとは、OpenTP1 システムを指します。

表 2-1 システムサービス定義を登録するファイル名とファイル数

定義名	定義ファイル名（絶対パス名）	定義ファイル数（1 ノード当たり）
システム環境定義	\$DCDIR/conf/env	0～1
システムサービス構成定義	\$DCCONFPATH/sysconf	1
ユーザサービス構成定義	\$DCCONFPATH/usrconf	1
システム共通定義	\$DCCONFPATH/betranrc	1
ロックサービス定義	\$DCCONFPATH/lck	0～1
タイマサービス定義	\$DCCONFPATH/tim	0～1
ネームサービス定義	\$DCCONFPATH/nam	0～1
プロセスサービス定義	\$DCCONFPATH/prc	0～1
スケジュールサービス定義	\$DCCONFPATH/scd	0～1
	\$DCCONFPATH/dcsdmlt※ ¹	0～1
	\$DCCONFPATH/マルチスケジューラグループ名※ ¹	0～マルチスケジューラグループ数
トランザクションサービス定義	\$DCCONFPATH/trn	1
XA リソースサービス定義	\$DCCONFPATH/xar	0～1
インタバルサービス定義	\$DCCONFPATH/itv	0～1
ステータスサービス定義	\$DCCONFPATH/sts	1
ジャーナルサービス定義	\$DCCONFPATH/jnl	0～1※ ²
システムジャーナルサービス定義	\$DCCONFPATH/任意※ ³	0～1※ ²
チェックポイントダンプサービス定義	\$DCCONFPATH/任意※ ³	0～チェックポイントダンプを取得するシステムサービス数※ ² ※ ⁴
ログサービス定義	\$DCCONFPATH/log	0～1
マルチノード構成定義	\$DCCONFPATH/nodeconf	0～1
マルチノード物理定義	\$DCCONFPATH/nodeaddr	0～1

定義名	定義ファイル名（絶対パス名）	定義ファイル数（1 ノード当たり）
グローバルアーカイブジャーナルサービス定義	\$DCCONFPATH/jar	0～1
アーカイブジャーナルサービス定義	\$DCCONFPATH/任意 ^{※5}	1～16 ^{※6}
DAM サービス定義	\$DCCONFPATH/dam	0～1
TAM サービス定義	\$DCCONFPATH/tam	0～1
クライアントサービス定義	\$DCCONFPATH/cltsrv	0～1 ^{※7}
IST サービス定義	\$DCCONFPATH/ist	0～1
RMM サービス定義	\$DCCONFPATH/rmm	0～1
監視対象 RM 定義	\$DCCONFPATH/任意 ^{※8}	0～1
拡張 RM 登録定義	\$DCDIR/conf/extendRM	0～1
XATMI 通信サービス定義	\$DCCONFPATH/_xatc	0～1
メッセージキューサービス定義	\$DCCONFPATH/que	0～1
ユーザサービスネットワーク定義	\$DCCONFPATH/usrnet	0～1
rap リスナーサービス定義	任意 ^{※9}	rap リスナー数
rap クライアントマネージャサービス定義	任意 ^{※9}	0～1
性能検証用トレース定義	\$DCCONFPATH/prf	0～1
XAR 性能検証用トレース定義	\$DCCONFPATH/_xr	0～1
JNL 性能検証用トレース定義	\$DCCONFPATH/_jl	0～1
LCK 性能検証用トレース定義	\$DCCONFPATH/_lk	0～1
TRN イベントトレース定義	\$DCCONFPATH/_tr	0～1
リアルタイム統計情報サービス定義	\$DCCONFPATH/rts	0～1
リアルタイム取得項目定義	\$DCCONFPATH/任意 ^{※10}	rtspat 定義コマンドの指定数 ^{※11}
MQA サービス定義 ^{※12}	\$DCCONFPATH/mqa	0～1
MQT サービス定義 ^{※12}	\$DCCONFPATH/mqt	0～1
テストサービス定義 ^{※13}	\$DCCONFPATH/uto	0～1
ユーザサービスデフォルト定義	\$DCCONFPATH/usrrc	0～1
ユーザサービス定義	\$DCCONFPATH/ユーザサーバ名 ^{※14}	ユーザサーバ数 ^{※15}

2. システムサービス定義の概要

注※1

マルチスケジューラ機能を使用する場合、マルチスケジューラデーモン独自に動作環境に関する定義を指定する必要があるときだけ、この定義ファイルを作成します。

dcscdmlt

このファイル名で示す定義ファイルを、すべてのマルチスケジューラデーモンが解析します。

ただし、マルチスケジューラグループ名の定義ファイルが存在する場合は、その定義内容を優先します。

マルチスケジューラグループ名

このファイル名で示す定義ファイルを、ファイル名と同一のマルチスケジューラグループに属するマルチスケジューラデーモンが解析します。

スケジューラサービス定義の `scdmulti` 定義コマンドの `-g` オプションで指定した名称です。

`dcscdmlt` で示す定義ファイルに同じ定義を指定した場合、マルチスケジューラグループ名で示す定義ファイルの指定値が有効になります。

注※2

システム共通定義の `jnl_fileless_option` オペランドに `Y` を指定している場合、定義ファイルを省略できます。

注※3

ノード内で一意になる名称を指定してください。

ここで指定したファイル名を、ジャーナルサービス定義 (`jnldfs`) に定義します。

なお、ほかの定義ファイルで使用しているファイル名は指定できません。

注※4

対象サービスごとに作成してください。

注※5

ノード内で一意になる名称を指定してください。

ここで指定したファイル名を、グローバルアーカイブジャーナルサービス定義 (`jnldfs`) に定義します。

なお、ほかの定義ファイルで使用しているファイル名は指定できません。

注※6

アーカイブジャーナルサービス定義を指定した場合にだけ作成してください。

注※7

0 の場合、警告メッセージが出力されますが、省略値で動作します。

注※8

同一マシン内で一意になる名称を指定してください。

ここで指定したファイル名を、RMM サービス定義 (`rmm_check_services`) に指定します。

なお、ほかの定義ファイルで使用しているファイル名は指定できません。

注※9

`$DCCONFPATH` 以外の名称を指定してください。

注※10

リアルタイム統計情報サービス定義の rtspat 定義コマンドで-f オプションに指定した名称です。

注※11

リアルタイム統計情報サービス定義の rtspat 定義コマンドで-f オプションに指定した定義ファイルの数だけ作成してください。

注※12

MQA サービス定義と MQT サービス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。

注※13

テストサービス定義の詳細については、マニュアル「OpenTP1 テスタ・UAP トレース使用の手引」を参照してください。

注※14

システム開始時に一緒に起動するユーザサーバの場合は、ユーザサービス構成定義（dcsvstart 定義コマンド）で指定するユーザサーバ名と同じ名称を指定してください。

必要に応じて起動するユーザサーバの場合は、サーバ起動コマンド（dcsvstart）で指定するユーザサーバ名と同じ名称を指定してください。

なお、ほかの定義ファイルで使用しているファイル名は指定できません。

注※15

ユーザサーバごとに作成してください。

2.2 定義の種類

2.2.1 システム環境定義の内容

システム環境定義の内容を、次の表に示します。

表 2-2 システム環境定義の内容

形式	オペランド／パス	定義内容	指定値
set	mode_conf	システムの開始方法	AUTO MANUAL1 《MANUAL2》
	static_shmpool_size	静的共用メモリの総量	〈符号なし整数〉 ((0~1945600)) 《4096》 (単位：キロバイト)
	dynamic_shmpool_size	動的共用メモリの最大使用時の総量	〈符号なし整数〉 ((0~1945600)) 《4096》 (単位：キロバイト)
	shmpool_attribute	共用メモリプールのメモリ固定の 要否	《free》 fixed
	user_command	ユーザ環境設定コマンド	〈パス名〉
	server_count	最大サーバ数	〈符号なし整数〉 ((32~9999)) 《64》
	user_server_ha	システムサーバ起動後、ユーザサー バを起動するかどうかを指定	Y 《N》
	system_terminate_watch_time	システム終了監視時間	〈符号なし整数〉 ((0~65535)) 《3600》 (単位：秒)
	start_scheduling_timing	システム開始時に RPC の受け付け の開始時期	BEFORE 《AFTER》
	system_init_watch_time	システム初期化の待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	user_command_online	システム開始完了コマンド	〈パス名〉
	preend_warning_watch_time	終了準備警告監視時間	〈符号なし整数〉 ((0~65535)) 《180》 (単位：秒)
	user_command_online_tplmng_r id	user_command_online に指定した システム開始完了コマンドを	Y 《N》

形式	オペランド／パス	定義内容	指定値
set	user_command_online_tplmngnrid	OpenTP1 管理者の UID/GID で実行するかどうか	Y 《N》
	default_value_option	システム定義のデフォルト値変更オプション	((0~1)) 《1》
putenv	DCCONFPATH	定義ファイルの格納ディレクトリ	〈パス名〉 《\$DCDIR/conf》
	DCADMDEBUG	デバッグ情報を取得するかどうか	《0》 1
	DCUAPCONFPATH	ユーザサービス定義ファイルまたはユーザサービスデフォルト定義ファイルを DCCONFPATH とは別のディレクトリに格納したい場合に格納するディレクトリ	〈パス名〉
dcputenv	DCCONFPATH	定義ファイルの格納ディレクトリ	〈パス名〉
	DCUAPCONFPATH	ユーザサービス定義ファイルまたはユーザサービスデフォルト定義ファイルを DCCONFPATH とは別のディレクトリに格納したい場合に格納するディレクトリ	〈パス名〉

2.2.2 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、次の表に示します。

表 2-3 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	オプション	定義内容	指定値
set	uap_conf		このノードでユーザサーバを実行するかどうかを指定	Y 《N》
	dam_conf		このノードで DAM サービスを使用するかどうかを指定	Y 《N》
	tam_conf		このノードで TAM サービスを使用するかどうかを指定	Y 《N》
	que_conf		このノードでメッセージキューサービス（MCF サービス）を使用するかどうかを指定	Y 《N》
	ha_conf		このノードで系切り替えをするかどうかを指定	Y 《N》
	jar_conf		このノードでグローバルアーカイブジャーナルサービスを使用するかどうかを指定	Y 《N》

形式	オペランド／コマンド	オプション	定義内容	指定値
set	clt_conf		このノードでクライアント拡張サービスを使用するかどうかを指定	Y 《N》
	ist_conf		このノードで IST サービスを使用するかどうかを指定	Y 《N》
	rmm_conf		このノードで RMM サービスを使用するかどうかを指定	Y 《N》
	xat_conf		このノードで XATMI 通信サービスを使用するかどうかを指定	Y 《N》
	mqa_conf		このノードで MQA サービスを使用するかどうかを指定	Y 《N》
コマンド	dcsvstart	-m	MCF サービスの起動	〈1～8 文字の英数字〉

2.2.3 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、次の表に示します。

表 2-4 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	定義内容	指定値
コマンド	dcsvstart	-u	ユーザサーバの起動	〈1～8 文字の識別子〉

2.2.4 システム共通定義の内容

システム共通定義の内容を、次の表に示します。

表 2-5 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	定義内容	指定値
set	rpc_trace		RPC トレースを取得するかどうかを指定	Y 《N》
	rpc_trace_name		RPC トレースを取得するファイル名	〈パス名〉 《\$DCDIR/spool/rpctr》
	rpc_trace_size		RPC トレースを取得するファイルのサイズ	〈符号なし整数〉 ((1024～2147483648)) 《4096》 (単位：バイト)
	name_port		ネームサービスのポート番号	〈符号なし整数〉

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	name_port		ネームサービスのポート番号	((5001～65535)) 《10000》
	system_id		OpenTP1 識別子	〈1～2 文字の識別子〉
	all_node		ネームサービスの存在するすべてのノード名	〈1～255 文字の識別子〉
			ネームサーバのポート番号	〈符号なし整数〉 ((5001～65535)) 《10000》
			サービス情報優先度指定機能を使用するかどうかを指定	:high
	node_id		ノード識別子	〈4 文字の識別子〉
	rpc_retry		RPC をリトライするかどうかを指定	Y 《N》
	rpc_retry_count		RPC 処理のリトライ回数の最大値	〈符号なし整数〉 ((1～3600)) 《18》
	rpc_retry_interval		RPC 処理のリトライ間隔	〈符号なし整数〉 ((1～3600)) 《10》 (単位：秒)
	multi_node_option		マルチノード機能を使用するかどうかを指定	Y 《N》
	prc_port		マルチノード連携制御機能で使用するポート番号	〈符号なし整数〉 ((5001～49999))
	rpc_delay_statistics		通信遅延時間統計情報を取得するかどうかを指定	Y 《N》
	my_host		使用するネットワークアダプタのホスト名	〈1～255 文字の識別子〉
	jp1_use		JP1/Base にイベントを登録するかどうかを指定	Y 《N》
	rpc_message_level		RPC のトラブルシュートメッセージの表示レベル	〈符号なし整数〉 ((0～2)) 《2》
	rpc_multi_tp1_in_same_host		同一ホスト内の複数 OpenTP1 を同一グローバルドメインとして運用するかどうかを指定	Y 《N》
	max_socket_descriptors		ソケット用ファイル記述子の最大数	〈符号なし整数〉 ((32～2032)) 《64》
	rpc_datacomp		サービス要求時とサービス要求応答時にユーザデータを圧縮して送信するかどうかを指定	Y 《N》

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	domain_masters_addr		通信先のドメイン名	〈1 文字以上の識別子〉
			ドメイン代表スケジュールサービスのホスト名 1 (第 1 優先)	〈1～255 文字の識別子〉
			ドメイン代表スケジュールサービスのポート番号 1 (第 1 優先)	〈符号なし整数〉 ((5001～65535))
			ドメイン代表スケジュールサービスのホスト名 2 (第 2 優先)	〈1～255 文字の識別子〉
			ドメイン代表スケジュールサービスのポート番号 2 (第 2 優先)	〈符号なし整数〉 ((5001～65535))
			ドメイン代表スケジュールサービスのホスト名 3 (第 2 優先)	〈1～255 文字の識別子〉
			ドメイン代表スケジュールサービスのポート番号 3 (第 2 優先)	〈符号なし整数〉 ((5001～65535))
	domain_masters_port		ドメイン代表スケジュールサービスのポート番号	〈符号なし整数〉 ((5001～65535))
	domain_use_dns		ドメイン指定の RPC を発行した際にドメインネームサービスに問い合わせるかどうかを指定	《Y》 N
	client_uid_check		TP1/Client/W または TP1/Client/P に対するユーザ認証機能を使用するかどうかを指定	HP-UX または Windows の場合 : 《Y》 N AIX, Linux または Solaris の場合 : Y 《N》
	rpc_port_base		最小ポート番号	〈符号なし整数〉 ((5001～65535))
	prf_trace		性能検証用トレース情報を取得するかどうかを指定	《Y》 N
	trn_prf_trace_level		トレースの取得レベル	((00000001～00000003)) 《00000003》

	core_suppress_watch_time		実時間監視タイムアウトプロセスのコアダンプ出力を抑止する時間間隔	〈符号なし整数〉 ((0～3600)) 《0》 (単位 : 秒)
	rpc_netmask		TCP/IP のネットワーク定義ファイルに指定したサブネットマスク値	〈「.」 語法の文字列のインターネットアドレス〉
	ipc_sockctl_highwater		ソケットの一時クローズ処理開始数パーセンテージ/ソケットの一時クローズ処理非対象数パーセンテージ	〈符号なし整数〉 ((0～100)) 《100, 0》

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	ipc_socketctl_watchtime		ソケット再利用可能監視時間	〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)
	ipc_conn_interval		コネクション確立監視時間	〈符号なし整数〉 ((8～65535)) 《8》 (単位：秒)
	ipc_send_interval		データ送信監視間隔	〈符号なし整数〉 ((5～32767)) 《5》 (単位：秒)
	ipc_send_count		データ送信監視回数	〈符号なし整数〉 ((1～32767)) 《5》
	ipc_header_rcv_time		通信制御データの受信監視時間	〈符号なし整数〉 ((5～32767)) 《10》 (単位：秒)
	name_notify		起動通知を行うかどうかを指定	Y 《N》
	all_node_ex	ネームサービスの存在するすべてのノード名		〈1～255 文字の識別子〉
		ネームサーバのポート番号		〈符号なし整数〉 ((5001～65535)) 《10000》
	rpc_server_busy_count		KFCA00356-W メッセージを出力するバンドル回数	〈符号なし整数〉 ((0～32767)) 《20》
	rpc_send_retry_count		TCP/IP コネクションの接続時にエラーが発生した場合のリトライ回数	〈符号なし整数〉 ((0～65535)) 《0》
	rpc_send_retry_interval		TCP/IP コネクションの接続時にエラーが発生した場合のリトライ間隔	〈符号なし整数〉 ((0～300000)) 《0》 (単位：ミリ秒)
	thdlock_sleep_time		スレッド間で排他が競合した場合のスレッドの待ち時間	〈符号なし整数〉 ((1～32767)) 《15》 (単位：ミリ秒)
	ipc_rcvbuf_size		TCP/IP の受信バッファサイズ	〈符号なし整数〉 ((0, 8192～1048576)) 《8192》 (単位：バイト)
	ipc_sndbuf_size		TCP/IP の送信バッファサイズ	〈符号なし整数〉 ((0, 8192～1048576)) 《8192》 (単位：バイト)

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	ipc_listen_sockbufset		TCP/IP の送受信バッファサイズを listen ソケットに設定するかどうかを指定	《Y》 N
	rpc_router_retry_count		ENOBUFFS, ENOMEM 発生時のリトライ回数	〈符号なし整数〉 ((0～65535)) 《30》
	rpc_router_retry_interval		ENOBUFFS, ENOMEM 発生時のリトライ間隔	〈符号なし整数〉 ((0～3600000)) 《0》 (単位：ミリ秒)
	ipc_backlog_count		コネクション確立要求を格納するキューの長さ	〈符号なし整数〉 ((0～4096)) 《0》
	statistics		共用メモリ上にシステム統計情報を取得するかどうかを指定	Y 《N》
	name_domain_file_use		OpenTP1 起動時および再開時(リラン)時に有効にするドメイン構成を指定	Y 《N》
	all_node_extend_number		ドメイン変更時の最大ノード数	〈符号なし整数〉 ((0～65535)) 《64》
	all_node_ex_extend_number		ドメイン定義ファイルを使用したドメイン変更時の最大ノード数	〈符号なし整数〉 ((0～65535)) 《64》
	prc_current_work_path		カレントワーキングディレクトリ作成用のディレクトリパス	〈パス名〉 《\$DCDIR》
	rpc_max_message_size		RPC 送受信電文の最大長	〈符号なし整数〉 ((1～8)) 《1》 (単位：メガバイト)
	uap_trace_file_put		UAP トレース情報をファイルに取得するかどうかを指定	Y 《N》
	dcstart_wakeup_retry_count		起動通知エラーリトライ回数	〈符号なし整数〉 ((0～60)) 《0》
	dcstart_wakeup_retry_interval		起動通知エラーリトライ間隔	〈符号なし整数〉 ((0～60)) 《0》 (単位：秒)
	nam_prf_trace_level		NAM イベントトレースの取得レベル	((00000000～00000007)) 《00000003》
	fil_prf_trace_option		FIL イベントトレースを取得するかどうかを指定	0 《1》

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	fil_prf_trace_delay_time		FIL イベントトレース取得条件となるファイルアクセス処理時間のしきい値	〈符号なし整数〉 ((1～65535)) 《3》 (単位：秒)
	jnl_prf_event_trace_level		JNL 性能検証用トレースの取得レベル	((00000000～00000002)) 《00000001》
	jnl_fileless_option		ジャーナルファイルレス機能を使用するかどうかを指定	Y 《N》
	prc_coredump_filter		core ファイルに共有メモリを含めるかの判断値	〈符号なし整数〉 ((1, 3))
	name_service_mode		ノード自動追加機能使用時の動作モード	manager agent 《normal》
	name_manager_node		マネジャノードのノード名	〈1～255 文字の識別子〉
			マネジャノードのネームサーバのポート番号	〈符号なし整数〉 ((5001～65535))
	name_remove_down_node		停止ノード情報の削除要否	Y 《N》
	name_node_add_policy		エージェントノードのノードリストにノーマルノードを登録するかどうかを指定	《using_only》 all
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)
コマンド	dcbindht	-h	ネットワークアダプタに対応するホスト名	〈1～255 文字の識別子〉
		-n	ネットワークアダプタを経由して通信するネットワーク名	〈1～64 文字の識別子〉
putenv	LANG		環境変数 LANG	〈文字列〉

2.2.5 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、次の表に示します。

表 2-6 ロックサービス定義の内容

形式	オペランド	定義内容	指定値
set	lck_limit_foruser	ユーザサーバの最大同時排他要求数	〈符号なし整数〉 ((0～327670)) 《512》

形式	オペランド	定義内容	指定値
set	lck_limit_fordam	DAM サービスの最大同時排他要求数	〈符号なし整数〉 ((0~327670)) 《512》
	lck_limit_fortam	TAM サービスの最大同時排他要求数	〈符号なし整数〉 ((0~327670)) 《512》
	lck_limit_formqa	MQA サービスの最大同時排他要求数	〈符号なし整数〉 ((0~327670)) 《0》
	lck_wait_timeout	排他待ち限界経過時間	〈符号なし整数〉 ((0~32767)) 《0》 (単位：秒)
	lck_deadlock_info	デッドロック情報とタイムアウト情報を出力するかどうかを指定	《Y》 N
	lck_deadlock_info_remove	システム開始時に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を削除するかどうかを指定	normal force 《no》
	lck_deadlock_info_remove_level	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レベル	〈符号なし整数〉 ((0~24855)) 《0》 (単位：日)
	lck_release_detect	プロセスがいつ排他解除したかを調べる方法	interval 《pipe》
	lck_release_detect_interval	排他解除検知インタバル最大時間	〈符号なし整数〉 ((10~60000)) 《250》 (単位：ミリ秒)
	lck_prf_trace_level	LCK 性能検証用トレース情報の取得レベル	((00000000~00000001)) 《00000000》

2.2.6 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、次の表に示します。

表 2-7 タイマサービス定義の内容

形式	オペランド	定義内容	指定値
set	tim_watch_count	最大時間監視サービス数	〈符号なし整数〉 ((0~65535)) 《128》

2.2.7 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、次の表に示します。

表 2-8 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	定義内容	指定値
set	name_total_size		サービス情報領域の大きさ	〈符号なし整数〉 ((1~32767))《64》 (単位：キロバイト)
	name_cache_size		サービス情報キャッシュ領域の大きさ	〈符号なし整数〉 ((1~32767))《16》 (単位：キロバイト)
	max_socket_descriptors		ソケット用ファイル記述子の最大数	〈符号なし整数〉 ((32~2032))
	name_global_lookup		グローバル検索機能を使用するかどうかを指定	Y 《N》
	name_service_extend		ネームサービスで管理するサーバ UAP の取得数を拡張するかどうかを指定	1 《0》
	name_audit_conf		ノード監視機能を使用するかどうかを指定	2 1 《0》
	name_audit_interval		監視サービスのノード監視が終了したあと、次の監視を開始するまでの時間	〈符号なし整数〉 ((1~65535))《60》 (単位：秒)
	name_audit_watch_time		ノードダウンを検知するまでの最大待ち時間	〈符号なし整数〉 ((8~65535))《8》 (単位：秒)
	name_rpc_control_list		ネームサービスの RPC 抑止リストの監視機能を使用するかどうかを指定	《Y》 N
	name_nodeid_check_message		自ノードと同じノード識別子を指定したノードから通信を受信した場合に、KFCA00677-W メッセージを出力するかどうかを指定	《Y》 N
	name_cache_validity_time		他ノードのサービス情報の有効時間	〈符号なし整数〉 ((0~65535))《1800》 (単位：秒)
	name_start_watch_time		OpenTP1 起動時のノードリスト要求応答待ち時間	〈符号なし整数〉 ((0~1800))《8》 (単位：秒)

形式	オペランド／コマンド	オプション	定義内容	指定値
set	name_start_retry_count		OpenTP1 起動時のノードリスト要求リトライ回数	〈符号なし整数〉 ((0～60)) 《0》
	name_start_retry_interval		OpenTP1 起動時のノードリスト要求リトライ間隔	〈符号なし整数〉 ((1～60)) 《10》 (単位：秒)
	name_start_error		ノードリスト要求でエラーが発生した回数が、ネームサービス定義の name_start_retry_count オペランドに指定した値を超えた場合の処理	stop 《continue》
	name_sync_ready_time		ノードリスト同期待ち時間	〈符号なし整数〉 ((0～65535)) 《60》 (単位：秒)
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0～65535)) (単位：秒)
コマンド	namnlfil	-r	再開始時だけノードリストファイルを読み込む	なし。
		-n	再開始時および（強制）正常開始時にノードリストファイルを読み込む	なし。
		引数	ノードリストファイル名	〈1～63 文字のパス名〉

2.2.8 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、次の表に示します。

表 2-9 プロセスサービス定義の内容

形式	オペランド／コマンド	定義内容	指定値
set	prc_process_count		最大同時起動サーバプロセス数 〈符号なし整数〉 ((1～10000)) 《64》
	prc_recovery_resident		部分回復するプロセスの起動方法として常駐か非常駐かを指定 《Y》 N
	prc_take_over_svpath		リラン時にユーザサーバおよびコマンドのパスを引き継ぐかどうかを指定 Y 《N》
	term_watch_time		連続異常終了限界経過時間 〈符号なし整数〉 ((0～32767)) 《10》 (単位：分)

形式	オペランド／コマンド	定義内容	指定値
set	max_socket_descriptors	ソケット用ファイル記述子の最大数	〈符号なし整数〉 ((32～2032))
	term_watch_count	連続異常終了限界回数	〈符号なし整数〉 ((1～3)) 《3》
	prc_prf_trace	プロセスサービスのイベントトレースを取得するかどうかを指定	《Y》 N
	prc_coresave_path	コアファイル格納パス	〈パス名〉 《\$DCDIR/spool/save》
	prc_corecompress	コアファイルの格納時に OpenTP1 で自動的に圧縮	Y 《N》
	watch_time	最大応答待ち時間	〈符号なし整数〉 ((0～65535)) (単位：秒)
コマンド	prcsvpath	プロセスサービスが起動するユーザーサーバのパス名	〈パス名〉 《\$DCDIR/aplib:\$DCDIR/bin》

2.2.9 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、次の表に示します。

表 2-10 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	定義内容	指定値
set	scd_server_count		スケジュールサービスを利用して動作する最大ユーザーサーバ数	〈符号なし整数〉 ((0～8192)) 《32》
	scd_hold_recovery		ユーザーサーバの閉塞状態を引き継ぐかどうかを指定	《Y》 F
	scd_hold_recovery_count		閉塞状態の引き継ぎが必要なサーバ、およびサービスの合計数	〈符号なし整数〉 ((0～58240)) 《64》
	scd_port		スケジュールサービスのポート番号	〈符号なし整数〉 ((5001～65535))
	scd_this_node_first		要求サーバが自ノードにある場合、自ノードのサーバを優先してスケジュールするかどうかを指定	Y 《N》
	scd_announce_server_status		自ノードのサーバの状態をすべての他ノードに不定期で通知するかどうかを指定	《Y》 N

形式	オペランド／コマンド	オプション	定義内容	指定値
set	max_socket_descriptors		ソケット用ファイル記述子の最大数	〈符号なし整数〉 ((32~2032))
	max_open_fds		スケジューラサービスプロセスでアクセスするファイル、およびパイプの最大数	〈符号なし整数〉 ((16~2016)) 《50》
	schedule_rate		サーバの負荷レベルが LEVEL0 のノードのスケジューラ比率	〈符号なし整数〉 ((50~100)) (単位：%)
	scd_retry_of_comm_error		障害ノード以外へのスケジューリングのリトライ回数	〈符号なし整数〉 ((0~128)) 《0》
	scd_advertise_control		システム共通定義の all_node オペランドに指定されたノードに対し、ユーザサーバ起動時にネーム情報を通知するタイミングを変更	《BEFORE》 AFTER
	scd_message_level		メッセージ格納バッファプールのメモリ不足が発生した場合に出力される KFCA00854-E メッセージ出力を抑止	1 《2》
	ipc_tcpnodelay		Nagle アルゴリズムを無効にするかどうかを指定	《Y》 N
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
コマンド	scdbufgrp	-g	スケジューラバッファグループ名	〈1~8 文字の識別子〉
		-e	メッセージ格納バッファプール長	〈符号なし整数〉 ((512~1610612736)) 《512》 (単位：バイト)
		-n	メッセージ格納バッファセル数	〈符号なし整数〉 ((1~3145728)) 《16》
		-l	メッセージ格納バッファセル長	〈符号なし整数〉 ((512~31457280)) 《512》 (単位：バイト)
		-s	メッセージ格納バッファ使用制限サイズ	〈符号なし整数〉 ((512~1610612736)) (単位：バイト)
		-p	メッセージ格納バッファ使用制限率	〈符号なし整数〉 ((1~100)) (単位：%)

形式	オペランド／コマンド	オプション	定義内容	指定値
コマンド	scdmulti	-m	マルチスケジューラデーモン数	〈符号なし整数〉 ((1～4096)) 《1》
		-p	ポート番号	〈符号なし整数〉 ((5001～65535))
		-g	マルチスケジューラグループ名	〈1～8 文字の識別子〉 《scdmltgp》
		-t	ほかのノードのユーザサーバに、サービス要求を負荷分散するかどうかを指定	なし。

2.2.10 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、次の表に示します。

表 2-11 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	定義内容	指定値
set	trn_tran_process_count		同時に起動するトランザクションプ ランチの数	〈符号なし整数〉 ((0～8192)) MCF サービスを使用する場合は、 次に示す指定範囲になります。 32 ビットの場合：((1～7484)) 64 ビットの場合：((1～6893))
	trn_recovery_process_count		トランザクションプランチの回復処 理を並行して行うプロセス数	〈符号なし整数〉 ((1～128)) 《4》
	trn_expiration_time		トランザクションプランチの限界経 過時間	〈符号なし整数〉 ((0～65535)) 《0》 (単位：秒)
	trn_expiration_time_suspend		トランザクションプランチの時間監 視の範囲	Y 《N》 F
	trn_tran_statistics		トランザクションプランチごとの統 計情報を取得するかどうかを指定	Y 《N》
	trn_tran_recovery_list		全面回復時、未決着トランザクシ ョン情報を取得するかどうかを指定	Y 《N》
	trn_cpu_time		トランザクションプランチが同期点 処理までに使用できる CPU 時間	〈符号なし整数〉 ((0～65535)) 《0》 (単位：秒)

形式	オペランド／コマンド	オプション	定義内容	指定値
set	trn_statistics_item		トランザクションブランチの統計情報項目	nothing base 《executiontime》 cputime
	trn_max_subordinate_count		一つのトランザクションブランチから生成する子トランザクションブランチの最大数	〈符号なし整数〉 ((0~1024)) 《32》
	trn_rm_open_close_scope		xa_open 関数および xa_close 関数の発行タイミング	《process》 transaction
	trn_optimum_item		トランザクションの最適化項目	《base》 asyncprepare
	trn_processing_in_rm_error		トランザクション回復時、リソースマネージャに対して発行した XA 関数が処理を続行できないコードでエラーリターンした場合の処置	《down》 retry force
	trn_recovery_list_remove		OpenTP1 開始時、未決着トランザクション情報ファイルを削除するかどうかを指定	normal force 《no》
	trn_recovery_list_remove_level		未決着トランザクション情報ファイルの削除レベル	〈符号なし整数〉 ((0~24855)) 《0》 (単位：日)
	trn_crm_use		CRM を使用するかどうかを指定	Y 《N》
	trn_max_crm_subordinate_count		CRM 経由の子トランザクションブランチの最大数	〈符号なし整数〉 ((0~1024)) 《8》
	trn_watch_time		トランザクション同期点処理時の最大通信待ち時間	〈符号なし整数〉 ((1~65535)) 《120》 (単位：秒)
	trn_rollback_information_put		トランザクションブランチロールバック時にロールバック情報を取得するかどうかを指定	《no》 self remote all
	trn_limit_time		トランザクションブランチ最大実行可能時間	〈符号なし整数〉 ((0~65535)) 《0》 (単位：秒)
	trn_rollback_response_receive		ロールバック完了通知を受信するかどうかを指定	《Y》 N
	trn_partial_recovery_type		UAP 障害時のトランザクション同期点処理方式	《type1》 type2 type3
	max_socket_descriptors		ソケット用ファイル記述子の最大数	〈符号なし整数〉 ((32~2032))

形式	オペランド／コマンド	オプション	定義内容	指定値
set	trn_recovery_failmsg_interval		仕掛り中トランザクション情報メッセージ間隔最小時間	〈符号なし整数〉 ((0~65535)) 《1800》 (単位：秒)
	trn_wait_rm_open		リソースマネージャのオープン処理 (xa_open 関数) でエラーが発生したときの処置	《continue》 stop retry_continue retry_stop
	trn_retry_interval_rm_open		xa_open 関数発行リトライインタバル時間	〈符号なし整数〉 ((1~3600)) 《10》 (単位：秒)
	trn_retry_count_rm_open		xa_open 関数発行リトライ回数	〈符号なし整数〉 ((1~65535)) 《18》
	thread_stack_size		OpenTP1 内部で使用するスレッドスタック領域のサイズ	〈符号なし整数〉 ((1024~524288)) 《49152*1》 (単位：バイト)
	polling_control_data		一時クローズ処理要求が到着していないかどうかを検査	Y 《N》
	thread_yield_interval		ソケットの再利用指示を受信できる契機を与えるインタバル時間	〈符号なし整数〉 ((1~86400)) 《90》 (単位：秒)
	groups		サービスグループのグループアクセスリストを設定	〈符号なし整数〉 ((0~4294967294))
	trn_xar_use		XA リソースサービスを使用するかどうかを指定	Y 《N》
	trn_start_recovery_mode		OpenTP1 開始処理の回復処理モードを指定	《stop》 wait continue
	trn_start_recovery_watch_time		オンライン前回復処理が完了するまでのリトライ上限時間	〈符号なし整数〉 ((0~65535)) 《600》 (単位：秒)
	trn_start_recovery_interval		オンライン前回復処理が完了するまでのリトライ間隔時間	〈符号なし整数〉 ((0~65535)) 《3》 (単位：秒)
	trn_xa_commit_error		複数のリソースをアクセスするトランザクションがコミットに決定したあと、コミットできなくなったりリソースマネージャが発生した場合の処理を指定	《down》 force
	trn_prf_event_trace_level		TRN イベントトレースの取得レベル	((00000000~00000007)) 《00000007》

形式	オペランド／コマンド	オプション	定義内容	指定値
set	trn_prf_event_trace_condition		取得する TRN イベントトレースの種類	《xafunc》 trnservice
	trn_completion_limit_time		トランザクション完了限界時間	〈符号なし整数〉 ((0~65535)) 《0》 (単位：秒)
	trn_extend_function		トランザクションサービスの機能の拡張レベル	〈16 進数字〉 ((00000000~00000001)) 《00000000》
	trn_rcv_open_close_scope		すべてのリソースマネージャ※2 に対する XA 関数の発行順序を変更するかどうかを指定	Y 《N》
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
コマンド	trnstring	-n	リソースマネージャ名	〈1~31 文字の識別子〉
		-i	リソースマネージャ拡張子	〈1~2 文字の識別子〉
		-o	トランザクションサービス用 xa_open 関数用文字列	〈1~256 文字の文字列〉
		-c	トランザクションサービス用 xa_close 関数用文字列	〈1~256 文字の文字列〉
		-O	ユーザサーバ用 xa_open 関数用文字列	〈1~256 文字の文字列〉
		-C	ユーザサーバ用 xa_close 関数用文字列	〈1~256 文字の文字列〉
		-d	このリソースマネージャがコミット最適化およびプリペア最適化に対応していない場合に指定	なし。
		-e	xa_start 関数でエラーが発生した場合に、リソースマネージャに対して xa_close 関数、xa_open 関数、xa_start 関数の順にリトライ処理を行うときに指定	なし。
		-m	OpenTP1 の開始処理で、OpenTP1 がオンラインになる前に、リソースマネージャのトランザクション回復処理を行うときに指定	なし。
		-r	OpenTP1 の開始処理、またはオンライン処理で、リソースマネージャに障害が発生した場合、リソースマネ	なし。

形式	オペランド／コマンド	オプション	定義内容	指定値
コマンド	trnstring	-r	ジャから未決着トランザクション情報の通知があるまで、トランザクションの回復処理を待ち合わせるときに指定	なし。
		-s	特定のリソースマネージャ※2 に対する XA 関数の発行順序を変更するかどうかを指定	なし。
putenv	任意		環境変数名 環境変数値	〈文字列〉
dcputenv	任意		環境変数名 環境変数値	〈文字列〉

注※1

AIX 版の uCosminexus TP1/Server Base(64)の場合、デフォルト値は 65536 となります。

注※2

OpenTP1 が提供するリソースマネージャは対象外です。

2.2.11 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、次の表に示します。

表 2-12 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	定義内容	指定値
set	xar_eventtrace_level		XAR イベントトレース情報の出力レベル	《ERR》 INF ALL
	xar_eventtrace_record		XAR イベントトレース情報ファイルの最大出力レコード数	〈符号なし整数〉 ((1～65535)) 《8192》
	xar_session_time		アイドル状態のトランザクションブランチの監視時間	〈符号なし整数〉 ((10～65535)) 《180》 (単位：秒)
	xar_msdtc_use		MSDTC 連携機能を使用するかどうかを指定	Y 《N》
	xar_prf_trace_level		XAR 性能検証用トレース情報の取得レベル	((00000000～00000003)) 《00000003》
コマンド	xarfile	-t	XAR ファイルのタイプ	online backup
		-a	物理 XAR ファイル名	〈1～63 文字のパス名〉

2.2.12 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、次の表に示します。

表 2-13 インタバルサービス定義の内容

形式	オペランド	定義内容	指定値
set	watch_time	最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)

2.2.13 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、次の表に示します。

表 2-14 ステータスサービス定義の内容

形式	オペランド	定義内容	指定値
set	sts_file_name_1	ステータスファイルの論理ファイル名	〈1~8 文字の識別子〉
		A 系ステータスファイル名	〈パス名〉
		B 系ステータスファイル名	〈パス名〉
	sts_file_name_2	ステータスファイルの論理ファイル名	〈1~8 文字の識別子〉
		A 系ステータスファイル名	〈パス名〉
		B 系ステータスファイル名	〈パス名〉
	sts_file_name_3	ステータスファイルの論理ファイル名	〈1~8 文字の識別子〉
		A 系ステータスファイル名	〈パス名〉
		B 系ステータスファイル名	〈パス名〉
	sts_file_name_4	ステータスファイルの論理ファイル名	〈1~8 文字の識別子〉
		A 系ステータスファイル名	〈パス名〉
		B 系ステータスファイル名	〈パス名〉
	sts_file_name_5	ステータスファイルの論理ファイル名	〈1~8 文字の識別子〉
		A 系ステータスファイル名	〈パス名〉
		B 系ステータスファイル名	〈パス名〉

形式	オペランド	定義内容	指定値
set	sts_file_name_6	ステータスファイルの論理ファイル名	〈1～8 文字の識別子〉
		A 系ステータスファイル名	〈パス名〉
		B 系ステータスファイル名	〈パス名〉
	sts_file_name_7	ステータスファイルの論理ファイル名	〈1～8 文字の識別子〉
		A 系ステータスファイル名	〈パス名〉
		B 系ステータスファイル名	〈パス名〉
	sts_initial_error_switch	ステータスサービスの障害時の処置 (ステータスファイルの実体がない、 またはステータスファイル障害を検知した場合)	《stop》 continue excontinue
	sts_single_operation_switch	ステータスサービスの障害時の処置 (現用ステータスファイル片系に入出力障害が発生し、かつスワップできない場合)	《stop》 continue
	sts_last_active_file	前回のオンラインの最新現用論理ファイル名	〈1～8 文字の識別子〉
	sts_last_active_side	前回オンラインで片系障害運転の場合に正常だった系	A B
	watch_time	最大応答待ち時間	〈符号なし整数〉 ((0～65535)) (単位：秒)

2.2.14 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、次の表に示します。

表 2-15 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	定義内容	指定値
set	jnl_tran_optimum_level		ジャーナルの出力方式に対する最適化レベル	〈符号なし整数〉 ((2～3)) 《2》
	jnl_arc_terminate_timeout		ノード終了時にアーカイブジャーナルサービスの終了を待ち合わせる最大時間	〈符号なし整数〉 ((0～3600)) 《0》 (単位：秒)
	max_socket_descriptors		ソケット用ファイル記述子の最大数	〈符号なし整数〉

形式	オペランド／コマンド	オプション	定義内容	指定値
set	max_socket_descriptors		ソケット用ファイル記述子の最大数	((32~2032))
	jnl_arc_ipc_buff_size		TCP/IP の送受信バッファサイズ	〈符号なし整数〉 ((8192~1048576)) 《61440》 (単位：バイト)
	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)
	jnl_message_id_change_level		メッセージ ID の変更レベル	〈符号なし整数〉 ((0~1)) 《0》
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
コマンド	jnldfsv	-r	システムジャーナルサービス定義のファイル名	〈1~8 文字の識別子〉
		-c	チェックポイントダンプサービス定義のファイル名 (必要なシステムサービスを複数指定可)	〈1~8 文字の識別子〉

2.2.15 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、次の表に示します。

表 2-16 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	定義内容	指定値
set	jnl_max_datasize		最大レコードデータ長	〈符号なし整数〉 ((32000~4000000)) 《32000》 (単位：バイト)
	jnl_cdinterval		ジャーナルブロック数	〈符号なし整数〉 ((100~32000)) 《1000》
	jnl_rerun_swap		システム再開始時にジャーナルをスワップするかどうかを指定	Y 《N》
	jnl_dual		ジャーナルファイルを二重化するかどうかを指定	Y 《N》
	jnl_singleoperation		ジャーナルファイルが片系しか使用できない場合にスワップするかどうかを指定	Y 《N》

形式	オペランド／コマンド	オプション	定義内容	指定値
set	jnl_rerun_reserved_file_open		全面回復時に予約ファイルをオープンするかどうかを指定	Y 《N》
	jnl_arc_name		グローバルアーカイブジャーナルサービスのリソースグループ名@ノード識別名	〈1～8 文字の識別子〉@ 〈4 文字の識別子〉
	jnl_arc_buff_size		アーカイブするための共用メモリバッファの容量	〈符号なし整数〉 ((2～31)) 《10》 (単位：メガバイト)
	jnl_arc_max_datasize		アーカイブ時の転送データの最大長	〈符号なし整数〉 ((1020～8188)) 《1020》 (単位：キロバイト)
	jnl_arc_terminate_check		ジャーナルサービスの正常終了および計画停止時、アーカイブ対象ジャーナルをすべてアーカイブするかどうかを指定	《Y》 N
	jnl_arc_rec_kind		アーカイブ対象になるジャーナルレコード種別	《a c f g i m o s u》
	jnl_arc_uj_code		アーカイブする UJ コード	〈符号なし整数〉 ((0～255)) 《0-255》
	jnl_arc_check_level		グローバルアーカイブジャーナル使用時にファイルグループがスワップ先として割り当てられる条件	《1》 2
	jnl_arc_trn_stat		同期点ジャーナルおよびトランザクション管理サービスの回復用ジャーナルをアーカイブの対象とするかどうかを指定	《Y》 N
	jnl_unload_check		スワップ先のファイルグループを決定する時にアンロード待ち状態をチェックするかどうかを指定	《Y》 N
	jnl_auto_unload		自動アンロード機能を使用するかどうかを指定	Y 《N》
	jnl_auto_unload_path		アンロードジャーナルファイルの格納ディレクトリ名	〈パス名〉 《\$DCDIR/spool/dcjnlinf/unload》
	jnl_max_file_dispersion		並列アクセス化する場合の最大分散数	〈符号なし整数〉 ((1～8)) 《1》
	jnl_min_file_dispersion		並列アクセス化する場合の最小分散数	〈符号なし整数〉 ((1～8)) 《1》

形式	オペランド／コマンド	オプション	定義内容	指定値
set	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
コマンド	jnladdfg	-g	システムジャーナルを構成するファイルグループ名	〈1~8 文字の識別子〉
		ONL	オンライン開始時にこのファイルグループを使用するかどうかを指定	なし。
	jnladdpf	-g	対応するファイルグループ名	〈1~8 文字の識別子〉
		-e	要素ファイル名	〈1~8 文字の識別子〉
		-a	物理ファイルのパス名	〈パス名〉
		-b	物理ファイルのパス名	〈パス名〉

2.2.16 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、次の表に示します。

表 2-17 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	定義内容	指定値
set	jnl_objservername		対象になるシステムサービス名	〈1~8 文字の識別子〉
	jnl_max_datasize		チェックポイントダンプ取得用バッファ長	〈符号なし整数〉 ((32000~4000000)) 《32768》 (単位：バイト)
	assurance_count		有効保証世代数	〈符号なし整数〉 ((1~2)) 《1》 (単位：世代ファイル)
	jnl_reduced_mode		縮退運転オプション	〈符号なし整数〉 ((0~2)) 《0》
	jnl_reserved_file_auto_open		予約ファイルを自動的にオープンするかどうかを指定	Y 《N》
	jnl_dual		チェックポイントダンプファイルを二重化するかどうかを指定	Y 《N》
	jnl_singleoperation		チェックポイントダンプファイル二重化の際、片系しか上書きできなくなった場合に上書きできるファイル	Y 《N》

形式	オペランド／コマンド	オプション	定義内容	指定値
set	jnl_singleoperation		グループとして割り当てるかどうかを指定	Y 《N》
	jnl_cdskip_limit		チェックポイントダンプ取得契機のスキップ回数の上限值	〈符号なし整数〉 ((1~32767)) (単位：回)
	jnl_cdskip_msg		チェックポイントダンプ取得契機をスキップする要因となった、トランザクションの情報を表示するかどうかを指定	Y 《N》
	cpd_message_id_change_level		メッセージ ID の変更レベル	〈符号なし整数〉 ((0~1)) 《0》
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
コマンド	jnladdfg	-j	サーバリカバリジャーナルファイル	〈1~8 文字の識別子〉
		-g	チェックポイントダンプまたはサーバリカバリジャーナルを構成するファイルグループ名	〈1~8 文字の識別子〉
		ONL	オンライン開始時にこのファイルグループを使用するかどうかを指定	なし。
	jnladdpf	-j	サーバリカバリジャーナルファイル	〈1~8 文字の識別子〉
		-g	対応するファイルグループ名	〈1~8 文字の識別子〉
		-a	物理ファイルのパス名	〈パス名〉
		-b	物理ファイルのパス名	〈パス名〉

2.2.17 ログサービス定義の内容

ログサービス定義の内容を、次の表に示します。

表 2-18 ログサービス定義の内容

形式	オペランド／パス	定義内容	指定値
set	log_filesize	メッセージログファイルの最大サイズ	〈符号なし整数〉 ((1~32767)) 《10240》 (単位：キロバイト)
	log_msg_console	メッセージログをリアルタイム出力機能を使用して出力するかどうかを指定	《Y》 N

形式	オペランド／パス	定義内容	指定値
set	log_msg_allno	メッセージログのシステム内のメッセージ通番を付けるかどうかを指定	Y 《N》
	log_msg_prcid	要求元のプロセス ID を付けるかどうかを指定	Y 《N》
	log_msg_prcno	メッセージログのプロセス内のメッセージ通番を付けるかどうかを指定	Y 《N》
	log_msg_sysid	OpenTP1 識別子を付けるかどうかを指定	《Y》 N
	log_msg_date	メッセージログの出力要求時の日付を付けるかどうかを指定	《Y》 N
	log_msg_time	メッセージログの出力要求時の時刻を付けるかどうかを指定	《Y》 N
	log_msg_hostname	メッセージログの出力要求元のホスト名を付けるかどうかを指定	《Y》 N
	log_msg_pgmid	メッセージログの出力要求元のプログラム ID を付けるかどうかを指定	《Y》 N
	log_netm_out	メッセージログを NETM に出力するかどうかを指定	Y 《N》
	log_netm_allno	メッセージログのシステム内の通番を付けるかどうかを指定	Y 《N》
	log_netm_prcid	要求元のプロセス ID を付けるかどうかを指定	Y 《N》
	log_netm_prcno	メッセージログのプロセス内の通番を付けるかどうかを指定	Y 《N》
	log_netm_sysid	OpenTP1 識別子を付けるかどうかを指定	《Y》 N
	log_netm_date	メッセージログの出力要求時の日付を付けるかどうかを指定	《Y》 N
	log_netm_time	メッセージログの出力要求時の時刻を付けるかどうかを指定	《Y》 N
	log_netm_hostname	メッセージログの出力要求元のホスト名を付けるかどうかを指定	《Y》 N
	log_netm_pgmid	メッセージログの出力要求元のプログラム ID を付けるかどうかを指定	《Y》 N
	log_jpl_allno	メッセージログのシステム内の通番を付けるかどうかを指定	Y 《N》
	log_jpl_prcid	要求元のプロセス ID を付けるかどうかを指定	Y 《N》

形式	オペランド／パス	定義内容	指定値
set	log_jpl_prcno	メッセージログのプロセス内の通番を付けるかどうかを指定	Y 《N》
	log_jpl_sysid	OpenTP1 識別子を付けるかどうかを指定	《Y》 N
	log_jpl_date	メッセージログの出力要求時の日付を付けるかどうかを指定	《Y》 N
	log_jpl_time	メッセージログの出力要求時の時刻を付けるかどうかを指定	《Y》 N
	log_jpl_hostname	メッセージログの出力要求元のホスト名を付けるかどうかを指定	《Y》 N
	log_jpl_pgmid	メッセージログの出力要求元のプログラム ID を付けるかどうかを指定	《Y》 N
	log_notify_out	メッセージログ通知機能を使用するかどうかを指定	Y 《N》
	log_notify_allno	メッセージログのシステム内の通番を付けるかどうかを指定	Y 《N》
	log_notify_prcid	要求元のプロセス ID を付けるかどうかを指定	Y 《N》
	log_notify_prcno	メッセージログのプロセス内の通番を付けるかどうかを指定	Y 《N》
	log_notify_sysid	OpenTP1 識別子を付けるかどうかを指定	《Y》 N
	log_notify_date	メッセージログの出力要求時の日付を付けるかどうかを指定	《Y》 N
	log_notify_time	メッセージログの出力要求時の時刻を付けるかどうかを指定	《Y》 N
	log_notify_hostname	メッセージログの出力要求元のホスト名を付けるかどうかを指定	《Y》 N
	log_notify_pgmid	メッセージログの出力要求元のプログラム ID を付けるかどうかを指定	《Y》 N
	log_jerr_rint	メッセージログ出力中にエラーが発生した場合のメッセージログ出力抑止回数	〈符号なし整数〉 ((1~65536)) 《32》
	log_syslog_out	メッセージを syslog に出力するレベル	〈符号なし整数〉 ((0~2)) 《1》
	log_syslog_allno	メッセージログのシステム内の通番を付けるかどうかを指定	Y 《N》

形式	オペランド／パス	定義内容	指定値
set	log_syslog_prcid	要求元のプロセス ID を付けるかどうかを指定	Y 《N》
	log_syslog_prcno	メッセージログのプロセス内の通番を付けるかどうかを指定	Y 《N》
	log_syslog_sysid	OpenTP1 識別子を付けるかどうかを指定	Y 《N》
	log_syslog_date	メッセージログの出力要求時の日付を付けるかどうかを指定	Y 《N》
	log_syslog_time	メッセージログの出力要求時の時刻を付けるかどうかを指定	Y 《N》
	log_syslog_hostname	メッセージログの出力要求元のホスト名を付けるかどうかを指定	Y 《N》
	log_syslog_pgmid	メッセージログの出力要求元のプログラム ID を付けるかどうかを指定	Y 《N》
	log_syslog_append_nodeid	ノード識別子を付けるかどうかを指定	Y 《N》
	log_syslog_elist	syslog 失敗リストのエレメント数	〈符号なし整数〉 ((0~65536)) 《0》
	log_syslog_elist_rint	syslog 失敗リストを定期的に出力する間隔	〈符号なし整数〉 ((0~65536)) 《0》 (単位：秒)
	log_syslog_synchro	メッセージログの出力失敗時、syslog に出力するメッセージログのフォーマットを関連オペランドに同期させるかどうかを指定	Y 《N》
	log_audit_out	監査ログ機能を使用するかどうかを指定	Y 《N》
	log_audit_path	監査ログファイルの出力先ディレクトリ	〈1~63 文字のパス名〉 《\$DCDIR/auditlog》
	log_audit_size	監査ログファイルの最大サイズ	〈符号なし整数〉 ((1~2047)) 《10》 (単位：メガバイト)
	log_audit_count	監査ログファイルの最大数	〈符号なし整数〉 ((1~256)) 《2》
	log_audit_message	監査ログを取得する項目のメッセージ ID	〈符号なし整数〉 ((33400~99999))
	watch_time	最大応答待ち時間	〈符号なし整数〉 ((0~65535))

形式	オペランド／パス	定義内容	指定値
set	watch_time	最大応答待ち時間	(単位：秒)
putenv	TZ	タイムゾーン	〈文字列〉
	DCSYSLOGOUT	syslog ヘッセージを出力するかどうかを指定	1 1 以外の文字列

2.2.18 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、次の表に示します。

表 2-19 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	定義内容	指定値
set	dcmstart_watch_time		dcmstart コマンド打ち切り時間	〈符号なし整数〉 ((0~65535)) 《600》 (単位：秒)
	dcmstop_watch_time		dcmstop コマンド打ち切り時間	〈符号なし整数〉 ((0~65535)) 《600》 (単位：秒)
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0~65535)) 《180》 (単位：秒)
コマンド	dcmarea	-m	マルチノードエリア識別子	〈1~8 文字の識別子〉
		-g	マルチノードサブエリア識別子	〈1~8 文字の識別子〉
		-w	ノード識別子	〈4 文字の識別子〉

2.2.19 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、次の表に示します。

表 2-20 マルチノード物理定義の内容

形式	コマンド	オプション	定義内容	指定値
コマンド	dcprcport	-w	ノード識別子	〈4 文字の識別子〉
		-h	OpenTP1 ノードがあるホスト名	〈1~64 文字の識別子〉
		-p	対応する OpenTP1 ノードの prc_port オペランドの指定値	〈符号なし整数〉 ((5001~49999))

2.2.20 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、次の表に示します。

表 2-21 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	定義内容	指定値
set	jnl_arc_terminate_timeout		ノード終了時にすべての被アーカイブジャーナルノードの接続を解除するまでの最大待ち合わせ時間	〈符号なし整数〉 ((0~3600)) 《0》 (単位：秒)
	max_socket_descriptors		ソケット用ファイル記述子の最大数	〈符号なし整数〉 ((32~2032))
	jnl_arc_ipc_buff_size		TCP/IP の送受信バッファサイズ	〈符号なし整数〉 ((8192~1048576)) 《61440》 (単位：バイト)
	jnl_watch_time		グローバルアーカイブ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)
	jnl_message_id_change_level		メッセージ ID の変更レベル	〈符号なし整数〉 ((0~1)) 《0》
コマンド	jnldfs	-a	アーカイブジャーナルサービス定義のファイル名	〈1~8 文字の識別子〉

2.2.21 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、次の表に示します。

表 2-22 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	定義内容	指定値
set	jnl_dual		アーカイブを二重化するかどうかを指定	Y 《N》
	jnl_singleoperation		片系クローズ状態のファイルグループを使用するかどうかを指定	Y 《N》
	jnl_rerun_swap		グローバルアーカイブジャーナルサービスのリラン時にスワップするかどうかを指定	Y 《N》
	jnl_max_file_dispersion		並列アクセス化する場合の最大分散数	〈符号なし整数〉 ((1~8)) 《1》

形式	オペランド／コマンド	オプション	定義内容	指定値
set	jnl_min_file_dispersion		並列アクセス化する場合の最小分散数	〈符号なし整数〉 ((1～8)) 《1》
	jnl_unload_check		スワップ先のファイルグループを決定する時にアンロード待ち状態をチェックするかどうかを指定	《Y》 N
	jnl_arc_max_datasize		アーカイブ時の転送データの最大長	〈符号なし整数〉 ((1020～8188)) 《1020》 (単位：キロバイト)
コマンド	jnladdfg	-g	アーカイブジャーナルファイルのファイルグループ名	〈1～8 文字の識別子〉
		ONL	グローバルアーカイブジャーナルサービス開始時にこのグループを指定するかどうかを指定	なし。
	jnladdpf	-g	アーカイブジャーナルファイルのファイルグループ名	〈1～8 文字の識別子〉
		-e	要素ファイル名	〈1～8 文字の識別子〉
		-a	物理ファイルのパス名	〈パス名〉
		-b	物理ファイルのパス名	〈パス名〉

2.2.22 DAM サービス定義の内容

DAM サービス定義の内容を、次の表に示します。

表 2-23 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	定義内容	指定値
set	dam_update_block		更新する最大ブロック数	〈符号なし整数〉 ((1～32768)) 《8》
	dam_added_file		オンライン中に追加する最大論理ファイル数	〈符号なし整数〉 ((1～128)) 《8》
	dam_update_block_over		トランザクションの途中で最大ブロック数を越えたときにエラーリターンするかどうかを指定	flush 《error》
	dam_message_level		DAM サービスが出力するメッセージのレベル	《1》 0

形式	オペランド／コ マンド	オプション	定義内容	指定値
set	dam_tran_process_count		同時に実行するトランザクションブ ランチの数	〈符号なし整数〉 ((0~8192))
	dam_cache_size		参照・更新ブロックをスタックして おくバッファ領域長（固定値として 設定しない場合）	〈符号なし整数〉 ((10~1000000)) (単位：キロバイト)
	dam_cache_size_fix		参照・更新ブロックをスタックして おくバッファ領域長（固定値として 設定する場合）	〈符号なし整数〉 ((10~1000000)) (単位：キロバイト)
	dam_cache_attribute		バッファ領域のメモリ固定要否	《free》 fixed
	dam_io_interval		デフォルト更新機能使用時の出力 専用プロセス実行間隔	〈符号なし整数〉 ((1~60)) (単位：秒) 《1》
	dam_transaction_access		DAM ファイルのデータ管理、およ び排他管理をするトランザクシ ョンの範囲	global 《branch》
	dam_io_error_occur		ディスク障害時の DAM サービスの 処置	stop 《continue》
	dam_cache_reuse_from		再利用するキャッシュブロックの 検索	《last》 first
	dam_default_cache_num		キャッシュブロック再利用境界デ フォルト値の指定	〈符号なし整数〉 ((0~4000000)) 《0》
	dam_ex_refer_read		排他あり参照指定で読み込むブロッ クを、トランザクション決着まで キャッシュバッファ領域上に残すか どうかを指定	《none》 stay
	dam_max_block_size		システム内の DAM ファイルの最大 ブロック長	〈符号なし整数〉 ((504~32760))* 《504》 (単位：バイト)
	dam_kb_size		dam_cache_size オペランドの指定 値（単位：キロバイト）を 1 キロバ イト当たり 1000 バイトで換算する か、1024 バイトで換算するかを 指定	《1000》 1024
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
コマンド	damcache	引数	キャッシュブロック再利用処理の境 界値を有効とする論理ファイル名	〈1~8 文字の識別子〉

形式	オペランド／コマンド	オプション	定義内容	指定値
コマンド	damcache	引数	キャッシュブロック再利用境界値	〈符号なし整数〉 ((0~4000000))
	damchlmnt	引数	キャッシュブロック数のしきい値を有効とする論理ファイル名	〈1~8 文字の識別子〉
			キャッシュブロック数のしきい値	〈符号なし整数〉 ((0~4000000))
	damfile	-d	ファイルのディファード更新処理の可否	なし。
		-n	回復対象外ファイルとするかどうかを指定	なし。
		-f	回復対象外ファイルをキャッシュレスアクセスとするかどうかを指定	なし。
		-c	DAM ファイルに障害が発生しても、DAM サービスの正常開始処理を続行する場合に指定	なし。
		引数	オンライン中にアクセスする論理ファイル名	〈1~8 文字の識別子〉
			オンライン中にアクセスする物理ファイル名	〈1~63 文字のパス名〉

注※

セクタ長×n-8 を満たす値を指定してください (n は正の整数)。

2.2.23 TAM サービス定義の内容

TAM サービス定義の内容を、次の表に示します。

表 2-24 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	定義内容	指定値
set	tam_max_tblnum		オンライン中に使用する TAM テーブルの最大数	〈符号なし整数〉 ((1~65535))
	tam_max_filesize		オンライン中に使用する TAM テーブルの最大容量	〈符号なし整数〉 ((136~1000000000)) (単位: バイト)
	tam_max_recsz		TAM テーブルの最大レコード長	〈符号なし整数〉 ((1~1000000000))

形式	オペランド／コマンド	オプション	定義内容	指定値
set	tam_max_recsz		TAM テーブルの最大レコード長	(単位：バイト)
	tam_jnl_err_flag		ジャーナルの読み込み障害時に TAM サービスを停止するかどうかを指定	《STOP》 CONTINUE
	tam_pool_attri		共用メモリプールのメモリ固定要否	HP-UX または Solaris の場合： 《fixed》 free AIX, Linux または Windows の場合：fixed 《free》
	tam_tbl_lock_mode		アクセス時にテーブルを排他するかどうかを指定	《LOCK》 NOLOCK
	tam_cbl_level		COBOL API の排他レベル	〈符号なし整数〉 ((0~2)) 《0》
	tam_max_trnnum		同時走行最大トランザクションブランチ数	〈符号なし整数〉 ((1~8192)) 《20》
	tam_max_trnfilnum		トランザクション内最大アクセステーブル数	〈符号なし整数〉 ((1~1024)) 《5》
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
コマンド	tamtable	-o	ローディング契機	《start》 cmd lib
		-a	アクセス形態	《read》 rewrite write
		-i	TAM テーブル更新時に入出力障害が発生してもアクセスを続行するかどうかを指定	なし。
		-j	TAM レコード更新時に部分ジャーナルを取得するかどうかを指定	なし。
		引数	オンライン中にアクセスする TAM テーブル名	〈1~32 文字の識別子〉
			オンライン中にアクセスする物理ファイル名	〈1~63 文字のパス名〉

2.2.24 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、次の表に示します。

表 2-25 クライアントサービス定義の内容

形式	オペランド	定義内容	指定値
set	parallel_count	常駐プロセス数と最大プロセス数	〈符号なし整数〉 ((1~1024)) 《1》
	balance_count	1 プロセスが処理するサービス要求数	〈符号なし整数〉 ((0~512)) 《3》
	trn_expiration_time	トランザクションブランチ限界経過時間	〈符号なし整数〉 ((0~65535)) 《0》 (単位：秒)
	trn_expiration_time_suspend	トランザクションブランチの時間監視の範囲	Y 《N》 F
	trn_cpu_time	トランザクションブランチが同期点処理までに使用できる CPU 時間	〈符号なし整数〉 ((0~65535)) 《0》 (単位：秒)
	open_rm	トランザクショナル RPC 機能使用時にクライアントサービス実行プログラムがオープンするリソースマネージャ名	OpenTP1_ALL 《OpenTP1_NONE》
	clt_inquire_time	常設コネクション問い合わせ間隔最大時間	〈符号なし整数〉 ((0~1048575)) 《180》 (単位：秒)
	clt_port	クライアント拡張サービスのポート番号	〈符号なし整数〉 ((5001~65535))
	clt_trn_conf	トランザクショナル RPC 実行プロセスを起動するかどうかを指定	《Y》 N
	clt_cup_conf	CUP 実行プロセスを起動するかどうかを指定	Y 《N》
	cup_parallel_count	CUP 実行プロセスの常駐プロセス数と最大プロセス数	〈符号なし整数〉 ((1~1024)) 《1》
	cup_balance_count	CUP 実行プロセスのサービス要求滞留値	〈符号なし整数〉 ((0~512)) 《3》
	clttrn_port	トランザクショナル RPC 実行プロセスのポート番号	〈符号なし整数〉 ((5001~65535))
	cltcon_port	CUP 実行プロセスのポート番号	〈符号なし整数〉 ((5001~65535))
	trn_statistics_item	トランザクションブランチの統計情報項目	nothing base 《executiontime》 cputime
	trn_optimum_item	トランザクションの最適化項目	《base》 asyncrepare

形式	オペランド	定義内容	指定値
set	trn_watch_time	トランザクション同期点処理時の最大通信待ち時間	〈符号なし整数〉 ((1~65535)) 《120》 (単位：秒)
	trn_rollback_information_put	トランザクションブランチロールバック時にロールバック情報を取得するかどうかを指定	《no》 self remote all
	trn_limit_time	トランザクションブランチ最大実行可能時間	〈符号なし整数〉 ((0~65535)) 《0》 (単位：秒)
	trn_rollback_response_receive	ロールバック完了通知を受信するかどうかを指定	《Y》 N
	trn_partial_recovery_type	UAP 障害時のトランザクション同期点処理方式	《type1》 type2 type3
	trn_completion_limit_time	トランザクション完了限界時間	〈符号なし整数〉 ((0~65535)) 《0》 (単位：秒)
	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)
	watch_time	最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)

2.2.25 IST サービス定義の内容

IST サービス定義の内容を、次の表に示します。

表 2-26 IST サービス定義の内容

形式	オペランド／コマンド	定義内容	指定値
set	ist_node	IST テーブルを操作するノード	〈4 文字の識別子〉
	ist_node_group	IST テーブルを操作するノードグループ	〈1~8 文字の識別子〉
コマンド	istdef	テーブル名	〈1~8 文字の識別子〉
		レコード長	〈符号なし整数〉 ((4~65536)) (単位：バイト)
		レコード数	〈符号なし整数〉 ((1~16384))

形式	オペランド／コマンド	定義内容	指定値
コマンド	istdef	レコード数	(単位：個)

2.2.26 RMM サービス定義の内容

RMM サービス定義の内容を、次の表に示します。

表 2-27 RMM サービス定義の内容

形式	オペランド	定義内容	指定値
set	rmm_check_services	監視対象リソースマネージャ名	〈1～7 文字の識別子〉
	rmm_system_behavior	監視対象リソースマネージャ開始失敗時のシステムの処置	《down》 giveup
	rmm_down_with_system	OpenTP1 ダウン時に、監視対象リソースマネージャも同時にダウンさせるかどうかを指定	《Y》 N
	rmm_sysdown_with_rm	監視対象リソースマネージャダウン時に、OpenTP1 も同時にダウンさせるかどうかを指定	Y 《N》

2.2.27 監視対象 RM 定義の内容

監視対象 RM 定義の内容を、次の表に示します。

表 2-28 監視対象 RM 定義の内容

形式	オペランド	定義内容	指定値
set	rmm_start_command	監視対象リソースマネージャ開始コマンド	〈1～127 バイトのパス名〉
	rmm_stop_command	監視対象リソースマネージャ終了コマンド	〈1～127 バイトのパス名〉
	rmm_abort_command	監視対象リソースマネージャ強制終了コマンド	〈1～127 バイトのパス名〉
	rmm_get_pid_command	監視対象プロセス ID 取得コマンド	〈1～127 バイトのパス名〉
	rmm_command_watch_time	監視対象リソースマネージャコマンド監視時間	〈符号なし整数〉 ((0～7200)) 《1800》 (単位：秒)
	rmm_command_uid	監視対象リソースマネージャの実行権限を持つユーザ ID	〈符号なし整数〉 ((0～59999))

形式	オペランド	定義内容	指定値
set	rmm_command_gid	監視対象リソースマネージャの実行権限を持つグループ ID	〈符号なし整数〉 ((0~59999))
	rmm_start_watch_time	監視対象リソースマネージャの開始処理を監視する時間	〈符号なし整数〉 ((0~7200)) 《300》 (単位：秒)

2.2.28 拡張 RM 登録定義の内容

拡張 RM 登録定義の内容を、次の表に示します。

表 2-29 拡張 RM 登録定義の内容

形式	コマンド	オプション	定義内容	指定値
コマンド	trnlncrm	-a	追加する OpenTP1 提供以外のリソースマネージャ名	〈1~31 文字の英数字〉
		-s	追加するリソースマネージャのスイッチ名	〈先頭が英字、またはアンダスコアで始まる 1~32 文字の英数字〉
		-o	追加する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名	〈英数字〉
		-C	コンパイル実行時に使用するコンパイルオプション名	〈1~512 文字の文字列〉
		-B	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	〈1~512 文字の文字列〉
		-l	trnlncrm コマンドの実行経過を標準出力に出力する場合に指定	なし。
		-f	OpenTP1 の状態に関係なく trnlncrm コマンドを強制的に実行する場合に指定	なし。

2.2.29 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、次の表に示します。

表 2-30 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	定義内容	指定値
set	xatinitapt		システムごとの自 AP 名称	〈1~24 文字の 16 進数字〉

形式	オペランド／コマンド	オプション	定義内容	指定値
set	xatinitaeq		システムごとの自 AE 修飾子	〈符号なし整数〉 ((0~2147483647))
	xat_aso_con_event_svcname		アソシエーションの確立通知を受信する SPP のサービスグループ名	〈1~31 文字の識別子〉
			アソシエーションの確立通知を受信する SPP のサービス名	〈1~31 文字の識別子〉
	xat_aso_discon_event_svcname		アソシエーションの正常解放通知を受信する SPP のサービスグループ名	〈1~31 文字の識別子〉
			アソシエーションの正常解放通知を受信する SPP のサービス名	〈1~31 文字の識別子〉
	xat_aso_failure_event_svcname		アソシエーションの異常解放通知を受信する SPP のサービスグループ名	〈1~31 文字の識別子〉
			アソシエーションの異常解放通知を受信する SPP のサービス名	〈1~31 文字の識別子〉
	max_open_fds		OSI TP 通信のアソシエーションによって使用される FDS の最大値	〈符号なし整数〉 ((16~2016)) 《50》
コマンド	xatsrvadd	-p	他 AP 名称	〈1~24 文字の 16 進数字〉
		-q	他 AE 修飾子	〈符号なし整数〉 ((0~2147483647))
		-s	他システムで提供されるサービス名	〈1~15 文字の 16 進数字〉

2.2.30 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、次の表に示します。

表 2-31 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	定義内容	指定値
set	que_xidnum		同時に実行するトランザクションの最大数	〈符号なし整数〉 ((1~4096)) 《256》
	que_io_maxrecsize		遅延書き込み最大レコード長	〈符号なし整数〉 ((0~32000)) 《0》 (単位：バイト)

形式	オペランド／コマンド	オプション	定義内容	指定値
コマンド	quegrp	-g	物理ファイルに割り当てるキューグループ ID	〈1～8 文字の識別子〉
		-f	物理ファイルのパス名	〈パス名〉
		-n	バッファキャッシュ用の入出力バッファ数	〈符号なし整数〉 ((2～1024)) 《128》
		-m	キューファイルの保持メッセージ数	〈符号なし整数〉 ((0～1024)) 《10》
		-w	物理ファイルの使用容量警告率	〈符号なし整数〉 ((0～95)) 《80》 (単位：%)
		-c	物理ファイルの使用容量に対する警告解除使用率	〈符号なし整数〉 ((0～95)) 《0》 (単位：%)

2.2.31 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、次の表に示します。

表 2-32 ユーザ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	定義内容	指定値
コマンド	dcsvgdef	-g	サービスグループ名	〈1～31 文字の識別子〉
		-h	ホスト名	〈1～255 文字の識別子〉
		-p	ポート番号	〈符号なし整数〉 ((1～65535))
		-t	あて先再選択間隔	〈符号なし整数〉 ((0～65534)) (単位：秒)
		-w	リモート API 機能を使用する場合に指定	なし。

2.2.32 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、次の表に示します。

表 2-33 rap リスナーサービス定義の内容

形式	オペランド	定義内容	指定値
set	rap_listen_port	rap リスナーポート番号	〈符号なし整数〉 ((5001~65535))
	rap_parallel_server	rap サーバ数	〈符号なし整数〉 ((1~1024)) 《1》
	rap_watch_time	メッセージ送受信最大監視時間	〈符号なし整数〉 ((0~65535)) 《180》 (単位：秒)
	rap_inquire_time	問い合わせ間隔最大時間	〈符号なし整数〉 ((0~1048575)) 《180》 (単位：秒)
	nice	プロセスの優先順位の変更	〈符号なし整数〉 ((0~39)) 《0》
	uap_trace_max	UAP トレース格納最大数	〈符号なし整数〉 ((0~4095)) 《32》
	uid	OpenTP1 管理者のユーザ識別子	〈符号なし整数〉 ((0~4294967294))
	rpc_response_statistics	レスポンス統計情報を取得するかどうかを指定	Y 《N》
	rpc_trace	RPC トレースを取得するかどうかを指定	Y N
	rpc_trace_name	RPC トレースを取得するファイル名	〈パス名〉
	rpc_trace_size	RPC トレースを取得するファイルのサイズ	〈符号なし整数〉 ((1024~2147483648)) (単位：バイト)
	trn_expiration_time	トランザクションブランチの限界経過時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	trn_expiration_time_suspend	トランザクションブランチの時間監視の範囲	Y N F
	trn_cpu_time	トランザクションブランチが同期点処理までに使用できる CPU 時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	trf_put	トランザクションが出力したジャーナルをトランザクションリカバリジャーナルファイルへ出力するかどうかを指定	Y N

形式	オペランド	定義内容	指定値
set	trn_statistics_item	トランザクションブランチの統計情報項目	nothing base executiontime cputime
	trn_optimum_item	トランザクションの最適化項目	base asyncprepare
	trn_watch_time	トランザクション同期点処理時の最大通信待ち時間	《符号なし整数》 ((1~65535)) (単位：秒)
	trn_rollback_information_put	トランザクションブランチロールバック時にロールバック情報を取得するかどうかを指定	no self remote all
	trn_limit_time	トランザクションブランチ最大実行可能時間	《符号なし整数》 ((0~65535)) (単位：秒)
	trn_rollback_response_receive	ロールバック完了通知を受信するかどうかを指定	《Y》 N
	trn_partial_recovery_type	UAP 障害時のトランザクション同期点処理方式	type1 type2 type3
	rap_inquire_timeout_message	クライアントからの要求最大待ち時間が満了したことを検知したとき、エラーメッセージを出力するかどうかを指定	《Y》 N
	rap_connection_assign_type	ダイナミックコネクションスケジュール機能を使用するかどうかを指定	dynamic 《static》
	rap_max_client	rap リスナーに同時接続する最大クライアント数	《符号なし整数》 ((128~1024)) 《256》
	rap_notify	rap クライアントマネージャに対する起動通知を行うかどうかを指定	Y 《N》
	rap_client_manager_node	rap クライアントが存在する OpenTP1 システムのホスト名	〈1~255 文字の識別子〉
		rap クライアントの存在する OpenTP1 システムの rap クライアントマネージャのポート番号	《符号なし整数》 ((1~65535))
	rap_max_buff_size	ソケットウィンドウサイズ	《符号なし整数》 ((4~2147483647)) 《4》 (単位：キロバイト)
	rap_io_retry_interval	電文送受信時のリトライ間隔	《符号なし整数》 ((1~999)) 《35》 (単位：ミリ秒)

形式	オペランド	定義内容	指定値
set	rap_sock_count	ソケット生成処理のリトライ回数	〈符号なし整数〉 ((0~65535)) 《1》
	rap_sock_interval	ソケット生成処理のリトライ間隔	〈符号なし整数〉 ((15~500)) 《30》 (単位：ミリ秒)
	rap_connect_retry_count	コネクション確立処理のリトライ回数	〈符号なし整数〉 ((8~2147483647)) 《8》
	rap_connect_retry_interval	コネクション確立処理のリトライ間隔	〈符号なし整数〉 ((10~999)) 《100》 (単位：ミリ秒)
	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〉 ((SOMAXCONN~2147483647)) 《SOMAXCONN》
	rap_msg_output_interval	クライアント数表示メッセージの出力間隔	〈符号なし整数〉 ((0~32767)) 《0》 (単位：分)
	rap_recovery_server	リカバリ要求用待機 rap サーバ数	〈符号なし整数〉 ((0~rap_parallel_server オペランドの指定値-1)) 《0》
	rap_connect_interval	コネクション確立処理の間隔	〈符号なし整数〉 ((0~999)) 《40》 (単位：ミリ秒)
	rpc_extend_function	RPC サービスの機能拡張レベル	〈16 進数字〉 ((00000000~0000000F)) 《00000000》
	max_socket_descriptors	ソケット用ファイル記述子の最大数	〈符号なし整数〉 ((32~2032))
	trn_completion_limit_time	トランザクション完了限界時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	rap_message_id_change_level	メッセージ ID の変更レベルを指定して、エラーメッセージの種類の E を W に変更	〈符号なし整数〉 ((0~2))
	rap_term_disconnect_time	rap リスナー終了時のコネクション切断待ち時間	〈符号なし整数〉 ((0~3600)) 《0》 (単位：秒)

形式	オペランド	定義内容	指定値
set	rap_stay_watch_time	rap サーバ割り当て待ち要求の滞留監視時間	〈符号なし整数〉 ((0~65535)) 《30》 (単位：秒)
	rap_stay_warning_interval	滞留警告メッセージの出力間隔	〈符号なし整数〉 ((3~65535)) 《180》 (単位：秒)
	log_audit_out_suppress	rap リスナーおよび rap サーバから出力される監査ログを抑止するかどうかを指定	Y 《N》
	log_audit_message	監査ログを取得する項目のメッセージ ID	〈符号なし整数〉 ((33400~99999))
	ipc_sockctl_highwater	ソケットの一時クローズ処理開始数パーセンテージ	〈符号なし整数〉 ((0~100))
		ソケットの一時クローズ処理非対象数パーセンテージ	〈符号なし整数〉 ((0~100))
	ipc_sockctl_watchtime	ソケット再利用可能監視時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	scs_prf_trace_level	リモート API 機能における性能検証用トレース取得レベル	〈符号なし整数〉 ((00000000~00000009)) 《00000001》
	rap_extend_function	リモート API 機能の機能拡張レベル	〈16 進数字〉 ((00000000~00000001)) 《00000000》
	watch_time	最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)

2.2.33 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、次の表に示します。

表 2-34 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	定義内容	指定値
set	rap_client_manager_port	rap クライアントマネージャのポート番号	〈符号なし整数〉 ((5001~65535))

形式	オペランド	定義内容	指定値
set	rap_listen_inf	rap リスナーを起動する OpenTP1 ノードのノード識別子	〈4 文字の識別子〉
		rap リスナーのポート番号	〈符号なし整数〉 ((5001～65535))
		リモート API 機能によるサービスの受信口になるホスト名	〈1～255 文字の英数字, ピリオドおよびハイフン〉
		リモート API 機能によるサービスの受信口になるポート番号	〈符号なし整数〉 ((1～65535))
	uid	OpenTP1 管理者のユーザ識別子	〈符号なし整数〉 ((0～4294967294))
	log_audit_out_suppress	rap クライアントマネージャから出力する監査ログを抑止するかどうかを指定	Y 《N》
	log_audit_message	監査ログを取得する項目のメッセージ ID	〈符号なし整数〉 ((33400～99999))
	rap_watch_time	メッセージ送受信最大監視時間	〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)

2.2.34 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、次の表に示します。

表 2-35 性能検証用トレース定義の内容

形式	オペランド	定義内容	指定値
set	prf_file_size	性能検証用トレース情報ファイルのサイズ	〈符号なし整数〉 ((1024～1048576)) 《10240》 (単位：キロバイト)
	prf_information_level	トレースファイル関連のメッセージの出力レベル	《1》 0
	prf_file_count	性能検証用トレース情報ファイルの世代数	〈符号なし整数〉 ((3～256)) 《3》
	prf_trace_backup	トレースファイルのバックアップを取得するかどうかを指定	《Y》 N
	prf_buff_size	性能検証用トレースバッファサイズ	〈符号なし整数〉 ((96～131072)) 《10240》 (単位：キロバイト)

2.2.35 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、次の表に示します。

表 2-36 XAR 性能検証用トレース定義の内容

形式	オペランド	定義内容	指定値
set	prf_file_size	XAR 性能検証用トレース情報ファイルのサイズ	〈符号なし整数〉 ((1024~1048576)) 《10240》 (単位：キロバイト)
	prf_information_level	XAR 性能検証用トレース関連のメッセージの表示レベルを指定	《1》 0
	prf_file_count	XAR 性能検証用トレース情報ファイルの世代数	〈符号なし整数〉 ((3~256)) 《3》

2.2.36 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、次の表に示します。

表 2-37 JNL 性能検証用トレース定義の内容

形式	オペランド	定義内容	指定値
set	prf_file_size	JNL 性能検証用トレース情報ファイルのサイズ	〈符号なし整数〉 ((1024~1048576)) 《1024》 (単位：キロバイト)
	prf_file_count	JNL 性能検証用トレース情報ファイルの世代数	〈符号なし整数〉 ((3~256)) 《3》
	prf_trace_backup	JNL 性能検証用トレース情報ファイルのバックアップを取得するかどうかを指定	《Y》 N

2.2.37 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、次の表に示します。

表 2-38 LCK 性能検証用トレース定義の内容

形式	オペランド	定義内容	指定値
set	prf_file_size	LCK 性能検証用トレース情報ファイルのサイズ	〈符号なし整数〉 ((1024~1048576)) 《5120》 (単位：キロバイト)

形式	オペランド	定義内容	指定値
set	prf_information_level	LCK 性能検証用トレース関連のメッセージの表示レベル	《1》 0
	prf_file_count	LCK 性能検証用トレース情報ファイルの世代数	〈符号なし整数〉 ((3~256)) 《3》

2.2.38 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、次の表に示します。

表 2-39 TRN イベントトレース定義の内容

形式	オペランド	定義内容	指定値
set	prf_file_size	TRN イベントトレース情報ファイルのサイズ	〈符号なし整数〉 ((1024~1048576)) 《10240》 (単位：キロバイト)
	prf_information_level	TRN イベントトレース関連のメッセージの表示レベルを指定	《1》 0
	prf_file_count	TRN イベントトレース情報ファイルの世代数	〈符号なし整数〉 ((3~256)) 《3》

2.2.39 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、次の表に示します。

表 2-40 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	定義内容	指定値
set	rts_trcput_interval		統計情報の取得間隔	〈符号なし整数〉 ((10~86400)) 《600》 (単位：秒)
	rts_service_max		最大取得サービス数	〈符号なし整数〉 ((1~1000)) 《64》
	rts_item_max		最大取得項目数	〈符号なし整数〉 ((1~1000)) 《64》
	rts_log_file		取得した統計情報を RTS ログファイルに出力するかどうかを指定	《Y》 N
	rts_log_file_name		RTS ログファイル名	〈1~63 文字のパス名〉

形式	オペランド／コマンド	オプション	定義内容	指定値
set	rts_log_file_name		RTS ログファイル名	《\$DCDIR/spool/dcrtsinf/rtslog》
	rts_log_file_size		RTS ログファイルのサイズ	〈符号なし整数〉 ((1024~1048576)) 《1024》 (単位：キロバイト)
	rts_log_file_count		RTS ログファイルの世代数	〈符号なし整数〉 ((1~10)) 《3》
	rts_log_file_backup		RTS ログファイルのバックアップファイルを作成するかどうかを指定	《Y》 N
	rts_swap_message		RTS ログファイルのスワップメッセージ (KFCA32740-I メッセージ) を出力するかどうかを指定	Y 《N》
コマンド	rtsput	-u	取得するリアルタイム統計情報の取得対象種別を設定	sys srv svc obj
		-s	サーバ名	〈1~8 文字で先頭が英字の英数字〉
		-v	サービス名	〈1~31 文字の識別子〉
		-o	取得対象名 1	〈1~8 文字の文字列〉
		-b	取得対象名 2	〈1~63 文字の文字列〉
		-e	項目 ID	〈符号なし整数〉 ((1000~9999))
		-f	リアルタイム取得項目定義ファイル名	〈1~8 文字の識別子〉

2.2.40 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、次の表に示します。

表 2-41 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	定義内容	指定値
set	rts_cpd_collct_cpd	チェックポイントダンプ情報	チェックポイントダンプ取得イベントを取得するかどうかを指定	Y 《N》
	rts_cpd_validt_cpd		チェックポイントダンプ有効化イベントを取得するかどうかを指定	Y 《N》
	rts_jnl_buf_full	ジャーナル情報	バッファ満杯イベントを取得するかどうかを指定	Y 《N》

形式	オペランド	定義種別	定義内容	指定値
set	rts_jnl_wait_buf	ジャーナル情報	空きバッファ待ちイベントを取得するかどうかを指定	Y 《N》
	rts_jnl_jnl_output		ジャーナル出力（ブロック）イベントを取得するかどうかを指定	Y 《N》
	rts_jnl_io_wait		入出力待ちイベントを取得するかどうかを指定	Y 《N》
	rts_jnl_write		write イベントを取得するかどうかを指定	Y 《N》
	rts_jnl_swap		スワップイベントを取得するかどうかを指定	Y 《N》
	rts_jnl_jnl_input		ジャーナル入力イベントを取得するかどうかを指定	Y 《N》
	rts_jnl_read		read イベントを取得するかどうかを指定	Y 《N》
	rts_lck_lock_acqst	ロック情報	ロック取得イベントを取得するかどうかを指定	Y 《N》
	rts_lck_lock_wait		ロック待ちイベントを取得するかどうかを指定	Y 《N》
	rts_lck_deadlock		デッドロックイベントを取得するかどうかを指定	Y 《N》
	rts_nam_global_cache_hit	ネーム情報	グローバルキャッシュヒットイベントを取得するかどうかを指定	Y 《N》
	rts_nam_local_cache_hit		ローカルキャッシュヒットイベントを取得するかどうかを指定	Y 《N》
	rts_nam_lookup		サービス情報の検索回数イベントを取得するかどうかを指定	Y 《N》
	rts_nam_node_lookup		指定参照先ノードへのサービス検索の送信回数イベントを取得するかどうかを指定	Y 《N》
	rts_nam_node_lookup_response		指定参照先ノードへのサービス検索の応答受信回数イベントを取得するかどうかを指定	Y 《N》
	rts_osl_stamem_acq	共用メモリ管理情報	静的共用メモリの使用サイズイベントを取得するかどうかを指定	Y 《N》
	rts_osl_stamem_pol		静的共用メモリプールの必要最大サイズイベントを取得するかどうかを指定	Y 《N》
	rts_osl_dynmem_acq		動的共用メモリの使用サイズイベントを取得するかどうかを指定	Y 《N》

形式	オペランド	定義種別	定義内容	指定値
set	rts_osl_dynmem_pol	共用メモリ管理 情報	動的共用メモリプールの必要最大サイズイベントを取得するかどうかを指定	Y 《N》
	rts_prc_prc_genert	プロセス情報	プロセス生成イベントを取得するかどうかを指定	Y 《N》
	rts_prc_uap_abnml		UAP 異常終了イベントを取得するかどうかを指定	Y 《N》
	rts_prc_sys_abnml		システムサーバ異常終了イベントを取得するかどうかを指定	Y 《N》
	rts_prc_prc_term		プロセス終了イベントを取得するかどうかを指定	Y 《N》
	rts_prc_prc_num		起動プロセス数の監視イベントを取得するかどうかを指定	Y 《N》
	rts_que_read	メッセージ キュー情報	read メッセージイベントを取得するかどうかを指定	Y 《N》
	rts_que_write		write メッセージイベントを取得するかどうかを指定	Y 《N》
	rts_que_read_err		read エラーイベントを取得するかどうかを指定	Y 《N》
	rts_que_write_err		write エラーイベントを取得するかどうかを指定	Y 《N》
	rts_que_wait_buf		空きバッファ待ちイベントを取得するかどうかを指定	Y 《N》
	rts_que_real_read		実 read イベントを取得するかどうかを指定	Y 《N》
	rts_que_real_write		実 write イベントを取得するかどうかを指定	Y 《N》
	rts_que_delay_wrt		遅延書き込み（回数）イベントを取得するかどうかを指定	Y 《N》
	rts_que_delay_rec		物理ファイル単位の遅延書き込み（レコード）イベントを取得するかどうかを指定	Y 《N》
	rts_que_delay_msg		物理ファイル単位の遅延書き込み（メッセージ）イベントを取得するかどうかを指定	Y 《N》
	rts_rpc_rpc_call	RPC 情報	RPC コール（同期応答型）イベントを取得するかどうかを指定	Y 《N》
	rts_rpc_rpc_call_chained		RPC コール（連鎖 RPC 型）イベントを取得するかどうかを指定	Y 《N》

形式	オペランド	定義種別	定義内容	指定値
set	rts_rpc_usr_srv c	RPC 情報	ユーザサービス実行イベントを取得 するかどうかを指定	Y 《N》
	rts_rpc_rpc_ovr tim		RPC タイムアウトイベントを取得す るかどうかを指定	Y 《N》
	rts_scd_scd_wa it	スケジュール 情報	スケジュール待ちイベントを取得す るかどうかを指定	Y 《N》
	rts_scd_schedul e		スケジュールイベントを取得するか どうかを指定	Y 《N》
	rts_scd_using_b uf		メッセージ格納バッファプールの使 用中サイズイベントを取得するかど うかを指定	Y 《N》
	rts_scd_lack_bu f		メッセージ格納バッファプールの不 足でスケジュールできなかったメッ セージサイズイベントを取得するか どうかを指定	Y 《N》
	rts_scd_scd_sta y		スケジュール滞留イベントを取得す るかどうかを指定	Y 《N》
	rts_scd_svc_scd _wait		サービス単位のスケジュール待ちイ ベントを取得するかどうかを指定	Y 《N》
	rts_scd_svc_usi ng_buf		サービス単位のメッセージ格納バッ ファプールの使用中サイズイベント を取得するかどうかを指定	Y 《N》
	rts_scd_parallel		同時実行サービス数イベントを取得 するかどうかを指定	Y 《N》
	rts_trn_commit	トランザクショ ン情報	コミットイベントを取得するかどう かを指定	Y 《N》
	rts_trn_rollback		ロールバックイベントを取得するか どうかを指定	Y 《N》
	rts_trn_cmt_cm d		コマンドによるコミットイベントを 取得するかどうかを指定	Y 《N》
	rts_trn_rbk_cm d		コマンドによるロールバックイベン トを取得するかどうかを指定	Y 《N》
	rts_trn_haz_cm d		コマンドによるハザードイベントを 取得するかどうかを指定	Y 《N》
	rts_trn_mix_cm d		コマンドによるミックスイベントを 取得するかどうかを指定	Y 《N》
	rts_trn_branch		ブランチ実行時間イベントを取得す るかどうかを指定	Y 《N》

形式	オペランド	定義種別	定義内容	指定値
set	rts_trn_sync_point	トランザクション情報	ブランチ同期点処理の実行時間イベントを取得するかどうかを指定	Y 《N》
	rts_dam_read	DAM 情報	read イベントを取得するかどうかを指定	Y 《N》
	rts_dam_read_err		read エラーイベントを取得するかどうかを指定	Y 《N》
	rts_dam_write		write イベントを取得するかどうかを指定	Y 《N》
	rts_dam_write_err		write エラーイベントを取得するかどうかを指定	Y 《N》
	rts_dam_fj		FJ 出力回数イベントを取得するかどうかを指定	Y 《N》
	rts_dam_trn_branch		同時実行 DAM トランザクションブランチ数イベントを取得するかどうかを指定	Y 《N》
	rts_dam_cache_block		DAM キャッシュブロックの確保回数イベントを取得するかどうかを指定	Y 《N》
	rts_dam_shm_pool		DAM キャッシュ用の共用メモリの使用率イベントを取得するかどうかを指定	Y 《N》
	rts_tam_real_renew	TAM 情報	TAM ファイル実更新イベントを取得するかどうかを指定	Y 《N》
	rts_tam_real_renew_time		TAM ファイル実更新時間イベントを取得するかどうかを指定	Y 《N》
	rts_tam_rec_refer		コミット、ロールバック（レコード参照）イベントを取得するかどうかを指定	Y 《N》
	rts_tam_rec_renew		コミット、ロールバック（レコード更新）イベントを取得するかどうかを指定	Y 《N》
	rts_tam_read		read イベントを取得するかどうかを指定	Y 《N》
	rts_tam_read_err		read エラーイベントを取得するかどうかを指定	Y 《N》
	rts_tam_write		write イベントを取得するかどうかを指定	Y 《N》
	rts_tam_write_err		write エラーイベントを取得するかどうかを指定	Y 《N》

形式	オペランド	定義種別	定義内容	指定値
set	rts_xar_start	XA リソース サービス情報	アプリケーションサーバから RAP サーバへのトランザクション開始要 求の回数イベントを取得するかどうかを指定	Y 《N》
	rts_xar_start_err		アプリケーションサーバから RAP サーバへのトランザクション開始要 求でのエラーイベントを取得するかどうかを指定	Y 《N》
	rts_xar_call		アプリケーションサーバから RAP サーバへのサービス要求の回数イベ ントを取得するかどうかを指定	Y 《N》
	rts_xar_call_err		アプリケーションサーバから RAP サーバへのサービス要求でのエラー イベントを取得するかどうかを指定	Y 《N》
	rts_xar_end		アプリケーションサーバから RAP サーバへのトランザクション終了要 求の回数イベントを取得するかどうかを指定	Y 《N》
	rts_xar_end_err		アプリケーションサーバから RAP サーバへのトランザクション終了要 求でのエラーイベントを取得するかどうかを指定	Y 《N》
	rts_xar_prepare		アプリケーションサーバから RAP サーバへのトランザクションのプリ ペア処理要求の回数イベントを取得 するかどうかを指定	Y 《N》
	rts_xar_prepare_err		アプリケーションサーバから RAP サーバへのトランザクションのプリ ペア処理要求でのエラーイベントを 取得するかどうかを指定	Y 《N》
	rts_xar_commit		アプリケーションサーバから RAP サーバへのトランザクションのコ ミット処理要求の回数イベントを取 得するかどうかを指定	Y 《N》
	rts_xar_commit_err		アプリケーションサーバから RAP サーバへのトランザクションのコ ミット処理要求でのエラーイベント を取得するかどうかを指定	Y 《N》
	rts_xar_rollback		アプリケーションサーバから RAP サーバへのトランザクションのロー ルバック処理要求の回数イベントを 取得するかどうかを指定	Y 《N》

形式	オペランド	定義種別	定義内容	指定値
set	rts_xar_rollback_err	XA リソースサービス情報	アプリケーションサーバから RAP サーバへのトランザクションのロールバック処理要求でのエラーイベントを取得するかどうかを指定	Y 《N》
	rts_xar_recover		アプリケーションサーバから RAP サーバへのプリペア済み、またはヒューリスティック完了したトランザクション通知要求の回数イベントを取得するかどうかを指定	Y 《N》
	rts_xar_recover_err		アプリケーションサーバから RAP サーバへのプリペア済み、またはヒューリスティック完了したトランザクション通知要求でのエラーイベントを取得するかどうかを指定	Y 《N》
	rts_xar_forget		アプリケーションサーバから RAP サーバへのトランザクション破棄要求の回数イベントを取得するかどうかを指定	Y 《N》
	rts_xar_forget_err		アプリケーションサーバから RAP サーバへのトランザクション破棄要求でのエラーイベントを取得するかどうかを指定	Y 《N》
	rts_mcf_ap_scd_stay	MCF 情報	スケジュール待ち情報を取得するかどうかを指定	Y 《N》
	rts_mcf_ap_usr_srv		ユーザサービス実行情報を取得するかどうかを指定	Y 《N》
	rts_mcf_in_msg_scd_wait		論理端末単位に受信メッセージの処理待ち情報を取得するかどうかを指定	Y 《N》
	rts_mcf_out_msg_sync_scd_wait		同期型送信メッセージの処理待ち情報を取得するかどうかを指定	Y 《N》
	rts_mcf_out_msg_resp_scd_wait		問い合わせ応答型送信メッセージの処理待ち情報を取得するかどうかを指定	Y 《N》
	rts_mcf_out_msg_prio_scd_wait		優先分岐型送信メッセージの処理待ち情報を取得するかどうかを指定	Y 《N》
	rts_mcf_out_msg_norm_scd_wait		一般分岐型送信メッセージの処理待ち情報を取得するかどうかを指定	Y 《N》

形式	オペランド	定義種別	定義内容	指定値
set	rts_mcf_que_sc d_wait_num	MCF 情報	入力キューの滞留数の情報を取得するかどうかを指定	Y 《N》

2.2.41 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、次の表に示します。

表 2-42 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	定義内容	指定値
set	nice		プロセスの優先順位の変更	〈符号なし整数〉 ((0~39)) 《0》
	parallel_count		常駐プロセス数と最大プロセス数	〈符号なし整数〉 ((0~1024)) 《1》
	hold		UAP が異常終了した場合にサービスグループまたはサービスを閉塞するかどうかを指定	《Y》 N
	hold_recovery		サービスグループまたはサービスの閉塞状態を全面回復時に引き継ぐかどうかを指定	《Y》 N
	deadlock_priority		UAP のデッドロックの優先順位	〈符号なし整数〉 ((1~127)) 《64》
	schedule_priority		このサービスグループのスケジュールの優先順位	〈符号なし整数〉 ((1~16)) 《8》
	message_buflen		最大メッセージ長	〈符号なし整数〉 ((1024~31457280)) 《4096》 (単位：バイト)
	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)
	trn_expiration_time		トランザクションブランチの限界経過時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	trn_expiration_time_suspend		トランザクションブランチの時間監視の範囲	Y N F
	watch_next_chain_time		連鎖 RPC 間隔監視時間	〈符号なし整数〉 ((0~65535)) 《180》

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	watch_next_chain_time		連鎖 RPC 間隔監視時間	(単位：秒)
	atomic_update		サービスをトランザクションとして実行するかどうかを指定	《Y》 N
	receive_from		スケジュールキュー，UNIX ドメイン，またはインターネットドメインを使用するかどうかを指定	《queue》 socket none
	uap_trace_max		UAP トレース格納最大数	〈符号なし整数〉 ((0～4095)) 《32》
	uap_trace_file_put		UAP トレース情報をファイルに取得するかどうかを指定	Y N
	term_watch_time		連続異常終了限界経過時間	〈符号なし整数〉 ((0～32767)) 《30》 (単位：分)
	mcf_jnl_buff_size		MCF のジャーナルバッファの大きさ	〈符号なし整数〉 ((4096～131072)) (単位：バイト)
	type		このサービスグループの種別	《other》 MHP RAP
	balance_count		1 プロセスが処理するサービス要求数	〈符号なし整数〉 ((0～512)) 《3》
	uid		ユーザ識別子	〈符号なし整数〉 ((0～4294967294))
	auto_restart		UAP が異常終了した場合のサービスグループの処置	Y 《N》
	critical		UAP が異常終了した場合のシステムの処置	Y 《N》
	lck_wait_priority		排他待ちの優先順位	〈符号なし整数〉 ((0～127)) 《0》
	mcf_psv_id		アプリケーション起動プロセス識別子	〈16 進数〉 ((01～ff))
	trn_cpu_time		トランザクションブランチが同期点処理までに使用できる CPU 時間	〈符号なし整数〉 ((0～65535)) (単位：秒)
	service_hold		サービス単位に閉塞管理するかどうかを指定	Y 《N》
	service_priority_control		サービス要求単位の優先順位に従ってスケジュールするかどうかを指定	Y 《N》

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	message_cell_size		スケジュールメッセージ格納セルサイズ	〈符号なし整数〉 ((512～31457280)) 《512》 (単位：バイト)
	max_socket_msg		ソケット受信型サーバが受信するメッセージ数の最大値	〈符号なし整数〉 ((1～500)) 《100》
	max_socket_msglen		ソケット受信型サーバが受信するメッセージ長の最大値	〈符号なし整数〉 ((1～30270)) 《10240》 (単位：キロバイト)
	trf_put		トランザクションが出力したジャーナルをトランザクションリカバリジャーナルファイルへ出力するかどうかを指定	Y 《N》
	mcf_mgrid		アプリケーション起動プロセスの属する MCF マネジャ識別子	〈識別子〉 ((A～Z, a～z)) 《A》
	mcf_service_max_count		MCF 通信関数発行回数上限値	〈符号なし整数〉 ((0～65535))
	trn_statistics_item		トランザクションブランチの統計情報項目	nothing base executiontime cputime
	node_down_restart		ユーザサーバを自動起動するかどうかを指定	《Y》 N
	rpc_response_statistics		レスポンス統計情報を取得するかどうかを指定	Y 《N》
	server_type		サービス関数の呼び出し時に従うパラダイム	《"betran"》 "xatmi" "xatmi_cbl"
	trn_rm_open_close_scope		xa_open 関数および xa_close 関数の発行タイミング	process transaction
	trn_optimum_item		トランザクションの最適化項目	base asyncprepare
	purge_msgget		サービスグループに割り当てた OS のメッセージキューを解放するかどうかを指定	Y 《N》
	cancel_normal_terminate		dcsvstop コマンドによる正常終了を抑制するかどうかを指定	Y 《N》
	prc_abort_signal		サーバのアボート用シグナル番号	〈符号なし整数〉 ((1～128)) 《3》
	rpc_service_retry_count		サービスリトライ機能でサービス関数をリトライする回数の最大値	〈符号なし整数〉 ((0～65535)) 《0》
	rpc_extend_function		RPC サービスの機能拡張レベル	〈16 進数字〉

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	rpc_extend_function		RPC サービスの機能拡張レベル	((00000000～0000000F)) 《00000000》
	max_socket_descriptors		ソケット用ファイル記述子の最大数	〈符号なし整数〉 ((32～2032))
	max_open_fds		UAP プロセスでアクセスするファイルおよびパイプの最大数	〈符号なし整数〉 ((16～2016)) 《50》
	service_term_watch_time		連続サービス異常終了限界経過時間	〈符号なし整数〉 ((0～32767)) 《0》 (単位：分)
	termed_after_service		サービス終了時に負荷が減少している場合に非常駐プロセスを終了させるかどうかを指定	Y 《N》
	xat_trn_expiration_time		他システムとの同期点処理の限界経過監視時間	〈符号なし整数〉 ((1～2147483647)) 《180》 (単位：秒)
	xat_osi_usr		OSI TP 通信用のスタブを使用するかどうかを指定	Y 《N》
	rpc_trace		RPC トレースを取得するかどうかを指定	Y N
	rpc_trace_name		RPC トレースを取得するファイル名	〈パス名〉
	rpc_trace_size		RPC トレースを取得するファイルのサイズ	〈符号なし整数〉 ((1024～2147483648)) (単位：バイト)
	trn_rollback_information_put		トランザクションブランチロールバック時にロールバック情報を取得するかどうかを指定	no self remote all
	schedule_method		ユーザサーバのスケジューリング方式	《msgque》 namedpipe
	service_wait_time		ユーザサーバの非常駐プロセスのサービス要求待ち時間	〈符号なし整数〉 ((1～4096)) (単位：秒)
	mcf_spp_oj		SPP での OJ の履歴情報を取得するかどうかを指定	《Y》 N
	adm_message_option		メッセージ出力指定	〈1 けたの 16 進数〉 《F》
	trn_watch_time		トランザクション同期点処理時の最大通信待ち時間	〈符号なし整数〉 ((1～65535)) (単位：秒)

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	trn_limit_time		トランザクションブランチ最大実行可能時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	trn_rollback_response_receive		ロールバック完了通知を受信するかどうかを指定	Y N
	trn_partial_recovery_type		UAP 障害時のトランザクション同期点処理方式	type1 type2 type3
	rpc_destination_mode		サービスの送信先を決定する方法	namdonly namd 《definition》
	rpc_rap_auto_connect		UAP とリモート API 制御プロセスの間のコネクションを自動的に管理するかどうかを指定	《Y》 N
	rpc_rap_inquire_time		リモート API 機能を使用して要求するサービスの問い合わせ間隔最大時間	〈符号なし整数〉 ((0~1048575)) 《0》 (単位：秒)
	rpc_request_cancel_for_timeout		クライアント UAP の応答待ち時間をサーバに意識させるかどうかを指定	《Y》 N
	status_change_when_termining		次回再開始時に最終的な状態の変化を反映するかどうかを指定	《Y》 N
	service_expiration_time		サービス関数開始から終了までの実行監視時間	〈符号なし整数〉 ((0~65535)) 《0》 (単位：秒)
	multi_schedule		マルチスケジューラ機能を使用してスケジューリングするかどうかを指定	Y 《N》
	make_queue_on_starting		非常駐のサービスグループに割り当てるスケジュールキューをサーバ起動時に割り当てるかどうかを指定	Y 《N》
	loadcheck_interval		このサービスグループの負荷レベルを監視するインタバル	〈符号なし整数〉 ((0~65535))
	levelup_queue_count		このサービスグループの負荷レベルを判断するサービス要求滞留数 (up)	〈符号なし整数〉 ((0~32767))
	leveldown_queue_count		このサービスグループの負荷レベルを判断するサービス要求滞留数 (down)	〈符号なし整数〉 ((0~32767))
	ipc_sockctl_highwater		ソケットの一時クローズ処理開始数パーセンテージ／ソケットの一時クローズ処理非対象数パーセンテージ	〈符号なし整数〉 ((0~100)) 《100, 0》

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	ipc_sockctl_watchtime		ソケット再利用可能監視時間	〈符号なし整数〉 ((0~65535)) 《180》 (単位：秒)
	ipc_conn_interval		コネクション確立監視時間	〈符号なし整数〉 ((8~65535)) 《8》 (単位：秒)
	ipc_send_interval		データ送信監視間隔	〈符号なし整数〉 ((5~32767)) 《5》 (単位：秒)
	ipc_send_count		データ送信監視回数	〈符号なし整数〉 ((1~32767)) 《5》
	ipc_header_rcv_time		通信制御データの受信監視時間	〈符号なし整数〉 ((5~32767)) 《10》 (単位：秒)
	rpc_send_retry_count		TCP/IP コネクションの接続時にエラーが発生した場合のリトライ回数	〈符号なし整数〉 ((0~65535)) 《0》
	rpc_send_retry_interval		TCP/IP コネクションの接続時にエラーが発生した場合のリトライ間隔	〈符号なし整数〉 ((0~300000)) 《0》 (単位：ミリ秒)
	ipc_rcvbuf_size		TCP/IP の受信バッファサイズ	〈符号なし整数〉 ((0, 8192~1048576)) 《8192》 (単位：バイト)
	ipc_sndbuf_size		TCP/IP の送信バッファサイズ	〈符号なし整数〉 ((0, 8192~1048576)) 《8192》 (単位：バイト)
	ipc_listen_sockbufset		TCP/IP の送受信バッファサイズをlisten ソケットに設定するかどうかを指定	Y 《N》
	polling_control_data		一時クローズ処理要求が到着していないかどうかを検査	Y 《N》
	thread_yield_interval		ソケットの再利用指示を受信できる契機を与えるインタバル時間	〈符号なし整数〉 ((1~86400)) 《90》 (単位：秒)
	groups		サービスグループのグループアクセスリストを設定	〈符号なし整数〉 ((0~4294967294))
	loadlevel_message		負荷レベル通知メッセージを出力するかどうかを指定	Y 《N》 A

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	ipc_backlog_count		コネクション確立要求を格納するキューの長さ	〈符号なし整数〉 ((0~4096)) 《0》
	rpc_buffer_pool_max		プーリングするバッファ数	〈符号なし整数〉 ((1~64)) 《64》
	schedule_delay_limit		スケジュール遅延限界経過時間	〈符号なし整数〉 ((0~32767)) 《0》 (単位：秒)
	schedule_delay_abort		スケジュール遅延時にシステムダウンするかどうかを指定	Y 《N》
	rap_autoconnect_con_error_msg		API の代理実行要求時， rap サーバとのコネクションの切断を検知した場合に， エラーメッセージを出力するかどうかを指定	《Y》 N
	core_shm_suppress		コアファイルへの共用メモリダンプの出力を抑止するかどうかを指定	Y 《N》
	xat_connect_resp_time		通信イベント処理用 SPP のアソシエーション確立の最大応答待ち時間	〈符号なし整数〉 ((0~65535)) 《180》 (単位：秒)
	scd_poolfull_check_interval		メッセージ格納バッファプールのメモリ不足が連続して発生した場合に， KFCA00853-E を出力するインタバル時間	〈符号なし整数〉 ((0~32767)) 《0》 (単位：秒)
	scd_poolfull_check_count		メッセージ格納バッファプールのメモリ不足が連続して発生した場合に， KFCA00853-E メッセージを出力する判断値	〈符号なし整数〉 ((1~32767)) 《10》 (単位：回)
	scd_pool_warning_use_rate		警告メッセージを出力する判断値になるメッセージ格納バッファプールの使用率の上限値	〈符号なし整数〉 ((0~99)) 《0》 (単位：%)
	scd_pool_warning_interval		メッセージ格納バッファプール使用率が超過したときの警告メッセージ出力インタバル時間	〈符号なし整数〉 ((0~32767)) 《0》 (単位：秒)
	ipc_tcpnodelay		Nagle アルゴリズムを無効にするかどうかを指定	《Y》 N
	stay_watch_queue_count		スケジュールキューの滞留監視判定を開始する際の判断になるサービス要求滞留数	〈符号なし整数〉 ((0~32767)) 《0》

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	stay_watch_check_rate		スケジュールキューの滞留監視判定処理で使用するサービス要求の処理率	〈符号なし整数〉 ((1~100)) (単位：%)
	stay_watch_abort		スケジュールキューの滞留監視判定式を満たした場合に、OpenTP1 をシステムダウンさせるかどうかを指定	Y 〈N〉
	stay_watch_start_interval		スケジュールキューの滞留監視インタバル時間	〈符号なし整数〉 ((1~32767)) 《10》 (単位：秒)
	stay_watch_check_interval		スケジュールキューの滞留監視判定インタバル時間	〈符号なし整数〉 ((1~65534)) 《10》 (単位：秒)
	trn_completion_limit_time		トランザクション完了限界時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	rap_message_id_change_level		メッセージ ID の変更レベルを指定して、エラーメッセージの種類の E を W に変更	〈符号なし整数〉 ((0~2))
	log_audit_out_suppress		監査ログを抑止するかどうかを指定	Y 〈N〉
	log_audit_message		監査ログを取得する項目のメッセージ ID	〈符号なし整数〉 ((33400~99999))
	mcf_prf_trace		ユーザサーバごとに MCF 性能検証用トレース情報を取得するかどうかを指定	《Y》 N
	scd_refresh_process		一つのサービス要求ごとに実行するプロセスを起動し直すかどうかを指定	Y 〈N〉
	rap_extend_function		リモート API 機能の機能拡張レベル	〈16 進数字〉 ((00000000~00000001)) 《00000000》
	prc_coredump_filter		core ファイルに共有メモリを含めるかの判断値	〈符号なし整数〉 ((1, 3))
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
コマンド	trnrmid	-n	リソースマネージャ名	〈1~31 文字の識別子〉
		-i	リソースマネージャ拡張子	〈1~2 文字の識別子〉

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
コマンド	trnrmid	-k	リソースマネージャをリソースマネージャ接続先選択機能の対象とする場合に指定するかどうかを指定	なし。
	scdbufgrp	-g	スケジュールバッファグループ名	〈1～8 文字の識別子〉
		-s	共用メッセージ格納バッファ使用制限サイズ	〈符号なし整数〉 ((512～1610612736)) (単位：バイト)
		-p	共用メッセージ格納バッファ使用制限率	〈符号なし整数〉 ((1～100)) (単位：%)
	scdmulti	-g	マルチスケジューラグループ名	〈1～8 文字の識別子〉 《scdmltgp》
	scdsvcdef	-c	サービス名	〈1～31 文字の識別子〉
		-p	同時実行可能なサービス数	〈符号なし整数〉 ((1～1024))
		-n	キューイング可能なサービス要求数	〈符号なし整数〉 ((1～65535))
		-l	キューイング可能なメッセージ格納バッファプール長	〈符号なし整数〉 ((512～31457280)) (単位：バイト)
putenv	任意		環境変数名 環境変数値	〈文字列〉
	XAT_CONNECT_RESP_TIME		通信イベント処理用 SPP のアソシエーション確立の最大応答待ち時間	〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)
dcputenv	任意		環境変数名 環境変数値	〈文字列〉

2.2.42 ユーザーサービス定義の内容

ユーザーサービス定義の内容を、次の表に示します。

表 2-43 ユーザー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	定義内容	指定値
set	service_group		サービスグループ名	〈1～31 文字の識別子〉

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	module		このサービスグループを実行する実行形式プログラム名	〈1～14 文字の識別子〉
	service		このサービスグループに属するサービス名とそのサービスを提供するエントリポイント名の組	〈1～31 文字の識別子〉
			UAP 共用ライブラリ名	〈1～255 文字のパス名〉
	nice		プロセスの優先順位の変更	〈符号なし整数〉 ((0～39))
	parallel_count		常駐プロセス数と最大プロセス数	〈符号なし整数〉 ((0～1024))
	hold		UAP が異常終了した場合にサービスグループまたはサービスを閉塞するかどうかを指定	Y N
	hold_recovery		サービスグループまたはサービスの閉塞状態を全面回復時に引き継ぐかどうかを指定	Y N
	deadlock_priority		UAP のデッドロックの優先順位	〈符号なし整数〉 ((1～127))
	schedule_priority		このサービスグループのスケジュールの優先順位	〈符号なし整数〉 ((1～16))
	message_buflen		最大メッセージ長	〈符号なし整数〉 ((1024～31457280)) (単位：バイト)
	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)
	trn_expiration_time		トランザクションプランチの限界経過時間	〈符号なし整数〉 ((0～65535)) (単位：秒)
	trn_expiration_time_suspend		トランザクションプランチの時間監視の範囲	Y N F
	watch_next_chain_time		連鎖 RPC 間隔監視時間	〈符号なし整数〉 ((0～65535)) (単位：秒)
	atomic_update		サービスをトランザクションとして実行するかどうかを指定	Y N
	receive_from		スケジュールキュー，または UNIX ドメインもしくはインターネットドメインを使用するかどうかを指定	queue socket none
	uap_trace_max		UAP トレース格納最大数	〈符号なし整数〉 ((0～4095))

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	uap_trace_file_put		UAP トレース情報をファイルに取得するかどうかを指定	Y N
	term_watch_time		連続異常終了限界経過時間	〈符号なし整数〉 ((0~32767)) (単位：分)
	mcf_jnl_buff_size		MCF のジャーナルバッファの大きさ	〈符号なし整数〉 ((4096~131072)) (単位：バイト)
	type		このサービスグループの種別	other MHP RAP
	balance_count		1 プロセスが処理するサービス要求数	〈符号なし整数〉 ((0~512))
	uid		ユーザ識別子	〈符号なし整数〉 ((0~4294967294))
	auto_restart		UAP が異常終了した場合のサービスグループの処置	Y N
	critical		UAP が異常終了した場合のシステムの処置	Y N
	lck_wait_priority		排他待ちの優先順位	〈符号なし整数〉 ((0~127))
	mcf_psv_id		アプリケーション起動プロセス識別子	〈16 進数〉 ((01~ff))
	trn_cpu_time		トランザクションブランチが同期点処理までに使用できる CPU 時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	service_hold		サービス単位に閉塞管理するかどうかを指定	Y N
	service_priority_control		サービス要求単位の優先順位に従ってスケジュールするかどうかを指定	Y N
	message_cell_size		スケジュールメッセージ格納セルサイズ	〈符号なし整数〉 ((512~31457280)) (単位：バイト)
	max_socket_msg		ソケット受信型サーバが受信するメッセージ数の最大値	〈符号なし整数〉 ((1~500))
	max_socket_msglen		ソケット受信型サーバが受信するメッセージ長の最大値	〈符号なし整数〉 ((1~30270)) (単位：キロバイト)

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	trf_put		トランザクションが出力したジャーナルをトランザクションリカバリジャーナルファイルへ出力するかどうかを指定	Y N
	mcf_mgrid		アプリケーション起動プロセスの属する MCF マネージャ識別子	〈識別子〉 ((A～Z, a～z))
	mcf_service_max_count		MCF 通信関数発行回数上限値	〈符号なし整数〉 ((0～65535))
	trn_statistics_item		トランザクションブランチの統計情報項目	nothing base executiontime cputime
	node_down_restart		ユーザサーバを自動起動するかどうかを指定	Y N
	rpc_response_statistics		レスポンス統計情報を取得するかどうかを指定	Y N
	server_type		サービス関数の呼び出し時に従うパラダイム	"betran" "xatmi" "xatmi_cbl"
	trn_rm_open_close_scope		xa_open 関数および xa_close 関数の発行タイミング	process transaction
	trn_optimum_item		トランザクションの最適化項目	base asyncprepare
	purge_msgget		サービスグループに割り当てた OS のメッセージキューを解放するかどうかを指定	Y N
	cancel_normal_terminate		dcsvstop コマンドによる正常終了を抑制するかどうかを指定	Y N
	prc_abort_signal		サーバのアボート用シグナル番号	〈符号なし整数〉 ((1～128))
	rpc_service_retry_count		サービスリトライ機能でサービス関数をリトライする回数の最大値	〈符号なし整数〉 ((0～65535))
	rpc_extend_function		RPC サービスの機能拡張レベル	〈16 進数字〉 ((00000000～0000000F))
	max_socket_descriptors		ソケット用ファイル記述子の最大数	〈符号なし整数〉 ((32～2032))
	max_open_fds		UAP プロセスでアクセスするファイルおよびパイプの最大数	〈符号なし整数〉 ((16～2016))
	service_term_watch_time		連続サービス異常終了限界経過時間	〈符号なし整数〉 ((0～32767)) (単位：分)

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	termed_after_service		サービス終了時に負荷が減少している場合に非常駐プロセスを終了させるかどうかを指定	Y N
	xat_trn_expiration_time		他システムとの同期点処理の限界経過監視時間	〈符号なし整数〉 ((1～2147483647)) (単位：秒)
	xat_osi_usr		OSI TP 通信用のスタブを使用するかどうかを指定	Y N
	rpc_trace		RPC トレースを取得するかどうかを指定	Y N
	rpc_trace_name		RPC トレースを取得するファイル名	〈パス名〉
	rpc_trace_size		RPC トレースを取得するファイルのサイズ	〈符号なし整数〉 ((1024～2147483648)) (単位：バイト)
	trn_rollback_information_put		トランザクションブランチロールバック時にロールバック情報を取得するかどうかを指定	no self remote all
	schedule_method		ユーザサーバのスケジューリング方式	msgque namedpipe
	service_wait_time		ユーザサーバの非常駐プロセスのサービス要求待ち時間	〈符号なし整数〉 ((1～4096)) (単位：秒)
	mcf_spp_oj		SPP での OJ の履歴情報を取得するかどうかを指定	Y N
	adm_message_option		メッセージ出力指定	〈1 けたの 16 進数〉
	trn_watch_time		トランザクション同期点処理時の最大通信待ち時間	〈符号なし整数〉 ((1～65535)) (単位：秒)
	trn_limit_time		トランザクションブランチ最大実行可能時間	〈符号なし整数〉 ((0～65535)) (単位：秒)
	trn_rollback_response_receive		ロールバック完了通知を受信するかどうかを指定	Y N
	trn_partial_recovery_type		UAP 障害時のトランザクション同期点処理方式	type1 type2 type3
	rpc_destination_mode		サービスの送信先を決定する方法	namdonly namd definition
	rpc_rap_auto_connect		UAP とリモート API 制御プロセスの間のコネクションを自動的に管理するかどうかを指定	Y N

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	rpc_rap_inquire_time		リモート API 機能を使用して要求するサービスの問い合わせ間隔最大時間	〈符号なし整数〉 ((0~1048575)) (単位：秒)
	rpc_request_cancel_for_timeout		クライアント UAP の応答待ち時間をサーバに意識させるかどうかを指定	Y N
	status_change_when_termining		次回再開始時に最終的な状態の変化を反映するかどうかを指定	Y N
	service_expiration_time		サービス関数開始から終了までの実行監視時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	multi_schedule		マルチスケジューラ機能を使用してスケジューリングするかどうかを指定	Y N
	make_queue_on_starting		非常駐のサービスグループに割り当てるスケジュールキューをサーバ起動時に割り当てるかどうかを指定	Y N
	loadcheck_interval		このサービスグループの負荷レベルを監視するインタバル	〈符号なし整数〉 ((0~65535)) (単位：秒)
	levelup_queue_count		このサービスグループの負荷レベルを判断するサービス要求滞留数 (up)	〈符号なし整数〉 ((0~32767))
	leveldown_queue_count		このサービスグループの負荷レベルを判断するサービス要求滞留数 (down)	〈符号なし整数〉 ((0~32767))
	ipc_sockctl_highwater		ソケットの一時クローズ処理開始数パーセンテージ／ソケットの一時クローズ処理非対象数パーセンテージ	〈符号なし整数〉 ((0~100))
	ipc_sockctl_watchtime		ソケット再利用可能監視時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	ipc_conn_interval		コネクション確立監視時間	〈符号なし整数〉 ((8~65535)) (単位：秒)
	ipc_send_interval		データ送信監視間隔	〈符号なし整数〉 ((5~32767)) (単位：秒)
	ipc_send_count		データ送信監視回数	〈符号なし整数〉 ((1~32767))

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	ipc_header_rcv_time		通信制御データの受信監視時間	〈符号なし整数〉 ((5～32767)) (単位：秒)
	rpc_send_retry_count		TCP/IP コネクションの接続時にエラーが発生した場合のリトライ回数	〈符号なし整数〉 ((0～65535))
	rpc_send_retry_interval		TCP/IP コネクションの接続時にエラーが発生した場合のリトライ間隔	〈符号なし整数〉 ((0～300000)) (単位：ミリ秒)
	ipc_rcvbuf_size		TCP/IP の受信バッファサイズ	〈符号なし整数〉 ((0, 8192～1048576)) (単位：バイト)
	ipc_sndbuf_size		TCP/IP の送信バッファサイズ	〈符号なし整数〉 ((0, 8192～1048576)) (単位：バイト)
	ipc_listen_sockbufset		TCP/IP の送受信バッファサイズをlisten ソケットに設定するかどうかを指定	Y 《N》
	polling_control_data		一時クローズ処理要求が到着していないかどうかを検査	Y N
	thread_yield_interval		ソケットの再利用指示を受信できる契機を与えるインタバル時間	〈符号なし整数〉 ((1～86400)) (単位：秒)
	groups		サービスグループのグループアクセスリストを設定	〈符号なし整数〉 ((0～4294967294))
	loadlevel_message		負荷レベル通知メッセージを出力するかどうかを指定	Y N A
	ipc_backlog_count		コネクション確立要求を格納するキューの長さ	〈符号なし整数〉 ((0～4096))
	rpc_buffer_pool_max		プーリングするバッファ数	〈符号なし整数〉 ((1～64))
	schedule_delay_limit		スケジュール遅延限界経過時間	〈符号なし整数〉 ((0～32767)) (単位：秒)
	schedule_delay_abort		スケジュール遅延時にシステムダウンするかどうかを指定	Y N
	rap_autoconnect_con_error_msg		API の代理実行要求時、rap サーバとのコネクションが切断されていた	Y N

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	rap_autoconnect_con_error_msg		場合に、エラーメッセージを出力するかどうかを指定	Y N
	core_shm_suppress		コアファイルへの共用メモリダンプの出力を抑止するかどうかを指定	Y N
	xat_connect_resp_time		通信イベント処理用 SPP のアソシエーション確立の最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	scd_poolfull_check_interval		メッセージ格納バッファプールのメモリ不足が連続して発生した場合に、KFCA00853-E を出力するインタバル時間	〈符号なし整数〉 ((0~32767)) (単位：秒)
	scd_poolfull_check_count		メッセージ格納バッファプールのメモリ不足が連続して発生した場合に、KFCA00853-E メッセージを出力する判断値	〈符号なし整数〉 ((1~32767)) (単位：回)
	scd_pool_warning_use_rate		警告メッセージを出力する判断値になるメッセージ格納バッファプールの使用率の上限値	〈符号なし整数〉 ((0~99)) (単位：%)
	scd_pool_warning_interval		メッセージ格納バッファプール使用率が超過したときの警告メッセージ出力インタバル時間	〈符号なし整数〉 ((0~32767)) (単位：秒)
	ipc_tcpnodelay		Nagle アルゴリズムを無効にするかどうかを指定	Y N
	stay_watch_queue_count		スケジュールキューの滞留監視判定を開始する際の判断になるサービス要求滞留数	〈符号なし整数〉 ((0~32767))
	stay_watch_check_rate		スケジュールキューの滞留監視判定処理で使用するサービス要求の処理率	〈符号なし整数〉 ((1~100)) (単位：%)
	stay_watch_abort		スケジュールキューの滞留監視判定式を満たした場合に、OpenTP1 をシステムダウンさせるかどうかを指定	Y N
	stay_watch_start_interval		スケジュールキューの滞留監視インタバル時間	〈符号なし整数〉 ((1~32767)) (単位：秒)
	stay_watch_check_interval		スケジュールキューの滞留監視判定インタバル時間	〈符号なし整数〉 ((1~65534)) (単位：秒)

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
set	trn_completion_limit_time		トランザクション完了限界時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
	rap_message_id_change_level		メッセージ ID の変更レベルを指定して、エラーメッセージの種類の E を W に変更	〈符号なし整数〉 ((0~2))
	log_audit_out_suppress		監査ログを抑止するかどうかを指定	Y 《N》
	log_audit_message		監査ログを取得する項目のメッセージ ID	〈符号なし整数〉 ((33400~99999))
	mcf_prf_trace		ユーザサーバごとに MCF 性能検証用トレース情報を取得するかどうかを指定	《Y》 N
	scd_refresh_process		一つのサービス要求ごとに実行するプロセスを起動し直すかどうかを指定	Y N
	rap_extend_function		リモート API 機能の機能拡張レベル	〈16 進数字〉 ((00000000~00000001)) 《00000000》
	prc_coredump_filter		core ファイルに共有メモリを含めるかの判断値	〈符号なし整数〉 ((1, 3))
	watch_time		最大応答待ち時間	〈符号なし整数〉 ((0~65535)) (単位：秒)
コマンド	trnrmid	-n	リソースマネージャ名	〈1~31 文字の識別子〉
		-i	リソースマネージャ拡張子	〈1~2 文字の識別子〉
		-k	リソースマネージャをリソースマネージャ接続先選択機能の対象とするかどうかを指定	なし。
	scdbufgrp	-g	スケジュールバッファグループ名	〈1~8 文字の識別子〉
		-s	共用メッセージ格納バッファ使用制限サイズ	〈符号なし整数〉 ((512~1610612736)) (単位：バイト)
		-p	共用メッセージ格納バッファ使用制限率	〈符号なし整数〉 ((1~100)) (単位：%)
	scdmulti	-g	マルチスケジューラグループ名	〈1~8 文字の識別子〉 《scdmulti》

形式	オペランド／コマンド／パス	オプション	定義内容	指定値
コマンド	scdsvcdef	-c	サービス名	〈1～31 文字の識別子〉
		-p	同時実行可能なサービス数	〈符号なし整数〉 ((1～1024))
		-n	キューイング可能なサービス要求数	〈符号なし整数〉 ((1～65535))
		-l	キューイング可能なメッセージ格納バッファプール長	〈符号なし整数〉 ((512～31457280)) (単位：バイト)
putenv	任意		環境変数名 環境変数値	〈文字列〉
	DCFPL_CONNECT_RETRY_COUNT		コネクション確立処理をリトライする回数	〈符号なし整数〉 ((8～2147483647))
	DCFPL_CONNECT_RETRY_INTERVAL		コネクション確立処理のリトライ間隔	〈符号なし整数〉 ((10～999)) (単位：ミリ秒)
	XAT_CONNECT_RESP_TIME		通信イベント処理用 SPP のアソシエーション確立の最大応答待ち時間	〈符号なし整数〉 ((0～65535)) (単位：秒)
dcputenv	任意		環境変数名 環境変数値	〈文字列〉

3

システムサービス定義の詳細

システムサービス定義の詳細について、それぞれの定義ごとに説明します。

システム環境定義

形式

set 形式

```
[set mode_conf=AUTO|MANUAL1|MANUAL2]
[set static_shmpool_size=静的共用メモリの総量]
[set dynamic_shmpool_size=動的共用メモリの最大使用時の総量]
[set shmpool_attribute=free|fixed]
[set user_command=ユーザ環境設定コマンド]
[set server_count=最大サーバ数]
[set user_server_ha=Y|N]
[set system_terminate_watch_time=システム終了監視時間]
[set start_scheduling_timing=BEFORE|AFTER]
[set system_init_watch_time=システム初期化の待ち時間]
[set user_command_online=システム開始完了コマンド]
[set preend_warning_watch_time=終了準備警告監視時間]
[set user_command_online_tplmgr_id=Y|N]
[set default_value_option=システム定義のデフォルト値変更オプション]
```

コマンド形式

なし。

putenv 形式

```
[putenv DCCONFPATH 定義ファイルの格納ディレクトリ]
[putenv DCADMDEBUG 0|1]
[putenv DCUAPCONFPATH ユーザサービス定義ファイルまたはユーザサービスデフォルト定義ファイルの格納ディレクトリ]
```

dcputenv 形式

```
[dcputenv DCCONFPATH 定義ファイルの格納ディレクトリ]
[dcputenv DCUAPCONFPATH ユーザサービス定義ファイルまたはユーザサービスデフォルト定義ファイルの格納ディレクトリ]
```

機能

OpenTP1 システムの実行環境を定義します。

プロセスサービスが OpenTP1 システム開始時にシステム環境定義を読み込みます。

システム環境定義を変更した場合、OpenTP1 を正常終了して dcreset コマンドを実行してください。

説明

set 形式のオペランド

●mode_conf=AUTO|MANUAL1|MANUAL2

～《MANUAL2》

OpenTP1 システムの開始方法を指定します。

AUTO

自動開始になります。

MANUAL1

手動開始になります。ただし、前回の OpenTP1 システムが異常終了した場合は、自動的に再開始します。

MANUAL2

手動開始になります。

前回の終了モード（正常終了・強制正常終了・計画停止 A・計画停止 B・強制停止・異常終了）とここで指定する開始方法の組み合わせによって開始形態が決まります。

前回の終了モードと、mode_conf オペランドの指定値の組み合わせによる開始形態の違いを、次に示します。

開始形態の決定条件		開始形態	
前回の終了モード	mode_conf オペランドの指定値	開始方法	開始モード
正常終了	AUTO	手動※1	正常開始
	MANUAL1	手動	
	MANUAL2		
強制正常終了	AUTO	手動※1	正常開始
	MANUAL1	手動	
	MANUAL2		
計画停止 A	AUTO	手動※1	再開始
	MANUAL1	手動	再開始※3
	MANUAL2		
計画停止 B	AUTO	手動※1	再開始
	MANUAL1	手動	再開始※3
	MANUAL2		
強制停止	AUTO	手動※1	再開始
	MANUAL1	手動	再開始※3
	MANUAL2		

開始形態の決定条件		開始形態	
前回の終了モード	mode_conf オペランドの指定値	開始方法	開始モード
異常終了	AUTO	自動	再開始
	MANUAL1	自動※2	
	MANUAL2	手動	再開始※3

注

系切り替え構成の際の mode_conf の指定内容については、マニュアル「OpenTP1 解説」の系切り替えの手順に関する記述を参照してください。

注※1

OS 起動時は自動開始となります。

注※2

OS 起動時は手動開始となります。

注※3

dcstart -n コマンドで強制的な正常開始もできます。ただし、強制的に正常開始すると、前回仕掛り中の情報は失われます。

●static_shmpool_size=静的共用メモリの総量

～〈符号なし整数〉((0～1945600))《4096》(単位：キロバイト)

OpenTP1 システムで、システムサービスがシステム開始から終了までの間、占有する共用メモリ（以後、静的共用メモリといいます）の総量を指定します。

静的共用メモリの総量は、dynamic_shmpool_size 指定値との合計が、1945600 を超えないように指定する必要があります。

必要に応じて起動するシステムサービスがある場合は、その容量も含めて指定します。

なお、HP-UX 版の 32 ビット版の OpenTP1 を使用する場合、次の点に注意してください。

- 上限値の 1945600 を 1038336 に読み替えてください。
32 ビット版の HP-UX では、共用メモリのセグメントサイズの上限値が 1 ギガバイトです。OpenTP1 は、内部的に必要とするサイズ（最大 10 メガバイト）を指定したサイズに加算して共用メモリの確保要求をするため、静的共用メモリと動的共用メモリの合計値が 1038336 を超えた場合、確保要求をするサイズが 1 ギガバイトを超えてしまい OpenTP1 システムの起動に失敗することがあります。
- マルチ OpenTP1 での運用や、共用メモリを確保するほかのプログラムを同一マシン上で実行する場合は、共用メモリの合計サイズに注意してください。
システム全体で使用できる共用メモリのサイズは、HP-UX (IPF) の場合 2 ギガバイト、HP-UX (PA-RISC) の場合 1.75 ギガバイトです。

●dynamic_shmpool_size=動的共用メモリの最大使用時の総量

～ 〈符号なし整数〉 ((0～1945600)) 《4096》 (単位：キロバイト)

OpenTP1 システムで、システムサービスがワークエリアイメージで動的にアクセスする共用メモリ（以後、動的共用メモリといいます）の最大使用時の総量を指定します。

動的共用メモリの最大使用時の総量は、static_shmpool_size 指定値との合計が、1945600 を超えないように指定する必要があります。

必要に応じて起動するシステムサービスがある場合は、その容量も含めて指定します。

なお、HP-UX 版の 32 ビット版の OpenTP1 を使用する場合、上限値の 1945600 を 1038336 に読み替えてください。また、システム全体で利用できる共用メモリサイズに制限があります。詳細については、システム環境定義の static_shmpool_size オペランドの説明を参照してください。

●shmpool_attribute=free|fixed

～ 《free》

OpenTP1 システムでは、静的共用メモリと動的共用メモリを合わせて、システムサービス用共用メモリプールとして確保します。この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するかどうかを指定します。

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しません。

実装するメモリ容量によっては、共用メモリのページングが発生するため、性能に悪影響を及ぼす場合があります。

fixed

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します。

共用メモリのページングが発生しないため、共用メモリのアクセスによる性能への悪影響を防止できます。

ただし、実装するメモリ容量によっては、共用メモリ以外のテキストやデータセグメントなどのページングが頻繁に発生する場合があるため、実装するメモリや全使用容量に対する共用メモリプールの占める割合を検討してください。

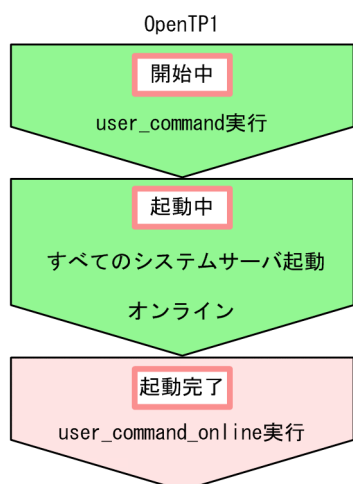
fixed は、OS が HP-UX または Solaris の場合だけ指定できます。なお、Solaris の場合、ご使用の環境によっては fixed を指定できないことがあります。詳細については、「リリースノート」を参照してください。

●user_command=ユーザ環境設定コマンド

～ 〈パス名〉

OpenTP1 の開始処理の最初に、実行するユーザ環境設定コマンドをパス名で指定します。ユーザ環境設定コマンドを指定すると、共用メモリの確保やファイルの初期化など、UAP 全体の環境を OpenTP1 の開始処理の最初に設定できます。

user_command オペランド, および user_command_online オペランドに指定されたコマンドの, 実行タイミングを次に示します。



(凡例)

 : OpenTP1の状態

ユーザ環境設定コマンドの実行環境は次のとおりです。

- ユーザ環境設定コマンドは, 系切り替え機能の使用の有無, および実行系, 待機系に関係なく, OpenTP1 の開始処理の最初に実行します。
- ユーザ ID/グループ ID はスーパーユーザの UID/GID (0/0) です。
- スーパーユーザで実行される可能性があるので, システム環境定義ファイル (\$DCDIR/conf/env) および user_command オペランドで指定したファイルが変更されないように, それぞれのファイルの書き込み権限を削除しておくことをお勧めします。
- シェルを使用する場合は, ボーンシェルだけ使用できます。
- 標準入力, 標準出力, および標準エラー出力は, OpenTP1 によってリダイレクトされます。
- オンラインで使用する機能は, 使用できません。
- ユーザ環境設定コマンドの起動が失敗した場合, および 0 以外で exit した場合は, システムダウンします。
- コアファイルは, _usrcmdN (N:1~3) に退避されます。ただし, ユーザ環境設定コマンドでカレントワーキングディレクトリを変更した場合, コアファイルは退避されません。
- 標準入力, 標準出力, 標準エラー出力は O_NONBLOCK 属性で open されています。

●server_count=最大サーバ数

～〈符号なし整数〉((32~9999))《64》

同一 OpenTP1 ノード内でシステム終了までに起動するサーバ数を指定します。

サーバ数は、OpenTP1 システムで動作する全システムサービス、および dcsvstart コマンドで指定するユーザサーバの数の合計値を指定します。同一名称のユーザサーバを複数回起動しても 1 として数えます。マルチサーバの場合は、サーバ数 1 と見なします。

server_count の指定値は、OS リソース（メモリ、ディスク容量、ポート番号など）に影響を与えます。このオペランドを変更した場合、「7.3.1 ユーザサーバの追加」を参照し、定義の見直しや OpenTP1 ファイルの再見積もりの必要有無を確認してください。

なお、多くのユーザサーバを起動する場合、dcsvstart コマンドで起動すると、ユーザサーバの起動時間が長くなります。ユーザサービス構成定義を使用して、並列にユーザサーバを起動することを推奨します。

また、多くのユーザサーバを起動している場合、OpenTP1 の停止までに時間が掛かるため、システム環境定義の system_terminate_watch_time オペランドの指定値を見直してください。

server_count の値を変更した場合は、OpenTP1 が停止した状態で dcsetup コマンドまたは dcreset コマンドを実行して定義の変更を OpenTP1 に反映してください。

●user_server_ha=Y|N

～《N》

系切り替え時、システムサーバの起動を待たないで、待機系のユーザサーバを起動するかどうかを指定します。

Y

システムサーバの起動を待たないで、ユーザサーバを起動します。

N

システムサーバの起動を待って、ユーザサーバを起動します。

●system_terminate_watch_time=システム終了監視時間

～〈符号なし整数〉((0～65535))《3600》(単位：秒)

dcstop コマンドで OpenTP1 を終了する時の監視時間を指定します。指定した監視時間内に OpenTP1 が終了しない場合は、OpenTP1 は強制停止します。

0 を指定した場合、終了時間は監視されません。

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 を指定して自動アンロード機能を使用する場合は、アンロードに掛かる時間を考慮して、system_terminate_watch_time オペランドの値を指定してください。

●start_scheduling_timing=BEFORE|AFTER

～《AFTER》

システム開始時に、RPC の受け付けの開始時期を指定します。

BEFORE

全ユーザサーバの起動前に、RPC の受け付けを開始します。BEFORE を指定した場合、RPC がユーザサーバ未起動のためにエラーになることがあります。

AFTER

全ユーザサーバの起動後に、RPC の受け付けを開始します。AFTER を指定した場合、全ユーザサーバが起動されるまで RPC はエラーになります。

このオペランドに BEFORE を指定した場合、ユーザサービス定義、ユーザサービスデフォルト定義の hold_recovery オペランドの指定に関係なく、システム全面回復時にユーザサーバの閉塞状態は引き継がれません。システム全面回復時にユーザサーバの閉塞状態を引き継ぐ場合は、スケジューラサービス定義の scd_hold_recovery オペランドに F を指定してください。各オペランドの指定による閉塞状態の引き継ぎについては、スケジューラサービス定義の [scd_hold_recovery](#) オペランドを参照してください。

●system_init_watch_time=システム初期化の待ち時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

OpenTP1 のシステム開始処理のうち、user_command の実行を含めたシステム初期化処理（システムサービス開始前の初期化処理）の待ち時間の最大値を指定します。

0 を指定した場合は、システム初期化処理が終わるまで無限に待ち続けます。

省略した場合は、システム共通定義の watch_time オペランドの値を仮定します。

●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パス名で指定します。システム開始完了コマンドを指定すると、dcstats コマンドなどの再開で引き継がれないプロセスを OpenTP1 の起動が完了するときに起動できます。

dcstart コマンドは、システム開始完了コマンドの完了を待ち合わせます。非同期に実行することで、dcstart コマンドを先に進めたい場合は、コマンドの最後に'&'を付けてください。

user_command_online オペランドに指定したプログラムが環境変数を必要とする場合、シェルスクリプトファイルに、プログラムが必要とする環境変数を宣言しておき、そのシェルスクリプトファイルを user_command_online オペランドに指定することをお勧めします。シェルスクリプトファイルに、プログラムが必要とする環境変数を宣言していないと、システム構成を変更するなどの要因で、環境変数が引き継がれなくなったとき、OpenTP1 の動作に影響を及ぼすことがあります。

システム開始完了コマンドの実行環境を次に示します。

- ユーザ ID およびグループ ID は次の ID で実行されます。なお、常に OpenTP1 管理者で実行したい場合は、システム環境定義の user_command_online_tplmngn_id オペランドを Y に指定してください。

- OpenTP1 管理者で dcstart コマンド実行した場合：OpenTP1 管理者の UID/GID
- 開始方法を自動開始で起動した場合※：スーパーユーザの UID/GID (0/0)
- dcmstart コマンドで開始した場合：スーパーユーザの UID/GID (0/0)
- 待機系が系切り替えによって起動した場合：OpenTP1 管理者の UID，スーパーユーザの GID (0)

注※

開始方法を自動開始にする方法については、マニュアル「OpenTP1 運用と操作」を参照してください。

- スーパーユーザで実行される可能性があるため、システム環境定義ファイル (\$DCDIR/conf/env) および user_command_online オペランドで指定したファイルが変更されないように、それぞれのファイルの書き込み権限を削除しておくことをお勧めします。
- シェルを使用する場合はボーンシェルだけ使用できます。
- 標準入力，標準出力，および標準エラー出力結果をファイルなどに出力する場合，シェルスクリプトファイル内でリダイレクトしてファイルに出力してください。
- オンラインで使用する機能が使用できます。
- システム開始完了コマンドの起動が失敗した場合，および 0 以外で exit した場合はシステムダウンします。系切り替え機能を使用した場合，システムダウンによって系切り替えが発生します。
ただし，システム開始完了コマンドを非同期に実行した場合はコマンドが 0 以外で exit してもシステムダウンしません。
- コアファイルは _usrcmdonN (N: 1~3) に退避されます。ただし，システム開始完了コマンドでカレントワーキングディレクトリを変更した場合，コアファイルは退避されません。

●preend_warning_watch_time=終了準備警告監視時間

～〈符号なし整数〉((0~65535))《180》(単位：秒)

dcstop コマンドの終了準備処理で警告メッセージ KFCA01859-W を出力するまでの時間を指定します。0 を指定した場合は時間監視をしません。

●user_command_online_tp1mgr_id=Y|N

～《N》

user_command_online オペランドに指定したシステム開始完了コマンドを OpenTP1 管理者の UID/GID で実行するかどうかを指定します。

Y

システム開始完了コマンドを OpenTP1 管理者の UID/GID で実行します。

N

システム開始完了コマンドの UID/GID は，次に示す UID/GID で実行されます。

- OpenTP1 管理者で dcstart コマンド実行した場合：OpenTP1 管理者の UID/GID

- ・ 開始方法を自動開始で起動した場合※：スーパーユーザの UID/GID (0/0)
- ・ dcmstart コマンドで開始した場合：スーパーユーザの UID/GID (0/0)
- ・ 待機系が系切り替えによって起動した場合：OpenTP1 管理者の UID, スーパーユーザの GID (0)

注※

開始方法を自動開始にする方法については、マニュアル「OpenTP1 運用と操作」を参照してください。

●default_value_option=システム定義のデフォルト値選択オプション

～〈符号なし整数〉((0～1))《1》

システム定義オペランドのうち、次に示すオペランドについて 07-50 以降のデフォルト値で動作するか、07-50 より前のデフォルト値で動作するか指定します。

0

07-50 より前のバージョンでのデフォルト値で動作します。

1

07-50 以降のバージョンで推奨するデフォルト値で動作します。

この指定値によってデフォルト値が変更されるのは次に示すオペランドだけです。

デフォルト値が変更されるオペランド		採用されるデフォルト値	
		default_value_option オペランドの指定値	
		0	1
システム共通定義 (betranrc)	tmn_prf_trace_level	00000001	00000003
	ipc_listen_sockbufset	N	Y
	fil_prf_trace_delay_time	10	3
ロックサービス定義 (lck)	lck_deadlock_info	N	Y
スケジューラサービス定義 (scd)	ipc_tcpnodelay	N	Y
ログサービス定義 (log)	log_filesize	1024	10240
性能検証用トレース定義 (prf)	prf_file_size	1024	10240
	prf_buff_size	96	1024
ユーザサービスデフォルト定義 (usrcc)	ipc_tcpnodelay	N	Y
MCF マネージャ定義	mcfmcomn -i	inc	msg

デフォルト値が変更されるオペランド		採用されるデフォルト値	
		default_value_option オペランドの指定値	
		0	1
MCF マネージャ定義	mcfmuap -e segsize	512	32768
	mcfmuap -c order	function	commit
	mcfmuap -c noansreply	no	yes
	mcfmuap -c errevt_recovery	no	yes
MCF 通信構成定義 (プロトコル固有定義 (TP1/NET/ TCP/IP))	mcftalccn -f releaselog	1	2
	mcftalccn -f cnerrlog	1	2
システムサービス共通情報定義	mcf_prf_trace_level	00000000	00000001

コマンド形式

なし。

putenv 形式のオペランド

●DCCONFPATH 定義ファイルの格納ディレクトリ

～ 〈パス名〉《\$DCDIR/conf》

各定義ファイルが格納されているディレクトリの絶対パス名を 246 バイト以下で指定します。ただし、環境変数は指定できません。

標準とは別のディレクトリ下にある定義ファイルのディレクトリを変更するときなどに使用します。

ただし、システム環境定義ファイル'env'を格納するディレクトリは変更できません。DCCONFPATH を指定する場合でも、システム環境定義ファイル'env'だけは\$DCDIR/conf になければなりません。

●DCADMDEBUG 0|1

～ 《0》

デバッグ情報を取得するかどうかを定義します。

OpenTP1 では UAP がダウンした場合、OpenTP1 の障害情報を取得するために netstat コマンドを発行します。netstat コマンドは UAP が連続ダウンした場合には複数投入されるので、構成によっては CPU を使い過ぎてマシン負荷が大きくなる場合があります。CPU 利用率を抑えたいときは、このオペランドの指定を 1 にしてください。

0

サーバダウン時に取得する netstat の情報を出力します。

サーバダウン時に取得する netstat の情報を出力しません。

●DCUAPCONFPATH ユーザサービス定義ファイルまたはユーザサービスデフォルト定義ファイルの格納ディレクトリ

～ 〈パス名〉

特定のユーザサービス定義ファイルまたはユーザサービスデフォルト定義ファイルだけを、DCCONFPATH 環境変数で指定したディレクトリ以外のディレクトリに格納したい場合、そのディレクトリの絶対パス名を 246 バイト以下で指定します。ただし、環境変数は指定できません。

DCCONFPATH 環境変数と DCUAPCONFPATH 環境変数で指定したディレクトリの両方にユーザサービスデフォルト定義ファイルを格納していた場合、DCCONFPATH 環境変数で指定したディレクトリに格納しているユーザサービスデフォルト定義が優先されます。

dcputenv 形式のオペランド

●DCCONFPATH 定義ファイルの格納ディレクトリ

～ 〈パス名〉

各定義ファイルが格納されているディレクトリの絶対パス名を指定します。環境変数を指定した場合、その値も取得します。指定するパス名は、環境変数の値も含め 246 バイト以下としてください。

標準とは別のディレクトリ下にある定義ファイルのディレクトリを変更するときなどに使用します。

●DCUAPCONFPATH ユーザサービス定義ファイルまたはユーザサービスデフォルト定義ファイルの格納ディレクトリ

～ 〈パス名〉

特定のユーザサービス定義ファイルまたはユーザサービスデフォルト定義ファイルだけを、DCCONFPATH 環境変数で指定したディレクトリ以外のディレクトリに格納したい場合、そのディレクトリを絶対パス名で指定します。

ディレクトリ名に環境変数を指定した場合、その値も取得します。指定するパス名は、環境変数の値も含め 246 バイト以下としてください。

DCCONFPATH 環境変数と DCUAPCONFPATH 環境変数で指定したディレクトリの両方にユーザサービスデフォルト定義ファイルを格納していた場合、DCCONFPATH 環境変数で指定したディレクトリに格納しているユーザサービスデフォルト定義が優先されます。

注意事項

従来どおり、ユーザサービス定義ファイルが DCCONFPATH 環境変数に指定したディレクトリにある場合、そのディレクトリのユーザサービス定義ファイルが有効となります。

ログイン環境に、DCCONFPATH 環境変数および DCUAPCONFPATH 環境変数を定義している場合は、ここで定義したものと同一定義値にしてください。

システムサービス構成定義

形式

set 形式

```
[set uap_conf=Y|N]
[set dam_conf=Y|N]
[set tam_conf=Y|N]
[set que_conf=Y|N]
[set ha_conf=Y|N]
[set jar_conf=Y|N]
[set clt_conf=Y|N]
[set ist_conf=Y|N]
[set rmm_conf=Y|N]
[set xat_conf=Y|N]
[set mqa_conf=Y|N]
```

コマンド形式

```
[dcsvstart -m システムサービス名]
```

機能

同一マシンで一つの OpenTP1 システムとしてシステム起動時に一緒に開始するサービス群の構成を定義します。

説明

set 形式のオペランド

●uap_conf=Y|N

～《N》

このノードで OpenTP1 システム起動時に、一緒にユーザサーバを起動するかどうかを指定します。

Y

このノードで OpenTP1 システム起動時に、一緒にユーザサーバを起動します。

N

このノードで OpenTP1 システム起動時に、一緒にユーザサーバを起動しません。

このオペランドに Y を指定した場合、一緒に起動するユーザサーバをユーザサービス構成定義に定義しておく必要があります。

●dam_conf=Y|N

～《N》

このノードで DAM サービスを使用するかどうかを指定します。

Y

このノードで DAM サービスを使用します。

N

このノードで DAM サービスを使用しません。

●**tam_conf=Y|N**

～《N》

このノードで TAM サービスを使用するかどうかを指定します。

Y

このノードで TAM サービスを使用します。

N

このノードで TAM サービスを使用しません。

●**que_conf=Y|N**

～《N》

このノードでメッセージキューサービス（MCF サービス）を使用するかどうかを指定します。

Y

このノードでメッセージキューサービスを使用します。

N

このノードでメッセージキューサービスを使用しません。

●**ha_conf=Y|N**

～《N》

このノードで系切り替え機能を使用するかどうかを指定します。

Y

このノードで系切り替え機能を使用します。

N

このノードで系切り替え機能を使用しません。

●**jar_conf=Y|N**

～《N》

このノードでグローバルアーカイブジャーナルサービスを実行するかどうかを指定します。

Y

グローバルアーカイブジャーナルサービスを実行します。

N

グローバルアーカイブジャーナルサービスを実行しません。

このオペランドで Y を指定する場合は、TP1/Multi が必要です。

●clt_conf=Y|N

～《N》

このノードでクライアント拡張サービスを使用するかどうかを指定します。

Y

クライアント拡張サービスを使用します。

N

クライアント拡張サービスを使用しません。

CUP からクライアント拡張サービスを使用する場合は、Y を指定してください。

Y を指定する場合、クライアントサービス定義が必要となります。

クライアント拡張サービスは、CUP からトランザクションを起動したり、常設コネクションを確立したりする場合に指定します。

リモート API 機能を使用する場合には、このオペランドに Y を指定する必要はありません。

注

Y を指定する場合、ユーザサービスデフォルト定義の trnrmid 定義コマンドでリソースマネージャ拡張子を定義しないでください。定義すると、OpenTP1 が起動できなくなります。

●ist_conf=Y|N

～《N》

このノードで IST サービスを使用するかどうかを指定します。

Y

このノードで IST サービスを使用します。

N

このノードで IST サービスを使用しません。

●rmm_conf=Y|N

～《N》

このノードで RMM サービスを使用するかどうかを指定します。

Y

このノードで RMM サービスを使用します。

N

このノードで RMM サービスを使用しません。

●xat_conf=Y|N

～《N》

このノードで XATMI 通信サービスを使用するかどうかを指定します。

Y

このノードで XATMI 通信サービスを使用します。

N

このノードで XATMI 通信サービスを使用しません。

TP1/NET/OSI-TP-Extended を使用して、XATMI インタフェースで OSI TP 通信を実行する場合は、Y を指定してください。TCP/IP 通信だけを実行する場合は、N を指定してください。

●mqa_conf=Y|N

～《N》

このノードで MQA サービスを使用するかどうかを指定します。

Y

このノードで MQA サービスを使用します。

N

このノードで MQA サービスを使用しません。

コマンド形式

次ページ以降に記述しています。

dcsvstart

名称

システムサービスの起動

形式

[dcsvstart -m システムサービス名]

機能

このノードで使用する MCF サービスを起動します。

オプション

●-m システムサービス名

～ 〈1～8 文字の英数字〉

このノードで使用する MCF サービスの名称を指定します。

これによって、MCF サービスが起動します。

MCF サービス名には、先頭 3 文字が'_mu'で始まる名称を指定します。

この MCF サービス名は、MCF 定義オブジェクト生成ユーティリティ（mcfmgr）で生成する MCF マネージャ定義の出力オブジェクトファイル名です。

ユーザサービス構成定義

形式

set 形式

なし。

コマンド形式

```
{{dcsvstart -u ユーザサーバ名 [, ユーザサーバ名] ...}}
```

機能

同一マシンで一つの OpenTP1 システムとしてシステム起動時に一緒に開始するユーザサーバ群の構成を定義します。

説明

set 形式

なし。

コマンド形式

次ページ以降に記述しています。

名称

ユーザサーバの起動

形式

```
{{dcsvstart -u ユーザサーバ名 [, ユーザサーバ名] ...}}
```

機能

このノードで使用する**ユーザサーバ**を起動します。

ユーザサーバの開始が OpenTP1 システムの開始と一緒に行われます。また、ユーザサーバの起動順序を制御できます。OpenTP1 の開始モードが正常開始の場合は、ユーザサービス構成定義の指定に従い、OpenTP1 開始時にユーザサーバを起動します。再開時は、前回のオンラインの終了状態を引き継いで開始するため、ユーザサービス構成定義の指定は無視します。

ユーザサービス定義でサービスグループの常駐・非常駐の指定に関係なく、ユーザサーバを起動しなければこのサービスグループをコールできません。

なお、運用コマンドで起動するユーザサーバは指定しません。

オプション

●-u ユーザサーバ名 [, ユーザサーバ名] ...

～ 〈1～8 文字の識別子〉

このノードに起動するユーザサーバの名称を指定します。ユーザサービス定義のファイル名がユーザサーバ名になります。

一つの dcsvstart 定義コマンドで指定できるユーザサーバ名の最大数は 50 です。

指定したユーザサーバ群は並列に起動され、すべて起動されてから次のコマンドを実行します。指定例を次に示します。

サーバ A とサーバ B を並列に起動する場合

```
dcsvstart -u サーバA, サーバB
```

サーバ A の起動後に、サーバ B を起動する場合

```
dcsvstart -u サーバA
```

```
dcsvstart -u サーバB
```

注

サーバ内で次の関数を発行したタイミングで起動完了とします。

- dc_adm_complete 関数
- dc_mcf_mainloop 関数
- dc_rpc_mainloop 関数

形式

set 形式

```
[set rpc_trace=Y|N]
[set rpc_trace_name="RPCトレースを取得するファイル名"]
[set rpc_trace_size=RPCトレースを取得するファイルのサイズ]
[set name_port=ネームサービスのポート番号]
[set system_id=OpenTP1識別子]
[set all_node="ノード名 [:ポート番号] [:high] "
              [, "ノード名 [:ポート番号] [:high] "...] ]
  set node_id=ノード識別子
[set rpc_retry=Y|N]
[set rpc_retry_count=RPC処理のリトライ回数最大値]
[set rpc_retry_interval=RPC処理のリトライ間隔]
[set multi_node_option=Y|N]
[set prc_port=マルチノード連携制御機能で使用するポート番号]
[set rpc_delay_statistics=Y|N]
[set my_host="ホスト名"|MYHOST]
[set jp1_use=Y|N]
[set rpc_message_level=RPCのトラブルシュートメッセージの表示レベル]
[set rpc_multi_tp1_in_same_host=Y|N]
[set max_socket_descriptors=ソケット用ファイル記述子の最大数]
[set rpc_datacomp=Y|N]
[set domain_masters_addr="ドメイン名=ホスト名1 [:ポート番号1]
                          [, ホスト名2 [:ポート番号2]
                          [, ホスト名3 [:ポート番号3] ] ] "
                          [, "ドメイン名=ホスト名1 [:ポート番号1]
                          [, ホスト名2 [:ポート番号2]
                          [, ホスト名3 [:ポート番号3] ] ] "...] ]
[set domain_masters_port=
      ドメイン代表スケジュールサービスのポート番号]
[set domain_use_dns=Y|N]
[set client_uid_check=Y|N]
[set rpc_port_base=最小ポート番号]
[set prf_trace=Y|N]
[set trn_prf_trace_level=トレースの取得レベル]
[set core_suppress_watch_time=実時間監視タイムアウトプロセスの
                              コアダンプ出力を抑止する時間間隔]
[set rpc_netmask=TCP/IPのネットワーク定義ファイルに指定した
                 サブネットマスク値]
[set ipc_socketctl_highwater=
      ソケットの一時クローズ処理開始数パーセンテージ
      [, ソケットの一時クローズ処理非対象数パーセンテージ] ]
[set ipc_socketctl_watchtime=ソケット再利用可能監視時間]
[set ipc_conn_interval=コネクション確立監視時間]
[set ipc_send_interval=データ送信監視間隔]
[set ipc_send_count=データ送信監視回数]
[set ipc_header_recv_time=通信制御データの受信監視時間]
[set name_notify=Y|N]
[set all_node_ex="ノード名 [:ポート番号] "
                 [, "ノード名 [:ポート番号] "...] ]
[set rpc_server_busy_count=KFCA00356-Wメッセージを出力する
                           バンドル回数]
```

```
[set rpc_send_retry_count=TCP/IP接続時の接続時にエラーが発生した
                        場合のリトライ回数]
[set rpc_send_retry_interval=TCP/IP接続時の接続時にエラーが
                        発生した場合のリトライ間隔]
[set thdlock_sleep_time=スレッド間で排他が競合した場合のスレッドの
                        待ち時間]
[set ipc_rcvbuf_size=TCP/IPの受信バッファサイズ]
[set ipc_sndbuf_size=TCP/IPの送信バッファサイズ]
[set ipc_listen_sockbufset=Y|N]
[set rpc_router_retry_count=ENOBUFFS, ENOMEM発生時のリトライ回数]
[set rpc_router_retry_interval=ENOBUFFS, ENOMEM発生時のリトライ間隔]
[set ipc_backlog_count=コネクション確立要求を格納するキューの長さ]
[set statistics=Y|N]
[set name_domain_file_use=Y|N]
[set all_node_extend_number=ドメイン変更時の最大ノード数]
[set all_node_ex_extend_number=ドメイン定義ファイルを使用したドメイン変更
                        時の最大ノード数]
[set prc_current_work_path=カレントワーキングディレクトリ作成用のディレク
                        トリパス]
[set rpc_max_message_size=RPC送受信電文の最大長]
[set uap_trace_file_put=Y|N]
[set dcstart_wakeup_retry_count=OpenTP1起動通知エラーリトライ回数]
[set dcstart_wakeup_retry_interval=OpenTP1起動通知エラーリトライ間隔]
[set nam_prf_trace_level=NAMイベントトレースの取得レベル]
[set fil_prf_trace_option=0|1]
[set fil_prf_trace_delay_time=FILイベントトレース取得条件となる
                        ファイルアクセス処理時間のしきい値]
[set jnl_prf_event_trace_level=JNL性能検証用トレースの取得レベル]
[set jnl_fileless_option=Y|N]
[set prc_coredump_filter=coreファイルに共有メモリを含めるかの判断値]
[set name_service_mode=manager|agent|normal]
[set name_manager_node=ノード名 [:ポート番号] ]
[set name_remove_down_node=Y|N]
[set name_node_add_policy=using_only|all]
[set watch_time=最大応答待ち時間]
```

コマンド形式

```
{{dcbindht  -h   ホスト名
             [-n   ネットワーク名 [, ネットワーク名] ...] }}
```

putenv 形式

```
[putenv LANG LANG設定値]
```

機能

OpenTP1 システム共通の実行環境を定義します。

説明

set 形式のオペランド

●rpc_trace=Y|N

～《N》

RPC トレースを取得するかどうかを指定します。

Y

RPC トレースを取得します。

N

RPC トレースを取得しません。

RPC トレースを取得した場合、処理速度が低下し、RPC がタイムアウトでエラーリターンすることがあります。その場合は、状況に応じて次の 1.または 2.の最大応答待ち時間（デフォルト値：180 秒）を十分な値まで増やして対処してください。

1. watch_time オペランド（システム共通定義、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で指定）に指定した最大応答待ち時間
2. クライアントから rap サーバに引き継いだ最大応答待ち時間

クライアントから rap サーバに最大応答待ち時間を引き継ぐかどうかは、クライアント環境定義の DCWATCHTIMINHERIT オペランド（TP1/Client/W, TP1/Client/P の場合）、または TP1/Client/J 環境定義の dcwatchtiminherit オペランド（TP1/Client/J の場合）で指定します。

DCWATCHTIMINHERIT オペランドまたは dcwatchtiminherit オペランドに Y が指定されている場合は、上記 2.の最大応答待ち時間を使用してください。

DCWATCHTIMINHERIT オペランドまたは dcwatchtiminherit オペランドに N が指定されているか、指定が省略されている場合は、上記 1.の最大応答待ち時間を使用してください。

リモート API 機能を使用する場合、このオペランドに Y を指定しても、rap クライアントの RPC トレースは取得されません。

このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです（1.> 2.> 3.）。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●rpc_trace_name="RPC トレースを取得するファイル名"

～〈パス名〉《\$DCDIR/spool/rpctr》

RPC トレースを取得するファイルのパス名を指定します。

パス名のうち、RPC トレースを取得するファイル名（デフォルトは rpctr）の最大長は、13 文字です。

パス名に環境変数を指定する場合、パス名の先頭に環境変数を指定してください（指定例：\$DCDIR/tmp/ファイル名）。

このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです（1.> 2.> 3.）。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●rpc_trace_size=RPC トレースを取得するファイルのサイズ

～〈符号なし整数〉((1024～2147483648))《4096》(単位：バイト)

RPC トレースを取得するファイルのサイズを指定します。

このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです（1.> 2.> 3.）。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

システム共通定義、ユーザサービスデフォルト定義、またはユーザサービス定義で、このオペランドに 4096 以下を指定していても、送受信された電文長がこの指定値を上回る場合、このオペランドで指定した値よりも大きなトレースファイルが作成されることがあります。

●name_port=ネームサービスのポート番号

～〈符号なし整数〉((5001～65535))《10000》

ネームサービスのアドレスとなるポート番号を指定します。

マルチ OpenTP1 システムを作成する場合は、OpenTP1 システムごとにネームサービスのポート番号を変えてください。

マルチ OpenTP1 システムではない場合でも、OpenTP1 システム以外のプログラムが特定のポート番号を使用しているときは、その特定のポート番号と異なるポート番号をネームサービスに指定してください。

このオペランドで指定するポート番号は、ほかのプログラムと重複しないようにしてください。

また、OS には任意に割り当てる番号がありますが、この番号も使用しないでください。OS が任意に割り当てる番号は、OS の種別やバージョンによって異なります。詳細については、使用している OS のマニュアルを参照してください。

●system_id=OpenTP1 識別子

～ 〈1～2 文字の識別子〉

OpenTP1 識別子を指定します。

このオペランドの指定値を変更した場合、チェックポイントダンプファイルの初期化が必要です。

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

OpenTP1 システムのすべてのノード名、およびそのネームサーバがウェルノウンポート番号として使用するポート番号を指定します。この指定を省略した場合は、自ノードだけで構成する OpenTP1 システムが仮定されます。

サービス情報優先度指定機能を使用する場合、サービス情報を優先して選択するノード（優先選択ノード）の指定の後ろに、サービス情報優先度指定キーワードである":high"を指定します。ポート番号を省略する場合は、ノード名の後ろに指定してください。このキーワードは必ず指定値の最後に指定してください。キーワードの区切り文字':'を省略した場合や、間違った位置にキーワードを指定した場合は、定義不正となります。定義不正となった場合、KFCA00606-E（ホスト未定義）または KFCA00607-E（ポート番号不正）のメッセージを出力し、OpenTP1 の起動は失敗します。

システムをマルチ OpenTP1 構成にしていない場合は、このオペランドのノード名に、自ホストを示すホスト名（または IP アドレス）や、ループバックアドレス（127 で始まる IP アドレス）を指定しないでください。指定した場合、トランザクションの性能に悪影響を与えたり、OpenTP1 がシステムダウンしたりするおそれがあります。

システムをマルチ OpenTP1 構成にしている場合は、このオペランドのノード名に、自ホストを示すホスト名（または IP アドレス）を指定できます。ただし、ポート番号は重複しないようにして指定してください。なお、ループバックアドレスなどの IP アドレスが 127 で始まるホスト名（例：127.0.0.1）は指定しないでください。指定した場合、グローバル検索機能などに影響を与えます。

name_notify オペランドに Y を指定して起動通知機能を使用する場合は、このオペランドに指定したノードで起動する OpenTP1 へ起動通知を行います。起動通知機能を使用すると、コネクション確立時期を管理でき、確実に送信できるようになります。

また、グローバルアーカイブジャーナルサービス機能を使用する場合は、このオペランドを次のように指定してください。

- 被アーカイブジャーナルノードには、アーカイブジャーナルノードのノード名を含むように指定する
- アーカイブジャーナルノードには、すべての被アーカイブジャーナルノードのノード名を含むように指定する

ノード名 ～ 〈1～255 文字の識別子〉

OpenTP1 システムのすべてのノード名を指定します。識別子に使用できる文字は英数字、ピリオド、およびハイフンです。ノード名とは、/etc/hosts など定まるホスト名のことで、ノード名に 256 文字以上指定した場合、指定した文字列の先頭から 255 文字までをノード名として扱います。

ポート番号 ～ 〈符号なし整数〉 ((5001～65535)) 《10000》

ネームサーバがウェルノウンポート番号として使用するポート番号を指定します。

ポート番号を省略した場合は、name_port オペランドで指定したネームサービスのポート番号が仮定されます。

サービス情報優先度指定キーワード ～:high

サービス情報優先度指定機能を使用する場合に、サービス情報を優先して選択するノード（優先選択ノード）に指定します。サービス情報優先度指定機能については、マニュアル「OpenTP1 解説」を参照してください。

ノード自動追加機能とサービス情報優先度指定機能を併用する場合、システム共通定義の name_service_mode オペランドに manager または agent を指定し、このオペランドに属性情報（優先度オプション）を指定します。

このオペランドに指定した属性情報は、ノードリストに登録されているノードに付与されます。

なお、このオペランドに指定したノードが、マネージャノードから受け取ったノードリストに登録されていない場合、指定したノードは RPC の実行範囲にはなりません。

指定できるノード数は、最大 512 です。

注意事項

- このオペランドに指定したノードが all_node_ex オペランドにも指定されていたり、このオペランド内に重複して指定されていたりする場合、OpenTP1 が最初に解析したオペランドに対して KFCA00604-W メッセージが出力されます。この場合、最初に解析したノードが有効となります。解析の優先順位は次のとおりです（優先順位が高い＞優先順位が低い）。
 - all_node オペランド、all_node_ex オペランド両方に同一ノードが指定されている場合
all_node オペランド＞all_node_ex オペランド
 - all_node (all_node_ex) オペランドに同一ノードが複数指定されている場合
先行指定ノード＞後続指定ノード
- 通信できない IP アドレス（以降、無効 IP アドレスと記載します）や無効 IP アドレスに変換されるホスト名をこのオペランドに指定し、かつ name_notify オペランドに Y を指定した場合、OpenTP1 起動時にこのオペランドに指定したノードに対して起動通知メッセージを送信します。無効 IP アドレスに対して起動通知メッセージを送信すると、処理が完了するまでに、1 ノードごとにコネクション確立監視時間 (ipc_conn_interval オペランドの指定値×2) が必要です。このため、OpenTP1 の起動処理に時間が掛かることがあります。このオペランドに無効 IP アドレスを指定する場合は、OpenTP1 の起動処理時間が増加することを考慮した上で運用してください。
なお、この現象は、次の方法で回避できます。

- ・ name_notify オペランドに N を指定し、自ノードとこのオペランドに指定したノードで、互いにノード監視機能を使用する。
 - ・ このオペランドから無効 IP アドレスの指定を削除する。
- ノード監視機能の使用方法および注意事項については、マニュアル「OpenTP1 解説」の、OpenTP1 のノード管理のノード監視機能の説明を参照してください。

●node_id=ノード識別子

～ 〈4 文字の識別子〉

自ノードの識別子を指定します。

OpenTP1 システム内で一意となる識別子を指定してください。

このオペランドの指定値を変更した場合、ジャーナルファイルの初期化が必要です。

●rpc_retry=Y|N

～ 〈N〉

OpenTP1 の RPC では、サービス要求時に相手サーバが起動していない場合があるため（系切り替え中、相手サーバ起動中など）、サービス要求がエラーになるおそれがあります。このとき、RPC の処理をリトライするかどうかを指定します。

Y

RPC の処理をリトライします。

N

RPC の処理をリトライしません。

RPC の処理をリトライする種類を次に示します。

・ ユーザサーバでの要求先検索のリトライ

ネームサービスを使用する RPC は、ネームサービスから要求先の情報を取得します。要求先サーバが未起動状態で、ネームサービスから要求先の情報が取得できない場合は、rpc_retry_interval オペランドに指定された間隔で、rpc_retry_count オペランドに指定された回数分、要求先検索をリトライします。

・ ユーザサーバでの要求送信のリトライ

RPC は要求先を決定したあと、相手サーバへサービス要求を送信します。次の場合、rpc_retry_interval オペランドに指定された間隔を待たないで、rpc_retry_count オペランドに指定された回数分、要求送信をリトライします。

- ・ 相手サーバのサービス開始の関数（dc_rpc_mainloop または CBLDCRSV('MAINLOOP')) が完了していないため、サービス提供できないなど、サービス要求の受け付けができない場合
- ・ サービス要求送信直前に相手サーバがダウンしたなど、通信障害が発生した場合

なお、サービス要求の受け付けができない、通信障害の発生などの問題がすぐに取り除かれないと、`rpc_retry_count` オペランドに指定された回数分リトライしても、サービス要求がエラーとなるおそれがあります。

- **システムサーバでの要求送信のリトライ**

OpenTP1 のシステムサーバは、システムサーバ間で通信を行います。通信障害が発生した場合は、`rpc_retry_interval` オペランドに指定された間隔で、`rpc_retry_count` オペランドに指定された回数分、要求送信をリトライします。

RPC の処理をリトライする種類と、`rpc_retry_count` オペランド、および `rpc_retry_interval` オペランドの指定の関係を次に示します。

リトライの種類	<code>rpc_retry_count</code> オペランドの指定	<code>rpc_retry_interval</code> オペランドの指定
ユーザサーバでの要求先検索のリトライ	○	○
ユーザサーバでの要求送信のリトライ	○	—
システムサーバでの要求送信のリトライ	○	○

(凡例)

- ：オペランドの指定は有効となります。
- ：オペランドの指定は無効となります。

- **`rpc_retry_count`=RPC 処理のリトライ回数最大値**

～ 〈符号なし整数〉 ((1～3600)) 《18》

要求先検索、および要求送信のリトライを行う場合のリトライ回数の最大値を指定します。

- ・ 要求先検索のリトライ：ネームサービスを使用する RPC で、ネームサービスから要求先の情報が取得できない場合
- ・ 要求送信のリトライ：サービス要求送信時に、サービス要求が受け付けられない場合、または通信障害が発生した場合

この指定は、`rpc_retry` オペランドに Y を指定した場合に有効です。

- **`rpc_retry_interval`=RPC 処理のリトライ間隔**

～ 〈符号なし整数〉 ((1～3600)) 《10》 (単位：秒)

要求先検索、および要求送信のリトライを行う場合のリトライ間隔を指定します。

この指定は、`rpc_retry` オペランドに Y を指定した場合に有効です。

- **`multi_node_option`=Y|N**

～ 《N》

該当する OpenTP1 ノードでマルチノード機能を使用するかどうかを指定します。

Y

マルチノード機能を使用します。

N

マルチノード機能を使用しません。

TP1/Multi が組み込まれていないのに Y を指定した場合は、警告メッセージを出力し、OpenTP1 を停止します。

●**prc_port=マルチノード連携制御機能で使用するポート番号**

～ 〈符号なし整数〉 ((5001～49999))

マルチノード連携制御機能で使用するポート番号を指定します。

multi_node_option オペランドに N を指定した場合は、このオペランドを指定する必要はありません。

マルチ OpenTP1 システムを作成する場合は、OpenTP1 システムごとにポート番号を変えてください。

マルチ OpenTP1 システムではない場合でも、OpenTP1 システム以外のプログラムが特定のポート番号を使用しているときは、その特定のポート番号と異なるポート番号を指定してください。

name_port オペランドで指定するネームサービスのポート番号と異なるポート番号を指定してください。

また、マルチノード物理定義で指定するポート番号には、ここで指定した値を指定してください。

このオペランドの指定値を変更した場合は、OpenTP1 を再度セットアップしてください。

このオペランドで指定するポート番号は、ほかのプログラムと重複しないようにしてください。

また、OS には任意に割り当てる番号がありますが、この番号も使用しないでください。OS が任意に割り当てる番号は、OS の種別やバージョンによって異なります。詳細については、使用している OS のマニュアルを参照してください。

●**rpc_delay_statistics=Y|N**

～ 〈N〉

通信遅延時間統計情報を取得するかどうかを指定します。

通信遅延時間とは、dc_rpc_call 関数のレスポンスタイムのうち、実際の通信（要求送信および応答送信）に掛かった時間のことです。

通信遅延時間統計情報は同期応答型 RPC の場合だけ取得します。

Y

通信遅延時間統計情報を取得します。

N

通信遅延時間統計情報を取得しません。

●my_host="ホスト名"|MYHOST

～ 〈1～255 文字の識別子〉

OpenTP1 が使用するネットワークアダプタをホスト名として指定します。識別子に使用できる文字は英数字、ピリオド、およびハイフンです（ただし、IP アドレス形式は除く）。ホスト名は、/etc/hosts ファイルまたは DNS など、IP アドレスとのマッピングができなければなりません。なお、ホスト名に localhost または IP アドレスが 127 で始まるホスト（例：127.0.0.1）は指定しないでください。

ホスト名は、1 個だけ指定できます。

このオペランドを省略、または MYHOST を指定した場合は、hostname コマンドが返す名称が指定されたと解釈されます。

また、hostname コマンドが返す名称が IP アドレスとマッピングできる環境設定（/etc/hosts、DNS など）が必要です。

64 文字以上のホスト名を指定した場合、TP1/Client/W または TP1/Client/P のブロードキャストによるユーザ認証（dc_clt_cltin_s 関数、dc_clt_cltin 関数）では、引数 set_host にホスト名が設定されません。

OpenTP1 は、次の判断をするためにこのオペランドを使用します。

- OpenTP1 が RPC を行う際、通信相手が同一マシン上に存在するのか、または別マシン上に存在するのかを判断するため。
- 複数の IP アドレスを持つマシンの場合、使用するホスト名を my_host と dcbindht に指定することで自 OpenTP1 が使用する IP アドレスを判断し、通信相手に自 OpenTP1 の IP アドレスを通知するため。

●jp1_use=Y|N

～ 〈N〉

JP1 イベントサービス機能にイベント登録をするかどうかを指定します。

Y

イベント登録をします。

N

イベント登録をしません。

JP1 イベントサービス機能へ登録するイベントについては、マニュアル「OpenTP1 運用と操作」を参照してください。

●rpc_message_level=RPC のトラブルシュートメッセージの表示レベル

～ 〈符号なし整数〉 ((0～2)) 〈2〉

RPC トラブルシューティングメッセージの表示レベルを指定します。

0

トラブルシューティングメッセージを表示しません。

1

表示レベル 1 以下のトラブルシューティングメッセージを表示します。

2

表示レベル 2 以下のトラブルシューティングメッセージを表示します。

RPC トラブルシューティングメッセージ (KFCA00325-E~KFCA00332-W) と表示レベルの関係を、次に示します。

メッセージ	表示レベル
KFCA00325-E	0
KFCA00326-W	2
KFCA00327-W	1
KFCA00328-W	2
KFCA00329-E	0
KFCA00330-E	0
KFCA00331-W	—
KFCA00332-W	1

(凡例)

—：表示されません。

RPC トラブルシューティングメッセージは、標準エラー出力に出力されます。

●rpc_multi_tp1_in_same_host=Y|N

～《N》

同一ホスト内に複数の OpenTP1 を稼働させ、これらを同一グローバルドメイン※として運用するかどうかを指定します。

注※

ここでのグローバルドメインとは、次のノード名の集合を指します。

システム共通定義の name_domain_file_use オペランドに N を指定している場合

システム共通定義の all_node オペランド、および all_node_ex オペランドで指定したノード名の集合です。

システム共通定義の name_domain_file_use オペランドに Y を指定している場合

ドメイン定義ファイルに指定したノード名の集合です。なお、ドメイン定義ファイルは次の場所に格納します。

- all_node のドメイン定義ファイル
\$DCCONFPATH/dcnamnd ディレクトリ下
- all_node_ex のドメイン定義ファイル
\$DCCONFPATH/dcnamndex ディレクトリ下

Y

同一ホスト内の複数の OpenTP1 を、同一グローバルドメインとして運用します。

N

同一ホスト内の複数の OpenTP1 を、同一グローバルドメインとして運用しません。

このオペランドの指定は、OpenTP1 を停止してから変更してください。変更後は、dcsetup コマンドを入力してから OpenTP1 を起動してください。

●max_socket_descriptors=ソケット用ファイル記述子の最大数

～〈符号なし整数〉((32～2032))《64》

OpenTP1 制御下のプロセス※で、ソケット用に使用するファイル記述子の最大値を指定します。

OpenTP1 制御下のプロセス※では、システムサービスやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数によって、ソケット用のファイル記述子の最大数を変更する必要があります。

注※

MCF サービス（MCF マネジャサービス、MCF 通信サービス、アプリケーション起動サービス、マッピングサービス、およびオンラインコマンドサービス）は対象外です。MCF サービスの場合は、「[システムサービス共通情報定義](#)」を参照してください。

ソケット用ファイル記述子の最大数の計算式を、次に示します。

$\uparrow (\text{自ノード内 UAP プロセス数}^{\ast 1} + \text{システムサービスプロセス数}^{\ast 2}) / 0.8 \uparrow$

$\uparrow \uparrow$: 小数点以下を切り上げます。

注※1

自ノード内 UAP プロセス数は、次に示す値の合計です。

- 自 OpenTP1 内の UAP プロセス数
- CUP から同時に起動するトランザクション数（クライアントサービス定義の parallel_count オペランドの指定値）

注※2

システムサービスプロセス数とは、自 OpenTP1 内のシステムサービスプロセスの数です。自 OpenTP1 内のシステムサービスプロセスは、rpcstat コマンドで表示されるサーバ名をカウントすることで求められます。rpcstat コマンドで表示されるサーバ名のうち、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造に記載されているシステムサービスプロセスをカウントしてください。

このオペランドの指定値が小さいと、OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため、プロセスが KFC A00307-E メッセージを出力して異常終了します。

このオペランドは、ユーザサービス定義とユーザサービスデフォルト定義の中でも指定できます。

指定値の優先順位は次のとおりです。

システムサーバの場合 (1.> 2.)

1. 各システムサービス定義
2. システム共通定義

ユーザサーバの場合 (1.> 2.> 3.)

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

max_socket_descriptors オペランドの算出式はシステムサーバごとに異なります。そのため、システムサーバごとの max_socket_descriptors オペランドの値は、各システムサービス定義の計算式で算出した値を設定してください。

●rpc_datacomp=Y|N

～《N》

OpenTP1 システムで、クライアント側からのサービス要求時、およびサーバ側からのサービス要求応答時に、**ユーザデータを圧縮して送信するかどうか**を指定します。

このオペランドの指定は、dc_rpc_call 関数でサービスを要求するクライアント側で有効になります。したがって、クライアント側のノードで rpc_datacomp=Y を指定していれば、サービスを受け付けるサーバのあるノードで rpc_datacomp=Y が指定されていなくても、サービス要求メッセージおよび応答メッセージのユーザデータを圧縮して送信します。ただし、サーバ側のシステムがユーザデータの圧縮機能をサポートしている場合に限ります。逆に、クライアント側で rpc_datacomp=Y を指定していなければ、サービスを受け付けるサーバのあるノードで rpc_datacomp=Y が指定されていても、サービス要求メッセージおよび応答メッセージのユーザデータを圧縮しません。

また、クライアント側のノードで rpc_datacomp=Y を指定しても、ユーザデータの圧縮効果がない場合、ユーザデータを圧縮しないでサービス要求を送信します。しかし、それに対する応答メッセージに圧縮効果がある場合は、応答のユーザデータは圧縮して返送します。

なお、rpc_datacomp=Y を指定した場合でも、サービス要求を受信する SPP のメッセージ格納バッファプール長 (message_store_buflen オペランド) は、圧縮前のユーザデータ長でサイズを計算してください。

Y

サービス要求時とサービス要求応答時にユーザデータを圧縮して送信します。

N

サービス要求時とサービス要求応答時にユーザデータを圧縮しないで送信します。

●domain_masters_addr="ドメイン名=ホスト名 1 [: ポート番号 1] [, ホスト名 2 [: ポート番号 2] [, ホスト名 3 [: ポート番号 3]]) " [, "ドメイン名=ホスト名 1 [: ポート番号 1] [, ホスト名 2 [: ポート番号 2] [, ホスト名 3 [: ポート番号 3]]) "...]

ドメイン指定の RPC を発行した際の、通信先のドメイン名とドメイン代表スケジュールサービスのホスト名 (およびポート番号) を指定します。この定義を指定すると、ドメイン指定の RPC を発行した際にドメインネームサービスを検索しないで通信ができるため、RPC の応答時間が短縮できます。

ポート番号の指定を省略した場合は、domain_masters_port オペランドの値を仮定します。

ドメイン名 ~ 〈1 文字以上の識別子〉

通信先のドメイン名を指定します。

ホスト名 1, ホスト名 2, ホスト名 3 ~ 〈1~255 文字の識別子〉

スケジュールサービスのホスト名を指定します。

ホスト名 1 には第 1 優先のホスト名を、ホスト名 2 以降には第 2 優先のホスト名を指定します。

ホスト名に 256 文字以上指定した場合、指定した文字列の先頭から 255 文字までをホスト名として扱います。

ポート番号 1, ポート番号 2, ポート番号 3 ~ 〈符号なし整数〉 ((5001~65535))

スケジュールサービスのポート番号を指定します。

ポート番号 1 には第 1 優先のポート番号を、ポート番号 2 以降には第 2 優先のポート番号を指定します。

●domain_masters_port=ドメイン代表スケジュールサービスのポート番号

~ 〈符号なし整数〉 ((5001~65535))

ドメイン指定の RPC を発行した際に、ドメインネームサービスを検索しないで通信するために必要な、通信先のドメイン代表スケジュールサービスのポート番号を指定します。

●domain_use_dns=Y|N

~ 〈Y〉

ドメイン指定の RPC を発行した際、domain_masters_addr オペランドに指定していないドメインへの検索をする場合に、ドメインネームサービスに問い合わせるかどうかを指定します。

Y

ドメインネームサービスに問い合わせます。

N

ドメインネームサービスに問い合わせません。

●client_uid_check=Y|N

～《HP-UX または Windows の場合：Y AIX, Linux または Solaris の場合：N》

TP1/Client/W または TP1/Client/P に対するユーザ認証機能を使用するかどうかを指定します。

Y

TP1/Client/W または TP1/Client/P に対するユーザ認証機能を使用します。なお、適用 OS が HP-UX で「高信頼性システム」、または「HP-UX Shadow Passwords」を適用する場合は、S を指定してください。適用 OS が HP-UX 以外の場合に S を指定すると定義不正になります。

N

TP1/Client/W または TP1/Client/P に対するユーザ認証機能を使用しません。

N を指定した場合、TP1/Client/W または TP1/Client/P からの認証要求は、どのようなログイン名を指定しても認証されます。このオペランドに Y (HP-UX の場合は S) を指定した場合に、適用 OS が AIX, Linux, Solaris, または HP-UX のときは、namd プロセスのユーザ ID が root になります。

●rpc_port_base=最小ポート番号

～〈符号なし整数〉((5001～65535))

OpenTP1 制御下のウェルノウンポート化されていないプロセスが利用する受信用ポート番号を、特定の範囲内の値に限定したい場合に、その最小値を指定します。

使用するポート番号の数は、プロセスサービス定義の prc_process_count オペランドで指定した値（最大同時起動サーバプロセス数）に 128 を加えた値になります。したがってこのオペランドを指定すると、受信用のポート番号は、指定された範囲内の値になります。このオペランドは、OpenTP1 のシステムサーバ、SUP、SPP、および MHP で有効です。

このオペランドは、プロセスサービス定義の prc_process_count オペランドの指定値に 128 を加えた値と合わせて、65535 を超えないように指定してください。65535 を超えた場合は、このオペランドの指定値から 65535 までの範囲のポート番号と、OS が任意に割り当てるポート番号が使用されます。

次に示す場合、このオペランドの指定は無視され、OpenTP1 制御下のウェルノウンポート化されていないプロセスのポート番号は、OS が任意に割り当てた番号になります。

- このオペランドを省略した場合
- プロセス起動時に、最小ポート番号から、最小ポート番号と最大同時起動サーバプロセス数に 128 を加えた値の合計値までの範囲から任意に選択したポート番号が使用中の場合

最小ポート番号から、最小ポート番号と最大同時起動サーバプロセス数に 128 を加えた値の合計値まで、またはこのオペランドの省略によって OS が任意に割り当てるポート番号の範囲に、OpenTP1 のウェルノウンポート、および他プログラムのウェルノウンポートを定義しないでください。

●prf_trace=Y|N

～《Y》

性能検証用トレース情報を取得するかどうかを指定します。

Y

性能検証用トレース情報を取得します。

N

性能検証用トレース情報を取得しません。

●trn_prf_trace_level=トレースの取得レベル

～((00000001～00000003))《00000003》

性能検証用トレース情報の取得で、トランザクション関係のトレースの取得レベルを指定します。トランザクション関係のトレースのイベント ID は 0x4000 から 0x4150 です。イベント ID の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

00000001

性能検証用トレース情報（イベント ID：0x4000～0x4017）を取得します。

00000002

トランザクションの開始時および終了時のイベントの性能検証用トレース情報（イベント ID：0x4100, 0x4150）を取得します。

00000003

00000001 および 00000002 両方の性能検証用トレース情報を取得します。

●core_suppress_watch_time=実時間監視タイムアウトプロセスのコアダンプ出力を抑止する時間間隔

～〈符号なし整数〉((0～3600))《0》(単位：秒)

実時間監視でタイムアウトを検知して強制停止させるプロセスについて、コアダンプの出力を抑止する時間間隔を秒単位で指定します。

以前に実時間監視タイムアウトを検知しコアダンプを出力してプロセスを停止させてから、このオペランドで指定した時間を経過するまでは、実時間監視タイムアウトでプロセスを強制停止させる際にコアダンプを出力しません。

このオペランドに 0 を指定した場合は、コアダンプ出力を抑止しないで、実時間監視タイムアウトで強制停止させるプロセスは、すべてコアダンプを出力します。

●rpc_netmask=TCP/IP のネットワーク定義ファイルに指定したサブネットマスク値

OpenTP1 が稼働する環境のネットワークでサブネットを使用している場合、TCP/IP のネットワーク定義ファイルに指定したサブネットマスク値を、「.」記法の文字列のインターネットアドレスで指定します

(例:「rpc_netmask=255.255.255.0」)。なお、インターネットアドレスの「.」と数字の間にスペースを挿入すると、スペース以降の値は解釈されないので注意してください。

TCP/IP のネットワーク定義ファイルにサブネットマスクを指定していない場合は、このオペランドを指定する必要はありません。

このオペランドは、dcbindht 定義コマンドを指定している場合に有効です。サブネットを使用したネットワーク上で稼働させ、かつ dcbindht 定義コマンドを指定している場合は、必ず指定しなければなりません。サブネットを使用したネットワーク上で稼働させている場合に、このオペランドを指定しなかったときは、dcbindht 定義コマンドに指定したネットワーク名と OpenTP1 の認識するネットワークアドレスが一致しないため、dcbindht 定義コマンドの指定値が有効になりません。

dcbindht 定義コマンドを使用したシステムを、サブネットを使用したネットワーク環境で運用する場合、OpenTP1 のグローバルドメイン下のネットワークはすべて同一サブネットマスクを使用していることが前提となります。OpenTP1 のグローバルドメイン下で、異なるサブネットマスクを使用したネットワークが存在する場合、dcbindht 定義コマンドの機能が正常に動作しないことがあります。

●ipc_socketctl_highwater=ソケットの一時クローズ処理開始数パーセンテージ [ソケットの一時クローズ処理非対象数パーセンテージ]

～ 〈符号なし整数〉 ((0～100)) 《100, 0》

max_socket_descriptors オペランドの指定値に対して、ソケットの一時クローズ処理開始数パーセンテージを指定します。

OpenTP1 は、プロセス内のソケット用に使用しているファイル記述子の数が、次の値を超えた時点で、一時クローズ処理を開始します。

max_socket_descriptors オペランドの指定値
× (ソケットの一時クローズ開始数パーセンテージ / 100)

ソケットの一時クローズ処理開始数パーセンテージに 0 を指定した場合、コネクションを確立するたびに一時クローズ処理が実行されます。一時クローズ処理については、マニュアル「OpenTP1 解説」を参照してください。

また、一時クローズ処理の対象外とするコネクション数のパーセンテージも指定できます。一時クローズ処理の対象外とするコネクション数は、次の値です。

max_socket_descriptors オペランドの指定値
× (ソケットの一時クローズ非対象数パーセンテージ / 100)

ソケットの一時クローズ非対象数パーセンテージには、ソケットの一時クローズ処理開始数パーセンテージ以下の値を指定してください。ソケットの一時クローズ処理開始数パーセンテージより大きい値を指定した場合は、ソケットの一時クローズ処理開始数パーセンテージと同じ値が指定されたものとして動作します。

OpenTP1 は、プロセス内で確立したコネクションを、確立した順に管理しています。ソケットの一時クローズ処理非対象数パーセンテージを指定した場合、最も古く確立されたコネクションから順に、一時クローズ処理要求が送信されます。

max_socket_descriptors オペランドの指定値が大きくなり、かつソケットの一時クローズ処理開始数パーセンテージに小さい値が指定されていると、一時クローズ要求が多発し、性能に影響を与えたり、通信障害になったりすることがあります。

このオペランドは、rap リスナーサービス定義、ユーザサービスデフォルト定義、またはユーザサービス定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義、または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●ipc_sockctl_watchtime=ソケット再利用可能監視時間

～ 〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)

プロセス内のソケット用に使用しているファイル記述子の数が、max_socket_descriptors オペランドの指定値になった時点から、一時クローズ処理によってソケットが再利用できるようになるまでの監視時間 (秒) を指定します。

UAP が異常終了した場合に閉塞するかどうかは hold オペランド、term_watch_time オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

一時クローズ処理は、コネクションを確立したプロセス間の合意によってコネクションを切断するため、一時クローズ処理の要求を送信したプロセスはその応答を受けるまではコネクションを切断できません。応答を受信したあと、コネクションは切断され、ソケットが再利用できるようになります。

ipc_sockctl_watchtime オペランドの指定値を経過しても、どのプロセスからも一時クローズ処理要求に対する応答が返らない場合、プロセスは強制停止されます。ipc_sockctl_watchtime オペランドに 0 を指定した場合、無限に待ちます。

このオペランドは、rap リスナーサービス定義、ユーザサービスデフォルト定義、またはユーザサービス定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義、または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●ipc_conn_interval=コネクション確立監視時間

～〈符号なし整数〉((8～65535))《8》(単位：秒)

データ送信時のコネクション確立監視時間を秒単位で指定します。

ノンブロッキングモードで呼び出した connect() システムコールに対する応答を受信するまでの監視時間を指定します。

このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●ipc_send_interval=データ送信監視間隔

～〈符号なし整数〉((5～32767))《5》(単位：秒)

データ送信時のデータ送信監視間隔を指定します。

このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●ipc_send_count=データ送信監視回数

～〈符号なし整数〉((1～32767))《5》

データ送信時の送信完了監視回数を指定します。

OpenTP1 のデータ送信監視時間は、次に示す秒数で監視します。

ipc_send_interval オペランドの指定値 × ipc_send_count オペランドの指定値

このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義

3. システム共通定義

●ipc_header_rcv_time=通信制御データの受信監視時間

～ 〈符号なし整数〉 ((5～32767)) 《10》 (単位：秒)

OpenTP1 が、データの受信が始まったことを TCP/IP から通知されてから、通信制御データを受信するまでの監視時間を指定します。

このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●name_notify=Y|N

～ 《N》

自ノードの OpenTP1 起動時に、all_node および all_node_ex オペランドで指定したノードで起動する OpenTP1 に対し**起動通知を行う**かどうか、または自ノードの OpenTP1 が他ノードの OpenTP1 からの**起動通知を受け付ける**かどうかを指定します。

Y

起動通知機能を使用します。

N

起動通知機能を使用しません。

なお、このオペランドに指定した値に関係なく、namunavl コマンドは動作します。

ノード自動追加機能を使用する場合の、このオペランドの動作を次に示します。

オペランドに Y を指定した場合

- ・ ノードリストに登録されているノード、および all_node_ex オペランドで指定したノードの OpenTP1 に対して起動通知を行います。また、起動通知を受け付けます。
- ・ マネジャノードでは、起動通知の送信元ノードがノードリストに登録されていないときは、送信元ノードをノードリストに追加します。

オペランドに N を指定した場合

他ノードの OpenTP1 に対して起動通知をしません。また、他ノードの OpenTP1 からの起動通知を受け付けません。ただし、マネジャノードでは、起動通知の送信元ノードがノードリストに登録されていないときは、送信元ノードをノードリストに追加します。

なお、マネジャノードでは、新たに起動したエージェントノードからのノードリスト要求を起動通知と解釈します。

注

- この機能は、ノードごとにユニークな IP アドレスが割り振られていることを前提としているため、同一 IP アドレスで複数の OpenTP1 が起動する（1LAN ボードだけ使用時など）システムでは、起動通知機能を使用しないでください。
- 起動通知機能を使用する場合、自ノードの all_node オペランドまたは all_node_ex オペランドで指定した起動通知の通知先ノードでは、all_node オペランドまたは all_node_ex オペランドに、自ノードを指定してください。

●all_node_ex="ノード名 [:ポート番号] " ["ノード名 [:ポート番号] "...]

あて先指定の RPC によって、all_node オペランドに指定したノード名以外に通信する可能性のある OpenTP1 システムのノード名およびそのネームサーバがウェルノウンポート番号として使用するポート番号を指定します。自ノードの OpenTP1 起動時に、all_node_ex オペランドおよび all_node オペランドで指定したノードで起動する OpenTP1 へ起動通知を行います。起動通知機能を使用すると、コネクション確立時期を管理でき、確実に送信できるようになります。

ノード名 ～ 〈1～255 文字の識別子〉

OpenTP1 システムのすべてのノード名を指定します。識別子に使用できる文字は英数字、ピリオド、およびハイフンです。ノード名とは、/etc/hosts など で定まるホスト名のことです。ノード名に 256 文字以上指定した場合、指定した文字列の先頭から 255 文字までをノード名として扱います。なお、IP アドレスを指定する場合、ループバックアドレスなど IP アドレスが 127 で始まるホスト名（例：127.0.0.1）は指定しないでください。

ポート番号 ～ 〈符号なし整数〉 ((5001～65535)) 《10000》

ネームサーバがウェルノウンポート番号として使用するポート番号を指定します。

ポート番号を省略した場合は、name_port オペランドで指定したネームサービスのポート番号が仮定されます。

注意事項

- このオペランドに指定したノードが all_node オペランドにも指定されていたり、このオペランド内に重複して指定されていたりする場合、OpenTP1 が最初に解析したオペランドに対して KFCA00604-W メッセージが出力されます。この場合、最初に解析したノードが有効となります。解析の優先順位は次のとおりです（優先順位が高い＞優先順位が低い）。
 - all_node オペランド、all_node_ex オペランド両方に同一ノードが指定されている場合
all_node オペランド＞all_node_ex オペランド
 - all_node (all_node_ex) オペランドに同一ノードが複数指定されている場合
先行指定ノード＞後続指定ノード
- ノード自動追加機能を使用する場合、次の点に注意してください。

システム共通定義の name_service_mode オペランドに manager または agent を指定したとき、このオペランドに指定したノードはノードリストに登録されません。ノードリストに登録するノードはこのオペランドに指定しないでください。

マネージャノードでは、このオペランドに指定したノードの情報をエージェントノードへ配布しません。ノードリストに登録するノードは、このオペランドに指定しないでください。

●rpc_server_busy_count=KFCA00356-W メッセージを出力するバンドル回数

～ 〈符号なし整数〉 ((0～32767)) 《20》

スケジューラサービスは、受け付けたサービス要求をスケジューラキューに登録します。スケジューラキュー登録処理は、スレッドを使用し、複数のサービス要求を並列にスケジューラキューに登録します。しかし、スケジューラキュー登録に使用するスレッドが不足したためスケジューラサービスがサービス要求を受け付けられない場合、KFCA00356-W メッセージを出力します。

このオペランドには、KFCA00356-W メッセージを出力するバンドル回数を指定します。KFCA00356-W メッセージを出力する契機は、初めてスケジューラサービスがサービス要求を受け付けられなかった場合、または、KFCA00356-W メッセージ出力後、スケジューラサービスがサービス要求を受け付けられなかった回数がこのオペランドの値以上になった場合に KFCA00356-W メッセージを出力します。

このオペランドに 0 を指定した場合は、スケジューラサービスがサービス要求を受け付けられなかった回数に関係なく、KFCA00356-W メッセージを出力しません。また、1 を指定した場合は、スケジューラサービスがサービス要求を受け付けられないたびに KFCA00356-W メッセージを出力します。

●rpc_send_retry_count=TCP/IP コネクションの接続時にエラーが発生した場合のリトライ回数

～ 〈符号なし整数〉 ((0～65535)) 《0》

サーバの応答送信時の TCP/IP コネクションの接続で、ECONNREFUSED (239), EHOSTUNREACH (242), ENETUNREACH (229) のどれかのエラーが発生した場合、このオペランドでリトライ回数を指定することによって、エラーを回避できることがあります。

0 を指定した場合は、TCP/IP コネクションの接続時に上記のエラーが発生しても接続をリトライしません。

このオペランドに 1～65535 を指定し、TCP/IP コネクションの接続時に上記のエラーが発生した場合、rpc_send_retry_interval オペランドに指定した時間を待ったあと、接続をリトライします。

このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●**rpc_send_retry_interval=TCP/IP コネクションの接続時にエラーが発生した場合のリトライ間隔**
～ 〈符号なし整数〉 ((0～300000)) 《0》 (単位：ミリ秒)

サーバの応答送信時の TCP/IP コネクションの接続で、ECONNREFUSED、EHOSTUNREACH、ENETUNREACH のどれかのエラーが発生した場合、このオペランドでリトライ間隔をミリ秒単位で指定することによって、エラーを回避できることがあります。

0 を指定した場合は、間隔を空けずに TCP/IP コネクションの接続をリトライします。1～19 は指定できません。1～19 を指定した場合は、20 を仮定します。

このオペランドは、rpc_send_retry_count オペランドに 1～65535 を指定した場合に有効になります。

このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義、各システムサービス定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.> 4.)。

- 1. ユーザサービス定義
- 2. ユーザサービスデフォルト定義
- 3. 各システムサービス定義
- 4. システム共通定義

●**thdlock_sleep_time=スレッド間で排他が競合した場合のスレッドの待ち時間**
～ 〈符号なし整数〉 ((1～32767)) 《15》 (単位：ミリ秒)

OpenTP1 制御下のプロセスが内部で用いる排他制御関数が、スレッド間で排他の競合が発生したため排他を獲得できなかった場合のスレッドの待ち時間を、ミリ秒単位で指定します。

このオペランドにデフォルト値 (15) より小さい値を指定すると、select システムコールの発行頻度が高くなり、CPU 利用率が増加します。

注

スレッドで排他の競合が発生した結果、排他を獲得できなかったスレッドが待つ場合、このオペランドの指定値に従い、OS の select システムコールを発行します。ただし、OS やマシン環境によって select システムコールの待ち時間精度は異なるため、実際の待ち時間がこのオペランドに指定された値より長くなるおそれがあります。

各 OS でのこのオペランドの指定値による待ち時間の測定結果の例を次の表に示します。

なお、マシン環境 (OS のバージョンやパッチ適用の有無など) によっては、待ち時間が次に示す表と異なる場合があります。select システムコールの待ち時間精度については、OS の仕様を確認してください。

OS 名	thdlock_sleep_time オペランドの指定値	待ち時間
AIX	指定値	指定値

OS 名	thdlock_sleep_time オペランドの指定値	待ち時間
Linux	指定値	指定値+約 1 ミリ秒
HP-UX	1～9	約 10 ミリ秒
	10～19	約 20 ミリ秒
Solaris	1～9	約 10 ミリ秒
	10～19	約 20 ミリ秒
Windows	1～15	約 16 ミリ秒
	16～31	約 31 ミリ秒

このオペランドは、各システムサービス定義およびシステムサービス共通情報定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.)。

1. 各システムサービス定義およびシステムサービス共通情報定義
2. システム共通定義

●ipc_listen_sockbufset=Y|N

～《Y》

OpenTP1 がプロセス間通信で使用する listen ソケットの生成時に、ipc_sendbuf_size オペランド、および ipc_recvbuf_size オペランドに指定した TCP/IP の送受信バッファサイズを listen ソケットに設定するかどうかを指定します。

Y

listen ソケットに、TCP/IP の送受信バッファサイズを設定します。

N

listen ソケットに、TCP/IP の送受信バッファサイズを設定しません。

ipc_sendbuf_size オペランド、および ipc_recvbuf_size オペランドに指定した TCP/IP の送受信バッファサイズは、OpenTP1 プロセスがコネクション確立要求を受信し、コネクションを確立したあとに設定します。コネクションを確立したあとに TCP/IP の送受信バッファサイズを変更するため、コネクションの接続元と接続先の間で TCP/IP のバッファサイズの値が異なり、通信の遅延が発生することがあります。

同一ノード内に TP1/Server Base と TP1/Client があり、TP1/Client から同一ノード内の TP1/Server Base に対して通信を行う場合は、このオペランドに Y を指定することをお勧めします。

このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義

3. システム共通定義

●`ipc_rcvbuf_size`=TCP/IP の受信バッファサイズ

～ 〈符号なし整数〉 ((0, 8192～1048576)) 《8192》 (単位: バイト)

コネクションごとに確保されるソケットの受信バッファのサイズを指定します。高速な通信媒体や MTU の大きな通信媒体を使用している場合、この値を大きくすることによって性能向上を図れます。

0 を指定した場合は、OS の受信バッファサイズを適用します。受信データサイズが 8192 バイトを超える場合、このオペランドに 0 を指定することで受信バッファのサイズが拡張され、性能向上を図れることがあります。

なお、0 を指定する場合、このノードと通信するすべてのノードで同じ値を指定してください。同じ値を指定しない場合、通信するノードとバッファサイズに差異が生じ、通信性能が劣化するおそれがあります。また、1～8191 は指定できません。1～8191 を指定した場合、0 を仮定します。

`ipc_rcvbuf_size` オペランドに指定した値よりも小さいデータをやり取りし合う通信処理の場合、`ipc_tcpnodelay` オペランドに Y を指定することを推奨します。

注

TCP は、受信したデータに対し、送達確認(ACK)パケットを返信します。

受信バッファのサイズに対し、受信したデータが小さいと、データを受信しても直ちに ACK を返信しません(遅延 ACK)。

このオペランドに大きな値を指定し、小さいデータをやり取りし合うような通信処理の場合、遅延 ACK の影響によって性能が悪くなるおそれがあります。遅延 ACK の詳細については、TCP/IP の文献を参照してください。

このオペランドの値は、OS で使用できる TCP/IP の受信バッファのサイズ以下の値を指定してください。指定する TCP/IP の受信バッファのサイズは、グローバルアーカイブジャーナル機能では有効になりません。グローバルアーカイブジャーナル機能の場合は、次のシステムサービス定義に TCP/IP の送受信バッファのサイズを指定してください。

- ジャーナルサービス定義の `jnl_arc_ipc_buff_size` オペランド
- グローバルアーカイブジャーナルサービス定義の `jnl_arc_ipc_buff_size` オペランド

このオペランドは、ユーザサービスデフォルト定義、ユーザサービス定義、各システムサービス定義でも指定できます。

指定値の優先順位は次のとおりです。

システムサーバの場合 (1.> 2.)

1. 各システムサービス定義
2. システム共通定義

ユーザサーバの場合 (1.> 2.> 3.)

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●ipc_sendbuf_size=TCP/IP の送信バッファサイズ

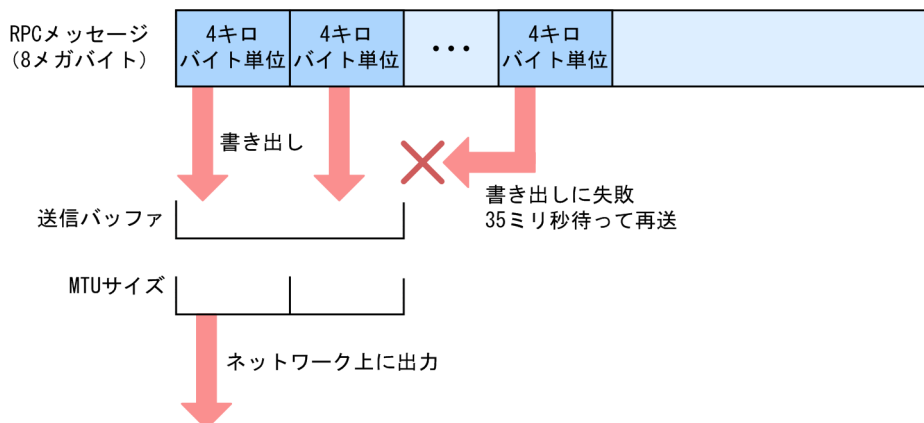
～ 〈符号なし整数〉 ((0, 8192～1048576)) 《8192》 (単位：バイト)

コネクションごとに確保されるソケットの送信バッファのサイズを指定します。高速な通信媒体や MTU の大きな通信媒体を使用している場合、この値を大きくすることによって性能向上を図れます。

0 を指定した場合は、OS の送信バッファサイズを適用します。送信データサイズが 8192 バイトを超える場合、このオペランドに 0 を指定することで送信バッファのサイズが拡張され、性能向上を図れることがあります。

RPC は、送信バッファにデータ送信する段階で 4 キロバイト単位で送信しており、ネットワークの状態によっては送信バッファ内のデータがネットワーク上に出力しきれないで、送信処理に失敗することがあります。例えば、rpc_max_message_size オペランドに 8 を指定して 8 メガバイトのメッセージを送信する場合、ネットワークの状態次第では 4 キロバイトの送信処理で 35 ミリ秒の待ちが無視できない回数発生し、通信性能に大きな影響を与えるおそれがあります。送信処理に失敗する場合の例を、次の図に示します。

図 3-1 送信バッファ内のデータがネットワーク上に出力しきれないため、送信処理に失敗する場合の例



このような場合は、このオペランドに 0 を指定し、送信バッファサイズを調整することを推奨します。

なお、0 を指定する場合、このノードと通信するすべてのノードで同じ値を指定してください。同じ値を指定しない場合、通信するノードとバッファサイズに差異が生じ、通信性能が劣化するおそれがあります。また、1～8191 は指定できません。1～8191 を指定した場合、0 を仮定します。

ipc_sendbuf_size オペランドに指定した値よりも小さいデータをやり取りし合う通信処理の場合、ipc_tcpnodelay オペランドに Y を指定することを推奨します。

注

このオペランドの値は、OS で使用できる TCP/IP の送信バッファのサイズ以下の値を指定してください。指定する TCP/IP の送信バッファのサイズは、グローバルアーカイブジャーナル機能では有効になりません。グローバルアーカイブジャーナル機能の場合は、次のシステムサービス定義に TCP/IP の送受信バッファのサイズを指定してください。

- ジャーナルサービス定義の jnl_arc_ipc_buff_size オペランド
- グローバルアーカイブジャーナルサービス定義の jnl_arc_ipc_buff_size オペランド

このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義、各システムサービス定義でも指定できます。

指定値の優先順位は次のとおりです。

システムサーバの場合 (1.> 2.)

1. 各システムサービス定義
2. システム共通定義

ユーザサーバの場合 (1.> 2.> 3.)

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●rpc_router_retry_count=ENOBUFS, ENOMEM 発生時のリトライ回数

～ 〈符号なし整数〉 ((0～65535)) 《30》

TP1/Server Base の内部で発行する accept システムコールで ENOBUFS が発生した場合、またはメッセージ受信処理で ENOMEM が発生した場合、メッセージ受信処理 (accept システムコールも含む) のリトライ処理回数を指定します。

●rpc_router_retry_interval=ENOBUFS, ENOMEM 発生時のリトライ間隔

～ 〈符号なし整数〉 ((0～3600000)) 《0》 (単位：ミリ秒)

TP1/Server Base の内部で発行する accept システムコールで ENOBUFS が発生した場合、またはメッセージ受信処理で ENOMEM が発生した場合、メッセージ受信処理 (accept システムコールも含む) のリトライ処理間隔を指定します。0 を指定した場合は、リトライ処理間隔を空けないで、メッセージ受信処理のリトライ処理を実行します。

なお、このオペランドに 1～9 を指定した場合は、10 を仮定します。

●ipc_backlog_count=コネクション確立要求を格納するキューの長さ

～ 〈符号なし整数〉 ((0～4096)) 《0》

コネクション確立要求を格納するキューの長さ (listen システムコールのバックログ数) を指定します。

0（デフォルト値）を指定した場合、listen システムコールのバックログ数に設定する実数値は OS によって異なります。デフォルト値を指定した場合の実数値については、「リリースノート」を参照してください。

実際のキューの長さは、指定した値より長くなる場合があります。

キューの長さの上限値、下限値は、各 OS によって異なります。OS によって、キューの長さの上限値、下限値が制限されている場合、設定した値が有効にならないことがあります。コネクション確立要求を格納するキューの詳細については、OS のマニュアル、または、TCP/IP の文献を参照してください。

このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです（1.> 2.> 3.）。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●statistics=Y|N

～《N》

共用メモリ上にシステム統計情報を取得するかどうかを指定します。

Y

共用メモリ上にシステム統計情報を取得します。

N

共用メモリ上にシステム統計情報を取得しません。

取得されるシステム統計情報を見るためには、dcstats コマンドでジャーナルファイルに出力するか、dcreport コマンドで標準出力に出力してください。

取得されるシステム統計情報は、dcstats コマンドの-s オプション指定時に取得されるシステム統計情報と同じです。

取得されるシステム統計情報は、OpenTP1 開始時からの累計なので、オーバフローすることがあります。したがって、長時間運用する場合は、dcstats コマンドでジャーナルファイルに出力するか、dcreport コマンドの-r オプションで定期的に値をリセットしてください。

●name_domain_file_use=Y|N

～《N》

OpenTP1 起動時および再開始（リラン）時に有効にするドメイン構成を指定します。

namchgl コマンドを使用してドメイン構成を変更する場合の定義ファイルの指定方法および注意事項については、マニュアル「OpenTP1 運用と操作」を参照してください。

Y

ドメイン定義ファイルに指定したドメイン構成を有効にします。

N

システム共通定義に指定したドメイン構成を有効にします。

ドメイン定義ファイルには、all_node を指定するファイルと、all_node_ex を指定するファイル、さらに優先選択ノードを指定する定義ファイルがあります。優先選択ノードの定義ファイルの指定方法は、all_node または all_node_ex を指定するファイルと同じです。ファイル名称は任意の名称です。このオペランドを有効にするには、それぞれのファイルを次の場所に格納してください。

all_node を指定するファイル

\$DCCONFPATH/dcnamnd ディレクトリ下

all_node_ex を指定するファイル

\$DCCONFPATH/dcnamndex ディレクトリ下

優先選択ノードを指定する定義ファイル

\$DCCONFPATH/dcnampr ディレクトリ下

これらのファイルのノード名やポート番号に不正があった場合は、KFCA00656-E（ノード名不正）、KFCA00657-E（ポート番号不正）または KFCA00666-E（ホスト未定義）のメッセージを出力し、OpenTP1 の起動に失敗します。

また、優先選択ノードの定義ファイルに指定したノードが all_node のドメイン定義ファイルに指定されていなかった場合は、KFCA00603-W（all_node に未定義）のメッセージを出力し、優先選択ノードの指定を無効として OpenTP1 を起動します。なお、同一ファイル内のほかの行で正しく定義されているノードについては、優先選択ノードの指定は有効となります。

それぞれのディレクトリ下に、複数のドメイン定義ファイルを格納した場合、同じディレクトリ下に格納されているすべてのドメイン定義ファイルを合わせて一つのドメイン（OpenTP1 システム）を構成します。次の場合、自ノードだけのドメインとして OpenTP1 システムを構成します。

- ・ドメイン定義ファイルが正しい場所に存在しない場合
- ・ドメイン定義ファイルを格納するディレクトリが存在しない場合
- ・ドメイン定義ファイルにノードが指定されていない場合

OpenTP1 の起動中にこのオペランドを変更した場合の動作は保証できません。

Y を指定した場合、namndchg コマンドがエラーリターンします。N を指定した場合、namchgfl コマンドがエラーリターンします。

name_domain_file_use オペランドの指定による機能差について次の表に示します。

OpenTP1 の状態	name_domain_file_use オペランドの指定値	
	Y	N
正常開始	システム共通定義の all_node オペランド、および all_node_ex オペランドを無視して、ドメイン定義ファイルの指定で OpenTP1 システムを構成する。	システム共通定義の all_node オペランド、および all_node_ex オペランドの指定で OpenTP1 システムを構成する。
再開始 (リラン)	再開始が発生した時点の、all_node オペランド、および all_node_ex オペランドを無視して、ドメイン定義ファイルの指定で OpenTP1 システムを構成する。	再開始時点の、all_node オペランド、および all_node_ex オペランドの指定で OpenTP1 システムを構成する。
稼働中	namchgfl コマンドでドメイン構成を変更できる。コマンド実行時点のドメイン定義ファイルの指定でドメイン構成を変更する。	namchgfl コマンドでドメイン構成を変更できない。OpenTP1 起動時点の name_domain_file_use オペランドの指定 (N) が環境不正となり、コマンドがエラーリターンする。
	namndchg コマンドでドメイン構成を変更できない。OpenTP1 起動時点の name_domain_file_use オペランドの指定 (Y) が環境不正となり、コマンドがエラーリターンする。	namndchg コマンドでドメイン構成を変更できる。コマンド実行時点のシステム共通定義の all_node オペランドおよび、all_node_ex オペランドの指定で OpenTP1 システムを構成する。

●all_node_extend_number=ドメイン変更時の最大ノード数

～〈符号なし整数〉((0～65535))《64》

namndchg コマンドまたは namchgfl コマンドでドメイン構成を変更する場合、変更後のドメインを構成する最大ノード数を指定します。

namndchg コマンドを使用する場合は、システム共通定義の all_node オペランドに指定するノードの最大数を指定してください。namchgfl コマンドを使用する場合は、all_node のドメイン定義ファイル中に指定するノードの最大数を指定してください。

このオペランドに指定されたノード数と、all_node オペランドに指定されたノード数または all_node のドメイン定義ファイル中に指定されたノード数とを合計した分の共用メモリの領域を確保します。

このオペランドに指定した値を超えるノード数を、システム共通定義またはドメイン定義ファイルに指定した場合、namndchg コマンドまたは namchgfl コマンドはエラーリターンし、KFCA00616-E および KFCA00654-E メッセージを出力します。この場合、ドメインは変更されないで、コマンド実行前のドメイン構成のままになります。

●all_node_ex_extend_number=ドメイン定義ファイルを使用したドメイン変更時の最大ノード数

～〈符号なし整数〉((0～65535))《64》

namchgfl コマンドに -e オプションを指定してドメイン構成を変更する場合、変更後のドメインを構成する最大ノード数を指定します。

all_node_ex のドメイン定義ファイル中に指定するノードの最大数を指定してください。

このオペランドに指定されたノード数と、all_node_ex のドメイン定義ファイル中に指定されたノード数とを合計した分の共用メモリの領域を確保します。

このオペランドに指定した値を超えるノード数をドメイン定義ファイルに指定した場合、namchgfl コマンドはエラーリターンし、KFCA00655-E メッセージを出力します。この場合、ドメインは変更されないで、コマンド実行前のドメイン構成のままになります。

●prc_current_work_path=カレントワーキングディレクトリ作成用のディレクトリパス

～ 〈パス名〉《\$DCDIR》

OpenTP1 プロセスのカレントワーキングディレクトリ作成用のディレクトリを絶対パスで指定します。実際のプロセスのカレントワーキングディレクトリは「このオペランドで指定したパス/tmp/home/プロセスごとに作成されたディレクトリ」になります。

このオペランドで指定したパスに誤りがある場合、またはこのオペランドの指定を省略した場合、カレントワーキングディレクトリ作成用のディレクトリは\$DCDIR になります。したがって、実際のプロセスのカレントワーキングディレクトリは「\$DCDIR/tmp/home/プロセスごとに作成されたディレクトリ」になります。

パス名に指定できる文字数は、最大 50 文字です。

OpenTP1 起動時、および dcreset コマンド実行時に、このオペランドで指定したディレクトリに tmp ディレクトリが存在する場合、いったん tmp ディレクトリが削除されたあと、再作成されます（コアファイルが存在する場合は、\$DCDIR/spool/save またはプロセスサービス定義の prc_coresave_path オペランドで指定したディレクトリに退避されます）。

なお、このオペランドで指定したディレクトリの tmp ディレクトリを削除の際、削除するディレクトリ下にユーザが作成したデータが多数存在すると、ディレクトリの削除に時間が掛かり、OpenTP1 の起動や dcreset コマンドの完了が遅くなります。

このため、ユーザプログラムから多数のデータを作成する環境の場合、次の対策を検討してください。

- このオペランドで指定したディレクトリ、および OpenTP1 ディレクトリ以外の場所にデータを作成する。
- OpenTP1 の停止前、および dcreset コマンド実行前に、このオペランドで指定したディレクトリのサーバ名下にある、ユーザプログラムが作成したデータを削除する。

このオペランドで指定するディレクトリには、次の権限が必要です。

- OpenTP1 管理者がディレクトリを作成できる権限
- OpenTP1 グループが上記のディレクトリにファイル（コアファイルなど）を作成できる権限

同一マシン上で複数の OpenTP1 を使用する場合、複数の OpenTP1 の prc_current_work_path オペランドに同じディレクトリを指定しないでください。指定したディレクトリが、ほかの OpenTP1 で使用されている場合、そのディレクトリは使用しないで、デフォルトのディレクトリを使用します。

このオペランドの指定を変更する場合、まず OpenTP1 を正常終了させてください。オペランドの変更後は dcreset コマンドを実行する必要があります。

このオペランドの指定を変更した場合、変更前にこのオペランドで指定していたディレクトリに、変更前に作成されたディレクトリが残っている場合があります。残っているディレクトリは、削除しても問題ありません。削除する場合は、OpenTP1 を正常終了させてから削除してください。その後、削除したディレクトリを再作成して再度使用する場合は、dcreset コマンドを実行する必要があります。

このオペランドとプロセスサービス定義の prc_coresave_path オペランドは、同じパーティション上に設定することをお勧めします。

このオペランドを指定する場合、オンラインテスト機能は使用しないでください。

●rpc_max_message_size=RPC 送受信電文の最大長

～ 〈符号なし整数〉 ((1～8)) 《1》 (単位：メガバイト)

RPC コールで送受信できるメッセージ（ユーザ電文）の最大長を指定します。

このオペランドを指定する場合は、システム共通定義の all_node オペランドに指定したすべてのノードで同じ値を指定してください。同じ値を指定しない場合、送信先ノードのノード間負荷バランス機能が実行されたときに、RPC コール (dc_rpc_call, dc_rpc_call_to) が DCRPCER_NET_DOWN でエラーリターンするおそれがあります。

このオペランドを指定して 1 メガバイトより大きなメッセージを送信する場合は、次のことに注意してください。

- SPP が起動している OpenTP1 ノードのバージョンは、このオペランドを使用できるバージョンにしてください。
SPP が起動しているすべての OpenTP1 ノードのバージョンが、このオペランドを使用できないバージョンの場合は、RPC コール (dc_rpc_call, dc_rpc_call_to) は DCRPCER_NO_SUCH_SERVICE_GROUP または DCRPCER_TRNCHK でエラーリターンします。
- サービス要求元の OpenTP1 ノードを開始してから、SPP を起動してください。
サービス要求元の OpenTP1 ノード開始前に SPP を起動した場合は、RPC コール (dc_rpc_call, dc_rpc_call_to) が DCRPCER_NO_SUCH_SERVICE_GROUP でエラーリターンするおそれがあります。

このオペランドを指定した場合、RPC コールで送信できるメッセージの最大長は、DCRPC_MAX_MESSAGE_SIZE の値 (1 メガバイト) ではなく、このオペランドに指定した値になります。ただし、rpc_max_message_size オペランドの値の指定を省略した場合、または rpc_max_message_size オペランドに 1 を指定した場合は、メッセージの最大長は、DCRPC_MAX_MESSAGE_SIZE の値 (1 メガバイト) になります。

このオペランドを指定しないで動作していた環境で、このオペランドの値に 2 以上を指定し、1 メガバイト以上のメッセージを送信した場合、メモリ不足が発生するおそれがあります。そのため、このオペランドを指定する場合は、実装するメモリ量やプロセスで使用できる最大メモリ量を考慮して指定してください。スケジューラサービスを使用した場合のメモリ量の最大値は次の式で求められます。

最大使用メモリ量 = `rpc_max_message_size`オペランド指定値×64×2（単位：メガバイト）

このオペランドの値に 2 以上を指定し、1 メガバイト以上のメッセージを送信した場合に RPC トレースを取得すると、メモリ使用量が増加します。また、RPC トレースの取得処理に時間が掛かるおそれがあります。

このオペランドを指定して次の機能を使用した場合の動作は保証できません。

- 通信先を指定した遠隔サービス (`dc_rpc_call_to`) のスケジューラデーモンのポート指定
- ネームサービスを使わない通信 (`dcsvgdef` 定義コマンド)
- ドメイン代表スケジューラサービス機能
- XATMI インタフェースを使用した通信
- ゲートウェイプログラムを使用した通信
- UAP テスタ機能（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）

このオペランドを指定して、1 メガバイト以上のメッセージを CUP へ一方通知する場合は、1 メガバイト以上のメッセージを受信できるバージョンの TP1/Client を使用してください。

このオペランドを指定して、データ圧縮機能を使用する場合は、圧縮後のメッセージ長が 1 メガバイト以下でも、RPC コール (`dc_rpc_call`, `dc_rpc_call_to`) の `in_len` の値が 1 メガバイトより大きければ、1 メガバイト以上のメッセージを送信すると判断します。

●`uap_trace_file_put=Y|N`

～《N》

UAP トレースのトレース情報を、ファイルに取得するかどうかを指定します。

Y

トレース情報を UAP トレースデータファイルに取得します。この場合、ユーザサーバを終了させないで、UAP トレースの編集出力ができるようになります。UAP トレースデータファイルについては、マニュアル「OpenTP1 運用と操作」を参照してください。

トレース情報をファイルに取得できなかった場合、プロセス固有領域にトレース情報を取得します。

N

トレース情報をプロセス固有領域に取得します。この場合、ユーザサーバをアボートしたときにだけ、UAP トレースの編集出力ができます。

Y を指定した場合、ユーザーバダウン時、または OpenTP1 終了後の再開始時に、UAP トレースデータファイルのバックアップファイルを各サーバで最大 6 世代取得します。バックアップファイルの内訳は、該当するサーバの正常終了時に 3 世代、コアファイルの出力を伴う異常終了時に 3 世代です。バックアップファイルの格納先は、プロセスサービス定義の `prc_coresave_path` オペランドに指定した、コアファイルの格納先です。

Windows の場合、UAP トレースの編集出力は `uatdump` コマンドで行います。`uatdump` コマンドについては、マニュアル「OpenTP1 テスタ・UAP トレース使用の手引」を参照してください。

このオペランドは、ユーザーサービス定義、ユーザーサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザーサービス定義
2. ユーザーサービスデフォルト定義
3. システム共通定義

●`dcstart_wakeup_retry_count=OpenTP1 起動通知エラーリトライ回数`

～〈符号なし整数〉((0～60))《0》

OpenTP1 の起動通知に失敗した場合に、リトライする回数を指定します。

`dcstart` コマンドで OpenTP1 を起動する場合、プロセスサーバに OpenTP1 の起動開始が通知されます。プロセスサーバへの起動通知に失敗すると、`dcstart` コマンドの機能によって、自動的にこのオペランドの指定回数分、起動通知をリトライします。

省略した場合、または 0 を指定した場合はリトライしません。

●`dcstart_wakeup_retry_interval=OpenTP1 起動通知エラーリトライ間隔`

～〈符号なし整数〉((1～60))《10》(単位：秒)

OpenTP1 の起動通知に失敗した場合に、リトライする間隔を指定します。

このオペランドを有効にするには、`dcstart_wakeup_retry_count` オペランドに 1 以上を指定する必要があります。

リトライの間隔は、プロセスサーバへの起動通知に失敗してから、次に起動通知が行われるまでの待ち時間になります。

●`nam_prf_trace_level=NAM イベントトレースの取得レベル`

～((00000000～00000007))《00000003》

性能検証用トレース情報の取得で、ネームサービス関係のトレース (NAM イベントトレース) の取得レベルを指定します。NAM イベントトレースのイベント ID は 0xf000 から 0xffff です。イベント ID の詳

細については、マニュアル「OpenTP1 運用と操作」を参照してください。なお、次の値以外を指定した場合は、00000003 を指定したと仮定されます。

00000000

NAM イベントトレースを取得しません。

00000001

サービスグループ情報の登録、削除などの NAM イベントトレース（イベント ID：0xf100～0xf1ff）を取得します。

00000002

ネームサーバ（namd）が行う他ノードとの通信処理に関する NAM イベントトレース（イベント ID：0xf000～0xf0ff）を取得します。

00000003

00000001 および 00000002 の NAM イベントトレースを取得します。

00000004

UAP および運用コマンドのプロセスとネームサーバ（namd）との通信処理に関する NAM イベントトレース（イベント ID：0xf200～0xf2ff）を取得します。

00000005

00000001 および 00000004 の NAM イベントトレースを取得します。

00000006

00000002 および 00000004 の NAM イベントトレースを取得します。

00000007

すべての NAM イベントトレース（00000001、00000002 および 00000004 の NAM イベントトレース情報）を取得します。

取得したトレースをファイル出力または編集出力するには、prfget コマンド、prfed コマンド、または dcalzprf コマンドを使用します。これらのコマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

●fil_prf_trace_option=0|1

～《1》

FIL イベントトレースを取得するかどうかを指定します。

0

FIL イベントトレースを取得しません。

OpenTP1 ファイルへのアクセス要求で、fil_prf_trace_delay_time オペランドの指定値以上の処理時間が掛かった場合、遅延情報として FIL イベントトレースを取得します。

このオペランドに 0 または 1 以外の値を指定しないでください。0 または 1 以外の値を指定した場合の動作は保証できません。

FIL イベントトレースで取得されるイベント ID は、0x6805, 0x6807, 0x6905, 0x6907, および 0x6909 です。

取得したトレースをファイル出力または編集出力するには、prfget コマンド、prfed コマンド、または dcalzprf コマンドを使用します。イベント ID の詳細、これらのコマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

●fil_prf_trace_delay_time=FIL イベントトレース取得条件となるファイルアクセス処理時間のしきい値 ～〈符号なし整数〉((1～65535))《3》(単位：秒)

FIL イベントトレースの取得条件となるファイルアクセス処理時間のしきい値を指定します。

注

しきい値を監視する時間精度は秒単位です。そのため、タイミングによっては、このオペランドに指定した値よりも短いファイルアクセス処理時間で FIL イベントトレースが取得されることがあります。しきい値が小さくなるほど、誤差の影響を受けやすくなりますので注意してください。

●jnl_prf_event_trace_level=JNL 性能検証用トレースの取得レベル ～((00000000～00000002))《00000001》

性能検証用トレース情報の取得で、JNL 性能検証用トレースの取得レベルを指定します。JNL 性能検証用トレースのイベント ID は 0xc000 から 0xcfff までです。イベント ID の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。なお、次の値以外を指定した場合は、00000001 を指定したと仮定されます。

00000000

JNL 性能検証用トレースを取得しません。

00000001

JNL 性能検証用トレース（イベント ID：0xc202, 0xc203, 0xc401, および 0xc402）を取得します。

00000002

JNL 性能検証用トレース（イベント ID：0xc001～0xc402）を取得します。

なお、取得レベルに 00000002 を指定した場合、すべてのトレース情報を取得できますが、オンライン性能に影響を与えるため、デバッグ時以外はデフォルトの出力レベルで運用することをお勧めします。

取得したトレースをファイル出力または編集出力するには、prfget コマンド、prfed コマンド、または dcalzprf コマンドを使用します。これらのコマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

●jnl_fileless_option=Y|N

～《N》

該当する OpenTP1 ノードでジャーナルファイルレス機能を使用するかどうかを指定します。

Y

ジャーナルファイルレス機能を使用します。

N

ジャーナルファイルレス機能を使用しません。

ジャーナルファイルレス機能を使用する場合、次の機能は使用できません。

- システムジャーナル機能
- グローバルアーカイブジャーナル機能
- トランザクション機能
- XA リソースサービスによるトランザクション機能
- DAM, TAM などの OpenTP1 提供の各種リソースマネージャ機能
- 各種統計情報取得機能

統計情報取得機能を使用してもエラーにはなりませんが、ジャーナルファイルレス機能を使用する場合は、統計情報が取得されません。

統計情報取得機能以外の機能を使用した場合は、OpenTP1 の開始処理がエラーになります。

トランザクション機能が使用できないため、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 atomic_update オペランドに Y を指定できません。Y を指定しても、N を指定したものとして動作します。

また、トランザクション機能が使用できないため、XA 連携機能も使用できません。このため、trnlncrm コマンドによる OpenTP1 に対するリソースマネージャの登録は不要です。

ジャーナルファイルレス機能を使用する場合に、使用できなくなる機能とコマンドなどの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

●prc_coredump_filter=core ファイルに共有メモリを含めるかの判断値

～〈符号なし整数〉((1, 3))

core ファイルに共有メモリを含めるかどうかを設定します。

1

含めません

3

含めます

このオペランドはシステム共通定義、ユーザサービスデフォルト定義、およびユーザサービス定義に指定できます。OpenTP1 のユーザサーバ、システムサーバで有効にするためには次のように設定します。

有効にしたいサーバ	システム共通定義	ユーザサービスデフォルト定義	ユーザサービス定義
システムサーバ	○	×	×
ユーザサーバ	○	○	○

(凡例)

○：含めます

×：含めません※

注※

システムサーバはサーバ単位に指定できません。また、ユーザサーバはサーバ単位に指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義

2. ユーザサービスデフォルト定義

3. システム共通定義

システム共通定義内のこのオペランドを変更した場合は、dcreset コマンドを実行してください。

このオペランドの指定値に従い、プロセス生成直後に OpenTP1 が /proc/self/coredump_filter を設定します。

/proc/self/coredump_filter の設定中にエラーが発生した場合は、KFCA00797-E メッセージが出力されます。

このオペランドを省略した場合は、OpenTP1 のプロセスに対して有効になっている OS の設定に従います。また、このオペランドは、Red Hat Enterprise Linux 5.2 以降に対して有効になります。対応 OS 以外の環境で指定した場合は無視されます。

OpenTP1 が提供する次のサービスでは、ユーザサービスデフォルト定義に指定した定義内容が有効になります。

- MQC ゲートウェイサービス
- リポジトリ管理サーバ
- リアルタイム統計情報サービス

- rap クライアントマネージャ, rap リスナー, および rap サーバ

注意事項

例えば、巨大な共有メモリを使用している環境で、core ファイルのサイズを極力抑えたい場合、このオペランドに 1 を指定することで core ファイルに共有メモリが含まれなくなり、core ファイルの大きさを抑えることができます。ただし、core ファイルに含まれている共有メモリは、その core を出力したプロセスがその瞬間参照していた共有メモリであって、トラブルシュートに大変有益な情報です。core を出力する状態自体が何か問題がある場合であり、この core に含まれている共有メモリ情報はその問題を解決するのに必要な情報です。

このオペランドに 1 を設定することで、そのトラブルシュートに必要な情報が一部失われ、問題解決に時間が掛かるおそれがあります。特に、システムサーバが core ファイルを出力する状態は、システムの異常な状態になっているおそれが高く、その瞬間の共有メモリがトラブルシュートに不可欠なケースがあります。

これらのことを踏まえた上で、このオペランドの指定値を決めてください。

●name_service_mode=manager|agent|normal

～《normal》

ノード自動追加機能を使用する場合の、ノードの動作モードを指定します。

このオペランドは、ノード自動追加機能を使用する場合に指定します。

manager

マネージャノードとして動作します。

agent

エージェントノードとして動作します。

normal

ノード自動追加機能を使用しません。ノーマルノードとして動作します。

manager を指定した場合、ネームサービス定義の name_audit_conf オペランドの指定を無視して、送受信型ノード監視機能によってエージェントノードとノーマルノードの稼働状況を監視します。そのため、同一ホストで複数のエージェントノードやノーマルノードを動作させる場合は、一つのホストに複数の異なる IP アドレスを設定し、それぞれのノードに異なる IP アドレスを設定してください。

agent を指定した場合、システム共通定義の name_manager_node オペランドにマネージャノードを指定してください。また、ネームサービス定義の name_audit_conf オペランドの指定内容によって、次の動作をします。

- name_audit_conf オペランドを省略、または 0 を指定した場合

送受信型ノード監視機能によって、マネージャノードの稼働状況を監視します。同時にノードリストの整合性を確保します。

- name_audit_conf オペランドに 1, または 2 を指定した場合

指定値を無視して送受信型ノード監視機能によって、マネージャノードの稼働状況を監視します。同時にノードリストの整合性を確保します。エージェントノード、およびノーマルノードとのノード監視は指定値に従って監視します。

manager または agent を指定した場合、ネームサービス定義の name_rpc_control_list オペランドに N が指定されたものとして動作します。

同一ホストで複数のマネージャノードを動作させる場合は、一つのホストに複数の異なる IP アドレスを設定し、それぞれのマネージャノードに異なる IP アドレスを設定してください。

ノードリストを引き継ぐ場合、OpenTP1 の開始は、ノードリストファイル中に格納された動作モードに従います。nammstr コマンドで動作モードを変更したとき、このオペランドに指定した動作モードと一致しないことがあります。OpenTP1 システムの現在の動作モードは、次のコマンドで確認できます。

- OpenTP1 のオフライン時：namnldsp コマンド
- OpenTP1 のオンライン時：namsvinf コマンドに -x オプションを指定

namnldsp コマンド、および namsvinf コマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

●name_manager_node=ノード名 [:ポート番号]

～ 〈1～255 文字の識別子〉 | 〈符号なし整数〉 ((5001～65535))

OpenTP1 システム内で稼働するマネージャノードのノード名、およびそのネームサーバがウェルノウンポート番号として使用するポート番号を指定します。

ポート番号を省略した場合は、name_port オペランドで指定したネームサービスのポート番号が仮定されます。

このオペランドは、ノード自動追加機能を使用する場合に、システム共通定義の name_service_mode オペランドに agent を指定したノード（エージェントノード）で指定します。agent 以外を指定したノードでこのオペランドを指定した場合は、無効になります。

エージェントノードは、OpenTP1 起動時に、このオペランドに指定したノードに対してノードリスト要求を送信します。ノードリストを正常に受信できた場合だけ、このオペランドで指定したノードをマネージャノードとして認識します。なお、このオペランドにマネージャノード以外のノード※を指定した場合は、ノードリストを正常に受信できないため、指定したノードをマネージャノードとして認識しません。また、マネージャノードからのノードリストを受信できないため、RPC の要求範囲は、オペランドを指定したノードに限定されます。

エージェントノードの開始時に、マネージャノードが起動していない場合や、ネットワークなどの障害によって、通信ができない状態の場合も、RPC の要求範囲がオペランドを指定したノードに限定されることがあります。

注※

name_service_mode オペランドに、manager 以外を指定しているノード、または name_service_mode オペランドをサポートしていないノード

参考

ノードリストを引き継ぐと、このようなマネジャノードとの通信障害などの影響を受けないで、前回オンラインの終了時と同等のノードリストが使用でき、RPC の要求範囲を確保できます。

●name_remove_down_node=Y|N

～《N》

マネジャノードが停止を検知したエーエージェントノード、およびノーマルノードのノード情報を、ノードリストから削除するかどうかを指定します。

このオペランドは、ノード自動追加機能を使用する場合に、マネジャノードで指定します。

Y

停止を検知したノードのノード情報をノードリストから削除し、マネジャノードの管理対象から除外します。

N

停止を検知したノードのノード情報をノードリストから削除しません。停止を検知したノードのノード情報を RPC 抑止リストへ登録します。

ノード情報をノードリストから削除したい場合は、マネジャノードで namndrm コマンドを実行してください。

次の場合は、ノードリストを削除しないで、次回のノードリストの整合性を確保するときに削除します。

- ・ マネジャノードがノードリストからノード情報を削除する際に、サービスグループ情報の検索など、ノードリストを参照する処理が並行して動作している場合
- ・ エーエージェントノードでノードリストの整合性を確保する際に、サービスグループ情報の検索など、ノードリストを参照する処理が並行して動作している場合

ノーマルノードを混在させて使用する運用の場合、このオペランドには N を指定してください。Y を指定すると、ノーマルノードの停止によって、マネジャノードのノードリストからノーマルノードの情報が削除されます。ノーマルノードの再開後も、マネジャノードとのコネクションが切断されているため、ノーマルノードのノード情報がノードリストに登録されません。また、マネジャノードやほかのエーエージェントノードのサービスグループ情報（起動、停止、負荷状態の変更）に関する通知がノーマルノードへ送信されません。ノード間の負荷バランスを保つために、このオペランドには N を指定してください。

●name_node_add_policy=using_only|all

～《using_only》

エージェントノードのノードリストに、ノーマルノードのノードの情報を登録するかどうかを指定します。

このオペランドは、ノード自動追加機能を使用する場合に、エージェントノードで指定します。

using_only

ノーマルノードの情報をノードリストに登録しません。

all

ノーマルノードの情報をノードリストに登録します。

このオペランドを省略、または using_only を指定したエージェントノードのノードリストには、ノーマルノードが登録されません。このエージェントノードとノーマルノード間では、サービスグループ情報（起動、停止、負荷状態の変更）に関する連携が行われません。ノーマルノードを使用し、かつ OpenTP1 システム内のノード間の負荷バランスを保つために、このオペランドに all を指定してください。

●watch_time=最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かるときがあります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しないときがあります。

このオペランドは、システム共通定義以外の幾つかのシステムサービス定義の中でも指定できます。各システムサービス定義で指定した場合は、各システムサービス定義で指定した値が優先されます。

このオペランドは、デフォルト値で使用してください。

システム共通定義だけにこのオペランドを指定すると、指定した定義値が OpenTP1 システム全体に有効となります。そのため、特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。チューニングを必要とする場合でも、ユーザーサービスに対する設定は、ユーザーサービスデフォルト定義で設定してください。

このオペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

次ページ以降に記述しています。

putenv 形式のオペランド

●LANG LANG 設定値

～〈文字列〉

環境変数 LANG を設定します。

LANG 設定値には、適用する OS で規定されている、言語種別を設定します。未設定の場合、または不当な値が設定された場合は、英語が設定されたものとします。

名称

OpenTP1 通信時または系切り替え機能使用時のホスト名の指定

形式

```
{{dcbindht -h ホスト名  
[-n ネットワーク名 [, ネットワーク名] ...] }}
```

機能

OpenTP1 が動作するホストが複数のネットワークアダプタで接続されている場合は、OpenTP1 がどのネットワークアダプタを使用して通信するかを指定します。一つのネットワークアダプタに複数の IP アドレスを付加できる場合は、OpenTP1 がどの IP アドレスを使用して通信するかを指定します。また、IP アドレスを引き継ぐ系切り替えを行うホストの場合も指定する必要があります。

この機能は、dcbindht 定義コマンドで指定されたホスト名に対応する IP アドレスで、OpenTP1 が動作しているということを、通信先の OpenTP1 に明示的に通知します。

ネットワークアダプタが一つで、かつ IP アドレスを一つしか持っていないマシンの場合、OpenTP1 が通信に使用する IP アドレスも物理的に一つに決まりますので、このコマンドを指定する必要ありません。

2：1 系切り替え構成や、相互系切り替え構成のように、IP アドレスを引き継ぐ系切り替えで、1 ホスト内に複数の OpenTP1 が稼働することがある場合、このコマンドを指定してください。なお、ルータなどを経由してネットワーク間を接続している場合は、ここで指定するネットワークアダプタを経由して通信するすべてのネットワークを指定しなければなりません。

指定を省略した場合、複数のネットワークアダプタ、または複数の IP アドレスのどれを使用してアクセスするかは、TCP/IP の制御で決定されます。ただし、複数の IP アドレスや複数のネットワークアダプタが接続されているマシンでも、指定を省略した場合、期待していない IP アドレスが通信先の OpenTP1 に通知され、通信時に通信障害が発生することがあります。

次に示すように、この定義のネットワーク名の指定を省略した場合は、OpenTP1 はすべての通信で、指定されたホスト名に対応する IP アドレスを使用します。この定義にネットワーク名の指定を省略した値を複数指定することはできません。複数指定した場合、先に定義された値が有効になり、あとに定義された値は無視されます。また、ネットワーク名が指定されている値とネットワーク名が省略されている値が混在する場合、ネットワーク名称が指定されている値が優先されます。

<システム共通定義>

```
dcbindht -h HOST_A -n NET_A,NET_B # 送信先ネットワークアドレスが  
# NET_A,NET_Bの場合、ホスト名称  
# HOST_Aをbindし、送信します。
```

```
dcbindht -h HOST_B # 送信先ネットワークアドレスが  
# NET_A,NET_B以外の場合、ホスト
```

```
dcbindht -h HOST_C      # 名称HOST_Bをbindし、送信します。
                          # この定義が指定されても、先に定義
                          # されたホスト名称HOST_Bが有効に
                          # なります。
```

サブネットを使用したネットワーク環境でこの定義を指定する場合は、システム共通定義の `rpc_netmask` オペランドも指定する必要があります。

サブネットを使用したネットワーク上で稼働させている場合、`rpc_netmask` オペランドを指定しないときには、`dcbindht` 定義コマンドに指定したネットワーク名と OpenTP1 の認識するネットワークアドレスが一致しないため、`dcbindht` 定義コマンドの指定値が有効になりません。なお、`rpc_netmask` に指定するサブネットマスクは、OpenTP1 のグローバルドメイン下のネットワークですべて同一でなければなりません。

オプション

●-h ホスト名

～〈1～255 文字の識別子〉

OpenTP1 の通信に使用する、ホスト名、またはネットワークアダプタに対応したホスト名を指定します。一つのネットワークアダプタに複数の IP アドレスを付加できる場合に、IP アドレスを引き継ぐ系切り替えを行うときは、引き継ぎたい IP アドレスに対応したホスト名を指定します。識別子に使用できる文字は英数字、ピリオド、およびハイフンです（ただし、IP アドレス形式は除く）。ホスト名は、`/etc/hosts` ファイルまたは DNS など、IP アドレスとのマッピングができなければなりません。ホスト名に `localhost` または IP アドレスが 127 で始まるホスト（例：127.0.0.1）を指定した場合、通信時に通信障害が発生するおそれがあります。

なお、`dc_rpc_call_to()`を使用する場合、このオプションに指定したホスト名と `my_host` オペランドに指定したホスト名が異なるときは、動作を保証しません。

●-n ネットワーク名

～〈1～64 文字の識別子〉

-h オプションで指定したネットワークアダプタまたは IP アドレスを経由して通信するネットワーク名を指定します。ネットワーク名は、`/etc/networks` ファイルまたは NIS など、ネットワーク番号とのマッピングができなければなりません。

このオプションは省略できます。省略した場合、OpenTP1 はすべての通信で-h オプションで指定されたホスト名に対応したネットワークアダプタまたは IP アドレスを使用します。

注意事項

- 複数の IP アドレスを持っているホストで、複数の IP アドレスを同一ネットワークに接続する場合、`dcbindht` 定義コマンドの-h オプションで指定するホスト名称を `my_host` オペランドにも指定してください。OpenTP1 では、コネクション接続先と `my_host` オペランドで指定された情報を対で管理し

ています。my_host オペランドに指定した値と異なるホスト名称を dcbindht 定義コマンドの -h オプションに指定した場合、接続先が同じであっても、複数のコネクションを使用する可能性があります。

- 1 ホスト内で複数の OpenTP1 が稼働し、IP アドレスを引き継ぐ系切り替え構成の場合は、必ずこの指定を定義してください。指定を省略した場合、目的の OpenTP1 と通信できないことがあります。

図 3-2 で示す構成では、OpenTP1 B から OpenTP1 A への通信には IP アドレス ip=a または ip=x を使用します。

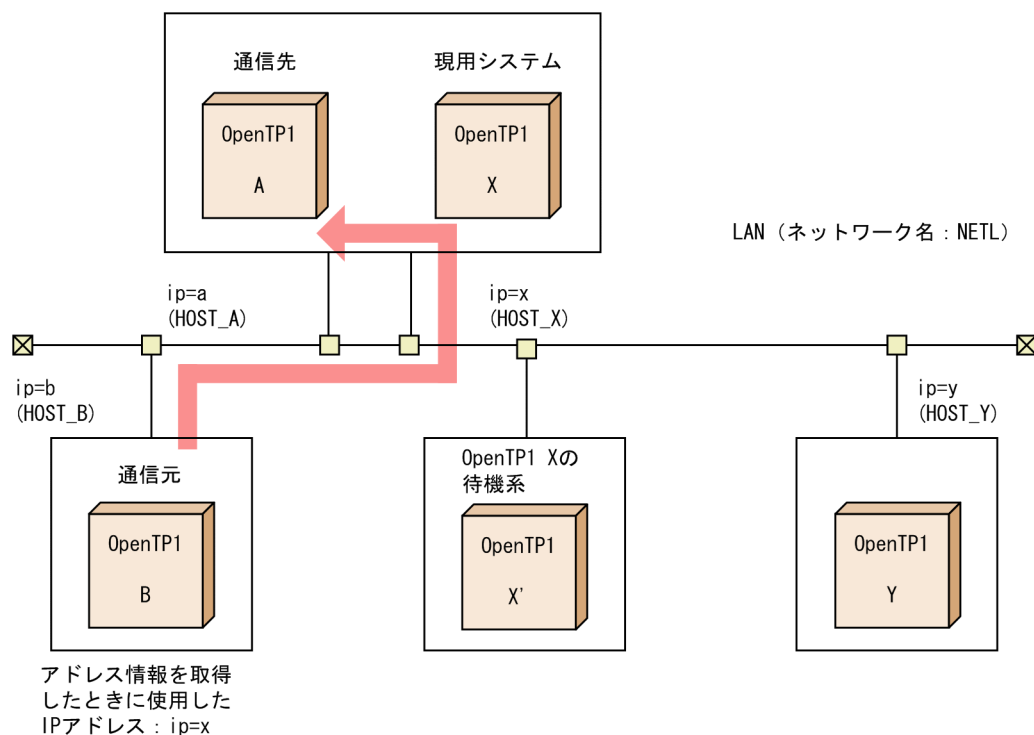
OpenTP1 は、サービスの要求を受けると、サービスのアドレス情報を取得するために使用した IP アドレスを記憶します。

IP アドレス ip=x を使用してアドレス情報を取得したあとに、系切り替えが発生すると、IP アドレス ip=x は OpenTP1 X から OpenTP1 X' に引き継がれますが、一方で、OpenTP1 B はアドレス情報を取得した時に使用した IP アドレスを ip=x と記憶したままです。この状態でサービス要求を送信すると、OpenTP1 A に送信したはずのサービス要求が、OpenTP1 X' に送信されてしまうことがあります (図 3-3)。

目的の OpenTP1 と通信するためには、次のように定義して通信に使用するネットワークアダプタを特定してください。

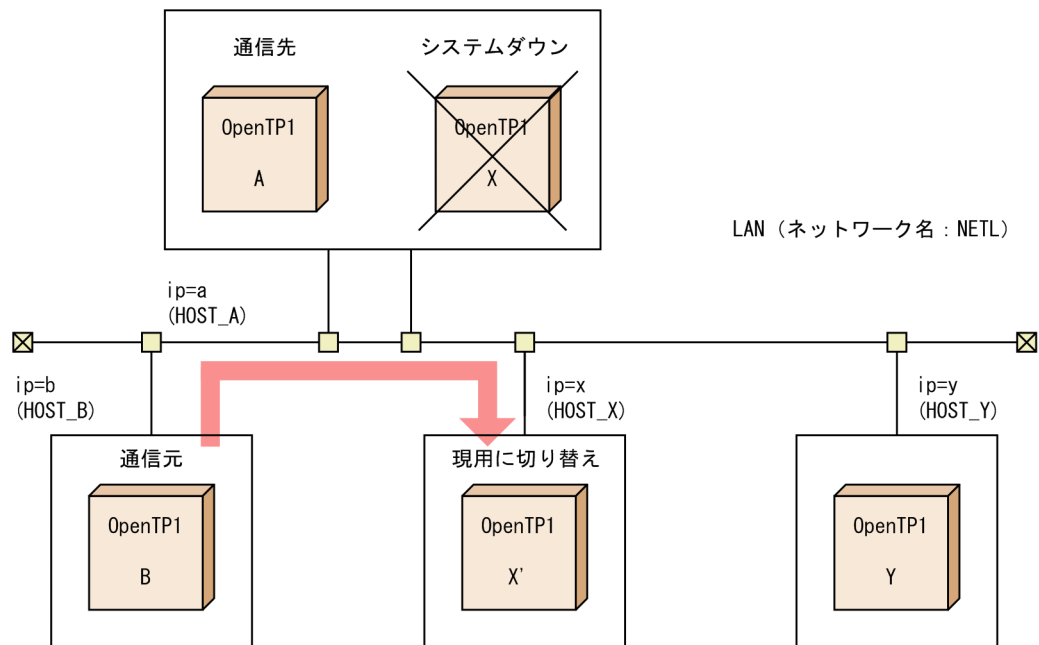
- OpenTP1 A では
dcbindht -h HOST_A -n NETL
- OpenTP1 X では
dcbindht -h HOST_X -n NETL

図 3-2 系切り替え前の通信形態



dcbindht 定義コマンドを指定すると，OpenTP1 A への通信には IP アドレス ip=a を使い，OpenTP1 X への通信には IP アドレス ip=x を使って通信します。

図 3-3 系切り替え後の通信形態



ロックサービス定義

形式

set 形式

```
[set lck_limit_foruser=ユーザーサーバの最大同時排他要求数]
[set lck_limit_fordam=DAMサービスの最大同時排他要求数]
[set lck_limit_fortam=TAMサービスの最大同時排他要求数]
[set lck_limit_formqa=MQAサービスの最大同時排他要求数]
[set lck_wait_timeout=排他待ち限界経過時間]
[set lck_deadlock_info=Y|N]
[set lck_deadlock_info_remove=normal|force|no]
[set lck_deadlock_info_remove_level=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レベル]
[set lck_release_detect=interval|pipe]
[set lck_release_detect_interval=排他解除検知インタバル最大時間]
[set lck_prf_trace_level=LCK性能検証用トレース情報の取得レベル]
```

コマンド形式

なし。

機能

OpenTP1 システムで排他制御機能を使用するための実行環境を定義します。

説明

set 形式のオペランド

●lck_limit_foruser=ユーザーサーバの最大同時排他要求数

～ 〈符号なし整数〉 ((0～327670)) 《512》

このノード上にあるユーザーサーバから同時に発生する排他要求数の最大値を指定します。0 を指定した場合は、排他制御機能は使用できません。

●lck_limit_fordam=DAM サービスの最大同時排他要求数

～ 〈符号なし整数〉 ((0～327670)) 《512》

このノード上にある DAM サービスから同時に発生する排他要求数の最大値を指定します。排他要求数の計算式を次に示します。

$$\text{lck\_limit\_fordam} = \sum_{i=1}^t T_i + \sum_{i=1}^p P_i + 1※$$

t：同時に実行するトランザクションの数

p：回復対象外アクセスで同時に実行するプロセスの数

T：1 トランザクションブランチでオープンするファイル数+参照，更新するブロック数

P：回復対象外アクセス処理の 1 プロセスでオープンするファイル数+参照，更新するブロック数

注※

計算式に+1 が反映されない場合，damrm コマンドの実行時に，KFCA01613-E メッセージが出力されることがあります。

●lck_limit_fortam=TAM サービスの最大同時排他要求数

～ 〈符号なし整数〉 ((0～327670)) 《512》

このノード上にある TAM サービスから発生する排他要求数の最大値を指定します。排他要求数の計算式を次に示します。

$$\text{lck\_limit\_fortam} = \sum_{i=1}^t (T_i + R_i)$$

t：同時に実行するトランザクションの数

T：1 トランザクションで参照，更新をするテーブル数

R：1 トランザクションで参照，更新をするレコード数

●lck_limit_formqa=MQA サービスの最大同時排他要求数

～ 〈符号なし整数〉 ((0～327670)) 《0》

このノード上にある MQA サービスから同時に発生する排他要求数の最大値を指定します。排他要求数の計算式については，マニュアル「TP1/Message Queue 使用の手引」の定義情報の作成手順に関する記述を参照してください。

●lck_wait_timeout=排他待ち限界経過時間

～ 〈符号なし整数〉 ((0～32767)) 《0》 (単位：秒)

排他要求が待ち状態になってから解除されるまでの排他待ち時間を監視する限界経過時間を指定します。

指定時間を経過しても待ち状態が解除されない場合，ロックサービスは待ち時間の超過として，排他要求をエラーリターンします。

0 を指定した場合は，排他待ちの時間監視をしません。

●lck_deadlock_info=Y|N

～ 《Y》

デッドロック情報とタイムアウト情報を出力するかどうかを指定します。

OpenTP1 はデッドロック，またはタイムアウトが発生するたびに\$DCDIR/spool/dclckinf の下にファイルを作成して，情報を出力します。不要となったファイルは削除してください。

Y

デッドロック情報とタイムアウト情報を出力します。

N

デッドロック情報とタイムアウト情報を出力しません。

●lck_deadlock_info_remove=normal|force|no

～《no》

OpenTP1 開始時に，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を削除するかどうかを指定します。

normal

OpenTP1 正常開始時に，削除します。

force

OpenTP1 開始時に，削除します。

no

OpenTP1 開始時に，削除しません。

●lck_deadlock_info_remove_level=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レベル

～〈符号なし整数〉((0～24855))《0》(単位：日)

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レベルを指定します。

lck_deadlock_info_remove オペランドで normal か force を指定した場合に有効です。

0

全ファイルを削除します。

1～24855

ロックサービス開始時刻から起算して「指定値×24」時間以前に作成されたファイルを削除します。

●lck_release_detect=interval|pipe

～《pipe》

排他が競合した場合，占有しているプロセスがいつ排他を解除したかを調べる方法を指定します。プロセスが排他待ちとなった場合に有効です。

interval

共用メモリの排他管理用領域を一定間隔で調べます。

50 ミリ秒から `lck_release_detect_interval` オペランドの指定値まで、徐々に間隔を大きくして占有プロセスが排他を解除したかを調べます。

排他占有プロセスが排他を解除しても、次のインタバルまで検索できないため、排他待ち時間が長くなる場合があります。

pipe

pipe ファイルを使用して排他解除を調べます。

占有プロセスから排他待ちプロセスに、排他解除が通知されます。したがって、排他占有プロセスが排他を解除したとき、排他待ちプロセスは排他解除を検知できます。

排他が競合したトランザクションのダイナミックステップ数は、interval 指定時より増加します。

OpenTP1 では、次の計算式の値分の pipe ファイルを同時に使用する場合がありますので、ファイル記述子の見積もりに次の計算式の値を加えてください。

同時に排他を占有するプロセス数+同時に排他待ちになるプロセス数+トランザクション回復プロセス数

●`lck_release_detect_interval`=排他解除検知インタバル最大時間

～〈符号なし整数〉((10～60000))《250》(単位：ミリ秒)

排他待ちプロセスが、排他解除を調べるインタバルの最大時間を指定します。`lck_release_detect` オペランドに interval を指定した場合に有効です。

50 より大きな値を指定すると、50 ミリ秒からこの指定値になるまで徐々に間隔を大きくして、プロセスの排他解除を調べます。50 以下の値を指定すると、指定された時間間隔でプロセスの排他解除を調べます。統計情報を基に指定値を決定してください。

●`lck_prf_trace_level`=LCK 性能検証用トレース情報の取得レベル

～((00000000～00000001))《00000000》

LCK 性能検証用トレースの取得レベルを指定します。

LCK 性能検証用トレースの詳細については、マニュアル「OpenTP1 解説」の障害の原因解析の説明を参照してください。

00000000

LCK 性能検証用トレースを取得しません。

00000001

LCK 性能検証用トレースを取得します。

取得したトレースをファイル出力または編集出力するには、`prfget` コマンド、`prfed` コマンド、または `dcalzprf` コマンドを使用します。ファイルの出力結果は、`$DCDIR/spool/dclckinf/prf` に保存されます。これらのコマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

コマンド形式

なし。

タイマサービス定義

形式

set 形式

```
[set tim_watch_count=最大時間監視サービス数]
```

コマンド形式

なし。

機能

時間監視サービスを行うための実行環境を定義します。

説明

set 形式のオペランド

●tim_watch_count=最大時間監視サービス数

～ 〈符号なし整数〉 ((0～65535)) 《128》

最大時間監視サービス数を指定します。

OpenTP1 システムでは、トランザクションブランチ処理内、および MHP 内で時間監視をします。また、サービス関数の実行時間を監視します。最大時間監視サービス数の計算式を次に示します。

$$\text{最大時間監視サービス数} = A + B + C + D$$

A：同時に起動するトランザクションブランチ数

トランザクションサービス定義の trn_tran_process_count オペランドの指定値。

B：非トランザクション MHP 限界経過時間の監視を指示した MHP プロセス数

次に示す条件を両方とも満たす MHP を同時に実行する数。

- アプリケーション属性定義 (mcfaalcap) の -n オプションで, trnmode オペランドに nontrn を指定
- アプリケーション属性定義 (mcfaalcap) の -v オプションまたは UAP 共通定義 (mcfmuap) の -u オプションで, ntmetim オペランドに 0 以外の値を指定

C：サービス関数実行時間の監視を指示した SPP プロセス数

ユーザサービス定義, ユーザサービスデフォルト定義の service_expiration_time オペランドが有効になる SPP プロセス数。SPP をマルチサーバで運用する場合は, 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の parallel_count オペランドの指定値を加算してください。

D：トランザクション完了限界時間の監視を指示したプロセス数

trn_completion_limit_time オペランド※が有効となる，次に示すプロセス数の合計値。

- トランザクションを実行する SUP，SPP および MHP のプロセス数。SPP および MHP をマルチサーバで運用する場合は，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の parallel_count オペランドの指定値を加算してください。
- トランザクショナル RPC 実行プロセス数。クライアントサービス定義の parallel_count オペランドの指定値を加算してください。
- トランザクションを実行する CUP 実行プロセス数。クライアントサービス定義の cup_parallel_count オペランドの指定値を加算してください。
- トランザクションを実行する rap サーバ数。rap サーバをマルチサーバで運用する場合は，rap リスナーサービス定義の rap_parallel_server オペランドの指定値を加算してください。

注※

ユーザサービス定義，ユーザサービスデフォルト定義，rap リスナーサービス定義，クライアントサービス定義，またはトランザクションサービス定義の trn_completion_limit_time オペランドです。

同一ノード内にトランザクションサービス，MCF，およびサービス関数実行時間の監視を指定した SPP がない場合は，0 を指定します。

0 を指定した場合，タイマサービスは時間監視サービスをしません。

ここで指定した値を超えて時間監視サービスを要求された場合，タイマサービスはサービスを提供できなくなります。トランザクションブランチ処理時間を監視できない場合はトランザクション開始を要求する API をエラーリターンさせます。また，サービス関数の実行時間またはトランザクション完了限界時間を監視できない場合は，エラーメッセージを出力して処理を続行します。

コマンド形式

なし。

ネームサービス定義

形式

set 形式

```
[set name_total_size=サービス情報領域の大きさ]
[set name_cache_size=サービス情報キャッシュ領域の大きさ]
[set max_socket_descriptors=ソケット用ファイル記述子の最大数]
[set name_global_lookup=Y|N]
[set name_service_extend=1|0]
[set name_audit_conf=2|1|0]
[set name_audit_interval=監視間隔時間]
[set name_audit_watch_time=ノードダウンを検知するまでの最大待ち時間]
[set name_rpc_control_list=Y|N]
[set name_nodeid_check_message=Y|N]
[set name_cache_validity_time=他ノードのサービス情報の有効時間]
[set watch_time=最大応答待ち時間]
[set name_start_watch_time=OpenTP1起動時のノードリスト要求応答待ち時間]
[set name_start_retry_count=OpenTP1起動時のノードリスト要求リトライ回数]
[set name_start_retry_interval=OpenTP1起動時のノードリスト要求リトライ間隔]
[set name_start_error= stop|continue]
[set name_sync_ready_time=ノードリスト同期待ち時間]
```

コマンド形式

```
[namnlfil - {r|n} ノードリストファイル名]
```

機能

RPC を可能にするサービス名とアドレスの対を管理するための実行環境を定義します。

ネームサービスは、ここで指定するサービス情報領域とサービス情報キャッシュ領域の大きさを加算した領域を、システム環境定義で指定した静的共用メモリの中から確保します。

説明

set 形式のオペランド

●name_total_size=サービス情報領域の大きさ

～ 〈符号なし整数〉 ((1～32767)) 《64》 (単位：キロバイト)

自ノードのネームサービスが確保するサービス情報領域の大きさを指定します。ノード自動追加機能を使用する場合は、必ず指定します。

サービス情報領域の大きさは、システムサービスとユーザサーバで登録するサービス情報の数を基に計算します。計算式については、「[付録 B.1 TP1/Server Base 用の共用メモリの見積もり式](#)」を参照してください。

ノード自動追加機能を使用する場合の計算式を次に示します。

システム共通定義の name_service_mode オペランドに manager, または agent を指定した場合

```
name_total_size= ↑ (7936+all_node_exのノード数×284
                    +all_node_ex_extend_numberオペランドの指定値
                      ×284
                    +163840
                    +((SPP数※1+rapサーバ数※2+XATMI数※3)×264)
                    +MCF※4+DAM※5+TAM※6+CLT※7)／1024 ↑
```

システム共通定義の name_service_mode オペランドを省略, または normal を指定した場合

```
name_total_size= ↑ (7616+(all_nodeのノード数×384
                    +all_node_exオペランドの指定値
                      ×284
                    +all_node_extend_numberオペランドの指定値×384
                    +all_node_ex_extend_numberオペランドの指定値
                      ×284)
                    +((SPP数※1+rapサーバ数※2+XATMI数※3)×264)
                    +MCF※4+DAM※5+TAM※6+CLT※7)／1024 ↑
```

(凡例)

↑ ↑ : 小数点以下を切り上げます。

注※1

サービスグループの数になります。

注※2

rap サーバを起動する場合は, rap サービス数を指定します。

注※3

XATMI インタフェースの関数を使用する場合は, XATMI の SPP で定義したサービスの総数になります。

注※4

TP1/Message Control を使用する場合は, 次の算出式で求めた値になります。

(1 + MCF 通信サーバの数 + MHP 数) × 264

注※5

TP1/FS/Direct Access を使用する場合は, 936 になります。

注※6

TP1/FS/Table Access を使用する場合は, 1424 になります。

注※7

クライアント拡張サービス機能を使用する場合は, 936 になります。

●name_cache_size=サービス情報キャッシュ領域の大きさ

～ 〈符号なし整数〉 ((1～32767)) 《16》 (単位: キロバイト)

自ノードのネームサービスが確保するサービス情報キャッシュ領域の大きさを指定します。

サービス情報キャッシュ領域の大きさは、他ノードに登録されているサービス情報のうち、自ノードから検索を要求するサービス情報の数を基に計算します。計算式については、「付録 B.1 TP1/Server Base 用の共用メモリの見積もり式」を参照してください。

ノード自動追加機能を使用する場合の計算式を次に示します。

システム共通定義の name_service_mode オペランドに manager, または agent を指定した場合

```
name_cache_size=↑(ノードリストに登録されているノードおよび  
all_node_exオペランドに指定したノード※1で起動する  
SPP数※2の総計  
×224) /1024↑
```

システム共通定義の name_service_mode オペランドを省略, または normal を指定した場合

```
name_cache_size=↑(all_nodeオペランドおよび  
all_node_exオペランドに指定したノード※3で起動する  
SPP数※2の総計  
×224) /1024↑
```

(凡例)

↑↑：小数点以下を切り上げます。

注※1

システム共通定義の all_node_ex オペランドに指定したノードか、またはドメイン定義ファイルに指定したノードです。

注※2

サービスグループの数になります。

注※3

システム共通定義の all_node オペランド, および all_node_ex オペランドに指定したノードか、またはドメイン定義ファイルに指定したノードです。

ネームサービスは、サービス情報領域が不足したとき、サービス情報キャッシュ領域をサービス情報領域として使用することがあります。逆に、サービス情報領域をサービス情報キャッシュ領域として使用することはありません。

サービス情報キャッシュ領域を大きくすると、他ノードのサービス情報を検索するときの応答性能が向上します。

ただし、必要以上に大きくすると、OpenTP1 システム全体のメモリを圧迫します。

●max_socket_descriptors=ソケット用ファイル記述子の最大数

～〈符号なし整数〉((32～2032))

OpenTP1 制御下のプロセス※でのソケット用に使用するファイル記述子の最大数を指定します。

OpenTP1 制御下のプロセス※は、システムサーバやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数、および通信する他ノードの数によって、このオペランドを変更する必要があります。

注※

MCF サービス（MCF マネージャサービス、MCF 通信サービス、およびアプリケーション起動サービス）以外の OpenTP1 プロセスが対象です。MCF サービスの場合は、「[システムサービス情報定義](#)」および「[システムサービス共通情報定義](#)」を参照してください。

ソケット用ファイル記述子の最大数の算出式を、次に示します。

システム共通定義の name_service_mode オペランドに manager、または agent を指定した場合
2032 を指定してください。

システム共通定義の name_service_mode オペランドを省略、または normal を指定した場合
 $\uparrow (A + B + C) \nearrow 0.8 \uparrow$

計算式の凡例を次に示します。

$\uparrow \uparrow$ ：小数点以下を切り上げます。

A：自ノード内で、次の値の合計値です。

1. OpenTP1 の開始から終了までに起動する UAP のプロセス数
2. クライアントサービス定義の parallel_count オペランド指定値
3. システムサービスプロセス数

システムサービスプロセス数については、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造の記述を参照してください。

4. 次のどちらかの値

システム共通定義の name_domain_file_use オペランドに N を指定している場合

システム共通定義の all_node オペランドと all_node_ex オペランドに指定しているノードの数

システム共通定義の name_domain_file_use オペランドに Y を指定している場合

ドメイン定義ファイル（\$DCCONFPATH/dcnamnd、および \$DCCONFPATH/dcnamndex ディレクトリ下のファイル）に指定しているノードの数

5. 自ノードで稼働する UAP で、dc_rpc_call_to 関数を実行する場合のあて先に指定しているノード数
ただし、A の 4.と重複したノードは除きます。

B：他ノードの OpenTP1 システムで、次の 1.~2.に該当するノードの数の合計値です。

1. 次のどちらかの値

システム共通定義の name_domain_file_use オペランドに N を指定している場合

自ノード名をシステム共通定義の all_node オペランド、または all_node_ex オペランドに指定しているノードの数を 4 倍した値

システム共通定義の name_domain_file_use オペランドに Y を指定している場合

自ノード名をドメイン定義ファイル (\$DCCONFPATH/dcnamnd, および \$DCCONFPATH/dcnamndex ディレクトリ下のファイル) に指定しているノードの数を 4 倍した値

2. 自ノード名を dc_rpc_call_to 関数のあて先に指定した UAP が稼働する他ノード数

ただし A の 4. および 5., B の 1. のどれかと重複したノードを除きます。

C: 自ノードのネームサービスのポート番号に接続する CUP プロセス数, およびスレッド数

このオペランドの指定が小さいと, OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため, プロセスが KFCA00307-E メッセージを出力して異常終了します。

指定値の優先順位は次のとおりです (1.> 2.)。

1. ネームサービス定義

2. システム共通定義

ここで指定を省略した場合, システム共通定義の値を仮定します。

●name_global_lookup=Y|N

～《N》

グローバル検索機能を使用するかどうかを指定します。

グローバル検索機能とは, all_node オペランドで指定された各ノード上のネームサービスがキャッシュしている他ノードのサービス情報を取得する機能です。

Y

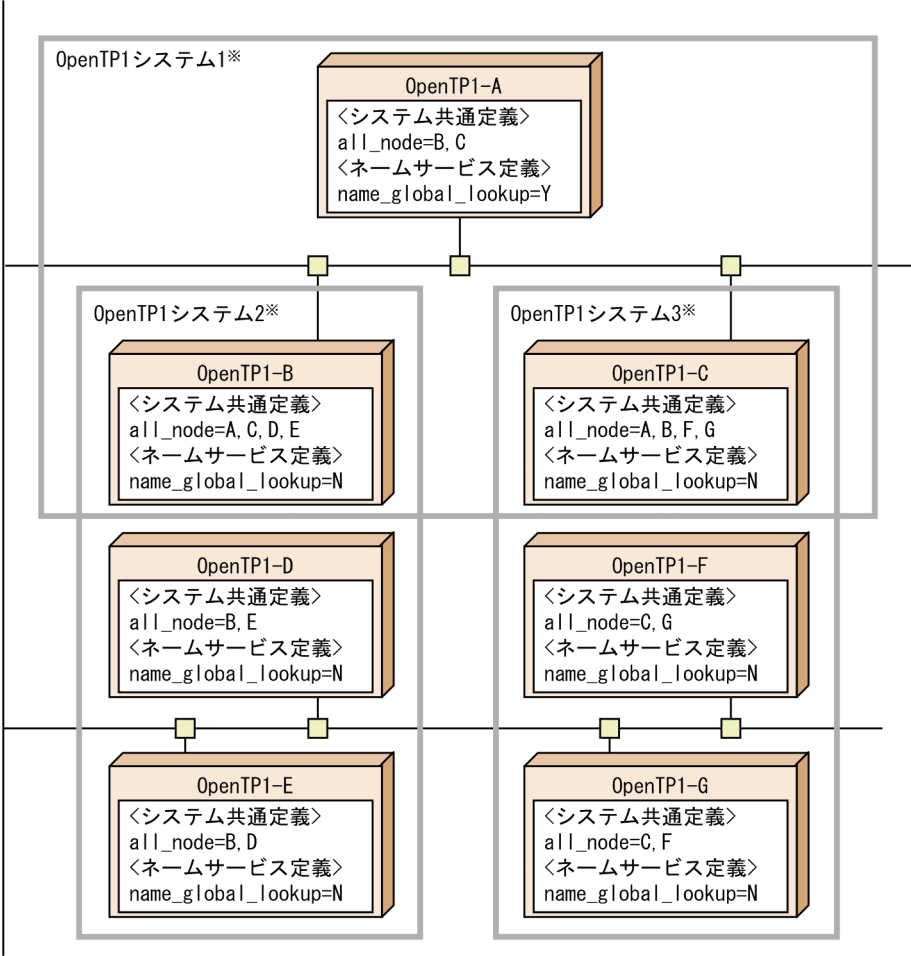
グローバル検索機能を使用します。

N

グローバル検索機能を使用しません。

グローバル検索機能使用時のシステム構成例を次の図に示します。

図 3-4 グローバル検索機能使用時のシステム構成



注 OpenTP1-B～OpenTP1-Gのバージョンは、03-02以降でなければなりません。
注※ OpenTP1システムとは、各OpenTP1ノードのall_nodeオペランドで指定されているOpenTP1ノード群を指します。

この図で示したシステム構成例の場合に、各 OpenTP1 ノードで指定するオペランドの値を次の表に示します。

表 3-1 グローバル検索機能を使用したシステムのオペランドの指定値

OpenTP1 ノード名	all_node オペランドの指定値	name_global_lookup オペランドの指定値
OpenTP1-A	B, C	Y
OpenTP1-B	A, C, D, E	N
OpenTP1-C	A, B, F, G	N
OpenTP1-D	B, E	N
OpenTP1-E	B, D	N
OpenTP1-F	C, G	N
OpenTP1-G	C, F	N

OpenTP1-A からの検索範囲は、OpenTP1-B～OpenTP1-G となります。つまり、OpenTP1 システム 1 だけでなく、OpenTP1 システム 2 および 3 までのサービスと通信できるようになります。

グローバル検索機能を使用した環境では、DCRPC_BINDTBL_SET 関数の引数 portno にネームサービスのポート番号を指定した、dc_rpc_call_to 関数は使用できません。

DCRPC_BINDTBL_SET 関数の引数 nid にノード識別子を指定した、dc_rpc_call_to 関数を使用する場合は、必ずグローバル検索機能の検索範囲（図の OpenTP1 システム 1～3）内のすべての OpenTP1 ノードのノード識別子を一意に定義してください。

サービス情報（閉塞、負荷状態など）は、検索元 OpenTP1（図の OpenTP1-A）まで通知されないため、図のように水平分散を利用したサービス要求を行うことをお勧めします。

ネームサービス定義の name_cache_size オペランドの指定値は、自ノードから検索を要求するサービス情報の数に、all_node オペランドで指定されたノードにキャッシュされるサービス情報の数まで含めて算出する必要があります。

グローバル検索機能を使用した OpenTP1（図の OpenTP1-A）に対し、TP1/Client/P、TP1/Client/W または TP1/Client/J からサービス要求が送信された場合、OpenTP1 システム 1 だけでなく、OpenTP1 システム 2 および 3 のサービス情報まで取得します。

グローバル検索機能の詳細については、マニュアル「OpenTP1 解説」を参照してください。

●name_service_extend=1|0

～《0》

ネームサービスで管理するサーバ UAP の取得数を拡張します。このオペランドに 1 を指定した場合、RPC 先の各ノードで起動する同一サービスグループ名のサーバ UAP に対し、RPC 先の対象を最大 512 にできます。ただし、RPC 先の各ノードは、all_node オペランドで指定した OpenTP1 システムに含まれている必要があります。このオペランドに 0 を指定した場合、または指定を省略した場合、RPC 先の対象は、最大 128 のままになります。

●name_audit_conf=2|1|0

～《0》

ノード監視機能を使用するかどうかを指定します。ノード監視機能の詳細については、マニュアル「OpenTP1 解説」を参照してください。

2

送受信型ノード監視機能を使用します。

1

一方送信型ノード監視機能を使用します。

0

ノード監視機能を使用しません。

このオペランドを指定する場合、次の点に注意してください。

- この機能は、ノードごとにユニークな IP アドレスが割り振られていることを前提としているため、同じ IP アドレスを使用するノードをシステム共通定義の `all_node` オペランドまたは `all_node_ex` オペランドに複数指定しているノードや、系切り替え後に同一の IP アドレスで複数の OpenTP1 が起動する（1LAN ボードだけ使用時など）ノードに対しては、ノード監視機能は使用できません。
- ノード監視機能の監視用通信処理で、ノードダウン検知の感度をチューニングする場合、次の定義オペランドを変更してください。

`name_audit_conf` に 1 を指定した場合

システム共通定義の `ipc_conn_interval` を変更してください。

`name_audit_conf` に 2 を指定した場合

ネームサービス定義の `name_audit_watch_time` を変更してください。

- ノード監視機能で同時に監視できるノード数は 60 ノードまでです。システム共通定義の `all_node` オペランド、および `all_node_ex` オペランドに指定したノード数が 60 を超える場合、60 ノード単位で監視を繰り返します。
- システム共通定義の `all_node` オペランド、および `all_node_ex` オペランドに多くのノードを指定している場合、ノード監視機能を使用すると UAP で実行する RPC に影響を及ぼす場合があります。この場合、`name_audit_interval` オペランドに小さい値を指定しないでください。
- `name_rpc_control_list` オペランドを省略するか、Y を指定した場合、`name_audit_interval` オペランドに指定した時間が経過していても、障害から復旧したノードが RPC 抑止リストから削除されることがあります。この場合、KFCA00651-I メッセージは出力されません。
- `name_rpc_control_list` オペランドを省略するか、Y を指定した場合で `name_audit_conf` オペランドに 2 を指定しているとき、KFCA00650-I メッセージが定期的に出力されることがあります。
- `name_audit_conf` オペランドに 1 または 2 を指定し、かつ `name_audit_interval` オペランドに 180 以下を指定した場合、`name_rpc_control_list` オペランドに N を指定することをお勧めします。
- `name_audit_conf` オペランドを省略するか、または 0 を指定した場合、`name_rpc_control_list` オペランドに N を指定したときは、ノード監視機能および RPC 抑止リストへの監視機能は無効になるので次に示す状態になります。
 - RPC 抑止リストに登録したノードから自ノードに対して通信が発生しないかぎり、RPC 抑止リストから該当ノードは削除されません。
 - RPC 抑止リストに登録したノードの `all_node` オペランドに自ノードを指定していない場合、自ノードの OpenTP1 を再起動しないかぎり、RPC 抑止リストから該当ノードが削除されません。

上記の状態を防ぐためには、`name_audit_conf` オペランドを省略するか、または 0 を指定した場合、`name_rpc_control_list` オペランドを省略するか、Y を指定することをお勧めします。

ノード自動追加機能を使用する場合、次の点に注意してください。

- システム共通定義の `name_service_mode` オペランドに `manager` を指定した場合、このオペランドの指定を無視して、送受信型ノード監視機能によって、エージェントノードとノーマルノードの稼働状況

を監視します。そのため、同一ホストで複数のエージェントノードやノーマルノードを動作させる場合は、一つのホストに複数の異なる IP アドレスを設定し、それぞれのノードに異なる IP アドレスを設定してください。

- システム共通定義の `name_service_mode` オペランドに `agent` を指定した場合、システム共通定義の `name_manager_node` オペランドにマネージャノードを指定してください。また、ネームサービス定義の `name_audit_conf` オペランドの指定内容によって、次の動作をします。
 - `name_audit_conf` オペランドを省略した場合、または 0 を指定したとき
送受信型ノード監視機能によって、マネージャノードの稼働状況を監視します。同時にノードリストの整合性を確保します。
 - `name_audit_conf` オペランドに 1、または、2 を指定したとき
送受信型ノード監視機能によって、指定値を無視してマネージャノードの稼働状況を監視します。同時にノードリストの整合性を確保します。エージェントノード、およびノーマルノードとのノード監視は指定値に従って監視します。
- 同一ホストで複数のマネージャノードを動作させる場合は、一つのホストに複数の異なる IP アドレスを設定し、それぞれのマネージャノードに異なる IP アドレスを設定してください。

●`name_audit_interval`=監視間隔時間

～〈符号なし整数〉((1～65535))《60》(単位：秒)

監視サービスのノード監視が終了したあと、次の監視を開始するまでの時間を指定します。

ノード自動追加機能を使用する場合の動作は次のとおりです。

- マネージャノードの場合、このオペランドの指定間隔でエージェントノードおよびノーマルノードの稼働状況を監視します。
- エージェントノードの場合、このオペランドの指定間隔で、次の動作をします。
 - マネージャノードの稼働状況を監視します。
 - 1.と同時にマネージャノードに対してノードリスト要求を行い、マネージャノードとの間でノードリストの整合性を取ります。
 - ほかのエージェントノードおよびノーマルノードの稼働状況の監視は、`name_audit_conf` オペランドの指定に従って行います。

このオペランドには、マネージャノードのネームサービス定義の `name_sync_ready_time` オペランドの指定値と同じ値を指定してください。指定値が異なると、エージェントノードのノードリストから不当にノード情報が削除されることがあります。また、マネージャノードとエージェントノードとの間で、ノードリストの整合性が取られるまでに時間が掛かるおそれがあります。

●`name_audit_watch_time`=ノードダウンを検知するまでの最大待ち時間

～〈符号なし整数〉((8～65535))《8》(単位：秒)

name_audit_conf オペランドに 2（送受信型ノード監視）を指定した場合、監視対象のノードから応答が返るまでの待ち時間の最大値を指定します。なお、name_audit_conf オペランドに 1 を指定した場合、システム共通定義の ipc_conn_interval オペランドに指定した値がノードダウンを検知するまでの時間となります。

ノード自動追加機能を使用する場合は、次の動作をします。

- マネジャノードでは、監視対象のノードから応答が返るまでの待ち時間の最大値となります。
- エージェントノードでは、次の動作をします。
 1. マネジャノードを認識するまでは、name_manager_node オペランドに指定したノードから応答が返るまでの応答待ち時間の最大値となります。
 2. マネジャノード認識後は、監視対象のノードから応答が返るまでの待ち時間の最大値となります。

●name_rpc_control_list=Y|N

～《Y》

RPC 抑止リストに登録されたノードの監視機能を使用するかどうかを指定します。

Y

RPC 抑止リストに登録されたノードの監視機能を使用します。監視する時間の間隔は 180 秒です。

N

RPC 抑止リストに登録されたノードの監視機能を使用しません。

ネームサービスでは、ノード監視機能とは別に 180 秒ごとに RPC 抑止リストに登録されたノードの起動状況を確認できます。この機能の使用有無を、このオペランドで指定します。

ノードの監視機能の設定を考慮して、この機能を使用してください。例えば、次に示す状態の場合、RPC 抑止リストに登録されたノードの監視機能を無効にする必要があります。

- name_audit_interval オペランドに指定した時間が経過していても、障害から復旧したノードが RPC 抑止リストから削除されることがあります。この場合、KFCA00651-I メッセージは出力されません。
- name_audit_conf オペランドに 2 を指定している場合、KFCA00650-I メッセージが定期的に出力されることがあります。

RPC 抑止リストに登録されたノードの監視機能を無効にした場合で、name_audit_interval オペランドに指定した時間が 180 秒以上のとき、障害から復旧したノードが RPC 抑止リストから削除されるまでの時間は、RPC 抑止リストに登録されたノードの監視機能を使用した場合よりも長くなります。

次に示す条件を満たす場合は、このオペランドに N を指定することをお勧めします。

- name_audit_conf オペランドに 1 または 2 を指定
- name_audit_interval オペランドに 180 以下を指定

name_audit_conf オペランドを省略するか、または 0 を指定した場合、name_rpc_control_list オペランドに N を指定したときは、ノード監視機能および RPC 抑止リストへの監視機能は無効になるので次に示す状態になります。

- RPC 抑止リストに登録したノードから自ノードに対して通信が発生しないかぎり、RPC 抑止リストから該当ノードは削除されません。
- RPC 抑止リストに登録したノードの all_node オペランドに自ノードを指定していない場合、自ノードの OpenTP1 を再起動しないかぎり、RPC 抑止リストから該当ノードが削除されません。

上記の状態を防ぐためには、name_audit_conf オペランドを省略するか、または 0 を指定した場合、name_rpc_control_list オペランドを省略するか、Y を指定することをお勧めします。

●name_nodeid_check_message=Y|N

～《Y》

自ノードと同じノード識別子を指定したノードから通信を受信した場合に、KFCA00677-W メッセージを出力するかどうかを指定します。

Y

KFCA00677-W メッセージを出力します。

N

KFCA00677-W メッセージを出力しません。

●name_cache_validity_time=他ノードのサービス情報の有効時間

～〈符号なし整数〉((0～65535))《1800》(単位：秒)

他ノードから取得したサーバ UAP のサービス情報の有効時間を指定します。ネームサービスでは、次に示す場合に、他ノードで起動しているサーバ UAP のサービス情報を取得します。

- RPC 先となるサービスグループに対して、そのノードで初めて RPC を実行した場合
- 他ノードでサーバ UAP が起動された場合

他ノードからサービス情報を取得した時点からこのオペランドに指定した時間内は、他ノードに対してサービス情報を検索しないで RPC を実行します。RPC 実行時に、サービス情報の有効時間がこのオペランドに指定した時間を超過していた場合、システム共通定義の all_node オペランドに指定したノードからサービス情報を新たに取得し、すでに登録されているサービス情報を更新します。また、サービス情報の有効時間内でも、他ノードからサービス情報の更新通知を受けると、サービス情報は更新されます。

name_cache_validity_time オペランドの指定値を変更する場合の注意事項を次に示します。

- 0 を指定した場合
サービス情報の有効時間満了時でもサービス情報を取得しません。
- デフォルト値より小さい値 (1～1799) を指定した場合

サービス情報の有効時間が短くなるため、RPC 実行時に他ノードのネームサービスとのサービス情報のやり取りが多くなり、他ノードとの通信トラフィックが高くなります。この場合、グローバルキャッシュを更新する間隔を短くすることで、システム共通定義の `all_node` オペランドで指定されたノードのサービス情報変化の検知が早まります。

極端に小さい値を指定した場合、各ノードのネームサービスの負荷が上がり、サービス情報の検索などネームサービスの機能が利用できなくなることがあります。

- デフォルト値より大きい値（1801～65535）を指定した場合

サービス情報の有効時間が長くなるため、RPC 実行時に他ノードのネームサービスとのサービス情報のやり取りが少なくなり、他ノードとの通信トラフィックが低くなります。他ノードで起動しているサーバ UAP のサービス情報を取得する契機が少なくなるため、任意のノードで起動しているサーバ UAP へ RPC が実行されないなどの現象が発生することがあります。

- ノード自動追加機能の使用時に、このオペランドに小さな値を指定すると、ノードリストの参照が頻繁に動作します。このとき、ノードリストからの削除が実行されないことが多くなります。

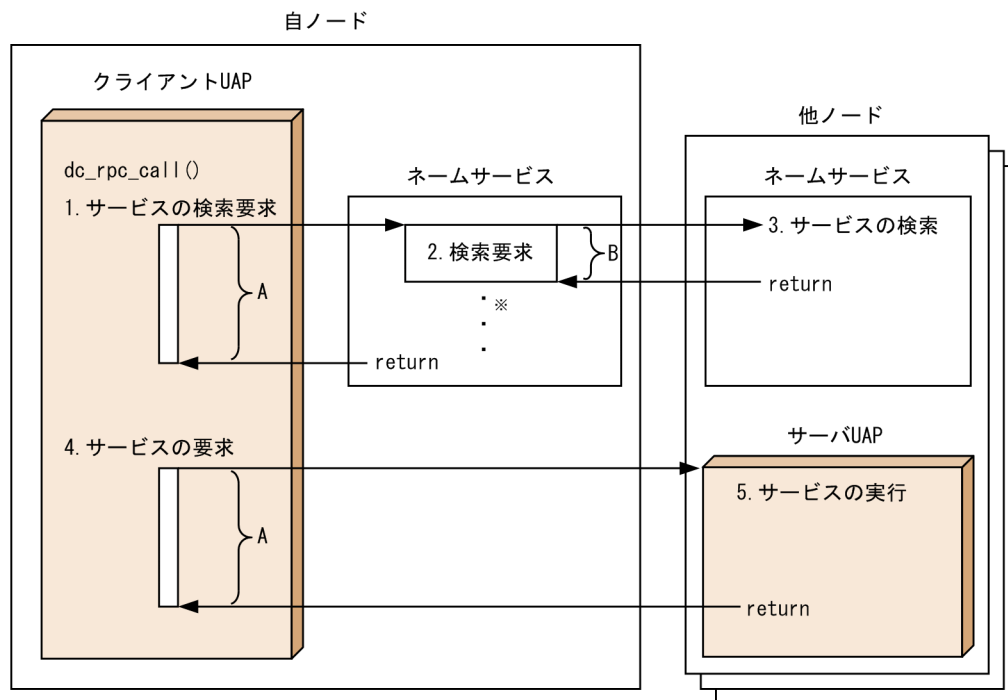
●`watch_time`=最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

クライアント UAP が他ノードのサーバ UAP にサービスを要求する形態の場合、このオペランドを指定してください。他ノードのサーバ UAP へサービスを要求する概要と、このオペランドが適用される RPC 通信個所を次の図に示します。

図 3-5 他ノードのサーバUAP へのサービス要求概要



(凡例) A: クライアントUAPに指定した最大応答待ち時間 (最大応答待ち時間A)
 クライアントUAPに指定した最大応答待ち時間は、ユーザサービス定義のwatch_timeオペランド、またはdc_rpc_set_watch_time関数で設定した時間になります。
 B: ネームサービス定義のwatch_timeオペランドで指定した値 (最大応答待ち時間B)

注※ 複数のノードに散在する同じサービスグループ名のサーバUAPにサービスを要求する場合は、そのノードの数分、検索要求があります。

1. クライアントUAPは、同じノードのネームサービスにサーバUAPのサービスの検索要求を送信します。この応答が返るまでの最大応答待ち時間Aは、クライアントUAPに指定した最大応答待ち時間になります。ただし、すべてのノードへのサービスの検索要求が送受信タイムアウトになった場合、検索要求をリトライします。リトライ回数、およびリトライ間隔は、システム共通定義のrpc_retry_countオペランド、およびrpc_retry_intervalオペランドに指定した値です。この二つのオペランドは、システム共通定義のrpc_retryオペランドにYを指定した場合に有効になります。rpc_retryオペランドにNを指定した場合、リトライ回数は1になります。
2. ネームサービスは、1. で受信したサービスの検索要求を他ノードのネームサービスに送信します。この応答が返るまでの最大応答待ち時間Bが、ネームサービス定義のwatch_timeオペランドに指定した値になります。もし、複数のノードのネームサービスにサービスの検索要求を送信する場合、それぞれのノードへの検索要求に最大応答待ち時間Bが適用されます。
3. サーバUAPのサービスを検索します。
4. クライアントUAPは、サーバUAPにサービス要求を送信します。この応答が返るまでの最大応答待ち時間Aは、クライアントUAPに指定した最大応答待ち時間になります。
5. サーバUAPのサービスを実行します。

このオペランドには、クライアント UAP の最大応答待ち時間よりも小さい値を指定してください。クライアント UAP の最大応答待ち時間と同じか、それよりも大きい値を指定した場合、他ノードのネームサービスの処理遅延によって、先にクライアント UAP プロセスのサービスの検索要求が送受信タイムアウトになります。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かるときがあります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しないときがあります。

省略した場合は、システム共通定義の watch_time オペランドの値を仮定します。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

●name_start_watch_time=OpenTP1 起動時のノードリスト要求応答待ち時間

～ 〈符号なし整数〉 ((0～1800)) 《8》 (単位：秒)

エージェントノードは OpenTP1 の起動時に、マネージャノードに対してノードリスト要求をします。ノードリスト要求を送信してから、応答を受信するまでの待ち時間の最大値を指定します。指定時間を過ぎても応答がない場合は、ノードリスト要求はエラーとなります。エラー発生後の動作は、name_start_error オペランドの指定値に従います。

このオペランドは、ノード自動追加機能を使用する場合に、エージェントノードで指定します。

0 を指定した場合、応答を受信するまで無限に待ち続けます。

大きな値を指定した場合、マネージャノードの動作状況によっては、OpenTP1 の開始処理に時間が掛かるおそれがあります。

●name_start_retry_count=OpenTP1 起動時のノードリスト要求リトライ回数

～ 〈符号なし整数〉 ((0～60)) 《0》

エージェントノードは OpenTP1 の起動時に、マネージャノードに対してノードリスト要求をします。このノードリスト要求でエラーが発生した場合のリトライ回数を指定します。発生したエラーの回数が、指定した値を超えたときは、name_start_error オペランドの指定に従った処理を実行します。

このオペランドは、ノード自動追加機能を使用する場合に、エージェントノードで指定します。

0 を指定した場合、リトライしません。

大きな値を指定した場合、マネージャノードの動作状況によっては、OpenTP1 の開始処理に時間が掛かるおそれがあります。

●name_start_retry_interval=OpenTP1 起動時のノードリスト要求リトライ間隔

～ 〈符号なし整数〉 ((1～60)) 《10》 (単位：秒)

エージェントノードは OpenTP1 の起動時に、マネージャノードに対するノードリスト要求をします。このノードリスト要求でエラーが発生した場合のリトライ間隔を指定します。

このオペランドは、ノード自動追加機能を使用する場合に、エージェントノードで指定します。また、ネームサービス定義の name_start_retry_count オペランドに 1 以上を指定した場合に有効となります。

大きな値を指定した場合、マネージャノードの動作状況によっては、OpenTP1 の開始処理に時間が掛かるおそれがあります。

●name_start_error=stop|continue

～《continue》

エージェントノードが OpenTP1 の起動時に、マネージャノードからノードリストを正常に受信できなかった場合の処理を指定します。

stop

OpenTP1 の開始処理を中止します。

このとき、アボートコード namdmnU を出力します。ただし、再開始（リラン）時は、OpenTP1 の開始処理を続行します。

continue

OpenTP1 の開始処理を続行します。

OpenTP1 開始完了後、ネームサービス定義の name_audit_interval オペランドに指定した間隔で、定期的にノードリスト要求を送信します。このときの応答待ち時間は、ネームサービス定義の name_audit_watch_time オペランドに指定した値となります。マネージャノードからノードリストを受信するまでの間は、オペランドを指定したノードだけで動作します。

このオペランドは、ノード自動追加機能を使用する場合に、エージェントノードで指定します。

●name_sync_ready_time=ノードリスト同期待ち時間

～〈符号なし整数〉((0～65535))《60》(単位：秒)

エージェントノードから RPC メッセージが受信できる状態になってから、エージェントノードとノードリストの整合性を取るまでの待ち時間を指定します。

このオペランドは、ノード自動追加機能を使用する場合に、マネージャノードで指定します。ただし、ノードリストを引き継ぐ場合、このオペランドの指定は無視します。

マネージャノードは、このオペランドに指定した時間が経過するまで、エージェントノードとノードリストの整合性を取りません。ただし、OpenTP1 システム内に新たに追加されたエージェントノードに対してはノードリストを返します。

このオペランドには、エージェントノードのネームサービス定義の name_audit_interval オペランド指定値と同じ値を指定してください。指定値が異なると、エージェントノードのノードリストから不当にノード情報が削除されることがあります。また、マネージャノードとエージェントノードとの間で、ノードリストの整合性が取られるまでに時間が掛かるおそれがあります。

コマンド形式

次ページ以降に記述しています。

名称

ノードリストファイルの指定

形式

`[namnlfil - {r|n} ノードリストファイル名]`

機能

ノード自動追加機能を使用する場合に、ノードリストの引き継ぎ機能で使用するノードリストファイルを指定します。

OpenTP1 の起動時に、ネームサービスは指定されたノードリストファイルに登録されているノードリストを採用してサービスを開始します。OpenTP1 のオンライン中は、一定間隔で使用中のノードリストをノードリストファイルへ書き込みます。

なお、次の場合は、ノードリストの引き継ぎ機能は無効として、OpenTP1 を起動します。

- この定義を省略した場合
- この定義に指定したノードリストファイルがない場合
- この定義に指定したノードリストファイルへのアクセスに失敗した場合

この定義コマンドは、システム共通定義の `name_service_mode` オペランドの指定が `manager` または `agent` の場合だけ有効となります。

オプション

●-r

再開始時だけノードリストファイルを読み込みます。ただし、システム共通定義の `name_service_mode` オペランドの指定が `agent` の場合は、再開始時にマネージャノードからのノードリストの取得に失敗したときだけ、ノードリストファイルを読み込みます。

●-n

再開始時および（強制）正常開始時にノードリストファイルを読み込みます。ただし、システム共通定義の `name_service_mode` オペランドの指定が `agent` の場合は、再開始時および（強制）正常開始時にマネージャノードからのノードリストの取得に失敗したときだけ、ノードリストファイルを読み込みます。

コマンド引数

●ノードリストファイル名

～ 〈1～63 文字のパス名〉

ノードリストを格納するノードリストファイルのファイル名を完全パスで指定します。

プロセスサービス定義

形式

set 形式

```
[set prc_process_count=最大同時起動サーバプロセス数]
[set prc_recovery_resident=Y|N]
[set prc_take_over_svpath=Y|N]
[set prc_take_over_dlpath=Y|N]
[set term_watch_time=連続異常終了限界経過時間]
[set max_socket_descriptors=ソケット用ファイル記述子の最大数]
[set term_watch_count=連続異常終了限界回数]
[set prc_prf_trace=Y|N]
[set prc_coresave_path=コアファイル格納パス]
[set prc_corecompress=Y|N]
[set watch_time=最大応答待ち時間]
```

コマンド形式

```
[prcsvpath パス名]
```

機能

OpenTP1 システム環境下で実行するサーバプロセスを管理するための実行環境を定義します。

プロセスサービス定義を変更した場合、OpenTP1 を正常終了して dcreset コマンドを実行してください。

説明

set 形式のオペランド

●prc_process_count=最大同時起動サーバプロセス数

～ 〈符号なし整数〉 ((1～10000)) 《64》

同一 OpenTP1 ノード内で同時に起動するサーバプロセス数の最大値を指定します。ただし、実際に起動できる最大プロセス数は、OpenTP1 を起動しているマシンに依存します。

なお、このオペランドを指定する場合は、実装するメモリ量やマシンで起動できる最大プロセスを考慮して、指定してください。

ここで指定するプロセス数は、OpenTP1 システムで動作する常駐、および非常駐の全サーバ（ユーザサーバ、およびシステムサービス）プロセス数を指定します。プロセス数には、次のコマンドが含まれます。

- dcstart
- dcstop
- dcsvstart

- dcsvstop
- dcstats

なお、ユーザサービス構成定義の dcsvstart 定義コマンドは含まれません。

マルチサーバの場合は、起動する本数を加算します。

非常駐プロセスを指定しているユーザサーバや scdrsprc コマンドを使用すると、一時的に最大でユーザサービス定義の parallel_count オペランドの指定値の倍のプロセス数（常駐プロセス数と最大プロセス数を指定している場合は最大プロセス数の倍、常駐プロセス数だけを指定している場合は常駐プロセス数の倍）が起動されることがあります。

非常駐プロセスを指定しているユーザサーバ、および scdrsprc コマンドで、倍のプロセス数が起動されるタイミングを次に示します。

非常駐プロセスを指定しているユーザサーバの場合

終了しようとしているプロセスが、dc_rpc_mainloop 関数または dc_mcf_mainloop 関数の終了後から dc_rpc_close 関数終了までの区間にある場合で、新たなサービス要求を受け付けたタイミング

scdrsprc コマンドで再起動するユーザサーバの場合

終了しようとしているプロセスが、dc_rpc_mainloop 関数または dc_mcf_mainloop 関数の終了後から dc_rpc_close 関数終了までの区間にある場合で、新たなプロセス起動を行ったタイミング

新たなプロセスが起動する契機については、マニュアル「OpenTP1 運用と操作」の scdrsprc コマンドの説明を参照してください。

システムサービスの場合は、ログサービス、ジャーナル関連（チェックポイントダンプ、システムジャーナルなど）、MCF、および MQ システムの数を指定します。

●prc_recovery_resident=Y|N

～《Y》

UAP が異常終了したときに部分回復するプロセスを常駐にするか、非常駐にするかを指定します。非常駐にすると、UAP がダウンするたびに部分回復するプロセスが起動されます。OpenTP1 を動作させるときにメモリに余裕がない場合は、Y を指定、またはこのオペランドを省略してください。Y を指定、またはこのオペランドを省略した場合は、部分回復する常駐プロセスの数は一つです。

Y

部分回復するプロセスを常駐化します。

N

部分回復するプロセスを非常駐化します。

●prc_take_over_svpath=Y|N

～《N》

ノード内で、ユーザサーバおよびコマンドのパスをリラン時に引き継ぐかどうかを指定します。

Y

リラン時にユーザサーバおよびコマンドのパスを引き継ぎます。

N

リラン時にユーザサーバおよびコマンドのパスを引き継ぎません。

Y を指定すると、ユーザサーバおよびコマンドのパスを変更したあと、OpenTP1 が正常開始するまでその変更が引き継がれます。

ユーザサーバおよびコマンドのパスの引き継ぎは、ノード内に限定されるので、系切り替え機能で切り替わった系には引き継がれません。

ユーザサーバおよびコマンドのパスの引き継ぎに失敗した場合は、警告メッセージを出力し、前回のオンライン中のユーザサーバおよびコマンドのパスの変更を引き継がないで処理を続行します。

●prc_take_over_dlpath=Y|N

～《N》

ノード内で、prcdlpath コマンドで設定した UAP 共用ライブラリのサーチパスをリラン時に引き継ぐかどうかを指定します。

Y

リラン時に UAP 共用ライブラリのサーチパスを引き継ぎます。

N

リラン時に UAP 共用ライブラリのサーチパスを引き継ぎません。

Y を指定すると、prcdlpath コマンドで UAP 共用ライブラリのサーチパスを変更したあと、OpenTP1 が正常開始するまでその変更が引き継がれます。prcdlpath コマンドを複数回実行した場合は、最後に実行した prcdlpath コマンドで設定されたサーチパスだけが引き継がれます。

UAP 共用ライブラリのサーチパスを変更したあと、OpenTP1 を正常開始した場合は、その変更は引き継がれません。そのため、正常開始後に変更後のサーチパスに指定したディレクトリから UAP 共用ライブラリを読み込みたい場合は、環境変数でライブラリサーチパスを設定してください。

UAP 共用ライブラリのサーチパスの引き継ぎは、ノード内に限定されるので、系切り替え機能で切り替わった系には引き継がれません。

UAP 共用ライブラリのサーチパスの引き継ぎに失敗した場合は、KFCA00734-W メッセージを出力し、前回のオンライン中に行った UAP 共用ライブラリのパスの変更を引き継がないで処理を実行します。

前回のオンライン中に、何度も prcdlpath コマンドでサーチパスを変更しながらユーザサーバを開始していた場合、サーチパスは引き継がれていても、リラン時にユーザサーバが正常に開始できないことがあります。

●term_watch_time=連続異常終了限界経過時間

～〈符号なし整数〉((0～32767))《10》(単位：分)

OpenTP1 が異常終了した回数を監視する限界経過時間を指定します。0 を指定した場合は、時間監視をしません。

指定時間内に 3 回連続して OpenTP1 がシステムダウンした場合、プロセスサービスは KFCA00715-E メッセージを出力して、システムの開始または再開を中断します。この場合は、システムダウンをした要因を取り除いてから、dcsetup コマンドで一度 OS への登録を削除、再度登録するか、または dcreset コマンドを入力してください。

システム環境定義の mode_conf オペランドが「MANUAL2」の場合、プロセスサービスの連続ダウン以外はこのオペランドの指定は無効となります。

●max_socket_descriptors=ソケット用ファイル記述子の最大数

～〈符号なし整数〉((32～2032))

OpenTP1 制御下のプロセス※でのソケット用に使用するファイル記述子の最大数を指定します。

OpenTP1 制御下のプロセス※は、システムサーバやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数、および通信する他ノードの数によって、このオペランドを変更する必要があります。

注※

MCF サービス (MCF マネジャサービス, MCF 通信サービス, およびアプリケーション起動サービス) 以外の OpenTP1 プロセスが対象です。MCF サービスの場合は、「[システムサービス情報定義](#)」および「[システムサービス共通情報定義](#)」を参照してください。

ソケット用ファイル記述子の最大数の算出式を、次に示します。

$\uparrow (\text{自ノード内 UAP プロセス数}^{\ast 1} + \text{プロセスサービスに対してサービスを要求してくるノード数}^{\ast 2} + \text{システムサービスプロセス数}^{\ast 3}) / 0.8 \uparrow$

↑↑：小数点以下を切り上げます。

注※1

自ノード内 UAP プロセス数は、次に示す値の合計です。

- 自 OpenTP1 内の UAP プロセス数
- CUP から同時に起動するトランザクション数 (クライアントサービス定義の parallel_count オペランドの指定値)

注※2

プロセスサービスに対してサービスを要求してくるプロセス数とは、TP1/Multi を使用した場合だけ加算してください。このノード数は、次に示す値の合計です。

- 自ノードに対して dc_adm_get～関数を呼び出す、他ノード内 UAP プロセス数
- 自ノードを構成要素とするマルチノードエリアおよびマルチノードサブエリアに対して入力する dcmstart, dcmstop, dcndls コマンドの数

注※3

システムサービスプロセス数とは、自 OpenTP1 内のシステムサービスプロセスの数です。自 OpenTP1 内のシステムサービスプロセスは、rpcstat コマンドで表示されるサーバ名をカウントすることで求められます。rpcstat コマンドで表示されるサーバ名のうち、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造に記載されているシステムサービスプロセスをカウントしてください。

このオペランドの指定が小さいと、OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため、プロセスが KFCA00307-E メッセージを出力して異常終了します。

指定値の優先順位は次のとおりです (1.> 2.)。

1. プロセスサービス定義
2. システム共通定義

ここで指定を省略した場合、システム共通定義の値を仮定します。

●term_watch_count=連続異常終了限界回数

～〈符号なし整数〉((1～3))《3》

このオペランドに 1～3 の値を指定すると、OpenTP1 システム連続異常終了限界回数指定機能を使用できます。

term_watch_time オペランドは、このオペランドの指定値が 3、または省略された場合に有効となります。このオペランドと term_watch_time オペランドの関係を次の表に示します。

表 3-2 term_watch_count オペランドと term_watch_time オペランドの関係

term_watch_count オペランドの指定値	term_watch_time オペランドの指定値	
	0	0 以外
1 または 2	term_watch_time オペランドの指定値に関係なく、term_watch_count オペランドの指定値で開始、再開を中断します。	term_watch_time オペランドの指定値に関係なく、term_watch_count オペランドの指定値で開始、再開を中断します。
3 または省略	開始、再開を中断しません。	term_watch_time オペランドの指定値に従い、3 回で開始、再開を中断します。

このオペランドで指定した回数まで連続して OpenTP1 がシステムダウンした場合、プロセスサービスは KFCA00715-E メッセージを出力して、システムの開始または再開を中断します。この場合は、システムダウンした要因を取り除いてから、dcsetup コマンドで一度 OS への登録を削除して再登録するか、または dcreset コマンドを入力してください。

システム環境定義の mode_conf オペランドが「MANUAL2」の場合、プロセスサービスの連続ダウン以外はこのオペランドの指定は無効となります。

●prc_prf_trace=Y|N

～《Y》

プロセスサービスのイベントトレースを取得するかどうかを指定します。

Y

プロセスサービスのイベントトレースを取得します。

N

プロセスサービスのイベントトレースを取得しません。

このオペランドで Y を指定した場合、プロセスサービスの動作を検証するためのイベントトレースが取得されます。イベント ID の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

イベントトレースは次に示すファイルに取得されます。

- 0xb001～0xb003：\$DCDIR/spool/dcprfinf の下のトレースファイル
- 0xb010 以降：\$DCDIR/spool/dcprcinf の下のトレースファイル

イベントトレースのうち、プロセスの生成と消滅については次に示すタイミングで情報が取得されます。

1. プロセス生成時に子プロセスで取得（イベント ID は 0xb001）
2. プロセス消滅時に取得（イベント ID は 0xb002）
3. プロセス生成時に親プロセスで取得（イベント ID は 0xb003）

prfed コマンド実行時に -d オプションを指定すると、1.では生成したプロセスのサーバ名を、2.では消滅した ID とその終了状態を、また 3.では生成したプロセス ID を、表示されるデータとして取得します。したがって、取得したトレース情報を prfed コマンドで出力する場合は、-d オプションを指定してください。-d オプションを指定すると次のようにトレース情報が出力されます。

1.の場合

Offset	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+a	+b	+c	+d	+e	+f	ASCII_code
+0000	aa	aa	aa	aa	aa	aa	aa	aa	aa								AAAAAAA

- aa：生成したプロセスのサーバ名（16 進数字）
- AA：生成したプロセスのサーバ名（ASCII コード）

2.の場合

Offset	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+a	+b	+c	+d	+e	+f	ASCII_code
+0000	bb	bb	bb	bb	cc	cc	cc	cc	cc								BBBCCCC

- bb：消滅したプロセス ID（16 進数字）

- BB：消滅したプロセス ID (ASCII コード)
- cc：消滅したプロセスの終了状態 (16 進数字)
- CC：消滅したプロセスの終了状態 (ASCII コード)

3.の場合

Offset	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+a	+b	+c	+d	+e	+f	ASCII_code
+0000	dd	dd	dd	dd													DDDD

- dd：生成したプロセス ID (16 進数字)
- DD：生成したプロセス ID (ASCII コード)

prfed コマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

なお、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、このオペランドの指定は無効となります。

●prc_coresave_path=コアファイル格納パス

～ 〈パス名〉《\$DCDIR/spool/save》

OpenTP1 プロセスのコアファイルを格納するディレクトリを絶対パス名で指定します。

ユーザサーバの場合は、UAP トレース編集結果もこのオペランドで指定したディレクトリに格納されます。

パス名に指定できる文字数は、最大 63 文字です。

このオペランドで指定したパス名に誤りがある場合や、このオペランドの指定を省略した場合は、コアファイルの格納先に \$DCDIR/spool/save を仮定します。

OpenTP1 管理者は、このオペランドで指定したディレクトリにコアファイルを書き込める権限を持っている必要があります。そうでない場合は、コアファイルが書き込まれないことがあります。

プロセスサーバのコアファイルの場合は、このオペランドの指定に関係なく、\$DCDIR/spool/save に出力されることがあります。

このオペランドの指定値を変更した場合は、dcreset コマンドを実行する必要があります。

●prc_corecompress=Y|N

～ 〈N〉

OpenTP1 プロセスのコアファイルの格納時に OpenTP1 で自動的に圧縮するかどうかを指定します。

Y

OpenTP1 プロセスのコアファイルの格納時に、OpenTP1 で自動的に圧縮します。

N

OpenTP1 プロセスのコアファイルの格納時に、OpenTP1 で自動的に圧縮しません。

このオペランドの指定を省略した場合は、圧縮しません。

圧縮処理は、OS が標準提供するコマンドを使用して、OpenTP1 がオンライン中に UAP がダウンした場合などに行います。そのため、このオペランドで Y を指定した場合で、UAP がダウンしたときは、OS が標準提供する圧縮処理用コマンドの性能に依存して、全体の性能が低下することがあります。

このオペランドで Y を指定した場合、コアファイルを格納するディレクトリに、圧縮処理前のファイル名が付いた 0 バイトのファイルが作成されます。このファイルは、退避した情報ファイルの世代管理に使用することがあるため、削除しないでください。

このオペランドの指定値を変更した場合は、dcreset コマンドを実行する必要があります。

●watch_time=最大応答待ち時間

～〈符号なし整数〉((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送ってからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かるときがあります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しないときがあります。

省略した場合は、システム共通定義の watch_time オペランドの値を仮定します。

このオペランドは、システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

次ページ以降に記述しています。

prcsvpath

名称

ユーザサーバのパスの指定

形式

[prcsvpath パス名]

機能

ユーザサーバおよびコマンドのパス名を指定します。

オプション

●パス名

～ 〈パス名〉《\$DCDIR/aplib:\$DCDIR/bin》

プロセスサービスが起動する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絶対パス名を指定します。

ユーザサーバを入れ替えるときに、ユーザサーバが格納されているディレクトリのパス名をオンライン中に prcpath コマンドで変更できます。prcpath コマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

パス名は、複数指定できます。複数のパス名を指定する場合は、コロン':'で区切って指定してください。また、追加する場合は、既存のパスを含めて指定してください。なお、パス名には環境変数は指定できません。

パス名に指定できる文字数は、最大 255 文字です。ただし、定義の 1 行の長さは最大 80 バイトです。80 バイトを超える場合については、「[1.4 定義の規則](#)」を参照してください。

スケジュールサービス定義

形式

set 形式

```
[set scd_server_count=最大ユーザサーバ数]
[set scd_hold_recovery=Y|F]
[set scd_hold_recovery_count=閉塞状態の引き継ぎが必要なサーバ、および
                             サービスの合計数]
[set scd_port=スケジュールサービスのポート番号]
[set scd_this_node_first=Y|N]
[set scd_announce_server_status=Y|N]
[set max_socket_descriptors=ソケット用ファイル記述子の最大数]
[set max_open_fds=スケジュールサービスプロセスでアクセスするファイル、およびパイプの最大
数]
[set schedule_rate=スケジュール比率]
[set scd_retry_of_comm_error=リトライ回数]
[set scd_advertise_control=BEFORE|AFTER]
[set scd_message_level=1|2]
[set ipc_tcpnodelay=Y|N]
[set watch_time=最大応答待ち時間]
```

コマンド形式

```
{{scdbufgrp -g スケジュールバッファグループ名
               { [-e メッセージ格納バッファプール長] |
                 [-n メッセージ格納バッファセル数] }
               { [-l メッセージ格納バッファセル長]
                 { [-s メッセージ格納バッファ使用制限サイズ] |
                   [-p メッセージ格納バッファ使用制限率] }}}
{{scdmulti [-m マルチスケジューラデーモン数]
            [-p ポート番号]
            [-g マルチスケジューラグループ名]
            [-t] }}
```

機能

OpenTP1 システム環境下で動作するサーバへのサービス要求のスケジューリングをするための実行環境を定義します。

スケジュールサービスは OS のメッセージ機能を使用して、スケジュールキューをサービスグループ単位に作成します。

説明

set 形式のオペランド

●scd_server_count=最大ユーザサーバ数

～ 〈符号なし整数〉 ((0～8192)) 《32》

スケジューラサービスを利用して動作するユーザサーバ数の最大値を指定します。

オンライン時に、サーバ起動コマンド（dcsvstart）で起動するユーザサーバの数も含めて指定します。

ユーザサーバ数に含めるサーバはキュー受信型 SPP，MHP，rap サーバ，RTSSPP，mqrspp です。

このオペランドを変更する場合は「7.3.1 ユーザサーバの追加」を参照し、定義の見直しや OpenTP1 ファイルの再見積もりの必要有無を確認してください。

●scd_hold_recovery=Y|F

～《Y》

システムを全面回復する場合に、ユーザサーバの閉塞状態を引き継ぐかどうかを指定します。

Y

ユーザサービス定義、ユーザサービスデフォルト定義の hold_recovery オペランドの指定によって、サーバまたはサービスの閉塞状態を引き継ぎます。ただし、システム環境定義の start_scheduling_timing オペランドに BEFORE を指定した場合は、ユーザサービス定義の hold_recovery オペランドの指定に関係なく、閉塞状態は引き継ぎません。

F

ユーザサービス定義、ユーザサービスデフォルト定義の hold_recovery オペランドの指定によって、サーバまたはサービスの閉塞状態を引き継ぎます。

閉塞状態のサーバまたはサービスの数がスケジューラサービス定義の scd_hold_recovery_count オペランドの指定値を超えた場合、超えた分のサーバまたはサービスの閉塞状態は引き継ぎません。また scd_hold_recovery_count オペランドに 0 を指定した場合も引き継ぎません。

SPP のソケット受信型サーバ，SUP，および MHP の場合は、指定する必要はありません。

scd_hold_recovery オペランドの指定と、ユーザサービス定義，ユーザサービスデフォルト定義の hold_recovery オペランド，システム環境定義の start_scheduling_timing オペランド，スケジューラサービス定義の scd_hold_recovery_count オペランドの指定の関係を、次の表に示します。

各オペランドの指定値				OpenTP1 の閉塞引き継ぎ 処理
start_scheduling_t iming	hold_recovery	scd_hold_recovery _count	scd_hold_recovery	
AFTER	Y	0	—	×
		0 以外	—	○
	N	0	—	×
		0 以外	—	×
BEFORE	Y	0	—	×
		0 以外	Y	×

各オペランドの指定値				OpenTP1 の閉塞引き継ぎ 処理
start_scheduling_t iming	hold_recovery	scd_hold_recovery _count	scd_hold_recovery	
BEFORE	Y	0 以外	F	○
	N	0	—	×
		0 以外	—	×

(凡例)

- ：閉塞状態を引き継ぎます。
- ×
- ：指定できません。指定しても無視されます。

●scd_hold_recovery_count=閉塞状態の引き継ぎが必要なサーバ、およびサービスの合計数

～〈符号なし整数〉((0～58240))《64》

システムを全面回復するときに、閉塞状態を引き継ぐことが必要なサーバ、およびサービスの合計数を指定します（指定値は、偶数になるように切り上げられます）。

ユーザサービス定義で service_hold オペランドに N が指定されたサーバは、一つのサーバとして数えます。service_hold オペランドに Y が指定されたサーバは、そのサーバが持つサービス数を数えて合計数を指定してください。ただし、service_hold オペランドに Y が指定されていても、ユーザサービス定義で hold オペランドに N が指定され、かつ service_term_watch_time オペランドに 0 が指定された場合は、一つのサーバとして数えます。

閉塞状態のサーバ、またはサービスの数がこの指定値を超えた場合、超えた分のサーバ、またはサービスの閉塞状態は引き継がれません。0 を指定した場合も引き継がれません。

システムを全面回復するとき、前回のオンライン処理のときに指定した値が引き継がれます。ただし、前回、または今回の指定値が 0 の場合は、指定値は引き継がれません。

0 以外を指定した場合は、閉塞情報がステータスファイルに格納されるので、指定値に応じてステータスファイルの容量を見積もる必要があります。指定値が小さいと、ステータスファイル内で閉塞情報の書き換え処理が多く発生します。

TP1/Server Base のバージョンが 07-07 以前の場合は、このオペランドの指定値によって、システム共通定義の rpc_max_message_size オペランドの指定値を変更する必要がある場合があります。次の計算式の算出値が 1 より大きい場合（2～8 の場合）は、算出値以上の値を rpc_max_message_size オペランドに指定してください。

↑ scd_hold_recovery_count オペランドの指定値／7280 ↑

↑ ↑：小数点以下を切り上げます。

rpc_max_message_size オペランドの指定値を変更する場合は、rpc_max_message_size オペランドの説明を参照してください。

●scd_port=スケジュールサービスのポート番号

～ 〈符号なし整数〉 ((5001～65535))

スケジュールサービスのポート番号を指定します。

このオペランドの指定を省略した場合、システムが任意に割り当てたポート番号で起動します。

このオペランドで指定するポート番号は、ほかのプログラムと重複しないようにしてください。

また、OS には任意に割り当てる番号がありますが、この番号も使用しないでください。OS が任意に割り当てる番号は、OS の種別やバージョンによって異なります。詳細については、使用している OS のマニュアルを参照してください。

●scd_this_node_first=Y|N

～ 〈N〉

要求されたサーバが自ノードにある場合に、自ノードのサーバを優先してスケジュールするかどうかを指定します。

Y

要求されたサーバが自ノードにある場合には、自ノードのサーバを優先してスケジュールします。

サーバが自ノードにない場合や、スケジュールできない状態（過負荷状態や閉塞状態など）の場合には、他ノードのサーバをスケジュールします。

N

要求されたサーバが自ノードにある場合でも、自ノードのサーバを優先しないで、ランダムにノードを選択してスケジュールします。

このオペランドで Y を指定し、scd_announce_server_status オペランドで N を指定すると、自ノードのサーバがスケジュールできる状態なら、負荷状態に関係なく、必ず自ノードでスケジュールします。

●scd_announce_server_status=Y|N

～ 〈Y〉

自ノードのサーバの状態を、すべての他ノードに不定期で通知するかどうかを指定します。ノード間で負荷バランスを取る場合に指定します。

Y

自ノードのサーバの状態を、すべての他ノードに不定期（30 秒以上の任意の間隔）で通知します。

N

自ノードのサーバの状態を、他ノードに通知しません。

このオペランドで Y を指定すると、同一サービスグループが複数ノードにある場合に、サーバの状態をノード間で通知し合うことで、負荷を考慮したノード間の負荷バランスが取れます。ただし、次のような場合には、N を指定してください。

- サービスグループが一つのノードで起動している場合
- 負荷を考慮しないで、ランダム選択でノード間の負荷バランスを取る場合

なお、このオペランドの指定は、分散処理するすべてのノードで同じにしてください。指定が異なるノードがあると、負荷が集中することがあります。

●max_socket_descriptors=ソケット用ファイル記述子の最大数

～〈符号なし整数〉((32～2032))

OpenTP1 制御下のプロセス※でのソケット用に使用するファイル記述子の最大数を指定します。

OpenTP1 制御下のプロセス※は、システムサーバやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数、および通信する他ノードの数によって、このオペランドを変更する必要があります。

このオペランドには、次の条件を満たす値を指定してください。

(「このオペランドの指定値」+同定義内の「max_open_fds オペランドの指定値」) ≤ 2048

条件を満たさない値を指定した場合は、このオペランドの指定値は次に示すように強制的に補正されます。

(「このオペランドの指定値」+同定義内の「max_open_fds オペランドの指定値」) = 2048

注※

MCF サービス (MCF マネジャサービス, MCF 通信サービス, およびアプリケーション起動サービス) 以外の OpenTP1 プロセスが対象です。MCF サービスの場合は、「[システムサービス情報定義](#)」および「[システムサービス共通情報定義](#)」を参照してください。

ソケット用ファイル記述子の最大数の算出式を、次に示します。

$(\text{全 UAP プロセス数}^{\ast 1} + \text{スケジューラサービスを要求してくるノード数}^{\ast 2} + \text{システムサービスプロセス数}^{\ast 3}) / 0.8$

↑↑：小数点以下を切り上げます。

注※1

全 UAP プロセス数は、次に示す値の合計です。

- 自 OpenTP1 内の UAP プロセス数
- 自 OpenTP1 内のサービスを利用する、他ノード内の UAP プロセス数
- CUP から同時に起動するトランザクション数 (クライアントサービス定義の parallel_count オペランドの指定値)

注※2

スケジュールサービスを要求してくるノード数とは、次に示す値の合計です。ただし、重複する場合は一つ分だけ加算してください。

- 自 OpenTP1 の all_node オペランドに指定したノード名の数
- 自ノード名を、OpenTP1 の all_node オペランドに指定している、他ノードの数

注※3

システムサービスプロセス数とは、自 OpenTP1 内のシステムサービスプロセスの数です。自 OpenTP1 内のシステムサービスプロセスは、rpcstat コマンドで表示されるサーバ名をカウントすることで求められます。rpcstat コマンドで表示されるサーバ名のうち、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造に記載されているシステムサービスプロセスをカウントしてください。

このオペランドの指定が小さいと、OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため、プロセスが KFCA00307-E メッセージを出力して異常終了します。

指定値の優先順位は次のとおりです (1.> 2.)。

1. スケジュールサービス定義
2. システム共通定義

ここで指定を省略した場合、システム共通定義の値を仮定します。

●max_open_fds=スケジュールサービスプロセスでアクセスするファイル、およびパイプの最大数 ～ 〈符号なし整数〉 ((16～2016)) 《50》

スケジュールサービスプロセスで使用するファイルおよびパイプの最大数を指定します。

スケジュールサービスでアクセスするファイルおよびパイプの最大数の計算式を、次に示します。

ステータスサービス定義で指定した OpenTP1 ファイルシステム数 + 20 + N

(N: ユーザサービス定義の schedule_method に namedpipe を指定したユーザサーバ数)

このオペランドには、次の条件を満たす値を指定してください。

(「max_open_fds オペランドの指定値」 + 「max_socket_descriptors オペランドの指定値」) ≤ 2048

●schedule_rate=スケジュール比率

～ 〈符号なし整数〉 ((50～100)) (単位: %)

TP1/Client/W または TP1/Client/P のクライアント環境定義の DCSCDDIRECT オペランドに Y を指定して、RPC をスケジューリングする際、サーバの負荷レベルが LEVEL0、および LEVEL1 のノードのうち、LEVEL0 のノードへのスケジュール比率を指定します。

なお、この機能は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

●scd_retry_of_comm_error=リトライ回数

～〈符号なし整数〉((0～128))《0》

サービス要求のスケジューリング時に通信障害が発生したとき、障害ノード以外へスケジューリングするリトライ回数を指定します。ただし、このオペランドの指定値が、サービス要求の対象となるサービスグループが起動しているノード数を上回っている場合は、サービス要求の対象となるサービスグループが起動しているノード数をリトライ回数の上限値とします。

0 を指定した場合は、リトライしません。

なお、この機能は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

●scd_advertise_control=BEFORE|AFTER

～《BEFORE》

システム共通定義の all_node オペランドに指定されたノードに対し、ユーザサーバ起動時にネーム情報を通知するタイミングを変更します。

BEFORE

ユーザサーバの起動処理中（オンライン前）に、システム共通定義の all_node オペランドに指定された OpenTP1 ノードに対してネーム情報を通知します。

AFTER

ユーザサーバ起動完了直後に、システム共通定義の all_node オペランドに指定された OpenTP1 ノードに対してネーム情報を通知します。

また、OpenTP1 の開始処理、または再開処理で起動されるユーザサーバについては、スケジュールサービスが開始されるタイミングで起動されているユーザサーバのネーム情報をまとめて通知します。ただし、システム環境定義の start_scheduling_timing オペランドに BEFORE を指定すると、OpenTP1 の開始処理、または再開処理では、ユーザサーバが起動される前にスケジュールサービスが開始されるため、通知されるネーム情報はなくなります。

注

このオペランドに AFTER を指定した場合、OpenTP1 の開始処理、または再開処理で起動されるユーザサーバ数が多いと、システム共通定義の all_node オペランドに指定された OpenTP1 ノードに対してまとめて通知するネーム情報量が増え、マシンおよびネットワークの負荷が一時的に高くなる場合があります。

●scd_message_level=1|2

～《2》

このオペランドに 1 を指定すると、メッセージ格納バッファプールのメモリ不足が発生した場合に出力される KFCA00854-E メッセージの出力を抑止できます。

●`ipc_tcpnodelay=Y|N`

～《Y》

OpenTP1 がノード間で使用する通信ソケット（INET ドメイン）に、`TCP_NODELAY` オプションを使用するかどうかを指定します。

`TCP_NODELAY` オプションを使用すると（このオペランドに Y を指定すると）、Nagle アルゴリズムが無効になるので、送信済みデータの応答待ちの状態でも遅延させることなくデータを送信できます。ただし、`TCP_NODELAY` オプションを使用することで、INET ドメイン通信時の送信効率が低下し、ネットワークの負荷が大きくなる場合があります。この場合は、ネットワークの帯域などを考慮して `ipc_sendbuf_size` オペランド、`ipc_recvbuf_size` オペランドのチューニングを検討するか、このオペランドに N を指定して `TCP_NODELAY` オプションを無効にするかを検討してください。

●`watch_time=最大応答待ち時間`

～〈符号なし整数〉((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しない場合があります。

省略した場合は、システム共通定義の `watch_time` オペランドの値を仮定します。

このオペランドは、システム共通定義の `watch_time` オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の `watch_time` オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

次ページ以降に記述しています。

名称

共用するメッセージ格納バッファプールの指定

形式

```
{ {scdbufgrp  -g スケジュールバッファグループ名
    { [-e メッセージ格納バッファプール長] |
      [-n メッセージ格納バッファセル数] }
    [-l メッセージ格納バッファセル長]
    { [-s メッセージ格納バッファ使用制限サイズ] |
      [-p メッセージ格納バッファ使用制限率] } } }
```

機能

一つのメッセージ格納バッファプールを共用するユーザサーバの集合をスケジュールバッファグループといいます。ユーザサーバは、スケジュールバッファグループごとにメッセージ格納バッファプールを共用します。

スケジュールサービス定義では、スケジュールバッファグループごとに、作成するメッセージ格納バッファプールを指定します。

オプション

●-g スケジュールバッファグループ名

～ 〈1～8 文字の識別子〉

メッセージ格納バッファプールを共用する、スケジュールバッファグループの名称を指定します。

スケジュールバッファグループ名は、ノード内で一意になるようにしてください。同じスケジュールバッファグループ名を複数指定した場合は、エラーとなります。

●-e メッセージ格納バッファプール長

～ 〈符号なし整数〉 ((512～1610612736)) (単位：バイト)

共用するメッセージ格納バッファプールの長さを指定します。-n オプションとは同時に指定できません。-n オプションと同時に指定した場合、および指定可能範囲外の値を指定した場合は、OpenTP1 の起動エラーとなります。

このオペランドの指定値には、-l オプションで指定するメッセージ格納バッファセル長（省略時は 512 バイト）の整数倍となる値を指定します。指定した値が-l オプションの指定値の整数倍でない場合は、-l オプションの指定値の整数倍に切り上げた値が仮定されます。ただし、切り上げた結果が最大値（1610612736 バイト）を超える場合は、OpenTP1 の起動エラーとなります。

メッセージ格納バッファプール長とメッセージ格納バッファセル数、およびメッセージ格納バッファセル長の関係を次に示します。

メッセージ格納バッファセル数（端数は切り上げ）

$$= \text{メッセージ格納バッファプール長} \div \text{メッセージ格納バッファセル長}$$

メッセージ格納バッファプールは静的共用メモリ上に確保されるため、システム環境定義の `static_shmpool_size` オペランドの指定値を考慮してください。

-e オプションで定義エラーが発生した場合、KFCA00803-E メッセージを出力し、OpenTP1 の起動を中止します。

●-n メッセージ格納バッファセル数

～〈符号なし整数〉((1～3145728))《16》

共用するメッセージ格納バッファプールを構成する、セルの数を指定します。-e オプションとは同時に指定できません。-e オプションと同時に指定した場合、および指定可能範囲外の値を指定した場合は、OpenTP1 の起動エラーとなります。

●-l メッセージ格納バッファセル長

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

共用するメッセージ格納バッファプールを構成する、セルの長さを指定します。

各セルには、16 バイトのセル管理テーブルを作成します。また、バッファプールサイズが 31457280 バイトを超えた場合、またはメッセージ格納バッファプールの使用サイズを制限する（-s, -p オプションを指定）場合、各セルに 32 バイトのセル管理テーブルを作成します。このため、指定するメッセージ格納バッファセル長にはこれらのセル管理テーブルサイズを考慮してください。指定可能範囲外の値を指定した場合は、OpenTP1 の起動エラーとなります。

このオペランドの指定値には、8 の整数倍となる値を指定してください。指定した値が 8 の整数倍でない場合は、その値を 8 の整数倍に切り上げた値が仮定されます。

メッセージ格納バッファプール長とメッセージ格納バッファセル数、およびメッセージ格納バッファセル長の関係を、次に示します。

メッセージ格納バッファプール長

$$= \text{メッセージ格納バッファセル長} \times \text{メッセージ格納バッファセル数}$$

メッセージ格納バッファプールとして確保できるバッファプールの最大長は、1610612736 バイトになります。そのため、「メッセージ格納バッファセル数×メッセージ格納バッファセル長」の値がこれを超えると、エラーとなります。また、メッセージ格納バッファプールは静的共用メモリ上に確保されるため、システム環境定義の `static_shmpool_size` オペランドの指定値を考慮してください。

●-s メッセージ格納バッファ使用制限サイズ

～ 〈符号なし整数〉 ((512～1610612736)) (単位：バイト)

このスケジュールバッファグループに属するユーザサーバのメッセージ格納バッファ使用制限サイズの省略時解釈値を指定します。-p オプションとは同時に指定できません。同時に指定した場合、OpenTP1 の起動エラーとなります。MHP がスケジュールバッファグループを利用する場合、この指定は無視されます。

このオペランドの指定値には、-l オプションで指定したメッセージ格納バッファセル長の整数倍となる値を指定します。指定した値がメッセージ格納バッファセル長の整数倍でない場合は、整数倍に切り下げた値になります。

切り下げた結果、値が 0 になった場合は OpenTP1 の起動を中止します。

-s オプションで定義エラーが発生した場合、KFCA00803-E メッセージを出力し、OpenTP1 の起動を中止します。

●-p メッセージ格納バッファ使用制限率

～ 〈符号なし整数〉 ((1～100)) (単位：%)

このスケジュールバッファグループに属するユーザサーバのメッセージ格納バッファ使用制限サイズの省略時解釈値を指定します。-s オプションとは同時に指定できません。同時に指定した場合、OpenTP1 の起動エラーとなります。MHP がスケジュールバッファグループを利用する場合、この指定は無視されます。

このオペランドの指定値によって計算されたメッセージ格納バッファ使用制限サイズの値が、-l オプションで指定したメッセージ格納バッファセル長の整数倍でない場合は、メッセージ格納バッファ使用制限サイズは整数倍に切り下げた値になります。切り下げた結果、値が 0 になった場合は OpenTP1 の起動を中止します。

-p オプションで定義エラーが発生した場合、KFCA00803-E メッセージを出力し、OpenTP1 の起動を中止します。

名称

マルチスケジューラデーモンの情報の指定

形式

```
{ {scdmulti  [-m マルチスケジューラデーモン数]
               [-p ポート番号]
               [-g マルチスケジューラグループ名]
               [-t] } }
```

機能

マルチスケジューラ機能を提供するマルチスケジューラデーモンに関する情報を指定します。

マルチスケジューラ機能を使用する場合、この定義コマンドとともに次の定義コマンドおよびオペランドを指定する必要があります。

RPC 受信側：ユーザサービス定義の scdmulti 定義コマンド

RPC 送信側：ユーザサービス定義の multi_schedule オペランド

オプション

● -m マルチスケジューラデーモン数

～ 〈符号なし整数〉 ((1～4096)) 《1》

マルチスケジューラデーモンのプロセス数を指定します。

このオプションで起動するプロセスは、-p オプションの指定値から順にポート番号が割り当てられます。

● -p ポート番号

～ 〈符号なし整数〉 ((5001～65535))

マルチスケジューラデーモンのベースとなるポート番号を指定します。

マルチスケジューラ機能では、このオプション指定値から「(-p オプション指定値) + (-m オプション指定値) - 1」までのポート番号を、順にマルチスケジューラデーモンに割り当てて起動します。

このオプションは、scdmulti 定義コマンドを複数指定する場合、またはスケジューラサービス定義の scd_port オペランドを指定する場合には省略できます。その場合は scdmulti 定義コマンドがすでに実行されていれば、「直前の scdmulti 定義コマンドで使用したポート番号 + 1」がこのオプションの指定値となります。scdmulti 定義コマンドがまだ実行されてなく、スケジューラサービス定義の scd_port オペランドが指定されていれば、「scd_port オペランドの指定値 + 1」の値をこのオプションの指定値とします。そのほかの場合は定義エラーとなります。

このオプションの指定値によって決定した各マルチスケジューラデーモンのポート番号が、他スケジューラデーモンのポート番号と重複した場合は、定義エラーとなります。

●-g マルチスケジューラグループ名
～ 〈1～8 文字の識別子〉《scdmltgp》

scdmulti 定義コマンドで起動されたマルチスケジューラデーモンのグループ名を指定します。

マルチスケジューラデーモンを複数グループに分け、グループ間のサービス要求メッセージ受信処理が競合しないようにする場合にだけ指定する必要があります。

このオプションを指定した場合、この scdmulti 定義コマンドで起動されたマルチスケジューラデーモンにサービス要求をスケジューリングさせたいユーザサーバのユーザサービス定義で、scdmulti 定義コマンドの-g オプションに、このオプションに指定したマルチスケジューラグループ名を指定する必要があります。

このオプションを省略した場合、"scdmltgp"が指定されたものとします。したがって、"scdmltgp"はマルチスケジューラグループ名として使用しないでください。

●-t
マルチスケジューラデーモンがサービス要求を受信したとき、かつ、自ノードのユーザサーバが高負荷状態のときに、マルチスケジューラ機能を使用していないほかのノードのユーザサーバ（ユーザサービス定義の scdmulti 定義コマンドで、マルチスケジューラ機能を指定していないユーザサーバ）にサービス要求を一部、転送し負荷分散するかどうかを指定します。

自ノードのユーザサーバの閉塞などによって、スケジュールできない場合には、このオプションを指定しなくても、サービス要求が負荷分散されます。

このオプションを指定すると、マルチスケジューラ機能を使用していないユーザサーバへの負荷分散処理はマスタスケジューラデーモン経由になるため、マスタスケジューラデーモンに負荷が集中することがあります。

サービス要求の負荷分散が行われる条件を次の表に示します。

表 3-3 サービス要求の負荷分散が行われる条件

scdmulti 定義コマンドの-tオプションの指定	サーバの状態		サービス要求の負荷分散
	ノード A (サービス要求の転送元) (マルチスケジューラ機能の指定あり)	ノード B (サービス要求の転送先) (マルチスケジューラ機能の指定なし)	
指定あり	通常状態	通常状態	×
		高負荷状態	×
		スケジュールできない状態	×
	高負荷状態	通常状態	○

scdmulti 定義コマンドの -t オプションの指定	サーバの状態		サービス要求の負荷分散
	ノード A (サービス要求の転送元) (マルチスケジューラ機能の指定あり)	ノード B (サービス要求の転送先) (マルチスケジューラ機能の指定なし)	
指定あり	高負荷状態	高負荷状態	×
		スケジュールできない状態	×
	スケジュールできない状態	通常状態	○
		高負荷状態	○
		スケジュールできない状態	×※
指定なし	通常状態	通常状態	×
		高負荷状態	×
		スケジュールできない状態	×
	高負荷状態	通常状態	×
		高負荷状態	×
		スケジュールできない状態	×
	スケジュールできない状態	通常状態	○
		高負荷状態	○
		スケジュールできない状態	×※

(凡例)

- ：サービス要求を負荷分散します。
- ×：サービス要求を負荷分散しません。

注※

ユーザサーバの閉塞が原因で、スケジュールできない状態の場合だけ、スケジューラサービスは、ユーザサーバがスケジュールできない状態であることを把握できます。

ユーザサーバの閉塞以外の原因（例：メッセージ格納バッファ不足）で、スケジュールできない状態の場合は、サービス要求の転送先からサービス要求元にエラーが通知されます。

注意事項

TP1/Client からマルチスケジューラ機能を使用してサービスを要求する場合については、マニュアル「OpenTP1 クライアント使用の手引」を参照してください。

マルチスケジューラデーモンが使用するポート番号は、必ず割り当て可能としてください。

-p オプションに指定するポート番号は、他プログラムで使用するウェルノウンポート番号と重複しない値で、かつ、OS が他プロセスに自動的に割り当てない値を指定してください。

マルチスケジューラデーモン起動時に、指定されたポート番号が割り当てられない場合は、スケジューラサービス開始処理をエラーとし、OpenTP1 システムの起動を中止します。

マルチスケジューラデーモンは、OpenTP1 のシステムサーバとして起動するため、全 scdmulti 定義コマンドの-m オプションの指定値の合計を、システム環境定義の server_count オペランド、およびプロセスサービス定義の prc_process_count オペランドに加える必要があります。server_count オペランドの指定値が不足した場合は、マスタスケジューラデーモンが「hm02102」、または「hm02301」でアボートします。

この定義コマンドは複数指定できます。ただし、-g オプションを省略したこの定義コマンドを複数指定した場合、または-g オプションに同一名称を指定したこの定義コマンドを複数指定した場合は、定義エラーとし、OpenTP1 システムの起動を中止します。

OpenTP1 システム内の同一サービスグループに、マルチスケジューラ機能を使用しているユーザサーバと、マルチスケジューラ機能を使用していないユーザサーバが混在する場合、マルチスケジューラ機能を使用したサービス要求は、マルチスケジューラ機能を使用しているユーザサーバに優先して負荷分散されます。

この定義コマンドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

トランザクションサービス定義

形式

set 形式

```
set trn_tran_process_count=同時に起動するトランザクションブランチの数
[set trn_recovery_process_count=並行回復プロセス数]
[set trn_expiration_time=トランザクションブランチ限界経過時間]
[set trn_expiration_time_suspend=Y|N|F]
[set trn_tran_statistics=Y|N]
[set trn_tran_recovery_list=Y|N]
[set trn_cpu_time=トランザクションブランチCPU監視時間]
[set trn_statistics_item=統計情報項目 [, 統計情報項目] ...]
[set trn_max_subordinate_count=最大子トランザクションブランチ数]
[set trn_rm_open_close_scope=process|transaction]
[set trn_optimum_item=トランザクション最適化項目
    [, トランザクション最適化項目] ...]
[set trn_processing_in_rm_error=down|retry|force]
[set trn_recovery_list_remove=normal|force|no]
[set trn_recovery_list_remove_level=未決着トランザクション情報
    ファイルの削除レベル]

[set trn_crm_use=Y|N]
[set trn_max_crm_subordinate_count=最大CRM経由
    子トランザクションブランチ数]
[set trn_watch_time=トランザクション同期点処理時の最大通信待ち時間]
[set trn_rollback_information_put=no|self|remote|all]
[set trn_limit_time=トランザクションブランチ最大実行可能時間]
[set trn_rollback_response_receive=Y|N]
[set trn_partial_recovery_type=type1|type2|type3]
[set max_socket_descriptors=ソケット用ファイル記述子の最大数]
[set trn_recovery_failmsg_interval=仕掛中トランザクション情報
    メッセージ間隔最小時間]

[set trn_wait_rm_open=continue|stop|retry_continue|retry_stop]
[set trn_retry_interval_rm_open=xa_open関数発行リトライ
    インタバル時間]

[set trn_retry_count_rm_open=xa_open関数発行リトライ回数]
[set thread_stack_size=openTP1内部で使用するスレッドスタック領域
    のサイズ]
[set polling_control_data=Y|N]
[set thread_yield_interval=ソケットの再利用指示を受信できる契機を
    与えるインタバル時間]

[set groups=グループ識別子 [, グループ識別子] ...]
[set trn_xar_use=Y|N]
[set trn_start_recovery_mode=stop|wait|continue]
[set trn_start_recovery_watch_time=リトライ経過時間]
[set trn_start_recovery_interval=リトライ間隔時間]
[set trn_xa_commit_error=down|force]
[set trn_prf_event_trace_level=TRNイベントトレースの取得レベル]
[set trn_prf_event_trace_condition=トレース取得種類
    [, トレース取得種類] ]

[set trn_completion_limit_time=トランザクション完了限界時間]
[set trn_extend_function=トランザクションサービスの機能の拡張レベル]
[set trn_rcv_open_close_scope=Y|N]
[set watch_time=最大応答待ち時間]
```

コマンド形式

```
trnstring -n リソースマネージャ名  
          [-i リソースマネージャ拡張子 [, リソースマネージャ拡張子] ...]  
          [-o "トランザクションサービス用xa_open関数用文字列"]  
          [-c "トランザクションサービス用xa_close関数用文字列"]  
          [-O "ユーザサーバ用xa_open関数用文字列"]  
          [-C "ユーザサーバ用xa_close関数用文字列"] [-d]  
          [-e] [-m] [-r] [-s]
```

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] }}
```

機能

トランザクション管理をするための実行環境を定義します。

説明

set 形式のオペランド

●trn_tran_process_count=同時に起動するトランザクションブランチの数

～ 〈符号なし整数〉 ((1～8192*))

注※

MCF サービスを使用する場合の指定範囲は、次のとおりです。

- 32 ビットの場合：((1～7484))
- 64 ビットの場合：((1～6893))

同時に起動するトランザクションブランチの数を指定します。

トランザクション処理を実行するユーザサーバプロセス数、回復プロセス数、および回復処理を待っているトランザクションブランチ数の総数を指定します。

トランザクションブランチを発生させたユーザサーバプロセスが異常終了した場合、回復プロセスがほかのトランザクションブランチの決着を処理中のとき、このトランザクションブランチは回復処理を待っている状態になります。この状態が長く続くと起動できるトランザクションブランチ数が少なくなることがあります。

このため、異常終了後にトランザクションブランチを発生できるユーザサーバプロセスの扱いや、異常発生頻度などを考慮してトランザクションブランチ数を設定する必要があります。指定数の目安を次に示します。

(トランザクション実行するユーザサーバプロセス数+回復プロセス数)

<指定するトランザクションブランチ数

≦ (トランザクション実行するユーザサーバプロセス数×2 +回復プロセス数)

なお、指定数が多くなればなるほどメモリ資源の効率が悪くなるので注意してください。

●trn_recovery_process_count=並行回復プロセス数

～ 〈符号なし整数〉 ((1～128)) 《4》

トランザクションブランチが異常終了した時に、トランザクションブランチの回復処理を並行してできる数を指定します。

トランザクションブランチが異常終了した場合は、ここで指定した数だけ並行してトランザクションブランチの回復処理をします。

●trn_expiration_time=トランザクションブランチ限界経過時間

～ 〈符号なし整数〉 ((0～65535)) 《0》 (単位：秒)

トランザクションブランチの処理を監視する限界経過時間を指定します。指定時間を超えた場合は、このトランザクションブランチのプロセスは異常終了して、ロールバックされます。0を指定した場合は、時間監視しません。

UAP が異常終了した場合に閉塞するかどうかは hold オペランド、term_watch_time オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

なお、このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

また、これらの定義のほかに関数を発行して監視時間を指定することもできます。

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

トランザクショナル RPC 実行プロセス (clttrnd) または CUP 実行プロセス (cltcond) で、この機能を有効にするには、クライアント環境定義またはクライアントサービス定義の trn_expiration_time オペランドを指定する必要があります。

注

サービス関数開始から終了までの実行監視時間の精度は秒単位です。そのため、タイミングによっては、このオペランドに指定した値よりも短い時間でプロセスを強制停止することがあります。サービス関数開始から終了までの実行監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、このオペランドには 3（単位：秒）以上の値の指定を推奨します。

●trn_expiration_time_suspend=Y|N|F

～《N》

トランザクションブランチの処理を監視するとき、次の処理時間も監視時間を含むかどうかを指定します。

1. 監視対象のトランザクションブランチが、RPC 機能を使ってほかのトランザクションブランチを呼び出し、その処理が終わるのを待つ時間
2. 連鎖 RPC で呼び出されたサーバ UAP が、次のサービス要求を待つ時間
3. 監視対象のトランザクションブランチが、非同期 RPC を使用してほかのトランザクションブランチを呼び出したあと、処理結果受信処理を行っている時間

Y

1., 2., 3.のすべてを監視時間を含みます。

N

3.だけを監視時間を含みます。

F

1., 2., 3.のどれも監視時間を含みません。

なお、このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです（1.> 2.> 3.）。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

このオペランドと各種タイマ監視との関係については、「[付録 A.2 トランザクションの時間監視](#)」を参照してください。

トランザクショナル RPC 実行プロセス（clttrnd）または CUP 実行プロセス（cltcond）で、この機能を有効にするには、クライアント環境定義またはクライアントサービス定義の trn_expiration_time_suspend オペランドを指定する必要があります。

●trn_tran_statistics=Y|N

～《N》

トランザクションブランチごとの統計情報を取得するかどうかを指定します。

Y

統計情報を取得します。

N

統計情報を取得しません。

Y を指定した場合、OpenTP1 システムの起動後から停止までの間、ユーザサービス定義の `trn_statistics_item` オペランドで、`nothing` 以外を指定されたユーザサーバが実行したトランザクションブランチの統計情報を取得します。

統計情報の取得開始または終了は、`trnstics` コマンドでも指示できます。その場合このオペランドの指定は無効となります。

統計情報を取得する場合、取得情報の種類が多くなるほど、トランザクション性能が低下します。そのため、統計情報の取得は、システムチューニング時やトラブル調査時など、トランザクション性能に影響を与えてもかまわない場合だけにしてください。

●`trn_tran_recovery_list=Y|N`

～《N》

OpenTP1 システムの全面回復時に、未決着トランザクション情報を取得するかどうかを指定します。OpenTP1 は、システムの全面回復のたびに `$DCDIR/spool/dctrninf` の下にファイルを作成して、この情報を出力します。不要となったファイルは削除してください。

Y

未決着トランザクション情報を取得します。

N

未決着トランザクション情報を取得しません。

●`trn_cpu_time=トランザクションブランチ CPU 監視時間`

～〈符号なし整数〉((0～65535))《0》(単位：秒)

トランザクションブランチが同期点処理までに使用できる CPU 時間を指定します。0 を指定した場合は、時間監視しません。指定時間を超えた場合は、該当するトランザクションブランチのプロセスを異常終了し、ロールバックします。

UAP が異常終了した場合に閉塞するかどうかは `hold` オペランド、`term_watch_time` オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

なお、このオペランドは、ユーザサービス定義または `rap` リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

トランザクショナル RPC 実行プロセス (clttrnd) または CUP 実行プロセス (cltcond) で、この機能を有効にするには、クライアント環境定義またはクライアントサービス定義の trn_cpu_time オペランドを指定する必要があります。

●trn_statistics_item=統計情報項目〔, 統計情報項目〕…

～《executiontime》

トランザクションブランチの統計情報を取得する項目を指定します。

nothing

統計情報を取得しません。

base

基本情報として、次の情報を取得します。

- トランザクションブランチの識別子
- トランザクションブランチの決着結果
- トランザクションブランチの実行プロセス種別
- トランザクションブランチの実行サーバ名
- トランザクションブランチの実行サービス名

executiontime

基本情報とトランザクションブランチの実行時間情報を取得します。

cputime

基本情報とトランザクションブランチの CPU 時間情報を取得します。

nothing の指定は、一つしかできません。また、nothing とほかの統計情報項目を同時に指定した場合、nothing の指定は無効になります。

トランザクションに関する統計情報を取得するときは次のどれかを指定してください。

- trn_tran_statistics オペランドに Y を指定
- trnstics コマンドで-s オプションを指定

なお、このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

トランザクショナル RPC 実行プロセス (clttrnd) または CUP 実行プロセス (cltcond) で、この機能を有効にするには、クライアント環境定義またはクライアントサービス定義の trn_statistics_item オペランドを指定する必要があります。

●trn_max_subordinate_count=最大子トランザクションブランチ数

～ 〈符号なし整数〉 ((0～1024)) 《32》

OpenTP1 下の一つのトランザクションブランチから生成する子トランザクションブランチの最大数を指定します。この最大数は、一つのトランザクションブランチから発行する dc_rpc_call 関数（ただし、呼び先で更新処理、または UJ 出力する）の最大数です。

●trn_rm_open_close_scope=process|transaction

～ 《process》

OpenTP1 以外が提供しているリソースマネージャに対して、XA インタフェースである二つの関数 (xa_open 関数と xa_close 関数) の発行タイミングを指定します。

process

dc_rpc_open 関数発行時に xa_open 関数を発行し、dc_rpc_close 関数発行時に xa_close 関数を発行します。

transaction

トランザクション開始時に xa_open 関数を発行し、トランザクション終了時に xa_close 関数を発行します。

process を指定した場合は、dc_rpc_open 関数の発行から dc_rpc_close 関数の発行まで、リソースマネージャの資源を占有します。したがって、トランザクションの性能を重視する場合は process を指定してください。

また、transaction を指定した場合は、トランザクションの開始と終了ごとに xa_open 関数、xa_close 関数を発行します。したがって、リソースマネージャの資源を有効利用したい場合は、transaction を指定してください。

なお、このオペランドは、ユーザサービス定義およびユーザサービスデフォルト定義にも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. トランザクション定義

●trn_optimum_item=トランザクション最適化項目 [、トランザクション最適化項目] …

～《base》

複数のユーザサーバで構成されるグローバルトランザクションの性能を向上させるための最適化項目を、次の文字列で指定します。

base

同期点処理全体（プリペア処理、コミット処理、およびロールバック処理）を最適化します。OpenTP1 のトランザクション制御は 2 相コミット方式で実行しているため、二つのトランザクションブランチ間のコミット制御には、4 回のプロセス間通信が必要となります。

次の条件をすべて満たす場合、親トランザクションブランチが子トランザクションブランチのコミット処理を代わりに実行することで、コミット制御に必要な 4 回のプロセス間通信を削減します。

1. 親トランザクションブランチと、子トランザクションブランチが同一 OpenTP1 下にあること。
2. 親トランザクションブランチが、子トランザクションブランチを同期応答型 RPC で呼び出していること。
3. 子トランザクションブランチでアクセスしたリソースマネージャの XA インタフェース用オブジェクトが、親トランザクションブランチにもリンケージされていること。

asyncprepare

base の指定条件を満たしていないため同期点処理全体の最適化ができない場合に、プリペア処理を最適化します。

次の条件をすべて満たす場合、親トランザクションブランチから発行された RPC によって子トランザクションブランチがサービス要求を実行したときに、RPC のリターン前にプリペア処理を実行することで、2 回のプロセス間通信を削減します。

1. base を指定した最適化ができないこと。
2. 親トランザクションブランチが、子トランザクションブランチを同期応答型 RPC で呼び出していること。

ただし、この最適化を実行した場合、親トランザクションブランチが発行した同期応答型 RPC の応答時間が遅くなります。また、子トランザクションブランチは、プリペア処理からコミット処理までの間隔（親トランザクションブランチからの指示がないとトランザクションを決着できない状態）が大きくなります。そのため、親トランザクションブランチの OpenTP1 がシステムダウンし、トランザクションブランチ間の連絡ができなくなると、ジャーナルファイルのスワップやチェックポイントダンプファイルの有効化が遅れ、子トランザクションブランチの OpenTP1 もシステムダウンする場合があります。

トランザクション最適化項目は、重複して指定できます。ただし、優先順位は次のようになります（1.> 2.）。

1. base
2. asyncprepare

なお、このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

トランザクショナル RPC 実行プロセス (clttrnd) または CUP 実行プロセス (cltcond) で、この機能を有効にするには、クライアント環境定義またはクライアントサービス定義の trn_optimum_item オペランドを指定する必要があります。

●trn_processing_in_rm_error=down|retry|force

～《down》

トランザクション回復サービスによるトランザクションの回復時に、リソースマネージャに対して発行した XA 関数 (xa_commit 関数, xa_forget 関数, xa_rollback 関数) が処理を続行できないコード (XAER_INVAL (-5), XAER_PROTO (-6)) でエラーリターンした場合の OpenTP1 の処置を指定します。

このような障害は、次のような場合に発生し、ほかのリソースと不整合になっているおそれがあります。

- リソースマネージャに障害が発生した場合
- OpenTP1 のトランザクション制御下で、XA 連携しているリソースマネージャが提供するトランザクション制御機能を使用した場合
- XA 連携時にリソースマネージャが禁止している機能を使用した場合

このような障害が発生した場合は、KFCA00907-E および KFCA00908-E のメッセージを出力します。

down

OpenTP1 システムをダウンさせます。

システムダウン後は、すべてのリソースを正常な状態に回復し、OpenTP1 システムを強制正常開始してください。

retry

トランザクションが決着できるまで、障害となったリソースマネージャに対して一定間隔でトランザクションの決着を指示します。

この指定は、リソースマネージャの障害を何らかの方法で回復できる場合、または指定を force に変更する前にリソースマネージャの状態を確認し、アクセス禁止などを処置する場合に指定します。

force

障害となったリソースマネージャを無視して、トランザクションを強制的に決着させます。

この指定は、トランザクション強制決着後、障害となったリソースマネージャをほかのトランザクションがアクセスしてもかまわない場合に指定してください。

このオペランドは、障害発生時に処置を決定できるよう、オンライン中に変更できます。そのため、システム開始時には retry を指定しておき、障害発生時に障害となったリソースマネージャに必要な処置をしたあと、指定を force に変更するというような運用ができます。

●trn_recovery_list_remove=normal|force|no

～《no》

OpenTP1 開始時に、未決着トランザクション情報ファイルを削除するかどうかを指定します。

normal

OpenTP1 正常開始時に削除します。

force

OpenTP1 開始時に削除します。

no

OpenTP1 開始時に削除しません。

●trn_recovery_list_remove_level=未決着トランザクション情報ファイルの削除レベル

～〈符号なし整数〉((0～24855))《0》(単位：日)

未決着トランザクション情報ファイルの削除レベルを指定します。このオペランドは、trn_recovery_list_remove オペランドで normal または force を指定した場合に有効です。

0

全ファイルを削除します。

1～24855

トランザクションサービス開始時刻から起算して、「指定値×24」時間より前に作成されたファイルを削除します。

●trn_crm_use=Y|N

～《N》

通信リソースマネージャ（CRM）を使用するかどうかを指定します。

Y

CRM を使用します。

N

CRM を使用しません。

●trn_max_crm_subordinate_count=最大 CRM 経由子トランザクションブランチ数

～〈符号なし整数〉((0～1024))《8》

OpenTP1 下の一つのトランザクションブランチから生成される, CRM 経由の RPC の最大数を指定します。

●trn_watch_time=トランザクション同期点処理時の最大通信待ち時間

～ 〈符号なし整数〉 ((1～65535)) 《120》 (単位：秒)

トランザクションの同期点処理で, トランザクションブランチ間で行う通信 (プリペア, コミット, ロールバック指示, または応答など) の受信待ち時間の最大値を指定します。

指定時間を過ぎても指示または応答がない場合は, 該当するトランザクションブランチが 2 相コミットの 1 相目完了前であればロールバックさせ, 1 相目完了後であればトランザクションサービスのシステムプロセスでトランザクション決着処理を再試行します。

このオペランドは, ユーザサービス定義または rap リスナーサービス定義, およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

これらのすべての定義でこのオペランドを省略した場合, トランザクションを行ったユーザサーバの watch_time の値を仮定します。ただし, trn_watch_time には 0 秒を指定できないため, ユーザサーバの watch_time に 0 秒を指定した場合は trn_watch_time のデフォルト値である 120 秒を仮定します。

トランザクショナル RPC 実行プロセス (clttrnd) または CUP 実行プロセス (cltcond) で, この機能を有効にするには, クライアント環境定義またはクライアントサービス定義の trn_watch_time オペランドを指定する必要があります。

●trn_rollback_information_put=no|self|remote|all

～ 《no》

トランザクションブランチがロールバックした場合に, ロールバック要因に関する情報をログに取得するかどうかを指定します。

no

ロールバック情報を取得しません。

self

ロールバック要因が発生したトランザクションブランチでだけ, ログにロールバック情報を取得します。

remote

self に加え, 他ノードのトランザクションブランチからロールバック要求されたトランザクションブランチでも, ログにロールバック情報を取得します。

all

remote に加え、自ノードのトランザクションブランチからロールバック要求されたトランザクションブランチでも、ログにロールバック情報を取得します。

このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

トランザクショナル RPC 実行プロセス (clttrnd) または CUP 実行プロセス (cltcond) で、この機能を有効にするには、クライアント環境定義またはクライアントサービス定義の trn_rollback_information_put オペランドを指定する必要があります。

このオペランドによって得たログを使用することで障害調査が容易になります。したがって、このオペランドに no を指定しているか、または何も指定していない場合は、このオペランドに all を指定することをお勧めします。

●trn_limit_time=トランザクションブランチ最大実行可能時間

～ 〈符号なし整数〉 ((0～65535)) 《0》 (単位：秒)

トランザクションブランチの最大実行可能時間を指定します。

トランザクションブランチを開始してから同期点処理が終了するまでの時間が、このオペランド指定時間を超えないように dc_rpc_call 関数、dc_rpc_poll_any_replies 関数および同期点処理内で行う通信のタイムアウト時間を次のように自動設定します。

- dc_rpc_call 関数および dc_rpc_poll_any_replies 関数のタイムアウト時間
「 $K \geq$ このオペランド指定時間」の場合は、要求処理を実行しないで、タイムアウトでエラーリターンします。
「 $K <$ このオペランド指定時間」でかつ「 $(\text{このオペランド指定時間} - K) \geq W$ 」の場合は、 W をタイムアウト時間とします。
「 $K <$ このオペランド指定時間」でかつ「 $(\text{このオペランド指定時間} - K) < W$ 」の場合は、 $(\text{このオペランド指定時間} - K)$ をタイムアウト時間とします。

K

現時刻－トランザクションブランチ開始時刻

W

dc_rpc_call 関数の場合は watch_time オペランド指定時間

dc_rpc_poll_any_replies 関数の場合は引数 timeout 指定時間

- 同期点処理内で行う通信のタイムアウト時間

「 $K \geq \text{このオペランド指定時間}$ 」の場合は、タイムアウト時間を 1 秒とします。

「 $K < \text{このオペランド指定時間}$ 」でかつ「 $(\text{このオペランド指定時間} - K) \geq W$ 」の場合は、 W をタイムアウト時間とします。

「 $K < \text{このオペランド指定時間}$ 」でかつ「 $(\text{このオペランド指定時間} - K) < W$ 」の場合は、 $(\text{このオペランド指定時間} - K)$ をタイムアウト時間とします。

K

現時刻 - トランザクションブランチ開始時刻

W

`trn_watch_time` オペランド指定時間

`trn_watch_time` オペランドを省略した場合は `watch_time` オペランド指定時間

上記の受信待ち以外の処理で時間が掛かった場合は、このオペランド指定時間以内にトランザクションブランチが終了しないことがあります。

同期点処理開始前にこのオペランドの指定時間が経過した場合、そのトランザクションはロールバックします。

0 を指定した場合は、時間監視をしません。

なお、このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

トランザクショナル RPC 実行プロセス (`clttrnd`) または CUP 実行プロセス (`cltcond`) で、この機能を有効にするには、クライアント環境定義またはクライアントサービス定義の `trn_limit_time` オペランドを指定する必要があります。

●`trn_rollback_response_receive=Y|N`

～《Y》

RPC 先トランザクションブランチにロールバック指示を送信したあと、ロールバック完了通知を受信するかどうかを指定します。

N を指定した場合、RPC 先トランザクションブランチからのロールバック完了通知を受信しないで (RPC 先トランザクションブランチのロールバック処理の完了を待たないで) 自トランザクションブランチを終了します。

なお、このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

トランザクショナル RPC 実行プロセス (clttrnd) または CUP 実行プロセス (cltcond) で、この機能を有効にするには、クライアント環境定義またはクライアントサービス定義の trn_rollback_response_receive オペランドを指定する必要があります。

●trn_partial_recovery_type=type1|type2|type3

～《type1》

UAP 障害時のトランザクション同期点処理方式を指定します。

RPC がタイムアウトし、RPC 発行先プロセスのアドレスが未解決の場合やトランザクション実行中の UAP がダウンした場合に、トランザクションブランチ間の連絡がスムーズにできないで、トランザクションの決着に時間が掛かることがあります。

このオペランドでは、次に示す障害が発生した場合のトランザクション同期点処理方式を、指定値に示す三つの方式から選択して指定します。

(障害 1) RPC がタイムアウトした場合

この場合、RPC 発行元トランザクションブランチは、サービス要求がどのプロセスで実行されているかがわからないため、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信できないで、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチがトランザクション同期点メッセージ待ちとなり、トランザクションの決着に時間が掛かります。

(障害 2) RPC 発行元 UAP が RPC の応答受信前にダウンした場合

この場合、RPC 発行元トランザクションブランチは、サービス要求がどのプロセスで実行されているかがわからないため、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信できないで、RPC 発行先トランザクションブランチはトランザクション同期点メッセージ待ちとなり、トランザクションの決着に時間が掛かります。

(障害 3) RPC 発行先 UAP からの応答受信後に RPC 発行元 UAP と RPC 発行先 UAP がほぼ同時にダウンした場合

この場合、それぞれのトランザクションブランチを引き継いだトランザクション回復プロセスは、相手 UAP プロセスのダウンを知らないため、すでに存在しない UAP プロセスにトランザクション同期点メッセージを送信してしまい、トランザクションの決着に時間が掛かることがあります。

type1

(障害 1) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

(障害 2) が発生した場合、RPC 発行元トランザクションブランチは、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信しないでトランザクションを決着します。RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

(障害 3) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

type2

(障害 1) が発生してトランザクションをコミットする場合は type1 と同じです。

(障害 1) が発生してトランザクションをロールバックする場合、または (障害 2) が発生した場合は、RPC 発行元トランザクションブランチは、RPC 発行先トランザクションブランチが存在するノードのトランザクションサービスプロセスにトランザクション同期点メッセージを送信後、トランザクションを決着します。トランザクション同期点メッセージを受信したトランザクションサービスプロセスは、該当するトランザクションブランチを処理中のプロセスに、トランザクション同期点指示を送信します。

(障害 3) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

type3

(障害 1) が発生してトランザクションをコミットする場合は、type1 と同じです。

(障害 1) が発生してトランザクションをロールバックする場合、(障害 2) が発生した場合、または (障害 3) が発生した場合、相手トランザクションブランチが存在するノードのトランザクションサービスプロセスに、トランザクション同期点メッセージを送信します。トランザクション同期点メッセージを受信したトランザクションサービスプロセスは、該当するトランザクションブランチを処理中のプロセスに、トランザクション同期点指示を送信します。

次に示す場合、このオペランドに type2 または type3 を指定しても、トランザクションの決着に時間が掛かることがあります。

1. RPC 実行中に、RPC 発行先 UAP の状態が変更となり (負荷増加, UAP 終了, UAP 閉塞など)、ほかのノードの同一 UAP にサービス要求が再転送された場合
2. 相手先の OpenTP1 がこのオプションをサポートしていないバージョンの場合
3. 相手先トランザクションブランチがトランザクション同期点メッセージ受信処理以外で時間が掛かっている場合

なお、このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義

3. トランザクションサービス定義

トランザクショナル RPC 実行プロセス (clttrnd) または CUP 実行プロセス (cltcond) で、この機能を有効にするには、クライアント環境定義またはクライアントサービス定義の `trn_partial_recovery_type` オペランドを指定する必要があります。

●max_socket_descriptors=ソケット用ファイル記述子の最大数

～ 〈符号なし整数〉 ((32～2032))

OpenTP1 制御下のプロセス※でのソケット用に使用するファイル記述子の最大数を指定します。

OpenTP1 制御下のプロセス※は、システムサーバやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数、および通信する他ノードの数によって、このオペランドを変更する必要があります。

注※

MCF サービス (MCF マネージャサービス, MCF 通信サービス, およびアプリケーション起動サービス) 以外の OpenTP1 プロセスが対象です。MCF サービスの場合は、「[システムサービス情報定義](#)」および「[システムサービス共通情報定義](#)」を参照してください。

ソケット用ファイル記述子の最大数の算出式を、次に示します。

$\uparrow (\text{自ノード内 UAP プロセス数}^{\ast 1} + \text{トランザクションサービスを要求してくるノード内の trn システムプロセス数}^{\ast 2} + \text{システムサービスプロセス数}^{\ast 3}) / 0.8 \uparrow$

↑↑ : 小数点以下を切り上げます。

注※1

自ノード内 UAP プロセス数は、次に示す値の合計です。

- 自 OpenTP1 内の UAP で、トランザクショナル RPC でサービス要求を送信する UAP プロセス数
- 自 OpenTP1 内のサービスに対し、トランザクショナル RPC でサービスを要求してくる他ノード内の UAP プロセス数
- CUP から同時に起動するトランザクション数 (クライアントサービス定義の `parallel_count` オペランドの指定値)

注※2

トランザクションサービスを要求してくるノードの trn システムプロセス数とは、次に示す値の合計です。

- 自 OpenTP1 の `all_node` オペランドに指定したノードのうち、自 OpenTP1 からトランザクショナル RPC でトランザクションを拡張するノードの、トランザクションサービス定義の `trn_recovery_process_count` オペランドの値に 1 を加えた数

- ・ 自ノード名を、OpenTP1 の all_node オペランドに指定している他ノードのうち、トランザクショナル RPC で自 OpenTP1 にトランザクションを拡張しているノードの、トランザクションサービス定義の trn_recovery_process_count オペランドの値に 1 を加えた数

注※3

システムサービスプロセス数とは、自 OpenTP1 内のシステムサービスプロセスの数です。自 OpenTP1 内のシステムサービスプロセスは、rpcstat コマンドで表示されるサーバ名をカウントすることで求められます。rpcstat コマンドで表示されるサーバ名のうち、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造に記載されているシステムサービスプロセスをカウントしてください。

このオペランドの指定が小さいと、OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため、プロセスが KFCA00307-E メッセージを出力して異常終了します。

ここで指定を省略した場合、システム共通定義の値を仮定します。

●trn_recovery_failmsg_interval=仕掛中トランザクション情報メッセージ間隔最小時間

～ 〈符号なし整数〉 ((0～65535)) 《1800》 (単位：秒)

KFCA00960-I メッセージを出力する間隔を指定します。

トランザクションを回復できなかった際、以前に KFCA00960-I メッセージを出力してからこのオペランドの指定時間が経過した場合には、再びメッセージを出力します。ただし次の場合はすぐに出力します。

- ・ トランザクション開始後、最初に回復に失敗したとき
- ・ OpenTP1 終了処理開始後、最初に回復に失敗したとき
- ・ OpenTP1 再開後、最初に回復に失敗したとき

指定値が 0 の場合には KFCA00960-I メッセージを出力しません。

複数のトランザクションブランチを回復している場合は、このオペランドはそれぞれのトランザクションブランチに適用されます。同一のトランザクションブランチのメッセージだけが、このオペランドで指定された間隔で出力されます。

なお、1 回のトランザクションの回復失敗で、KFCA00960-I メッセージが 2 個以上出力される場合があります。これは、回復失敗の要因が複数存在するためです。KFCA00960-I メッセージは、KFCA00990-I メッセージよりもあとに出力される場合があります。

●trn_wait_rm_open=continue|stop|retry_continue|retry_stop

～ 《continue》

OpenTP1 起動時に、OpenTP1 以外が提供するリソースマネージャのオープン処理(xa_open 関数)でエラーが発生したときの処置を指定します。

continue

xa_open 関数エラーを無視して OpenTP1 開始処理を続行します。

stop

xa_open 関数エラーが発生した場合、OpenTP1 開始処理を中止します。

retry_continue

xa_open 関数エラーが発生した場合、OpenTP1 開始処理を中断し、trn_retry_interval_rm_open オペランドで指定したインタバルで trn_retry_count_rm_open オペランドで指定した回数だけ xa_open 関数発行をリトライします。xa_open 関数エラーが解消されなかったときは、エラーを無視して OpenTP1 開始処理を続行します。

retry_stop

xa_open 関数エラーが発生した場合、OpenTP1 開始処理を中断し、trn_retry_interval_rm_open オペランドで指定したインタバルで trn_retry_count_rm_open オペランドで指定した回数だけ xa_open 関数発行をリトライします。xa_open 関数エラーが解消されなかったときは、OpenTP1 開始処理を中止します。

●trn_retry_interval_rm_open=xa_open 関数発行リトライインタバル時間

～ 〈符号なし整数〉 ((1～3600)) 《10》 (単位：秒)

trn_wait_rm_open オペランドで retry_continue か retry_stop を指定したときの xa_open 関数発行リトライをする間隔を指定します。

●trn_retry_count_rm_open=xa_open 関数発行リトライ回数

～ 〈符号なし整数〉 ((1～65535)) 《18》

trn_wait_rm_open オペランドで retry_continue か retry_stop を指定したときの xa_open 関数発行リトライをする回数を指定します。

●thread_stack_size=OpenTP1 内部で使用するスレッドスタック領域のサイズ

～ 〈符号なし整数〉 ((1024～524288)) 《49152※》 (単位：バイト)

注※

AIX 版の uCosminexus TP1/Server Base(64)の場合は、デフォルト値が 65536 となります。

OpenTP1 の内部で使用するスレッドスタック領域のサイズを指定します。リソースマネージャ監視サービスのスレッドスタック領域は 262144 バイトに拡張されているため、XA 連携するリソースマネージャで 262144 バイト以上のスレッドスタック領域が必要な場合だけ指定してください。このオペランドを指定した場合のメモリ使用量の算出式を次に示します。

スレッドスタック領域のサイズ (単位：バイト) = 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に 2 を加えた数 × thread_stack_size オペランドの指定値 × 2

●polling_control_data=Y|N

～ 《N》

トランザクションリカバリデーモンで、トランザクション回復要求の受信待ちの状態のときに、定期的に待ち状態に割り込んで、一時クローズ処理要求が到着していないかどうかを検査することを指示します。

Y

トランザクション回復要求の受信待ち状態に定期的に割り込み、一時クローズ処理要求が到着していないかどうかを検査します。

N

トランザクション回復要求の受信待ち状態に割り込まないで、トランザクション回復要求が到着するまで待ち状態を継続します。

●thread_yield_interval=ソケットの再利用指示を受信できる契機を与えるインタバル時間

～〈符号なし整数〉((1～86400))《90》(単位：秒)

トランザクションリカバリデーモンで、トランザクション回復要求の受信待ちの状態のときに、定期的に待ち状態に割り込んで、一時クローズ処理要求が到着していないかどうかを検査する場合の検査インタバル時間を秒単位で指定します。

また、トランザクション回復要求を待つことなく次々にトランザクション回復要求を受け付けている場合には、連続してトランザクション回復要求を受け付けた時間が一定時間を超えたところで、一時クローズ処理要求が到着していないかどうかを検査します。このオペランドは、このインタバル値としても使用されます（連続してトランザクション回復要求を受け付けた時間がこのオペランド指定値を超えたときの一時クローズ処理要求到着検査には、シグナルによる割り込みは発生しません）。

このオペランドの指定値は、一時クローズ処理要求を送信したプロセスが応答を待つ時間（180 秒：ipc_socketctl_watchtime オペランドのデフォルト値）よりも小さい値を設定する必要があります。

トランザクション回復要求待ち状態に割り込んで、一時クローズ処理要求受信を検査する機能は、polling_control_data オペランドに Y を指定した場合にだけ動作します。連続してトランザクション回復要求を受け付けたときに一時クローズ処理要求受信を検査する機能は、polling_control_data オペランドの指定に関係なく、動作します。

このオペランドに最大値を指定した場合は、polling_control_data オペランドの指定に関係なく、一時クローズ処理要求が到着していないかどうかの検査を実行しません。

●groups=グループ識別子〔グループ識別子〕

～〈符号なし整数〉((0～4294967294))

このサービスグループのグループアクセスリストを設定します。

OS に登録されているグループ ID を指定してください。このオペランドには 16 個まで指定できます。

このオペランドの指定に関係なく、OpenTP1 管理者のグループ ID は setgid() システムコールによって自動的に設定されます。

最大値は OS に依存するので、使用している OS のマニュアルを参照してください。

●trn_xar_use=Y|N

～《N》

XA リソースサービスを使用するかどうかを指定します。XA リソースサービスを使用する場合、[XA リソースサービス定義](#)で実行環境を定義する必要があります。また、システム共通定義の jnl_fileless_option オペランドに Y を指定している場合は、XA リソースサービスを使用できません。

Y

XA リソースサービスを使用します。

N

XA リソースサービスを使用しません。

●trn_start_recovery_mode=stop|wait|continue

～《stop》

trnstring 定義コマンドで-m オプションを指定したリソースマネージャの OpenTP1 開始処理でのトランザクション回復処理方式を指定します。このオペランドは再開始時に変更または削除できます。

stop

一定時間 (trn_start_recovery_watch_time オペランドの指定値) 以上のリトライ処理で回復処理を行っても、回復処理が完了しない場合、OpenTP1 の開始処理を中止します。このとき、アボートコード「t860004」を出力します。

wait

trnstring 定義コマンドの-m オプションで指定したリソースマネージャの中で、OpenTP1 に関連するすべてのトランザクション回復処理が完了するまで、リトライ処理が実行されます。このリトライ処理を中断するには、dcstop コマンドで OpenTP1 を強制停止してください。

continue

一定時間 (trn_start_recovery_watch_time オペランドの指定値) 以上のリトライ処理で回復処理を行っても、回復処理が完了しない場合、回復処理を中断し、OpenTP1 の開始処理を続行します。

●trn_start_recovery_watch_time=リトライ経過時間

～〈符号なし整数〉((0～65535))《600》(単位：秒)

trnstring 定義コマンドで-m オプションを指定した場合の、オンライン前回復処理が完了するまでのリトライ上限時間を指定します。0 を指定した場合は、リトライをしません。このオペランドは、trn_start_recovery_mode オペランドで wait 以外を指定した場合に有効です。このオペランドは再開始時に変更または削除できます。

●trn_start_recovery_interval=リトライ間隔時間

～〈符号なし整数〉((0～65535))《3》(単位：秒)

trnstring 定義コマンドで-m オプションを指定した場合の、オンライン前回復処理が完了するまでのリトライ間隔時間を指定します。0 を指定した場合は、すぐに次のリトライ処理を行います。

trn_start_recovery_watch_time オペランドで 0 を指定した場合は、このオペランドで指定した値は無効になります。このオペランドは再開始時に変更または削除できます。

●trn_xa_commit_error=down|force

～《down》

複数のリソースをアクセスするトランザクションがコミットに決定したあと、コミットできなくなったりソースマネージャが発生した場合※の処理を指定します。

この現象が発生した場合、該当トランザクションでアクセスしたリソース間の整合性は失われているおそれがあります。

注※

リソースマネージャに対して発行した xa_commit 関数がロールバックした際のコード (XAER_RMERR, または XA_RB から始まるコード) でリターンした場合

down

OpenTP1 システムをダウンさせます。

システムダウン後に、リソースの整合性を回復し、OpenTP1 を強制正常開始してください。

force

リソースの不整合を無視してシステムを続行します。

この場合、コミットを行う関数 (dc_trn_chained_commit, dc_trn_unchained_commit(), tx_commit()) は正常終了します。不整合が発生しても問題がないリソースの場合、またはテスト中の場合などに指定します。

●trn_prf_event_trace_level=TRN イベントトレースの取得レベル

～((00000000～00000007))《00000007》

TRN イベントトレースの取得レベルを指定します。このオペランドは、イベントトレースの取得機能を制御します。このオペランドの指定を省略した場合は、入口情報、正常出口情報、およびエラー出口情報をトレースとして取得します。TRN イベントトレースの詳細については、マニュアル「OpenTP1 解説」のトレース機能の記述を参照してください。

00000000

TRN イベントトレースを取得しません。

00000001

イベントの入口でトレースを取得します (入口情報)。

00000002

イベントの出口でトレースを取得します (正常出口情報)。

trn_prf_event_trace_condition オペランドに、xafunc を指定した場合は、XA 関数のリターン値が 0 以上のときにトレースを取得します。

00000004

イベントの出口で障害が発生したときにトレースを取得します（エラー出口情報）。

00000007

イベントの入口、および出口（障害発生時を含む）でトレースを取得します（入口情報、正常出口情報、およびエラー出口情報）。

これらの指定値は、論理和でも指定できます。例えば、00000003 を指定すると、00000001 と 00000002 の両方を指定したときと同じ動作になります。

取得したトレースをファイル出力または編集出力するには、prfget コマンド、prfed コマンド、または dcalzprf コマンドを使用します。ファイルの出力結果は、\$DCDIR/spool/dctrninf/trace/prf に保存されます。これらのコマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

●trn_prf_event_trace_condition=トレース取得種類 [, トレース取得種類]

～《xafunc》

取得する TRN イベントトレースの種類を指定します。取得したいトレースによって、複数の種類を指定できます。このオペランドの指定を省略した場合は、XA 関数についてのトレースを取得します。

xafunc

XA 関数についてのトレースを取得します。

trnservice

トランザクションサービスの動作状況についてのトレースを取得します。

なお、このオペランドは、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

●trn_completion_limit_time=トランザクション完了限界時間

～〈符号なし整数〉((0～65535))《0》(単位：秒)

トランザクションブランチの開始から終了までの最大実行時間を指定します。指定時間を超えた場合、このトランザクションブランチのプロセスが異常終了したあとに、トランザクションブランチが回復プロセスによってコミットまたはロールバックのどちらかに決着して終了します。0 を指定した場合は、トランザクションブランチの最大実行時間を監視しません。

UAP が異常終了した場合に閉塞するかどうかは hold オペランド、term_watch_time オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

このオペランドの監視対象区間は、dc_trn_begin 関数などの呼び出しやサービス関数の開始によるトランザクションの開始から、トランザクションの同期点処理終了情報 (TJ) の取得後であるトランザクションブランチの終了までです。ただし、トランザクションの最適化が行われた場合、サーバ側のトランザクションブランチに対する監視が終了するのは、クライアント側に応答を返したあとです。このオペランドの監視対象区間の詳細および各種タイマ監視との関係については、「付録 A.2 トランザクションの時間監視」を参照してください。

なお、このオペランドは、ユーザサービス定義または rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

トランザクショナル RPC 実行プロセス (clttrnd) または CUP 実行プロセス (cltcond) で、この機能を有効にするには、クライアントサービス定義の trn_completion_limit_time オペランドを指定する必要があります。

また、子トランザクションブランチを処理しているプロセスが、サービス関数処理以外のタイミングでこのオペランドの指定時間を超えて異常終了した場合、RPC コールの呼び出し元は、応答待ち時間が満了するまで待ち合わせることがあります。

●trn_extend_function=トランザクションサービスの機能の拡張レベル

～ 〈16 進数字〉 ((00000000～00000001)) 《00000000》

トランザクションサービスの機能の拡張レベルを、次の中から指定します。

次の指定値で示す機能の拡張レベルを複数指定する場合、それぞれの指定値の論理和を指定してください。

00000000

トランザクションサービスの機能を拡張しません。

00000001

00000001 は、トランザクションブランチが関連するすべての OpenTP1 ノードのトランザクションサービス定義で指定してください。

トランザクションブランチが関連するすべての OpenTP1 ノードのうち、このオペランドの指定がないノードが一つ以上ある場合、1 相コミットでリソースマネージャから XAER_NOTA が返されたときは、ルートトランザクションブランチのリターン値は変更されません。

1 相コミットで、リソースマネージャから XAER_NOTA が返された場合の関数のリターン値を次の表に示します。

関数	リターン値	
	trn_extend_function=0000 0000 の場合	trn_extend_function=00000001 の場合
dc_trn_chained_commit (CBLDCTRN('C-COMMIT'))	DC_OK (00000)	DCTRNER_HAZARD (00904), DCTRNER_HAZARD_NO_BEGIN (00927)
dc_trn_unchained_commit (CBLDCTRN('U-COMMIT'))	DC_OK (00000)	DCTRNER_HAZARD (00904)
tx_commit (TXCOMMIT)	TX_OK (TX-OK)	TX_HAZARD (TX-HAZARD), TX_HAZARD_NO_BEGIN (TX-HAZARD-NO-BEGIN)
dc_mcf_commit (CBLDCMCF('COMMIT'))	DCMCFRTN_00000 (00000)	DCMCFRTN_HAZARD (70908)

注

括弧内の英数字は、COBOL 言語を使用した場合の関数名およびステータスコードです。

●trn_rcv_open_close_scope=Y|N

～《N》

複数のリソースマネージャにアクセスしていたトランザクションの回復処理で、XA 関数 (xa_open(), xa_commit(), xa_rollback(), xa_close()) の発行順序を変更するかどうかを指定します。発行順序の変更は、OpenTP1 システムに登録された OpenTP1 が提供するリソースマネージャを除く、すべてのリソースマネージャに対して行われます。一部のリソースマネージャの回復処理で発行される xa_open() がエラーとなり、回復処理が正常に実行されない場合に指定します。

Y

XA 関数の発行順序を変更します。

N

XA 関数の発行順序を変更しません。

このオペランドの指定による、XA 関数の発行順序の例を示します。

例：

次の条件を満たす例で説明します。

- 同一ノード内の SUP でトランザクションを開始し、SUP から SPP に RPC コールでサービスを要求し、同一リソースマネージャに拡張子 (-i) を使用してアクセスするグローバルトランザクションを構成した。

- 次のように定義している。

トランザクションサービス定義：

```
trnstring -n TEST_RM -i _A
```

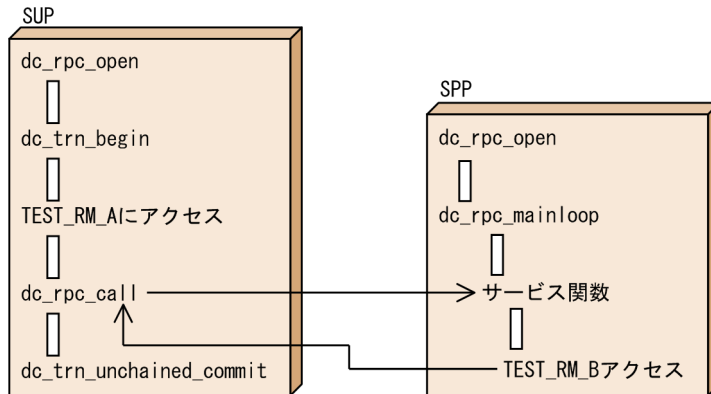
```
trnstring -n TEST_RM -i _B
```

ユーザーサービス定義 (SUP) : trnrmid -n TEST_RM -i _A

ユーザーサービス定義 (SPP) : trnrmid -n TEST_RM -i _B

この例での、グローバルトランザクションの回復処理を次に示します。

図 3-6 グローバルトランザクションの回復処理の例



この例の場合の、オペランドの指定による XA 関数の発行順序の違いを次に示します。

trn_rcv_open_close_scope オペランドを省略、または N を指定

- ・ トランザクション回復プロセス (trnrvd)
 1. xa_open(TEST_RM_A)
 2. xa_open(TEST_RM_B)
 3. xa_commit(TEST_RM_A)または xa_rollback(TEST_RM_A)
 4. xa_commit(TEST_RM_B)または xa_rollback(TEST_RM_B)
 5. xa_close(TEST_RM_A)
 6. xa_close(TEST_RM_B)

trnstring 定義コマンドで-s オプションを省略した場合も、発行順序は同様になります。

trn_rcv_open_close_scope オペランドに Y を指定

- ・ トランザクション回復プロセス (trnrvd)
 1. xa_open(TEST_RM_A)
 2. xa_commit(TEST_RM_A)または xa_rollback(TEST_RM_A)
 3. xa_close(TEST_RM_A)
 4. xa_open(TEST_RM_B)
 5. xa_commit(TEST_RM_B)または xa_rollback(TEST_RM_B)
 6. xa_close(TEST_RM_B)

trnstring 定義コマンドに-s オプションを指定した場合も、発行順序は同様になります。

複数のリソースマネージャが登録されている状態で、特定のリソースマネージャにだけこの機能を使用したい場合は、trnstring 定義コマンドの-s オプションを使用してください。詳細については、[trnstring 定義コマンド](#)の説明を参照してください。

●watch_time=最大応答待ち時間

～〈符号なし整数〉((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しない場合があります。

省略した場合は、システム共通定義の watch_time オペランドの値を仮定します。

このオペランドは、システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

次ページ以降に記述しています。

putenv 形式のオペランド

●環境変数名 環境変数値

～〈文字列〉

XA インタフェースでリソースマネージャと連携する場合に、トランザクション回復サービスおよびリソースマネージャ監視サービスがリソースマネージャにアクセスするために必要な環境変数を指定します。

指定する必要がある環境変数名と環境変数値はリソースマネージャによって異なります。

なお、OpenTP1 では、'dc'で始まる環境変数名は使用しないでください。

dcputenv 形式のオペランド

●環境変数名 環境変数値

～〈文字列〉

XA インタフェースでリソースマネージャと連携する場合に、トランザクション回復サービスおよびリソースマネージャ監視サービスがリソースマネージャにアクセスするために必要な環境変数を指定します。環境変数を指定した場合、その値も取得します。

指定する必要がある環境変数名と環境変数値はリソースマネージャによって異なります。

なお、OpenTP1 では、'dc'で始まる環境変数名は使用しないでください。

trnstring

名称

リソースマネージャのアクセス用文字列または拡張子の指定

形式

```
trnstring  -n リソースマネージャ名  
           [-i リソースマネージャ拡張子 [, リソースマネージャ拡張子] ...]  
           [-o "トランザクションサービス用xa_open関数用文字列"]  
           [-c "トランザクションサービス用xa_close関数用文字列"]  
           [-O "ユーザサーバ用xa_open関数用文字列"]  
           [-C "ユーザサーバ用xa_close関数用文字列"] [-d]  
           [-e] [-m] [-r] [-s]
```

機能

リソースマネージャにアクセスするための文字列または拡張子を定義します。

OpenTP1 のトランザクションサービスでは、X/Open の XA インタフェースに準拠し、リソースマネージャと連携してトランザクションを実行します。XA インタフェースでは、リソースマネージャが規定した xa_open 関数用文字列、および xa_close 関数用文字列を通知する必要があります。

ユーザサーバが、OpenTP1 下のトランザクション内で、OpenTP1 以外が提供するリソースマネージャにアクセスする場合（トランザクション制御用オブジェクトファイルをユーザサーバにリンケージした場合）、そのリソースマネージャの 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、および xa_close 関数は、このコマンドで指定された文字列を引数として、ユーザサーバ起動、および終了時にトランザクションサービスが発行します。

トランザクションサービスは、ユーザサーバ異常終了時、または OpenTP1 再開始時に、ユーザサーバが行っていたトランザクションを回復します。その際 xa_open 関数および xa_close 関数を使用するため、トランザクションサービスには、ユーザサーバがアクセスする、すべてのリソースマネージャの xa_open 関数用、xa_close 関数用文字列を定義する必要があります。

-i オプションは、一つのリソースマネージャに複数の拡張子を割り当てて、複数の制御単位としてアクセスする場合にだけ指定してください。OpenTP1 は、このオプションを指定したリソースマネージャに対して、リソースマネージャ名を「リソースマネージャ名+リソースマネージャ拡張子」に修飾して管理します。そのため、リソースマネージャ名だけではアクセスできません。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で trnrmid 定義コマンドを指定する必要があります。

-i オプションに指定するリソースマネージャを識別するための文字列は、すべての trnstring 定義コマンドで重複しないように指定してください。一つの定義ファイル内に同一リソースマネージャ名で、同一リソースマネージャ拡張子の trnstring 定義コマンドを複数指定した場合、最後に指定した定義が有効となります。

-o, -c オプションには、トランザクションサービスプロセスから xa_open 関数, xa_close 関数を発行する場合に使用する文字列を指定してください。-O, -C オプションには、ユーザーバプロセスから xa_open 関数, xa_close 関数を発行する場合に使用する文字列を指定してください。

-e オプションは、静的リソースマネージャに対するトランザクションブランチ開始時に xa_start 関数でエラーが発生した場合、リソースマネージャに対して xa_close 関数, xa_open 関数, xa_start 関数の順にリトライ処理を行うときに指定します。なお、-e オプションを指定しない場合、xa_open 関数, xa_start 関数の順にリトライ処理を行います。

指定する文字列は、各リソースマネージャのマニュアルを参照してください。

NULL 文字列を通知する必要がある場合は、-o, -c, -O, または -C オプションを省略してください。

次の両方の条件を満たす場合は、このコマンドを定義する必要はありません。

- 一つのリソースマネージャに対して、複数のリソースマネージャ拡張子を割り当てて、複数の制御単位としてアクセスする必要がない場合
- xa_open 関数用文字列と xa_close 関数用文字列の両方が NULL 文字列を通知する必要がある場合

必要な文字列を定義しなかった場合、または不当な文字列を指定した場合は、ユーザーサーバの起動、終了、またはトランザクションの回復がエラーとなります。

オプション

●-n リソースマネージャ名

～ 〈1～31 文字の識別子〉

OpenTP1 下のトランザクションでアクセスするリソースマネージャ名を指定します。

OpenTP1 が提供するリソースの場合、OpenTP1_MCF だけ指定が必要です。ただし、OpenTP1_MCF でも次の場合には指定を省略できます。

- MCF マネージャ環境定義 mcfmenv の id オペランドを省略した場合、または A を指定した場合

●-i リソースマネージャ拡張子

～ 〈1～2 文字の識別子〉

一つのリソースマネージャを複数の制御単位としてアクセスする場合、リソースマネージャ名を「リソースマネージャ名+リソースマネージャ拡張子」に修飾するための文字列を指定します。

複数の拡張子が同じ xa_open 関数用文字列、または xa_close 関数用文字列を使用する場合は、このオプションに複数の拡張子を指定できます。その場合は、拡張子と拡張子の間をコンマ', 'で区切ります。

すべての trnstring 定義コマンドで指定した拡張子は重複しないようにしてください。

トランザクションに関係する MCF を管理するために、MCF に対してのリソースマネージャ拡張子を指定しなければなりません。このオプションに指定する MCF のリソースマネージャ拡張子は、ユーザサーバがアクセスできるプロセス識別子と、該当するノードで動作する MCF のマネージャ定義で指定した MCF マネージャプロセス識別子をすべて指定してください。

●-o "トランザクションサービス用 xa_open 関数用文字列"

～ 〈1～256 の文字列〉

OpenTP1 のトランザクションサービスプロセスから xa_open 関数を発行する場合に使用する文字列を指定します。

文字列に指定可能な文字として「ASCII 印字可能文字（ASCII コード 32～126）」を使用できます。

ただし、「"（ASCII コード 33）」は引用符以外には使用できません。

指定する文字列は、各リソースマネージャのマニュアルを参照してください。

NULL 文字列を通知する必要がある場合は、このオプションを省略してください。

●-c "トランザクションサービス用 xa_close 関数用文字列"

～ 〈1～256 の文字列〉

OpenTP1 のトランザクションサービスプロセスから xa_close 関数を発行する場合に使用する文字列を指定します。

文字列に指定可能な文字として「ASCII 印字可能文字（ASCII コード 32～126）」を使用できます。

ただし、「"（ASCII コード 33）」は引用符以外には使用できません。

指定する文字列は、各リソースマネージャのマニュアルを参照してください。

NULL 文字列を通知する必要がある場合は、このオプションを省略してください。

●-O "ユーザサーバ用 xa_open 関数用文字列"

～ 〈1～256 の文字列〉

ユーザサーバプロセスから xa_open 関数を発行する場合に使用する文字列を指定します。

文字列に指定可能な文字として「ASCII 印字可能文字（ASCII コード 32～126）」を使用できます。

ただし、「"（ASCII コード 33）」は引用符以外には使用できません。

指定する文字列は、各リソースマネージャのマニュアルを参照してください。

NULL 文字列を通知する必要がある場合は、このオプションを省略してください。

●-C "ユーザーバ用 xa_close 関数用文字列"

～ 〈1～256 の文字列〉

ユーザーバプロセスから xa_close 関数を発行する場合に使用する文字列を指定します。

文字列に指定可能な文字として「ASCII 印字可能文字 (ASCII コード 32～126)」を使用できます。

ただし、「" (ASCII コード 33)」は引用符以外には使用できません。

指定する文字列は、各リソースマネージャのマニュアルを参照してください。

NULL 文字列を通知する必要がある場合は、このオプションを省略してください。

●-d

このリソースマネージャが、コミット最適化およびプリペア最適化に対応していない場合に、必ず指定します。コミット最適化およびプリペア最適化に対応していないリソースマネージャでこの指定を省略すると、トランザクションが決着できなかったり、リソースマネージャがダウンしたりすることがあります。

このオプションを指定したリソースマネージャがトランザクションに参加した場合、トランザクションブランチはコミット最適化およびプリペア最適化をしません。

●-e

-n オプションで指定したリソースマネージャに対するトランザクションブランチ開始時に xa_start 関数でエラーが発生した場合、リソースマネージャに対して xa_close 関数、xa_open 関数、xa_start 関数の順にリトライ処理を行うときに指定します。このオプションを指定しない場合、xa_start 関数でエラーが発生したとき、xa_open 関数、xa_start 関数の順にリトライ処理を行います。

●-m

このリソースマネージャ内の未決着トランザクションの回復処理を、OpenTP1 がオンラインになる前に行う場合に指定します。

このオプションを指定すると、OpenTP1 開始時にリソースマネージャ監視サービスがリソースマネージャに対して xa_recover 関数を発行し、取得した未決着トランザクション情報を基に、trn_start_recovery_mode、trn_start_recovery_watch_time および trn_start_recovery_interval オペランドに従ってトランザクション回復処理を行います。

●-r

OpenTP1 の開始処理、またはオンライン処理で、リソースマネージャから未決着トランザクション情報の通知があるまで、トランザクションの回復処理を待ち合わせるときに指定します。トランザクション回復プロセスからリソースマネージャに対して発行する xa_recover 関数が未決着トランザクション情報を取得するまで、トランザクションの回復処理は行われません。このオプションを指定したときは、指定したリソースマネージャの数分のトランザクション並行回復プロセスを確保してください (トランザクション並行回復プロセス数は、trn_recovery_process_count オペランドで指定します)。このオプションは再開時に変更または削除できます。

なお、このオプションを有効にするには、オープン文字列※を指定する必要があります。

注※

Oracle9i までは、OPS_FAILOVER=T をオープン文字列に指定してください。

Oracle 10g 以降は、RAC_FAILOVER=T をオープン文字列に指定してください。

●-S

特定のリソースマネージャにアクセスしていたトランザクションの回復処理で、XA 関数 (xa_open(), xa_commit(), xa_rollback(), xa_close()) の発行順序を変更するかどうかを指定します。発行順序の変更は、OpenTP1 システムに登録された OpenTP1 が提供するリソースマネージャを除く、特定のリソースマネージャに対してだけ行われます。一部のリソースマネージャの回復処理で発行される xa_open() がエラーとなり、回復処理が正常に実行されない場合に指定します。

OpenTP1 システムに登録されているすべてのリソースマネージャ (OpenTP1 が提供するリソースマネージャは対象外) にこの機能を使用したい場合は、trn_rcv_open_close_scope オペランドに Y を指定してください。オペランドの詳細、およびオプションの指定による XA 関数の発行順序の例については、[trn_rcv_open_close_scope オペランド](#)の説明を参照してください。

XA リソースサービス定義

形式

set 形式

```
[set xar_eventtrace_level=ERR|INF|ALL]
[set xar_eventtrace_record=XAR イベントトレース情報ファイルの
    最大出力レコード数]
[set xar_session_time=アイドル状態のトランザクションブランチの監視時間]
[set xar_msdtc_use=Y|N]
[set xar_prf_trace_level=XAR 性能検証用トレース情報の取得レベル]
```

コマンド形式

```
xarfile -t online|backup
        -a 物理ファイル名
```

機能

XA リソースサービスを使用してトランザクション管理をするための実行環境を定義します。

説明

set 形式のオペランド

●xar_eventtrace_level=ERR|INF|ALL

～《ERR》

XAR イベントトレース情報の出力レベルを指定します。

ERR

エラー出口情報を XAR イベントトレースに出力します。

INF

エラー情報出口および正常出口情報を XAR イベントトレースに出力します。

ALL

エラー出口情報、正常出口情報およびその他の情報を含めた、すべての XAR イベントトレース情報を XAR イベントトレースに出力します。

本番環境では、ERR を指定することをお勧めします。ERR 以外を指定した場合、I/O 回数が増加するため、処理速度が低下します。

●xar_eventtrace_record=XAR イベントトレース情報ファイルの最大出力レコード数

～〈符号なし整数〉((1～65535))《8192》

XAR イベントトレース情報ファイルの最大出力レコード数を指定します。

●**xar_session_time=アイドル状態のトランザクションブランチの監視時間**

～ 〈符号なし整数〉 ((10～65535)) 《180》 (単位：秒)

End()から Prepare()までのアイドル状態のトランザクションブランチを監視する時間を指定します。指定された時間を超えた場合は、メッセージを出力してトランザクションブランチをロールバックします。アイドル状態を監視するインタバル時間は、10 秒です。そのため、検知するのに時間が掛かる場合があります。

●**xar_msdtc_use=Y|N**

～ 《N》

MSDTC 連携機能を使用するかどうかを指定します。

Y

MSDTC 連携機能を使用します。

N

MSDTC 連携機能を使用しません。

●**xar_prf_trace_level=XAR 性能検証用トレース情報の取得レベル**

～((00000000～00000003)) 《00000003》

XAR 性能検証用トレースの取得レベルを指定します。複数の取得レベルを指定したい場合は、それぞれの指定値の論理和を指定してください。

XAR 性能検証用トレースの詳細については、マニュアル「OpenTP1 解説」の障害の原因解析の説明を参照してください。

00000000

XAR 性能検証用トレースを取得しません。

00000001

アプリケーションサーバからのトランザクション要求の入口、および出口でトレース（イベント ID：0x4a00～0x4a0f）を取得します。

00000002

OpenTP1 のトランザクション処理の入口、および出口でトレース（イベント ID：0x4b00～0x4b0f）を取得します。

取得したトレースをファイル出力または編集出力するには、prfget コマンド、prfed コマンド、または dcalzprf コマンドを使用します。ファイルの出力結果は、\$DCDIR/spool/dcxarinf に保存されます。これらのコマンドの詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

コマンド形式

次ページ以降に記述しています。

xarfile

名称

XAR ファイルの指定

形式

```
xarfile  -t  online|backup  
          -a  物理XARファイル名
```

機能

XA リソースサービスで使用する物理 XAR ファイル名を指定します。

物理 XAR ファイルには、オンライン用 XAR ファイルとバックアップ用 XAR ファイルの二つがあり、両方指定する必要があります。

オンライン用 XAR ファイルとバックアップ用 XAR ファイルを、それぞれ複数指定することはできません。

オンライン用 XAR ファイルとバックアップ用 XAR ファイルに指定するファイルは、同一のレコード数、同一のレコード長である必要があります。また、レコード数は、トランザクションサービス定義の `trn_tran_process_count` オペランドで指定した値よりも大きいファイルを指定する必要があります。

オプション

●-t online|backup

XAR ファイルのタイプを指定します。

online

オンライン用 XAR ファイルとして使用します。

backup

バックアップ用 XAR ファイルとして使用します。

●-a 物理 XAR ファイル名

～ 〈1～63 文字のパス名〉

物理 XAR ファイル名を指定します。

インタバルサービス定義

形式

set 形式

```
[set watch_time=最大応答待ち時間]
```

コマンド形式

なし。

機能

各システムサーバへ提供するインタバルタイマ機能の実行環境を定義します。

説明

set 形式のオペランド

●watch_time=最大応答待ち時間

～〈符号なし整数〉((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しない場合があります。

省略した場合は、システム共通定義の watch_time オペランドの値が仮定されます。

このオペランドは、システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

なし。

ステータスサービス定義

形式

set 形式

```
set sts_file_name_1="論理ファイル名","A系ステータスファイル名"  
                  ,"B系ステータスファイル名"  
[set sts_file_name_2="論理ファイル名","A系ステータスファイル名"  
                  ,"B系ステータスファイル名"]  
[set sts_file_name_3="論理ファイル名","A系ステータスファイル名"  
                  ,"B系ステータスファイル名"]  
[set sts_file_name_4="論理ファイル名","A系ステータスファイル名"  
                  ,"B系ステータスファイル名"]  
[set sts_file_name_5="論理ファイル名","A系ステータスファイル名"  
                  ,"B系ステータスファイル名"]  
[set sts_file_name_6="論理ファイル名","A系ステータスファイル名"  
                  ,"B系ステータスファイル名"]  
[set sts_file_name_7="論理ファイル名","A系ステータスファイル名"  
                  ,"B系ステータスファイル名"]  
[set sts_initial_error_switch=stop|continue|excontinue]  
[set sts_single_operation_switch=stop|continue]  
[set sts_last_active_file="論理ファイル名"]  
[set sts_last_active_side=A|B]  
[set watch_time=最大応答待ち時間]
```

コマンド形式

なし。

機能

OpenTP1 システム環境下で動作する各システムサービスの状態管理をするための実行環境を定義します。一つのノードに一つのステータスサービスがあります。

ステータスサービス定義ではステータスファイルの名称を最大7個指定できます。

OpenTP1 システムの正常開始時に、ステータスサービス定義で最初に指定したステータスファイルが現用になります。残りのステータスファイルは予備になります。

再開始時には、前回の現用ファイルを引き継ぎます。

ステータスファイルは、信頼性を高めるために A 系と B 系の二重で管理します。

現用のステータスファイルには、A 系と B 系で同じ内容が書き込まれます。

このため、ステータスファイルの実体は、すべて同じレコード長のファイルを指定します。また、一つの論理ファイルごとに同じ容量のファイルを指定します。

A 系または B 系の物理ファイルは、複数のディスクに分散させて作成してください。一つのディスク上に同じ系のすべての物理ファイルを作成すると、ディスクの全面障害が発生した場合、その系は使用できま

せん。例えば、一つのディスク上に同じ系の物理ファイルを作成した場合、ディスク障害発生時に A 系、B 系の二重書きできる世代がなくなるため、OpenTP1 を緊急停止させるか、または片系運転状態となり危険です。

OpenTP1 システムの正常開始時、再開時またはオンライン中にステータスファイル障害が発生した場合、OpenTP1 システムを停止させないために、ステータスサービスがステータスファイルをスワップします。

ステータスファイルをスワップするには、現用ファイルのほかに、予備用のステータス論理ファイルを定義する必要があります。

説明

set 形式のオペランド

●**sts_file_name_1～sts_file_name_7**="論理ファイル名", "A 系ステータスファイル名", "B 系ステータスファイル名"

最大 7 個のステータスファイルを指定できます。

1 個を現用ファイルとして、残り 6 個を予備ファイルとして作成します。

それぞれのステータスファイルでは、信頼性を高めるために、A 系ステータスファイルと B 系ステータスファイルの二つの同一ファイルを作成します。

なお、sts_file_name_1 から sts_file_name_7 に指定する論理ファイル名、A 系ステータスファイル名、および B 系ステータスファイル名は重複して指定できません。

"論理ファイル名" ～ 〈1～8 文字の識別子〉

ステータスファイルの論理ファイル名を指定します。

"A 系ステータスファイル名" ～ 〈パス名〉

論理ファイルを構成する A 系のステータスファイル名を絶対パス名で指定します。

"B 系ステータスファイル名" ～ 〈パス名〉

論理ファイルを構成する B 系のステータスファイル名を絶対パス名で指定します。

なお、この B 系ステータスファイルには、A 系ステータスファイルと同じレコード長、およびレコード数のファイルを指定します。

A 系ステータスファイル名、および B 系ステータスファイル名の絶対パス名には、環境変数は使用できません。

また、ステータスファイルには、絶対パス名が異なっても、同じファイル名を指定できません。

●**sts_initial_error_switch=stop|continue|excontinue**

～ 《stop》

ステータスサービス開始時に、定義で指定したステータスファイルに次に示す異常を検知した場合の、ステータスサービスの処置を指定します。

- ステータスファイルの実体がありません。
- ステータスファイル障害を検知しました。

stop

OpenTP1 システムの再開始を保証するため、ステータスサービスを開始しないで OpenTP1 システムを停止させる場合に指定します。

stop を指定した場合、次の処理をします。

- 定義に指定したステータスファイルのうち、一つでも異常を検知すると、OpenTP1 システムは停止します。異常が発生したステータスファイルの対策を実施したあと、OpenTP1 システムを再起動してください。
- 前回のオンラインで現用だったステータス論理ファイルの両系に異常が発生した場合は、OpenTP1 システムの再開始はできません。

continue

定義に指定したステータス論理ファイルに異常が発生しても、ステータスサービスを開始させる場合に指定します。

continue を指定した場合、次の処理をします。

- 前回のオンラインで現用だったステータス論理ファイルの片系に異常が発生すると、正常な系のファイルの内容を予備の A 系、B 系ファイルに複写します。その後、予備ファイルを現用に切り替えて、ステータスサービスを開始します。複写できる予備ファイルがない場合、sts_single_operation_switch オペランドの指定に従い、ステータスサービスを停止するか、継続するかを決定します。
- ステータスサービスの開始時、OpenTP1 システムが現用に選択したステータスファイルが、前回のオンラインの最新現用ファイルかどうか確定できない場合、sts_last_active_file オペランドに指定された論理ファイル名と比較します。
ファイル名が一致した場合：ステータスサービスを開始します。
ファイル名が一致しない場合：ステータスサービスを停止します。
なお、sts_last_active_file オペランドの指定がなく、最新の現用ファイルが確定できない場合は、ステータスサービスを開始しません。
最新の現用ファイルかどうか確定できない場合とは、定義した論理ファイルが、ステータスサービスの開始時に、一つでも次の表に示す状態になった場合のことです。

表 3-4 現用ファイルが確定できないときの論理ファイルの状態 (continue の場合)

項番	A 系の状態	B 系の状態
1	BLOCKADE	BLOCKADE
2	BLOCKADE	OPEN (INIT)
3	BLOCKADE	NONE

項番	A 系の状態	B 系の状態
4	OPEN (INIT)	BLOCKADE
5	OPEN (INIT)	NONE
6	NONE	OPEN (INIT)
7	NONE	BLOCKADE
8	NONE	NONE

3. A 系、B 系ともに正常なステータス論理ファイルが一つもない場合、ステータスサービスは開始しないで、OpenTP1 システムは停止します。

excontinue

このオペランドで continue を指定した場合と同じ処理をします。ただし、sts_single_operation_switch オペランドで stop を指定、または省略した場合、片方の系が異常で、他方の系が初期化状態である論理ファイルについては現用決定対象ファイルから除き、現用ファイル決定処理をします。

最新の現用ファイルかどうか確定できない場合とは、定義した論理ファイルが、ステータスサービスの開始時に、一つでも次の表に示す状態になった場合のことです。

表 3-5 現用ファイルが確定できないときの論理ファイルの状態 (excontinue の場合)

項番	A 系の状態	B 系の状態
1	BLOCKADE	BLOCKADE
2	BLOCKADE	NONE
3	NONE	BLOCKADE
4	NONE	NONE

OpenTP1 開始前に前回のオンライン時に現用だった論理ファイルの片系を stsinit コマンドで初期化した場合、現用ファイルの決定が正しくできなくなるので、注意してください。

なお、異常の発生したステータスファイルの対策については、マニュアル「OpenTP1 運用と操作」を参照してください。

●sts_single_operation_switch=stop|continue

～《stop》

ステータスサービス開始中または開始後に、現用ステータスファイルの片系に入出力障害が発生し、かつスワップできる両系そろった論理ファイルが存在しない場合のステータスサービスの処置を指定します。

stop

ステータスサービスを停止する場合に指定します。

continue

現用ステータスファイルの正常な系だけで、ステータスサービスを継続する場合に指定します。

片系運転状態になると、KFCA01044-I が出力されます。片系運転中に次のような障害が発生すると、OpenTP1 を再開できなくなります。そのため、片系運転状態になった場合は、すぐに障害について対策してください。

- 片系運転中に、正常な系に障害が発生した場合
- 片系運転中で、ファイルを更新するときに、システムダウンが発生した場合

●sts_last_active_file="論理ファイル名"

～ 〈1～8 文字の識別子〉

前回までのオンラインで最新の現用ファイルだったものを論理ファイル名で指定します。このオペランドは、sts_initial_error_switch オペランドに continue または excontinue を指定したときだけ指定できます。また、OpenTP1 システムが選択した現用ファイルが前回までのオンラインで最新の現用ファイルかどうか確定できない場合に有効となります。

このオペランドは、次のように指定してください。

- 現用ファイルが変更されるたびにオペランドを修正できる場合
全ステータスファイルの初期化直後に OpenTP1 システムを開始したときは、最も小さい番号の sts_file_name オペランドに指定した論理ファイル名を指定してください。
そのあと、現用ファイルがスワップによって変更された場合は、この定義を新しい現用ファイル名に変更してください。
- 現用ファイルが変更されてもオペランドを修正できない場合
システムを開始したときに、KFCA01011-I および KFCA01005-E（理由コード=0000000015 または 0000000016）が出力され、システム起動エラーとなったときは、前回のオンラインの最新現用ファイルをログファイルで調査します。
KFCA01011-I で出力された論理ファイル名と一致するときは、その論理ファイル名をこのオペランドに指定し、OpenTP1 を再開してください。

●sts_last_active_side=A|B

前回のオンライン時、現用ステータスファイルの片系に障害が発生して閉塞状態になった場合、正常だった系を指定します。

この指定は、OpenTP1 再開時に、前回のオンライン中に障害が発生した系を現用を選択して誤動作するのを防ぐために使用します。

片系に障害が発生して閉塞状態になった場合は、必ず指定してください。

この指定は、sts_single_operation_switch オペランドに continue を指定したときだけ有効です。

●watch_time=最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しない場合があります。

省略した場合は、システム共通定義の `watch_time` オペランドの値を仮定します。

このオペランドは、システム共通定義の `watch_time` オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の `watch_time` オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

なし。

ジャーナルサービス定義

形式

set 形式

```
[set jnl_tran_optimum_level=ジャーナルの出力方式に対する  
最適化レベル]  
[set jnl_arc_terminate_timeout=待ち合わせ最大時間]  
[set max_socket_descriptors=ソケット用ファイル記述子の最大数]  
[set jnl_arc_ipc_buff_size=TCP/IPの送受信バッファサイズ]  
[set jnl_watch_time=ジャーナルサービスの通信応答待ち時間]  
[set jnl_message_id_change_level=メッセージIDの変更レベル]  
[set watch_time=最大応答待ち時間]
```

コマンド形式

```
jnldfs -r システムジャーナルサービス定義のファイル名  
-c チェックポイントダンプサービス定義のファイル名  
[ [, チェックポイントダンプサービス定義のファイル名] ...]
```

機能

システムジャーナルサービスとチェックポイントダンプサービスを使用するための実行環境を定義します。

なお、システム再開始時に定義の変更はできません。

説明

set 形式のオペランド

●jnl_tran_optimum_level=ジャーナルの出力方式に対する最適化レベル

～ 〈符号なし整数〉 ((2～3)) 《2》

トランザクション実行時に、ジャーナルファイルに取得するトランザクションの回復に必要なジャーナルの出力方式に対する最適化レベルを指定します。レベル 3 は、業務形態に注意して指定してください。

2

グローバルトランザクション単位の最適化

次の場合にジャーナルを同期出力します。次に示すジャーナル以外は、非同期出力します。

- ・ グローバルトランザクション内のルートトランザクションブランチで PJ を出力する場合
- ・ 他ノードに分岐したトランザクションブランチで PJ と HJ を出力する場合

これらのジャーナル以外を非同期出力すると、トランザクション実行時のジャーナル入出力数が減り、性能が上がります。

なお、システム異常終了後の再開始処理時、非同期出力でファイルに出力されなかったジャーナルは、トランザクションを回復するとき、システムが再度出力します。

条件付きジャーナル非同期出力

実行中のトランザクションが次の条件を含んでいない場合、すべてのジャーナルを非同期出力します。これによって、トランザクション実行時のジャーナル入出力がなくなるので、性能が上がります。

なお、実行中のトランザクションが次の場合は、レベル 2 でジャーナルを取得します。

- DAM を使用するグローバルトランザクションで、デフォード更新機能を使っていない DAM ファイルを更新した場合
- 複数ノードにわたるグローバルトランザクションを実行した場合
- ISAM を使用するグローバルトランザクションで更新した場合
- MCF を使用するグローバルトランザクションで更新した場合

非同期出力を要求されたトランザクションジャーナルは、ジャーナルファイルに出力されます。出力される時期を次に示します。

- トランザクション内でアクセスしたリソースマネージャ (DAM, TAM, MCF) がファイルを実更新するとき (一定間隔で行われます)
- チェックポイントダンプ取得時
- ジャーナルバッファがいっぱいになったとき

ただし、システム異常終了後の再開始時、ジャーナルファイルにジャーナルが出力されなかったトランザクションはロールバックします。システム異常終了前にトランザクションが正常 (コミット完了済み) に終了していてもロールバックします。トランザクションのコミット API が正常リターンしたことを期待して、ユーザ処理する UAP を実行すると、その実行中に OpenTP1 が異常終了した場合、トランザクションとユーザ処理の状態不一致が発生することがあります。そのため、システム再開始後にロールバックしては困る場合は、レベル 3 を指定しないようにする必要があります。

●jnl_arc_terminate_timeout=待ち合わせ最大時間

～ 〈符号なし整数〉 ((0～3600)) 《0》 (単位：秒)

ジャーナルサービスを終了させてから、アーカイブサービスが停止するまで、最大待ち時間を秒単位で指定します。指定時間が過ぎても使用中のアーカイブサービスがある場合は、サービスは中断されます。また、ジャーナルサービスも終了します。

0 を指定した場合は、被アーカイブノードの終了処理が完了するまで、アーカイブサービスとジャーナルサービスは終了されません。

●max_socket_descriptors=ソケット用ファイル記述子の最大数

～ 〈符号なし整数〉 ((32～2032))

OpenTP1 制御下のプロセス※でのソケット用に使用するファイル記述子の最大数を指定します。

OpenTP1 制御下のプロセス※は、システムサーバやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数、および通信する他ノードの数によって、このオペランドを変更する必要があります。

注※

MCF サービス（MCF マネージャサービス、MCF 通信サービス、およびアプリケーション起動サービス）以外の OpenTP1 プロセスが対象です。MCF サービスの場合は、「[システムサービス情報定義](#)」および「[システムサービス共通情報定義](#)」を参照してください。

ソケット用ファイル記述子の最大数の算出式を、次に示します。

$\uparrow (\text{自ノード内 UAP プロセス数}^{\ast 1} + \text{ジャーナルサービスに対してサービスを要求してくるノード数}^{\ast 2} + \text{システムサービスプロセス数}^{\ast 3}) / 0.8 \uparrow$

↑↑：小数点以下を切り上げます。

注※1

自ノード内 UAP プロセス数は、次に示す値の合計です。

- 自 OpenTP1 内の UAP プロセス数
- CUP から同時に起動するトランザクション数（クライアントサービス定義の parallel_count オペランドの指定値）

注※2

ジャーナルサービスに対してサービスを要求してくるノード数は、ジャーナルのアーカイブ機能を使用する場合だけ加算してください。このノード数は、次に示す値の合計です。

- 自 OpenTP1 の all_node オペランドに指定したノード名の数
- 自ノード名を、OpenTP1 の all_node オペランドに指定している、他ノードの数

注※3

システムサービスプロセス数とは、自 OpenTP1 内のシステムサービスプロセスの数です。自 OpenTP1 内のシステムサービスプロセスは、rpcstat コマンドで表示されるサーバ名をカウントすることで求められます。rpcstat コマンドで表示されるサーバ名のうち、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造に記載されているシステムサービスプロセスをカウントしてください。

このオペランドの指定が小さいと、OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため、プロセスが KFCA00307-E メッセージを出力して異常終了します。

指定値の優先順位は次のとおりです（1.> 2.）。

1. ジャーナルサービス定義
2. システム共通定義

ここで指定を省略した場合、システム共通定義の値を仮定します。

●jnl_arc_ipc_buff_size=TCP/IP の送受信バッファサイズ

～ 〈符号なし整数〉 ((8192～1048576)) 《61440》 (単位：バイト)

アーカイブ機能を使用して、ジャーナルをアーカイブノードに転送するジャーナル転送プロセスが使用する TCP/IP の送受信バッファのサイズを指定します。

高速な通信媒体を使用する場合、値を大きくすることで性能が向上できます。このオペランドの値には、各 OS で指定できる TCP/IP のバッファサイズの上限值以下の値を指定してください。また、このオペランドの値は、アーカイブノード側のグローバルアーカイブジャーナルサービス定義の jnl_arc_ipc_buff_size オペランドの指定値と合わせてください。

●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)

ジャーナルサービスが RPC によってプロセス間で通信する場合の、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。指定時間を過ぎても応答がない場合、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。この場合、OpenTP1 が終了しないことがあります。

特別なチューニングが不要な場合は、このオペランドの内容を変更しないでください。

jnl_watch_time 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

注※

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

●jnl_message_id_change_level=メッセージ ID の変更レベル

～ 〈符号なし整数〉 ((0～1)) 《0》

ジャーナルサービスが出力するメッセージで、重要な情報となるメッセージの種別を変更します。

メッセージログを監視し、メッセージの種類によって管理方法を区別したい場合にこのオペランドを使用してください。

このオペランドによって変更されるのはメッセージ ID だけであり、メッセージの内容は変更されません。

- 0
- メッセージ ID を変更しないで、従来どおりのメッセージ ID で出力します。
- 1
- メッセージ ID を変更し、出力します。

このオペランドの指定値と出力されるメッセージの関係を次に示します。

jnl_message_id_change_level オペランドの指定値	0	1
出力されるメッセージ ID	KFCA01224-I	KFCA01230-W
	KFCA01250-I	KFCA01231-W

●watch_time=最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しない場合があります。

省略した場合は、システム共通定義の watch_time オペランドの値を仮定します。

このオペランドは、システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

次ページ以降に記述しています。

名称

ジャーナル関係のファイルの指定

形式

```
jnldfs -r システムジャーナルサービス定義のファイル名
        -c チェックポイントダンプサービス定義のファイル名
        [ [ , チェックポイントダンプサービス定義のファイル名 ] ... ]
```

機能

ジャーナルサービスを構成する各システムサービス定義のファイル名を指定します。

ファイル名は、OpenTP1 システム内で一意でなければなりません。

オプション

●-r システムジャーナルサービス定義のファイル名

～ 〈1～8 文字の識別子〉

システムジャーナルサービス定義のファイル名を指定します。

●-c チェックポイントダンプサービス定義のファイル名 [, チェックポイントダンプサービス定義のファイル名] ...

～ 〈1～8 文字の識別子〉

チェックポイントダンプサービス定義のファイル名を指定します。

チェックポイントダンプを取得するシステムサービスごとにファイルを作成します。このコマンドの対象となるシステムサービスを次に示します。

- MCF サービス
- トランザクションサービス
- MQA サービス

なお、ファイル名として、cpdN (N：整数) は使用できません。

注意事項

システム共通定義の jnl_fileless_option オペランドに Y を指定している場合、jnldfs 定義コマンドは省略できます。

システムジャーナルサービス定義

形式

set 形式

```
[set jnl_max_datasize=最大レコードデータ長]
[set jnl_cdinterval=ジャーナルブロック数]
[set jnl_rerun_swap=Y|N]
[set jnl_dual=Y|N]
[set jnl_singleoperation=Y|N]
[set jnl_rerun_reserved_file_open=Y|N]
[set jnl_arc_name=グローバルアーカイブジャーナルサービスの
    リソースグループ名@ノード識別子]
[set jnl_arc_buff_size=アーカイブ用バッファサイズ]
[set jnl_arc_max_datasize=アーカイブ時の転送データの最大長]
[set jnl_arc_terminate_check=Y|N]
[set jnl_arc_rec_kind=ジャーナルレコード種別
    [ジャーナルレコード種別] ...]
[set jnl_arc_uj_code="UJコード [, UJコード] ..."]
[set jnl_arc_check_level=1|2]
[set jnl_arc_trn_stat=Y|N]
[set jnl_unload_check=Y|N]
[set jnl_auto_unload=Y|N]
[set jnl_auto_unload_path=
    "アンロードジャーナルファイルの格納ディレクトリ名"
    [, "アンロードジャーナルファイルの格納ディレクトリ名", ...] ]
[set jnl_max_file_dispersion=並列アクセス化する場合の最大分散数]
[set jnl_min_file_dispersion=並列アクセス化する場合の最小分散数]
[set watch_time=最大応答待ち時間]
```

コマンド形式

```
{{{{jnladdfg -g ファイルグループ名 [ONL] }}
  {{jnladdpf -g ファイルグループ名
    [-e 要素ファイル名]
    [-a 物理ファイル名]
    [-b 物理ファイル名] }}}}
```

機能

システムジャーナルファイルにジャーナルを取得するための実行環境を定義します。

説明

set 形式のオペランド

●jnl_max_datasize=最大レコードデータ長

～ 〈符号なし整数〉 ((32000～4000000)) 《32000》 (単位：バイト)

ジャーナルに取得するレコードデータの最大長を指定します。

計算式を次に示します。次の条件のうちの最大値を指定してください。

1. UJ を取得する場合

取得する UAP 履歴情報の最大の長さ + 8 の計算値以上

2. DAM サービスを使用する場合

(i) dam_update_block_over オペランドに flush を指定したとき

$a \times \{(b \times 2) + 24\} + 128$ の計算値以上

(ii) (i) 以外するとき

$a \times (b + 24) + 128$ の計算値以上

a : DAM サービス定義の dam_update_block オペランドの指定値

b : damload コマンド, または dc_dam_create 関数で作成した DAM ファイルのうち, 最大のブロック長 + 8 (damload コマンド, または dc_dam_create 関数の blksize の値 + 8)

(iii) (i) または (ii) で算出した値が 4000000 を超えるとき

4000000 以内のなるべく大きな値

DAM サービスで出力しようとするジャーナルレコード長が jnl_max_datasize オペランドの指定値を超える場合, ジャーナルレコードを jnl_max_datasize オペランドの指定値内のサイズに分割して出力します。ただし, ジャーナルレコードが分割されるとディスクへのジャーナル入出力が増加するおそれがありますので, 4000000 以内のなるべく大きな値としてください。

3. TAM サービスを使用する場合

TAM で取得する CJ のデータ長以上

$a \times 2 + 96$ の計算値以上

a : TAM サービス定義の tam_max_recsz オペランドの指定値を 4 で切り上げた値

4. MCF サービスを使用する場合

MCF で取得する GJ, IJ, MJ, OJ, CJ のデータ長のうちの最大値以上

(i) GJ のデータ長

MCF アプリケーション定義の mcfaalcap の-j オプションの gj オペランドに yes を指定したときに取得します。

●32 ビット版のとき

$\uparrow (204 + \text{seg}) / 4 \uparrow \times 4$

●64 ビット版のとき

$\uparrow (204 + \text{seg}) / 8 \uparrow \times 8$

(ii) IJ のデータ長

MCF アプリケーション定義の mcfaalcap の-j オプションの ij オペランドに yes を指定したときに取得します。

●32 ビット版のとき

$\uparrow (172 + \text{seg}) / 4 \uparrow \times 4$

●64 ビット版のとき

$$\uparrow (172 + \text{seg}) / 8 \uparrow \times 8$$

(iii) MJ のデータ長

mcftactmj コマンドを実行したときに取得します。

●32 ビット版のとき

$$\uparrow (180 + \text{seg}) / 4 \uparrow \times 4$$

●64 ビット版のとき

$$\uparrow (180 + \text{seg}) / 8 \uparrow \times 8$$

(iv) OJ のデータ長

MHP からメッセージ送信をした場合、MCF アプリケーション定義の mcfaalcap の-j オプションの oj オペランドに yes を指定したときに取得します。SPP からメッセージ送信をした場合、ユーザーサービス定義の mcf_spp_oj オペランドに Y を指定したときに取得します。

●32 ビット版のとき

$$\uparrow (204 + \text{seg}) / 4 \uparrow \times 4$$

●64 ビット版のとき

$$\uparrow (204 + \text{seg}) / 8 \uparrow \times 8$$

(v) CJ (メッセージキューサーバ) のデータ長

ITQ, OTQ にディスクキューを使用した場合に取得します。

次に示す三つの計算式のうち最大の値をデータ長としてください。

・計算式 1 (MCF 通信プロセスやアプリケーション起動プロセスで送受信するメッセージ単位で取得する CJ)

●32 ビット版のとき

$$\uparrow \{88 + (24 \times (\uparrow \text{msg} / \text{ql} \uparrow + \uparrow 960 / \text{ql} \uparrow)) + \text{qio}\} / 4 \uparrow \times 4$$

●64 ビット版のとき

$$\uparrow \{88 + (24 \times (\uparrow \text{msg} / \text{ql} \uparrow + \uparrow 1072 / \text{ql} \uparrow)) + \text{qio}\} / 8 \uparrow \times 8$$

・計算式 2 (トランザクション単位で取得する CJ)

●32 ビット版のとき

$$\uparrow \{260 + \Sigma \{24 \times (\uparrow \text{msg} / \text{ql} \uparrow + \uparrow 960 / \text{ql} \uparrow)\} + \text{qio}\} / 4 \uparrow \times 4$$

●64 ビット版のとき

$$\uparrow \{260 + \Sigma \{24 \times (\uparrow \text{msg} / \text{ql} \uparrow + \uparrow 1072 / \text{ql} \uparrow)\} + \text{qio}\} / 8 \uparrow \times 8$$

・計算式 3 (論理端末やサービスグループ単位で取得する CJ)

●32 ビット版のとき

$$260 + (24 \times \text{rmsg})$$

●64 ビット版のとき

$$\uparrow \{260 + 24 \times \text{rmsg}\} / 8 \uparrow \times 8$$

(凡例)

$\uparrow$: 小数点以下を切り上げます。

seg：セグメント長

msg：ディスクキューを使用した受信メッセージ長または送信メッセージ長

ql：キューファイル物理レコード長

qio：メッセージキューサービス定義の que_io_maxrecsize オペランドの指定値

Σ：該当するトランザクションで、受信または送信したメッセージに対する見積もりの総和

rmsg：すべての論理端末、サービスグループのうちの構成変更準備停止時における滞留メッセージ数の最大値（構成変更準備停止を行う直前にそれぞれ-m オプションを指定した mcftlsle コマンド、または mcftlssg コマンドを実行することで、滞留メッセージの最大値を取得できます）

5. MQA サービスを使用する場合

(i) 32 ビットのとき

$$j \geq 716 + 4 \times m$$

(ii) 64 ビットのとき

$$j \geq 744 + 4 \times m$$

(凡例)

j：jnl_max_datasize オペランドの指定値

m：mqa_max_msg_recnum オペランドの指定値

TP1/Message Queue を使用する場合は、マニュアル「TP1/Message Queue 使用の手引」の mqa_max_msg_recnum オペランドの説明を参照してください。

●jnl_cdinterval=ジャーナルブロック数

～〈符号なし整数〉((100～32000))《1000》

チェックポイントダンプを取得する間隔を、ジャーナルブロック数で指定します。

ここで指定した数のジャーナルをジャーナルファイルに格納した時点で、チェックポイントダンプが取得されます。

ここで指定するブロック数以上の容量がジャーナルファイルに必要です。

●jnl_rerun_swap=Y|N

～《N》

OpenTP1 システム再開始時にジャーナルをスワップするかどうかを指定します。

スワップすることで、物理的にジャーナルを分けられます。

Y

ジャーナルをスワップします。

N

ジャーナルをスワップしません。

●jnl_dual=Y|N

～《N》

ジャーナルファイルを二重化するかどうかを指定します。

Y

ジャーナルファイルを二重化します。

N

ジャーナルファイルを二重化しません。

●jnl_singleoperation=Y|N

～《N》

ジャーナルファイルを二重化する場合、片系しかスワップ先として使用できなくなったときに、スワップ先として割り当てられているかどうかを指定します。

Y

片系しかスワップ先として使用できない場合でもスワップ先として割り当てます（片系運転可）。

N

片系しかスワップ先として使用できない場合はスワップ先として割り当てません（片系運転不可）。

●jnl_rerun_reserved_file_open=Y|N

～《N》

全面回復時にオープンされたファイルグループがすべて上書き禁止の状態で、スワップ先がない場合、予約ファイルがあれば予約ファイルをオープンするかどうかを指定します。

Yを指定しても、上書きできるファイルグループがある場合は、スワップ先がなくても予約ファイルをオープンしません。

Y

予約ファイルをオープンします。

全面回復のとき、スワップ先がないためにシステムが再度停止することを防ぎます。

N

予約ファイルをオープンしません。

●jnl_arc_name=グローバルアーカイブジャーナルサービスのリソースグループ名@ノード識別子

～〈1～8 文字の識別子〉@〈4 文字の識別子〉

該当するジャーナルをアーカイブする先の「ノード識別子（システム共通定義の node_id で指定する名称）」と、そのノードで定義された「グローバルアーカイブジャーナルサービスのリソースグループ名（アーカイブジャーナルサービス定義のファイル名）」を、@で連結したものを指定します。

●jnl_arc_buff_size=アーカイブ用バッファサイズ

～〈符号なし整数〉((2～80))《10》(単位：メガバイト)

該当するジャーナルをアーカイブするための、共用メモリバッファの容量を指定します。ここで指定する値は、グローバルアーカイブジャーナルサービスが異常終了と再開に掛かる時間の間に発生するジャーナル量から求めます。また、ここで指定する値は、次の計算式で求められた値以上の値を指定してください。

$$\text{jnl\_arc\_buff\_size オペランドの値} = \lceil (\text{jnl\_arc\_max\_datasize} \times 1024) / 1048576 \rceil \times 3$$

この計算式で求められた値より jnl_arc_buff_size オペランドの指定値が小さい場合は、OpenTP1 の開始時に KFCA01113-E の理由コード 427 を出力し、OpenTP1 の開始処理を中止します。

このオペランドは、jnl_arc_name オペランドを定義している場合にだけ有効です。

●jnl_arc_max_datasize=アーカイブ時の転送データの最大長

～〈符号なし整数〉((1020～8188))《1020》(単位：キロバイト)

アーカイブノードにジャーナルデータをアーカイブする場合の最大転送データ長を指定します。このオペランドの値は、次の計算式で求めた値以上の値を指定してください。

$$\text{jnl\_arc\_max\_datasize} \geq \lceil (\lceil (\text{jnl\_max\_datasize} + 336) / 4096 \rceil \times 4096) / 1024 \rceil$$

計算式で求められた値より jnl_arc_max_datasize の指定値が小さい場合は、OpenTP1 の開始時に KFCA01113-E メッセージの理由コード 424 を出力して OpenTP1 の開始処理を中止します。

また、このオペランドに 1020 より大きな値を指定する場合は、次の計算式で求められた値以上の値をシステム共通定義の rpc_max_message_size オペランドに指定してください。

$$\text{rpc\_max\_message\_size} \geq \lceil (\text{jnl\_arc\_max\_datasize} \times 1024 + 4096) / 1048576 \rceil$$

計算式で求められた値より rpc_max_message_size が小さい場合は、OpenTP1 の開始時に KFCA01113-E の理由コード 425 を出力して OpenTP1 の開始処理を中止します。このオペランドは、jnl_arc_name オペランドを定義している場合にだけ有効です。また、このオペランドの指定値以上の値を、アーカイブジャーナルサービス定義の jnl_arc_max_datasize オペランドに指定してください。

アーカイブジャーナルサービス定義の jnl_arc_max_datasize オペランドの値がこのオペランドの指定値より小さく、かつ次の計算値で求めた値より小さい場合は、アーカイブノードとのコネクションの確立時に、KFCA04133-W の理由コード 427 を出力し、アーカイブ機能を停止します。

$$\begin{aligned} &\text{アーカイブジャーナルサービス定義の jnl\_arc\_max\_datasize} \\ &\geq \lceil (\lceil (\text{jnl\_max\_datasize} + 336) / 4096 \rceil \times 4096) / 1024 \rceil \end{aligned}$$

●jnl_arc_terminate_check=Y|N

～〈Y〉

該当するジャーナルサービスが正常終了および計画停止するとき、アーカイブ対象ジャーナルをすべてアーカイブするかどうかを指定します。

Y

すべてアーカイブして終了します。

N

アーカイブを中断して終了します。

このオペランドは、jnl_arc_name オペランドを定義している場合にだけ有効です。

●jnl_arc_rec_kind=ジャーナルレコード種別〔ジャーナルレコード種別〕 …

～《acfgimosu》

アーカイブ対象となるジャーナルレコード種別を指定します。

ジャーナルレコード種別の指定値と内容を次に示します。

レコード種別	指定値	内 容	ジャーナルの種類
FJ	f	DAM ファイルの更新情報	回復用ジャーナル
CJ	c	回復対象テーブルの更新情報	
SJ	s	システム統計情報	統計用ジャーナル
AJ	a	送信完了情報	
IJ	i	入力キュー登録メッセージ	
OJ	o	出力キュー登録メッセージ	
MJ	m	メッセージジャーナル	
GJ	g	receive 情報	
UJ	u	ユーザ任意の情報	ユーザジャーナル

このオペランドの指定を省略した場合、すべてのジャーナルレコード種別がアーカイブ対象となります。

また、jnl_arc_trn_stat オペランドで Y を指定した場合、同期点ジャーナルおよびトランザクション管理サービスの回復用ジャーナルは、このオペランドの指定に関係なく、すべてアーカイブの対象となります。

このオペランドを指定した場合は、jnl_arc_check_level オペランドの指定に注意してください。詳細は、jnl_arc_check_level オペランドの説明を参照してください。

このオペランドは、jnl_arc_name オペランドを指定している場合にだけ有効です。

●jnl_arc_uj_code="UJ コード [, UJ コード] …"

～〈符号なし整数〉((0～255))《0-255》

UJ がアーカイブ対象となっている場合に、アーカイブする UJ コードを指定します。コンマ', 'で区切ると、複数の UJ コードを指定できます。

UJ コードの先頭に「0x」を指定した場合は 16 進数、「0」を指定した場合は 8 進数として指定できます。また、UJ コードをハイフン '-' でつなぐと、範囲指定できます。

(例)

15 : 10 進数で「15」

0x15 : 16 進数で「21」

015 : 8 進数で「13」

0-255 : 「0」から「255」までのすべての値

このオペランドの指定を省略した場合、すべての UJ コードがアーカイブ対象となります。

このオペランドは、jnl_arc_name オペランドを指定していて、かつ jnl_arc_rec_kind オペランドを省略または u を指定している場合にだけ有効です。

●jnl_arc_check_level=1|2

～《1》

グローバルアーカイブジャーナル機能を使用した場合に、ファイルグループがスワップ先に割り当てられる条件を指定します。

1

アンロード済みまたはアーカイブ済み状態のファイルグループをスワップ先に割り当てます。

2

アンロード済みかつアーカイブ済み状態のファイルグループをスワップ先に割り当てます。

このオペランドで 1 を指定し、jnl_arc_rec_kind オペランドにアーカイブするジャーナル種別を指定した場合は、オンラインの回復に必要なジャーナルファイルが失われることがあります。そのため、jnl_arc_rec_kind オペランドでアーカイブするジャーナル種別を選択した場合には、jnl_arc_check_level オペランドで 2 を指定してください。この時、アーカイブ済みのファイルグループをスワップ先として割り当てるには、jnlunlfg コマンドでアンロードするか、jnlchgfg コマンドでファイルグループを強制的にアンロード済み状態に変更してください。また、アーカイブ機能の障害時には、jnlchgfg コマンドでファイルグループを強制的にアンロード済み状態に変更してください。

このオペランドは、jnl_arc_name オペランドを指定している場合にだけ有効です。

●jnl_arc_trn_stat=Y|N

～《Y》

同期点ジャーナルおよびトランザクション管理サービスの回復用ジャーナルを、アーカイブの対象とすることを指定します。

Y

アーカイブの対象とします。

N

アーカイブの対象としません。

N を指定した場合、アーカイブジャーナルファイルを使った次の運用コマンドの動作は保証しません。

- damfrc コマンド (DAM 論理ファイルの回復)
- tamfrc コマンド (TAM ファイルの回復)
- jnlcolc コマンド (ファイル回復用ジャーナルの集積)
- jnlmkrf コマンド (ジャーナル関係のファイル回復)
- jnlrput コマンド (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) の-e オプション

このオペランドは、jnl_arc_name オペランドおよび jnl_arc_rec_kind オペランドを指定している場合にだけ有効です。

●jnl_unload_check=Y|N

～《Y》

スワップ先のファイルグループを決定する時に、アンロード待ち状態をチェックするかどうかを指定します。

Y

アンロード待ち状態をチェックします。

N

アンロード待ち状態をチェックしません。

Y を指定した場合、OpenTP1 が強制終了、または異常終了したあとに OpenTP1 を強制正常開始したとき、前回のオンラインで現用として使用されていたジャーナルファイルはクローズ状態となります。jnlunlfg コマンドまたは jnlchgfg コマンドでファイルグループのステータスを変更するまで、ジャーナルファイルは使用できません。

N を指定した場合、アンロードしていないジャーナルは失われます。そのため、通常は、テストなどでジャーナルの運用（ユーザファイルの回復や稼働統計情報の編集など）を必要としない場合に指定します。また、ジャーナルが失われるため、アンロードジャーナルファイルを指定するコマンドが、一部使用できなくなります。詳細は、マニュアル「OpenTP1 運用と操作」を参照してください。

グローバルアーカイブジャーナルサービス機能を使用して、被アーカイブノードとして動作させる環境で N を指定した場合、転送が完了していないジャーナルが上書きされ、アーカイブノード側で世代抜けエラーが発生する場合があります。被アーカイブノードとして起動する場合は、N を指定しないでください。

●jnl_auto_unload=Y|N

～《N》

自動アンロード機能を使用するかどうかを指定します。自動アンロード機能の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

Y

自動アンロード機能を使用します。

N

自動アンロード機能を使用しません。

jnl_unload_check オペランドに N が指定されている場合は、jnl_auto_unload オペランドに Y を指定しても自動アンロード機能を使用できません。

●jnl_auto_unload_path="アンロードジャーナルファイルの格納ディレクトリ名" ["アンロードジャーナルファイルの格納ディレクトリ名",...]

～〈パス名〉《\$DCDIR/spool/dcjnlinf/unload》

自動アンロード機能を使用する場合に、アンロードジャーナルファイルを格納するディレクトリを絶対パス名で指定します。パス名に指定できる文字数は、最大 80 バイトです。複数のディレクトリを指定できます。指定できるディレクトリの最大数は 32 です。

このオペランドは、jnl_unload_check オペランドおよび jnl_auto_unload オペランドに Y を指定した場合にだけ有効です。

このオペランドで指定したパス名に誤りがある場合は、自動アンロード機能を使用できません。また、このオペランドの指定を省略した場合は、\$DCDIR/spool/dcjnlinf/unload を仮定します。

このオペランドに指定するディレクトリには、アンロードジャーナルファイルを格納できるディスク容量が必要です。アンロードジャーナルファイルを格納するディレクトリに必要なディスク容量の算出式を次に示します。

ディスク容量（単位：バイト）＝ジャーナルファイルレコード数
×ジャーナルファイルレコード長
×格納するアンロードジャーナルファイル数
×1.2

●jnl_max_file_dispersion=並列アクセス化する場合の最大分散数

～〈符号なし整数〉((1～8))《1》

システムジャーナルを並列アクセス化する場合の最大分散数を指定します。

一つのファイルグループで使用する最大要素ファイル数を指定してください。1 を指定した場合、または指定を省略した場合は、jnl_min_file_dispersion オペランドの指定値は無効になります。

●jnl_min_file_dispersion=並列アクセス化する場合の最小分散数

～〈符号なし整数〉((1～8))《1》

システムジャーナルを並列アクセス化する場合の最小分散数を指定します。

ファイルグループを構成する要素ファイルが障害などで使用できなくなった場合でも、現用として使用できる最小要素ファイル数を指定してください。使用できる要素ファイル数が最小要素ファイル数未満になった場合、ファイルグループは使用できなくなります。

並列アクセス化する場合の最小分散数は次の範囲で指定してください。

最大分散数 ≥ 最小分散数 ≥ 1

●watch_time=最大応答待ち時間

～〈符号なし整数〉((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しない場合があります。

省略した場合は、システム共通定義の watch_time オペランドの値を仮定します。

このオペランドは、システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

次ページ以降に記述しています。

jnladdfg

名称

システムジャーナルファイルのファイルグループ名と属性の定義

形式

```
{{jnladdfg -g ファイルグループ名 [ONL] }}
```

機能

システムジャーナルファイルのファイルグループ名と属性を定義します。

jnladdfg 定義コマンドは、システムジャーナルサービス定義内に最低 2 個、最大 256 個指定できます。

ただし、ONL を指定した jnladdfg 定義コマンドの場合は、最低 2 個必要です。

なお、ファイルグループ名は、システムジャーナルサービス定義内で一意でなければなりません。

オプション

●-g システムジャーナルファイルのファイルグループ名

～ 〈1～8 文字の識別子〉

システムジャーナルファイルのファイルグループ名を指定します。

このオプションの指定値を変更した場合は、システムジャーナルファイルの初期化が必要です。

●ONL

このファイルグループを、オンライン開始と同時に**オープン状態**にする場合に指定します。

省略した場合は、オンライン開始時には**クローズ状態**の予約のファイルグループになります。

注意事項

システム共通定義の jnl_fileless_option オペランドに Y を指定している場合、jnladdfg 定義コマンドは省略できます。

名称

システムジャーナルファイルのファイルグループを構成する物理ファイルの定義

形式

```
{ {jnladdpf -g ファイルグループ名  
    [-e 要素ファイル名]  
    -a 物理ファイル名  
    [-b 物理ファイル名] } }
```

機能

システムジャーナルファイルのファイルグループを構成する物理ファイルを定義します。

jnladdpf 定義コマンドは、ファイルグループに対して最低 1 個、最大 8 個指定できます。

なお、物理ファイル名は、OpenTP1 システム内で一意でなければなりません。

オプション

●-g システムジャーナルファイルのファイルグループ名

～ 〈1～8 文字の識別子〉

システムジャーナルファイルのファイルグループ名を指定します。

ここで指定するファイルグループ名は、あらかじめ jnladdfg 定義コマンドで定義しておきます。なお、このオプションの指定値を変更した場合は、システムジャーナルファイルの初期化が必要です。

●-e 要素ファイル名

～ 〈1～8 文字の識別子〉

ファイルグループを構成する要素ファイル名を指定します。

要素ファイル名はシステムジャーナルサービス定義内で一意でなければなりません。

並列アクセス化 (jnl_max_file_dispersion オペランドに 2 以上を指定) をする場合は、必ずこのオプションを指定してください。並列アクセス化をしない場合は、指定する必要はありません。

並列アクセス化をする場合に -e オプションを省略すると、定義解析でエラーになります。一つのファイルグループに指定する要素ファイル数は次の範囲で指定してください。

```
jnl_max_file_dispersionオペランドの指定値 ≥ 要素ファイル数 ≥ jnl_min_file_dispersionオペランド  
の指定値
```


ここで指定する要素ファイル数が jnl_max_file_dispersion オペランドの指定値を超えた場合、超えた分はオンラインでは使用されません。また、ここで指定する要素ファイル数が jnl_min_file_dispersion オペランドの指定値に満たない場合、そのファイルグループは使用できません。

●-a 物理ファイル名

～ 〈パス名〉

このファイルグループを構成する物理ファイル名を絶対パス名で指定します。ただし、環境変数は使用できません。

この物理ファイルはジャーナル関係のファイル用として作成した OpenTP1 ファイルを使用します。

set 形式の jnl_dual オペランドでファイルの二重化を指定した場合は、ここに A 系ファイルの物理ファイル名を指定してください。

二重化を指定しても -a の物理ファイル名しか指定しない場合は、コマンド全体が無視されます。

●-b 物理ファイル名

～ 〈パス名〉

set 形式の jnl_dual オペランドでファイルの二重化を指定した場合に、B 系ファイルの物理ファイル名を絶対パス名で指定します。ただし、環境変数は使用できません。この物理ファイルはジャーナル関係のファイル用として作成した OpenTP1 ファイルを使用します。

二重化の指定をしていない場合に -a および -b の物理ファイル名を指定すると、-b の方が無効となります。また、二重化する場合もしない場合も、-b の物理ファイル名だけを指定すると、コマンド全体が無視されます。

注意事項

システム共通定義の jnl_fileless_option オペランドに Y を指定している場合、jnladdpf 定義コマンドは省略できます。

チェックポイントダンプサービス定義

形式

set 形式

```
set jnl_objservername="対象となるシステムサービス名"  
[set jnl_max_datasize=チェックポイントダンプ取得用バッファ長]  
[set assurance_count=有効保証世代数]  
[set jnl_reduced_mode=縮退運転オプション]  
[set jnl_reserved_file_auto_open=Y|N]  
[set jnl_dual=Y|N]  
[set jnl_singleoperation=Y|N]  
[set jnl_cdskip_limit=チェックポイントダンプ取得契機の  
スキップ回数の上限值]  
[set jnl_cdskip_msg=Y|N]  
[set cpd_message_id_change_level=メッセージIDの変更レベル]  
[set watch_time=最大応答待ち時間]
```

コマンド形式

```
{{{jnladdfg [-j srf]  
-g ファイルグループ名 [ONL] }}  
{jnladdpf [-j srf]  
-g ファイルグループ名 -a 物理ファイル名  
[-b 物理ファイル名] }}}
```

機能

チェックポイントダンプファイルにチェックポイントダンプを取得するための実行環境を定義します。トランザクションサービスについては必ず定義します。そのほかのサービスについては、そのサービスを使用する場合にだけ定義します。

説明

set 形式のオペランド

●jnl_objservername="対象となるシステムサービス名"

～ 〈1～8 文字の識別子〉

対象となるシステムサービス名を指定します。

MCF サービスの場合

システムサービス構成定義の dcsvstart -m で指定した名称を指定します。

トランザクションサービスの場合

常に_tjl を指定します。

MQA サービスの場合

常に_mqa を指定します。

●jnl_max_datasize=チェックポイントダンプ取得用バッファ長

～〈符号なし整数〉((32000～4000000))《32768》(単位：バイト)

チェックポイントダンプ情報をチェックポイントダンプファイルに書き出したり，読み込んだりするバッファの大きさを指定します。

チェックポイントダンプファイルの入出力回数を制御できます。

指定するバッファの大きさが大きいほどチェックポイントダンプファイルに対する入出力回数が少なくなります。

●assurance_count=有効保証世代数

～〈符号なし整数〉((1～2))《1》

システム回復時のチェックポイントダンプファイルの入力障害などに備え，オンライン中に保存しておくチェックポイントダンプファイルの世代数を指定します。ここで指定した世代数のファイルは，上書きされません。

これによって，最新世代のチェックポイントダンプファイルに障害が発生しても，残りの有効保証世代を使って回復ができ，システムの信頼性が向上します。

チェックポイントダンプファイルを取得するには，有効保証世代数+1 のファイルグループが必要です。

複数の有効保証世代を指定する場合，上書きできないジャーナルが増えることによって，ジャーナルの次回取得先ファイルが不足することがあります。これを防ぐため，ジャーナルファイルのブロック数は，次の値を目安にして設定してください。

チェックポイントダンプを取得する間隔となるジャーナルブロック数×（有効保証世代数+1）

●jnl_reduced_mode=縮退運転オプション

～〈符号なし整数〉((0～2))《0》

チェックポイントダンプの縮退機能を使用するかどうかを指定します。

オンライン中，または再開処理中のファイル障害等によって，運用に必要なファイル数を下回った場合でも，最低二つのファイルが使用できれば処理を続行します。この場合，assurance_count オペランドの指定は無視されます。

0

縮退機能を使用しません。

1

縮退機能を使用します。

縮退機能を使用し、縮退状態でチェックポイントダンプ取得契機が発生するたびに、警告メッセージを出力します。

●jnl_reserved_file_auto_open=Y|N

～《N》

オンライン中にファイル障害等によって、運用に必要なファイル数を下回った場合、未使用の予約ファイルを自動的にオープンして処理を続行するかどうかを指定します。

Y

予約ファイルを自動的にオープンします。

N

予約ファイルを自動的にオープンしません。

この指定は、縮退運転オプションより優先されます。

●jnl_dual=Y|N

～《N》

チェックポイントダンプファイルを二重化するかどうかを指定します。

二重化する場合は、一つのファイルグループに二つの物理ファイル（A系とB系）を指定してください。

Y

チェックポイントダンプファイルを二重化します。

N

チェックポイントダンプファイルを二重化しません。

●jnl_singleoperation=Y|N

～《N》

チェックポイントダンプファイルを二重化している際、上書きできる物理ファイルが片系だけになった場合に、上書きできるファイルグループとして割り当てるかどうかを指定します。

Y

上書きできる物理ファイルが片系だけの場合は、上書きできるファイルグループとして割り当てます（片系運転可）。

N

上書きできる物理ファイルが片系だけの場合でも、上書きできるファイルグループとして割り当てません（片系運転不可）。

ファイルグループは予約状態となります。

●jnl_cdskip_limit=チェックポイントダンプ取得契機のスキップ回数の上限値

～ 〈符号なし整数〉 ((1～32767)) (単位：回)

チェックポイントダンプ取得契機のスキップ回数の上限値を指定します。チェックポイントダンプ取得契機のスキップ回数がこのオペランドで指定した回数に達すると、スキップ要因となっているトランザクションの情報を KFCA32550-I メッセージで出力し、そのトランザクションを実行しているプロセスを強制停止します。

このオペランドは、チェックポイントダンプサービス定義の jnl_objservername オペランドに _tjl を指定した場合に有効です。_tjl 以外を指定した場合、このオペランドの指定は無視されます。

このオペランドの指定を省略した場合は、チェックポイントダンプ取得契機のスキップ回数を監視しません。

このオペランドの指定値が大きすぎると、チェックポイントダンプの取得処理が完了できないため、使用（上書き）できるシステムジャーナルファイルが不足してしまいます。この場合、OpenTP1 は異常終了します。また、指定値が小さすぎると、プロセスの強制停止によるトランザクションのロールバックが増加することがあります。

スキップ回数の上限値の見積もり式については、マニュアル「OpenTP1 運用と操作」のチェックポイントダンプ取得契機のスキップ回数の監視の説明を参照してください。

●jnl_cdskip_msg=Y|N

～ 〈N〉

チェックポイントダンプ取得契機をスキップする要因となった、トランザクションの情報を出力するかどうかを指定します。

このオペランドは、チェックポイントダンプサービス定義の jnl_objservername オペランドに _tjl を指定した場合に有効です。_tjl 以外を指定した場合、このオペランドの指定は無視されます。

Y

スキップ要因となっているトランザクションの情報として、KFCA32550-I メッセージを出力します。

N

スキップ要因となっているトランザクションの情報を出力しません。

●cpd_message_id_change_level=メッセージ ID の変更レベル

～ 〈符号なし整数〉 ((0～1)) 《0》

チェックポイントダンプサービスが出力するメッセージで、重要な情報となるメッセージの種別を変更します。

メッセージログを監視し、メッセージの種類によって管理方法を区別したい場合にこのオペランドを使用してください。

このオペランドは、チェックポイントダンプサービス定義の jnl_objservername オペランドに_tjl を指定した場合に有効です。_tjl 以外を指定した場合、このオペランドの指定は無視されます。

0
メッセージ ID を変更しないで、従来どおりのメッセージ ID で出力します。

1
メッセージ ID を変更し、出力します。

このオペランドの指定値と出力されるメッセージの関係を次に示します。

cpd_message_id_change_level オペランドの指定値	0	1
出力されるメッセージ ID	KFCA02179-I	KFCA02161-W

●**watch_time=最大応答待ち時間**
～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しない場合があります。

省略した場合は、システム共通定義の watch_time オペランドの値を仮定します。

このオペランドは、システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

次ページ以降に記述しています。

jnladdfg

名称

チェックポイントダンプファイルのファイルグループ名と属性の定義

形式

```
{{jnladdfg [-j srf]
          -g ファイルグループ名 [ONL] }}
```

機能

チェックポイントダンプファイルとサーバリカバリジャーナルファイルのファイルグループ名と属性を定義します。ただし、サーバリカバリジャーナルファイルを OpenTP1 ファイルで作成しない場合、-j オプションを指定した jnladdfg 定義コマンドと jnladdpf 定義コマンドは指定してはなりません。また、-j オプションを指定した jnladdfg 定義コマンドは、定義内に一つしか指定できません。

jnladdfg 定義コマンドは、チェックポイントダンプサービス定義内に最低 2 個、最大 60 個指定できます。チェックポイントダンプファイルの ONL を指定した jnladdfg 定義コマンドの場合は、最低 2 個、最大 30 個指定できます。

なお、ファイルグループ名は、すべてのチェックポイントダンプサービス定義内で一意でなければなりません。

オプション

●-j srf

～ 〈1～8 文字の識別子〉

系切り替え構成で、サーバリカバリジャーナルファイルを使用する場合には、サーバリカバリジャーナルファイルを、共有ディスク装置上の OpenTP1 ファイルとして作成する必要があります。その際、必ずこのオプションを指定してください。

マニュアル「OpenTP1 解説」のサーバリカバリジャーナルファイルについての記述もあわせて参照してください。

●-g チェックポイントダンプファイルのファイルグループ名

～ 〈1～8 文字の識別子〉

チェックポイントダンプファイルのファイルグループ名を指定します。

●ONL

このファイルグループを、オンライン開始と同時にオープン状態にする場合に指定します。

省略した場合は、オンライン開始時にはクローズ状態の予約のファイルグループになります。

サーバリカバリジャーナルファイルの場合、この指定を省略してもオンライン開始と同時にオープン状態になります。

名称

チェックポイントダンプファイルのファイルグループを構成する物理ファイルの定義

形式

```
{jnladdpf [-j srf]
          -g ファイルグループ名 -a 物理ファイル名
          [-b 物理ファイル名 ] }
```

機能

チェックポイントダンプファイルのファイルグループを構成する物理ファイルを定義します。

jnladdpf 定義コマンドを定義する場合、このコマンドで指定するファイルグループ名を指定した jnladdfg 定義コマンドを定義しなければなりません。

jnladdpf 定義コマンドは、ファイルグループに対して 1 個だけ指定します。

なお、物理ファイル名は、OpenTP1 システム内で一意でなければなりません。

また、ONL を指定していない jnladdfg 定義コマンドに対する jnladdpf 定義コマンドは、省略できます。省略した場合は、運用コマンドの jnladdpf 定義コマンドで物理ファイルを割り当てます。ただし、サーバリカバリジャーナルファイルの場合は、省略できません。

オプション

●-j srf

～ 〈1～8 文字の識別子〉

系切り替え構成で、サーバリカバリジャーナルファイルを使用する場合には、サーバリカバリジャーナルファイルを、共有ディスク装置上の OpenTP1 ファイルとして作成する必要があります。その際、必ずこのオプションを指定してください。

マニュアル「OpenTP1 解説」のサーバリカバリジャーナルファイルについての記述もあわせて参照してください。

●-g チェックポイントダンプファイルのファイルグループ名

～ 〈1～8 文字の識別子〉

チェックポイントダンプファイルのファイルグループ名を指定します。

ここで指定するファイルグループ名は、あらかじめ jnladdfg 定義コマンドで定義しておきます。

●-a 物理ファイル名

～ 〈パス名〉

-g オプションで指定した、ファイルグループを構成する物理ファイル名を絶対パス名で指定します。ただし、環境変数は使用できません。

チェックポイントダンプファイルを二重化して運用する場合は、ファイルグループの A 系を構成する物理ファイル名を指定します。

なお、この物理ファイルはジャーナル関係のファイル用として作成した OpenTP1 ファイルを指定します。

●-b 物理ファイル名

～ 〈パス名〉

-g オプションで指定した、ファイルグループの B 系を構成する物理ファイル名を絶対パス名で指定します。チェックポイントダンプファイルを二重化して運用する場合だけ指定します。

なお、この物理ファイルはジャーナル関係のファイル用として作成した OpenTP1 ファイルを指定します。

また、-j オプションと-b オプションを同時に指定した場合、-b オプションは無効になります。

ログサービス定義

形式

set 形式

```
[set log_filesize=メッセージログファイルの最大サイズ]
[set log_msg_console=Y|N]
[set log_msg_allno=Y|N]
[set log_msg_prcid=Y|N]
[set log_msg_prcno=Y|N]
[set log_msg_sysid=Y|N]
[set log_msg_date=Y|N]
[set log_msg_time=Y|N]
[set log_msg_hostname=Y|N]
[set log_msg_pgmid=Y|N]
[set log_netm_out=Y|N]
[set log_netm_allno=Y|N]
[set log_netm_prcid=Y|N]
[set log_netm_prcno=Y|N]
[set log_netm_sysid=Y|N]
[set log_netm_date=Y|N]
[set log_netm_time=Y|N]
[set log_netm_hostname=Y|N]
[set log_netm_pgmid=Y|N]
[set log_jp1_allno=Y|N]
[set log_jp1_prcid=Y|N]
[set log_jp1_prcno=Y|N]
[set log_jp1_sysid=Y|N]
[set log_jp1_date=Y|N]
[set log_jp1_time=Y|N]
[set log_jp1_hostname=Y|N]
[set log_jp1_pgmid=Y|N]
[set log_notify_out=Y|N]
[set log_notify_allno=Y|N]
[set log_notify_prcid=Y|N]
[set log_notify_prcno=Y|N]
[set log_notify_sysid=Y|N]
[set log_notify_date=Y|N]
[set log_notify_time=Y|N]
[set log_notify_hostname=Y|N]
[set log_notify_pgmid=Y|N]
[set log_jerr_rint=メッセージログ出力抑止回数]
[set log_syslog_out=syslog出力レベル]
[set log_syslog_allno=Y|N]
[set log_syslog_prcid=Y|N]
[set log_syslog_prcno=Y|N]
[set log_syslog_sysid=Y|N]
[set log_syslog_date=Y|N]
[set log_syslog_time=Y|N]
[set log_syslog_hostname=Y|N]
[set log_syslog_pgmid=Y|N]
[set log_syslog_append_nodeid=Y|N]
[set log_syslog_elist=syslog失敗リストの要素数]
[set log_syslog_elist_rint=syslog失敗リストを定期的に出力する間隔]
[set log_syslog_synchro=Y|N]
```

```
[set log_audit_out=Y|N]
[set log_audit_path=監査ログファイルの出力先ディレクトリ]
[set log_audit_size=監査ログファイルの最大サイズ]
[set log_audit_count=監査ログファイルの最大数]
[set log_audit_message=監査ログを取得する項目のメッセージID
                        [, 監査ログを取得する項目のメッセージID] ...]
[set watch_time=最大応答待ち時間]
```

コマンド形式

なし。

putenv 形式

```
[putenv TZ タイムゾーン]
[putenv DCSYSLOGOUT 1|1以外の文字列]
```

機能

メッセージログの出力など、次に示すログサービス機能に関する環境を定義します。

- メッセージログ出力機能：メッセージログをメッセージログファイルに出力します。
- リアルタイム出力機能：メッセージログを標準出力に出力します。
- NETM 出力機能：メッセージログを JP1/Cm2 に出力します。
- JP1 イベントサービスへの登録機能：JP1 イベントサービス機能にイベントを登録します。
- メッセージログ通知機能：メッセージログを専用のアプリケーションプログラムへ通知します。
- syslog 出力機能：メッセージログを syslog へ出力します。
- 監査ログ機能：監査ログを監査ログファイルへ出力します。

説明

set 形式のオペランド

●log_filesize=メッセージログファイルの最大サイズ

～ 〈符号なし整数〉 ((1～32767)) 《10240》 (単位：キロバイト)

メッセージログファイルの最大サイズを指定します。ログサービスごとに二つのメッセージログファイルがあります。指定した最大サイズに達したら、メッセージログファイルを切り替えます。

●log_msg_console=Y|N

～ 《Y》

リアルタイム出力機能を使用するかどうかを指定します。

Y

リアルタイム出力機能を使用します。

N

リアルタイム出力機能を使用しません。

Y を指定したときだけ、以下のオペランド (log_msg_xxxx) の指定が有効になります。

●log_msg_allno=Y|N

～《N》

リアルタイム出力機能を使用する時に、メッセージログのシステム内のメッセージ通番を付けるかどうかを指定します。

Y

メッセージログのシステム内のメッセージ通番を付けます。

N

メッセージログのシステム内のメッセージ通番を付けません。

●log_msg_prcid=Y|N

～《N》

リアルタイム出力機能を使用する時に、要求元のプロセスのプロセス ID を付けるかどうかを指定します。

Y

要求元のプロセスのプロセス ID を付けます。

N

要求元のプロセスのプロセス ID を付けません。

●log_msg_prcno=Y|N

～《N》

リアルタイム出力機能を使用する時に、メッセージログのプロセス内のメッセージ通番を付けるかどうかを指定します。

Y

メッセージログのプロセス内のメッセージ通番を付けます。

N

メッセージログのプロセス内のメッセージ通番を付けません。

●log_msg_sysid=Y|N

～《Y》

リアルタイム出力機能を使用する時に、OpenTP1 識別子を付けるかどうかを指定します。

Y

OpenTP1 識別子を付けます。

N

OpenTP1 識別子を付けません。

●log_msg_date=Y|N

～《Y》

リアルタイム出力機能を使用する時に、メッセージログの出力要求時の日付を付けるかどうかを指定します。

Y

メッセージログの出力要求時の日付を付けます。

N

メッセージログの出力要求時の日付を付けません。

●log_msg_time=Y|N

～《Y》

リアルタイム出力機能を使用する時に、メッセージログの出力要求時の時刻を付けるかどうかを指定します。

Y

メッセージログの出力要求時の時刻を付けます。

N

メッセージログの出力要求時の時刻を付けません。

●log_msg_hostname=Y|N

～《Y》

リアルタイム出力機能を使用する時に、メッセージログの出力要求元のホスト名を付けるかどうかを指定します。

Y

メッセージログの出力要求元のホスト名を付けます。

N

メッセージログの出力要求元のホスト名を付けません。

●log_msg_pgmid=Y|N

～《Y》

リアルタイム出力機能を使用する時に、メッセージログの出力要求元のプログラム ID を付けるかどうかを指定します。

Y

メッセージログの出力要求元のプログラム ID を付けます。

N

メッセージログの出力要求元のプログラム ID を付けません。

●log_netm_out=Y|N

～《N》

メッセージログを NETM に出力するかどうかを指定します。

Y

NETM に出力します。

N

NETM に出力しません。

Y を指定したときだけ、以下のオペランド (log_netm_xxxx) の指定が有効になります。

●log_netm_allno=Y|N

～《N》

NETM に出力する場合、メッセージログのシステム内の通番を付けるかどうかを指定します。

Y

メッセージログのシステム内の通番を付けます。

N

メッセージログのシステム内の通番を付けません。

●log_netm_prcid=Y|N

～《N》

NETM に出力する場合、メッセージログのプロセス ID を付けるかどうかを指定します。

Y

メッセージログのプロセス ID を付けます。

N

メッセージログのプロセス ID を付けません。

●log_netm_prcno=Y|N

～《N》

NETM に出力する場合、メッセージログのプロセス内の通番を付けるかどうかを指定します。

Y

メッセージログのプロセス内の通番を付けます。

N

メッセージログのプロセス内の通番を付けません。

●log_netm_sysid=Y|N

～《Y》

NETM に出力する場合、OpenTP1 識別子を付けるかどうかを指定します。

Y

OpenTP1 識別子を付けます。

N

OpenTP1 識別子を付けません。

●log_netm_date=Y|N

～《Y》

NETM に出力する場合、メッセージログの出力要求時の日付を付けるかどうかを指定します。

Y

メッセージログの出力要求時の日付を付けます。

N

メッセージログの出力要求時の日付を付けません。

●log_netm_time=Y|N

～《Y》

NETM に出力する場合、メッセージログの出力要求時の時刻を付けるかどうかを指定します。

Y

メッセージログの出力要求時の時刻を付けます。

N

メッセージログの出力要求時の時刻を付けません。

●log_netm_hostname=Y|N

～《Y》

NETM に出力する場合、メッセージログの出力要求元のホスト名を付けるかどうかを指定します。

Y

メッセージログの出力要求元のホスト名を付けます。

N

メッセージログの出力要求元のホスト名を付けません。

●log_netm_pgmid=Y|N

～《Y》

NETM に出力する場合、メッセージログの出力要求元のプログラム ID を付けるかどうかを指定します。

Y

メッセージログの出力要求元のプログラム ID を付けます。

N

メッセージログの出力要求元のプログラム ID を付けません。

●log_jp1_allno=Y|N

～《N》

JP1 イベントサービス機能に出力する場合、メッセージログのシステム内の通番を付けるかどうかを指定します。

Y

メッセージログのシステム内の通番を付けます。

N

メッセージログのシステム内の通番を付けません。

●log_jp1_prcid=Y|N

～《N》

JP1 イベントサービス機能に出力する場合、メッセージログのプロセス ID を付けるかどうかを指定します。

Y

メッセージログのプロセス ID を付けます。

N

メッセージログのプロセス ID を付けません。

●log_jp1_prcno=Y|N

～《N》

JP1 イベントサービス機能に出力する場合、メッセージログのプロセス内の通番を付けるかどうかを指定します。

Y

メッセージログのプロセス内の通番を付けます。

N

メッセージログのプロセス内の通番を付けません。

●log_jp1_sysid=Y|N

～《Y》

JP1 イベントサービス機能に出力する場合、OpenTP1 識別子を付けるかどうかを指定します。

Y

OpenTP1 識別子を付けます。

N

OpenTP1 識別子を付けません。

●log_jp1_date=Y|N

～《Y》

JP1 イベントサービス機能に出力する場合、メッセージログの出力要求時の日付を付けるかどうかを指定します。

Y

メッセージログの出力要求時の日付を付けます。

N

メッセージログの出力要求時の日付を付けません。

●log_jp1_time=Y|N

～《Y》

JP1 イベントサービス機能に出力する場合、メッセージログの出力要求時の時刻を付けるかどうかを指定します。

Y

メッセージログの出力要求時の時刻を付けます。

N

メッセージログの出力要求時の時刻を付けません。

●log_jp1_hostname=Y|N

～《Y》

JP1 イベントサービス機能に出力する場合、メッセージログの出力要求元のホスト名を付けるかどうかを指定します。

Y

メッセージログの出力要求元のホスト名を付けます。

N

メッセージログの出力要求元のホスト名を付けません。

●log_jp1_pgmid=Y|N

～《Y》

JP1 イベントサービス機能に出力する場合、メッセージログの出力要求元のプログラム ID を付けるかどうかを指定します。

Y

メッセージログの出力要求元のプログラム ID を付けます。

N

メッセージログの出力要求元のプログラム ID を付けません。

●log_notify_out=Y|N

～《N》

メッセージログ通知機能を使用するかどうかを指定します。

Y

メッセージログ通知機能を使用します。

N

メッセージログ通知機能を使用しません。

Y を指定したときだけ、以下のオペランド (log_notify_xxxx) の指定が有効になります。

●log_notify_allno=Y|N

～《N》

メッセージログ通知機能を使用するときに、メッセージログのシステム内の通番を付けるかどうかを指定します。

Y

メッセージログのシステム内の通番を付けます。

N

メッセージログのシステム内の通番を付けません。

●log_notify_prcid=Y|N

～《N》

メッセージログ通知機能を使用するときに、要求元のプロセスのプロセス ID を付けるかどうかを指定します。

Y

要求元のプロセスのプロセス ID を付けます。

N

要求元のプロセスのプロセス ID を付けません。

●log_notify_prcno=Y|N

～《N》

メッセージログ通知機能を使用するときに、メッセージログのプロセス内の通番を付けるかどうかを指定します。

Y

メッセージログのプロセス内の通番を付けます。

N

メッセージログのプロセス内の通番を付けません。

●log_notify_sysid=Y|N

～《Y》

メッセージログ通知機能を使用するときに、OpenTP1 識別子を付けるかどうかを指定します。

Y

OpenTP1 識別子を付けます。

N

OpenTP1 識別子を付けません。

●log_notify_date=Y|N

～《Y》

メッセージログ通知機能を使用するときに、メッセージログの出力要求時の日付を付けるかどうかを指定します。

Y

メッセージログの出力要求時の日付を付けます。

N

メッセージログの出力要求時の日付を付けません。

●log_notify_time=Y|N

～《Y》

メッセージログ通知機能を使用するときに、メッセージログの出力要求時の時刻を付けるかどうかを指定します。

Y

メッセージログの出力要求時の時刻を付けます。

N

メッセージログの出力要求時の時刻を付けません。

●log_notify_hostname=Y|N

～《Y》

メッセージログ通知機能を使用するときに、メッセージログの出力要求元のホスト名を付けるかどうかを指定します。

Y

メッセージログの出力要求元のホスト名を付けます。

N

メッセージログの出力要求元のホスト名を付けません。

●log_notify_pgmid=Y|N

～《Y》

メッセージログ通知機能を使用するときに、メッセージログの出力要求元のプログラム ID を付けるかどうかを指定します。

Y

メッセージログの出力要求元のプログラム ID を付けます。

N

メッセージログの出力要求元のプログラム ID を付けません。

●log_jerr_rint=メッセージログ出力抑止回数

～〈符号なし整数〉((1～65536))《32》

JP1 イベントサービス機能へメッセージログ出力中に、エラーが発生した場合のメッセージログ出力抑止回数を指定します。

JP1 イベントサービス機能へメッセージログ出力中に、エラーが発生した場合、エラー発生後のメッセージログ出力は、log_jerr_rint オペランドに指定した回数分抑止し、再び出力を試みます。

●log_syslog_out=syslog 出力レベル

～〈符号なし整数〉((0～2))《1》

ログファイルへ出力される全メッセージについて、syslog へ出力要求するレベル（種類）を指定します。

0

メッセージを syslog へ出力要求しません。

1

メッセージの種類が-E および-W の場合だけ syslog へ出力要求します。

2

全メッセージを syslog へ出力要求します。

1 または 2 の指定は、環境変数 DCSYSLOGOUT に 1 が指定されている場合に有効となります。

以降に指定する log_syslog_xxxx オペランドは、log_syslog_out オペランドに 1 または 2 を指定した場合に有効となります。

このオペランドを指定することで、UAP から発行される dc_logprint 関数で出力するメッセージを制御できます。ただし、それ以外のメッセージについては完全には制御できません。そのため、環境変数 DCSYSLOGOUT に 1 を指定した場合、このオペランドに 0 または 1 を指定しても、抑止対象のメッセージが syslog へ出力されることがあります。

syslog に出力されるメッセージを完全に制御する場合、syslogd の構成ファイルの設定を変更する必要があります。

OpenTP1 では、次の情報を付加して syslog へ出力要求を行います。syslog への出力を完全に制御する場合は、これらの情報を基に syslogd の構成ファイルの設定を変更してください。

- syslog の facility : LOG_USER
- syslog の level
 - メッセージの種類が-E の場合 : LOG_ERR
 - メッセージの種類が-W の場合 : LOG_WARNING
 - メッセージの種類が-I の場合 : LOG_INFO
 - メッセージの種類が-R の場合 : LOG_NOTICE

syslog へのメッセージ出力可否を、OS 別に次に示します。

OS	syslog への出力	
	日本語環境	英語環境
AIX	○	○
HP-UX (IPF)	○	○
HP-UX (PA-RISC)	×	○
Linux	○	○
Solaris	×	○
Windows	—	—

(凡例)

○：出力できます。

×：出力できません。

－：対象外です。

●log_syslog_allno=Y|N

～《N》

syslog 出力機能を使用するときに、メッセージログのシステム内の通番を付けるかどうかを指定します。

Y

メッセージログのシステム内の通番を付けます。

N

メッセージログのシステム内の通番を付けません。

●log_syslog_prcid=Y|N

～《N》

syslog 出力機能を使用するときに、要求元のプロセスのプロセス ID を付けるかどうかを指定します。

Y

要求元のプロセスのプロセス ID を付けます。

N

要求元のプロセスのプロセス ID を付けません。

●log_syslog_prcno=Y|N

～《N》

syslog 出力機能を使用するときに、メッセージログのプロセス内の通番を付けるかどうかを指定します。

Y

メッセージログのプロセス内の通番を付けます。

N

メッセージログのプロセス内の通番を付けません。

●log_syslog_sysid=Y|N

～《N》

syslog 出力機能を使用するときに、OpenTP1 識別子を付けるかどうかを指定します。

Y

OpenTP1 識別子を付けます。

N

OpenTP1 識別子を付けません。

●**log_syslog_date=Y|N**

～《N》

syslog 出力機能を使用するときに、メッセージログの出力要求時の日付を付けるかどうかを指定します。

Y

メッセージログの出力要求時の日付を付けます。

N

メッセージログの出力要求時の日付を付けません。

●**log_syslog_time=Y|N**

～《N》

syslog 出力機能を使用するときに、メッセージログの出力要求時の時刻を付けるかどうかを指定します。

Y

メッセージログの出力要求時の時刻を付けます。

N

メッセージログの出力要求時の時刻を付けません。

●**log_syslog_hostname=Y|N**

～《N》

syslog 出力機能を使用するときに、メッセージログの出力要求元のホスト名を付けるかどうかを指定します。

Y

メッセージログの出力要求元のホスト名を付けます。

N

メッセージログの出力要求元のホスト名を付けません。

●**log_syslog_pgmid=Y|N**

～《N》

syslog 出力機能を使用するときに、メッセージログの出力要求元のプログラム ID を付けるかどうかを指定します。

Y

メッセージログの出力要求元のプログラム ID を付けます。

N

メッセージログの出力要求元のプログラム ID を付けません。

●log_syslog_append_nodeid=Y|N

～《N》

syslog 出力機能を使用するときに、ノード識別子を付けるかどうかを指定します。

Y

ノード識別子を付けます。

N

ノード識別子を付けません。

●log_syslog_elist=syslog 失敗リストの要素数

～〈符号なし整数〉((0～65536))《0》

ログサービスが syslog ファイルへの出力に失敗した場合に、該当するメッセージをリトライタイミングまで保管するメモリキューの要素数を指定します。

syslog 失敗リストが満杯になった状態で、新たにメッセージを保管する必要がある場合は、保管されている最も古いメッセージを削除したあと、新しいメッセージを保管します。

1 エLEMENTのサイズは 512 バイトです。

このオペランドは、1 以上の値を指定し、log_syslog_out オペランドでも 1 以上の値を指定した場合だけ、有効になります。

syslog 失敗リストに保管されているメッセージを定期的に出力する間隔は、log_syslog_elist_rint オペランドで指定します。

この機能は、AIX および Linux で使用できます。なお、Linux でこの機能を使用する場合は、拡張 SYSLOG 機能※が必要となります。拡張 SYSLOG 機能がインストールされていない場合や、拡張 SYSLOG 機能が動作しない環境では、この機能は使用できません。

注※

拡張 SYSLOG 機能はサポートサービス (SD-LS100-FR1N1 または SD-LS200-FR1N1) で提供するプログラムです。

●log_syslog_elist_rint=syslog 失敗リストを定期的に出力する間隔

～〈符号なし整数〉((0～65536))《0》(単位: 秒)

syslog 失敗リストに保管されているメッセージを、定期的に syslog ファイルに出力するための間隔を指定します。

このオペランドで 0 を指定した場合、syslog 失敗リストに保管されているメッセージを syslog ファイルに定期的に出力することはできません。

このオペランドは、log_syslog_out オペランドおよび log_syslog_elist オペランドで 1 以上の値を指定した場合だけ有効になります。

この機能は、AIX および Linux で使用できます。なお、Linux でこの機能を使用する場合は、拡張 SYSLOG 機能※が必要となります。拡張 SYSLOG 機能をインストールしていない場合や、拡張 SYSLOG 機能が動作しない環境では、この機能は使用できません。

注※

拡張 SYSLOG 機能はサポートサービス (SD-LS100-FR1N1 または SD-LS200-FR1N1) で提供するプログラムです。

●log_syslog_synchro=Y|N

～《N》

syslog 出力機能を使用しているときにログサーバへの通信に失敗した場合、該当するメッセージログを関連オペランド(log_syslog_xxxx)に従ったフォーマットで syslog に出力するかどうかを指定します。

ただし、log_syslog_allno オペランドに限り無効であり、指定した場合は空白になります。

Y

関連オペランドに従ったフォーマットで出力します。

N

関連オペランドに従ったフォーマットで出力しません。

●log_audit_out=Y|N

～《N》

監査ログ機能を使用するかどうかを指定します。

Y

監査ログ機能を使用します。

N

監査ログ機能を使用しません。

Y を指定したときだけ、以降のオペランド (log_audit_xxxx) の指定が有効になります。

●log_audit_path=監査ログファイルの出力先ディレクトリ

～〈1～63 文字のパス名〉\$DCDIR/auditlog》

監査ログ機能を使用する場合に、監査ログファイルを出力するディレクトリを絶対パスで指定します。ただし、共有ディスク上のディレクトリは指定しないでください。なお、最下層のディレクトリは、dcauditsetup コマンドによって作成されますが、上位ディレクトリは事前に準備しておく必要があります。

このオペランドで指定したパスを構成する各ディレクトリのアクセス権限は、ユーザ、グループ、およびその他のユーザのすべてに実行権限が必要です。実行権限がない場合、監査ログの出力でエラーになることがあります。

このオペランドに指定するディレクトリには、監査ログファイルを格納できるディスク容量が必要です。監査ログファイルを格納するディレクトリに必要なディスク容量の計算式を次に示します。

$$\text{ディスク容量 (単位: メガバイト)} = \text{監査ログファイルの最大サイズ} \times \text{監査ログファイルの最大数}$$

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

●log_audit_size=監査ログファイルの最大サイズ
～ 〈符号なし整数〉 ((1～2047)) 《10》 (単位: メガバイト)

監査ログ機能を使用する場合に、監査ログファイルの最大サイズを指定します。監査ログファイルのサイズが指定値に達した場合には、出力先ファイルの切り替えを行います。

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

●log_audit_count=監査ログファイルの最大数
～ 〈符号なし整数〉 ((1～256)) 《2》

監査ログ機能を使用する場合に、監査ログファイルの最大数を指定します。

監査ログの出力先ファイルを切り替える時点で、ファイル数（ログファイル数+バックアップログファイル数）がこのオペランドの指定値を超える場合は、最も古いファイルを削除します。

このオペランドの指定値とバックアップログファイル名の対応を次の表に示します。

表 3-6 log_audit_count オペランドの指定値とバックアップログファイル名

log_audit_count オペランドの指定値	バックアップログファイル名
1	バックアップログファイルは作成されません。
2～256	audit001.log～audit255.log

このオペランドの指定値が大きいほど出力先ファイル切り替え時のオーバーヘッドが大きくなるため注意してください。

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

●**log_audit_message=監査ログを取得する項目のメッセージ ID** [監査ログを取得する項目のメッセージ ID] ...

～ 〈符号なし整数〉 ((33400～99999))

監査ログを取得する項目のメッセージ ID を 33400～99999 の範囲で指定します。指定できるメッセージ ID は最大 2048 個です。

このオペランドで指定できるメッセージ ID については、「[付録 C 監査イベントを取得する定義](#)」を参照してください。

なお、このオペランドは、ユーザサービス定義、rap リスナーサービス定義、およびユーザサービスデフォルト定義でも指定できます。指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義または rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. ログサービス定義

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

●**watch_time=最大応答待ち時間**

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かることがあります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しないことがあります。

省略した場合は、システム共通定義の watch_time オペランドの値を仮定します。

このオペランドは、システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

なし。

putenv 形式のオペランド

●TZ タイムゾーン

～〈文字列〉

環境変数 TZ を設定します。

タイムゾーンとは、時刻などを表示する場合の環境を設定するものです。

日本の標準時間を表示する場合は、タイムゾーンに JST-9 を指定します。

このオペランドで指定したタイムゾーンと、コマンドを実行する環境のタイムゾーンは一致させてください。一致していない場合、syslog の時刻が正しく出力されません。

このオペランドを変更する場合は、dcsetup コマンド (-dn) ※で一度 OS への登録を削除し、dcsetup コマンド※で再度登録してください。

注※

dcreset コマンドでは変更が反映されません。

●DCSYSLOGOUT 1|1 以外の文字列

syslog ヘッセージを出力するかどうかを指定します。

syslog ヘッセージを出力する場合は、環境変数 DCSYSLOGOUT に 1 を指定します。1 以外の文字列を指定した場合や、このオペランドの指定を省略した場合は、syslog ヘッセージを出力しません。

出力制御については、log_syslog_out オペランドの説明を参照してください。

注意事項

- 1. syslog に出力できるメッセージ長は、511 バイトまでです。512 バイト以降のメッセージは切り捨てられます。なお、HP-UX(PA-RISC)版および Solaris 版については、255 バイトまでです。
- 2. ログサービス定義のオペランドによっては Y を指定すると、ユーザメッセージの前にヘッダ情報が付きます。このため、ヘッダ情報の分、syslog に出力できるユーザメッセージのデータ量は少なくなります。

Y を指定した場合にヘッダ情報が付与されるオペランドと、ヘッダ情報として付与されるバイト数を次の表に示します。なお、バイト数には区切りのスペースを含みます。さらにヘッダ情報には、メッセージ ID の 12 バイトが付与されます。

オペランド	ヘッダ情報として付与されるバイト数
log_syslog_allno	8
log_syslog_prcid	6 または 11 (11 は AIX 版の場合だけ)
log_syslog_prcno	8

オペランド	ヘッダ情報として付与されるバイト数
log_syslog_sysid	3
log_syslog_date	11
log_syslog_time	9
log_syslog_hostname	9
log_syslog_pgmid	4
log_syslog_append_nodeid	5

3.512 バイト以降のメッセージが切り捨てられた場合、syslog に出力されている最後の文字が複数バイトのときは、正しく表示されないことがあります。

4. ログサービスに対して一度に大量のメッセージの出力を要求すると、出力先のファイル種別や関連製品の特性※によっては、一時的にバッファ不足などのエラーとなり、メッセージが出力できない場合があります。この場合、ログサービスとしては、一時的に出力先を標準エラー出力に切り替えたり、メッセージの出力を抑止したりするなどの制御をします。このため、一定期間、目的のファイルにメッセージが出力されないことがあります。

一方、メッセージの出力頻度や全体のメッセージ量は、システムのスループットにも依存します。このため、一般的に、CPU や関連するハードウェアの性能が上がると単位時間当たりのメッセージ出力量も増えることがあります。

また、運用面では、一度に起動または停止するユーザサーバの数が非常に多い場合や、UAP からのメッセージが多量に出力されたり一時期に集中して出力されたりするような場合は、メッセージの量が問題になることがあります。

このような場合は、メッセージの出力量を削減したり、メッセージの出力先を変更したりしてください。

注※

製品によっては、処理できる最大の処理件数が明示されているものもあります。出力先に応じ、関連製品の仕様を確認してください。

5. メッセージログ通知機能を使用する場合も、4.の注意事項と同様ですのでご注意ください。ログサービスから UAP へのメッセージログの送信で一時的にバッファ不足などのエラーが発生した場合、ログサービスとしてはメッセージを再送しないで次のメッセージを処理します。このため、UAP で受信できる処理能力を超えて一度に多量のメッセージ出力要求がログサービスに対して行われるような場合は、メッセージが消失するおそれがあります。

マルチノード構成定義

形式

set 形式

```
[set dcmstart_watch_time=dcmstart打ち切り時間]
[set dcmstop_watch_time=dcmstop打ち切り時間]
[set watch_time=最大応答待ち時間]
```

コマンド形式

```
{{dcmarea {-m マルチノードエリア識別子|-g マルチノードサブエリア識別子}
-w ノード識別子 [, ノード識別子] ...}}
```

機能

マルチノードエリアまたはマルチノードサブエリアの構成を定義します。

マルチノード連携制御機能のうち、次の機能に関するサービス要求の最大応答待ち時間を定義します。

- dcmstart コマンド
- dcmstop コマンド
- dcndls コマンド

説明

set 形式のオペランド

●dcmstart_watch_time=dcmstart 打ち切り時間

～ 〈符号なし整数〉 ((0～65535)) 《600》 (単位：秒)

dcmstart コマンドの実行から処理を打ち切るまでの時間を指定します。

指定時間を過ぎても開始処理中の OpenTP1 ノードがある場合は、メッセージを出力してその OpenTP1 ノードの監視と dcmstart コマンドは打ち切られます。

0 を指定した場合は、すべての OpenTP1 ノードが開始処理を完了するか、開始できないことを確認するまで監視が続きます。

1～59 を指定した場合は、60 を仮定します。

●dcmstop_watch_time=dcmstop 打ち切り時間

～ 〈符号なし整数〉 ((0～65535)) 《600》 (単位：秒)

dcmstop コマンドの実行から処理を打ち切るまでの時間を指定します。

指定時間を過ぎても開始中の OpenTP1 ノードがある場合は、メッセージを出力してその OpenTP1 ノードの監視と dcmstop コマンドは打ち切られます。

0 を指定した場合は、すべての OpenTP1 ノードが開始処理を完了するか、開始できないことを確認するまで監視が続きます。

●watch_time=最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しない場合があります。

省略した場合は、システム共通定義の watch_time オペランドの値を仮定します。

このオペランドは、マルチノード連携制御機能を使用する OpenTP1 ノード以外は指定する必要はありません。

このオペランドは、デフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

このオペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

次ページ以降に記述しています。

名称

マルチノードエリア／サブエリアに属するノード識別子を指定する

形式

```
{ {dcmarea  {-m マルチノードエリア識別子|-g マルチノードサブエリア識別子}
           -w ノード識別子  [, ノード識別子] ...} }
```

機能

同一のマルチノードエリアまたはマルチノードサブエリアに属する OpenTP1 ノードの構成を定義します。

次のことはオンライン中にできません。

- マルチノードエリアまたはマルチノードサブエリアの設定
 - マルチノードエリアまたはマルチノードサブエリアへの OpenTP1 ノードの追加および削除
- 次に示す前提条件に従ってください。この条件を満たしていない場合は、マルチノード機能の動作を保証しません。

(前提条件)

- マルチノードエリア識別子は 1 個指定します。
- マルチノードエリアに属する OpenTP1 ノードの個数は 128 以下です。
- 同一マルチノードエリア（サブエリア）識別子に対して複数のコマンドを指定した場合、それら複数のコマンドの論理和とします。
- ノード識別子は、システム共通定義に指定したものを指定します。
- ホストとホスト名が 1 対 1 に対応しているホストを、マルチ OpenTP1 構成にする場合、そのホスト内の各 OpenTP1 ノードは、別々のマルチノードエリアおよびマルチノードサブエリアに属します。
- マルチノードサブエリアに属する OpenTP1 ノードは、必ずマルチノードエリアに属します。
- マルチノード構成定義には、全マルチノードサブエリアの構成を記述します。
- マルチノードエリアまたはマルチノードサブエリアを構成する各 OpenTP1 ノードのマルチノード構成定義は一致させます。

コマンド引数

●-m マルチノードエリア識別子

～ 〈1～8 文字の識別子〉

マルチノードエリア識別子を指定します。

●-g マルチノードサブエリア識別子

～ 〈1～8 文字の識別子〉

マルチノードサブエリア識別子を指定します。

●-w ノード識別子 [, ノード識別子] ...

～ 〈4 文字の識別子〉

マルチノードエリアまたはマルチノードサブエリアに属する OpenTP1 ノードのノード識別子を指定します。各 OpenTP1 ノードのシステム共通定義に指定したノード識別子でなくてはなりません。

マルチノード物理定義

形式

set 形式

なし。

コマンド形式

```
{{dcprcport -w ノード識別子  
            -h ホスト名  
            -p ポート番号}}
```

機能

マルチノード構成定義に定義した各 OpenTP1 ノードについて、OpenTP1 ノードがあるホスト名と、そのノードでマルチノード連携制御機能が使用するポート番号を定義します。

説明

set 形式

なし。

コマンド形式

次ページ以降に記述しています。

名称

マルチノードのホスト名とポート番号を指定する

形式

```
{{dcprcport -w ノード識別子  
            -h ホスト名  
            -p ポート番号}}
```

機能

マルチノード構成定義に定義した各 OpenTP1 ノードについて、OpenTP1 ノードがあるホスト名と、そのノードでマルチノード連携制御機能が使用するポート番号を定義します。

系切り替え機能を使用している OpenTP1 ノードは、実行系と待機系を別に定義してください。

系切り替え機能を使用していない OpenTP1 ノードは、ノード識別子を重複しないように指定してください。

マルチノード構成定義で定義した OpenTP1 ノードは、必ずこの定義に指定してください。また、この定義に指定できる OpenTP1 ノードは、最大 128 個です。

オプション

●-w ノード識別子

～ 〈4 文字の識別子〉

マルチノード構成定義で定義したノード識別子を指定します。

●-h ホスト名

～ 〈1～64 文字の識別子〉

ノード識別子に対応する OpenTP1 ノードがあるホスト名を指定します。

ホスト名は、/etc/hosts に登録されていなければなりません。

保守用 LAN があるときは、保守用 LAN のホスト名を指定してください。保守用 LAN がないときは、システム共通定義の my_host オペランドで指定したホスト名（省略時は標準ホスト名）を指定してください。

●-p ポート番号

～ 〈符号なし整数〉 ((5001～49999))

ノード識別子に対応するマルチノード連携制御機能で使用するポート番号を指定します。

マルチノード連携制御機能で使用するポート番号は、システム共通定義の `prc_port` オペランドで指定した値を設定してください。

グローバルアーカイブジャーナルサービス定義

形式

set 形式

```
[set jnl_arc_terminate_timeout=待ち合わせ最大時間]
[set max_socket_descriptors=ソケット用ファイル記述子の最大数]
[set jnl_arc_ipc_buff_size=TCP/IPの送受信バッファサイズ]
[set jnl_watch_time=グローバルアーカイブジャーナルサービスの通信応答待ち時間]
[set jnl_message_id_change_level=メッセージIDの変更レベル]
```

コマンド形式

```
jnldfs -a グローバルアーカイブジャーナルサービスのリソースグループ名
[ [, グローバルアーカイブジャーナルサービスの
  リソースグループ名] ...]
```

機能

グローバルアーカイブジャーナルサービスを使用するための実行環境を定義します。この機能を使用するには、TP1/Multi が必要です。

set 形式のオペランド

●jnl_arc_terminate_timeout=待ち合わせ最大時間

～ 〈符号なし整数〉 ((0～3600)) 《0》 (単位：秒)

アーカイブノードを終了させてから、接続しているすべての被アーカイブノードの接続を解除するまでの最大待ち時間を秒単位で指定します。指定時間内に接続が解除された場合は、アーカイブノードは終了します。指定時間を過ぎても接続している被アーカイブノードがある場合は、アーカイブノードは終了しません。

0 を指定した場合は、すべての被アーカイブノードの接続が解除されるまでアーカイブノードの終了処理が続行されます。

●max_socket_descriptors=ソケット用ファイル記述子の最大数

～ 〈符号なし整数〉 ((32～2032))

OpenTP1 制御下のプロセス※でのソケット用に使用するファイル記述子の最大数を指定します。

OpenTP1 制御下のプロセス※は、システムサーバやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数、および通信する他ノードの数によって、このオペランドを変更する必要があります。

注※

MCF サービス（MCF マネジャサービス、MCF 通信サービス、およびアプリケーション起動サービス）以外の OpenTP1 プロセスが対象です。MCF サービスの場合は、「[システムサービス情報定義](#)」および「[システムサービス共通情報定義](#)」を参照してください。

ソケット用ファイル記述子の最大数の算出式を、次に示します。

$\uparrow (\text{グローバルアーカイブジャーナルサービスに対してサービスを要求してくるノード数}^{\ast 1} + \text{システムサービスプロセス数}^{\ast 2}) / 0.8 \uparrow$

$\uparrow \uparrow$: 小数点以下を切り上げます。

注※1

グローバルアーカイブジャーナルサービスに対してサービスを要求してくるノード数とは、次に示す値の合計です。

- 自 OpenTP1 の all_node オペランドに指定したノード名の数
- 自ノード名を、OpenTP1 の all_node オペランドに指定している、他ノードの数

注※2

システムサービスプロセス数とは、自 OpenTP1 内のシステムサービスプロセスの数です。自 OpenTP1 内のシステムサービスプロセスは、rpcstat コマンドで表示されるサーバ名をカウントすることで求められます。rpcstat コマンドで表示されるサーバ名のうち、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造に記載されているシステムサービスプロセスをカウントしてください。

このオペランドの指定が小さいと、OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため、プロセスが KFCA00307-E メッセージを出力して異常終了します。

指定値の優先順位は次のとおりです（1.> 2.）。

1. グローバルアーカイブジャーナルサービス定義
2. システム共通定義

ここで指定を省略した場合、システム共通定義の値を仮定します。

●jnl_arc_ipc_buff_size=TCP/IP の送受信バッファサイズ

～ 〈符号なし整数〉 ((8192～1048576)) 《61440》 (単位：バイト)

被アーカイブノードから転送されるジャーナルを受信するアーカイブジャーナル受信プロセスが使用する TCP/IP の送受信バッファのサイズを指定します。

高速な通信媒体を使用する場合、値を大きくすることで性能を向上できます。このオペランドの値は、各 OS で指定できる TCP/IP のバッファサイズの上限值以下の値を指定してください。また、このオペランドの値は、被アーカイブノード側のジャーナルサービス定義の jnl_arc_ipc_buff_size オペランドの指定値と合わせてください。

●jnl_watch_time=グローバルアーカイブ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)

グローバルアーカイブジャーナルサービスが RPC によってプロセス間で通信する場合の、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。指定時間を過ぎても応答がない場合、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。この場合、OpenTP1 が終了しないことがあります。

特別なチューニングが不要な場合は、このオペランドの内容を変更しないでください。

jnl_watch_time 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※
- jnlardis 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

注※

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

●jnl_message_id_change_level=メッセージ ID の変更レベル

～ 〈符号なし整数〉 ((0～1)) 《0》

グローバルアーカイブジャーナルサービスが出力するメッセージで、重要な情報となるメッセージの種別を変更します。

メッセージログを監視し、メッセージの種類によって管理方法を区別したい場合にこのオペランドを使用してください。

このオペランドによって変更されるのはメッセージ ID だけであり、メッセージの内容は変更されません。

0

メッセージ ID を変更しないで、従来どおりのメッセージ ID で出力します。

1

メッセージ ID を変更し、出力します。

このオペランドの指定値と出力されるメッセージの関係を次に示します。

jnl_message_id_change_level オペランドの指定値	0	1
出力されるメッセージ ID	KFCA01224-I	KFCA01230-W
	KFCA01250-I	KFCA01231-W

コマンド形式

次ページ以降に記述しています。

名称

グローバルアーカイブジャーナルサービスのリソースグループ名の指定

形式

```
jnldfsv -a グローバルアーカイブジャーナルサービスのリソースグループ名  
[ [ , グローバルアーカイブジャーナルサービスの  
リソースグループ名 ] ... ]
```

機能

グローバルアーカイブジャーナルサービスのリソースグループ名を指定します。

オプション

●-a グローバルアーカイブジャーナルサービスのリソースグループ名

～ 〈1～8 文字の識別子〉

グローバルアーカイブジャーナルサービスのリソースグループ名（アーカイブジャーナルサービス定義のファイル名）を指定します。

リソースグループ名は 16 個まで指定できます。リソースグループ名は重複するとエラーとなります。

アーカイブジャーナルサービス定義

形式

set 形式

```
[set jnl_dual=Y|N]
[set jnl_singleoperation=Y|N]
[set jnl_rerun_swap=Y|N]
[set jnl_max_file_dispersion=並列アクセス化する場合の最大分散数]
[set jnl_min_file_dispersion=並列アクセス化する場合の最小分散数]
[set jnl_unload_check=Y|N]
[set jnl_arc_max_datasize=アーカイブ時の転送データの最大長]
```

コマンド形式

```
{{{jnladdfg -g アーカイブジャーナルファイル
             のファイルグループ名 [ONL] }}
 {{jnladdpf -g アーカイブジャーナルファイルの
             ファイルグループ名
             [-e 要素ファイル名]
             [-a 物理ファイル名]
             [-b 物理ファイル名] }}}}
```

機能

グローバルアーカイブジャーナルサービスを使用するための実行環境を定義します。この機能を使用するには、TP1/Multi が必要です。

set 形式のオペランド

●jnl_dual=Y|N

～《N》

アーカイブジャーナルファイルを二重化するかどうかを指定します。

Y

二重化します。

N

二重化しません。

●jnl_singleoperation=Y|N

～《N》

jnl_dual オペランドに Y を指定した場合に、片系クローズ状態のファイルグループを使用するかどうかを指定します。

Y

片系クローズのファイルグループでも使用します。

N

片系クローズのファイルグループは使用しません。

●jnl_rerun_swap=Y|N

～《N》

グローバルアーカイブジャーナルサービスのあるノードがリランしたとき、アーカイブジャーナルファイルをスワップするかどうかを指定します。

Y

スワップします。

N

スワップしません。

●jnl_max_file_dispersion=並列アクセス化する場合の最大分散数

～〈符号なし整数〉((1～8))《1》

アーカイブジャーナルを並列アクセス化する場合の最大分散数を指定します。

一つのファイルグループで使用する最大要素ファイル数を指定してください。1 を指定した場合、または指定を省略した場合は、jnl_min_file_dispersion オペランドの指定値は無効になります。

●jnl_min_file_dispersion=並列アクセス化する場合の最小分散数

～〈符号なし整数〉((1～8))《1》

アーカイブジャーナルを並列アクセス化する場合の最小分散数を指定します。

ファイルグループを構成する要素ファイルが障害などで使用できなくなった場合でも、現用として使用可能とする最小要素ファイル数を指定してください。

使用可能な要素ファイル数が最小要素ファイル数未満になった場合、ファイルグループは使用できなくなります。並列アクセス化する場合の最小分散数は次の範囲で指定してください。

- 最大分散数 ≥ 最小分散数 ≥ 1

●jnl_unload_check=Y|N

～《Y》

スワップ先のファイルグループを決定する時に、アンロード待ち状態をチェックするかどうかを指定します。

Y

アンロード待ち状態をチェックします。

N

アンロード待ち状態をチェックしません。

Y を指定した場合、OpenTP1 が強制終了、または異常終了したあとに OpenTP1 を強制正常開始したとき、前回のオンラインで現用として使用されていたジャーナルファイルはクローズ状態となります。jnlunlfg コマンドまたは jnlchgfg コマンドでファイルグループのステータスを変更するまで、ジャーナルファイルは使用できません。

N を指定した場合、アンロードしていないアーカイブジャーナルは失われます。そのため、通常は、テストなどでジャーナルの運用（稼働統計情報の編集など）を必要としない場合に指定します。また、ジャーナルが失われるため、アンロードジャーナルファイルを指定するコマンドが、一部使用できなくなります。詳細は、マニュアル「OpenTP1 運用と操作」を参照してください。

●jnl_arc_max_datasize=アーカイブ時の転送データの最大長

～〈符号なし整数〉((1020～8188))《1020》(単位：キロバイト)

被アーカイブノードからアーカイブされるジャーナルデータの最大転送データ長を指定します。このオペランドの指定値は、該当するリソースグループに接続する被アーカイブノードの中で最も大きな転送データ長の値を指定してください。

また、このオペランドに 1020 より大きな値を指定する場合は、次の計算式によって求めた値より大きな値をシステム共通定義の rpc_max_message_size オペランドに指定してください。

$$\text{rpc\_max\_message\_size} \geq \uparrow (\text{jnl\_arc\_max\_datasize} \times 1024 + 4096) / 1048576 \uparrow$$

計算式で求められた値より rpc_max_message_size の指定値が小さい場合は、OpenTP1 の開始時に KFCA01113-E メッセージの理由コード 425 を出力して OpenTP1 の開始処理を中止します。

コマンド形式

次ページ以降に記述しています。

jnladdfg

名称

アーカイブジャーナルファイルのファイルグループ名と属性の定義

形式

```
{{jnladdfg  -g  アーカイブジャーナルファイルのファイルグループ名  
            [ONL]                                     }}
```

機能

アーカイブジャーナルファイルのファイルグループ名と属性を定義します。

jnladdfg 定義コマンドは、アーカイブジャーナルサービス定義内に最低 2 個、最大 256 個指定できます。

ただし、ONL を指定した jnladdfg 定義コマンドの場合は、最低 2 個必要です。

なお、ファイルグループ名は、アーカイブジャーナルサービス定義内で一意でなければなりません。

オプション

●-g アーカイブジャーナルファイルのファイルグループ名

～ 〈1～8 文字の識別子〉

アーカイブジャーナルファイルのファイルグループ名を指定します。

●ONL

このファイルグループを、グローバルアーカイブジャーナルサービスの開始と同時に**オープン状態**にする場合に指定します。

省略した場合は、グローバルアーカイブジャーナルサービスの開始時点には、クローズ状態の予約のファイルグループになります。このファイルグループを使用する場合は、jnlopnfg コマンドでオープンしなければなりません。

名称

アーカイブジャーナルファイルのファイルグループを構成する物理ファイルの定義

形式

```
{{jnladdpf  -g アーカイブジャーナルファイルのファイルグループ名
            [-e 要素ファイル名]
            -a 物理ファイル名
            [-b 物理ファイル名] }}
```

機能

アーカイブジャーナルファイルのファイルグループを構成する、要素ファイルおよび物理ファイルを定義します。

jnladdpf 定義コマンドは、ファイルグループに対して最低 1 個、最大 8 個指定できます。

なお、物理ファイル名は、OpenTP1 システム内で一意でなければなりません。

オプション

●-g アーカイブジャーナルファイルのファイルグループ名

～ 〈1～8 文字の識別子〉

アーカイブジャーナルファイルのファイルグループの名称を指定します。

ここで指定するファイルグループ名は、あらかじめ jnladdfg 定義コマンドで定義しておきます。

●-e 要素ファイル名

～ 〈1～8 文字の識別子〉

このファイルグループを構成する要素ファイル名を指定します。

要素ファイル名はアーカイブジャーナルサービス定義内で一意でなければなりません。

並列アクセス化 (jnl_max_file_dispersion オペランドに 2 以上を指定) をする場合は、必ずこのオプションを指定してください。並列アクセス化をしない場合は、指定する必要はありません。

並列アクセス化をする場合に-e オプションを省略すると、定義解析でエラーになります。一つのファイルグループに指定する要素ファイル数は次の範囲で指定してください。

```
jnl_max_file_dispersionオペランドの指定値 ≥ 要素ファイル数 ≥ jnl_min_file_dispersionオペランドの指定値
```

ここで指定する要素ファイル数が `jnl_max_file_dispersion` オペランドの指定値を超えた場合、超えた分はオンラインでは使用されません。また、ここで指定する要素ファイル数が `jnl_min_file_dispersion` オペランドの指定値に満たない場合、そのファイルグループは使用できません。

●-a 物理ファイル名

～ 〈パス名〉

この要素ファイルを構成する物理ファイル名を絶対パス名で指定します。ただし、環境変数は使用できません。この物理ファイルはジャーナル関係のファイル用として作成した OpenTP1 ファイルを使用します。

set 形式の `jnl_dual` でファイルの二重化を指定した場合は、ここに A 系ファイルの物理ファイル名を指定してください。

●-b 物理ファイル名

～ 〈パス名〉

set 形式の `jnl_dual` でファイルの二重化を指定した場合に、B 系ファイルの物理ファイル名を絶対パス名で指定します。ただし、環境変数は使用できません。この物理ファイルはジャーナル関係のファイル用として作成した OpenTP1 ファイルを使用します。

二重化の指定をしていない場合に -a および -b の物理ファイル名を指定すると、-a の方が有効となります。また、二重化する場合もしない場合も、-b の物理ファイル名だけを指定すると、コマンド全体が無視されます。

DAM サービス定義

形式

set 形式

```
[set dam_update_block=更新する最大ブロック数]
[set dam_added_file=オンライン中に追加する最大論理ファイル数]
[set dam_update_block_over=flush|error]
[set dam_message_level=1|0]
[set dam_tran_process_count=同時に実行するトランザクション
                             ブランチの数]
[set dam_cache_size | dam_cache_size_fix=バッファ領域長]
[set dam_cache_attribute=free|fixed]
[set dam_io_interval=実行間隔時間]
[set dam_transaction_access=global|branch]
[set dam_io_error_occur=stop|continue]
[set dam_cache_reuse_from=last|first]
[set dam_default_cache_num=キャッシュブロック再利用境界デフォルト値]
[set dam_ex_refer_read=none|stay]
[set dam_max_block_size=最大ブロック長]
[set dam_kb_size=1000|1024]
[set watch_time=最大応答待ち時間]
```

コマンド形式

```
[damcache 論理ファイル名 キャッシュブロック再利用境界値]
[damchmt 論理ファイル名 キャッシュブロック数のしきい値]
{{damfile [-{d|n [-f]}] [-c] 論理ファイル名 物理ファイル名}}
```

機能

ファイル回復機能付き直接アクセスファイルを管理するための実行環境を定義します。

説明

set 形式のオペランド

●dam_update_block=更新する最大ブロック数

～ 〈符号なし整数〉 ((1～32768)) 《8》

更新するブロックの最大値を指定します。DAM サービスは、この指定値によってシステムサービス用共用メモリプール（動的共用メモリ）上に DAM サービス分のメモリを確保します。そのため、算出式で求めた値よりも大きな値を指定した場合は、システムサービス用共用メモリプール（動的共用メモリ）を圧迫することがあります。

このオペランドに指定する値の算出方法を次に示します。

回復対象の DAM ファイルにアクセスする場合

同一のトランザクション内で発生する次の A, B, および C を合計した値をトランザクションごとに算出し、その中で最大の値をこのオペランドに指定します。

- A : dc_dam_read 関数で更新目的で入力するブロック数
- B : dc_dam_write 関数で出力するブロック数
- C : dc_dam_read 関数で排他あり参照指定で入力するブロック数（ただし、DAM サービス定義の dam_ex_refer_read オペランドに stay を指定した場合だけ）

トランザクションの開始から完了まで更新目的で入出力する場合の算出方法を次に示します

表 3-7 トランザクションの開始から完了まで更新目的で入出力する場合の算出方法

アクセス関数の実行順序	アクセスブロック数	A	B	C	算出値 (A+B+C)
トランザクション開始	—	0	0	0	0
dc_dam_read 関数 (更新目的)	2	2	0	0	2
dc_dam_rewrite 関数	2	0	0	0	0
dc_dam_write 関数	3	0	3	0	5
トランザクション完了	—	0	0	0	0
最終値		2	3	0	5

(凡例)

— : 該当しません

この例では、算出値 (A+B+C) が 5 となるため、dam_update_block オペランドに 5 以上の値を指定してください。

回復対象外の DAM ファイルにアクセスする場合

一つの UAP で発生する次の D の値、または一つの UAP で発生する次の E および F を合計した値から G を引いた値を UAP ごとに算出し、その中で最大の値をこのオペランドに指定します。

- D : UAP で実行する一つの DAM アクセス関数でアクセスするブロック数の最大値
- E : dc_dam_read 関数で更新目的で入力するブロック数
- F : E に示す dc_dam_read 関数を実行したあとに、dc_dam_write 関数で出力するブロック数
- G : F に示す dc_dam_write 関数を実行するまでに、E に示した dc_dam_read 関数で更新目的で入力したブロックに対して、dc_dam_rewrite 関数を実行したブロック数

dc_dam_write 関数だけ実行する場合の算出方法を次に示します。

表 3-8 dc_dam_write 関数だけ実行する場合の算出方法

アクセス関数の実行順序	アクセスブロック数	D	E	F	G	算出値 (E+F-G)
dc_dam_start 関数	—	0	0	0	0	0

アクセス関数の実行順序	アクセスブロック数	D	E	F	G	算出値 (E+F-G)
dc_dam_write 関数	3	3	0	0	0	0
dc_dam_write 関数	5	5	0	0	0	0
dc_dam_end 関数	—	0	0	0	0	0
最終値		5	0	0	0	0

(凡例)

—：該当しません

この例では、D が 5 となり、算出値 (E+F-G) は D よりも小さい 0 になるため、dam_update_block オペランドには 5 以上の値を指定してください。

dc_dam_read 関数（更新目的）と dc_dam_write 関数を実行した場合の算出方法を次に示します。

表 3-9 dc_dam_read 関数（更新目的）と dc_dam_write 関数を実行した場合の算出方法

アクセス関数の実行順序	アクセスブロック数	D	E	F	G	算出値 (E+F-G)
dc_dam_start 関数	—	0	0	0	0	0
dc_dam_read 関数（更新目的）	3	3	3	0	0	3
dc_dam_write 関数	5	5	0	5	0	8
dc_dam_end 関数	—	0	0	0	0	0
最終値		5	3	5	0	8

(凡例)

—：該当しません

この例では、D が 5 となり、算出値 (E+F-G) が D よりも大きい 8 になるため、dam_update_block オペランドには 8 以上の値を指定してください。

dc_dam_read 関数（更新目的）実行前に dc_dam_write 関数を実行した場合の算出方法を次に示します。

表 3-10 dc_dam_read 関数（更新目的）実行前に dc_dam_write 関数を実行した場合の算出方法

アクセス関数の実行順序	アクセスブロック数	D	E	F	G	算出値 (E+F-G)
dc_dam_start 関数	—	0	0	0	0	0
dc_dam_write 関数	8	8	0	0	0	0
dc_dam_read 関数（更新目的）	4	0	4	0	0	3
dc_dam_write 関数	5	0	0	5	0	9
dc_dam_end 関数	—	0	0	0	0	0

アクセス関数の実行順序	アクセスブロック数	D	E	F	G	算出値 (E+F-G)
最終値		8	4	5	0	9

(凡例)

－：該当しません

この例では、D が 8 となり、算出値 (E+F-G) が D よりも大きい 9 になるため、dam_update_block オペランドには 9 以上の値を指定してください。

dc_dam_read 関数（更新目的）実行後、一部のブロックに対して dc_dam_rewrite 関数を実行した場合の算出方法を次に示します。

表 3-11 dc_dam_read 関数（更新目的）実行後、一部のブロックに対して dc_dam_rewrite 関数を実行した場合の算出方法

アクセス関数の実行順序	アクセスブロック数	D	E	F	G	算出値 (E+F-G)
dc_dam_start 関数	－	0	0	0	0	0
dc_dam_read 関数（更新目的）	4	4	4	0	0	4
dc_dam_rewrite 関数	3	0	0	0	3	1
dc_dam_write 関数	5	5	0	5	0	6
dc_dam_end 関数	－	0	0	0	0	0
最終値		5	4	5	3	6

(凡例)

－：該当しません

この例では、D が 5 となり、算出値 (E+F-G) が D よりも大きい 6 になるため、dam_update_block オペランドには 6 以上の値を指定してください。

更新するブロック数が最大ブロック数を超えた場合、アクセス関数（dc_dam_read 関数、dc_dam_write 関数、dc_dam_rewrite 関数）は、DCDAMER_JNLOV※¹ または DCDAMER_ACSOV※² でエラーリターンします。

注※1

COBOL-UAP 作成用プログラムの場合は、ステータスコードに「01613」を設定します。

注※2

COBOL-UAP 作成用プログラムの場合は、ステータスコードに「01648」を設定します。

●dam_added_file=オンライン中に追加する最大論理ファイル数

～〈符号なし整数〉((1～128))《8》

オンライン中に追加する論理ファイルの最大値を指定します。

オンライン実行中には、指定した数まで論理ファイルを追加できます。

●dam_update_block_over=flush|error

～《error》

トランザクションの途中で更新するブロック数が、dam_update_block オペランドで指定した最大ブロック数を超える場合に、アクセス関数をエラーリターンするかどうかを指定します。

flush

アクセス関数をエラーリターンしないで、処理を続行します。ただし、ブロックを更新（dc_dam_rewrite 関数）しないで更新目的を入力（dc_dam_read 関数）したブロックが最大ブロックに達している場合は、エラーリターンします。

flush を指定した場合の注意を次に示します。

- 最大ブロック数を越えたときに、一時的に DAM ファイルへ出力します。このため、アクセス関数の処理時間が増加します。
- DAM ファイルを一時的に更新するため、当該トランザクションがロールバックした場合、一時的な更新を無効にする必要があります。このため、一時的な出力をする直前のブロック情報を DAM の回復用ジャーナル（FJ）に取得します。これによって、取得するジャーナルの量が多くなります。
- 回復用ジャーナル（FJ）に一時的な出力をする直前のブロック情報を設定するため、dc_dam_write 関数に指定したブロックの入力が起こります。ただし、dc_dam_write 関数に指定したブロックが、すでに DAM サービス用共用メモリに存在する場合、ブロックの入力は起こりません。

error

アクセス関数をエラーリターンします。

●dam_message_level=1|0

～《1》

DAM サービスが出力するメッセージのレベルを指定します。

0

すべてのメッセージを出力します。

1

排他エラーのメッセージ（KFCA01610-W）を出力しません。

●dam_tran_process_count=同時に実行するトランザクションブランチの数

～〈符号なし整数〉((0～8192))

DAM ファイルにアクセスするトランザクションブランチを同時に実行する数を指定します。ここで指定した数に合わせて領域が確保されます。

1 以上を指定した場合

オンラインシステムを開始するときに、指定した数の領域を確保します。指定した数の領域を確保できない場合は、確保できた領域だけでオンラインシステムを開始します。

確保した領域は、オンラインシステムが終了すると解放されます。

0 を指定した場合

トランザクションを実行するときに領域を確保します。

確保した領域は、トランザクションが決着すると解放されます。

定義を省略した場合

UAP を起動するときに領域を確保します。

確保した領域は、UAP が終了すると解放されます。

0 を指定した場合や領域が足りなくなった場合は、トランザクションを実行するときに領域を確保するため、処理が遅くなります。

なお、この指定値を大きくすると、共用メモリを圧迫します。使用できるメモリの容量を考えて指定してください。

●dam_cache_size | dam_cache_size_fix=バッファ領域長

～〈符号なし整数〉((10～1000000)) (単位：キロバイト)

システム起動時にオンラインで参照・更新したブロックをスタックしておくためのバッファ領域の大きさを、dam_cache_size オペランドまたは dam_cache_size_fix オペランドに指定します。

dam_cache_size_fix オペランドの指定は、dam_cache_size オペランドの指定より優先されます。

dam_cache_size を指定する場合は、次の「バッファ領域長の仮定値計算式」で示す値より大きい値を指定します。

バッファ領域長の仮定値計算式

バッファ領域長 = A + B

A : $(Mb / 64 + 1) \times 128 \times n \times Tr$

B : $\{\downarrow (A / 576) \downarrow\} \times 32 + 64$

Mb : 次のどちらか大きい値に 8 を加えた値

- DAM サービス定義で定義した論理ファイルの中で、最大のブロック長を持つファイルのブロック長 (damload コマンド, dc_dam_create 関数に指定したブロック長)
- DAM サービス定義の dam_max_block_size オペランドで指定した最大ブロック長

n : DAM サービス定義の dam_update_block オペランドで指定した最大更新ブロック数

Tr : DAM サービス定義の dam_tran_process_count オペランドで指定したトランザクションブランチ数

↓↓ : 小数点以下を切り捨てます。

注

DAM サービス定義の damfile 定義コマンドに -f オプションを指定して、キャッシュレスアクセスの DAM ファイルを使用する場合は、Mb に次のどちらか大きい値に 8 を加えた値を指定してバッファ領域長を計算し、その算出値を dam_cache_size_fix オペランドに指定してください。

- DAM サービス定義で定義したキャッシュレスアクセスではない論理ファイルの中で、最大のブロック長を持つファイルのブロック長 (damload コマンド, dc_dam_create 関数に指定したブロック長)
- DAM サービス定義の dam_max_block_size オペランドの指定値

すべての DAM ファイルをキャッシュレスアクセスとして指定する場合は、dam_cache_size_fix オペランドに 10 を指定してください。

dam_tran_process_count オペランドに 0 を指定した場合は、同時に実行するトランザクションブランチの数として「8」を仮定して計算します。dam_tran_process_count オペランドの指定を省略した場合は、同時に実行するトランザクションブランチの数としてトランザクションサービス定義の trn_tran_process_count オペランドの指定値で計算します。

dam_cache_size オペランドを省略した場合や、DAM サービスが「バッファ領域長の仮定値計算式」を基に自動的に算出した値より小さい値を指定した場合は、DAM サービスが算出した値でバッファ領域を確保します。

「バッファ領域長の仮定値計算式」で求める値は、DAM サービス定義に指定した各オペランドを基に、最も大きいバッファ領域サイズを算出します。したがって、「バッファ領域長の仮定値計算式」で算出した値ではバッファを確保できない場合があります。その場合、DAM サービス内部でバッファ領域が確保できるまで、確保する値を 1/2 にしながら確保処理を実行します。

dam_cache_size オペランドおよび dam_cache_size_fix オペランドの指定値が指定範囲内でない場合には、ともに KFCA01644-I メッセージが出力されます。

容量不足のため dam_cache_size オペランドおよび dam_cache_size_fix オペランドの指定値でバッファ領域を確保できない場合には、ともに KFCA01648-E メッセージが出力され DAM サービスや UAP がダウンします。

なお、デフォード更新処理指定の DAM ファイルを指定している場合は、仮定値計算式で算出したバッファ領域長の値に加えて次の見積もり式に示す容量が必要です。

見積もり式

実行間隔時間内※に実行され遅延書き込み処理に滞留するトランザクション数×おのこのトランザクションで更新される DAM ファイルのブロック長の総和

注※

DAM サービス定義の dam_io_interval オペランドに指定した実行間隔時間

次に、dam_cache_size オペランドと dam_cache_size_fix オペランドの違いについて説明します。

dam_cache_size を指定した場合

dam_cache_size を指定した場合、dam_cache_size の指定値、または前述の「バッファ領域長の仮定値計算式」の値のどちらか大きい方の値が採用されます。dam_cache_size オペランドの指定を省略した場合、および dam_cache_size オペランドの指定値が指定範囲内でない場合は、KFCA02530-I メッ

セージが出力され仮定値計算式の値が採用されます。dam_cache_size_fix オペランドを指定した場合は、優先的に dam_cache_size_fix オペランドの指定値が採用されます。

dam_cache_size_fix を指定した場合

優先的に dam_cache_size_fix オペランドの指定値を採用します。dam_cache_size_fix オペランドの指定を省略した場合、および dam_cache_size_fix オペランドの指定値が指定範囲内でない場合は、dam_cache_size オペランドが指定されていればそれを採用し、指定されていなければ KFCA02530-I メッセージを出力して仮定値計算式の値を採用します。

dam_cache_size_fix を指定することの利点

dam_cache_size オペランドに指定する値や、「バッファ領域長の仮定値計算式」で算出される値は、次の値を基に算出します。

- DAM サービス定義に指定した DAM ファイルの最大ブロック長(damfile 定義コマンドで指定したファイルの最大ブロック長)
- 同時実行トランザクション数(dam_tran_process_count オペランド指定値)
- トランザクション内で更新する最大ブロック数(dam_update_block オペランド指定値)

計算式では、上記の各値を掛け合わせることで求めています。つまり、最大ブロック長を持つ DAM ファイルが一つのトランザクションで 1 ブロックだけ更新される場合でも、上記計算式を用いて算出するため、十分に動作できるバッファサイズよりもさらに大きい値でバッファを確保することになります。このような場合、最適なバッファ領域長を dam_cache_size_fix オペランドに指定することで、最適なバッファを確保できます。ただし、dam_cache_size_fix オペランドを使用する場合は、その指定値に十分注意してください。dam_cache_size_fix オペランドに指定した値が、実際に必要となるバッファ長より小さかった場合、オンライン処理中にバッファ領域不足が発生し、処理が続行できなくなります。また、使用する DAM ファイルのうち、ディファード更新指定ファイルがある場合、実行間隔時間(dam_io_interval オペランドの指定値)内に滞留する量(滞留するブロック数とそのブロック長の総和)も考慮して、バッファ領域長を見積もる必要があります。

dam_cache_size オペランドおよび dam_cache_size_fix オペランドの相互指定関係での採用値について次の表に示します。

dam_cache_size_fix の指定	dam_cache_size の指定		
	指定が正しい場合	指定を省略した場合	指定を誤った場合 (KFCA01644-I 出力)
指定が正しい場合	(2)	(2)	(2)
指定を省略した場合	(1)	(3)	(3)
指定を誤った場合 (KFCA01644-I 出力)	(1)	(3)	(3)

(凡例)

- (1) : dam_cache_size への指定値を採用
- (2) : dam_cache_size_fix への指定値を採用

(3)：仮定値計算式の値を採用，かつ，KFCA02530-I メッセージ出力

バッファ領域不足が発生し処理が続行できなくなった場合，アボートコードが出力されるので，アボートコードに対する対策方法に従ってください。アボートコードについては，マニュアル「OpenTP1 メッセージ」を参照してください。

なお，DAM ファイルの構成を変更するなど，システムの定義および環境を変更する場合にも，再度指定値を見直してください。

リソースマネージャ用共用メモリの所要量は，dam_cache_size_fix オペランドで指定したキャッシュサイズになります。

●dam_cache_attribute=free|fixed

～《free》

共用メモリに確保したバッファ領域をメモリ上に固定するかどうかを指定します。このオペランドを指定する場合は，実装するメモリの容量や，全使用量に対するバッファ領域の割合などを考慮してください。この指定値は，デフォルト更新機能に関係なく指定してください。

free

バッファ領域をメモリ上に固定しません。実装する実メモリの容量が少ない場合，共用メモリのページングが発生して処理速度が低下することがあります。

fixed

バッファ領域をメモリ上に固定します。共用メモリのページングによって処理速度が低下することはありません。ただし，実装する実メモリの容量が少ない場合，共用メモリ以外のテキストやデータセグメントなどのページングが発生することがあります。

fixed は，OS が HP-UX または Solaris の場合だけ指定できます。なお，Solaris の場合，ご使用の環境によっては fixed を指定できないことがあります。詳細については，「リリースノート」を参照してください。

●dam_io_interval=実行間隔時間

～〈符号なし整数〉((1～60))《1》(単位：秒)

デフォルト更新機能を使用するとき，出力専用プロセスを実行する間隔を指定します。この指定値は，デフォルト更新機能を使用しないときは，指定する必要はありません。

出力専用プロセスの実行間隔を大きくし過ぎるとバッファ領域不足が発生し，ほかのトランザクションで更新または出力しようとしたブロックの書き込みができなくなって，メモリ不足で UAP がダウンすることがあります。この値は動作するトランザクションがピークに達した場合を想定したときの実行間隔時間内に終了するトランザクションの数と各トランザクションで更新，および出力するブロック数（バッファ所要量）で決めてください。

●dam_transaction_access=global|branch

～《branch》

DAM ファイルのデータ管理および排他管理をするトランザクションの単位を指定します。

global

グローバルトランザクション単位で、ファイルデータ管理および排他管理をします。global を指定すると、論理ファイルのオープン時にファイル排他を指定できません。

branch

トランザクションブランチ単位で、ファイルデータ管理および排他管理をします。

トランザクションブランチが異なれば、同一グローバルトランザクションでもアクセスはエラーとなります。

●dam_io_error_occur=stop|continue

～《continue》

ディスク更新時にディスク障害（入出力エラー）が発生した場合の DAM サービスの処置を指定します。

DAM サービスでは、トランザクションの同期点処理時にディスク更新をします。または、トランザクションの途中更新機能を使用している場合の DAM サービスアクセス関数の延長で、ディスク更新をします。ここでは、ディスク更新時に障害が発生した場合の **DAM サービスの処置**を指定します。

ディファード更新時および回復対象外ファイル更新時の、ディスク障害が発生した場合の処置については、指定値による違いはありません。ディファード更新時にディスク障害が発生した場合は、KFCA01646-E、KFCA01642-I のメッセージを出力して障害ファイル以外のファイルについて処理を続行します。回復対象外ファイル更新時にディスク障害が発生した場合は、KFCA01646-E、KFCA01642-I のメッセージを出力して dc_dam_rewrite 関数、dc_dam_write 関数が DCDAMER_IOER をリターンします。

stop

DAM アクセスを行っている UAP を異常終了します。このとき、クリティカル状態であるため、システムを停止します。KFCA01622-E または KFCA01646-E のメッセージで示される障害が起きたファイルに対して、障害要因を取り除くかまたは新しい別のボリュームを割り当て、damfrc コマンドを実行してファイルを回復させてください。

ファイルの回復手順を次に示します。

1. damfrc コマンドの実行

OpenTP1 の再開始を行う前に、必ずバックアップしてあったデータを damrstr コマンドでリストアし、damfrc コマンドを実行してください。システムダウン時点までのジャーナル情報を基にファイルを回復します。

2. OpenTP1 の再開始

OpenTP1 を再開始します。

3. ファイルの閉塞解除

OpenTP1 を再開始しても、障害が起きたファイルは障害閉塞状態になっています。そのため、damrles コマンドを実行して閉塞を解除してください。

4. トランザクションの決着

damrles コマンド実行後、ファイルへのアクセスが可能となり、OpenTP1 ではトランザクションの回復(決着)処理が行われ、そこでトランザクションが完了し、データの整合性が取られます。

なお、damrles コマンドが実行されるまでは、KFCA01623-E（ファイル切り離し中）または KFCA01624-E（ファイル閉塞中）のメッセージが出力され続け、トランザクションの回復ができない理由をユーザに知らせます。

continue

ディスク障害の発生が DAM アクセス中の場合、KFCA01618-E、KFCA01622-E、および KFCA01642-I のメッセージを出力後、UAP を異常終了させます。

また、トランザクションの同期点処理中の場合は、KFCA01622-E および KFCA01642-I のメッセージを出力したあと、トランザクションを完了させます（コミット決着：OpenTP1 ログ上は正常に終了し、関数のリターン値も DC_OK に見えます）。この場合、トランザクションは正常に完了していますが、障害の起きたファイルはトランザクションで更新したデータが反映されていません。

このため、業務処理を続行するとデータの不整合となりますので、必ず、障害の起きたファイルをいったんオンラインから切り離し、damrstr コマンド、および damfrc コマンドによる論理的なファイル回復を行い、回復したファイルを damadd コマンドで追加後、damrles コマンドで閉塞を解除してください。ファイルの切り離し、回復、追加をする場合には、UAP を停止させてください。

●dam_cache_reuse_from=last | first

～《last》

DAM ファイルをアクセスするトランザクションで、新たにキャッシュブロック用領域が必要となった場合、どのブロックから優先的に再利用対象として検索するかを指定します。なお、DAM サービスではブロックを新しく更新し、参照用キャッシュチェーンにつなぐ場合、参照用キャッシュチェーンの先頭につないでいます。

last

再利用するキャッシュブロックを、DAM サービス内部で管理している参照用キャッシュチェーンの後ろにあるキャッシュブロックから優先して検索します。last を指定すると、最も古く参照用キャッシュチェーンにつながれたキャッシュブロックが再利用されます。

first

再利用するキャッシュブロックを、DAM サービス内部で管理している参照用キャッシュチェーンの先頭にあるキャッシュブロックから優先して検索します。first を指定すると、最も新しくキャッシュチェーンにつながれたキャッシュブロックが再利用されます。

last を指定した場合でも、キャッシュブロックの検索のタイミングでトランザクションによってアクセスされているキャッシュブロックは、再利用対象外として扱われます。したがって、その次に古いキャッシュブロックが再利用対象として検索されます。また、最も古くからキャッシュチェーンに残っているかどうかを判断していて、LRU 方式のように最近アクセスされたもののチェーンの並べ替えはしていません。あくまで、アクセスされ、キャッシュチェーンにつながれた時期が古いものから検索されます。

●dam_default_cache_num=キャッシュブロック再利用境界デフォルト値

～〈符号なし整数〉((0～4000000))《0》

キャッシュブロック再利用処理の境界値のデフォルト値を指定します。このオペランドに指定した値が、damcache 定義コマンドでキャッシュブロック再利用境界値を指定しなかった場合の論理ファイルの境界値となります。

●dam_ex_refer_read=none | stay

～《none》

DAM ファイルをアクセスするトランザクションで、**排他あり参照指定で読み込むブロックを、トランザクション決着までキャッシュバッファ領域上に残すかどうか**を指定します。

このオペランドを使用する場合、次のことに注意してください。

- DAM サービス定義の dam_update_block オペランド（一つのトランザクションで更新できるブロック数）には、排他あり参照 read を実行するブロック数を加算してください。
- トランザクションの途中更新機能を使用している場合、この機能を使用した排他あり参照目的の入力後に、更新目的の入力をし、ブロックの更新（dc_dam_rewrite 関数）を実行すると、トランザクションの途中で更新ブロックのデータ実更新が発生することがあります。更新ブロックのデータ実更新が発生した場合は、トランザクション決着時までキャッシュバッファ領域上に必ず残すとは保証されません。更新しないキャッシュブロックについてはトランザクション決着時まで残ります。
- 排他あり参照目的の入力は、トランザクション決着まで排他を解放することはできません。

none

排他あり参照指定で入力したキャッシュブロックは、ユーザプログラムにデータを返却したあと、再利用可能なブロックとして扱います。ただし、キャッシュバッファ領域に余裕がある場合、すぐに再利用されることはありません。キャッシュバッファ領域の空き領域が少なくなった場合に再利用される可能性があります。

次のような場合は、none を指定してください。

- アクセスする DAM ブロックに特定性がない場合

stay

排他あり参照指定で入力したキャッシュブロックでも、更新指定のときと同じようにトランザクション決着時までキャッシュバッファ領域上に残します。

次のような場合に stay を指定し、排他あり参照 read を実行すると、キャッシュヒット率が向上しやすくなります。

- 特定の DAM ブロックに対して、頻繁に排他あり参照 read を実行する場合

なお、stay を指定した場合、次のことに注意してください。

- DAM サービス定義に指定した dam_update_block オペランドに、排他あり参照目的の入力を実行するブロック数を加算する必要があります。また、dam_update_block オペランドの値を加算して、OpenTP1 の動的共用メモリサイズを大きくする必要があります。
- dam_update_block オペランドの値を加算しなかった場合、次のような影響があります。

1. トランザクションの途中更新機能を使用している場合

一つのトランザクション内で、排他あり参照目的の入力、更新目的の入力、ブロックの更新 (dc_dam_write 関数) を実行したブロック数の合計が、dam_update_block オペランドの指定値を超えると、途中更新処理が実行され、データ実更新が発生します。したがって、排他あり参照目的の入力ブロック数が多い場合、途中更新処理を実行する回数が多くなり、その結果、データの実更新が頻繁に発生します。

2. DAM サービス定義の dam_update_block_over オペランドに error を指定している場合

一つのトランザクション内で、排他あり参照目的の入力、更新目的の入力、ブロックの更新 (dc_dam_write()) を実行したブロック数の合計が、dam_update_block オペランドの指定値を超えると、そのときに実行した API が DCDAMER_JNLOV でエラーリターンします。したがって、排他あり参照目的の入力ブロック数が多い場合、DCDAMER_JNLOV でエラーリターンする回数が多くなります。

●dam_max_block_size=最大ブロック長

～ 〈符号なし整数〉 ((504～32760))※ 《504》 (単位：バイト)

注※

セクタ長×n-8 を満たす値を指定してください (n は正の整数)。

システム内の DAM ファイルの最大ブロック長を指定します。ただし、条件によって、指定値と異なる値が最大ブロック長に採用されることがあります。条件と採用される最大ブロック長の関係を次の表に示します。

dam_max_block_size オペランド	damfile 定義コマンド	
	指定あり	指定なし
指定が正しい場合	a または b のうち、大きい値を採用	a の値を採用
指定を省略した場合	b の値を採用	504 を仮定
指定を誤った場合 (504～32760 の範囲外)	b の値を採用※1	504 を仮定※1
指定を誤った場合 (セクタ長×n-8 以外 (n は正の整数))	b の値を採用※2	504 を仮定※2

(凡例)

- a：DAM サービス定義の dam_max_block_size オペランドの値
- b：DAM サービス定義に定義されている論理ファイルの中で最大ブロック長を持つファイルのブロック長

注※1

KFCA00216-E および KFCA01644-I を出力します。

注※2

KFCA02565-I を出力します。

●**dam_kb_size=1000|1024**

～《1000》

DAM サービス用共用メモリとして確保されるバッファ領域のサイズを DAM サービス定義 (\$DCDIR/conf/dam) の dam_cache_size オペランド (単位: キロバイト) に指定した場合, 自動的にバイト換算されます。そのときに, 1 キロバイト当たり 1000 バイトで換算するか, 1024 バイトで換算するかを指定します。

1000

1 キロバイト当たり 1000 バイトで換算します。

1024

1 キロバイト当たり 1024 バイトで換算します。

●**watch_time=最大応答待ち時間**

～〈符号なし整数〉((0～65535)) (単位: 秒)

RPC によってプロセス間で通信する場合, サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で, このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって, 大きな値を指定した場合, OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は, RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は, 応答を受信するまで待ち続けます。0 を指定した場合, OpenTP1 が終了しない場合があります。

省略した場合は, システム共通定義の watch_time オペランドの値を仮定します。

このオペランドは, システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は, このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも, 極端に小さな値または大きな値を指定すると, OpenTP1 ダウンに至る障害が発生する場合がありますので, ご注意ください。

コマンド形式

次ページ以降に記述しています。

名称

キャッシュブロックの再利用境界値の指定

形式

[damcache 論理ファイル名 キャッシュブロック再利用境界値]

機能

DAM ファイルのための共用メモリが不足したとき、DAM ファイルのキャッシュブロックのうち、未使用状態のキャッシュブロックを再利用するときの、再利用境界値を指定します。

コマンド引数

●論理ファイル名

～ 〈1～8 文字の識別子〉

キャッシュブロック再利用処理の境界値を有効とする論理ファイル名を指定します。ここで指定する論理ファイル名は、あらかじめ damfile 定義コマンドで定義しておく必要があります。

●キャッシュブロック再利用境界値

～ 〈符号なし整数〉 ((0～4000000))

キャッシュブロック再利用処理の境界値を指定します。キャッシュブロック再利用境界値の指定を省略した場合は、DAM サービス定義の dam_default_cache_num オペランドに指定された値をデフォルト値とします。

ここで指定するキャッシュブロック再利用境界値に達していない間は、アクセスする DAM ファイルではなく、ほかの DAM ファイルのキャッシュブロックを優先して再利用します。キャッシュブロック数が再利用境界値を超えた場合は、アクセスする DAM ファイルのキャッシュブロックから再利用します。0 を指定した場合は、従来の処理と同じように、アクセスする DAM ファイルのキャッシュブロックチェーンにつながっているキャッシュブロックから再利用します。

注意事項

- damcache 定義コマンドは、damfile 定義コマンドで定義した論理ファイルに対して有効です。そのため、damcache 定義コマンドの指定前に、あらかじめ damfile 定義コマンドで対象となる論理ファイル名を定義する必要があります。damfile 定義コマンドで対象となる論理ファイルを指定する前にキャッシュブロック再利用境界値を指定しても、値は有効となりません。
- damcache 定義コマンドでキャッシュブロック再利用境界値を指定しなかった論理ファイルのキャッシュブロック再利用境界値は、dam_default_cache_num オペランドで指定した値が採用されます。

- 一つの論理ファイル名に対して、damcache 定義コマンドを複数回指定した場合は、動作は保証されません。

名称

キャッシュブロック数のしきい値の指定

形式

[damchfmt 論理ファイル名 キャッシュブロック数のしきい値]

機能

一つの DAM ファイルで管理するキャッシュブロック数の上限を指定します。

コマンド引数

●論理ファイル名

～ 〈1～8 文字の識別子〉

キャッシュブロック数のしきい値を有効とする論理ファイル名を指定します。ここで指定する論理ファイル名は、あらかじめ damfile 定義コマンドで定義しておく必要があります。

●キャッシュブロック数のしきい値

～ 〈符号なし整数〉 ((0～4000000))

指定した論理ファイルが管理する、キャッシュブロック数のしきい値を指定します。この指定を省略した場合は、共用メモリ資源がなくなるまでキャッシュブロックを確保します。

注意事項

- このコマンドは、damfile 定義コマンドで定義した論理ファイルに対して有効です。そのため、このコマンドの指定前に、あらかじめ damfile 定義コマンドで対象となる論理ファイル名を定義しておく必要があります。damfile 定義コマンドで対象となる論理ファイルを指定する前に、このコマンドでキャッシュブロック数のしきい値を指定しても、その値は有効になりません。
- 一つの論理ファイル名に対して、このコマンドを複数回指定した場合、動作は保証されません。

名称

論理ファイルの指定

形式

```
{{damfile [-{d|n [-f]}] [-c] 論理ファイル名 物理ファイル名}}
```

機能

オンライン中に DAM サービスを使用してアクセスするファイルを論理ファイル名と物理ファイル名の対で指定します。

DAM サービスは、指定された論理ファイル名によってアクセスを制御します。

オプション

●-d

指定されたファイルをデフォード更新処理します。

省略した場合は、デフォード更新処理しません（同期点で DAM ファイルを実更新します）。

デフォード更新指定の DAM ファイルと通常更新の DAM ファイルを、一つのトランザクションから更新出力した場合には、どちらの DAM ファイルも同期点で実更新します。

●-n

指定されたファイルを回復対象外ファイルとして処理する場合に指定します。

●-f

指定されたファイルをキャッシュレスアクセスとして処理する場合に指定します。このオプションは、必ず -n オプションと同時に指定してください。

●-c

指定された DAM ファイルに障害が発生しても、DAM サービスの正常開始処理を続行する場合に指定します。ただし、正常開始処理を続行できない場合は、DAM サービスの正常開始処理を終了します。DAM サービス開始後、この DAM ファイルは障害閉塞状態になります。dc_dam_open 関数でアクセスした場合、DCDAMER_OHOLD でエラーリターンします。障害が発生した DAM ファイルを UAP で使用する場合は、いったん、damrm コマンドでオンラインから切り離れたあと、障害要因を取り除き、再度、damadd コマンドでオンラインに登録してください。

コマンド引数

●論理ファイル名

～ 〈1～8 文字の識別子〉

論理ファイル名を指定します。

●物理ファイル名

～ 〈1～63 文字のパス名〉

論理ファイル名に対応する物理ファイル名を絶対パス名で指定します。ただし、環境変数は使用できません。

注意事項

- DAM サービスでは、次の二つの値を比較して、大きい方の値を DAM ファイルの最大ブロック長とします。
 - damfile 定義コマンドで定義されている論理ファイルの中で、最大のブロック長を持つファイルのブロック長の値
 - dam_max_block_size オペランドに指定した値

damfile 定義コマンドで論理ファイルが定義されていないかつ、dam_max_block_size オペランドが指定されていない場合は、DAM ファイルの最大ブロック長は、504 バイトになります。
- DAM サービスでは、OpenTP1 システムでの DAM ファイルの最大ブロック長を、DAM サービス定義に damfile 定義コマンドで定義されている論理ファイルの中で、最大のブロック長を持つファイルのブロック長としています。DAM サービス定義に論理ファイルがまったく定義されていない場合は、最大ブロック長は 504 バイトになります。
- -f オプションを指定した場合、DAM サービス専用共用メモリの DAM ファイルデータ領域を使用しないで、必ずファイルの I/O を実行します。したがって、システムがオンラインの間にアクセスする DAM ファイルのブロック数によっては、性能が低下することがあります。また、オンラインで使用する DAM ファイルがすべてキャッシュレスアクセス指定の場合、dam_cache_size_fix オペランドで 10 を指定すると、最小限の DAM キャッシュメモリでシステムを開始できます。この場合の注意事項については、[dam_cache_size_fix オペランド](#)を参照してください。
- damfile 定義コマンドの指定数と dam_added_file オペランドに指定した値の合計が 3600 を超える場合、DAM サービスが起動できないことがあります。

TAM サービス定義

形式

set 形式

```
set tam_max_tblnum=オンライン中に使用するTAMテーブルの最大数
set tam_max_filesize=オンライン中に使用するTAMテーブルの最大容量
set tam_max_recsizе=TAMテーブルの最大レコード長
[set tam_jnl_err_flag=STOP|CONTINUE]
[set tam_pool_attri=fixed|free]
[set tam_tbl_lock_mode=LOCK|NOLOCK]
[set tam_cbl_level=COBOL APIの排他レベル]
[set tam_max_trnnum=同時走行最大トランザクションブランチ数]
[set tam_max_trnfilnum=トランザクション内最大アクセステーブル数]
[set watch_time=最大応答待ち時間]
```

コマンド形式

```
{{tamtable [-o ローディング契機] [-a アクセス形態] [-i] [-j]
           TAMテーブル名 物理ファイル名}}
```

機能

TAM テーブルを管理するための実行環境を定義します。

説明

set 形式のオペランド

●tam_max_tblnum=オンライン中に使用する TAM テーブルの最大数

～ 〈符号なし整数〉 ((1～65535))

オンライン中に使用する TAM テーブルの最大数を指定します。オンライン中は、ここで指定した数までテーブルを追加できます。

tamtable 定義コマンドで指定した TAM テーブル数に、tamadd コマンドでオンライン中に動的に追加する TAM テーブル数を加算したものを、ここで指定します。

●tam_max_filesize=オンライン中に使用する TAM テーブルの最大容量

～ 〈符号なし整数〉 ((136～1000000000)) (単位：バイト)

オンライン中に使用する TAM テーブルの中で、最大の TAM テーブルサイズ※を指定します。

tamtable 定義コマンドで指定した TAM テーブルだけでなく、tamadd コマンドでオンライン中に動的に追加する TAM テーブルのサイズも含めて、それらの中で最大の値を指定してください。

なお、初期データがある状態で tamcre コマンドに -u オプションを指定し、かつ -x オプションを指定しないで実行して TAM ファイルを作成した場合、シノニム領域の最適化によって見積もり式から算出された TAM テーブルの容量よりも実際に作成される TAM テーブルの最大容量が小さくなります。ただし、この削減量は事前に見積もることができないため、tam_max_filesize オペランドには、見積もり式から算出された TAM テーブルの容量の最大値以上の値を指定して下さい。

TAM の共用メモリ量を削減したいなどで、tam_max_filesize オペランドの指定値を実際の値に近づけた場合は fills コマンドの結果から作成された TAM テーブルの容量をレコード長×レコード数で確認し、TAM テーブルの容量の最大値以上の値を指定して下さい。

注※

TAM テーブルのサイズについては、マニュアル「OpenTP1 運用と操作」にある、TAM ファイルのサイズの見積もり式の説明を参照してください。

●tam_max_recsz=TAM テーブルの最大レコード長

～ 〈符号なし整数〉 ((1～1000000000)) (単位：バイト)

オンライン中に使用する TAM テーブルの最大レコード長を指定します。

●tam_jnl_err_flag=STOP|CONTINUE

～ 《STOP》

TAM サービスの再開始時にジャーナルの読み込み障害が発生した場合、TAM サービスを停止するかどうかを指定します。

STOP

TAM サービスを停止します。

CONTINUE

TAM テーブルをすべて閉塞状態にして TAM サービスを開始します。

●tam_pool_attr=fixed|free

～ 《HP-UX または Solaris の場合：fixed AIX, Linux または Windows の場合：free》

共用メモリに TAM サービス専用のメモリプールをメモリ上に固定するかどうかを指定します。TAM サービスのメモリプールは、TAM テーブルのロードや、アクセスするテーブルの管理などに使用されます。

このオペランドを指定する場合には、実装するメモリの容量や、全使用容量に対する共用メモリプールの割合などを考慮してください。

fixed

TAM サービス用メモリプールを固定します。

共用メモリのページングによって処理速度が低下することはありません。ただし、実装するメモリの容量が少ない場合、共用メモリ以外のテキストやデータセグメントなどのページングが発生することがあります。

fixed は、OS が HP-UX または Solaris の場合だけ指定できます。なお、Solaris の場合、ご使用の環境によっては fixed を指定できないことがあります。詳細については、「リリースノート」を参照してください。

free

TAM サービス用メモリプールを固定しません。

実装するメモリの容量が少ない場合、共用メモリのページングが発生して処理速度が低下することがあります。

●tam_tbl_lock_mode=LOCK|NOLOCK

～《LOCK》

TAM テーブルアクセス形態が‘参照型’または‘追加・削除できない更新型’のテーブルのテーブル排他モードを指定します。

LOCK

アクセス時のテーブルを排他します。

NOLOCK

アクセス時のテーブルを排他しません。

この指定をした場合は、アクセス時の性能は向上します。また、UAP で排他する指定をしても、テーブル排他はしません。

●tam_cbl_level=COBOL API の排他レベル

～〈符号なし整数〉((0～2))《0》

COBOL API の排他が競合した場合の、排他レベルを指定します。

なお、この指定は COBOL API だけに有効です。

0

無条件に排他が解除されるのを待ちます。

COBOL API のデータ名 I の指定は無効になります。

1

排他が解除されるのを待たないで、エラーリターンします。

COBOL API のデータ名 I の指定は無効になります。

2

COBOL API のデータ名 I の排他解除の指定に従います。

●tam_max_trnnum=同時走行最大トランザクションブランチ数

～〈符号なし整数〉((1～8192))《20》

TAM テーブルをアクセスするトランザクションブランチの同時走行数を指定します。

この指定値が大きいと、TAM サーバの共用メモリも大きくなります。共用メモリが確保できない場合は TAM サーバを開始できない場合があります。また、指定値が小さいと、API 関数でメモリ不足が発生する場合があります。

●tam_max_trnfilnum=トランザクション内最大アクセステーブル数

～ 〈符号なし整数〉 ((1～1024)) 《5》

1 トランザクションブランチでアクセスする最大 TAM テーブル数を指定します。

この指定値が大きいと、TAM サーバの共用メモリも大きくなります。共用メモリが確保できない場合は TAM サーバを開始できない場合があります。また、指定値が小さいと、API 関数でメモリ不足が発生する場合があります。

●watch_time=最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しない場合があります。

このオペランドは、システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

省略した場合の指定値は、次のようになります。

- ・ システム共通定義の watch_time オペランドの指定も省略した場合は、0 を仮定します。
- ・ システム共通定義の watch_time オペランドを指定した場合は、その指定値を仮定します。

システム共通定義、TAM サービス定義の両方に指定した場合の指定値の優先順位は、次のとおりです (1. > 2.)。

1. TAM サービス定義
2. システム共通定義

コマンド形式

次ページ以降に記述しています。

tamtable

名称

TAM テーブルの属性の指定

形式

```
{{tamtable  [-o ローディング契機]  [-a アクセス形態]  [-i]  [-j]  
            TAMテーブル名  物理ファイル名}}
```

機能

オンライン中に TAM サービスを使用してアクセスするテーブル名と属性を指定します。

オプション

●-o ローディング契機

～ 《start》

TAM テーブルをロードする契機を指定します。

start

TAM サービスを開始するときにロードします。

cmd

tamload コマンドを入力したときにロードします。

lib

dc_tam_open 関数を発行し、TAM テーブルをオープンしたときにロードします。

●-a アクセス形態

～ 《read》

TAM テーブルのアクセス形態を指定します。

read

参照型

rewrite

追加・削除できない更新型

write

追加・削除できる更新型

reclck

テーブル排他を確保しない、追加・削除できる更新型

●-i

TAM テーブルの実更新時に、入出力エラーが発生してもアクセスを続行する場合に指定します。このオプションを指定すると、入出力エラーが発生しても同一オンラインでは、その TAM テーブルは障害閉塞状態にはなりません。したがって、アクセスできます。

ただし、入出力エラーが発生後にオンラインダウンした場合、そのテーブルをオンラインから切り離して TAM サービスを再開します。

●-j

TAM レコード更新時に、更新された部分だけのジャーナルを取得します。

このオプションを指定すると、TAM レコード更新時のジャーナルの取得量を低減できます。

●TAM テーブル名

～ 〈1～32 文字の識別子〉

TAM テーブル名を指定します。

●物理ファイル名

～ 〈1～63 文字のパス名〉

TAM テーブルを格納する物理ファイルを絶対パス名で指定します。ただし、環境変数は使用できません。

クライアントサービス定義

形式

set 形式

```
[set parallel_count=常駐プロセス数 [, 最大プロセス数] ]
[set balance_count=1プロセスが処理するサービス要求数]
[set trn_expiration_time=トランザクションブランチ限界経過時間]
[set trn_expiration_time_suspend=Y|N|F]
[set trn_cpu_time=トランザクションブランチCPU監視時間]
[set open_rm=OpenTP1_ALL|OpenTP1_NONE]
[set clt_inquire_time=常設コネクション問い合わせ間隔最大時間]
[set clt_port=クライアント拡張サービスのポート番号]
[set clt_trn_conf=Y|N]
[set clt_cup_conf=Y|N]
[set cup_parallel_count=常駐プロセス数 [, 最大プロセス数] ]
[set cup_balance_count=サービス要求滞留値]
[set clttrn_port=トランザクショナルRPC実行プロセスのポート番号]
[set cltcon_port=CUP実行プロセスのポート番号]
[set trn_statistics_item=統計情報項目 [, 統計情報項目] ...]
[set trn_optimum_item=トランザクション最適化項目
                        [, トランザクション最適化項目] ...]
[set trn_watch_time=トランザクション同期点処理時の最大通信待ち時間]
[set trn_rollback_information_put=no|self|remote|all]
[set trn_limit_time=トランザクションブランチ最大実行可能時間]
[set trn_rollback_response_receive=Y|N]
[set trn_partial_recovery_type=type1|type2|type3]
[set trn_completion_limit_time=トランザクション完了限界時間]
[set message_store_buflen=メッセージ格納バッファプール長]
[set watch_time=最大応答待ち時間]
```

コマンド形式

なし。

機能

OpenTP1 のクライアント機能を支援するためのサーバの実行環境を定義します。

CUP からトランザクションを起動するときに次のオペランドを指定します。

説明

set 形式のオペランド

●parallel_count=常駐プロセス数 [, 最大プロセス数]

～ 〈符号なし整数〉 ((1～1024)) 《1》

CUP（サーバ側で同時に処理される）から起動するトランザクションの数を指定します。常駐プロセス数を指定すると、OpenTP1 起動時に指定した数の実行プロセスが一緒に起動され、常に CUP からのトラ

ンザクションの開始要求を受け付ける状態になります。複数の CUP からトランザクションの開始要求を受け付けたときは、並行して処理するので処理性能が向上します。

最大プロセス数を指定すると、常駐プロセス数を超える処理は非常駐プロセスを動的に起動して処理します。指定した最大プロセス数以内でサーバプロセスの起動を制御するため、OpenTP1 システムの性能が低下するのを防ぎます。

プロセス数の指定条件を次に示します。

1. 常駐プロセス数と最大プロセス数は 1 以上の値を指定します。
2. 最大プロセス数を指定する場合は、常駐プロセス数以上の値を指定します。
3. 最大プロセス数を省略した場合は、すべてのプロセスが常駐プロセスとなります。
4. すべてのプロセスを非常駐プロセス（必要に応じて起動）とすることはできません。

●balance_count=1 プロセスが処理するサービス要求数

～ 〈符号なし整数〉 ((0～512)) 《3》

このユーザサーバに対応するスケジュールキューに、処理されないで残っているサービス要求を、1 プロセスで幾つ処理させるかを指定します。サービス要求が、(このオペランドの指定値) × (起動プロセス数) を超えてスケジュールキューに残っている場合には、非常駐プロセスを起動してサービス要求を処理させます。このオペランドは、parallel_count オペランドで非常駐プロセスを起動するように指定したサービスグループにだけ有効です。

0 を指定すると、サービス要求のあった時点で起動されているすべてのプロセスがサービス処理中の場合、非常駐プロセスが起動されます。

●trn_expiration_time=トランザクションブランチ限界経過時間

～ 〈符号なし整数〉 ((0～65535)) 《0》 (単位：秒)

トランザクションブランチの処理を監視する限界経過時間を指定します。CUP からトランザクションを起動したあとに、指定時間を経過してもトランザクションのコミットまたはロールバックの指示がない場合は、このトランザクションブランチのプロセスが異常終了して、ロールバックします。

なお、このオペランドは、クライアント環境定義の DCCLTTREXPTM オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです (1.> 2.)。

1. クライアント環境定義
2. クライアントサービス定義

なお、RPC 機能を使用した場合に、他プロセスで実行するトランザクションブランチの処理時間も監視時間を含むかどうかは、trn_expiration_time_suspend オペランドで指定してください。

●trn_expiration_time_suspend=Y|N|F

～《N》

トランザクションブランチの処理を監視するとき、次の処理時間も監視時間を含むかどうかを指定します。

1. 監視対象のトランザクションブランチが、RPC 機能を使ってほかのトランザクションブランチを呼び出し、その処理が終わるのを待つ時間
2. 連鎖 RPC で呼び出されたサーバ UAP が次のサービス要求を待つ時間
3. 監視対象のトランザクションブランチが、非同期 RPC を使用してほかのトランザクションブランチを呼び出したあと、処理結果受信処理を行っている時間

Y

1., 2., 3.のすべてを監視時間を含みます。

N

3.だけを監視時間を含みます。

F

1., 2., 3.のどれも監視時間を含みません。

なお、このオペランドは、クライアント環境定義の DCCLTTREXPSP オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです (1.> 2.)。

1. クライアント環境定義
2. クライアントサービス定義

このオペランドと各種タイマ監視との関係については、「付録 A.2 トランザクションの時間監視」を参照してください。

●trn_cpu_time=トランザクションブランチ CPU 監視時間

～〈符号なし整数〉((0～65535))《0》(単位：秒)

トランザクションブランチが同期点処理までに使用できる CPU 時間を指定します。0 を指定した場合は、時間監視しません。指定時間を超えた場合は、該当するトランザクションブランチのプロセスを異常終了し、ロールバックします。

なお、このオペランドは、クライアント環境定義の DCCLTTTRCPUTM オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです (1.> 2.)。

1. クライアント環境定義

2. クライアントサービス定義

●open_rm=OpenTP1_ALL|OpenTP1_NONE

～《OpenTP1_NONE》

CUP からのトランザクショナル RPC 機能を使用した場合に、クライアントサービス実行プログラムがオープンするリソースマネージャの名称を指定します。これによって、CUP からのトランザクショナル RPC 機能で同期点処理が最適化できるようになり、トランザクションの性能が向上します。

このオペランドを指定すると、parallel_count オペランドに指定したプロセス数の分だけ、リソースマネージャの資源を占有します。

指定できるリソースマネージャ名を次に示します。

OpenTP1_ALL

OpenTP1 開始時、クライアントサービス実行形式プログラムで、OpenTP1 が提供するリソースマネージャのうち、OpenTP1 に登録されているすべてのリソースマネージャに対してオープン処理を実行します。

OpenTP1_NONE

OpenTP1 開始時、クライアントサービス実行形式プログラムで、リソースマネージャに対してオープン処理を実行しません（同期点処理が最適化できません）。

●clt_inquire_time=常設コネクション問い合わせ間隔最大時間

～〈符号なし整数〉((0~1048575))《180》(単位：秒)

CUP がサーバに対して問い合わせをしてから、次の問い合わせをするまでの間隔の最大時間を指定します。

常設コネクション問い合わせ間隔最大時間は、CUP 実行プロセスで監視するタイマであり、指定時間を超えても問い合わせがない場合、CUP 実行プロセス側で強制的に常設コネクションを解放します。

また、トランザクション内で常設コネクション問い合わせ間隔最大時間の満了を検出した場合は、該当するトランザクションをロールバックします。

0 を指定すると、CUP からの問い合わせを無限に待ちます。

なお、このオペランドは、クライアント環境定義の DCCLTINQUIRETIME オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです (1.> 2.)。

1. クライアント環境定義
2. クライアントサービス定義

●clt_port=クライアント拡張サービスのポート番号

～ 〈符号なし整数〉 ((5001～65535))

クライアント拡張サービスのポート番号を指定します。

他システムサーバで使用しているウェルノウンポート番号と異なるポート番号を指定してください。

このオペランドを省略した場合、システム共通定義の rpc_port_base オペランドが指定されていれば、rpc_port_base の指定値から「rpc_port_base の指定値+ prc_process_count の値」の範囲のポート番号が任意に割り当てられます。rpc_port_base オペランドの指定がない場合には、システムが任意に割り当てたポート番号を使用します。

このオペランドで指定するポート番号は、ほかのプログラムと重複しないようにしてください。

また、OS には任意に割り当てる番号がありますが、この番号も使用しないでください。OS が任意に割り当てる番号は、OS の種別やバージョンによって異なります。詳細については、使用している OS のマニュアルを参照してください。

●clt_trn_conf=Y|N

～ 〈Y〉

自 OpenTP1 ノードで、トランザクショナル RPC 実行プロセスを起動するかどうかを指定します。このオペランドを省略、または Y を指定した場合、parallel_count で指定されたプロセス数分のトランザクショナル RPC 実行プロセスが起動されます。

●clt_cup_conf=Y|N

～ 〈N〉

自 OpenTP1 ノードで、CUP 実行プロセスを起動するかどうかを指定します。このオペランドで Y を指定した場合、cup_parallel_count で指定されたプロセス数分の CUP 実行プロセスが起動されます。

CUP から dc_clt_connect 関数、または dc_clt_connect_s 関数などで常設コネクションを確立したい場合は、Y を指定してください。

●cup_parallel_count=常駐プロセス数 [, 最大プロセス数]

～ 〈符号なし整数〉 ((1～1024)) 〈1〉

CUP（サーバ側で同時に処理される）から確立される常設コネクションの数を指定します。

常駐プロセス数を指定すると、OpenTP1 起動時に指定した数の CUP 実行プロセスと一緒に起動され、常に CUP からの常設コネクション確立要求を受け付ける状態になります。複数の CUP から常設コネクション確立要求を受け付けた時は、並行して処理するので、処理性能が向上します。

最大プロセス数を指定すると、常駐プロセス数を超える処理は非常駐プロセスを動的に起動して処理します。指定した最大プロセス数以内でサーバプロセスの起動を制御するため、OpenTP1 システムの性能が低下するのを防ぎます。

プロセス数の指定条件を次に示します。

1. 常駐プロセス数と最大プロセス数は 1 以上の値を指定します。
2. 最大プロセス数を指定する場合、常駐プロセス数以上の値を指定します。
3. 最大プロセス数を省略した場合、すべてのプロセスが常駐プロセスとなります。
4. すべてのプロセスを非常駐プロセス（必要に応じて起動）とすることはできません。

一方、クライアントサービス定義の `parallel_count` オペランドは、トランザクショナル RPC 実行プロセスの常駐プロセス数および最大プロセス数を指定します。

●`cup_balance_count`=サービス要求滞留値

～〈符号なし整数〉((0～512))《3》

処理されないで残っている常設コネクション確立要求数の滞留値を指定します。

CUP から常設コネクション確立要求を受け付け、その要求をスケジュールキューに登録後、コネクション確立要求がこのオペランドの指定値を超えると、スケジュール機能が非常駐プロセスを起動して、常設コネクション確立要求を処理します。このオペランドは、`cup_parallel_count` オペランドで最大プロセス数を指定した場合だけ有効です。

0 を指定すると、常設コネクション確立要求のあった時点で起動されているすべてのプロセスが CUP との常設コネクションを確立していた場合、非常駐プロセスが起動されます。

一方、クライアントサービス定義の `balance_count` オペランドは、トランザクショナル RPC 実行プロセスのサービス要求滞留値を指定します。

●`clttrn_port`=トランザクショナル RPC 実行プロセスのポート番号

～〈符号なし整数〉((5001～65535))

トランザクショナル RPC 実行プロセスのポート番号を指定します。

他システムサーバで使用しているウェルノウンポート番号と異なるポート番号を指定してください。

このオペランドを省略した場合、システム共通定義の `rpc_port_base` オペランドが指定されていれば、`rpc_port_base` の指定値から「`rpc_port_base` の指定値 + `prc_process_count` の値」の範囲のポート番号が任意に割り当てられます。

`rpc_port_base` オペランドの指定がない場合には、システムが任意に割り当てたポート番号を使用します。

ここで指定したポート番号から、parallel_count オペランドで指定した数がポート番号として割り当てられます。したがって、このオペランドの指定値と parallel_count オペランドの指定値の和が、65535 を超えないように注意してください。

このオペランドで指定するポート番号は、ほかのプログラムと重複しないようにしてください。

また、OS には任意に割り当てる番号がありますが、この番号も使用しないでください。OS が任意に割り当てる番号は、OS の種別やバージョンによって異なります。詳細については、使用している OS のマニュアルを参照してください。

●cltcon_port=CUP 実行プロセスのポート番号

～ 〈符号なし整数〉 ((5001～65535))

CUP 実行プロセスのポート番号を指定します。

他システムサーバで使用しているウェルノウンポート番号と異なるポート番号を指定してください。

このオペランドを省略した場合、システム共通定義の rpc_port_base オペランドが指定されていれば、rpc_port_base の指定値から「rpc_port_base の指定値 + prc_process_count の値」の範囲のポート番号が任意に割り当てられます。

rpc_port_base オペランドの指定がない場合には、システムが任意に割り当てたポート番号を使用します。

ここで指定したポート番号から、cup_parallel_count オペランドで指定した数がポート番号として割り当てられます。したがって、このオペランドの指定値と cup_parallel_count オペランドの指定値の和が、65535 を超えないように注意してください。

このオペランドで指定するポート番号は、ほかのプログラムと重複しないようにしてください。

また、OS には任意に割り当てる番号がありますが、この番号も使用しないでください。OS が任意に割り当てる番号は、OS の種別やバージョンによって異なります。詳細については、使用している OS のマニュアルを参照してください。

●trn_statistics_item=統計情報項目〔, 統計情報項目〕 …

～ 《executiontime》

トランザクションブランチの統計情報を取得する項目を指定します。

nothing

統計情報を取得しません。

base

基本情報として、次の情報を取得します。

- トランザクションブランチの識別子
- トランザクションブランチの決着結果

- トランザクションブランチの実行プロセス種別
- トランザクションブランチの実行サーバ名
- トランザクションブランチの実行サービス名

executiontime

基本情報とトランザクションブランチの実行時間情報を取得します。

cputime

基本情報とトランザクションブランチの CPU 時間情報を取得します。

nothing の指定は、一つしかできません。また、nothing とほかの統計情報項目を同時に指定した場合、nothing の指定は無効になります。

トランザクションに関する統計情報を取得するときは次のどれかを指定してください。

- トランザクションサービス定義の trn_tran_statistics オペランドに Y を指定
- trnstics コマンドで-s オプションを指定

なお、このオペランドは、クライアント環境定義の DCCLTTRSTATISITEM オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです (1.> 2.)。

1. クライアント環境定義
2. クライアントサービス定義

●trn_optimum_item=トランザクション最適化項目 [, トランザクション最適化項目] ...

～《base》

複数のユーザサーバで構成されるグローバルトランザクションの性能を向上させるための最適化項目を、次の文字列で指定します。

base

同期点処理全体（プリペア処理、コミット処理、およびロールバック処理）を最適化します。OpenTP1 のトランザクション制御は 2 相コミット方式で実行しているため、二つのトランザクションブランチ間のコミット制御には、4 回のプロセス間通信が必要となります。

次の条件をすべて満たす場合、親トランザクションブランチが子トランザクションブランチのコミット処理を代わりに実行することで、コミット制御に必要な 4 回のプロセス間通信を削減します。

1. 親トランザクションブランチと、子トランザクションブランチが同一 OpenTP1 下にあること。
2. 親トランザクションブランチが、子トランザクションブランチを同期応答型 RPC で呼び出していること。

3. 子トランザクションブランチでアクセスしたリソースマネージャの XA インタフェース用オブジェクトが、親トランザクションブランチにもリンケージされていること。

asyncprepare

base の指定条件を満たしていないため同期点処理全体の最適化ができない場合に、プリペア処理を最適化します。

次の条件をすべて満たす場合、親トランザクションブランチから発行された RPC によって子トランザクションブランチがサービス要求を実行したときに、RPC のリターン前にプリペア処理を実行することで、2 回のプロセス間通信を削減します。

1. base を指定した最適化ができないこと。
2. 親トランザクションブランチが、子トランザクションブランチを同期応答型 RPC で呼び出していること。

ただし、この最適化を実行した場合、親トランザクションブランチが発行した同期応答型 RPC の応答時間が遅くなります。また、子トランザクションブランチは、プリペア処理からコミット処理までの間隔（親トランザクションブランチからの支持がないとトランザクションを決着できない状態）が大きくなります。そのため、親トランザクションブランチの OpenTP1 がシステムダウンし、トランザクションブランチ間の連絡ができなくなると、ジャーナルファイルのスワップやチェックポイントダンプファイルの有効化が遅れ、子トランザクションブランチの OpenTP1 もシステムダウンする場合があります。

トランザクション最適化項目は、重複して指定できます。ただし、優先順位は次のようになります（1.> 2.）。

1. base
2. asyncprepare

なお、このオペランドは、クライアント環境定義の DCCLTTROPTIITEM オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです（1.> 2.）。

1. クライアント環境定義
2. クライアントサービス定義

●trn_watch_time=トランザクション同期点処理時の最大通信待ち時間

～ 〈符号なし整数〉 ((1～65535)) 《120》 (単位：秒)

トランザクションの同期点処理で、トランザクションブランチ間で行う通信（プリペア、コミット、ロールバック指示、または応答など）の受信待ち時間の最大値を指定します。

指定時間を過ぎても指示または応答がない場合は、該当するトランザクションブランチが 2 相コミットの 1 相目完了前であればロールバックさせ、1 相目完了後であればトランザクションサービスのシステムプロセスでトランザクション決着処理を再試行します。

なお、このオペランドは、クライアント環境定義の DCCLTTRWATCHTIME オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです (1.> 2.)。

1. クライアント環境定義
2. クライアントサービス定義

●trn_rollback_information_put=no|self|remote|all

～《no》

トランザクションブランチがロールバックした場合に、ロールバック要因に関する情報をログに取得するかどうかを指定します。

no

ロールバック情報を取得しません。

self

ロールバック要因が発生したトランザクションブランチでだけ、ログにロールバック情報を取得します。

remote

self に加え、他ノードのトランザクションブランチからロールバック要求されたトランザクションブランチでも、ログにロールバック情報を取得します。

all

remote に加え、自ノードのトランザクションブランチからロールバック要求されたトランザクションブランチでも、ログにロールバック情報を取得します。

なお、このオペランドは、クライアント環境定義の DCCLTTRRBINFO オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです (1.> 2.)。

1. クライアント環境定義
2. クライアントサービス定義

●trn_limit_time=トランザクションブランチ最大実行可能時間

～〈符号なし整数〉((0～65535))《0》(単位：秒)

トランザクションブランチの最大実行可能時間を指定します。

トランザクションブランチを開始してから同期点処理が終了するまでの時間が、このオペランド指定時間を超えないように dc_rpc_call 関数、dc_rpc_poll_any_replies 関数および同期点処理内で行う通信のタイムアウト時間を次のように自動設定します。

- dc_rpc_call 関数および dc_rpc_poll_any_replies 関数のタイムアウト時間

「 $K \geq$ このオペランド指定時間」の場合は、要求処理を実行しないで、タイムアウトでエラーリターンします。

「 $K <$ このオペランド指定時間」でかつ「 $(\text{このオペランド指定時間} - K) \geq W$ 」の場合は、 W をタイムアウト時間とします。

「 $K <$ このオペランド指定時間」でかつ「 $(\text{このオペランド指定時間} - K) < W$ 」の場合は、 $(\text{このオペランド指定時間} - K)$ をタイムアウト時間とします。

K

現時刻 - トランザクションブランチ開始時刻

W

dc_rpc_call 関数の場合は watch_time オペランド指定時間

dc_rpc_poll_any_replies 関数の場合は引数 timeout 指定時間

- 同期点処理内で行う通信のタイムアウト時間

「 $K \geq$ このオペランド指定時間」の場合は、タイムアウト時間を 1 秒とします。

「 $K <$ このオペランド指定時間」でかつ「 $(\text{このオペランド指定時間} - K) \geq W$ 」の場合は、 W をタイムアウト時間とします。

「 $K <$ このオペランド指定時間」でかつ「 $(\text{このオペランド指定時間} - K) < W$ 」の場合は、 $(\text{このオペランド指定時間} - K)$ をタイムアウト時間とします。

K

現時刻 - トランザクションブランチ開始時刻

W

trn_watch_time オペランド指定時間

trn_watch_time オペランドを省略した場合は watch_time オペランド指定時間

上記の受信待ち以外の処理で時間が掛かった場合は、このオペランド指定時間以内にトランザクションブランチが終了しないことがあります。

同期点処理開始前にこのオペランドの指定時間が経過した場合、そのトランザクションはロールバックします。

0 を指定した場合は、時間監視をしません。

なお、このオペランドは、クライアント環境定義の DCCLTTRLIMITTIME オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです (1.> 2.)。

1. クライアント環境定義
2. クライアントサービス定義

●trn_rollback_response_receive=Y|N

～《Y》

RPC 先トランザクションブランチにロールバック指示を送信したあと、**ロールバック完了通知を受信するかどうか**を指定します。N を指定した場合、RPC 先トランザクションブランチからのロールバック完了通知を受信しないで（RPC 先トランザクションブランチのロールバック処理の完了を待たないで）自トランザクションブランチを終了します。

なお、このオペランドは、クライアント環境定義の DCCLTTRRBRCV オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです（1.> 2.）。

1. クライアント環境定義
2. クライアントサービス定義

●trn_partial_recovery_type=type1|type2|type3

～《type1》

UAP 障害時のトランザクション同期点処理方を指定します。

クライアントサービス定義でこのオペランドを指定する場合は、CUP からトランザクションを開始する場合だけに有効です。

RPC がタイムアウトし、RPC 発行先プロセスのアドレスが未解決の場合やトランザクション実行中の UAP がダウンした場合に、トランザクションブランチ間の連絡がスムーズにできないで、トランザクションの決着に時間が掛かることがあります。

このオペランドでは、次に示す障害が発生した場合のトランザクション同期点処理方を、指定値に示す三つの方式から選択して指定します。

（障害 1）RPC がタイムアウトした場合

この場合、RPC 発行元トランザクションブランチは、サービス要求がどのプロセスで実行されているかがわからないため、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信できないで、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチがトランザクション同期点メッセージ待ちとなり、トランザクションの決着に時間が掛かります。

（障害 2）RPC 発行元 UAP が RPC の応答受信前にダウンした場合

この場合、RPC 発行元トランザクションブランチは、サービス要求がどのプロセスで実行されているかがわからないため、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信できないで、RPC 発行先トランザクションブランチはトランザクション同期点メッセージ待ちとなり、トランザクションの決着に時間が掛かります。

(障害 3) RPC 発行先 UAP からの応答受信後に RPC 発行元 UAP と RPC 発行先 UAP がほぼ同時にダウンした場合

この場合、それぞれのトランザクションブランチを引き継いだトランザクション回復プロセスは、相手 UAP プロセスのダウンを知らないため、すでに存在しない UAP プロセスにトランザクション同期点メッセージを送信してしまい、トランザクションの決着に時間が掛かることがあります。

type1

(障害 1) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

(障害 2) が発生した場合、RPC 発行元トランザクションブランチは、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信しないでトランザクションを決着します。RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

(障害 3) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

type2

(障害 1) が発生してトランザクションをコミットする場合は type1 と同じです。

(障害 1) が発生してトランザクションをロールバックする場合、または (障害 2) が発生した場合は、RPC 発行元トランザクションブランチは、RPC 発行先トランザクションブランチが存在するノードのトランザクションサービスプロセスにトランザクション同期点メッセージを送信後、トランザクションを決着します。トランザクション同期点メッセージを受信したトランザクションサービスプロセスは、該当するトランザクションブランチを処理中のプロセスに、トランザクション同期点指示を送信します。

(障害 3) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

type3

(障害 1) が発生してトランザクションをコミットする場合は、type1 と同じです。

(障害 1) が発生してトランザクションをロールバックする場合、(障害 2) が発生した場合、または (障害 3) が発生した場合、相手トランザクションブランチが存在するノードのトランザクションサービスプロセスに、トランザクション同期点メッセージを送信します。トランザクション同期点メッセージを受信したトランザクションサービスプロセスは、該当するトランザクションブランチを処理中のプロセスに、トランザクション同期点指示を送信します。

次に示す場合、このオペランドに type2 または type3 を指定しても、トランザクションの決着に時間が掛かることがあります。

1. RPC 実行中に、RPC 発行先 UAP の状態が変更となり(負荷増加, UAP 終了, UAP 閉塞など), ほかのノードの同一 UAP にサービス要求が再転送された場合
2. 相手先の OpenTP1 がこのオプションをサポートしていないバージョンの場合

3. 相手先トランザクションブランチがトランザクション同期点メッセージ受信処理以外で時間が掛かっている場合

なお、このオペランドは、クライアント環境定義の DCCLTTRRECOVERYTYPE オペランドでも指定できます。クライアント環境定義については、マニュアル「OpenTP1 クライアント使用の手引 TP1/Client/W, TP1/Client/P 編」を参照してください。

指定値の優先順位は次のとおりです (1.> 2.)。

1. クライアント環境定義
2. クライアントサービス定義

●trn_completion_limit_time=トランザクション完了限界時間

～〈符号なし整数〉((0～65535))《0》(単位：秒)

トランザクションブランチの開始から終了までの最大実行時間を指定します。指定時間を超えた場合、このトランザクションブランチのプロセスが異常終了したあとに、トランザクションブランチが回復プロセスによってコミットまたはロールバックのどちらかに決着して終了します。0 を指定した場合は、トランザクションブランチの最大実行時間を監視しません。

UAP が異常終了した場合に閉塞するかどうかは hold オペランド、term_watch_time オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

このオペランドの監視対象区間は、dc_trn_begin 関数などの呼び出しやサービス関数の開始によるトランザクションの開始から、トランザクションの同期点処理終了情報 (TJ) の取得後であるトランザクションブランチの終了までです。ただし、トランザクションの最適化が行われた場合、サーバ側のトランザクションブランチに対する監視が終了するのは、クライアント側に応答を返したあとです。このオペランドの監視対象区間の詳細および各種タイマ監視との関係については、「[付録 A.2 トランザクションの時間監視](#)」を参照してください。

●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)

CUP からのトランザクショナル RPC 要求、常設コネクション要求を、トランザクショナル RPC 実行プロセス、CUP 実行プロセスに渡す前にいったん格納しておく共用メモリのプールの大きさを指定します。

次の計算式で求めた値を指定してください。

メッセージ格納バッファプール長 = $D \times C$

D：制御データ長 (256 バイト)

C：同時に発生するトランザクショナル RPC 要求数、常設コネクション要求数のうち大きい値

なお、この計算式の算出結果が 8196 より小さい場合は、指定する必要はありません。

●watch_time=最大応答待ち時間

～〈符号なし整数〉((0～65535)) (単位：秒)

CUP から、トランザクションを開始した場合や、常設コネクションを確立した場合、CUP から発行される RPC はクライアント拡張サービスの実行プロセスが代理実行します。このオペランドでは、実行プロセスが代理実行する RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

指定時間を過ぎても応答がない場合は、受信タイムアウトとして CUP にエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。

クライアント環境定義に DCWATCHTIMINHERIT=Y と定義されている場合は、CUP 側のタイマ値が有効になり、このオペランドに指定されたタイマ値は無視されます。

このオペランドの指定を省略した場合、システム共通定義の watch_time オペランドに指定した値を仮定します。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

なし。

注意事項

CUP から起動するトランザクションは、parallel_count オペランドで指定したプロセス数だけ同時に並行実行します。このプロセスのサーバ名は_clttrn です。trnls コマンドでこのプロセスのトランザクション状態を表示できます。

また、トランザクションブランチ限界経過時間に達した場合などの要因によって、プロセスがキャンセルされた場合に表示するメッセージにも、このサーバ名が表示されます。

IST サービス定義

形式

set 形式

```
[set ist_node=ノード名 [, ノード名] ...]  
[set ist_node_group=ノードグループ名]
```

コマンド形式

```
istdef テーブル名 レコード長 レコード数
```

機能

IST サービスを使用するための実行環境を定義します。

説明

set 形式のオペランド

●ist_node=ノード名 [, ノード名] ...

～《4 文字の識別子》

IST テーブルを操作するノード名を指定します。

各ノードのシステム共通定義の node_id オペランドで指定した、ノード識別子を指定します。IST テーブルに更新が発生した場合、ここで指定したノードに反映されます。

ノード名は最大 128 個指定できます。

なお、ist_node_group オペランドを指定しない場合は、ist_node オペランドに一つ以上のノード名を必ず指定してください。

●ist_node_group=ノードグループ名

～《1～8 文字の識別子》

IST テーブルを操作するノードグループ名を指定します。

マルチノード構成定義の dcmarea -m オプションで指定したマルチノードエリア識別子、または-g オプションで指定したマルチノードサブエリア識別子を指定します。

ist_node オペランドと ist_node_group オペランドの両方を指定した場合は、ist_node_group オペランドを優先します。

なお、ist_node オペランドを指定しない場合は、ist_node_group オペランドを必ず指定してください。

コマンド形式

次ページ以降に記述しています。

istdef

名称

IST サービスのテーブル名の定義

形式

istdef テーブル名 レコード長 レコード数

機能

IST サービスを使用してアクセスする IST テーブル名を指定します。

テーブル名, レコード長, レコード数はすべて指定してください。また, IST サービス定義内に IST テーブルを, 最低一つ定義してください。

1 ノードに定義できるテーブル数は, 最大 64 です。ただし, 定義したすべての IST テーブルのテーブル長の合計を 64 キロバイト以下にしてください。計算式を次に示します。

$\sum_{1}^n (\text{各テーブルの1レコード長} \times \text{レコード数}) \leq 64 \text{キロバイト}$

n : 定義したテーブル数

オプション

●テーブル名

～ 〈1～8 文字の識別子〉

すべての IST テーブル内で一意となる名称を指定します。

●レコード長

～ 〈符号なし整数〉 ((4～65536)) (単位 : バイト)

IST テーブルのレコード長を 4 の倍数で指定します。

IST テーブルの全レコードは固定長です。

●レコード数

～ 〈符号なし整数〉 ((1～16384)) (単位 : 個)

IST テーブルが持つレコード数を指定します。

RMM サービス定義

形式

set 形式

```
set rmm_check_services=監視対象RM名  
[set rmm_system_behavior=down|giveup]  
[set rmm_down_with_system=Y|N]  
[set rmm_sysdown_with_rm=Y|N]
```

コマンド形式

なし。

機能

RMM サービスを使用するリソースマネージャ（監視対象 RM）を定義します。

説明

set 形式のオペランド

●rmm_check_services=監視対象 RM 名

～ 〈1～7 文字の識別子〉

RMM サービスで監視するリソースマネージャを指定します。監視対象リソースマネージャの定義ファイルと同じ名称を指定します。

●rmm_system_behavior=down|giveup

～ 《down》

RMM サービスが監視対象リソースマネージャの開始、または再開に失敗した場合、システム全体の開始処理を続行するかどうかを指定します。

down

OpenTP1 を開始しません。

giveup

OpenTP1 開始処理を続行します。

●rmm_down_with_system=Y|N

～ 《Y》

OpenTP1 のダウン時、または dcstop -f コマンド実行時に、監視対象リソースマネージャも同時にダウンさせるかどうかを指定します。

Y

ダウンさせます。

N

ダウンさせません。

システムサービス構成定義の ha_conf オペランドに Y を指定した場合は、このオペランドの指定を省略するか、または Y を指定してください。

●rmm_sysdown_with_rm=Y|N

～《N》

監視対象リソースマネージャのダウン時に、OpenTP1 も同時にダウンさせるかどうかを指定します。

Y

ダウンさせます。

N

ダウンさせません。監視対象リソースマネージャは再起動します。

システムサービス構成定義の ha_conf オペランドに Y を指定した場合は、このオペランドの指定を省略するか、または N を指定してください。

コマンド形式

なし。

監視対象 RM 定義

形式

set 形式

```
set rmm_start_command=監視対象RM開始コマンド
set rmm_stop_command=監視対象RM終了コマンド
set rmm_abort_command=監視対象RM強制終了コマンド
set rmm_get_pid_command=監視対象プロセスID取得コマンド
[set rmm_command_watch_time=監視対象RMコマンド監視時間]
set rmm_command_uid=コマンド実行用ユーザID
set rmm_command_gid=コマンド実行用グループID
[set rmm_start_watch_time=監視対象RM開始処理監視時間]
```

コマンド形式

なし。

機能

RMM サービスのコマンドを定義します。

コマンドの作成については、マニュアル「OpenTP1 運用と操作」を参照してください。

説明

set 形式のオペランド

●rmm_start_command=監視対象 RM 開始コマンド

～ 〈1～127 バイトのパス名〉

監視対象リソースマネージャを開始するためのコマンド名を指定します。

コマンド名は、ルートディレクトリからの絶対パス名で指定してください。

●rmm_stop_command=監視対象 RM 終了コマンド

～ 〈1～127 バイトのパス名〉

監視対象リソースマネージャを終了するためのコマンド名を指定します。

コマンド名は、ルートディレクトリからの絶対パス名で指定してください。

●rmm_abort_command=監視対象 RM 強制終了コマンド

～ 〈1～127 バイトのパス名〉

監視対象リソースマネージャを強制終了するためのコマンド名を指定します。

コマンド名は、ルートディレクトリからの絶対パス名で指定してください。

●rmm_get_pid_command=監視対象プロセス ID 取得コマンド

～ 〈1～127 バイトのパス名〉

監視対象プロセスのプロセス ID を取得するためのコマンド名を指定します。

コマンド名は、ルートディレクトリからの絶対パス名で指定してください。

●rmm_command_watch_time=監視対象 RM コマンド監視時間

～ 〈符号なし整数〉 ((0～7200)) 《1800》 (単位：秒)

監視対象リソースマネージャの各コマンドの監視時間を指定します。指定した時間を過ぎてもコマンドが終了しなかった場合、RMM サービスはコマンド失敗と判断してコマンドを無効にします。

0 を指定した場合、監視時間は無限となります。

このオペランドを指定する場合は、dcstop コマンドで指定した監視時間内になるように指定してください。

●rmm_command_uid=コマンド実行用ユーザ ID

～ 〈符号なし整数〉 ((0～59999))

各監視対象リソースマネージャのコマンド（開始コマンド、終了コマンド、強制終了コマンド、プロセス ID 取得コマンド）を実行するユーザの ID を指定します。

RMM サービスでは、この指定に従って各コマンドのユーザ ID を設定して実行します。このユーザ ID は、各コマンド中で実行されるコマンドや、子プロセスにも引き継がれます。

●rmm_command_gid=コマンド実行用グループ ID

～ 〈符号なし整数〉 ((0～59999))

各監視対象リソースマネージャのコマンド（開始コマンド、終了コマンド、強制終了コマンド、プロセス ID 取得コマンド）を実行するグループの ID を指定します。

RMM サービスでは、この指定に従って各コマンドのグループ ID を設定して実行します。このグループ ID は、各コマンド中で実行されるコマンドや、子プロセスにも引き継がれます。

●rmm_start_watch_time=監視対象 RM 開始処理監視時間

～ 〈符号なし整数〉 ((0～7200)) 《300》 (単位：秒)

RMM サービスが、監視対象リソースマネージャの再開始を待つ時間を指定します。0 を指定した場合は、監視対象リソースマネージャが再開始されるまで待ち続けます。

このオペランドは、監視対象リソースマネージャを自動再開始機能で再開始する場合に有効です。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。

コマンド形式

なし。

拡張 RM 登録定義

形式

set 形式

なし。

コマンド形式

```
trnlnkrm [ [-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] ...'] ...]  
          [-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]
```

機能

dcsetup コマンド内で OpenTP1 提供のリソースマネージャを登録したあと、OpenTP1 提供以外のリソースマネージャを登録するための実行環境を定義します。

拡張 RM 登録定義がない場合、OpenTP1 提供以外のリソースマネージャを登録しません。

拡張 RM 登録定義は、ボーンシェル (/bin/sh) の書式で記述してください。

説明

set 形式

なし。

コマンド形式

次ページ以降に記述しています。

名称

OpenTP1 提供以外のリソースマネージャの登録

形式

```
trnlInkrm [ [-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] ...'] ...]  
            [-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]
```

機能

dcsetup コマンド内で OpenTP1 提供のリソースマネージャを登録したあと、OpenTP1 提供以外のリソースマネージャを登録します。

オプション

●-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] ...

～ 〈1～31 文字の英数字〉

追加する 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 提供のリソースマネージャは指定しないでください。

このオプションで指定したリソースマネージャに対しては、RM スイッチ名、および RM 関連オブジェクト名を指定してください。

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。

●-s RM スイッチ名 [,RM スイッチ名] ...

～ 〈先頭が英字、またはアンダスコアで始まる 1～32 文字の英数字〉

追加する OpenTP1 提供以外のリソースマネージャのスイッチ名を指定します。

スイッチ名については、追加するリソースマネージャの仕様書を参照してください。複数の RM スイッチ名を指定する場合は、RM スイッチ名と RM スイッチ名との間をコンマ (,) で区切ります。

RM スイッチ名と RM 名は、指定した順に対応します。

●-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] …'] …

～ 〈英数字〉

追加する OpenTP1 提供以外の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) の名称を指定します。

一つのリソースマネージャに対して複数の RM 関連オブジェクト名を指定できます。RM 関連オブジェクト名については、追加するリソースマネージャの仕様書を参照してください。

RM 関連オブジェクト名の中にコンマ (,) を記述する必要がある場合は、コンマ (,) の前に'¥'を記述してください。

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。

複数のリソースマネージャに対する RM 関連オブジェクト名を指定する場合は、一つのリソースマネージャに対する RM 関連オブジェクト名の集まりをアポストロフィ (') で囲み、それぞれの集まりの間をコンマ (,) で区切ります。

RM 関連オブジェクト名と RM は、指定した順に対応します。

●-C 'コンパイルオプション名 [コンパイルオプション名] …'

～ 〈1～512 文字の文字列〉

コンパイル実行時に使用するコンパイルオプションを指定します。

指定するコンパイルオプション名はアポストロフィ (') で囲み、コンパイルオプション名にコンマ (,) を指定する必要がある場合は、コンマ (,) の前に'¥'を記述してください。

複数のコンパイルオプション名を指定する場合は、コンパイルオプション名とコンパイルオプション名との間を空白で区切ります。

通常、このオプションを指定する必要はありません。

●-B 'リンケージオプション名 [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を指定します。

指定するリンケージオプション名はアポストロフィ (') で囲み、リンケージオプション名にコンマ (,) を指定する必要がある場合は、コンマ (,) の前に'¥'を記述してください。

複数のリンケージオプション名を指定する場合は、リンケージオプション名とリンケージオプション名との間を空白で区切ります。trlnlkrm コマンドでは、cc コマンドを使用してリンケージを行っています。このため、指定できるリンケージオプションは、cc コマンドで指定できるオプションとなります。

通常、このオプションを指定する必要はありません。

●-l

trnlncrm コマンドの実行経過を、標準出力に出力します。

●-f

OpenTP1 の状態に関係なく、trnlncrm コマンドを強制的に実行します。ただし、OpenTP1 のトランザクションサービス制御プログラム、およびクライアントサービス実行形式プログラムを再作成するため、OpenTP1 がオンライン中の場合は実行できません。

このオプションは、OpenTP1 を正常終了以外（計画停止 A、計画停止 B、強制停止、および異常終了）で終了したあと、使用するリソースマネージャを変更して OpenTP1 を正常開始する場合にだけ指定してください。

このオプションを指定した trnlncrm コマンドが正常終了したあとは、OpenTP1 は再開始できません。

注意事項

拡張 RM 登録定義は、ボーンシェル (/bin/sh) の書式で記述してください。

複数の trnlncrm コマンドを記述した場合は、先頭から順に実行します。

trnlncrm コマンドが正常終了したあとは、OpenTP1 は再開始できません。必ず正常開始してください。

オプションを複数の行に分けて指定する場合は、オプションとオプションとの間に継続符号'¥'を挿入して改行してください。一つのオプションの指定値の途中（例えば、-o オプションで複数の RM 関連オブジェクト名を指定する場合の、RM 関連オブジェクト名と RM 関連オブジェクト名との間）で改行すると、コマンドが正しく実行されない場合があります。

XATMI 通信サービス定義

形式

set 形式

```
[set xatinitapt="自AP名称"]  
[set xatinitaeq="自AE修飾子"]  
[set xat_aso_con_event_svcname="サービスグループ名",  
                                "サービス名"]  
[set xat_aso_discon_event_svcname="サービスグループ名",  
                                   "サービス名"]  
[set xat_aso_failure_event_svcname="サービスグループ名",  
                                    "サービス名"]  
[set max_open_fds=OSI TP通信のアソシエーションによって使用される  
                  FDSの最大値]  
[set max_socket_descriptors=XATMI通信サービスとUAPとの通信で使用  
                             されるFDSの最大値]
```

コマンド形式

```
xatsrvadd -p "他AP名称" -q "他AE修飾子"  
          -s "サービス名" [, "サービス名"] ...
```

機能

TP1/NET/OSI-TP-Extended を使った、XATMI インタフェースでの OSI TP 通信の実行環境を定義します。

説明

set 形式のオペランド

●xatinitapt="自 AP 名称"

～ 〈1～24 文字の 16 進数字〉

OpenTP1 システムごとに付ける自 AP 名称を指定します。

TP1/NET/OSI-TP-Extended のプロトコル固有定義で指定した、自システムが発呼側となる場合の、発呼側の AP 名称と同じ名称を指定してください。

TP1/NET/OSI-TP-Extended のプロトコル固有定義については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended 編」を参照してください。

●xatinitaeq="自 AE 修飾子"

～ 〈符号なし整数〉 ((0～2147483647))

OpenTP1 システムごとに付ける自 AE 修飾子を指定します。

TP1/NET/OSI-TP-Extended のプロトコル固有定義で指定した、自システムが発呼側となる場合の、発呼側の AE 修飾子と同じ修飾子を指定してください。

TP1/NET/OSI-TP-Extended のプロトコル固有定義については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended 編」を参照してください。

●**xat_aso_con_event_svcname="サービスグループ名", "サービス名"**

アソシエーションの確立通知を受信する SPP のサービスグループ名とサービス名を指定します。

サービスグループ名 ～ 〈1～31 文字の識別子〉

アソシエーションの確立通知を受信する、SPP のサービスグループ名を指定します。

サービス名 ～ 〈1～31 文字の識別子〉

アソシエーションの確立通知を受信する、SPP のサービス名を指定します。

このオペランドで指定するサービスグループ名とサービス名は、xat_aso_discon_event_svcname オペランドおよび xat_aso_failure_event_svcname オペランドで指定した、サービスグループ名およびサービス名と同じでもかまいません。同じサービスグループ名とサービス名を指定すると、アソシエーションの確立通知と解放通知を一つのサービス関数で受信できます。

このオペランドの指定を省略すると、通信イベントは通知されません。

●**xat_aso_discon_event_svcname="サービスグループ名", "サービス名"**

アソシエーションの正常解放通知を受信する SPP のサービスグループ名とサービス名を指定します。

サービスグループ名 ～ 〈1～31 文字の識別子〉

アソシエーションの正常解放通知を受信する、SPP のサービスグループ名を指定します。

サービス名 ～ 〈1～31 文字の識別子〉

アソシエーションの正常解放通知を受信する、SPP のサービス名を指定します。

このオペランドで指定するサービスグループ名とサービス名は、xat_aso_con_event_svcname オペランドおよび xat_aso_failure_event_svcname オペランドで指定した、サービスグループ名およびサービス名と同じでもかまいません。同じサービスグループ名とサービス名を指定すると、アソシエーションの確立通知と解放通知を一つのサービス関数で受信できます。

このオペランドの指定を省略すると、通信イベントは通知されません。

●**xat_aso_failure_event_svcname="サービスグループ名", "サービス名"**

アソシエーションの異常解放通知を受信する SPP のサービスグループ名とサービス名を指定します。

サービスグループ名 ～ 〈1～31 文字の識別子〉

アソシエーションの異常解放通知を受信する、SPP のサービスグループ名を指定します。

サービス名 ～ 〈1～31 文字の識別子〉

アソシエーションの異常解放通知を受信する、SPP のサービス名を指定します。

このオペランドで指定するサービスグループ名とサービス名は、`xat_aso_con_event_svcname` オペランドおよび `xat_aso_discon_event_svcname` オペランドで指定した、サービスグループ名およびサービス名と同じでもかまいません。同じサービスグループ名とサービス名を指定すると、アソシエーションの確立通知と解放通知を一つのサービス関数で受信できます。

このオペランドの指定を省略すると、通信イベントは通知されません。

●`max_open_fds=OSI TP 通信のアソシエーションによって使用される FDS の最大値`

～〈符号なし整数〉((16～2016))《50》

OSI TP 通信のアソシエーションで使われる FDS の最大値を指定します。TP1/NET/OSI-TP-Extended のアソシエーション(コネクション)の数に依存します。

●`max_socket_descriptors=XATMI 通信サービスと UAP との通信で使用される FDS の最大値`

～〈符号なし整数〉((4～2047))《64》

XATMI 通信サーバと OSI TP 通信を利用する UAP との間の通信で使用される FDS の最大値を指定します。次に示す計算式以上の値が適切です。

↑ OSI TP 通信を利用する UAP の数/0.8 ↑

↑ ↑ : 小数点以下を切り上げます。

コマンド形式

次ページ以降に記述しています。

名称

他システムが提供するサービス名の指定

形式

```
xatsrvadd -p "他AP名称" -q "他AE修飾子"  
          -s "サービス名" [, "サービス名"] ...
```

機能

TP1/NET/OSI-TP-Extended を使った、XATMI インタフェースでの OSI TP 通信を実行する場合に、他 AP 名称と他 AE 修飾子で示す他システムが提供するサービス名を指定します。

オプション

●-p "他 AP 名称"

～ 〈1～24 文字の 16 進数字〉

TP1/NET/OSI-TP-Extended のプロトコル固有定義で指定した、自システムが発呼側となる場合の、着呼側の AP 名称を指定します。

TP1/NET/OSI-TP-Extended のプロトコル固有定義については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended 編」を参照してください。

●-q "他 AE 修飾子"

～ 〈符号なし整数〉 ((0～2147483647))

TP1/NET/OSI-TP-Extended のプロトコル固有定義で指定した、自システムが発呼側となる場合の、着呼側の AE 修飾子を指定します。

TP1/NET/OSI-TP-Extended のプロトコル固有定義については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended 編」を参照してください。

●-s "サービス名" [, "サービス名"] ...

～ 〈1～15 文字の 16 進数字〉

他 AP 名称と他 AE 修飾子で示した他システムで提供される、サービス名を指定します。システム内で一意となる名称を指定してください。

複数のサービス名を指定する場合は、サービス名とサービス名との間をコンマ (,) で区切ります。

メッセージキューサービス定義

形式

set 形式

```
[set que_xidnum=同時実行トランザクション数]
[set que_io_maxrecsize=遅延書き込み最大レコード長]
```

コマンド形式

```
{{quegrp -g キューグループID -f 物理ファイル名
  [-n 入出力バッファ数] [-m 保持メッセージ数]
  [-w 使用容量警告率] [-c 警告解除使用率] }}
```

機能

入出力メッセージのキューイングを管理するための実行環境を定義します。

ここでは、物理ファイルに関する情報だけを定義します。

キューファイルに関する情報は、MCF からの要求によってメッセージキューサービスが動的に追加します。

キューファイルとは、メッセージキューサービスが物理ファイル上に割り当てる論理的なファイルのことです。この論理的なファイルには、**入力キュー**用のキューファイルおよび**出力キュー**用のキューファイルの二つがあります。入力キュー用のキューファイルはサービスグループ単位に作成し、出力キュー用のキューファイルは論理端末単位に作成します。ただし、入力キューおよび出力キューを同一の物理ファイルに割り当てることはできません。

同一物理ファイル上に割り当てられた入力キューまたは出力キューは、同一定義下の環境（入出力バッファ数、保持メッセージ数など）を使用します。

説明

set 形式のオペランド

●que_xidnum=同時実行トランザクション数

～ 〈符号なし整数〉 ((1～4096)) 《256》

同時に実行するトランザクションの最大数を指定します。

メッセージキューを使用する MCF のメッセージ送受信を同時に実行するトランザクション数として、不足が発生しない数を指定します。

●que_io_maxrecsize=遅延書き込み最大レコード長

～ 〈符号なし整数〉 ((0～32000)) 《0》 (単位：バイト)

入出力メッセージを、物理ファイルへ遅延書き込みする場合、そのメッセージの最大レコード長を指定します。

0 を指定した場合は、遅延書き込みをしません。

遅延書き込み要求を実行するためには、1 トランザクションのメッセージの合計長より大きい値を指定する必要があります。

メッセージの合計長の計算式を次に示します。

メッセージの合計長=送受信メッセージ長+ユーザ情報長+セグメント数×16 +物理ファイル上のレコード数×36

ユーザ情報長とは、MCF がメッセージ単位に付加する情報の合計のことです。32 ビットの場合、960 バイト×メッセージ数、64 ビットの場合、1276 バイト×メッセージ数で計算してください。

物理ファイル上のレコード数=↑ (送受信メッセージ長+ユーザ情報長+メッセージ数×16) /物理ファイルのレコード長-36 ↑

↑ ↑ : 小数点以下を切り上げます。

コマンド形式

次ページ以降に記述しています。

名称

メッセージキューファイルの指定

形式

```
{ {quegrp -g キューグループID -f 物理ファイル名  
    [-n 入出力バッファ数] [-m 保持メッセージ数]  
    [-w 使用容量警告率] [-c 警告解除使用率] }
```

機能

物理ファイルと物理ファイルに対応する環境（入出力バッファ数、保持メッセージ数など）の組をキューグループといい、キューグループごとに同一資源を使用します。キューグループに割り当てた ID をキューグループ ID といいます。

キューグループ ID と物理ファイル名は、ほかのキューグループと重複した指定はできません。

物理ファイル名は、ほかのメッセージキューサービス定義と重複した指定はできません。

オプション

●-g キューグループ ID

～ 〈1～8 文字の識別子〉

物理ファイルに割り当てるキューグループ ID を指定します。

MCF が、どの物理ファイルを使用するかを識別するために使用します。

入出力キュー定義 (mcfmqgid -q) の quegrp オペランド、論理端末定義 (mcftalcle -k) の quegrp オペランド（出力キューの場合）、およびアプリケーション属性定義 (mcfaalcap -g) の quegrp オペランド（入力キューの場合）にこのキューグループ ID を指定します。

●-f 物理ファイル名

～ 〈パス名〉

キューグループを構成する物理ファイル名を絶対パス名で指定します。ただし、環境変数は使用できません。

●-n 入出力バッファ数

～ 〈符号なし整数〉 ((2～1024)) 《128》

キューグループを構成するバッファキャッシュ用の入出力バッファの個数を指定します。物理ファイルに複数のキューファイルを割り当てて、バッファキャッシュを行います。バッファキャッシュ用の入出力バッ

ファは共用メモリを使用します。入出力バッファ数の指定が大きいほどディスクに対する入出力回数が少なくなります。

入出力バッファ数の計算式を次に示します。

$L = \lceil \text{平均メッセージ長} / \text{物理ファイルのレコード長} \rceil \times 1 \text{ トランザクションの平均メッセージ数} \times \text{同じ物理ファイルに割り当てているキューファイルを同時にアクセスするトランザクション数}$

$N = L + \text{遅延書き込み時間間隔で、メッセージ有効化時遅延書き込み対象となるレコード数}^*$

$M = L + \lceil \text{遅延書き込み対象メッセージ長} / \text{物理ファイルのレコード長} \rceil \times \text{保持メッセージ数} \times \text{遅延書き込み対象となるメッセージの割合}^* \times \text{物理ファイルに割り当てているキューファイル数}$

↑↑：小数点以下を切り上げます。

注※

統計情報で値を取得しています。

N と M のうち、どちらか小さい方の値を指定します。

●-m 保持メッセージ数

～〈符号なし整数〉((0～1024))《10》

メッセージ送信後、UAP の dc_mcf_resend 関数を使用して、ユーザがメッセージを再送したい場合に備えてキューファイルに残しておくメッセージの数を、キューグループごとに指定します。キューグループには最大で「指定した保持メッセージ数×キューファイル数（このキューグループを使用するサービスグループ数または論理端末数）」分のメッセージが保持されます。

MCF 通信サービスがキューファイルからメッセージを読み出して送信したあとに、通信障害（相手システムとの回線および相手システム内でメッセージを失う）などが発生した場合、メッセージの再送が必要になることがあります。再送するために再読み出しできるメッセージ数は、論理端末単位にここで指定した保持メッセージ数の範囲内です。

送信したメッセージ数がこのオペランドの指定数を超えた場合は、古いメッセージから再読み出しおよびメッセージの再送ができなくなります。

保持メッセージ数は、キューグループを出力キューとして使用する場合にだけ有効です。また、メッセージの再送を使用しない場合、メッセージを保持する必要はありません。キューグループを入力キューとして使用する場合、またはメッセージの再送を使用しない場合は、0 を指定してください。

●-w 使用容量警告率

～〈符号なし整数〉((0～95))《80》(単位：%)

物理ファイルの使用容量に対して警告を出すときの警告率を物理ファイルごとに指定します。

物理ファイルの使用容量（取り出し待ちメッセージと保持メッセージの容量）が、物理ファイルの容量のある一定の割合を超えた場合に、警告メッセージを出力します。

●-c 警告解除使用率

～ 〈符号なし整数〉 ((0～95)) 《0》 (単位：%)

物理ファイルの使用容量（取り出し待ちメッセージと保持メッセージの容量）に対して警告された状態を解除する使用率を、物理ファイルごとに指定します。

物理ファイルの使用率が、このオペランドに指定した値を下回った場合に、警告状態を解除します。解除後、再び使用容量警告率を超えたときは、警告メッセージを出力します。このオペランドに使用容量警告率以上の値を指定した場合、このオペランド値には使用容量警告率を仮定します。

ユーザサービスネットワーク定義

形式

set 形式

なし。

コマンド形式

```
{{dcsvgdef -g サービスグループ名 [, サービスグループ名] ...  
           {-h ホスト名 [:ポート番号] [, ホスト名 [:ポート番号]] ...  
           {-p ポート番号}  
           {-t あて先再選択間隔} [-w] } }}
```

機能

他ノードの TP1/Server Base 管理下の SPP のサービスを、リモート API 機能を介して利用する場合に、その SPP のサービスグループ名とリモート API 機能によるサービスの受信口となるホスト名およびポート番号を指定します。

または、all_node オペランドに指定していないノードの TP1/Server Base 管理下の SPP のサービスを利用する場合に、その SPP のサービスグループ名とその SPP の存在するグローバルドメイン内の任意のノードのホスト名、およびスケジュールサービス定義の scd_port オペランドに指定したポート番号を指定します。複数のホスト名、ポート番号を指定することもできます（リモート API 機能を利用する場合は複数指定できません）。

ユーザサービス定義の rpc_destination_mode オペランドに namd、または definition が指定された UAP から呼び出される dc_rpc_call のサービス要求を、ネームサーバへのあて先検索要求なしで実行します。

rpc_destination_mode オペランドに namd が指定された UAP は、ネームサーバへのあて先検索要求がエラーになった場合、この定義コマンドで指定された情報を基にサービス要求を実施します。

rpc_destination_mode オペランドに definition が指定され、かつこの定義コマンドで複数のホストが指定された UAP は、この定義コマンドで指定された情報を基にサービス要求を実施し、エラーになった場合、ネームサーバへのあて先検索要求を実施します。この定義コマンドに指定されているホストが一つだけの場合は、ネームサーバへのあて先検索要求は実施しません。

OpenTP1 は、ユーザサービス定義の rpc_destination_mode オペランドに namd または definition が指定された UAP から呼び出される dc_rpc_call について、第 1 引数に指定されたサービスグループ名を、ユーザサービスネットワーク定義に指定されたサービスグループ名の中から検索します。サービスグループ名が一致する定義が見つかった場合、その定義に指定されているホストとポート番号へサービス要求を送信します。

複数のホスト名を指定した場合、OpenTP1 はランダムにホストを選択してサービス要求を送信します。サービス要求の送信で障害が発生すると、残りのホスト名から再度ランダムにホストを選択します。すべてのホストへのサービス要求の送信が障害になると、dc_rpc_call はエラーリターンします。サービス要

求が成功すると、UAP 内で以降に呼び出す同じサービスグループ名への dc_rpc_call は、あて先再選択間隔を指定していなければ、障害が発生するまで同じホストにサービス要求の送信を継続します。継続中のサービス要求の送信で障害になった場合、今回障害になったホストを除いたすべてのホスト名からランダムに選んだホストに対してサービス要求の送信を試みます。

なお、次の条件が成立する場合、次回に呼び出した dc_rpc_call で、OpenTP1 はランダムにホストを選択してサービス要求を送信します。

rpc_destination_mode オペランドの指定を definition, または省略した場合

- すべてのホストへのサービス要求の送信が障害になり、ネームサーバへのあて先検索要求もエラーとなったため dc_rpc_call がエラーリターンしたとき
- すべてのホストへのサービス要求の送信が障害になり、ネームサーバへのあて先検索要求が成功して dc_rpc_call が成功したとき

rpc_destination_mode オペランドの指定が namd で、次の条件をすべて満たす場合

- ネームサーバへのあて先検索要求がエラー
- すべてのホストへのサービス要求の送信が障害
- dc_rpc_call がエラーリターン
- 次回に呼び出した dc_rpc_call でネームサーバへのあて先検索要求がエラー

複数の dcsvgdef 定義コマンドに同じサービスグループ名を指定した場合、ユーザサービスネットワーク定義ファイルで先に記述されている dcsvgdef 定義コマンドの指定が有効になります。リモート API 機能を介して要求するサービスの情報か、all_node オペランドに指定していないノード上のサービスの情報の区別は、-w オプションの指定の有無によって判断します。

dcsvgdef 定義コマンドのサービスグループ名に対し、XATMI インタフェースでサービスを要求しないでください。この制限に反した場合の動作は保証されません。

-w オプションを指定しない dcsvgdef 定義コマンドのサービスグループ (SPP) が、atomic_update=N であり、このサービスグループに対して、トランザクション内から dc_rpc_call を発行した場合、dc_rpc_call は DCRPCER_TRNCHK でエラーリターンします。この場合、SPP の atomic_update オペランドに 'Y' を指定するか、dc_rpc_call の flags に DCRPC_TPNOTRAN を指定する必要があります。

-w オプションを指定しない dcsvgdef 定義コマンドのサービスグループ名に対して、dc_rpc_call を実行し、性能検証用トレースを取得していた場合、サーバ側の性能検証用トレース情報とリンクしません。つまり、クライアント UAP で取得した性能検証用トレースの通番は、サーバ側に引き継がれないため、サーバ側で取得した性能検証用トレースには、新たに採番した通番が出力されます。

-w オプションを指定した dcsvgdef 定義コマンドのサービスグループ名に対し、非同期 RPC を要求した場合は、このユーザサービスネットワーク定義を無効とし、従来どおりネーム情報検索によって処理します。また、-w オプションを指定した dcsvgdef 定義コマンドのサービスグループに対し、トランザクションとしてサービス要求しても、無条件に非トランザクションモードで処理します。

-w オプションを指定した dcsvgdef 定義コマンドのサービスグループ名に対して、dc_rpc_call を実行した場合、RPC トレースは取得しません。ユーザサービスネットワーク定義でリモート API 機能を介したノード上のサービスとして定義したサービスグループへの dc_rpc_call については、クライアント UAP のシステム定義で RPC トレースの取得を指定しても、クライアントのトレース情報を取得しません。

-w オプションを指定した dcsvgdef 定義コマンドのサービスグループ名に対して、dc_rpc_call を実行した場合、レスポンス統計情報／通信遅延時間統計情報は取得しません。ユーザサービスネットワーク定義によってリモート API 機能を介したノード上のサービスとして、定義したサービスグループへの dc_rpc_call は、システム定義でレスポンス統計情報／通信遅延時間統計情報の取得を指定しても、統計情報を取得しません。

-w オプションを指定し、TP1/Server Base 間でリモート API 機能を使用する場合（例えば、アプリケーションゲートウェイ型ファイアウォールなどのゲートウェイを介して RPC をする場合など）、トランザクション属性で dc_rpc_call 関数を発行してもトランザクションにはなりません。したがって、トランザクション内から連鎖 RPC を開始し、同期点処理で連鎖 RPC を終了させる運用は、リモート API 機能を使用した場合には正しく動作しません。flags 引数に DCNOFLAGS を指定した dc_rpc_call 関数で、明示的に連鎖 RPC を終了するようにしてください。

説明

set 形式

なし。

コマンド形式

次ページ以降に記述しています。

名称

通信相手のサービス情報の指定

形式

```
{ {dcsvgdef  -g サービスグループ名 [, サービスグループ名] ...  
             {-h ホスト名 [:ポート番号] [, ホスト名 [:ポート番号]] ...  
             [-p ポート番号]  
             [-t あて先再選択間隔] [-w] } }
```

機能

他ノードの TP1/Server Base 管理下の SPP のサービスを、リモート API 機能を介して利用する場合に、その SPP のサービスグループ名とリモート API 機能によるサービスの受信口となるホスト名およびポート番号を指定します。また、リモート API 機能を使用するかどうかも指定します。

または、all_node オペランドに指定していないノードの TP1/Server Base 管理下の SPP のサービスを利用する場合に、その SPP のサービスグループ名とその SPP の存在するグローバルドメイン内の任意のノードのホスト名、およびスケジュールサービス定義の scd_port オペランドに指定したポート番号を指定します。複数のホスト名、ポート番号を指定することもできます（リモート API 機能を利用する場合は複数指定できません）。

なお、SPP が稼働するサーバ側の TP1/Server Base のバージョンは、03-03 以降でなければなりません。

ユーザサービス定義の rpc_destination_mode オペランドに namd、または definition が指定された UAP から呼び出される dc_rpc_call のサービス要求を、ネームサーバへのあて先検索要求なしで実行します。

rpc_destination_mode オペランドに namd が指定された UAP は、ネームサーバへのあて先検索要求がエラーになった場合、この定義コマンドで指定された情報を基にサービス要求を実施します。

rpc_destination_mode オペランドに definition が指定され、かつこの定義コマンドで複数のホストが指定された UAP は、この定義コマンドで指定された情報を基にサービス要求を実施し、エラーになった場合、ネームサーバへのあて先検索要求を実施します。この定義コマンドに指定されているホストが一つだけの場合は、ネームサーバへのあて先検索要求は実施しません。

OpenTP1 は、ユーザサービス定義の rpc_destination_mode オペランドに namd または definition が指定された UAP から呼び出される dc_rpc_call について、第 1 引数に指定されたサービスグループ名を、ユーザサービスネットワーク定義に指定されたサービスグループ名の中から検索します。サービスグループ名が一致する定義が見つかった場合、その定義に指定されているホストとポート番号へサービス要求を送信します。

複数のホスト名を指定した場合、OpenTP1 はランダムにホストを選択してサービス要求を送信します。サービス要求の送信で障害が発生すると、残りのホスト名から再度ランダムにホストを選択します。すべてのホストへのサービス要求の送信が障害になると、dc_rpc_call はエラーリターンします。サービス要

求が成功すると、UAP 内で以降に発行する同じサービスグループ名への dc_rpc_call は、あて先再選択間隔を指定していなければ、障害が発生するまで同じホストにサービス要求の送信を継続します。継続中のサービス要求の送信で障害になった場合、障害になったホストを除いたすべてのホスト名からランダムに選んだホストに対してサービス要求の送信を試みます。

なお、次の条件が成立する場合、次回に呼び出した dc_rpc_call で、OpenTP1 はランダムにホストを選択してサービス要求を送信します。

rpc_destination_mode オペランドの指定を definition, または省略した場合

- すべてのホストへのサービス要求の送信が障害になり、ネームサーバへのあて先検索要求もエラーとなったため dc_rpc_call がエラーリターンしたとき
- すべてのホストへのサービス要求の送信が障害になり、ネームサーバへのあて先検索要求が成功して dc_rpc_call が成功したとき

rpc_destination_mode オペランドの指定が namd で、次の条件をすべて満たす場合

- ネームサーバへのあて先検索要求がエラー
- すべてのホストへのサービス要求の送信が障害
- dc_rpc_call がエラーリターン
- 次回に呼び出した dc_rpc_call でネームサーバへのあて先検索要求がエラー

複数のホスト名を指定した場合の、dc_rpc_call の動作例を次の図に示します。

図 3-7 dcsvgdef 定義コマンドで複数のホスト名を指定した場合の dc_rpc_call の動作例

定義

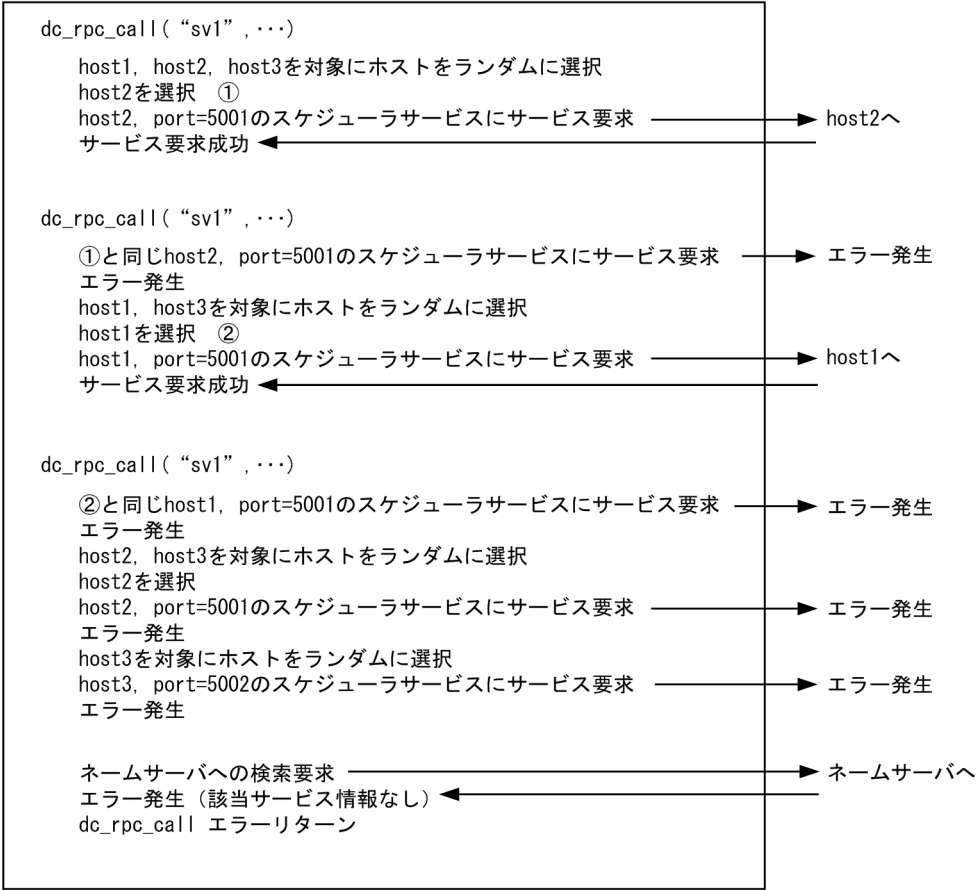
ユーザサービスネットワーク定義

dcsvgdef -g sv1 -h host1,host2,host3:5002 -p 5001

ユーザサービス定義

rpc_destination_mode = definition

dc_rpc_callの動作



複数の dcsvgdef 定義コマンドに同じサービスグループ名を指定した場合、ユーザサービスネットワーク定義ファイルで先に記述されている dcsvgdef 定義コマンドの指定が有効になります。リモート API 機能を介して要求するサービスの情報が、all_node オペランドに指定していないノード上のサービスの情報の区別は、-w オプションの指定の有無によって判断します。

dcsvgdef 定義コマンドのサービスグループ名に対し、XATMI インタフェースでサービスを要求しないでください。この制限に反した場合の動作は保証されません。

-w オプションを指定しない dcsvgdef 定義コマンドのサービスグループ（SPP）が、atomic_update=N であり、このサービスグループに対して、トランザクション内から dc_rpc_call を発行した場合、dc_rpc_call は DCRPCER_TRNCHK でエラーリターンします。この場合、SPP の atomic_update オペランドに'Y'を指定するか、dc_rpc_call の flags に DCRPC_TPNOTRAN を指定する必要があります。

-w オプションを指定しない dcsvgdef 定義コマンドのサービスグループ名に対して、dc_rpc_call を実行し、性能検証用トレースを取得していた場合、サーバ側の性能検証用トレース情報とリンクしません。つまり、クライアント UAP で取得した性能検証用トレースの通番は、サーバ側に引き継がれないため、サーバ側で取得した性能検証用トレースには、新たに採番した通番が出力されます。

-w オプションを指定した dcsvgdef 定義コマンドのサービスグループ名に対して、非同期 RPC を要求した場合は、このユーザサービスネットワーク定義を無効とし、従来どおりネーム情報検索によって処理します。また、-w オプションを指定した dcsvgdef 定義コマンドのサービスグループに対し、トランザクションとしてサービス要求しても、無条件に非トランザクションモードで処理します。

-w オプションを指定した dcsvgdef 定義コマンドのサービスグループ名に対して、dc_rpc_call を実行した場合、RPC トレースは取得しません。ユーザサービスネットワーク定義でリモート API 機能を介したノード上のサービスとして定義したサービスグループへの dc_rpc_call については、クライアント UAP のシステム定義で RPC トレースの取得を指定しても、クライアントのトレース情報を取得しません。

-w オプションを指定した dcsvgdef 定義コマンドのサービスグループ名に対して、dc_rpc_call を実行した場合、レスポンス統計情報／通信遅延時間統計情報は取得しません。ユーザサービスネットワーク定義によってリモート API 機能を介したノード上のサービスとして、定義したサービスグループへの dc_rpc_call は、システム定義でレスポンス統計情報／通信遅延時間統計情報の取得を指定しても、統計情報を取得しません。

-w オプションを指定し、TP1/Server Base 間でリモート API 機能を使用する場合（例えば、アプリケーションゲートウェイ型ファイアウォールなどのゲートウェイを介して RPC をするなど）、トランザクション属性で dc_rpc_call 関数を発行してもトランザクションにはなりません。したがって、トランザクション内から連鎖 RPC を開始し、同期点処理で連鎖 RPC を終了させる運用は、リモート API 機能を使用した場合には正しく動作しません。flags 引数に DCNOFLAGS を指定した dc_rpc_call 関数で、明示的に連鎖 RPC を終了するようにしてください。

オプション

●-g サービスグループ名

～ 〈1～31 文字の識別子〉

リモート API 機能を介して利用するサービス、またはシステム共通定義の all_node オペランドで指定していないノード上のサービスの、サービスグループ名を指定します。このオプションは、「サービスグループ名の先頭にある文字（1 文字以上） + *」の形式で、複数のサービスグループを一括して指定できます。

なお、all_node オペランドで指定していないノード上のサービスのサービスグループ名を指定する場合、このサービスはユーザサービス定義の receive_from オペランドで queue を指定した SPP でなければなりません。

●-h ホスト名:ポート番号

～ 〈1～255 文字の識別子〉

リモート API 機能によるサービスの受信口となるホスト名、またはシステム共通定義の all_node オペランドで指定していないノードの OpenTP1 の通信に使用するホスト名を指定します。識別子に使用できる文字は英数字ピリオド、およびハイフンです（ただし、IP アドレス形式は除く）。ホスト名は、/etc/hosts ファイルまたは DNS などで、IP アドレスとのマッピングができなければなりません。

ホスト名の後ろにコロンで区切ってポート番号を指定できます。ポート番号の有効範囲は 5001～65535 です。ポート番号を省略した場合、-p オプションに指定したポート番号が仮定されます。-h オプションと-p オプションの両方のポート番号を省略することはできません。両方のポート番号を省略した場合、KFCA00340-W のメッセージが出力されます。

コンマで区切り、複数のホスト名を指定できます。リモート API 機能を利用する場合（-w オプションを指定した場合）、複数のホスト名を指定することはできません。リモート API 機能を利用する場合に、複数のホスト名を指定したときは、KFCA00340-W のメッセージが出力されます。

このオプションにホスト名を一つだけ指定した場合、ユーザサービス定義の rpc_destination_mode オペランドに definition を指定しても、ネームサーバへのあて先検索要求は実行されません。

●-p ポート番号

～〈符号なし整数〉((1～65535))

リモート API 機能によるサービスの受信口となるポート番号、またはシステム共通定義の all_node オペランドで指定していないノード上の OpenTP1 のスケジュールサービス定義の scd_port オペランドで指定したポート番号を指定します。

-w オプションを指定し、リモート API 機能によるサービスの受信口となるポート番号を指定する場合は、1 から 65535 の範囲のポート番号を指定できます。-w オプションを指定しないでスケジュールサービス定義の scd_port オペランドに指定したポート番号を指定する場合は、5001 から 65535 の範囲のポート番号を指定できます。指定できない範囲のポート番号を指定した場合、KFCA00340-W のメッセージが出力されます。

●-t あて先再選択間隔

～〈符号なし整数〉((0～65534)) (単位：秒)

-h オプションに複数のホスト名を指定し、サービス要求の送信を継続しているとき、あて先をランダムに再選択する時間間隔を秒単位で指定します。

-h オプションにホスト名を一つだけ指定した場合、このオプションで指定した値は有効になりません。-h オプションに指定したあて先へのサービス要求の送信が成功し、あて先との通信を開始したあと、あて先へのサービス要求の送信ごとに、このオプションに指定した時間以上経過しているかどうか判定します。このオプションに指定した時間以上経過していた場合、あて先をランダムに再選択します。このオプションに指定した時間以上経過していても、あて先へのサービス要求の送信が行われていなければ再選択は行いません。

なお、あて先の再選択で、再選択前と同一のあて先を選択する場合があります。このオプションに 0 を指定した場合、サービス要求の送信ごとにあて先をランダムに再選択します。このオプションを省略した場合、サービス要求が成功した送信先で障害が発生するまで同じあて先にサービス要求の送信を継続します。

あて先と通信を開始したあとの経過時間を次に示します。

- ユーザサービス定義の `rpc_destination_mode` オペランドに `definition` を指定した場合

経過時間の開始は、`-h` オプションに指定されたあて先にサービス要求の送信が成功した時刻です。サービス要求の送信で障害が発生した場合、経過時間の開始は、残りのホスト名からランダムにあて先を選択し、サービス要求の送信が成功した時刻になります。ネームサーバのあて先検索で取得したあて先にサービス要求の送信が成功した場合、この時間は経過時間に含まれません。

ユーザサービス定義の `rpc_destination_mode` オペランドに `definition` を指定した場合の経過時間について次に示します。

図 3-8 ユーザサービス定義の rpc_destination_mode オペランドに definition を指定した場合の経過時間

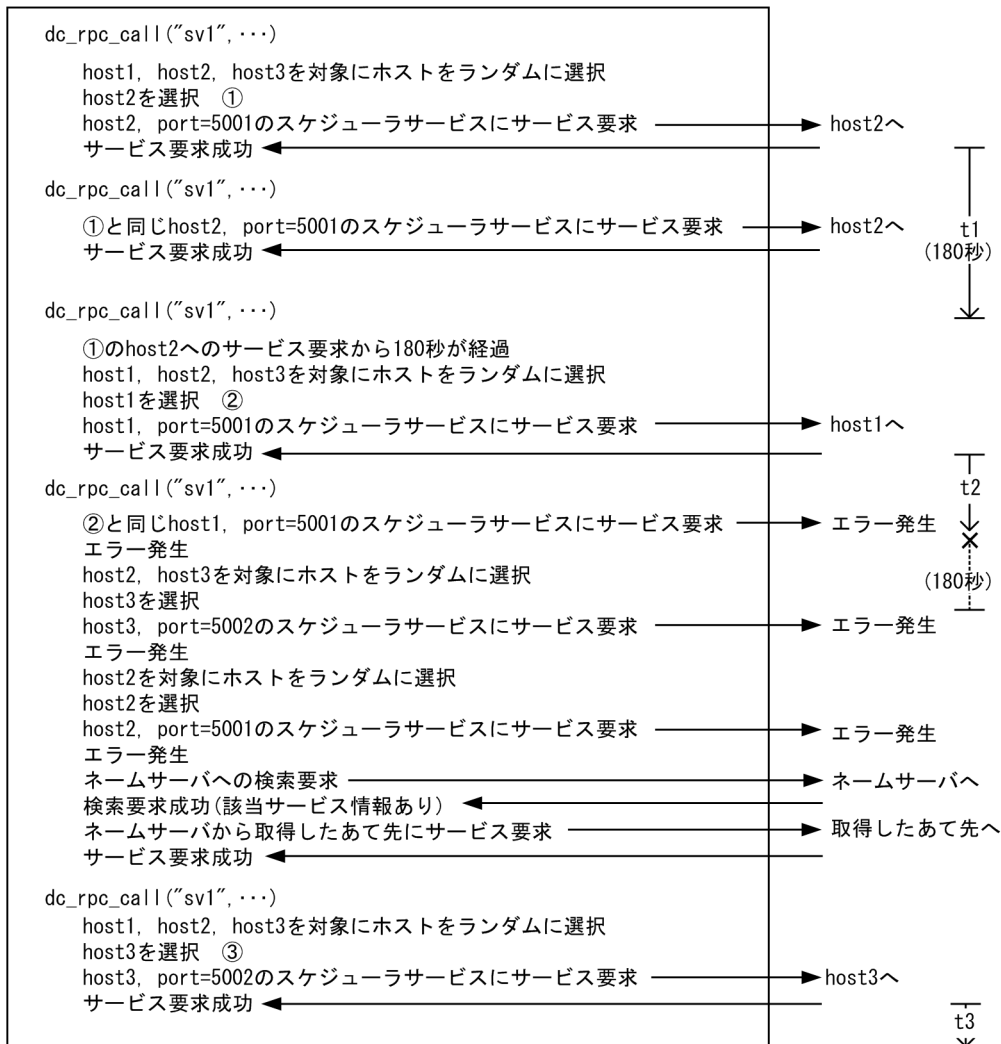
定義

```

ユーザサービスネットワーク定義
dcsvgdef -g sv1 -h host1,host2,host3:5002 -p 5001 -t 180

ユーザサービス定義
set rpc_destination_mode = definition
    
```

dc_rpc_callの動作



(凡例)

- t1 : host2へのサービス要求が成功してからの経過時間
- t2 : host1へのサービス要求が成功してからの経過時間
次のサービス要求がエラーとなったため、経過時間は打ち切られる。
- t3 : host3へのサービス要求が成功してからの経過時間

- ユーザサービス定義の rpc_destination_mode オペランドに namd を指定した場合

経過時間の開始は、-h オプションに指定されたあて先にサービス要求の送信が成功した時刻です。サービス要求の送信を継続し、ネームサーバのあて先検索で取得したあて先にサービス要求の送信が成功した場合、この時間も経過時間に含みます。

ユーザサービス定義の rpc_destination_mode オペランドに namd を指定した場合の経過時間について次に示します。

図 3-9 ユーザサービス定義の `rpc_destination_mode` オペランドに `namd` を指定した場合の経過時間

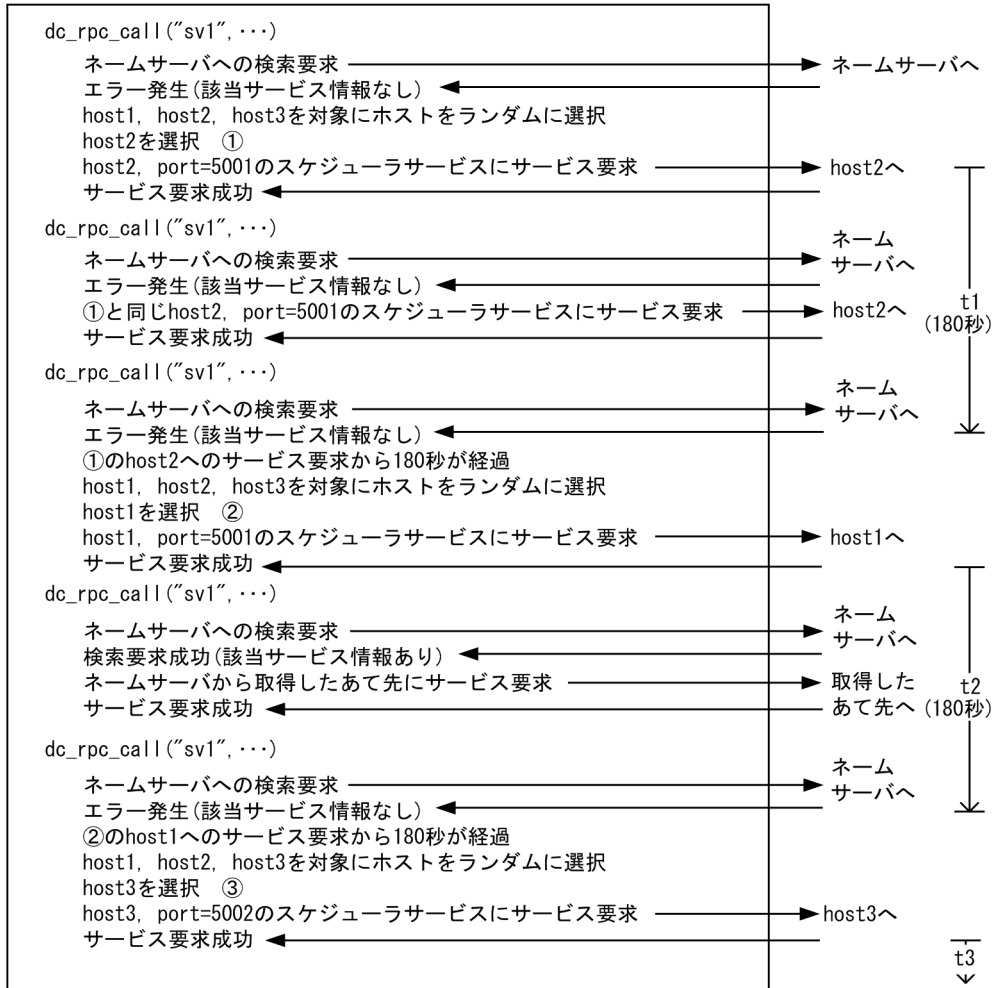
定義

```

ユーザサービスネットワーク定義
dcsvgdef -g sv1 -h host1,host2,host3:5002 -p 5001 -t 180

ユーザサービス定義
set rpc_destination_mode = namd
    
```

`dc_rpc_call` の動作



(凡例)

- t1 : host2へのサービス要求が成功してからの経過時間
- t2 : host1へのサービス要求が成功してからの経過時間
ネームサーバから取得したあて先へのサービス要求も経過時間に含む。
- t3 : host3へのサービス要求が成功してからの経過時間

• 連鎖 RPC を使用した場合

経過時間の開始は、`-h` オプションに指定されたあて先に連鎖 RPC 開始サービス要求の送信が成功した時刻です。あて先をランダムに再選択するかどうかの判定は、新たな連鎖 RPC 開始サービス要求の送信で行います。連鎖 RPC 中のサービス要求の送信、連鎖 RPC 終了サービス要求の送信では、判定を行いません。

連鎖 RPC を使用した場合の経過時間について次に示します。

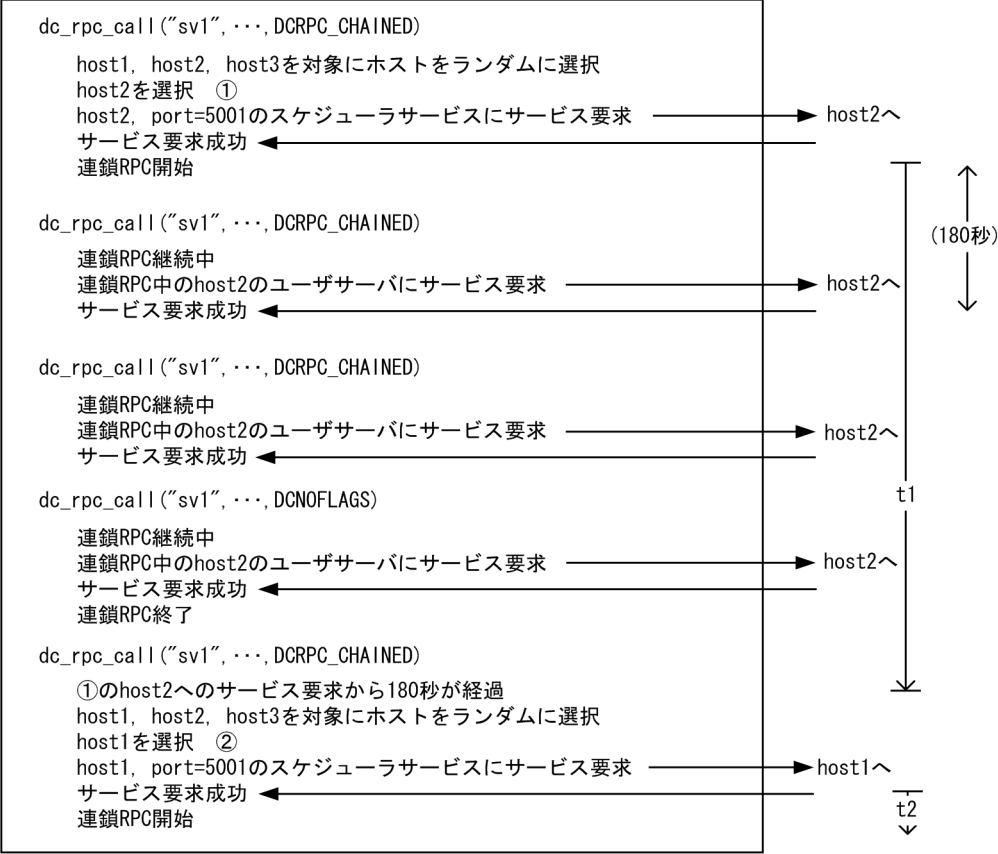
図 3-10 連鎖 RPC を使用した場合の経過時間

定義

```
ユーザサービスネットワーク定義
dcsvgdef -g sv1 -h host1,host2,host3:5002 -p 5001 -t 180

ユーザサービス定義
set rpc_destination_mode = definition
```

dc_rpc_call の動作



- (凡例)
- t1 : host2へのサービス要求が成功してからの経過時間
連鎖RPC中のサービス要求及び連鎖RPC終了のサービス要求も経過時間に含む。
 - t2 : host1へのサービス要求が成功してからの経過時間

●-w

リモート API 機能を使用する場合に指定します。このオプションを指定すると、-h および -p オプションで指定した値が、リモート API 機能によるサービスの受信口の情報であることを示します。

rap リスナーサービス定義

形式

set 形式

```
set rap_listen_port=rapリスナーポート番号
[set rap_parallel_server=rapサーバ数]
[set rap_watch_time=メッセージ送受信最大監視時間]
[set rap_inquire_time=問い合わせ間隔最大時間]
[set nice=プロセスの優先順位の変更]
[set uap_trace_max=UAPトレース格納最大数]
set uid=openTP1管理者のユーザ識別子
[set rpc_response_statistics=Y|N]
[set rpc_trace=Y|N]
[set rpc_trace_name="RPCトレースを取得するファイル名"]
[set rpc_trace_size=RPCトレースを取得するファイルのサイズ]
[set trn_expiration_time=トランザクションブランチ限界経過時間]
[set trn_expiration_time_suspend=Y|N|F]
[set trn_cpu_time=トランザクションブランチCPU監視時間]
[set trf_put=Y|N]
[set trn_statistics_item=統計情報項目 [, 統計情報項目] ...]
[set trn_optimum_item=トランザクション最適化項目
    [, トランザクション最適化項目] ...]
[set trn_watch_time=トランザクション同期点処理時の最大通信待ち時間]
[set trn_rollback_information_put=no|self|remote|all]
[set trn_limit_time=トランザクションブランチ最大実行可能時間]
[set trn_rollback_response_receive=Y|N]
[set trn_partial_recovery_type=type1|type2|type3]
[set rap_inquire_timeout_message=Y|N]
[set rap_connection_assign_type=dynamic|static]
[set rap_max_client=rapリスナーに同時接続する最大クライアント数]
[set rap_notify=Y|N]
[set rap_client_manager_node="ホスト名:ポート番号"
    [, "ホスト名:ポート番号"] ...]
[set rap_max_buff_size=ソケットウィンドウサイズ]
[set rap_io_retry_interval=電文送受信時のリトライ間隔]
[set rap_sock_count=ソケット生成処理のリトライ回数]
[set rap_sock_interval=ソケット生成処理のリトライ間隔]
[set rap_connect_retry_count=コネクション確立処理のリトライ回数]
[set rap_connect_retry_interval=コネクション確立処理のリトライ間隔]
[set rap_listen_backlog=接続待ちキューに保留できる最大要求数]
[set rap_msg_output_interval=クライアント数表示メッセージの出力間隔]
[set rap_recovery_server=リカバリ要求用待機rapサーバ数]
[set rap_connect_interval=コネクション確立処理の間隔]
[set rpc_extend_function=RPCサービスの機能拡張レベル]
[set max_socket_descriptors=ソケット用ファイル記述子の最大数]
[set trn_completion_limit_time=トランザクション完了限界時間]
[set rap_message_id_change_level=メッセージIDの変更レベル]
[set rap_term_disconnect_time=
    rapリスナー終了時のコネクション切断待ち時間]
[set rap_stay_watch_time=rapサーバ割り当て待ち要求の滞留監視時間]
[set rap_stay_warning_interval=滞留警告メッセージの出力間隔]
[set log_audit_out_suppress=Y|N]
[set log_audit_message=監査ログを取得する項目のメッセージID
    [, 監査ログを取得する項目のメッセージID] ...]
```

```
[set ipc_sockctl_highwater=
    ソケットの一時クローズ処理開始数パーセンテージ
    [, ソケットの一時クローズ処理非対象数パーセンテージ] ]
[set ipc_sockctl_watchtime=ソケット再利用可能監視時間]
[set scs_prf_trace_level=リモートAPI機能における性能検証用トレース取得レベル]
[set rap_extend_function=リモートAPI機能の機能拡張レベル]
[set watch_time=最大応答待ち時間]
```

コマンド形式

なし。

機能

リモート API 機能を使用する場合の rap リスナーサービスを指定します。この定義を指定してから、rapdfgen コマンドを実行すると、リモート API 機能を使用するのに必要な rap リスナー用ユーザサービス定義および rap サーバ用ユーザサービス定義が自動的に作成されます。

説明

set 形式のオペランド

●rap_listen_port=rap リスナーポート番号

～ 〈符号なし整数〉 ((5001～65535))

rap リスナーおよび rap サーバが使用するウェルノウンポート番号を指定します。rap リスナーを複数起動する場合はそれぞれのポート番号が重複しないように指定してください。

このオペランドで指定するポート番号は、ほかのプログラムと重複しないようにしてください。

また、OS には任意に割り当てる番号がありますが、この番号も使用しないでください。OS が任意に割り当てる番号は、OS の種別やバージョンによって異なります。詳細については、使用している OS のマニュアルを参照してください。

●rap_parallel_server=rap サーバ数

～ 〈符号なし整数〉 ((1～1024)) 《1》

該当する rap リスナーが管理する rap サーバのサーバ数を指定します。

●rap_watch_time=メッセージ送受信最大監視時間

～ 〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)

rap リスナーおよび rap サーバがメッセージの送受信を開始し始めてから、送受信が完了するまでの監視時間を指定します。監視時間が経過してもメッセージの送受信が完了しない場合はメッセージを出力し、ネットワーク障害と同様の処理を行います。

0 を指定した場合は、時間監視をしません。

●rap_inquire_time=問い合わせ間隔最大時間

～〈符号なし整数〉((0～1048575))《180》(単位：秒)

クライアントが rap リスナーおよび rap サーバに問い合わせ応答をしてから次の問い合わせが来るまでの間隔の最大待ち時間を指定します。この値は rap リスナーまたは rap サーバ側で監視するタイマであり、指定時間を超えても問い合わせがない場合、クライアントがダウンしたとして処理します。

0 を指定した場合は、無限に待ちます。

常駐 SPP からリモート API 機能を使用している場合、このオペランドまたはユーザサービス定義の rpc_rap_inquire_time オペランドに 0 を指定しないでください。指定した場合、rap リスナーが存在するノードの OpenTP1 システムを終了させようとしても、rap リスナーが終了できないため、常駐 SPP が終了するまで無限に待ち続けます。

SPP または SUP のユーザサービス定義で rpc_rap_inquire_time オペランドの指定を省略した場合、このオペランドで指定した値が最大待ち時間となります。どちらも省略した場合は、180 秒を仮定します。

●nice=プロセスの優先順位の変更

～〈符号なし整数〉((0～39))《0》

このサービスグループプロセスの nice 値に加える値を指定します。プロセスの nice 値は正の整数で、この値が大きくなると CPU の優先順位は低くなります。nice の詳細については、ユーザの使用する OS の、該当するマニュアルを参照してください。

●uap_trace_max=UAP トレース格納最大数

～〈符号なし整数〉((0～4095))《32》

UAP トレースの最大格納数を指定します。

0 を指定した場合

UAP トレースを取得しません。

1 以上を指定した場合

指定値+1 の UAP トレースを取得します。

UAP トレースデータファイルまたは退避コアファイルに、UAP トレースの情報として次の情報を取得します。

- サービス関数の開始
- サービス関数の終了
- OpenTP1 のライブラリ関数の入り口または出口※

注※

OpenTP1 のライブラリ関数がリターンするときは、入り口情報を取得した領域を出口情報で上書きするため、UAP トレースの格納数は 1 となります。ただし、TP1/Message Queue、およびオンライン

テスト (TP1/Online Tester) を使用している場合は、OpenTP1 のライブラリ関数の入出力情報と
出力情報で別の領域となるため、UAP トレースの格納数は 2 となります。

UAP トレースを参照することで、次に示す処理の流れを解析できます。

- UAP が正常終了までに行った処理の流れ
- UAP の異常終了時、障害が発生するまでに UAP が行った処理の流れ

この解析によって、ユーザは UAP に障害が起こった原因を解析でき、UAP を修正したり、システムを再
構築したりする目安にできます。

UAP トレースの詳細については、マニュアル「OpenTP1 テスタ・UAP トレース使用の手引」を参照
してください。

●uid=OpenTP1 管理者のユーザ識別子

～ 〈符号なし整数〉 ((0～4294967294))

このサービスグループのプロセス所有者となるユーザ ID を指定します。

この場合、OpenTP1 管理者のユーザ識別子を指定してください。

最大値は OS に依存するので、使用している OS のマニュアルを参照してください。

●rpc_response_statistics=Y|N

～ 〈N〉

レスポンス統計情報を取得するかどうかを指定します。

Y

レスポンス統計情報を取得します。

N

レスポンス統計情報を取得しません。

レスポンス統計情報は、RPC コールのレスポンスタイム、サービスの実行時間、およびサーバの CPU 時
間などを取得します。

●rpc_trace=Y|N

RPC トレースを取得するかどうかを指定します。

Y

RPC トレースを取得します。

N

RPC トレースを取得しません。

このオペランドは、システム共通定義、またはユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

RPC トレースを取得した場合、処理速度が低下し、RPC がタイムアウトでエラーリターンすることがあります。その場合は、状況に応じて次の 1.または 2.の最大応答待ち時間（デフォルト値：180 秒）を十分な値まで増やして対処してください。

1. watch_time オペランド（システム共通定義、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で指定）に指定した最大応答待ち時間
2. クライアントから rap サーバに引き継いだ最大応答待ち時間

クライアントから rap サーバに最大応答待ち時間を引き継ぐかどうかは、クライアント環境定義の DCWATCHTIMINHERIT オペランド (TP1/Client/W, TP1/Client/P の場合)、または TP1/Client/J 環境定義の dcwatchtiminherit オペランド (TP1/Client/J の場合) で指定します。

DCWATCHTIMINHERIT オペランドまたは dcwatchtiminherit オペランドに Y が指定されている場合は、上記 2.の最大応答待ち時間を使用してください。

DCWATCHTIMINHERIT オペランドまたは dcwatchtiminherit オペランドに N が指定されているか、指定が省略されている場合は、上記 1.の最大応答待ち時間を使用してください。

また、rap リスナーサービス定義の rpc_trace オペランドを指定した場合、rap サーバが代理実行したサービス要求の送受信電文だけを取得でき、rap クライアントから rap サーバが受信した電文は取得できません。

●rpc_trace_name="RPC トレースを取得するファイル名"

～ 〈パス名〉

RPC トレースを取得するファイルのパス名を指定します。

パス名のうち、RPC トレースを取得するファイル名（デフォルトは rpctr）の最大長は、13 文字です。

パス名に環境変数を指定する場合、パス名の先頭に環境変数を指定してください（指定例：\$DCDIR/tmp/ファイル名）。

このオペランドは、システム共通定義でも指定できます。rap リスナーサービス定義で指定した場合は、rap リスナーサービス定義で指定した値が優先されます。

●rpc_trace_size=RPC トレースを取得するファイルのサイズ

～ 〈符号なし整数〉 ((1024～2147483648)) (単位：バイト)

RPC トレースを取得するファイルのサイズを指定します。

このオペランドは、システム共通定義でも指定できます。rap リスナーサービス定義で指定した場合は、rap リスナーサービス定義で指定した値が優先されます。

●trn_expiration_time=トランザクションブランチ限界経過時間

～〈符号なし整数〉((0～65535)) (単位：秒)

トランザクションブランチの処理を監視する限界経過時間を指定します。指定時間を超えた場合は、このトランザクションブランチのプロセスは異常終了して、ロールバックされます。0 を指定した場合は、時間監視しません。

なお、このオペランドは、トランザクションサービス定義またはユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

また、これらの定義のほかに関数を発行して監視時間を指定することもできます。

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

●trn_expiration_time_suspend=Y|N|F

トランザクションブランチの処理を監視するとき、次の処理時間も監視時間を含むかどうかを指定します。

1. 監視対象のトランザクションブランチが、RPC 機能を使ってほかのトランザクションブランチを呼び出し、その処理が終わるのを待つ時間
2. 連鎖 RPC で呼び出されたサーバ UAP が、次のサービス要求を待つ時間
3. 監視対象のトランザクションブランチが、非同期 RPC を使用してほかのトランザクションブランチを呼び出したあと、処理結果受信処理を行っている時間

Y

- 1., 2., 3.のすべてを監視時間を含みます。

N

- 3.だけを監視時間を含みます。

F

- 1., 2., 3.のどれも監視時間を含みません。

このオペランドは、トランザクションサービス定義、またはユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

このオペランドと各種タイマ監視との関係については、「[付録 A.2 トランザクションの時間監視](#)」を参照してください。

●trn_cpu_time=トランザクションブランチ CPU 監視時間

～〈符号なし整数〉((0～65535)) (単位：秒)

トランザクションブランチが同期点処理までに使用できる CPU 時間を指定します。0 を指定した場合は、時間監視しません。指定時間を超えた場合は、該当するトランザクションブランチのプロセスを異常終了し、ロールバックします。

なお、このオペランドは、トランザクションサービス定義またはユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

●trf_put=Y|N

該当するサーバで起動されたトランザクションが出力したジャーナルをトランザクションリカバリジャーナルに出力するかどうかを指定します。

Y

トランザクションリカバリジャーナルファイルに出力します。

N

トランザクションリカバリジャーナルファイルに出力しません。

トランザクションリカバリジャーナルファイルを作成すると、長時間トランザクションによって発生するジャーナル障害を防止でき、またリラン時間を短縮できます。ただし、この機能は、性能、およびメモリへのオーバヘッドがあるので、長時間トランザクションを使用するサーバの場合に指定してください。

系切り替え機能を使用する OpenTP1 システムでは、この機能を使用できないため、rap リスナーサービス定義およびユーザサービス定義の trf_put オペランドには N を指定してください。

●trn_statistics_item=統計情報項目〔、統計情報項目〕…

トランザクションブランチの統計情報を取得する項目を指定します。

nothing

統計情報を取得しません。

base

基本情報として、次の情報を取得します。

- トランザクションブランチの識別子
- トランザクションブランチの決着結果
- トランザクションブランチの実行プロセス種別
- トランザクションブランチの実行サーバ名
- トランザクションブランチの実行サービス名

executiontime

基本情報とトランザクションブランチの実行時間情報を取得します。

cputime

基本情報とトランザクションブランチの CPU 時間情報を取得します。

nothing の指定は、一つしかできません。また、nothing とほかの統計情報項目を同時に指定した場合、nothing の指定は無効になります。

トランザクションに関する統計情報を取得するときは次のどれかを指定してください。

- trn_tran_statistics オペランドに Y を指定
- trnstics コマンドで-s オプションを指定

なお、このオペランドは、トランザクションサービス定義またはユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

- 1.rap リスナーサービス定義
- 2.ユーザサービスデフォルト定義
- 3.トランザクションサービス定義

●trn_optimum_item=トランザクション最適化項目 [, トランザクション最適化項目] …

複数のユーザサーバで構成されるグローバルトランザクションの性能を向上させるための最適化項目を、次の文字列で指定します。

base

同期点処理全体（プリペア処理、コミット処理、およびロールバック処理）を最適化します。OpenTP1 のトランザクション制御は 2 相コミット方式で実行しているため、二つのトランザクションブランチ間のコミット制御には、4 回のプロセス間通信が必要となります。

次の条件をすべて満たす場合、親トランザクションブランチが子トランザクションブランチのコミット処理を代わりに実行することで、コミット制御に必要な 4 回のプロセス間通信を削減します。

1. 親トランザクションブランチと、子トランザクションブランチが同一 OpenTP1 下にあること。
2. 親トランザクションブランチが、子トランザクションブランチを同期応答型 RPC で呼び出していること。
3. 子トランザクションブランチでアクセスしたリソースマネージャの XA インタフェース用オブジェクトが、親トランザクションブランチにもリンケージされていること。

asyncprepare

base の指定条件を満たしていないため同期点処理全体の最適化ができない場合に、プリペア処理を最適化します。

次の条件をすべて満たす場合、親トランザクションブランチから発行された RPC によって子トランザクションブランチがサービス要求を実行したときに、RPC のリターン前にプリペア処理を実行することで、2 回のプロセス間通信を削減します。

1. base を指定した最適化ができないこと。
2. 親トランザクションブランチが、子トランザクションブランチを同期応答型 RPC で呼び出していること。

ただし、この最適化を実行した場合、親トランザクションブランチが発行した同期応答型 RPC の応答時間が遅くなります。また、子トランザクションブランチは、プリペア処理からコミット処理までの間隔（親トランザクションブランチからの指示がないとトランザクションを決着できない状態）が大きくなります。そのため、親トランザクションブランチの OpenTP1 がシステムダウンし、トランザクションブランチ間の連絡ができなくなると、ジャーナルファイルのスワップやチェックポイントダンプファイルの有効化が遅れ、子トランザクションブランチの OpenTP1 もシステムダウンする場合があります。

トランザクション最適化項目は、重複して指定できます。ただし、優先順位は次のようになります（1.> 2.）。

1. base
2. asyncprepare

なお、このオペランドは、トランザクションサービス定義またはユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです（1.> 2.> 3.）。

1. rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

●trn_watch_time=トランザクション同期点処理時の最大通信待ち時間

～ 〈符号なし整数〉 ((1～65535)) (単位：秒)

トランザクションの同期点処理で、トランザクションブランチ間で行う通信（プリペア、コミット、ロールバック指示、または応答など）の受信待ち時間の最大値を指定します。

指定時間を過ぎても指示または応答がない場合は、該当するトランザクションブランチが2相コミットの1相目完了前であればロールバックさせ、1相目完了後であればトランザクションサービスのシステムプロセスでトランザクション決着処理を再試行します。

このオペランドは、トランザクションサービス定義またはユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです（1.> 2.> 3.）。

1. rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

これらのすべての定義でこのオペランドを省略した場合、トランザクションを行ったユーザサーバの watch_time の値を仮定します。ただし、trn_watch_time には0秒を指定できないため、ユーザサーバの watch_time に0秒を指定した場合は trn_watch_time のデフォルト値である120秒を仮定します。

●trn_rollback_information_put=no|self|remote|all

トランザクションブランチがロールバックした場合に、ロールバック要因に関する情報をログに取得するかどうかを指定します。

no

ロールバック情報を取得しません。

self

ロールバック要因が発生したトランザクションブランチでだけ、ログにロールバック情報を取得します。

remote

self に加え、他ノードのトランザクションブランチからロールバック要求されたトランザクションブランチでも、ログにロールバック情報を取得します。

all

remote に加え、自ノードのトランザクションブランチからロールバック要求されたトランザクションブランチでも、ログにロールバック情報を取得します。

このオペランドは、トランザクション定義またはユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです（1.> 2.> 3.）。

1. rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

●trn_limit_time=トランザクションブランチ最大実行可能時間

～〈符号なし整数〉((0～65535)) (単位：秒)

トランザクションブランチの最大実行可能時間を指定します。

トランザクションブランチを開始してから同期点処理が終了するまでの時間が、このオペランド指定時間を超えないように dc_rpc_call 関数、dc_rpc_poll_any_replies 関数および同期点処理内で行う通信のタイムアウト時間を次のように自動設定します。

- dc_rpc_call 関数および dc_rpc_poll_any_replies 関数のタイムアウト時間

「 $K \geq$ このオペランド指定時間」の場合は、要求処理を実行しないで、タイムアウトでエラーリターンします。

「 $K <$ このオペランド指定時間」でかつ「 $(\text{このオペランド指定時間} - K) \geq W$ 」の場合は、 W をタイムアウト時間とします。

「 $K <$ このオペランド指定時間」でかつ「 $(\text{このオペランド指定時間} - K) < W$ 」の場合は、 $(\text{このオペランド指定時間} - K)$ をタイムアウト時間とします。

K

現時刻－トランザクションブランチ開始時刻

W

dc_rpc_call 関数の場合は watch_time オペランド指定時間

dc_rpc_poll_any_replies 関数の場合は引数 timeout 指定時間

- 同期点処理内で行う通信のタイムアウト時間

「 $K \geq$ このオペランド指定時間」の場合は、タイムアウト時間を 1 秒とします。

「 $K <$ このオペランド指定時間」でかつ「 $(\text{このオペランド指定時間} - K) \geq W$ 」の場合は、 W をタイムアウト時間とします。

「 $K <$ このオペランド指定時間」でかつ「 $(\text{このオペランド指定時間} - K) < W$ 」の場合は、 $(\text{このオペランド指定時間} - K)$ をタイムアウト時間とします。

K

現時刻－トランザクションブランチ開始時刻

W

trn_watch_time オペランド指定時間

trn_watch_time オペランドを省略した場合は watch_time オペランド指定時間

上記の受信待ち以外の処理で時間が掛かった場合は、このオペランド指定時間以内にトランザクションブランチが終了しないことがあります。

同期点処理開始前にこのオペランドの指定時間が経過した場合、そのトランザクションはロールバックします。

0 を指定した場合は、時間監視をしません。

なお、このオペランドは、トランザクションサービス定義またはユーザサービスデフォルト定義でも指定できます。指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

●trn_rollback_response_receive=Y|N

～《Y》

RPC 先トランザクションブランチにロールバック指示を送信したあと、**ロールバック完了通知を受信するかどうか**を指定します。N を指定した場合、RPC 先トランザクションブランチからのロールバック完了通知を受信しないで (RPC 先トランザクションブランチのロールバック処理の完了を待たないで) 自トランザクションブランチを終了します。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●trn_partial_recovery_type=type1|type2|type3

UAP 障害時のトランザクション同期点処理方式を指定します。

RPC がタイムアウトし、RPC 発行先プロセスのアドレスが未解決の場合やトランザクション実行中の UAP がダウンした場合に、トランザクションブランチ間の連絡がスムーズにできないで、トランザクションの決着に時間が掛かることがあります。

このオペランドでは、次に示す障害が発生した場合のトランザクション同期点処理方式を、指定値に示す三つの方式から選択して指定します。

(障害 1) RPC がタイムアウトした場合

この場合、RPC 発行元トランザクションブランチは、サービス要求がどのプロセスで実行されているかがわからないため、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信できません。そのため、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチがトランザクション同期点メッセージ待ちとなり、トランザクションの決着に時間が掛かります。

(障害 2) RPC 発行元 UAP が RPC の応答受信前にダウンした場合

この場合、RPC 発行元トランザクションブランチは、サービス要求がどのプロセスで実行されているかがわからないため、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信できません。そのため、RPC 発行先トランザクションブランチはトランザクション同期点メッセージ待ちとなり、トランザクションの決着に時間が掛かります。

(障害 3) RPC 発行先 UAP からの応答受信後に RPC 発行元 UAP と RPC 発行先 UAP がほぼ同時にダウンした場合

この場合、それぞれのトランザクションブランチを引き継いだトランザクション回復プロセスは、相手 UAP プロセスのダウンを知らないため、すでに存在しない UAP プロセスにトランザクション同期点メッセージを送信してしまい、トランザクションの決着に時間が掛かることがあります。

type1

(障害 1) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

(障害 2) が発生した場合、RPC 発行元トランザクションブランチは、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信しないでトランザクションを決着します。RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

(障害 3) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

type2

(障害 1) が発生してトランザクションをコミットする場合は type1 と同じです。

(障害 1) が発生してトランザクションをロールバックする場合、または (障害 2) が発生した場合は、RPC 発行元トランザクションブランチは、RPC 発行先トランザクションブランチが存在するノードのトランザクションサービスプロセスにトランザクション同期点メッセージを送信後、トランザクションを決着します。トランザクション同期点メッセージを受信したトランザクションサービスプロセスは、該当するトランザクションブランチを処理中のプロセスに、トランザクション同期点指示を送信します。

(障害 3) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

type3

(障害 1) が発生してトランザクションをコミットする場合は、type1 と同じです。

(障害 1) が発生してトランザクションをロールバックする場合、(障害 2) が発生した場合、または (障害 3) が発生した場合、相手トランザクションブランチが存在するノードのトランザクションサービスプロセスに、トランザクション同期点メッセージを送信します。トランザクション同期点メッセージを受信したトランザクションサービスプロセスは、該当するトランザクションブランチを処理中のプロセスに、トランザクション同期点指示を送信します。

次に示す場合、このオペランドに type2 または type3 を指定しても、トランザクションの決着に時間が掛かることがあります。

1. RPC 実行中に、RPC 発行先 UAP の状態が変更となり(負荷増加, UAP 終了, UAP 閉塞など)、ほかのノードの同一 UAP にサービス要求が再転送された場合
2. 相手先の OpenTP1 がこのオプションをサポートしていないバージョンの場合

3. 相手先トランザクションブランチがトランザクション同期点メッセージ受信処理以外で時間が掛かっている場合

なお、このオペランドは、トランザクションサービス定義またはユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. rap リスナーサービス定義
2. ユーザサービスデフォルト定義
3. トランザクションサービス定義

●rap_inquire_timeout_message=Y|N

～《Y》

クライアントからの要求最大待ち時間が満了したことを検知したとき、エラーメッセージを出力するかどうかを指定します。このオペランドに Y を指定するか、省略した場合は、エラーメッセージを出力します。

Y

クライアントからの要求最大待ち時間が満了したことを検知したとき、エラーメッセージを出力します。

N

クライアントからの要求最大待ち時間が満了したことを検知したとき、エラーメッセージを出力しません。

なお、このオペランドで抑止されるメッセージは次のとおりです。

- KFCA26956-W メッセージの理由コード=71
- KFCA26965-E メッセージの理由コード=71

●rap_connection_assign_type=dynamic|static

～《static》

ダイナミックコネクションスケジュール機能を使用するかどうかを指定します。

dynamic

ダイナミックコネクションスケジュール機能を使用します。

static

ダイナミックコネクションスケジュール機能を使用しません。

ダイナミックコネクションスケジュール機能を使用しサーバ負荷が高くなると、スタティックコネクションスケジュール機能に比べてレスポンス性能が劣化する場合があります。また、ダイナミックコネクションスケジュール機能を使用している場合、コネクト要求以外の要求は rap サーバで処理します(コネクト要求だけ rap リスナーが受け持ちます)。したがって、クライアントが要求を行ったとき割り当てられる rap

サーバがない場合、その要求は rap リスナーで保留されたままになります。rap サーバから代理実行要求される dc_rpc_call に長い時間が掛かる場合や、トランザクション状態で走行するクライアントがある場合などは、起動する rap サーバ数に注意してください。起動する rap サーバが少ない場合、クライアントからの要求はタイムアウトでエラーリターンすることがあります。

●rap_max_client=rap リスナーに同時接続する最大クライアント数

～ 〈符号なし整数〉 ((128～1024)) 《256》

rap リスナーに同時接続する最大クライアント数を指定します。

●rap_notify=Y|N

～ 《N》

自ノードの rap リスナー起動時に、rap_client_manager_node オペランドで指定したノードで起動する rap クライアントマネージャに対し起動通知を行うかどうかを指定します。

Y

起動通知します。

N

起動通知しません。

Y を指定した場合、rap_client_manager_node オペランドの指定を省略できません。

●rap_client_manager_node="ホスト名:ポート番号" ["ホスト名:ポート番号"] …

rap_notify オペランドに Y を指定して起動通知機能を使用する場合は、このオペランドに指定したノードで起動する OpenTP1 の rap クライアントマネージャへ起動通知を行います。起動通知機能を使用すると、コネクション確立時期を管理でき、確実に送信できるようになります。

rap_notify オペランドに Y を指定した場合、このオペランドの指定を省略できません。

ホスト名

～ 〈1～255 文字の英数字、ピリオド、およびハイフン〉

自ノードの rap リスナーに接続する rap クライアントを持つ、OpenTP1 システムのホスト名を指定します。ホスト名とは、/etc/hosts に定義したホスト名のことです。

ポート番号

～ 〈符号なし整数〉 ((1～65535))

自ノードの rap リスナーに接続する rap クライアントを持つ、OpenTP1 システムの rap クライアントマネージャのポート番号を指定します。

●rap_max_buff_size=ソケットウィンドウサイズ

～ 〈符号なし整数〉 ((4～2147483647)) 《4》 (単位：キロバイト)

ソケットのウィンドウサイズを指定します。

●**rap_io_retry_interval=電文送受信時のリトライ間隔**

～ 〈符号なし整数〉 ((1～999)) 《35》 (単位：ミリ秒)

電文送受信時のリトライ間隔を指定します。

●**rap_sock_count=ソケット生成処理のリトライ回数**

～ 〈符号なし整数〉 ((0～65535)) 《1》

ソケット生成処理で EAGAIN, EWOULDBLOCK エラーが発生した場合、ソケット生成処理のリトライ回数を指定します。

●**rap_sock_interval=ソケット生成処理のリトライ間隔**

～ 〈符号なし整数〉 ((15～500)) 《30》 (単位：ミリ秒)

ソケット生成処理で EAGAIN, EWOULDBLOCK エラーが発生した場合、ソケット生成処理のリトライ間隔を指定します。

●**rap_connect_retry_count=コネクション確立処理のリトライ回数**

～ 〈符号なし整数〉 ((8～2147483647)) 《8》

通信制御部のコネクション確立処理で、ECONNREFUSED エラーが発生した場合、コネクション確立処理のリトライ回数を指定します。

●**rap_connect_retry_interval=コネクション確立処理のリトライ間隔**

～ 〈符号なし整数〉 ((10～999)) 《100》 (単位：ミリ秒)

通信制御部のコネクション確立処理で、ECONNREFUSED エラーが発生した場合、コネクション確立処理のリトライ間隔を指定します。

●**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〉 ((SOMAXCONN～2147483647)) 《SOMAXCONN》

接続待ちキューに保留できる最大要求数を指定します。

ここで最小値およびデフォルト値にしている SOMAXCONN は OpenTP1 をコンパイルした環境の値です。詳細については、「リリースノート」を参照してください。

なお、このオペランドの設定が有効になるのは次の場合だけです。

OpenTP1をコンパイルした環境のSOMAXCONN ≤ 実際の動作環境のSOMAXCONN

このオペランドの設定は、rap リスナーが発行する listen() システムコールのバックログ数に設定するだけで、バックログの実際の制御は OS にゆだねられます。

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。

●rap_msg_output_interval=クライアント数表示メッセージの出力間隔

～〈符号なし整数〉((0～32767))《0》(単位：分)

rap リスナー終了時に接続中のクライアントがあった場合に出力される、クライアント数表示メッセージの出力間隔を指定します。クライアント数表示メッセージの出力後、このオペランドで指定した時間が経過してもまだ接続中のクライアントがあった場合は、再びクライアント数表示メッセージが出力されます。

このオペランドに 0 を指定した場合は、クライアント数表示メッセージを一度だけ出力します。

また、rap リスナーは 3 秒ごとにイベントを監視しています。

●rap_recovery_server=リカバリ要求用待機 rap サーバ数

～〈符号なし整数〉((0～rap_parallel_server オペランドの指定値-1))《0》

XA リソースサービスを使用する場合、クライアントからのリカバリ要求用待機 rap サーバ数を指定します。このオペランドに指定した数の rap サーバは、通常のサービス要求を受け付けずに、リカバリ要求だけを実行するために待機します。リカバリ要求用待機 rap サーバの必要数は、J2EE サーバ単位に 1 サーバを目安としてください。

●rap_connect_interval=コネクション確立処理の間隔

～〈符号なし整数〉((0～999))《40》(単位：ミリ秒)

リモート API サービス開始時の rap リスナーと rap サーバとの間のコネクション確立処理で、コネクション確立処理の間隔を指定します。

コネクション確立処理は、接続待ちキューに保留できる最大要求数ごとに、このオペランドで指定した値の間隔で行います。

●rpc_extend_function=RPC サービスの機能拡張レベル

～〈16 進数字〉((00000000～0000000F))《00000000》

RPC サービスの機能の拡張レベルを、次の中から指定します。

次の指定値で示す機能の拡張レベルを複数指定する場合、それぞれの指定値の論理和を指定してください。

00000000

RPC サービスの機能を拡張しません。

00000001

サービス要求実行中の SPP が異常終了した場合に、dc_rpc_call 関数、dc_rpc_call_to 関数および dc_rpc_poll_any_replies 関数で、リターン値「DCRPCER_SERVICE_TERMINATED (00378)」を返します。

00000002

トランザクション内で開始した非トランザクション（flags に DCRPC_TPNOTRAN を指定した dc_rpc_call 関数、および dc_rpc_call_to 関数）の連鎖 RPC を、同期点処理で終了しないで、flags に DCNOFLAGS を指定した dc_rpc_call 関数、および dc_rpc_call_to 関数で明示的に終了するまで継続します。

00000004

非同期応答型 RPC の応答メッセージを受け取っていない状態で、トランザクションの同期点処理をした場合に、非トランザクション属性の非同期応答型 RPC の応答メッセージは破棄しないで、トランザクション属性の非同期応答型 RPC の応答メッセージだけを削除します。

00000008

SPP プロセスで、dc_rpc_call 関数、および dc_rpc_call_to 関数の呼び出し元の応答受信タイムアウトを検出したため、処理を中断してサービス要求を破棄した場合に、KFCA00339-W のメッセージを出力します。

●max_socket_descriptors=ソケット用ファイル記述子の最大数

～〈符号なし整数〉((32～2032))

OpenTP1 制御下のプロセス※で、ソケット用に使用するファイル記述子の最大値を指定します。

OpenTP1 制御下のプロセス※では、システムサーバやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数および通信する他ノードの数によって、ソケット用のファイル記述子の最大数を変更する必要があります。

注※

MCF サービス（MCF マネジャサービス、MCF 通信サービス、およびアプリケーション起動サービス）以外の OpenTP1 プロセスが対象です。MCF サービスの場合は、「[システムサービス情報定義](#)」および「[システムサービス共通情報定義](#)」を参照してください。

ソケット用ファイル記述子の最大数の計算式を、次に示します。

$\uparrow (\text{このユーザサーバが通信する UAP プロセス数}^{\ast 1} + \text{システムサービスプロセス数}^{\ast 2}) / 0.8 \uparrow$

$\uparrow \uparrow$: 小数点以下を切り上げます。

注※1

rap リスナーは、rap リスナーが管理する rap サーバと通信します。rap リスナーが通信する rap サーバ数と rap サーバが通信する UAP プロセス数の、どちらか大きい値を採用してください。

rap サーバが通信する UAP プロセス数は、次に示す値の合計です。

- rap サーバが通信する自 OpenTP1 内の UAP プロセス数
- rap サーバが通信する他ノード内の UAP プロセス数

注※2

システムサービスプロセス数とは、自 OpenTP1 内のシステムサービスプロセスの数です。自 OpenTP1 内のシステムサービスプロセスは、rpcstat コマンドで表示されるサーバ名をカウントすることで求められます。rpcstat コマンドで表示されるサーバ名のうち、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造に記載されているシステムサービスプロセスをカウントしてください。

このオペランドの指定値が小さいと、OpenTP1 制御下の他プロセスとの接続が設定できなくなるため、プロセスが KFCA00307-E メッセージを出力して異常終了します。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

- 1.rap リスナーサービス定義
- 2.ユーザサービスデフォルト定義
- 3.システム共通定義

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●trn_completion_limit_time=トランザクション完了限界時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

トランザクションブランチの開始から終了までの最大実行時間を指定します。指定時間を超えた場合、このトランザクションブランチのプロセスが異常終了したあとに、トランザクションブランチが回復プロセスによってコミットまたはロールバックのどちらかに決着して終了します。0 を指定した場合は、トランザクションブランチの最大実行時間を監視しません。

このオペランドの監視対象区間は、rap サーバが dc_trn_begin 関数などの API を代理実行することによるトランザクションの開始から、トランザクションの同期点処理終了情報 (TJ) の取得後であるトランザクションブランチの終了までです。このオペランドの監視対象区間の詳細および各種タイマ監視との関係については、「付録 A.2 トランザクションの時間監視」を参照してください。

なお、このオペランドは、トランザクションサービス定義またはユーザサービスデフォルト定義でも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

- 1.rap リスナーサービス定義
- 2.ユーザサービスデフォルト定義
- 3.トランザクションサービス定義

●rap_message_id_change_level=メッセージ ID の変更レベル

～ 〈符号なし整数〉 ((0～2))

リモート API 機能を使用する場合に出力される可能性の高いエラーメッセージでは、メッセージの種類の E を W に変更するために、メッセージ ID を変更できます。変更する内容によって変更レベルを指定してください。

メッセージログを監視し、メッセージの種類が E であるか W であるかによって管理方法を区別したい場合にこのオペランドを使用してください。

このオペランドによって変更されるのはメッセージ ID だけであり、メッセージの内容は変更されません。

0
メッセージ ID を変更しないで、従来どおりのメッセージ ID で出力します。

1
特定の理由コードが出力される条件で、メッセージの種類 E を W に変更するために、メッセージ ID を変更できます。

2
1 を指定した場合に加え、メッセージの種類 E を W に変更するために、メッセージ ID を変更できます。

このオペランドの指定値と出力されるメッセージの関係を次の表に示します。

表 3-12 rap_message_id_change_level オペランドの指定値と出力されるメッセージの関係

変更できるメッセージ ID		KFCA26965-E を KFCA27790-W に変更			KFCA26970-E を KFCA27791-W に変更			KFCA26971-E を KFCA27792-W に変更		
rap_message_id_change_level オペ ランドの指定値		0	1	2	0	1	2	0	1	2
理由コード	8	—	—	—	E	W	W	—	—	—
	22	E	E	E	E	E	E	E	E	E
	24	—	—	—	E	E	E	—	—	—
	31	E	E	W	E	E	W	E	E	W
	32	E	E	W	E	E	W	E	E	W
	35	E	E	W	—	—	—	E	E	E
	36	E	W	W	E	W	W	E	W	W
	37	E	E	E	E	E	E	E	E	E
	38	—	—	—	E	W	W	—	—	—
	71	E	W	W	—	—	—	—	—	—
	81	—	—	—	E	E	E	—	—	—
	82	—	—	—	E	W	W	—	—	—
	83	—	—	—	E	E	E	—	—	—

変更できるメッセージ ID		KFCA26965-E を KFCA27790-W に変更			KFCA26970-E を KFCA27791-W に変更			KFCA26971-E を KFCA27792-W に変更		
rap_message_id_change_level オペラ ンドの指定値		0	1	2	0	1	2	0	1	2
理由コード	91	E	W	W	—	—	—	—	—	—

(凡例)

- E：メッセージの種類が E のメッセージが出力されます。
- W：メッセージの種類が W のメッセージが出力されます。
- ：該当しません。

このオペランドの指定を省略した場合は、ユーザサービスデフォルト定義の値が仮定されます。

●rap_term_disconnect_time=rap リスナー終了時のコネクション切断待ち時間

～〈符号なし整数〉((0～3600))《0》(単位：秒)

rap リスナー終了時の rap クライアントとのコネクション切断待ち時間を指定します。rap リスナー終了時、このオペランドに指定した時間が経過したら、コネクションを切断して KFCA27763-W または FCA27765-W メッセージを出力します。ただし、rap サーバが API を代理実行中は、コネクションを切断しません。このオペランドを省略または 0 を指定した場合、rap クライアントからコネクション切断要求が到着するか、または問い合わせ間隔最大待ち時間が満了するまで、コネクションを切断しません。

このオペランドには、システム環境定義の system_terminate_watch_time オペランドの指定値より小さい値を指定することをお勧めします。次の場合、rap リスナーの終了待ちが原因で、dcstop コマンドがタイムアウトすることがあります。

- ・このオペランドに system_terminate_watch_time オペランドの指定値より大きい値を指定した場合
- ・このオペランドに 0 を指定した場合
- ・このオペランドを省略した場合

なお、rap リスナーは 3 秒ごとにイベントを監視しているため、この待ち時間には最大で 3 秒の誤差が発生することがあります。

●rap_stay_watch_time=rap サーバ割り当て待ち要求の滞留監視時間

～〈符号なし整数〉((0～65535))《30》(単位：秒)

rap クライアントの要求が滞留した場合の監視時間を指定します。rap クライアントの要求を実行するための空き rap サーバがないとき、要求は rap サーバが空くのを待ちます。この割り当て待ち要求の滞留時間がこのオペランド指定値を超えた場合、KFCA27764-W メッセージを出力します。このオペランドに 0 を指定した場合は、滞留時間を監視しません。

このオペランドには、rap クライアント側で指定する最大応答待ち時間より小さい値を指定することをお勧めします。このオペランドに rap クライアント側の最大応答待ち時間より大きい値を指定した場合、ク

クライアント側がすでにタイムアウトと判断した要求に対しても、KFCA27764-W メッセージを出力することがあります。

なお、rap リスナーは 3 秒ごとにイベントを監視しているため、この監視時間には最大で 3 秒の誤差が発生することがあります。

●rap_stay_warning_interval=滞留警告メッセージの出力間隔

～ 〈符号なし整数〉 ((3～65535)) 《180》 (単位：秒)

rap クライアントの要求が滞留した場合に、KFCA27764-W メッセージが出力される間隔を指定します。KFCA27764-W メッセージが一度出力されたあと、このオペランドの指定値を経過するまでの間、KFCA27764-W メッセージは出力されません。rap_stay_watch_time オペランドに 0 を指定した場合、このオペランドは無視されます。

なお、rap リスナーは 3 秒ごとにイベントを監視しているため、この出力間隔には最大で 3 秒の誤差が発生することがあります。

●log_audit_out_suppress=Y|N

～ 《N》

rap リスナーおよび rap サーバから出力される監査ログを抑止する場合に指定します。

Y

rap リスナーおよび rap サーバから出力される監査ログを抑止します。

N

rap リスナーおよび rap サーバから出力される監査ログを抑止しません。

このオペランドは、ログサービス定義の log_audit_out オペランドに Y を指定した場合だけ有効です。

●log_audit_message=監査ログを取得する項目のメッセージ ID [監査ログを取得する項目のメッセージ ID] ...

～ 〈符号なし整数〉 ((33400～99999))

OpenTP1 が取得する監査ログのうち、rap リスナーサービス定義に指定できる監査ログのメッセージ ID を指定します。指定できるメッセージ ID は、最大 2048 個です。

このオペランドで指定できるメッセージ ID については、「[付録 C 監査イベントを取得する定義](#)」を参照してください。

このオペランドを省略し、ユーザサービスデフォルト定義でも指定を省略した場合、ログサービス定義の値を仮定します。このオペランドは、ログサービス定義の log_audit_out オペランドに Y を指定し、rap リスナーサービス定義の log_audit_out_suppress オペランドに N を指定した場合に有効です。

●`ipc_socketctl_highwater`=ソケットの一時クローズ処理開始数パーセンテージ〔ソケットの一時クローズ処理非対象数パーセンテージ〕

～〈符号なし整数〉((0～100))

`max_socket_descriptors` オペランドの指定値に対して、ソケットの一時クローズ処理開始数パーセンテージを指定します。

OpenTP1 は、プロセス内のソケット用に使用しているファイル記述子の数が、次の値を超えた時点で、一時クローズ処理を開始します。

`max_socket_descriptors`オペランドの指定値
× (ソケットの一時クローズ開始数パーセンテージ／100)

ソケットの一時クローズ処理開始数パーセンテージに 0 を指定した場合、コネクションを確立するたびに一時クローズ処理が実行されます。一時クローズ処理については、マニュアル「OpenTP1 解説」を参照してください。

また、一時クローズ処理の対象外とするコネクション数のパーセンテージも指定できます。一時クローズ処理の対象外とするコネクション数は、次の値です。

`max_socket_descriptors`オペランドの指定値
× (ソケットの一時クローズ非対象数パーセンテージ／100)

ソケットの一時クローズ非対象数パーセンテージには、ソケットの一時クローズ処理開始数パーセンテージ以下の値を指定してください。ソケットの一時クローズ処理開始数パーセンテージより大きい値を指定した場合は、ソケットの一時クローズ処理開始数パーセンテージと同じ値が指定されたものとして動作します。

OpenTP1 は、プロセス内で確立したコネクションを、確立した順に管理しています。ソケットの一時クローズ処理非対象数パーセンテージを指定した場合、最も古く確立されたコネクションから順に、一時クローズ処理要求が送信されます。

`max_socket_descriptors` オペランドの指定値が大きくなり、かつソケットの一時クローズ処理開始数パーセンテージに小さい値が指定されていると、一時クローズ要求が多発し、性能に影響を与えたり、通信障害になったりすることがあります。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●`ipc_socketctl_watchtime`=ソケット再利用可能監視時間

～〈符号なし整数〉((0～65535)) (単位：秒)

プロセス内のソケット用に使用しているファイル記述子の数が、`max_socket_descriptors` オペランドの指定値になった時点から、一時クローズ処理によってソケットが再利用できるようになるまでの監視時間(秒)を指定します。

一時クローズ処理は、コネクションを確立したプロセス間の合意によってコネクションを切断するため、一時クローズ処理の要求を送信したプロセスはその応答を受けるまではコネクションを切断できません。応答を受信したあと、コネクションは切断され、ソケットが再利用できるようになります。

ipc_sockctl_watchtime オペランドの指定値を経過しても、どのプロセスからも一時クローズ処理要求に対する応答が返らない場合、プロセスは強制停止されます。ipc_sockctl_watchtime オペランドに 0 を指定した場合、無限に待ちます。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●scs_prf_trace_level=リモート API 機能における性能検証用トレース取得レベル

～〈符号なし整数〉((00000000～00000009))《00000001》

リモート API 機能における性能検証用トレースの取得レベルを指定します。指定した値が次に示すビットを含む場合、情報を取得します。性能検証用トレースの取得レベルを複数指定する場合、それぞれの指定値の論理和を指定してください。

00000000

リモート API 機能における性能検証用トレースを取得しません。

00000001

イベント ID が 0x5001～0x5008 の性能検証用トレースを取得します。

00000008

イベント ID が 0x5200, または 0x5201 の性能検証用トレースを取得します。

イベント ID の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

●rap_extend_function=リモート API 機能の機能拡張レベル

～〈16 進数字〉((00000000～00000001))《00000000》

リモート API 機能の機能拡張レベルを指定します。機能の拡張レベルを複数指定する場合、それぞれの指定値の論理和を指定してください。

00000000

リモート API 機能を機能拡張しません。

00000001

rap クライアントと rap サーバ間の dc_rpc_call 関数の処理でタイムアウト事象が発生した場合の dc_rpc_call 関数のリターン値を、DCRPCER_NET_DOWN (-306) から DCRPCER_TIMED_OUT (-307) に変更します。

リターン値が変更となるのは次の場合です。

- rap クライアントに指定された watch_time オペランドの指定時間内に、サービス要求先サーバからの応答が返らなかった場合
- rap クライアントからトランザクショナル RPC を実行した時点で rap クライアントに指定された trn_limit_time オペランドの指定値を満たした場合

このオペランドの指定は、rap サーバが rap クライアントとして動作した場合に有効となります。SUP、SPP または MHP が rap クライアントとなる場合は、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指定してください。

●watch_time=最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

rap サーバが RPC を代理実行する場合、サービス要求を送信してからサービスの応答が返るまでの最大待ち時間を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かるときがあります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。0 を指定した場合は、応答を受信するまで待ち続けます。また、0 を指定した場合、OpenTP1 が終了しないときがあります。

このオペランドは、システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

このオペランドを省略し、ユーザサービスデフォルト定義でも指定を省略した場合、システム共通定義の値を仮定します。ただし、rap クライアント側で最大応答待ち時間を引き継ぐ設定にした場合は、rap クライアント側で設定した値が有効になります。

コマンド形式

なし。

注意事項

- rap リスナーサービス定義は、rapdfgen コマンドの入力となる定義ファイルであり、\$DCCONFPATH の下に置いた場合、新たに作成された rap リスナー用ユーザサービス定義に上書きされるので、\$DCCONFPATH の下には置かないでください。\$DCCONFPATH の下に置いた場合の動作は保証しません。
- rap リスナー用ユーザサービス定義および rap サーバ用ユーザサービス定義は、\$DCCONFPATH の下にあることを前提としているので、\$DCCONFPATH の下から\$DCUAPCONFPATH の下に移動しないでください。\$DCUAPCONFPATH の下に移動した場合の動作は保証しません。

- リモート API 機能を使用する場合、次に示すシステムサービス定義の指定値に注意してください。
 - システム環境定義

server_count オペランドに rap リスナー数と rap サーバ数を加算してください。また、system_terminate_watch_time オペランドの指定値は rap サービスの終了時間を考慮して見直してください。
 - ユーザサービス構成定義

TP1/Server Base と同期して起動する rap リスナーサービスは、dcsvstart 定義コマンドで定義してください。この場合のユーザサーバ名は、rap リスナーサービス名を指定してください。
 - システム共通定義

all_node オペランドで指定するノード名に、クライアントホスト名を指定する必要はありません。また、name_port オペランドおよび prc_port オペランドで指定するポート番号と rap リスナーが使用するポート番号は重複しないようにしてください。重複した場合の動作は保証できません。
 - プロセスサービス定義

prc_process_count オペランドに、rap サーバのプロセス数を考慮して見直してください。
 - スケジュールサービス定義

scd_server_count オペランドに、rap サーバ数を考慮して見直してください。
- 次に示すオペランドは、rap リスナーサービス定義で指定を省略した場合、ユーザサービスデフォルト定義の値が有効になります。

rpc_trace
 rpc_trace_name
 rpc_trace_size
 rpc_extend_function
 trn_expiration_time
 trn_expiration_time_suspend
 trn_cpu_time
 trn_statistics_item
 trn_optimum_item
 trn_rollback_information_put
 trn_watch_time
 trn_limit_time
 trn_rollback_response_receive
 trn_partial_recovery_type
 trn_completion_limit_time
 max_socket_descriptors
 log_audit_message
 watch_time

- 次に示すオペランドは、rap リスナーサービス定義に指定しても無効となります。

- rap リスナーの場合

module
type
atomic_update
receive_from
auto_restart
critical
trf_put
node_down_restart
term_watch_time
max_open_fds
rpc_destination_mode
status_change_when_termining

- rap サーバの場合

module
atomic_update
type
hold
hold_recovery
server_security
service
balance_count
auto_restart
critical
service_hold
service_priority_control
node_down_restart
server_type
term_watch_time
max_open_fds
message_store_bufLen
schedule_delay_limit
schedule_delay_abort
scd_pool_warning_use_rate
scd_pool_warning_interval

rap リスナーおよび rap サーバは、これら以外のオペランドに関して、ユーザサービスデフォルト定義のデフォルト値が指定されているものとして動作します。

- OpenTP1 の制限のため、次に示す場合は、各オペランドの指定値を調整してください。
 - rap_connection_assign_type オペランドで dynamic を指定する場合
rap_parallel_server オペランド、rap_max_client オペランドおよび max_socket_descriptors オペランドの和が 1993 以下になるように調整してください。
 - rap_connection_assign_type オペランドで static を指定する場合
rap_max_client オペランドおよび max_socket_descriptors オペランドの和が 1993 以下になるように調整してください。
- TP1/Client/J を使用する場合は、rpc_extend_function オペランドの 00000002 ビットをオンにしないでください。00000002 ビットがオンになっている場合の動作は保証できません。
- rap リスナーサービス定義に、ユーザサービス定義の rpc_rap_auto_connect オペランドを指定しないでください。rap サーバは rpc_rap_auto_connect オペランドに Y が指定されたものとして動作していますが、rpc_rap_auto_connect オペランドに N が指定された場合の rap サーバの動作は保証できません。
- rap リスナーの開始処理中に、1 プロセス当たりでオープンできるファイル数を超えると、rap リスナーが KFCA00105-E（アポートコードは r902109）のメッセージを出力して異常終了します。1 プロセス当たりでオープンできるファイル数を超えないように、max_open_fds オペランドおよび max_socket_descriptors オペランドの値を見積もってください。

OpenTP1 のプロセスでオープンするファイル数の最大値は、ユーザサービス定義の max_open_fds オペランドと max_socket_descriptors オペランドの指定値の合計になります。なお、max_socket_descriptors オペランドの値は、指定値がそのまま適用されますが、max_open_fds オペランドの値は、rapdfgen コマンドで定義を生成したときに、次の計算式によって値が自動的に設定されます。

- rap リスナー用ユーザサービス定義
$$\text{max\_open\_fds} = \text{rap\_parallel\_server} + \text{rap\_max\_client} + 23$$
- rap サーバ用ユーザサービス定義
$$\text{max\_open\_fds} = 22$$

1 プロセス当たりでオープンできるファイル数については、「リリースノート」または使用している OS のマニュアルを参照してください。

rap クライアントマネージャサービス定義

形式

set 形式

```
set rap_client_manager_port=rapクライアントマネージャのポート番号
set rap_listen_inf="ノード識別子:ポート番号=ホスト名:ポート番号"
                  [, "ノード識別子:ポート番号=ホスト名:ポート番号"] ...
set uid=openTP1管理者のユーザ識別子
[set log_audit_out_suppress=Y|N]
[set log_audit_message=監査ログを取得する項目のメッセージID
                      [, 監査ログを取得する項目のメッセージID] ...]
[set rap_watch_time=メッセージ送受信最大監視時間]
```

コマンド形式

なし。

機能

リモート API クライアントマネージャ機能を使用するための実行環境を定義します。

説明

set 形式のオペランド

●rap_client_manager_port=rap クライアントマネージャのポート番号

～ 〈符号なし整数〉 ((5001～65535))

rap リスナーの起動を監視する rap クライアントマネージャのポート番号を指定します。

このオペランドで指定するポート番号は、ほかのプログラムと重複しないようにしてください。

また、OS には任意に割り当てる番号がありますが、この番号も使用しないでください。OS が任意に割り当てる番号は、OS の種別やバージョンによって異なります。詳細については、使用している OS のマニュアルを参照してください。

●rap_listen_inf="ノード識別子:ポート番号=ホスト名:ポート番号" [, "ノード識別子:ポート番号=ホスト名:ポート番号"] ...

監視する rap リスナーを指定します。コンマで区切り、最大 1024 の rap リスナーを指定できます。

"ノード識別子:ポート番号 1=ホスト名:ポート番号 2"

●uid=openTP1 管理者のユーザ識別子

～ 〈符号なし整数〉 ((0～4294967294))

このサービスグループのプロセス所有者となるユーザ ID を指定します。

この場合、OpenTP1 管理者のユーザ識別子を指定してください。

最大値は OS に依存するので、使用している OS のマニュアルを参照してください。

●log_audit_out_suppress=Y|N

～《N》

rap クライアントマネージャから出力される監査ログを抑止する場合に指定します。

Y

rap クライアントマネージャから出力される監査ログの出力を抑止します。

N

rap クライアントマネージャから出力される監査ログの出力を抑止しません。

このオペランドは、ログサービス定義の log_audit_out オペランドに Y を指定した場合だけ有効です。

●log_audit_message=監査ログを取得する項目のメッセージ ID [、監査ログを取得する項目のメッセージ ID] ...

～〈符号なし整数〉((33400～99999))

OpenTP1 が取得する監査ログのうち、rap クライアントマネージャサービス定義に指定できる監査ログのメッセージ ID を指定します。指定できるメッセージ ID は、最大 2048 個です。

このオペランドで指定できるメッセージ ID については、「[付録 C 監査イベントを取得する定義](#)」を参照してください。

このオペランドを省略し、ユーザサービスデフォルト定義でも指定を省略した場合、ログサービス定義の値を仮定します。このオペランドは、ログサービス定義の log_audit_out オペランドに Y を指定し、rap クライアントマネージャサービス定義の log_audit_out_suppress オペランドに N を指定した場合に有効です。

●rap_watch_time=メッセージ送受信最大監視時間

～〈符号なし整数〉((0～65535))《180》(単位：秒)

rap クライアントマネージャがメッセージの送受信を開始してから、送受信が完了するまでの監視時間を指定します。監視時間が経過してもメッセージの送受信が完了しない場合は、メッセージを出力し、ネットワーク障害と同様の処理を行います。

0 を指定した場合は、時間監視をしません。

コマンド形式

なし。

注意事項

- rap クライアントマネージャサービス定義は、rapdfgen コマンドの入力となる定義ファイルであり、\$DCCONFPATH の下に置いた場合、新たに作成された rap クライアントマネージャ用ユーザサービス定義に上書きされるので、\$DCCONFPATH の下には置かないでください。\$DCCONFPATH の下に置いた場合の動作は保証しません。
- rap クライアントマネージャ用ユーザサービス定義は、\$DCCONFPATH の下にあることを前提としているので、\$DCCONFPATH の下から\$DCUAPCONFPATH の下に移動しないでください。\$DCUAPCONFPATH の下に移動した場合の動作は保証しません。
- rap クライアントマネージャサービス定義の指定値を変更する場合は、rap クライアントマネージャが正常終了したあとに変更してください。

性能検証用トレース定義

形式

set 形式

```
[set prf_file_size=性能検証用トレース情報ファイルのサイズ]
[set prf_information_level=1|0]
[set prf_file_count=性能検証用トレース情報ファイルの世代数]
[set prf_trace_backup=Y|N]
[set prf_buff_size=性能検証用トレースバッファサイズ]
```

コマンド形式

なし。

機能

性能検証用トレース情報を取得する場合の実行環境を定義します。

説明

set 形式のオペランド

●prf_file_size=性能検証用トレース情報ファイルのサイズ

～〈符号なし整数〉((1024～1048576))《10240》(単位：キロバイト)

性能検証用トレース情報ファイルのサイズを指定します。

トレース情報は、通常ファイルに出力します。このファイルは性能検証用トレース定義の prf_file_count オペランドの指定値分の世代を用意し、すべての世代を使用したらオーバーラップして上書きします。このため、ある程度時間が経つと古いトレース情報がなくなります。prf_file_size オペランドの指定値を大きくすることで、ファイルを上書きする時間を延長できます。

トレースファイルの 1 ファイルサイズの算出式を、次に示します (単位：バイト)。

1 ファイルサイズ=128 + (1 トランザクションに必要なトレースデータ長※×実行トランザクション数)

注※

1 トランザクションに必要なトレースデータ長は、マニュアル「OpenTP1 運用と操作」を参照して計算してください。

●prf_information_level=1|0

～《1》

性能検証用トレース関連のメッセージの表示レベルを指定します。

0

表示レベル 0 の性能検証用トレース関連のメッセージ（警告および障害メッセージ）を syslog に出力します。

1

表示レベル 1 以下の性能検証用トレース関連のメッセージを syslog に出力します。

性能検証用トレース関連のメッセージと表示レベルの関係を次に示します。

メッセージ	表示レベル
KFCA26700-W	0
KFCA26705-W	0
KFCA26710-I	1

表示レベル 0 のメッセージの出力を抑止する場合、システム共通定義、または性能検証用トレース定義で環境変数 DCSYSLOGOUT に 0 を指定してください。システム共通定義および性能検証用トレース定義の両方に、環境変数 DCSYSLOGOUT が指定されている場合、システム共通定義の指定が優先されます。

環境変数 DCSYSLOGOUT の詳細については、「[ログサービス定義](#)」の環境変数 DCSYSLOGOUT の説明を参照してください。

●prf_file_count=性能検証用トレース情報ファイルの世代数

～〈符号なし整数〉((3～256))《3》

性能検証用トレース情報ファイルの世代数を指定します。

●prf_trace_backup=Y|N

～《Y》

OpenTP1 の終了時に prf トレースファイルのバックアップを取得するかどうかを指定します。

Y

prf トレースファイルのバックアップを取得します。

N

prf トレースファイルのバックアップを取得しません。

バックアップファイルを取得する場合、バックアップファイルは、\$DCDIR/spool/save 下に作成されます。

バックアップする対象ファイルを次の表に示します。

トレースファイル名称	ファイル名	バックアップファイル名※1
性能検証用トレース情報ファイル	prf_nnn※2	prf_nnn.bk1, prf_nnn.bk2

トレースファイル名称	ファイル名	バックアップファイル名※1
XAR 性能検証用トレース情報ファイル	_xr_nnn※2	_xr_nnn.bk1, _xr_nnn.bk2
JNL 性能検証用トレース情報ファイル	_jl_nnn※2	_jl_nnn.bk1, _jl_nnn.bk2
LCK 性能検証用トレース情報ファイル	_lk_nnn※2	_lk_nnn.bk1, _lk_nnn.bk2
MCF 性能検証用トレース情報取得ファイル	_mc_nnn※2	_mc_nnn.bk1, _mc_nnn.bk2
TRN イベントトレース情報ファイル	_tr_nnn※2	_tr_nnn.bk1, _tr_nnn.bk2
NAM イベントトレース情報ファイル	_nm_001, _nm_002, _nm_003	_nm_nnn.bk1, _nm_nnn.bk2
プロセスサービスイベントトレース情報ファイル	_pr_001, _pr_002, _pr_003	_pr_nnn.bk1, _pr_nnn.bk2
FIL イベントトレース情報ファイル	_fl_001, _fl_002, _fl_003	_fl_nnn.bk1, _fl_nnn.bk2

注※1

nnn：各トレースファイルのファイル名に対応した値です。

注※2

nnn：それぞれ次に示す定義の prf_file_count オペランドで指定した値を上限とした 001 から始まる値です。

性能検証用トレース：性能検証用トレース定義

XAR 性能検証用トレース：XAR 性能検証用トレース定義

JNL 性能検証用トレース：JNL 性能検証用トレース定義

LCK 性能検証用トレース：LCK 性能検証用トレース定義

MCF 性能検証用トレース：MCF 性能検証用トレース定義

TRN イベントトレース：TRN イベントトレース定義

各トレースファイルの詳細については、マニュアル「OpenTP1 解説」を参照してください。

prf トレースファイルはトラブルシュートとして有効な情報です。バックアップを取得していなかったために、障害発生時の情報が残っていなかった場合、問題解決に時間が掛かるおそれがあります。バックアップを取得しない場合は、prf_file_size オペランドの拡張、および prf_file_count オペランドの拡張を検討してください。

なお、JNL 性能検証用トレース情報ファイルの場合、このオペランドは、JNL 性能検証用トレース定義でも指定できます。指定値の優先順位は次のとおりです（1.> 2.）。

1. JNL 性能検証用トレース定義

2. 性能検証用トレース定義

●prf_buff_size=性能検証用トレースバッファサイズ

～〈符号なし整数〉((96～131072))《10240》(単位：キロバイト)

性能検証用トレース情報を格納するバッファのサイズを指定します。

OpenTP1 のシステムサービスプロセスや UAP プロセスからの性能検証用トレース情報は、いったん共用メモリ上のバッファに格納された上でトレースファイルに出力されます。このとき、一時的な負荷の増大などで、一度に大量のトレース情報取得が行われたりファイルへの出力が滞ると、バッファが満杯になることがあります。バッファが満杯になるとその状態が解消するまで以降のトレース情報は捨てられることになります。

このオペランドは、バッファのサイズを大きくすることでバッファあふれによるトレース情報抜けの発生を低減できます。

コマンド形式

なし。

注意事項

この定義の使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

XAR 性能検証用トレース定義

形式

set 形式

```
[set prf_file_size=XAR性能検証用トレース情報ファイルのサイズ]
[set prf_information_level=1|0]
[set prf_file_count=XAR性能検証用トレース情報ファイルの世代数]
```

コマンド形式

なし。

機能

XAR 性能検証用トレース情報を取得する場合の実行環境を定義します。

説明

set 形式のオペランド

●prf_file_size=XAR 性能検証用トレース情報ファイルのサイズ

～ 〈符号なし整数〉 ((1024～1048576)) 《10240》 (単位：キロバイト)

XAR 性能検証用トレース情報ファイルのサイズを指定します。

トレース情報は、通常ファイルに出力します。このファイルは XAR 性能検証用トレース定義の prf_file_count オペランドの指定値分の世代を用意し、すべての世代を使用したらオーバーラップして上書きします。このため、ある程度時間が経つと古いトレース情報がなくなります。このオペランドの指定値を大きくすることで、ファイルを上書きする時間を延長できます。

トランザクション実行中に取得するトレースファイルの 1 ファイルサイズの算出式を、次に示します (単位：バイト)。

1 ファイルサイズ※=128 + (384× (4 + 1 トランザクションで実行する RPC コール回数) ×実行トランザクション数)

注※

トレース量は、トランザクションの最適化などの条件によって異なります。

●prf_information_level=1|0

～ 《1》

XAR 性能検証用トレース関連のメッセージの表示レベルを指定します。

0

表示レベル 0 の XAR 性能検証用トレース関連のメッセージ（警告および障害メッセージ）を syslog に出力します。

1

表示レベル 1 以下の XAR 性能検証用トレース関連のメッセージを syslog に出力します。

XAR 性能検証用トレース関連のメッセージ ID と表示レベルの関係を次に示します。

メッセージ	表示レベル
KFCA26700-W	0
KFCA26705-W	0
KFCA26710-I	1

表示レベル 0 のメッセージの出力を抑止する場合、システム共通定義、または性能検証用トレース定義で環境変数 DCSYSLOGOUT に 0 を指定してください。システム共通定義および性能検証用トレース定義の両方に、環境変数 DCSYSLOGOUT が指定されている場合、システム共通定義の指定が優先されます。

環境変数 DCSYSLOGOUT の詳細については、「[ログサービス定義](#)」の環境変数 DCSYSLOGOUT の説明を参照してください。

●prf_file_count=XAR 性能検証用トレース情報ファイルの世代数
～ 〈符号なし整数〉 ((3～256)) 《3》

XAR 性能検証用トレース情報ファイルの世代数を指定します。

コマンド形式

なし。

注意事項

この定義の使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

JNL 性能検証用トレース定義

形式

set 形式

```
[set prf_file_size=JNL性能検証用トレース情報ファイルのサイズ]  
[set prf_file_count=JNL性能検証用トレース情報ファイルの世代数]  
[set prf_trace_backup=Y|N]
```

コマンド形式

なし。

機能

JNL 性能検証用トレース情報を取得する場合の実行環境を定義します。

説明

set 形式のオペランド

●prf_file_size=JNL 性能検証用トレース情報ファイルのサイズ

～〈符号なし整数〉((1024～1048576))《1024》(単位：キロバイト)

JNL 性能検証用トレース情報ファイルのサイズを指定します。

トレース情報は、通常ファイルに出力します。このファイルは JNL 性能検証用トレース定義の prf_file_count オペランドの指定値分の世代を用意し、すべての世代を使用したらオーバーラップして上書きします。このため、ある程度時間が経つと古いトレース情報がなくなります。このオペランドの指定値を大きくすることで、ファイルを上書きする時間を延長できます。

トレースファイルの 1 ファイルサイズの算出式を次に示します (単位：バイト)。

- システム共通定義の jnl_prf_event_trace_level オペランドに 00000001 を指定した場合
1 ファイルサイズ=128×a + 128×b
- システム共通定義の jnl_prf_event_trace_level オペランドに 00000002 を指定した場合
 - 適用 OS が UNIX の場合
1 ファイルサイズ=512×a + 128×b + 256×c
 - 適用 OS が Windows の場合
1 ファイルサイズ=576×a + 128×b + 256×c

(凡例)

a：1 トランザクション内で発生するジャーナルライト処理回数

b：1 トランザクション内で発生するバッファ空き待ち回数

c:1 トランザクションで取得されるジャーナルレコード数

●prf_file_count=JNL 性能検証用トレース情報ファイルの世代数

～〈符号なし整数〉((3～256))《3》

JNL 性能検証用トレース情報ファイルの世代数を指定します。

このオペランドは、性能検証用トレース定義でも指定できます。指定値の優先順位は次のとおりです (1. > 2.)。

1. JNL 性能検証用トレース定義
2. 性能検証用トレース定義

●prf_trace_backup=Y|N

～《Y》

OpenTP1 の終了時に JNL 性能検証用トレース情報ファイルのバックアップを取得するかどうかを指定します。

Y

JNL 性能検証用トレース情報ファイルのバックアップを取得します。

N

JNL 性能検証用トレース情報ファイルのバックアップを取得しません。

バックアップファイルを取得する場合、バックアップファイルは、\$DCDIR/spool/save 下に作成されます。

バックアップする対象ファイルを次に示します。

トレースファイル名称	ファイル名	バックアップファイル名※1
JNL 性能検証用トレース情報ファイル	_jl_nnn※2	_jl_nnn.bk1, _jl_nnn.bk2

注※1

nnn: トレースファイルのファイル名に対応した値です。

注※2

nnn: JNL 性能検証用トレース定義の prf_file_count オペランドで指定した値を上限とした 001 から始まる値です。

JNL 性能検証用トレース情報ファイルの詳細については、マニュアル「OpenTP1 解説」を参照してください。

JNL 性能検証用トレース情報ファイルはトラブルシュートとして有効な情報です。バックアップを取得していなかったために、障害発生時の情報が残っていなかった場合、問題解決に時間が掛かるおそれがあります。バックアップを取得しない場合は、prf_file_size オペランドの拡張、および prf_file_count オペランドの拡張を検討してください。

このオペランドは、性能検証用トレース定義でも指定できます。指定値の優先順位は次のとおりです（1. > 2.）。

1. JNL 性能検証用トレース定義
2. 性能検証用トレース定義

コマンド形式

なし。

注意事項

この定義の使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

LCK 性能検証用トレース定義

形式

set 形式

```
[set prf_file_size=LCK性能検証用トレース情報ファイルのサイズ]
[set prf_information_level=1|0]
[set prf_file_count=LCK性能検証用トレース情報ファイルの世代数]
```

コマンド形式

なし。

機能

LCK 性能検証用トレース情報を取得する場合の実行環境を定義します。

説明

set 形式のオペランド

●prf_file_size=LCK 性能検証用トレース情報ファイルのサイズ

～ 〈符号なし整数〉 ((1024～1048576)) 《5120》 (単位：キロバイト)

LCK 性能検証用トレース情報ファイルのサイズを指定します。

トレース情報は、通常ファイルに出力します。このファイルは LCK 性能検証用トレース定義の prf_file_count オペランドの指定値分の世代を用意し、すべての世代を使用したらオーバーラップして上書きします。このため、ある程度時間が経つと古いトレース情報がなくなります。このオペランドの指定値を大きくすることで、ファイルを上書きする時間を延長できます。

トランザクション実行中に取得するトレースファイルの 1 ファイルサイズの算出式を次に示します (単位：バイト)。

1 ファイルサイズ = $128 + ((1024 \times (D + T + M) + R \times 128) \times \text{実行トランザクション数})$

D : 1 トランザクションブランチでの DAM サービスへのアクセス数 (参照・更新)

T : 1 トランザクションブランチでの TAM サービスへのアクセス数 (参照・更新)

M : 1 トランザクションブランチでの MQA サービスへのアクセス数 (参照・更新)

R : 1 トランザクションブランチ当たりの排他制御関数実行回数

なお、算出式で取得したファイルサイズはおおよそのサイズです。UAP の構成などによっては、算出値以上となる場合があります。

●prf_information_level=1|0

～ 《1》

LCK 性能検証用トレース関連のメッセージの表示レベルを指定します。

- 0
- 表示レベル 0 の LCK 性能検証用トレース関連のメッセージ（警告および障害メッセージ）を syslog に出力します。
- 1
- 表示レベル 1 以下の LCK 性能検証用トレース関連のメッセージを syslog に出力します。

LCK 性能検証用トレース関連のメッセージと表示レベルの関係を次に示します。

メッセージ	表示レベル
KFCA26700-W	0
KFCA26705-W	0
KFCA26710-I	1

表示レベル 0 のメッセージの出力を抑止する場合、システム共通定義、または性能検証用トレース定義で環境変数 DCSYSLOGOUT に 0 を指定してください。システム共通定義および性能検証用トレース定義の両方に、環境変数 DCSYSLOGOUT が指定されている場合、システム共通定義の指定が優先されます。

環境変数 DCSYSLOGOUT の詳細については、「[ログサービス定義](#)」の環境変数 DCSYSLOGOUT の説明を参照してください。

●prf_file_count=LCK 性能検証用トレース情報ファイルの世代数
～ 〈符号なし整数〉 (3～256) 《3》

LCK 性能検証用トレース情報ファイルの世代数を指定します。

コマンド形式

なし。

注意事項

この定義の使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

TRN イベントトレース定義

形式

set 形式

```
[set prf_file_size=TRN イベントトレース情報ファイルのサイズ]
[set prf_information_level=1|0]
[set prf_file_count=TRN イベントトレース情報ファイルの世代数]
```

コマンド形式

なし。

機能

TRN イベントトレース情報を取得する場合の実行環境を定義します。

説明

set 形式のオペランド

●prf_file_size=TRN イベントトレース情報ファイルのサイズ

～ 〈符号なし整数〉 ((1024～1048576)) 《10240》 (単位：キロバイト)

TRN イベントトレース情報ファイルのサイズを指定します。

トレース情報は、通常ファイルに出力します。このファイルは TRN イベントトレース定義の prf_file_count オペランドの指定値分の世代を用意し、すべての世代を使用したらオーバーラップして上書きします。このため、ある程度時間が経つと古いトレース情報がなくなります。prf_file_size オペランドの指定値を大きくすることで、ファイルを上書きする時間を延長できます。

トランザクション実行中に取得するトレースファイルの 1 ファイルサイズの算出式を、次に示します (単位：バイト)。

1 ファイルサイズ※=128 + (12×アクセスするリソースマネージャ数×320×実行トランザクション数)

注※

2 相コミットのトランザクションの場合、1 トランザクションブランチあたりに取得するトレース量は「12×アクセスするリソースマネージャ数」となります。ただし、トレース量は、ユーザサーバにリンクされている XA インタフェースオブジェクトファイルや、トランザクションの最適化などの条件によって異なります。

●prf_information_level=1|0

～ 《1》

TRN イベントトレース関連のメッセージの表示レベルを指定します。

0

表示レベル 0 の TRN イベントトレース関連のメッセージ（警告および障害メッセージ）を syslog に出力します。

1

表示レベル 1 以下の TRN イベントトレース関連のメッセージを syslog に出力します。

TRN イベントトレース関連のメッセージ ID と表示レベルの関係を次に示します。

メッセージ ID	表示レベル
KFCA26700-W	0
KFCA26705-W	0
KFCA26710-I	1

表示レベル 0 のメッセージの出力を抑止する場合、システム共通定義、または性能検証用トレース定義で環境変数 DCSYSLOGOUT に 0 を指定してください。システム共通定義および性能検証用トレース定義の両方に、環境変数 DCSYSLOGOUT が指定されている場合、システム共通定義の指定が優先されます。

環境変数 DCSYSLOGOUT の詳細については、「[ログサービス定義](#)」の環境変数 DCSYSLOGOUT の説明を参照してください。

●prf_file_count=TRN イベントトレース情報ファイルの世代数

～〈符号なし整数〉((3～256))《3》

TRN イベントトレース情報ファイルの世代数を指定します。

コマンド形式

なし。

注意事項

この定義の使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

リアルタイム統計情報サービス定義

形式

set 形式

```
[set rts_trcput_interval=統計情報の取得間隔]
[set rts_service_max=最大取得サービス数]
[set rts_item_max=最大取得項目数]
[set rts_log_file=Y|N]
[set rts_log_file_name=RTSログファイル名]
[set rts_log_file_size=RTSログファイルのサイズ]
[set rts_log_file_count=RTSログファイルの世代数]
[set rts_log_file_backup=Y|N]
[set rts_swap_message=Y|N]
```

コマンド形式

```
[rtsput -u {sys|srv|svc|obj}
        [-s サーバ名] [-v サービス名]
        [-o 取得対象名1] [-b 取得対象名2]
        [-e 項目ID [, 項目ID] ...]
        [-f リアルタイム取得項目定義ファイル名] ]
```

機能

リアルタイム統計情報サービスで、統計情報を取得するための実行環境を定義します。

説明

set 形式のオペランド

●rts_trcput_interval=統計情報の取得間隔

～ 〈符号なし整数〉 ((10～86400)) 《600》 (単位：秒)

リアルタイム統計情報を取得し編集する時間間隔を、秒単位で指定します。

●rts_service_max=最大取得サービス数

～ 〈符号なし整数〉 ((1～1000)) 《64》

リアルタイム統計情報サービスで、統計情報を取得する対象の最大数を指定します。

●rts_item_max=最大取得項目数

～ 〈符号なし整数〉 ((1～1000)) 《64》

リアルタイム統計情報サービスが、一つの取得対象で統計情報を取得できるイベントの最大数を指定します。

●rts_log_file=Y|N

～《Y》

RTS サービス用の共用メモリに取得した統計情報を RTS ログファイルに出力するかどうかを指定します。

Y

取得した統計情報を RTS ログファイルに出力します。

N

取得した統計情報を RTS ログファイルに出力しません。

●rts_log_file_name=RTS ログファイル名

～〈1～63 文字のパス名〉《\$DCDIR/spool/dcrtsinf/rtslog》

統計情報を出力する RTS ログファイル名を絶対パスで指定します。

このオペランドで指定したパスに、すでに同名のファイルが存在する場合は、RTS ログファイルを作り直して使用します。

RTS ログファイルの出力先に RTS ログファイル名と同じ名称のファイル、またはディレクトリを作成しないでください。同じ名称のファイル、またはディレクトリを作成した場合の動作は保証できません。

ユーザサービスデフォルト定義の uid オペランドに OpenTP1 管理者以外のユーザを指定する場合、rts_log_file_name オペランドには、uid オペランドに指定したユーザが書き込み権限を持つパスを指定してください。

リアルタイム統計情報サービスは、どのユーザ ID で起動したかには関係なく、RTS ログファイルの出力先に対する書き込み権限があれば正常に動作し、権限がなければ KFCA32734-W メッセージを出力して RTS ログファイルへの出力機能を停止します。また、rtssetup コマンドで作成する定義ファイル (RTSSUP および RTSSPP) には uid オペランド、および groups オペランドを指定していません。このため、ユーザサービスデフォルト定義に uid オペランド、および groups オペランドが指定されている場合は、指定されたユーザおよびグループでリアルタイム統計情報サービスを起動します。

●rts_log_file_size=RTS ログファイルのサイズ

～〈符号なし整数〉((1024～1048576))《1024》(単位：キロバイト)

統計情報を出力する RTS ログファイルのサイズを指定します。

RTS ログファイルは、rts_log_file_count オペランドで指定した値分の世代を用意します (rts_log_file_size オペランドの指定値×rts_log_file_count オペランドの指定値分のディスク容量が必要です)。すべての世代を使用した場合は、オーバーラップして上書きします。このため、ある程度の時間が経過すると、古いリアルタイム統計情報はなくなります。RTS ログファイルを上書きする時間を延長したい場合は、rts_log_file_size オペランドに大きい値を指定してください。

また、rts_log_file_size オペランドには、一度の契機で出力するリアルタイム統計情報のサイズより大きな値を指定してください。一度の契機で出力するリアルタイム統計情報のサイズは次に示す計算式で求めます。

$$\text{一度の契機で出力するリアルタイム統計情報のサイズ} \\ = (96 + (40 \times \text{rts\_item\_maxオペランドの値})) \times \text{リアルタイム統計情報の取得対象の数}^{\ast}$$

注※

リアルタイム統計情報の取得対象の数は、リアルタイム統計情報サービス定義の rtspout 定義コマンドで指定した取得対象の数です。

ただし、-u オプションに srv を指定した場合、-s オプションの引数にユーザサーバを指定したときのリアルタイム統計情報の取得対象の数は、「service オペランドに定義したサービス数 + 2」になります。rtsstats コマンドで取得対象を変更する場合は、その数も考慮してください。

rtspout 定義コマンドは同じサービスやサーバに対して重複して指定できます。重複して指定した場合、リアルタイム統計情報サービスは、各 rtspout 定義コマンドで指定した取得項目をマージして取得します。重複して指定している場合、取得対象の数は、重複した取得対象ごとに一つと数えられるため、増加しません。

システム全体の統計情報は、リアルタイム統計情報の動作上、取得対象としてカウントしませんが、RTS ログファイルの出力に関してはカウントする必要があります。そのため、システム全体の統計情報を取得している場合は、上記の計算式の「リアルタイム統計情報の取得対象の数」に 1 を加算して計算してください。

●rts_log_file_count=RTS ログファイルの世代数

～ 〈符号なし整数〉 ((1～10)) 《3》

統計情報を出力する RTS ログファイルの世代数を指定します。

●rts_log_file_backup=Y|N

～ 《Y》

リアルタイム統計情報サービスの開始時に、RTS ログファイルのバックアップファイルを作成するかどうかを指定します。

Y

RTS ログファイルのバックアップファイルを作成します。

バックアップファイルは、RTS ログファイル名に「.bk」を付与した名称で、RTS ログファイルの出力先ディレクトリに作成します。

N

RTS ログファイルのバックアップファイルを作成しません。

RTS ログファイルのバックアップファイルを作成する場合、出力先ディレクトリに最大で次のディスク容量が必要になります。

RTS ログファイルの出力先ディレクトリのディスク容量=rts_log_file_size オペランドの指定値×
rts_log_file_count オペランドの指定値×2

RTS ログファイルの出力先ディレクトリのディスク容量に余裕があることを確認してください。

RTS ログファイルの出力先に、バックアップファイルと同じ名称のファイル、またはバックアップファイルと同じ名称のディレクトリを作成しないでください。同じ名称のファイルを作成した場合は、バックアップファイルで上書きします。同じ名称のディレクトリを作成した場合は、RTS ログファイルのバックアップを作成できません。

●rts_swap_message=Y|N

～《N》

リアルタイム統計情報を出力する RTS ログファイルを切り替える場合に、RTS ログファイルのスワップメッセージ (KFCA32740-I メッセージ) を出力するかどうかを指定します。

Y

RTS ログファイルのスワップメッセージを出力します。

N

RTS ログファイルのスワップメッセージを出力しません。

コマンド形式

次ページ以降に記述しています。

注意事項

- rts_service_max オペランド、および rts_item_max オペランドの値は、一度リアルタイム統計情報サービスを開始したあと、OpenTP1 のオンライン中に値を変更しても反映されません。定義を変更したい場合は、一度 OpenTP1 を終了させてください。
- RTS ログファイルの統計情報の出力処理に時間が掛かった場合、rts_trcput_interval オペランドに指定した時間間隔を超えて統計情報を取得する場合があります。
- 環境変数 DCDIR の設定値が 41 文字を超えた場合、rts_log_file_name オペランドを省略しないでください。定義解析時にエラーになります。

名称

統計情報取得サービスの指定

形式

```
[rtspu    -u {sys|srv|svc|obj}
          [-s サーバ名] [-v サービス名]
          [-o 取得対象名1] [-b 取得対象名2]
          [-e 項目ID [,項目ID] ...]
          [-f リアルタイム取得項目定義ファイル名] ]
```

機能

リアルタイム統計情報サービスで統計情報を取得する対象と項目を指定します。

オプション

●-u {sys | srv | svc | obj}

取得するリアルタイム統計情報の取得対象種別を指定します。

sys

システム全体の統計情報を取得します。

sys を指定した場合、リアルタイム統計情報サービス定義で指定した rts_service_max オペランドの値は消費しません。

srv

指定したサーバの統計情報を取得します。

-s オプションにユーザサーバ名を指定した場合、次に示す統計情報を取得します。

- サーバ単位での統計情報
- 指定したユーザサーバのユーザサービス定義の service オペランドに定義されているすべてのサービスの統計情報
- サービス以外の処理の統計情報

このため、取得対象の数は、service オペランドに指定したサービスの数に 2 を加えたものになります。システムサーバや SUP などの、サービスを持たないサーバ名を指定した場合は、サーバ単位の統計情報だけを取得します。

SVC

指定したユーザサーバのサービス単位で統計情報を取得します。

obj

-o オプションと-b オプションを組み合わせ、統計情報の取得対象オブジェクトを指定します。取得対象オブジェクトとオプションの組み合わせについては、-o オプションおよび-b オプションの説明を参照してください。

●-s サーバ名

～ 〈1～8 文字で先頭が英字の英数字〉

統計情報を取得するサーバの名称を指定します。

-u オプションに srv または svc を指定した場合、必ず-s オプションを指定してください。-u オプションに sys または obj を指定した場合、-s オプションは指定できません。

●-v サービス名

～ 〈1～31 文字の識別子〉

統計情報を取得するサービスの名称を指定します。

-s オプションで指定したサーバの指定したサービスの統計情報を取得します。

-u オプションに svc を指定した場合、必ず-v オプションを指定してください。-u オプションに sys, srv, または obj を指定した場合、-v オプションは指定できません。

●-o 取得対象名 1

～ 〈1～8 文字の文字列〉

●-b 取得対象名 2

～ 〈1～63 文字の文字列〉

-u オプションに obj を指定した場合に、リアルタイム統計情報を取得する取得対象オブジェクト名を指定します。

-o オプションおよび-b オプションで指定できる取得対象オブジェクトを次の表に示します。-u オプションに obj を指定した場合は、必ず次の表に従って指定してください。

表 3-13 rtspout 定義コマンドのオプションと指定できる取得対象オブジェクト

-o オプションの指定値	-b オプションの指定値	指定できる取得対象オブジェクト
ポート番号※	IP アドレス※	指定したサービス情報の参照先ノード
論理端末名	指定を省略	指定した論理端末
指定を省略	サービスグループ名	指定したサービスグループ

注※

システム共通定義の all_node オペランド、またはドメイン定義ファイルに指定したノードの IP アドレスとポート番号を指定してください。

-u オプションに sys, srv, または svc を指定した場合, -o オプションおよび-b オプションは指定できません。

●-e 項目 ID

～ 〈符号なし整数〉 ((1000～9999))

統計情報を取得する項目の項目 ID を指定します。

項目 ID 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。

●-f リアルタイム取得項目定義ファイル名

～ 〈1～8 文字の識別子〉

取得する統計情報の項目を定義したリアルタイム取得項目定義ファイルを指定します。

注意事項

- -u オプションに srv を, -s オプションにユーザサーバ名を指定した場合, 指定されたサーバのユーザサービス定義ファイルを解析してサービス名を取得します。-s オプションにユーザサーバ名を指定する場合は, 必ず対応するユーザサービス定義ファイルを準備してください。
- rtspout 定義コマンドで指定できる統計情報の取得対象は, リアルタイム統計情報サービス定義の rts_service_max オペランドに指定した数までです。また, 一つの取得対象で取得できるイベントの数は, リアルタイム統計情報サービス定義の rts_item_max オペランドに指定した数までです。
- -f オプションおよび-e オプションの指定を省略した場合, 統計情報を取得するための領域を確保します。ただし, 取得する統計情報は任意区間でのユーザサーバ実行時間の情報だけです。
- rtspout 定義コマンドは, 同じサービスやサーバに対して重複して指定できます。重複して指定した場合, リアルタイム統計情報サービスは, 各定義で指定した取得項目をマージして取得します。rtspout 定義コマンドおよびリアルタイム取得項目定義ファイルの指定例を表 3-14 に, 表 3-14 のとおりに指定した場合のリアルタイム統計情報サービスで取得する項目を表 3-15 に示します。

rtspout 定義コマンドの定義例の詳細については, 「8.2 リアルタイム統計情報の取得項目の定義例」を参照してください。

表 3-14 rtspout 定義コマンドおよびリアルタイム取得項目定義ファイルの指定例

rtspout 定義コマンドの指定	定義ファイルで指定した取得項目
rtspout -u srv -s SERVER_A -f 定義ファイル 1	項目 A, 項目 C
rtspout -u svc -s SERVER_A -v SERVICE_B -f 定義ファイル 2	項目 B
rtspout -u svc -s SERVER_A -v SERVICE_C -f 定義ファイル 3	項目 A, 項目 D

表 3-15 リアルタイム統計情報サービスで取得する項目

サーバ名	サービス名	項目 A	項目 B	項目 C	項目 D
SERVER_A	サービス全体	○	×	○	×
	サービス外	○	×	○	×
	SRVICE_A	○	×	○	×
	SRVICE_B	○	○	○	×
	SRVICE_C	○	×	○	○

(凡例)

○：取得します。

×：取得しません。

- -u オプションに svc を指定した場合、-s オプションに指定したサーバ名や、-v オプションに指定したサービス名が存在するかのチェックは行いません。また、-u オプションに obj を指定した場合、-o オプションや-b オプションに指定した値が妥当であるかどうかのチェックは行いません。不要な取得対象を登録した場合は、rtsls コマンドを実行して取得対象の構成を確認し、-d オプションを指定した rtsstats コマンドを実行して削除してください。
- リアルタイム統計情報サービスの開始後にユーザサービス定義に追加または削除したサービスは、rtspu 定義コマンドの-u オプションに srv を指定している場合でも、リアルタイム統計情報の取得対象に反映されません。リアルタイム統計情報サービスの開始後に取得対象の設定を変更するには、rtsstats コマンドを使用するか、またはリアルタイム統計情報サービス (RTSSUP) を再起動してください。
また、リアルタイム統計情報サービスの開始後に、サービス関数動的ローディング機能で追加または削除したサービスも、リアルタイム統計情報の取得対象に反映されません。リアルタイム統計情報サービスの開始後に取得対象の設定を変更するには、rtsstats コマンドを使用するか、またはリアルタイム統計情報サービス (RTSSUP) を再起動してください。

形式

set 形式

```
[set rts_cpd_collct_cpd=Y|N]
[set rts_cpd_validt_cpd=Y|N]
[set rts_jnl_buf_full=Y|N]
[set rts_jnl_wait_buf=Y|N]
[set rts_jnl_jnl_output=Y|N]
[set rts_jnl_io_wait=Y|N]
[set rts_jnl_write=Y|N]
[set rts_jnl_swap=Y|N]
[set rts_jnl_jnl_input=Y|N]
[set rts_jnl_read=Y|N]
[set rts_lck_lock_acqst=Y|N]
[set rts_lck_lock_wait=Y|N]
[set rts_lck_deadlock=Y|N]
[set rts_nam_global_cache_hit=Y|N]
[set rts_nam_local_cache_hit=Y|N]
[set rts_nam_lookup=Y|N]
[set rts_nam_node_lookup=Y|N]
[set rts_nam_node_lookup_responce=Y|N]
[set rts_osl_stamem_acq=Y|N]
[set rts_osl_stamem_pol=Y|N]
[set rts_osl_dynmem_acq=Y|N]
[set rts_osl_dynmem_pol=Y|N]
[set rts_prc_prc_genert=Y|N]
[set rts_prc_uap_abnml=Y|N]
[set rts_prc_sys_abnml=Y|N]
[set rts_prc_prc_term=Y|N]
[set rts_prc_prc_num=Y|N]
[set rts_que_read=Y|N]
[set rts_que_write=Y|N]
[set rts_que_read_err=Y|N]
[set rts_que_write_err=Y|N]
[set rts_que_wait_buf=Y|N]
[set rts_que_real_read=Y|N]
[set rts_que_real_write=Y|N]
[set rts_que_delay_wrt=Y|N]
[set rts_que_delay_rec=Y|N]
[set rts_que_delay_msg=Y|N]
[set rts_rpc_rpc_call=Y|N]
[set rts_rpc_rpc_call_chained=Y|N]
[set rts_rpc_usr_srvc=Y|N]
[set rts_rpc_rpc_ovrtim=Y|N]
[set rts_scd_scd_wait=Y|N]
[set rts_scd_schedule=Y|N]
[set rts_scd_using_buf=Y|N]
[set rts_scd_lack_buf=Y|N]
[set rts_scd_scd_stay=Y|N]
[set rts_scd_svc_scd_wait=Y|N]
[set rts_scd_svc_using_buf=Y|N]
[set rts_scd_parallel=Y|N]
[set rts_trn_commit=Y|N]
```

```
[set rts_trn_rollback=Y|N]
[set rts_trn_cmt_cmd=Y|N]
[set rts_trn_rbk_cmd=Y|N]
[set rts_trn_haz_cmd=Y|N]
[set rts_trn_mix_cmd=Y|N]
[set rts_trn_branch=Y|N]
[set rts_trn_sync_point=Y|N]
[set rts_dam_read=Y|N]
[set rts_dam_read_err=Y|N]
[set rts_dam_write=Y|N]
[set rts_dam_write_err=Y|N]
[set rts_dam_fj=Y|N]
[set rts_dam_trn_branch=Y|N]
[set rts_dam_cache_block=Y|N]
[set rts_dam_shm_pool=Y|N]
[set rts_tam_real_renew=Y|N]
[set rts_tam_real_renew_time=Y|N]
[set rts_tam_rec_refer=Y|N]
[set rts_tam_rec_renew=Y|N]
[set rts_tam_read=Y|N]
[set rts_tam_read_err=Y|N]
[set rts_tam_write=Y|N]
[set rts_tam_write_err=Y|N]
[set rts_xar_start=Y|N]
[set rts_xar_start_err=Y|N]
[set rts_xar_call=Y|N]
[set rts_xar_call_err=Y|N]
[set rts_xar_end=Y|N]
[set rts_xar_end_err=Y|N]
[set rts_xar_prepare=Y|N]
[set rts_xar_prepare_err=Y|N]
[set rts_xar_commit=Y|N]
[set rts_xar_commit_err=Y|N]
[set rts_xar_rollback=Y|N]
[set rts_xar_rollback_err=Y|N]
[set rts_xar_recover=Y|N]
[set rts_xar_recover_err=Y|N]
[set rts_xar_forget=Y|N]
[set rts_xar_forget_err=Y|N]
[set rts_mcf_ap_scd_stay=Y|N]
[set rts_mcf_ap_usr_srvc=Y|N]
[set rts_mcf_in_msg_scd_wait=Y|N]
[set rts_mcf_out_msg_sync_scd_wait=Y|N]
[set rts_mcf_out_msg_resp_scd_wait=Y|N]
[set rts_mcf_out_msg_prio_scd_wait=Y|N]
[set rts_mcf_out_msg_norm_scd_wait=Y|N]
[set rts_mcf_que_scd_wait_num=Y|N]
```

コマンド形式

なし。

機能

リアルタイム統計情報サービスで取得する統計情報の項目を定義します。

説明

set 形式のオペランド

チェックポイントダンプ情報

●**rts_cpd_collct_cpd=Y|N**

～《N》

チェックポイントダンプ取得イベントを取得するかどうかを指定します。

Y

チェックポイントダンプ取得イベントを取得します。

N

チェックポイントダンプ取得イベントを取得しません。

●**rts_cpd_validt_cpd=Y|N**

～《N》

チェックポイントダンプ有効化イベントを取得するかどうかを指定します。

Y

チェックポイントダンプ有効化イベントを取得します。

N

チェックポイントダンプ有効化イベントを取得しません。

ジャーナル情報

●**rts_jnl_buf_full=Y|N**

～《N》

バッファ満杯イベントを取得するかどうかを指定します。

Y

バッファ満杯イベントを取得します。

N

バッファ満杯イベントを取得しません。

●**rts_jnl_wait_buf=Y|N**

～《N》

空きバッファ待ちイベントを取得するかどうかを指定します。

Y

空きバッファ待ちイベントを取得します。

N

空きバッファ待ちイベントを取得しません。

●**rts_jnl_jnl_output=Y|N**

～《N》

ジャーナル出力（ブロック）イベントを取得するかどうかを指定します。

Y

ジャーナル出力（ブロック）イベントを取得します。

N

ジャーナル出力（ブロック）イベントを取得しません。

●**rts_jnl_io_wait=Y|N**

～《N》

入出力待ちイベントを取得するかどうかを指定します。

Y

入出力待ちイベントを取得します。

N

入出力待ちイベントを取得しません。

●**rts_jnl_write=Y|N**

～《N》

ジャーナル情報の write イベントを取得するかどうかを指定します。

Y

write イベントを取得します。

N

write イベントを取得しません。

●**rts_jnl_swap=Y|N**

～《N》

スワップイベントを取得するかどうかを指定します。

Y

スワップイベントを取得します。

N

スワップイベントを取得しません。

●**rts_jnl_jnl_input=Y|N**

～《N》

ジャーナル入力イベントを取得するかどうかを指定します。

Y

ジャーナル入力イベントを取得します。

N

ジャーナル入力イベントを取得しません。

●**rts_jnl_read=Y|N**

～《N》

ジャーナル情報の read イベントを取得するかどうかを指定します。

Y

read イベントを取得します。

N

read イベントを取得しません。

ロック情報

●**rts_lck_lock_acqst=Y|N**

～《N》

ロック取得イベントを取得するかどうかを指定します。

Y

ロック取得イベントを取得します。

N

ロック取得イベントを取得しません。

●**rts_lck_lock_wait=Y|N**

～《N》

ロック待ちイベントを取得するかどうかを指定します。

Y

ロック待ちイベントを取得します。

N

ロック待ちイベントを取得しません。

●**rts_lck_deadlock=Y|N**

～《N》

デッドロックイベントを取得するかどうかを指定します。

Y

デッドロックイベントを取得します。

N

デッドロックイベントを取得しません。

ネーム情報

●**rts_nam_global_cache_hit=Y|N**

～《N》

グローバルキャッシュヒットイベントを取得するかどうかを指定します。

Y

グローバルキャッシュヒットイベントを取得します。

N

グローバルキャッシュヒットイベントを取得しません。

●**rts_nam_local_cache_hit=Y|N**

～《N》

ローカルキャッシュヒットイベントを取得するかどうかを指定します。

Y

ローカルキャッシュヒットイベントを取得します。

N

ローカルキャッシュヒットイベントを取得しません。

●**rts_nam_lookup=Y|N**

～《N》

サービス情報の検索回数イベントを取得するかどうかを指定します。

Y

サービス情報の検索回数イベントを取得します。

N

サービス情報の検索回数イベントを取得しません。

●**rts_nam_node_lookup=Y|N**

～《N》

指定参照先ノードへのサービス検索の送信回数イベントを取得するかどうかを指定します。

Y

指定参照先ノードへのサービス検索の送信回数イベントを取得します。

N

指定参照先ノードへのサービス検索の送信回数イベントを取得しません。

●**rts_nam_node_lookup_responce=Y|N**

～《N》

指定参照先ノードへのサービス検索の応答送信回数イベントを取得するかどうかを指定します。

Y

指定参照先ノードへのサービス検索の応答受信回数イベントを取得します。

N

指定参照先ノードへのサービス検索の応答受信回数イベントを取得しません。

共用メモリ管理情報

●**rts_osl_stamem_acq=Y|N**

～《N》

静的共用メモリの使用サイズイベントを取得するかどうかを指定します。

Y

静的共用メモリの使用サイズイベントを取得します。

N

静的共用メモリの使用サイズイベントを取得しません。

●**rts_osl_stamem_pol=Y|N**

～《N》

静的共用メモリプールの必要最大サイズイベントを取得するかどうかを指定します。

Y

静的共用メモリプールの必要最大サイズイベントを取得します。

N

静的共用メモリプールの必要最大サイズイベントを取得しません。

●**rts_osl_dynmem_acq=Y|N**

～《N》

動的共用メモリの使用サイズイベントを取得するかどうかを指定します。

Y

動的共用メモリの使用サイズイベントを取得します。

N

動的共用メモリの使用サイズイベントを取得しません。

●**rts_osl_dynmem_pol=Y|N**

～《N》

動的共用メモリプールの必要最大サイズイベントを取得するかどうかを指定します。

Y

動的共用メモリプールの必要最大サイズイベントを取得します。

N

動的共用メモリプールの必要最大サイズイベントを取得しません。

プロセス情報

●**rts_prc_prc_genert=Y|N**

～《N》

プロセス生成イベントを取得するかどうかを指定します。

Y

プロセス生成イベントを取得します。

N

プロセス生成イベントを取得しません。

●**rts_prc_uap_abnml=Y|N**

～《N》

UAP 異常終了イベントを取得するかどうかを指定します。

Y

UAP 異常終了イベントを取得します。

N

UAP 異常終了イベントを取得しません。

●**rts_prc_sys_abnml=Y|N**

～《N》

システムサーバ異常終了イベントを取得するかどうかを指定します。

Y

システムサーバ異常終了イベントを取得します。

N

システムサーバ異常終了イベントを取得しません。

●**rts_prc_prc_term=Y|N**

～《N》

プロセス終了イベントを取得するかどうかを指定します。

Y

プロセス終了イベントを取得します。

N

プロセス終了イベントを取得しません。

●**rts_prc_prc_num=Y|N**

～《N》

起動プロセス数の監視イベントを取得するかどうかを指定します。

Y

起動プロセス数の監視イベントを取得します。

N

起動プロセス数の監視イベントを取得しません。

メッセージキュー情報

●**rts_que_read=Y|N**

～《N》

read メッセージイベントを取得するかどうかを指定します。

Y

read メッセージイベントを取得します。

N

read メッセージイベントを取得しません。

●**rts_que_write=Y|N**

～《N》

write メッセージイベントを取得するかどうかを指定します。

Y

write メッセージイベントを取得します。

N

write メッセージイベントを取得しません。

●**rts_que_read_err=Y|N**

～《N》

メッセージキュー情報の read エラーイベントを取得するかどうかを指定します。

Y

read エラーイベントを取得します。

N

read エラーイベントを取得しません。

●**rts_que_write_err=Y|N**

～《N》

メッセージキュー情報の write エラーイベントを取得するかどうかを指定します。

Y

write エラーイベントを取得します。

N

write エラーイベントを取得しません。

●**rts_que_wait_buf=Y|N**

～《N》

メッセージキュー情報の空きバッファ待ちイベントを取得するかどうかを指定します。

Y

空きバッファ待ちイベントを取得します。

N

空きバッファ待ちイベントを取得しません。

●**rts_que_real_read=Y|N**

～《N》

メッセージキュー情報の実 read イベントを取得するかどうかを指定します。

Y

実 read イベントを取得します。

N

実 read イベントを取得しません。

●**rts_que_real_write=Y|N**

～《N》

メッセージキュー情報の実 write イベントを取得するかどうかを指定します。

Y

実 write イベントを取得します。

N

実 write イベントを取得しません。

●**rts_que_delay_wrt=Y|N**

～《N》

遅延書き込み（回数） イベントを取得するかどうかを指定します。

Y

遅延書き込み（回数） イベントを取得します。

N

遅延書き込み（回数） イベントを取得しません。

●**rts_que_delay_rec=Y|N**

～《N》

物理ファイル単位の遅延書き込み（レコード） イベントを取得するかどうかを指定します。

Y

物理ファイル単位の遅延書き込み（レコード） イベントを取得します。

N

物理ファイル単位の遅延書き込み（レコード） イベントを取得しません。

●**rts_que_delay_msg=Y|N**

～《N》

物理ファイル単位の遅延書き込み（メッセージ）イベントを取得するかどうかを指定します。

Y

物理ファイル単位の遅延書き込み（メッセージ）イベントを取得します。

N

物理ファイル単位の遅延書き込み（メッセージ）イベントを取得しません。

RPC 情報

●rts_rpc_rpc_call=Y|N

～《N》

RPC コール（同期応答型）イベントを取得するかどうかを指定します。

Y

RPC コール（同期応答型）イベントを取得します。

N

RPC コール（同期応答型）イベントを取得しません。

●rts_rpc_rpc_call_chained=Y|N

～《N》

RPC コール（連鎖 RPC 型）イベントを取得するかどうかを指定します。

Y

RPC コール（連鎖 RPC 型）イベントを取得します。

N

RPC コール（連鎖 RPC 型）イベントを取得しません。

●rts_rpc_usr_srvc=Y|N

～《N》

ユーザサービス実行イベントを取得するかどうかを指定します。

Y

ユーザサービス実行イベントを取得します。

N

ユーザサービス実行イベントを取得しません。

●rts_rpc_rpc_ovrtim=Y|N

～《N》

RPC タイムアウトイベントを取得するかどうかを指定します。

Y

RPC タイムアウトイベントを取得します。

N

RPC タイムアウトイベントを取得しません。

スケジュール情報

●rts_scd_scd_wait=Y|N

～《N》

スケジュール待ちイベントを取得するかどうかを指定します。

Y

スケジュール待ちイベントを取得します。

N

スケジュール待ちイベントを取得しません。

●rts_scd_schedule=Y|N

～《N》

スケジュールイベントを取得するかどうかを指定します。

Y

スケジュールイベントを取得します。

N

スケジュールイベントを取得しません。

●rts_scd_using_buf=Y|N

～《N》

メッセージ格納バッファプールの使用中サイズイベントを取得するかどうかを指定します。

Y

メッセージ格納バッファプールの使用中サイズイベントを取得します。

N

メッセージ格納バッファプールの使用中サイズイベントを取得しません。

●rts_scd_lack_buf=Y|N

～《N》

メッセージ格納バッファプールの不足で、スケジュールできなかったメッセージサイズイベントを取得するかどうかを指定します。

Y

メッセージ格納バッファプールの不足で、スケジュールできなかったメッセージサイズイベントを取得します。

N

メッセージ格納バッファプールの不足で、スケジュールできなかったメッセージサイズイベントを取得しません。

●rts_scd_scd_stay=Y|N

～《N》

スケジュール滞留イベントを取得するかどうかを指定します。

Y

スケジュール滞留イベントを取得します。

N

スケジュール滞留イベントを取得しません。

●rts_scd_svc_scd_wait=Y|N

～《N》

サービス単位のスケジュール待ちイベントを取得するかどうかを指定します。

Y

サービス単位のスケジュール待ちイベントを取得します。

N

サービス単位のスケジュール待ちイベントを取得しません。

●rts_scd_svc_using_buf=Y|N

～《N》

サービス単位のメッセージ格納バッファプールの使用中サイズイベントを取得するかどうかを指定します。

Y

サービス単位のメッセージ格納バッファプールの使用中サイズイベントを取得します。

N

サービス単位のメッセージ格納バッファプールの使用中サイズイベントを取得しません。

●rts_scd_parallel=Y|N

～《N》

同時実行サービス数イベントを取得するかどうかを指定します。

Y

同時実行サービス数イベントを取得します。

N

同時実行サービス数イベントを取得しません。

トランザクション情報

●rts_trn_commit=Y|N

～《N》

コミットイベントを取得するかどうかを指定します。

Y

コミットイベントを取得します。

N

コミットイベントを取得しません。

●rts_trn_rollback=Y|N

～《N》

ロールバックイベントを取得するかどうかを指定します。

Y

ロールバックイベントを取得します。

N

ロールバックイベントを取得しません。

●rts_trn_cmt_cmd=Y|N

～《N》

コマンドによるコミットイベントを取得するかどうかを指定します。

Y

コマンドによるコミットイベントを取得します。

N

コマンドによるコミットイベントを取得しません。

●rts_trn_rbk_cmd=Y|N

～《N》

コマンドによるロールバックイベントを取得するかどうかを指定します。

Y

コマンドによるロールバックイベントを取得します。

N

コマンドによるロールバックイベントを取得しません。

●**rts_trn_haz_cmd=Y|N**

～《N》

コマンドによるハザードイベントを取得するかどうかを指定します。

Y

コマンドによるハザードイベントを取得します。

N

コマンドによるハザードイベントを取得しません。

●**rts_trn_mix_cmd=Y|N**

～《N》

コマンドによるミックスイベントを取得するかどうかを指定します。

Y

コマンドによるミックスイベントを取得します。

N

コマンドによるミックスイベントを取得しません。

●**rts_trn_branch=Y|N**

～《N》

ブランチ実行時間イベントを取得するかどうかを指定します。

Y

ブランチ実行時間イベントを取得します。

N

ブランチ実行時間イベントを取得しません。

●**rts_trn_sync_point=Y|N**

～《N》

ブランチ同期点処理の実行時間イベントを取得するかどうかを指定します。

Y

ブランチ同期点処理の実行時間イベントを取得します。

N

ブランチ同期点処理の実行時間イベントを取得しません。

DAM 情報

●rts_dam_read=Y|N

～《N》

DAM 情報の read イベントを取得するかどうかを指定します。

Y

read イベントを取得します。

N

read イベントを取得しません。

●rts_dam_read_err=Y|N

～《N》

DAM 情報の read エラーイベントを取得するかどうかを指定します。

Y

read エラーイベントを取得します。

N

read エラーイベントを取得しません。

●rts_dam_write=Y|N

～《N》

DAM 情報の write イベントを取得するかどうかを指定します。

Y

write イベントを取得します。

N

write イベントを取得しません。

●rts_dam_write_err=Y|N

～《N》

DAM 情報の write エラーイベントを取得するかどうかを指定します。

Y

write エラーイベントを取得します。

N

write エラーイベントを取得しません。

●**rts_dam_fj=Y|N**

～《N》

FJ 出力回数イベントを取得するかどうかを指定します。

Y

FJ 出力回数イベントを取得します。

N

FJ 出力回数イベントを取得しません。

●**rts_dam_trn_branch=Y|N**

～《N》

同時実行 DAM トランザクションブランチ数イベントを取得するかどうかを指定します。

Y

同時実行 DAM トランザクションブランチ数イベントを取得します。

N

同時実行 DAM トランザクションブランチ数イベントを取得しません。

●**rts_dam_cache_block=Y|N**

～《N》

DAM キャッシュブロックの確保回数イベントを取得するかどうかを指定します。

Y

DAM キャッシュブロックの確保回数イベントを取得します。

N

DAM キャッシュブロックの確保回数イベントを取得しません。

●**rts_dam_shm_pool=Y|N**

～《N》

DAM キャッシュ用の共用メモリの使用率イベントを取得するかどうかを指定します。

Y

DAM キャッシュ用の共用メモリの使用率イベントを取得します。

N

DAM キャッシュ用の共用メモリの使用率イベントを取得しません。

TAM 情報

●rts_tam_real_renew=Y|N

～《N》

TAM ファイル実更新イベントを取得するかどうかを指定します。

Y

TAM ファイル実更新イベントを取得します。

N

TAM ファイル実更新イベントを取得しません。

●rts_tam_real_renew_time=Y|N

～《N》

TAM ファイル実更新時間イベントを取得するかどうかを指定します。

Y

TAM ファイル実更新時間イベントを取得します。

N

TAM ファイル実更新時間イベントを取得しません。

●rts_tam_rec_refer=Y|N

～《N》

コミット、ロールバック（レコード参照）イベントを取得するかどうかを指定します。

Y

コミット、ロールバック（レコード参照）イベントを取得します。

N

コミット、ロールバック（レコード参照）イベントを取得しません。

●rts_tam_rec_renew=Y|N

～《N》

コミット、ロールバック（レコード更新）イベントを取得するかどうかを指定します。

Y

コミット、ロールバック（レコード更新）イベントを取得します。

N

コミット，ロールバック（レコード更新）イベントを取得しません。

●**rts_tam_read=Y|N**

～《N》

TAM 情報の read イベントを取得するかどうかを指定します。

Y

read イベントを取得します。

N

read イベントを取得しません。

●**rts_tam_read_err=Y|N**

～《N》

TAM 情報の read エラーイベントを取得するかどうかを指定します。

Y

read エラーイベントを取得します。

N

read エラーイベントを取得しません。

●**rts_tam_write=Y|N**

～《N》

TAM 情報の write イベントを取得するかどうかを指定します。

Y

write イベントを取得します。

N

write イベントを取得しません。

●**rts_tam_write_err=Y|N**

～《N》

TAM 情報の write エラーイベントを取得するかどうかを指定します。

Y

write エラーイベントを取得します。

N

write エラーイベントを取得しません。

XA リソースサービス情報

●rts_xar_start=Y|N

～《N》

アプリケーションサーバから RAP サーバへのトランザクション開始要求の回数イベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクション開始要求の回数イベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクション開始要求の回数イベントを取得しません。

●rts_xar_start_err=Y|N

～《N》

アプリケーションサーバから RAP サーバへのトランザクション開始要求でのエラーイベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクション開始要求でのエラーイベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクション開始要求でのエラーイベントを取得しません。

●rts_xar_call=Y|N

～《N》

アプリケーションサーバから RAP サーバへのサービス要求の回数イベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのサービス要求の回数イベントを取得します。

N

アプリケーションサーバから RAP サーバへのサービス要求の回数イベントを取得しません。

●rts_xar_call_err=Y|N

～《N》

アプリケーションサーバから RAP サーバへのサービス要求でのエラーイベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクション開始要求のエラーイベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクション開始要求のエラーイベントを取得しません。

●**rts_xar_end=Y|N**

～《N》

アプリケーションサーバから RAP サーバへのトランザクション終了要求の回数イベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクション終了要求の回数イベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクション終了要求の回数イベントを取得しません。

●**rts_xar_end_err=Y|N**

～《N》

アプリケーションサーバから RAP サーバへのトランザクション終了要求でのエラーイベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクション終了要求でのエラーイベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクション終了要求でのエラーイベントを取得しません。

●**rts_xar_prepare=Y|N**

～《N》

アプリケーションサーバから RAP サーバへのトランザクションのプリペア処理要求の回数イベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクションのプリペア処理要求の回数イベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクションのプリペア処理要求の回数イベントを取得しません。

●**rts_xar_prepare_err=Y|N**

～《N》

アプリケーションサーバから RAP サーバへのトランザクションのプリペア処理要求でのエラーイベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクションのプリペア処理要求でエラーイベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクションのプリペア処理要求でのエラーイベントを取得しません。

●**rts_xar_commit=Y|N**

～《N》

アプリケーションサーバから RAP サーバへのトランザクションのコミット処理要求の回数イベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクションのコミット処理要求の回数イベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクションのコミット処理要求の回数イベントを取得しません。

●**rts_xar_commit_err=Y|N**

～《N》

アプリケーションサーバから RAP サーバへのトランザクションのコミット処理要求でのエラーイベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクションのコミット処理要求でのエラーイベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクションのコミット処理要求でのエラーイベントを取得しません。

●rts_xar_rollback=Y|N

～《N》

アプリケーションサーバから RAP サーバへのトランザクションのロールバック処理要求の回数イベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクションのロールバック処理要求の回数イベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクションのロールバック処理要求の回数イベントを取得しません。

●rts_xar_rollback_err=Y|N

～《N》

アプリケーションサーバから RAP サーバへのトランザクションのロールバック処理要求でのエラーイベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクションのロールバック処理要求でのエラーイベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクションのロールバック処理要求でのエラーイベントを取得しません。

●rts_xar_recover=Y|N

～《N》

アプリケーションサーバから RAP サーバへのプリペア済み、またはヒューリスティック完了したトランザクション通知要求の回数イベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのプリペア済み、またはヒューリスティック完了したトランザクション通知要求の回数イベントを取得します。

N

アプリケーションサーバから RAP サーバへのプリペア済み、またはヒューリスティック完了したトランザクション通知要求の回数イベントを取得しません。

●rts_xar_recover_err=Y|N

～《N》

アプリケーションサーバから RAP サーバへのプリペア済み、またはヒューリスティック完了したトランザクション通知要求でのエラーイベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのプリペア済み、またはヒューリスティック完了したトランザクション通知要求でのエラーイベントを取得します。

N

アプリケーションサーバから RAP サーバへのプリペア済み、またはヒューリスティック完了したトランザクション通知要求でのエラーイベントを取得しません。

●rts_xar_forget=Y|N

～《N》

アプリケーションサーバから RAP サーバへのトランザクション破棄要求の回数イベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクション破棄要求の回数イベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクション破棄要求の回数イベントを取得しません。

●rts_xar_forget_err=Y|N

～《N》

アプリケーションサーバから RAP サーバへのトランザクション破棄要求でのエラーイベントを取得するかどうかを指定します。

Y

アプリケーションサーバから RAP サーバへのトランザクション破棄要求でのエラーイベントを取得します。

N

アプリケーションサーバから RAP サーバへのトランザクション破棄要求でのエラーイベントを取得しません。

MCF 情報

●rts_mcf_ap_scd_stay=Y|N

～《N》

スケジュール待ち情報を取得するかどうかを指定します。

Y

スケジュール待ち情報を取得します。

N

スケジュール待ち情報を取得しません。

●**rts_mcf_ap_usr_srv=Y|N**

～《N》

ユーザサービス実行情報を取得するかどうかを指定します。

Y

ユーザサービス実行情報を取得します。

N

ユーザサービス実行情報を取得しません。

●**rts_mcf_in_msg_scd_wait=Y|N**

～《N》

論理端末単位に受信メッセージの処理待ち情報を取得するかどうかを指定します。

Y

論理端末単位に受信メッセージの処理待ち情報を取得します。

N

論理端末単位に受信メッセージの処理待ち情報を取得しません。

●**rts_mcf_out_msg_sync_scd_wait=Y|N**

～《N》

同期型送信メッセージの処理待ち情報を取得するかどうかを指定します。

Y

同期型送信メッセージの処理待ち情報を取得します。

N

同期型送信メッセージの処理待ち情報を取得しません。

●**rts_mcf_out_msg_resp_scd_wait=Y|N**

～《N》

問い合わせ応答型送信メッセージの処理待ち情報を取得するかどうかを指定します。

Y

問い合わせ応答型送信メッセージの処理待ち情報を取得します。

N

問い合わせ応答型送信メッセージの処理待ち情報を取得しません。

●**rts_mcf_out_msg_prio_scd_wait=Y|N**

～《N》

優先分岐型送信メッセージの処理待ち情報を取得するかどうかを指定します。

Y

優先分岐型送信メッセージの処理待ち情報を取得します。

N

優先分岐型送信メッセージの処理待ち情報を取得しません。

●**rts_mcf_out_msg_norm_scd_wait=Y|N**

～《N》

一般分岐型送信メッセージの処理待ち情報を取得するかどうかを指定します。

Y

一般分岐型送信メッセージの処理待ち情報を取得します。

N

一般分岐型送信メッセージの処理待ち情報を取得しません。

●**rts_mcf_que_scd_wait_num=Y|N**

～《N》

入力キューの滞留数の情報を取得するかどうかを指定します。

Y

入力キューの滞留数の情報を取得します。

N

入力キューの滞留数の情報を取得しません。

コマンド形式

なし。

注意事項

rtspout 定義コマンドや rtsstats コマンドで指定した対象では取得できない項目の定義に Y を指定した場合、統計情報を取得するための領域は確保しますが、リアルタイム統計情報は取得されません。

ユーザサービスデフォルト定義

形式

set 形式

```
[set nice=プロセスの優先順位の変更]
[set parallel_count=常駐プロセス数 [, 最大プロセス数] ]
[set hold=Y|N]
[set hold_recovery=Y|N]
[set deadlock_priority=デッドロックの優先順位]
[set schedule_priority=スケジュールの優先順位]
[set message_buflen=最大メッセージ長]
[set message_store_buflen=メッセージ格納バッファプール長]
[set trn_expiration_time=トランザクションブランチ限界経過時間]
[set trn_expiration_time_suspend=Y|N|F]
[set watch_next_chain_time=連鎖RPC間隔監視時間]
[set atomic_update=Y|N]
[set receive_from=queue|socket|none]
[set uap_trace_max=UAPトレース格納最大数]
[set uap_trace_file_put=Y|N]
[set term_watch_time=連続異常終了限界経過時間]
[set mcf_jnl_buff_size=MCFのジャーナルバッファの大きさ]
[set type=other|MHP]
[set balance_count=1プロセスが処理するサービス要求数]
[set uid=ユーザ識別子]
[set auto_restart=Y|N]
[set critical=Y|N]
[set lck_wait_priority=排他待ちの優先順位]
[set mcf_psv_id=アプリケーション起動プロセス識別子]
[set trn_cpu_time=トランザクションブランチCPU監視時間]
[set service_hold=Y|N]
[set service_priority_control=Y|N]
[set message_cell_size=スケジュールメッセージ格納セル長]
[set max_socket_msg=ソケット受信型サーバ受信メッセージ数の最大値]
[set max_socket_msglen=ソケット受信型サーバ受信メッセージ長の最大値]
[set trf_put=Y|N]
[set mcf_mgrid=アプリケーション起動プロセスの属するMCFマネジャ識別子]
[set mcf_service_max_count=MCF通信関数発行回数上限値]
[set trn_statistics_item=統計情報項目 [, 統計情報項目] ...]
[set node_down_restart=Y|N]
[set rpc_response_statistics=Y|N]
[set server_type="betran"|"xatmi"|"xatmi_cbl"]
[set trn_rm_open_close_scope=process|transaction]
[set trn_optimum_item=トランザクション最適化項目
    [, トランザクション最適化項目] ...]
[set purge_msgget=Y|N]
[set cancel_normal_terminate=Y|N]
[set prc_abort_signal=アボート用シグナル番号]
[set rpc_service_retry_count=サービスのリトライ回数の最大値]
[set rpc_extend_function=RPCサービスの機能拡張レベル]
[set max_socket_descriptors=ソケット用ファイル記述子の最大数]
[set max_open_fds=UAPプロセスでアクセスするファイルおよびパイプの最大数]
[set service_term_watch_time=連続サービス異常終了限界経過時間]
[set termed_after_service=Y|N]
[set xat_trn_expiration_time=同期点処理の限界経過監視時間]
```

```

[set xat_osi_usr=Y|N]
[set rpc_trace=Y|N]
[set rpc_trace_name="RPCトレースを取得するファイル名"]
[set rpc_trace_size=RPCトレースを取得するファイルのサイズ]
[set trn_rollback_information_put=no|self|remote|all]
[set schedule_method=msgque|namedpipe]
[set service_wait_time=ユーザサーバの非常駐プロセスのサービス
    要求待ち時間]

[set mcf_spp_oj=Y|N]
[set adm_message_option=メッセージ出力指定]
[set trn_watch_time=トランザクション同期点処理時の最大通信待ち時間]
[set trn_limit_time=トランザクションブランチ最大実行可能時間]
[set trn_rollback_response_receive=Y|N]
[set trn_partial_recovery_type=type1|type2|type3]
[set rpc_destination_mode=namdonly|namd|definition]
[set rpc_rap_auto_connect=Y|N]
[set rpc_rap_inquire_time=リモートAPI機能を使用して要求するサービス
    の問い合わせ間隔最大時間]

[set rpc_request_cancel_for_timeout=Y|N]
[set status_change_when_termining=Y|N]
[set service_expiration_time=サービス関数開始から終了までの
    実行監視時間]

[set multi_schedule=Y|N]
[set make_queue_on_starting=Y|N]
[set loadcheck_interval=負荷監視インタバル時間]
[set levelup_queue_count=U1,U2]
[set leveldown_queue_count=D0,D1]
[set ipc_sockctl_highwater=
    ソケットの一時クローズ処理開始数パーセンテージ
    [,ソケットの一時クローズ処理非対象数パーセンテージ] ]

[set ipc_sockctl_watchtime=ソケット再利用可能監視時間]
[set ipc_conn_interval=コネクション確立監視時間]
[set ipc_send_interval=データ送信監視間隔]
[set ipc_send_count=データ送信監視回数]
[set ipc_header_rcv_time=通信制御データの受信監視時間]
[set rpc_send_retry_count=TCP/IPコネクションの接続時にエラーが発生した
    場合のリトライ回数]
[set rpc_send_retry_interval=TCP/IPコネクションの接続時にエラーが発生し
    た場合のリトライ間隔]

[set ipc_rcvbuf_size=TCP/IPの受信バッファサイズ]
[set ipc_sndbuf_size=TCP/IPの送信バッファサイズ]
[set ipc_listen_sockbufset=Y|N]
[set polling_control_data=Y|N]
[set thread_yield_interval=ソケットの再利用指示を受信できる契機を
    与えるインタバル時間]

[set groups=グループ識別子 [,グループ識別子] ...]
[set loadlevel_message=Y|N|A]
[set ipc_backlog_count=コネクション確立要求を格納するキューの長さ]
[set rpc_buffer_pool_max=プーリングするバッファ数]
[set schedule_delay_limit=スケジュール遅延限界経過時間]
[set schedule_delay_abort=Y|N]
[set rap_autoconnect_con_error_msg=Y|N]
[set core_shm_suppress=Y|N]
[set xat_connect_resp_time=通信イベント処理用SPPのアソシエーション確立の
    最大応答待ち時間]
[set scd_poolfull_check_interval=KFCA00853-Eメッセージ出力インタバル
    時間]
[set scd_poolfull_check_count=KFCA00853-Eメッセージ出力判断値]

```

```

[set scd_pool_warning_use_rate=警告メッセージの出力判断値になるメッセージ
格納バッファプール使用率の上限値]
[set scd_pool_warning_interval=メッセージ格納バッファプール使用率超過時の
警告メッセージ出力インタバル時間]
[set ipc_tcpnodelay=Y|N]
[set stay_watch_queue_count=スケジュールキューの滞留監視判定を開始する際
の判断になるサービス要求滞留数]
[set stay_watch_check_rate=スケジュールキューの滞留監視判定処理で使用する
サービス要求の処理率]
[set stay_watch_abort=Y|N]
[set stay_watch_start_interval=スケジュールキューの滞留監視インタバル
時間]
[set stay_watch_check_interval=スケジュールキューの滞留監視判定インタバル
時間]
[set trn_completion_limit_time=トランザクション完了限界時間]
[set rap_message_id_change_level=メッセージIDの変更レベル]
[set log_audit_out_suppress=Y|N]
[set log_audit_message=監査ログを取得する項目のメッセージID]
[, 監査ログを取得する項目のメッセージID] ...]
[set mcf_prf_trace=Y|N]
[set scd_refresh_process=Y|N]
[set rap_extend_function=リモートAPI機能機能拡張レベル]
[set prc_coredump_filter=coreファイルに共有メモリを含めるかの判断値]
[set watch_time=最大応答待ち時間]

```

コマンド形式

```

[trnrmid -n リソースマネージャ名
-i リソースマネージャ拡張子 [, リソースマネージャ拡張子] ...
[-k] ]
[scdbufgrp -g スケジュールバッファグループ名
{ [-s メッセージ格納バッファ使用制限サイズ] |
[-p メッセージ格納バッファ使用制限率] }]
[scdmulti [-g マルチスケジュールラグループ名] ]
[scdsvcdef [-c サービス名]
[-p 同時実行可能なサービス数]
[-n キューイング可能なサービス要求数]
[-l キューイング可能なメッセージ格納バッファプール長] ]

```

putenv 形式

```

{{ [putenv 環境変数名 環境変数値] }}
[putenv XAT_CONNECT_RESP_TIME 通信イベント処理用SPPのアソシエーション
確立の最大応答待ち時間]

```

dcputenv 形式

```

{{ [dcputenv 環境変数名 環境変数値] }}

```

機能

ユーザサービス定義の省略時解釈値を定義します。

また、rap リスナーサービス定義の省略時解釈値を定義します。ただし、rap リスナーサービス定義に反映されないオペランドおよび指定値がありますので、詳細は「[rap リスナーサービス定義](#)」の注意事項を参照してください。

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》

ユーザサービス定義の nice オペランドの省略時解釈値を指定します。

●parallel_count=常駐プロセス数 [, 最大プロセス数]

～〈符号なし整数〉((0～1024))《1》

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

●hold=Y|N

～《Y》

ユーザサービス定義の hold オペランドの省略時解釈値を指定します。

●hold_recovery=Y|N

～《Y》

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

システム環境定義の start_scheduling_timing オペランドに BEFORE を指定した場合、このオペランドの指定に関係なく、閉塞状態は引き継がれません。閉塞状態を引き継ぐ場合は、スケジュールサービス定義の scd_hold_recovery オペランドに F を指定してください。各オペランド指定による閉塞状態の引き継ぎについては、スケジュールサービス定義の [scd_hold_recovery](#) オペランドを参照してください。

●deadlock_priority=デッドロックの優先順位

～〈符号なし整数〉((1～127))《64》

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

●schedule_priority=スケジュールの優先順位

～〈符号なし整数〉((1～16))《8》

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

●message_buflen=最大メッセージ長

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

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

●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)

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

●trn_expiration_time=トランザクションブランチ限界経過時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●trn_expiration_time_suspend=Y|N|F

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

このオペランドと各種タイマ監視との関係については、「[付録 A.2 トランザクションの時間監視](#)」を参照してください。

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●watch_next_chain_time=連鎖 RPC 間隔監視時間

～ 〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)

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

このオペランドと各種タイマ監視との関係については、「[付録 A.2 トランザクションの時間監視](#)」を参照してください。

●atomic_update=Y|N

～ 《Y》

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

●receive_from=queue|socket|none

～ 《queue》

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

●uap_trace_max=UAP トレース格納最大数

～ 〈符号なし整数〉 ((0～4095)) 《32》

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

●uap_trace_file_put=Y|N

ユーザサービス定義の uap_trace_file_put オペランドの省略時解釈値を指定します。ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●term_watch_time=連続異常終了限界経過時間

～ 〈符号なし整数〉 ((0～32767)) 《30》 (単位：分)

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

●mcf_jnl_buff_size=MCF のジャーナルバッファの大きさ

～ 〈符号なし整数〉 ((4096～131072))) (単位：バイト)

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

●type=other|MHP

～ 《other》

ユーザサービス定義の type オペランドの省略時解釈値を指定します。

●balance_count=1 プロセスが処理するサービス要求数

～ 〈符号なし整数〉 ((0～512)) 《3》

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

●uid=ユーザ識別子

～ 〈符号なし整数〉 ((0～4294967294))

ユーザサービス定義の uid オペランドの省略時解釈値を指定します。

省略した場合、OpenTP1 管理者のユーザ ID を仮定します。

最大値は OS に依存するので、使用している OS のマニュアルを参照してください。

●auto_restart=Y|N

～ 《N》

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

●critical=Y|N

～ 《N》

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

●**lck_wait_priority=排他待ちの優先順位**

～ 〈符号なし整数〉 ((0～127)) 《0》

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

●**mcf_psv_id=アプリケーション起動プロセス識別子**

～ 〈16 進数字〉 ((01～ff))

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

●**trn_cpu_time=トランザクションブランチ CPU 監視時間**

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●**service_hold=Y|N**

～ 《N》

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

●**service_priority_control=Y|N**

～ 《N》

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

●**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)

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

●**max_socket_msg=ソケット受信型サーバ受信メッセージ数の最大値**

～ 〈符号なし整数〉 ((1～500)) 《100》

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

●**max_socket_msglen=ソケット受信型サーバ受信メッセージ長の最大値**

～ 〈符号なし整数〉 ((1～30270)) 《10240》 (単位：キロバイト)

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

●trf_put=Y|N

～《N》

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

●mcf_mgrid=アプリケーション起動プロセスの属する MCF マネジャ識別子

～〈識別子〉((A～Z, a～z))《A》

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

●mcf_service_max_count=MCF 通信関数発行回数上限値

～〈符号なし整数〉((0～65535))

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

●trn_statistics_item=統計情報項目〔, 統計情報項目〕…

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●node_down_restart=Y|N

～《Y》

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

●rpc_response_statistics=Y|N

～《N》

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

●server_type="betran"|"xatmi"|"xatmi_cbl"

～《"betran"》

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

●trn_rm_open_close_scope=process|transaction

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●trn_optimum_item=トランザクション最適化項目〔, トランザクション最適化項目〕…

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●**purge_msgget=Y|N**

～《N》

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

●**cancel_normal_terminate=Y|N**

～《N》

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

●**prc_abort_signal=アボート用シグナル番号**

～〈符号なし整数〉((1～128))《3》

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

●**rpc_service_retry_count=サービスのリトライ回数の最大値**

～〈符号なし整数〉((0～65535))《0》

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

●**rpc_extend_function=RPC サービスの機能拡張レベル**

～〈16 進数字〉((00000000～0000000F))《00000000》

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

●**max_socket_descriptors=ソケット用ファイル記述子の最大数**

～〈符号なし整数〉((32～2032))

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●**max_open_fds=UAP プロセスでアクセスするファイルおよびパイプの最大数**

～〈符号なし整数〉((16～2016))《50》

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

●**service_term_watch_time=連続サービス異常終了限界経過時間**

～〈符号なし整数〉((0～32767))《0》(単位：分)

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

●**termed_after_service=Y|N**

～《N》

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

●**xat_trn_expiration_time=同期点処理の限界経過監視時間**

～〈符号なし整数〉((1～2147483647))《180》(単位：秒)

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

●**xat_osi_usr=Y|N**

～《N》

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

●**rpc_trace=Y|N**

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

ここで指定を省略し、rap リスナーサービス定義およびユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●**rpc_trace_name="RPC トレースを取得するファイル名"**

～〈パス名〉

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

パス名のうち、RPC トレースを取得するファイル名（デフォルトは rpctr）の最大長は、13 文字です。

パス名に環境変数を指定する場合、パス名の先頭に環境変数を指定してください（指定例：\$DCDIR/tmp/ファイル名）。

ここで指定を省略し、rap リスナーサービス定義およびユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●**rpc_trace_size=RPC トレースを取得するファイルのサイズ**

～〈符号なし整数〉((1024～2147483648)) (単位：バイト)

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

ここで指定を省略し、rap リスナーサービス定義およびユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●**trn_rollback_information_put=no|self|remote|all**

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●**schedule_method=msgque|namedpipe**

～《msgque》

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

●**service_wait_time=ユーザサーバの非常駐プロセスのサービス要求待ち時間**

～〈符号なし整数〉((1～4096)) (単位：秒)

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

●**mcf_spp_oj=Y|N**

～《Y》

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

●**adm_message_option=メッセージ出力指定**

～〈1けたの16進数〉《F》

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

●**trn_watch_time=トランザクション同期点処理時の最大通信待ち時間**

～〈符号なし整数〉((1～65535)) (単位：秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。さらに、トランザクションサービス定義でも省略した場合、トランザクションを行ったユーザサーバの watch_time の値を仮定します。ただし、watch_time に 0 を指定した場合は 120 秒を仮定します。

●**trn_limit_time=トランザクションブランチ最大実行可能時間**

～〈符号なし整数〉((0～65535)) (単位：秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●**trn_rollback_response_receive=Y|N**

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●trn_partial_recovery_type=type1|type2|type3

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●rpc_destination_mode=namdonly|namd|definition

～《definition》

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

●rpc_rap_auto_connect=Y|N

～《Y》

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

●rpc_rap_inquire_time=リモート API 機能を使用して要求するサービスの問い合わせ間隔最大時間

～〈符号なし整数〉((0～1048575))《0》(単位：秒)

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

●rpc_request_cancel_for_timeout=Y|N

～《Y》

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

●status_change_when_termining=Y|N

～《Y》

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

●service_expiration_time=サービス関数開始から終了までの実行監視時間

～〈符号なし整数〉((0～65535))《0》(単位：秒)

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

●multi_schedule=Y|N

～《N》

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

●make_queue_on_starting=Y|N

～《N》

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

●loadcheck_interval=負荷監視インタバル時間

～ 〈符号なし整数〉 ((0～65535))

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

●levelup_queue_count=U1,U2

～ 〈符号なし整数〉 ((0～32767))

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

●leveldown_queue_count=D0,D1

～ 〈符号なし整数〉 ((0～32767))

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

U1

サーバの負荷レベルが LEVEL1 に上がったと判断するサービス要求滞留数

U2

サーバの負荷レベルが LEVEL2 に上がったと判断するサービス要求滞留数

D0

サーバの負荷レベルが LEVEL0 に下がったと判断するサービス要求滞留数

D1

サーバの負荷レベルが LEVEL1 に下がったと判断するサービス要求滞留数

●ipc_sockctl_highwater=ソケットの一時クローズ処理開始数パーセンテージ [ソケットの一時クローズ処理非対象数パーセンテージ]

～ 〈符号なし整数〉 ((0～100)) 《100, 0》

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_sockctl_watchtime=ソケット再利用可能監視時間

～ 〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_conn_interval=コネクション確立監視時間

～ 〈符号なし整数〉 ((8～65535)) 《8》 (単位：秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_send_interval=データ送信監視間隔

～ 〈符号なし整数〉 ((5～32767)) 《5》 (単位：秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_send_count=データ送信監視回数

～ 〈符号なし整数〉 ((1～32767)) 《5》

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_header_recv_time=通信制御データの受信監視時間

～ 〈符号なし整数〉 ((5～32767)) 《10》 (単位：秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●rpc_send_retry_count=TCP/IP コネクションの接続時にエラーが発生した場合のリトライ回数

～ 〈符号なし整数〉 ((0～65535)) 《0》

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●rpc_send_retry_interval=TCP/IP コネクションの接続時にエラーが発生した場合のリトライ間隔

～ 〈符号なし整数〉 ((0～300000)) 《0》 (単位：ミリ秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_listen_sockbufset=Y|N

～ 《N》

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

●**ipc_rcvbuf_size=TCP/IP の受信バッファサイズ**

～ 〈符号なし整数〉 ((0, 8192～1048576)) 《8192》 (単位：バイト)

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

●**ipc_sndbuf_size=TCP/IP の送信バッファサイズ**

～ 〈符号なし整数〉 ((0, 8192～1048576)) 《8192》 (単位：バイト)

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

●**polling_control_data=Y|N**

～ 《N》

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

●**thread_yield_interval=ソケットの再利用指示を受信できる契機を与えるインタバル時間**

～ 〈符号なし整数〉 ((1～86400)) 《90》 (単位：秒)

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

●**groups=グループ識別子 [グループ識別子]**

～ 〈符号なし整数〉 ((0～4294967294))

ユーザサービス定義の groups オペランドの省略時解釈値を指定します。

最大値は OS に依存するので、使用している OS のマニュアルを参照してください。

●**loadlevel_message=Y|N|A**

～ 《N》

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

●**ipc_backlog_count=コネクション確立要求を格納するキューの長さ**

～ 〈符号なし整数〉 ((0～4096)) 《0》

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

●**rpc_buffer_pool_max=プーリングするバッファ数**

～ 〈符号なし整数〉 ((1～64)) 《64》

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

●**schedule_delay_limit=スケジュール遅延限界経過時間**

～ 〈符号なし整数〉 ((0～32767)) 《0》 (単位：秒)

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

●**`schedule_delay_abort=Y|N`**

～《N》

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

●**`rap_autoconnect_con_error_msg=Y|N`**

～《Y》

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

●**`core_shm_suppress=Y|N`**

～《N》

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

●**`xat_connect_resp_time=通信イベント処理用 SPP のアソシエーション確立の最大応答待ち時間`**

～〈符号なし整数〉((0～65535))《180》(単位：秒)

ユーザサービス定義の set 形式の `xat_connect_resp_time` オペランドの省略時解釈値を指定します。このオペランドと `putenv` 形式の `XAT_CONNECT_RESP_TIME` オペランドの両方を指定した場合は、このオペランドの指定が優先され、`putenv` 形式の `XAT_CONNECT_RESP_TIME` オペランドの指定は無視されます。

●**`scd_poolfull_check_interval=KFCA00853-E メッセージ出力インタバル時間`**

～〈符号なし整数〉((0～32767))《0》(単位：秒)

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

●**`scd_poolfull_check_count=KFCA00853-E メッセージ出力判断値`**

～〈符号なし整数〉((1～32767))《10》(単位：回)

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

●**`scd_pool_warning_use_rate=警告メッセージの出力判断値になるメッセージ格納バッファプール使用率の上限値`**

～〈符号なし整数〉((0～99))《0》(単位：%)

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

●**`scd_pool_warning_interval=メッセージ格納バッファプール使用率超過時の警告メッセージ出力インタバル時間`**

～〈符号なし整数〉((0～32767))《0》(単位：秒)

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

●`ipc_tcpnodelay=Y|N`

～《Y》

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

●`stay_watch_queue_count=スケジュールキューの滞留監視判定を開始する際の判断になるサービス要求滞留数`

～〈符号なし整数〉((0～32767))《0》

ユーザサービス定義の `stay_watch_queue_count` オペランドの省略時解釈値を指定します。ここで 0 を指定または指定を省略し、ユーザサービス定義でも指定を省略した場合、スケジュールキューの滞留監視をしません。

●`stay_watch_check_rate=スケジュールキューの滞留監視判定処理で使用するサービス要求の処理率`

～〈符号なし整数〉((1～100)) (単位：%)

ユーザサービス定義の `stay_watch_check_rate` オペランドの省略時解釈値を指定します。このオペランドにデフォルト値はありません。ユーザサービス定義を省略する場合は、このオペランドを必ず指定してください。指定を省略した場合、定義不正のためサーバ起動処理が中止されます。

●`stay_watch_abort=Y|N`

～《N》

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

●`stay_watch_start_interval=スケジュールキューの滞留監視インタバル時間`

～〈符号なし整数〉((1～32767))《10》 (単位：秒)

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

●`stay_watch_check_interval=スケジュールキューの滞留監視判定インタバル時間`

～〈符号なし整数〉((1～65534))《10》 (単位：秒)

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

●`trn_completion_limit_time=トランザクション完了限界時間`

～〈符号なし整数〉((0～65535)) (単位：秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●rap_message_id_change_level=メッセージ ID の変更レベル

～ 〈符号なし整数〉 ((0～2)) 《0》

ユーザサービス定義または rap リスナーサービス定義の rap_message_id_change_level オペランドの省略時解釈値を指定します。このオペランドの指定値と出力されるメッセージの対応については rap リスナーサービス定義の rap_message_id_change_level オペランドを参照してください。

●log_audit_out_suppress=Y|N

～ 《N》

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

●log_audit_message=監査ログを取得する項目のメッセージ ID [監査ログを取得する項目のメッセージ ID] ...

～ 〈符号なし整数〉 ((33400～99999))

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

●mcf_prf_trace=Y|N

～ 《Y》

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

●scd_refresh_process=Y|N

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

●rap_extend_function=リモート API 機能の機能拡張レベル

～ 〈16 進数字〉 ((00000000～00000001)) 《00000000》

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

●prc_coredump_filter=core ファイルに共有メモリを含めるかの判断値

～ 〈符号なし整数〉 ((1, 3))

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

●watch_time=最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

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

ここで指定を省略し、ユーザサービス定義でも省略した場合、システム共通定義の値を仮定します。

コマンド形式

次のページに記述しています。

putenv 形式のオペランド

●環境変数名 環境変数値

～ 〈文字列〉

ユーザサービス定義の環境変数名の省略時解釈値を指定します。

●XAT_CONNECT_RESP_TIME 通信イベント処理用 SPP のアソシエーション確立の最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) 《180》 (単位：秒)

ユーザサービス定義の putenv 形式の XAT_CONNECT_RESP_TIME オペランドの省略時解釈値を指定します。このオペランドと set 形式の xat_connect_resp_time オペランドの両方を指定した場合は、このオペランドの指定は無視され、set 形式の xat_connect_resp_time オペランドの指定が優先されます。

dcputenv 形式のオペランド

●環境変数名 環境変数値

～ 〈文字列〉

ユーザサービス定義の環境変数名の省略時解釈値を指定します。

trnrmid

名称

リソースマネージャ拡張子の指定

形式

```
[trnrmid -n リソースマネージャ名  
-i リソースマネージャ拡張子 [, リソースマネージャ拡張子] ...  
[-k] ]
```

機能

ユーザサービス定義の trnrmid 定義コマンドの省略時解釈値を指定します。

オプション

●-n リソースマネージャ名

～ 〈1～31 文字の識別子〉

ユーザサービス定義のリソースマネージャ名の省略時解釈値を指定します。

●-i リソースマネージャ拡張子

～ 〈1～2 文字の識別子〉

ユーザサービス定義のリソースマネージャ拡張子の省略時解釈値を指定します。

●-k

ユーザサービス定義の省略時解釈値を指定します。

注意事項

ユーザサービス定義とユーザサービスデフォルト定義の両方で、trnrmid 定義コマンドに-i オプションを指定する場合は注意してください。例えば、次のように指定した場合、ユーザサービス定義とユーザサービスデフォルト定義の両方の指定が有効となります。

- ・ ユーザサービス定義の指定：trnrmid -n RM 名称 -i s1
- ・ ユーザサービスデフォルト定義の指定：trnrmid -n RM 名称 -i s2

scdbufgrp

名称

スケジュールバッファグループの指定

形式

```
scdbufgrp -g スケジュールバッファグループ名  
           { [-s メッセージ格納バッファ使用制限サイズ] |  
             [-p メッセージ格納バッファ使用制限率] }
```

機能

ユーザサービス定義の scdbufgrp 定義コマンドの省略時解釈値を指定します。

オプション

●-g スケジュールバッファグループ名

～ 〈1～8 文字の識別子〉

ユーザサービス定義のスケジュールバッファグループ名の省略時解釈値を指定します。

●-s メッセージ格納バッファ使用制限サイズ

～ 〈符号なし整数〉 ((512～1610612736)) (単位：バイト)

ユーザサービス定義のメッセージ格納バッファ使用制限サイズの省略時解釈値を指定します。

●-p メッセージ格納バッファ使用制限率

～ 〈符号なし整数〉 ((1～100)) (単位：%)

ユーザサービス定義のメッセージ格納バッファ使用制限率の省略時解釈値を指定します。

注意事項

ユーザサービスデフォルト定義に scdbufgrp 定義コマンドを指定した場合、すべてのユーザサービス定義の message_store_bufllen オペランドの指定値（メッセージ格納バッファプール長）が無視されます。

ユーザサービス定義の message_store_bufllen オペランドとユーザサービスデフォルト定義の指定とメッセージ格納バッファプール長の関係については、ユーザサービス定義の [scdbufgrp](#) 定義コマンドを参照してください。

scdmulti

名称

マルチスケジューラ機能の指定

形式

scdmulti [-g マルチスケジューラグループ名]

機能

ユーザサービス定義の scdmulti 定義コマンドの省略時解釈値を指定します。

オプション

- g マルチスケジューラグループ名
 ～ 〈1～8 文字の識別子〉《scdmltgp》

ユーザサービス定義のマルチスケジューラグループ名の省略時解釈値を指定します。

名称

サービス単位にスケジュールサービスの動作を指定

形式

```
[scdsvcdef [-c サービス名]
            [-p 同時実行可能なサービス数]
            [-n キューイング可能なサービス要求数]
            [-l キューイング可能なメッセージ格納バッファプール長] ]
```

機能

ユーザサービス定義の scdsvcdef 定義コマンドの省略時解釈値を指定します。

オプション

●-c サービス名

～ 〈1～31 文字の識別子〉

ユーザサービス定義のサービス名の省略時解釈値を指定します。

●-p 同時実行可能なサービス数

～ 〈符号なし整数〉 ((1～1024))

ユーザサービス定義の同時実行可能なサービス数の省略時解釈値を指定します。

●-n キューイング可能なサービス要求数

～ 〈符号なし整数〉 ((1～65535))

ユーザサービス定義のキューイング可能なサービス要求数の省略時解釈値を指定します。

●-l キューイング可能なメッセージ格納バッファプール長

～ 〈符号なし整数〉 ((512～31457280)) (単位：バイト)

ユーザサービス定義のキューイング可能なメッセージ格納バッファプール長の省略時解釈値を指定します。

ユーザサービス定義

形式

set 形式

```
set service_group="サービスグループ名"
set module="実行形式プログラム名"
set service="サービス名=エントリポイント名 [;UAP共用ライブラリ名] "
    [, "サービス名=エントリポイント名 [;UAP共用ライブラリ名] " ] ...
[set nice=プロセスの優先順位の変更]
[set parallel_count=常駐プロセス数 [, 最大プロセス数] ]
[set hold=Y|N]
[set hold_recovery=Y|N]
[set deadlock_priority=デッドロックの優先順位]
[set schedule_priority=スケジュールの優先順位]
[set message_buflen=最大メッセージ長]
[set message_store_buflen=メッセージ格納バッファプール長]
[set trn_expiration_time=トランザクションブランチ限界経過時間]
[set trn_expiration_time_suspend=Y|N|F]
[set watch_next_chain_time=連鎖RPC間隔監視時間]
[set atomic_update=Y|N]
[set receive_from=queue|socket|none]
[set uap_trace_max=UAPトレース格納最大数]
[set uap_trace_file_put=Y|N]
[set term_watch_time=連続異常終了限界経過時間]
[set mcf_jnl_buff_size=MCFのジャーナルバッファの大きさ]
[set type=other|MHP]
[set balance_count=1プロセスが処理するサービス要求数]
[set uid=ユーザ識別子]
[set auto_restart=Y|N]
[set critical=Y|N]
[set lck_wait_priority=排他待ちの優先順位]
[set mcf_psv_id=アプリケーション起動プロセス識別子]
[set trn_cpu_time=トランザクションブランチCPU監視時間]
[set service_hold=Y|N]
[set service_priority_control=Y|N]
[set message_cell_size=スケジュールメッセージ格納セル長]
[set max_socket_msg=ソケット受信型サーバ受信メッセージ数の最大値]
[set max_socket_msglen=ソケット受信型サーバ受信メッセージ長の最大値]
[set trf_put=Y|N]
[set mcf_mgrid=アプリケーション起動プロセスの属するMCFマネジャ識別子]
[set mcf_service_max_count=MCF通信関数発行回数上限値]
[set trn_statistics_item=統計情報項目 [, 統計情報項目] ...]
[set node_down_restart=Y|N]
[set rpc_response_statistics=Y|N]
[set server_type="betran"|"xatmi"|"xatmi_cbl"]
[set trn_rm_open_close_scope=process|transaction]
[set trn_optimum_item=トランザクション最適化項目
    [, トランザクション最適化項目] ...]
[set purge_msgsget=Y|N]
[set cancel_normal_terminate=Y|N]
[set prc_abort_signal=アボート用シグナル番号]
[set rpc_service_retry_count=サービスのリトライ回数の最大値]
[set rpc_extend_function=RPCサービスの機能拡張レベル]
[set max_socket_descriptors=ソケット用ファイル記述子の最大数]
```

```

[set max_open_fds=UAPプロセスでアクセスするファイルおよびパイプの最大数]
[set service_term_watch_time=連続サービス異常終了限界経過時間]
[set termed_after_service=Y|N]
[set xat_trn_expiration_time=同期点処理の限界経過監視時間]
[set xat_osi_usr=Y|N]
[set rpc_trace=Y|N]
[set rpc_trace_name="RPCトレースを取得するファイル名"]
[set rpc_trace_size=RPCトレースを取得するファイルのサイズ]
[set trn_rollback_information_put=no|self|remote|all]
[set schedule_method=msgque|namedpipe]
[set service_wait_time=ユーザサーバの非常駐プロセスのサービス
                        要求待ち時間]

[set mcf_spp_oj=Y|N]
[set adm_message_option=メッセージ出力指定]
[set trn_watch_time=トランザクション同期点処理時の最大通信待ち時間]
[set trn_limit_time=トランザクションブランチ最大実行可能時間]
[set trn_rollback_response_receive=Y|N]
[set trn_partial_recovery_type=type1|type2|type3]
[set rpc_destination_mode=namdonly|namd|definition]
[set rpc_rap_auto_connect=Y|N]
[set rpc_rap_inquire_time=リモートAPI機能を使用して要求するサービスの
                        問い合わせ間隔最大時間]

[set rpc_request_cancel_for_timeout=Y|N]
[set status_change_when_termining=Y|N]
[set service_expiration_time=サービス関数開始から終了までの
                        実行監視時間]

[set multi_schedule=Y|N]
[set make_queue_on_starting=Y|N]
[set loadcheck_interval=負荷監視インタバル時間]
[set levelup_queue_count=U1,U2]
[set leveldown_queue_count=D0,D1]
[set ipc_sockctl_highwater=
                        ソケットの一時クローズ処理開始数パーセンテージ
                        [, ソケットの一時クローズ処理非対象数パーセンテージ] ]

[set ipc_sockctl_watchtime=ソケット再利用可能監視時間]
[set ipc_conn_interval=コネクション確立監視時間]
[set ipc_send_interval=データ送信監視間隔]
[set ipc_send_count=データ送信監視回数]
[set ipc_header_rcv_time=通信制御データの受信監視時間]
[set rpc_send_retry_count=TCP/IPコネクションの接続時にエラーが発生した
                        場合のリトライ回数]
[set rpc_send_retry_interval=TCP/IPコネクションの接続時にエラーが発生し
                        た場合のリトライ間隔]

[set ipc_rcvbuf_size=TCP/IPの受信バッファサイズ]
[set ipc_sndbuf_size=TCP/IPの送信バッファサイズ]
[set ipc_listen_sockbufset=Y|N]
[set polling_control_data=Y|N]
[set thread_yield_interval=ソケットの再利用指示を受信できる契機を
                        与えるインタバル時間]

[set groups=グループ識別子 [, グループ識別子] ...]
[set loadlevel_message=Y|N|A]
[set ipc_backlog_count=コネクション確立要求を格納するキューの長さ]
[set rpc_buffer_pool_max=プーリングするバッファ数]
[set schedule_delay_limit=スケジュール遅延限界経過時間]
[set schedule_delay_abort=Y|N]
[set rap_autoconnect_con_error_msg=Y|N]
[set core_shm_suppress=Y|N]
[set xat_connect_resp_time=通信イベント処理用SPPのアソシエーション確立の

```

```

                                最大応答待ち時間]
[set scd_poolfull_check_interval=KFCA00853-Eメッセージ出力インタバル
                                時間]
[set scd_poolfull_check_count=KFCA00853-Eメッセージ出力判断値]
[set scd_pool_warning_use_rate=警告メッセージの出力判断値になるメッセージ
                                格納バッファプール使用率の上限値]
[set scd_pool_warning_interval=メッセージ格納バッファプール使用率超過時の
                                警告メッセージ出力インタバル時間]
[set ipc_tcpnodelay=Y|N]
[set stay_watch_queue_count=スケジュールキューの滞留監視判定を開始する際
                                の判断になるサービス要求滞留数]
[set stay_watch_check_rate=スケジュールキューの滞留監視判定処理で使用する
                                サービス要求の処理率]
[set stay_watch_abort=Y|N]
[set stay_watch_start_interval=スケジュールキューの滞留監視インタバル
                                時間]
[set stay_watch_check_interval=スケジュールキューの滞留監視判定インタバル
                                時間]
[set trn_completion_limit_time=トランザクション完了限界時間]
[set rap_message_id_change_level=メッセージIDの変更レベル]
[set log_audit_out_suppress=Y|N]
[set log_audit_message=監査ログ取得メッセージID]
                                [, 監査ログ取得メッセージID] ...]
[set mcf_prf_trace=Y|N]
[set scd_refresh_process=Y|N]
[set rap_extend_function=リモートAPI機能機能拡張レベル]
[set prc_coredump_filter=coreファイルに共有メモリを含めるかの判断値]
[set watch_time=最大応答待ち時間]

```

コマンド形式

```

[trnrmid  -n   リソースマネージャ名
          -i   リソースマネージャ拡張子 [, リソースマネージャ拡張子] ...
          [-k] ]
[scdbufgrp  -g   スケジュールバッファグループ名
             { [-s   メッセージ格納バッファ使用制限サイズ] |
               [-p   メッセージ格納バッファ使用制限率] }]
[scdmulti  [-g   マルチスケジュールラグループ名] ]
[scdsvcdef  [-c   サービス名]
             [-p   同時実行可能なサービス数]
             [-n   キューイング可能なサービス要求数]
             [-l   キューイング可能なメッセージ格納バッファプール長] ]

```

putenv 形式

```

{{ [putenv 環境変数名 環境変数値] }}
[putenv  DCFPL_CONNECT_RETRY_COUNT   コネクション確立処理をリトライする
                                         回数]
[putenv  DCFPL_CONNECT_RETRY_INTERVAL コネクション確立処理のリトライ
                                         間隔]
[putenv  XAT_CONNECT_RESP_TIME       通信イベント処理用SPPのアソシエーション
                                         確立の最大応答待ち時間]

```

dcputenv 形式

```
{{ [dcputenv 環境変数名 環境変数値] }}
```

機能

ユーザサーバの実行環境をユーザサーバごとに定義します。

RPC でサービスを要求する場合は、「サービスグループ名+サービス名」を指定します。サービスグループ名は UAP (SPP, MHP) の実行形式プログラムに対応します。また、サービス名は実行形式プログラムを構成する個々の関数 (サービス関数) に対応します。

OpenTP1 は、指定されたサービスグループ名ごとに対応するスケジュールキューへサービス要求を登録します。

ユーザサービス定義の項目を省略した場合は、ユーザサービスデフォルト定義の値を省略時解釈値とします。

ユーザサービス定義のファイル名がユーザサーバ名になります。

このユーザサーバ名をユーザサービス構成定義で指定します。

説明

set 形式のオペランド

●service_group="サービスグループ名"

～ 〈1～31 文字の識別子〉

サービスグループ名を指定します。

サービスグループ名は、ネットワークで接続されるすべての OpenTP1 システムの中で一意になるように指定してください。

SUP の場合は、指定する必要はありません。

●module="実行形式プログラム名"

～ 〈1～14 文字の識別子〉

このサービスグループを実行する実行形式プログラム名を指定します。

この実行形式プログラムはプロセスサービス定義で指定したディレクトリにあります。

OS が HP-UX の場合に、リンケージの際のバインドモードが"immediate"でない実行形式ファイルを指定すると、OpenTP1 の動作は保証しません。アプリケーションのバインドモードが"immediate"かどうかは、OS の chatr コマンドを使用して確認してください。

●service="サービス名=エントリポイント名 [UAP 共用ライブラリ名] " [, "サービス名=エントリポイント名 [UAP 共用ライブラリ名] ") ...

サービス名, およびエントリポイント名 ～〈1～31 文字の識別子〉

このサービスグループに属するサービス名とそのサービスを提供するエントリポイント名の組をすべてのサービスについて指定します。なお、サービス名とエントリポイント名の間にある「=」の前後に、空白、タブコードなどを指定しないでください。

エントリポイント名とは、C の関数名であり、COBOL のプログラム名、または入り口名のことです。スタブを使うサービスの場合は、エントリポイント名に RPC インタフェース定義に指定する名称と同じ指定をします。

サービス名とエントリポイント名は、1 対 1 で対応させてください。複数のサービス名に一つのエントリポイント名を対応させることはできません。なお、「サービス名=エントリポイント名」の指定回数に上限はありません。

UAP 共用ライブラリ名 ～〈1～255 文字のパス名〉

サービス関数動的ローディング機能を使うサービスの場合、UAP 共用ライブラリのパス名を指定します。UAP 共用ライブラリ名に、空白、タブコードなどを指定しないでください。

なお、定義チェックでは、UAP 共用ライブラリ名に空白やタブコードなどを指定していないかどうかだけが対象とされます。空白やタブコードを指定していないときは、無条件に UAP 共用ライブラリのローディングが実行されます。

サービス関数動的ローディング機能を使う場合は、RPC インタフェース定義にはエントリポイント名の指定は不要です。UAP 共用ライブラリを省略した場合、スタブを使うサービスとします。

AIX 版 OpenTP1 で、サービス関数動的ローディング機能を使う場合、SPP のリンケージライブラリには libbetran2.a を、MHP のリンケージライブラリには libbetran2.a および libmcf2.a を使用してください。これらのライブラリが使用されなかった場合は、動作を保証しません。ただし、AIX 版の uCosminexus TP1/Server Base(64)の場合には、libbetran.a と libbetran2.a が使用できます。

- スタブだけを使うサービスの場合

次の指定例のとおり指定した場合、サービスは RPC インタフェース定義に指定したエントリポイント名の順で検索されます。

指定例：

```
set service = "serviceA=serviceA"
```

- サービス関数動的ローディング機能だけを使うサービスの場合

このオペランドにサービス関数動的ローディング機能を使うサービスを指定した場合、サービスはこのオペランドに指定したサービスの順で検索されます。

- UAP ライブラリに環境変数を使用する場合

指定例：

```
set service = "serviceA=serviceA; $USRDIR/lib/usrlib.so"
putenv USRDIR /OLTP1/usrdir
```

UAP 共用ライブラリ名のパス名に環境変数を指定する場合、パス名の先頭だけ指定できます。また、ここで指定する環境変数は、ユーザサービスデフォルト定義またはユーザサービス定義に指定できます。

このオペランドに限り、使用する環境変数がこのオペランドの指定値より下位に設定されていても有効となります。なお、環境変数を指定した場合は、環境変数展開後のパス名で最大 255 文字指定できます。

- UAP ライブラリに名称を絶対パスで指定する場合

指定例：

```
set service = "serviceA=serviceA;/OLTP1/usrdir/lib/usrlib.so"
```

UAP 共用ライブラリ名に、UAP 共用ライブラリ名称の絶対パスを指定した場合、OpenTP1 は指定した UAP 共用ライブラリを検索して使用します。

- UAP ライブラリにライブラリ名称だけを指定する場合

指定例：

```
set service = "serviceA=serviceA;usrlib.so"
```

UAP ライブラリにライブラリ名称だけを指定した場合、OpenTP1 は指定した UAP 共用ライブラリを OS のライブラリサーチパス（環境変数）を基に検索して使用します。

OS のライブラリサーチパス（環境変数）は、prcdlpath コマンドで変更できます。prcdlpath 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。

UAP 共用ライブラリ名称だけを指定する場合は、次の指定例のように、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どちらかで、UAP 共用ライブラリが格納されているディレクトリパスを OS のライブラリサーチパス（環境変数）に指定してください。

指定例：

この例では、UAP 共用ライブラリが格納されているディレクトリパスを /usr/local/lib とします。

putenv 形式の場合：putenv OS のライブラリサーチパス（環境変数） OpenTP1 ホームディレクトリ/lib:/usr/local/lib

dcputenv 形式の場合：dcputenv OS のライブラリサーチパス（環境変数） \$DCDIR/lib:/usr/local/lib

- スタブおよびサービス関数動的ローディング機能を使用するサービスを併用する場合

このオペランドに、スタブおよびサービス関数動的ローディング機能を使用するサービスを指定した場合、サービスはこのオペランドに指定したサービスの順で検索されます。

指定例：

```
set service = "serviceA=serviceA","serviceB=serviceB;/OLTP1/usrdir/lib/usrlib.so"
```

RPC のサービス関数、および RPC インタフェース定義の詳細については、マニュアル「OpenTP1 プログラム作成の手引」を参照してください。

SUP の場合は、指定する必要はありません。

●nice=プロセスの優先順位の変更

～ 〈符号なし整数〉 ((0～39))

このサービスグループプロセスの nice 値に加える値を指定します。プロセスの nice 値は正の整数で、この値が大きくなると CPU の優先順位は低くなります。nice の詳細については、ユーザの使用する OS の、該当するマニュアルを参照してください。

●parallel_count=常駐プロセス数 [, 最大プロセス数]

～ 〈符号なし整数〉 ((0～1024))

このサービスグループ内のサービスを並行処理するプロセス数を指定します。

常駐プロセス数を指定すると、同じサービスグループ内のサービスを処理するサーバプロセスを同時に複数起動するマルチサーバ機能が使用できます。マルチサーバ機能を使用することで、サービス要求に対する処理性能が向上します。なお、常駐プロセス数が 0 でも最大プロセス数が指定されていれば、マルチサーバ機能は使用できます。

最大プロセス数を指定すると、常駐プロセス数を超える処理は非常駐プロセス数を動的に増減して処理されます。指定した最大プロセス数以内でサーバプロセスの起動を制御するため、OpenTP1 システムの性能が低下するのを防げます。

プロセス数の指定条件を次に示します。

1. 常駐プロセス数と最大プロセス数の両方に 0 を指定することはできません。
2. 最大プロセス数を指定する場合は、常駐プロセス数以上の値を指定します。
3. 最大プロセス数を省略した場合は、すべてのプロセスが常駐プロセスとなります。
4. すべてのプロセスを非常駐プロセス（必要に応じて起動）とする場合は、常駐プロセス数に 0 を指定し、最大プロセス数に非常駐プロセス数を指定します。

なお、常駐プロセス数に 0 を指定し、最大プロセス数を省略した場合は、定義エラーになります。

このサービスグループ内のサービスでトランザクション処理をする場合、サービスを処理するプロセスは、トランザクションが終了するまで次のサービスを処理できません。したがって、最大プロセス数に 1 を指定した場合、一つのトランザクション内では、このサービスグループ内の複数のサービスを呼び出すことはできません。

非常駐プロセスを指定しているユーザサーバや scdrsproc コマンドを使用すると、一時的に最大でユーザサービス定義の parallel_count オペランドの指定値の倍のプロセス数（常駐プロセス数と最大プロセス数を指定している場合は最大プロセス数の倍、常駐プロセス数だけを指定している場合は常駐プロセス数の倍）が起動されることがあります。

非常駐プロセスを指定しているユーザサーバ、および scdrsproc コマンドで、倍のプロセス数が起動されるタイミングを次に示します。

非常駐プロセスを指定しているユーザサーバの場合

終了しようとしているプロセスが、dc_rpc_mainloop 関数または dc_mcf_mainloop 関数の終了後から dc_rpc_close 関数終了までの区間にある場合で、新たなサービス要求を受け付けたタイミング

scdrsprc コマンドで再起動するユーザサーバの場合

終了しようとしているプロセスが、dc_rpc_mainloop 関数または dc_mcf_mainloop 関数の終了後から dc_rpc_close 関数終了までの区間にある場合で、新たなプロセス起動を行ったタイミング

新たなプロセスが起動する契機については、マニュアル「OpenTP1 運用と操作」の scdrsprc コマンドの説明を参照してください。

SPP のソケット受信型サーバ、および SUP の場合は、指定する必要はありません。

●hold=Y|N

このサービスグループ下のサービスを実行中にサーバプロセスが異常終了した場合、このサービスグループ、またはサービスを閉塞するかどうかを指定します。

Y

サービスグループ、またはサービスを閉塞します。

N

サービスグループ、またはサービスを閉塞しません。

サービスグループを閉塞するか、サービスを閉塞するかは、service_hold オペランドの指定に従います。

サービスグループの閉塞とは、対象となるサービスグループのスケジュール要求を受けてもサーバプロセスを起動しないことです。ただし、サービスの閉塞の場合は、同じサービスグループ内のほかのサービスが閉塞していなければ、サーバプロセスを起動します。このとき、サービス要求はサービス要求元にエラーリターンします。

SPP のソケット受信型サーバ、および SUP の場合は、指定する必要はありません。MHP の場合、サービス開始前に異常終了したときに、このオペランドの指定に従った動作になります。サービス開始後の動作はアプリケーション属性定義（mcfaalcap 定義コマンド）の指定に従います。

●hold_recovery=Y|N

全面回復時にサービスグループ、およびサービスの閉塞状態を引き継ぐかどうかを指定します。

Y

サービスグループ、およびサービスの閉塞状態を引き継ぎます。

N

サービスグループ、およびサービスの閉塞状態を引き継ぎません。

SPP のソケット受信型サーバ、SUP、および MHP の場合は、指定する必要はありません。

システム環境定義の `start_scheduling_timing` オペランドに `BEFORE` を指定した場合、このオペランドの指定に関係なく、閉塞状態は引き継がれません。閉塞状態を引き継ぐ場合には、スケジュールサービス定義の `scd_hold_recovery` オペランドに `F` を指定してください。各オペランド指定による閉塞状態の引き継ぎについては、スケジュールサービス定義の `scd_hold_recovery` オペランドを参照してください。

●`deadlock_priority`=デッドロックの優先順位

～ 〈符号なし整数〉 ((1～127))

UAP のデッドロックの優先順位を指定します。

指定する値が小さいほど優先順位は高くなります。デッドロックが発生すると、優先順位のいちばん低い UAP の排他要求をエラーリターンします。

●`schedule_priority`=スケジュールの優先順位

～ 〈符号なし整数〉 ((1～16))

サービスグループのスケジュールの優先順位を指定します。

指定する値が小さいほど優先順位は高くなります。

プロセスサービス定義で指定した最大同時サーバプロセス数の上限までサーバプロセスが起動された状態で、新たなサービス要求が発生したときに優先順位を比較します。

優先順位が最も低いサービスグループのサーバプロセスを順次停止して、これより高い優先順位を持つサービスグループのサーバプロセスを起動します。

SPP のソケット受信型サーバ、および SUP の場合は、指定する必要はありません。

●`message_buflen`=最大メッセージ長

～ 〈符号なし整数〉 ((1024～31457280)) (単位：バイト)

共用メモリにあるクライアントからのメッセージをユーザサーバが受信するためのユーザデータ格納領域の大きさを指定します。プロセス固有の領域です。

OpenTP1 システムは各メッセージに 512 バイトの制御情報を付加するため、その値を加算した大きさを指定してください。

ソケット受信型サーバを除く SPP の場合は、`dc_rpc_call` 関数の `in_len` 設定値 + 512 バイトを指定してください。

MHP の場合は、1024 バイトを指定してください。

ソケット受信型サーバ、および SUP の場合は、指定する必要はありません。

●`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)

クライアントからのメッセージをいったん格納しておく共用メモリのプールの大きさを指定します。

クライアントからのメッセージをユーザサーバに渡す前にためておく領域です。

クライアントから送信されるメッセージの中で最大値以上を指定します。

また、OpenTP1 システムは各メッセージに 512 バイトの制御情報を付加するため、その値を加算した大きさを指定してください。

プールに格納されたメッセージは、FIFO 方式でスケジュールされてユーザサーバに渡されます。ユーザサーバのメッセージの取り出しが遅れると、このプールにメッセージがたまり、以後クライアントからのサービス要求は受け付けられません。このため余裕のある値を指定してください。

次の機能を使用する場合は、message_cell_size オペランドで指定した値の整数倍になる値を指定してください。

- プライオリティスケジューリング機能 (service_priority_control オペランドに Y を指定する)
- サービス単位の閉塞機能 (service_hold オペランドに Y を指定する)
- サービス単位のスケジュール制御機能 (scdsvcdef 定義コマンドを指定する)

整数倍になっていない場合、ここで指定した値を message_cell_size オペランドの整数倍に切り上げて、スケジュールメッセージ格納バッファプールサイズとして使用します。

次の計算式で求めた値を指定してください。算出値が 1024 より小さい場合は、1024 を指定してください。

ソケット受信型サーバを除く SPP の場合

$$\text{メッセージ格納バッファプール長} = (M+C) \times S$$

- M：該当するユーザサーバで受信する最大メッセージ長
ユーザデータの圧縮 (rpc_datacomp オペランド) を使用している場合でも、圧縮前のメッセージ (ユーザデータ) 長を使用してください。
- C：制御情報の付加分 (512 バイト)
- S：滞留の数

MHP の場合

MHP プロセスがすべて常駐、または balance_count オペランドで 0 を指定しているとき

$$\text{メッセージ格納バッファプール長} = C \times P$$

それ以外るとき

$$\text{メッセージ格納バッファプール長} = C \times P \times B$$

- C：制御情報の付加分 (512 バイト)
- P：parallel_count オペランドで指定する最大プロセス数
- B：balance_count オペランドの指定値

SPP のソケット受信型サーバ、および SUP の場合は、指定する必要はありません。

なお、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、scdbufgrp 定義コマンドを指定した場合、message_store_bufllen オペランドの指定値は無視されます。

●trn_expiration_time=トランザクションブランチ限界経過時間

～〈符号なし整数〉((0～65535)) (単位：秒)

トランザクションブランチの処理を監視する限界経過時間を指定します。指定時間を超えた場合は、このトランザクションブランチのプロセスを異常終了させて、ロールバックします。0 を指定した場合は、時間監視をしません。

UAP が異常終了した場合に閉塞するかどうかは hold オペランド、term_watch_time オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

これらの定義のほかに関数を発行して監視時間を指定することもできます。関数の詳細については、マニュアル「OpenTP1 プログラム作成の手引」を参照してください。

なお、RPC 機能を使用した場合に、他プロセスで実行するトランザクションブランチの処理時間も監視時間に含むかどうかは、trn_expiration_time_suspend オペランドで指定してください。

また、メッセージ制御機能 (TP1/Message Control) を使用した同期型のメッセージ処理を行う場合、メッセージ送受信時間を監視時間に含むかどうかは、trn_expiration_time_suspend オペランドで指定してください。

●trn_expiration_time_suspend=Y|N|F

トランザクションブランチの処理を監視するとき、次の処理時間も監視時間に含むかどうかを指定します。

1. 監視対象のトランザクションブランチが、RPC 機能を使ってほかのトランザクションブランチを呼び出し、その処理が終わるのを待つ時間
2. 連鎖 RPC で呼び出されたサーバ UAP が、次のサービス要求を待つ時間
3. 監視対象のトランザクションブランチが、非同期 RPC を使用してほかのトランザクションブランチを呼び出したあと、処理結果受信処理を行っている時間
4. 監視対象のトランザクションブランチが、メッセージ制御機能 (TP1/Message Control) を使用した同期型のメッセージ処理を行う場合のメッセージ送受信時間

Y

- 1., 2., 3., 4.のすべてを監視時間に含みます。

N

- 3.だけを監視時間に含みます。

F

1., 2., 3., 4.のどれも監視時間に含みません。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

このオペランドと各種タイマ監視との関係については、「[付録 A.2 トランザクションの時間監視](#)」を参照してください。

●watch_next_chain_time=連鎖 RPC 間隔監視時間

～〈符号なし整数〉((0～65535)) (単位：秒)

サーバ UAP が連鎖 RPC で呼び出された場合、前回のサービス要求に応答を返してから次のサービス要求を受けるまで、またはトランザクションが終了するまでの最大時間間隔を秒単位で指定します。指定した時間を超えると、UAP は異常終了します。

UAP が異常終了した場合に閉塞するかどうかは hold オペランド、term_watch_time オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

この指定値は、サーバとなる UAP だけに有効です。0 を指定すると、連鎖 RPC での次の要求が来るまで、またはトランザクションが終了するまで待ちます。

このオペランドと各種タイマ監視との関係については、「[付録 A.2 トランザクションの時間監視](#)」を参照してください。

●atomic_update=Y|N

このサービスグループのプロセスでトランザクションを発生させるかどうかを指定します。

Y

このプロセスはトランザクションの範囲に入れます。

トランザクションの範囲に入る方法には次の二つがあります。

1. このプロセスで dc_trn_begin 関数を新たに発行した場合
2. ほかの UAP からのサービス要求を受けて、OpenTP1 システムが自動的に行う場合

N

このプロセスはトランザクションの範囲に入れません。

dc_trn_begin 関数や dc_trn_unchained_commit 関数などのトランザクションサービス関数は使用できません。また、ジャーナル出力もできません。

なお、システム共通定義の jnl_fileless_option オペランドに Y を指定している場合、atomic_update オペランドに Y を指定しても N として動作します。

●receive_from=queue|socket|none

サービス関数が受け取るメッセージを何を使って受信するかを指定します。

queue

スケジュールサービスのスケジュールキューを使用します。

SPP および MHP の場合に指定します。

スケジュールキューを使用することで、dc_rpc_call 関数を発行したときには起動されていないサーバプロセスを新たに起動して呼んだり、同じサーバプロセスを複数起動してロードバランスを測ったりする機能が使用できます。

socket

UNIX ドメイン、またはインターネットドメインを使用します。

SPP の場合に指定します。

このサーバはオンデマンド起動、およびノード間負荷バランス機能を使用できません。

none

スケジュールサービスのスケジュールキュー、UNIX ドメイン、およびインターネットドメインのどれも使用しません。

SUP の場合に指定します。

●uap_trace_max=UAP トレース格納最大数

～ 〈符号なし整数〉 ((0～4095))

UAP トレースの最大格納数を指定します。

0 を指定した場合

UAP トレースを取得しません。

1 以上を指定した場合

指定値+1 の UAP トレースを取得します。

UAP トレースデータファイルまたは退避コアファイルに、UAP トレースの情報として次の情報を取得します。

- サービス関数の開始
- サービス関数の終了
- OpenTP1 のライブラリ関数の入り口または出口※

注※

OpenTP1 のライブラリ関数がリターンするときは、入り口情報を取得した領域を出口情報で上書きするため、UAP トレースの格納数は 1 となります。ただし、TP1/Message Queue、およびオンラインテスタ (TP1/Online Tester) を使用している場合は、OpenTP1 のライブラリ関数の入り口情報と出口情報で別の領域となるため、UAP トレースの格納数は 2 となります。

UAP トレースを参照することで、次に示す処理の流れを解析できます。

- UAP が正常終了までに行った処理の流れ
- UAP の異常終了時、障害が発生するまでに UAP が行った処理の流れ

この解析によって、ユーザは UAP に障害が起こった原因を解析でき、UAP を修正したり、システムを再構築したりする目安にできます。

UAP トレースの詳細については、マニュアル「OpenTP1 テスタ・UAP トレース使用の手引」を参照してください。

●uap_trace_file_put=Y|N

UAP トレースのトレース情報を、ファイルに取得するかどうかを指定します。

Y

トレース情報を UAP トレースデータファイルに取得します。この場合、ユーザサーバを終了させないで、UAP トレースの編集出力ができるようになります。UAP トレースデータファイルについては、マニュアル「OpenTP1 運用と操作」を参照してください。

トレース情報をファイルに取得できなかった場合、プロセス固有領域にトレース情報を取得します。

N

トレース情報をプロセス固有領域に取得します。この場合、ユーザサーバをアボートしたときにだけ、UAP トレースの編集出力ができます。

Y を指定した場合、ユーザサーバダウン時、または OpenTP1 終了後の再開始時に、UAP トレースデータファイルのバックアップファイルを各サーバ最大で 6 世代取得します。バックアップファイルの内訳は、該当するサーバの正常終了時に 3 世代、コアファイルの出力を伴う異常終了時に 3 世代です。バックアップファイルの格納先は、プロセスサービス定義の `prc_coresave_path` オペランドに指定した、コアファイルの格納先です。

Windows の場合、UAP トレースの編集出力は `uatdump` コマンドで行います。`uatdump` コマンドについては、マニュアル「OpenTP1 テスタ・UAP トレース使用の手引」を参照してください。

このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義でも指定できます。指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●term_watch_time=連続異常終了限界経過時間

～〈符号なし整数〉((0～32767)) (単位：分)

このユーザサーバのサーバプロセスが異常終了した回数を監視する限界経過時間を指定します。0 を指定した場合は、時間監視をしません。

ソケット受信型サーバを除く SPP は、hold オペランドに N を指定した場合に有効です。

SPP のソケット受信型サーバ、および SUP は auto_restart オペランドに Y を指定した場合に有効です。

MHP は、hold オペランドに N を指定し、サービス開始前にユーザサーバのサーバプロセスが異常終了した場合に有効です。

OpenTP1 では、時間監視によって UAP プロセスを停止させることがあります。時間監視によって UAP プロセスが停止したとき、サーバの閉塞する条件を次の表に示します。

表 3-16 hold オペランド、term_watch_time オペランドの指定値と、サーバの閉塞の関係

ユーザサーバの種類	hold オペランドの指定値	term_watch_time オペランドの指定値	サーバが閉塞するかどうか
SUP SPP MHP (サービス開始前)	Y	すべて (hold オペランドに Y を指定した場合は、term_watch_time オペランドの指定に関係なく、無条件で閉塞します)	要因に関係なく無条件で閉塞します。
		0	閉塞しません。
	N	0 以外	term_watch_time オペランドに指定した時間内に、3 回異常終了すると閉塞します (要因によっては異常終了回数に数えない場合があります)。
MHP (サービス開始後)	すべて	すべて	hold オペランドおよび term_watch_time オペランドの指定に関係なく、アプリケーション属性定義 (mcfaalcap 定義コマンド) の指定で閉塞するかどうかが決まります。

hold オペランドに N を指定し、term_watch_time オペランドに 0 を指定しているときに、UAP の問題による異常終了があると、UAP プロセスの再起動と UAP プロセスの異常終了を繰り返します。例えば、非常駐プロセスだけのユーザサーバ (parallel_count オペランドの常駐プロセス数に 0 を指定) で hold オペランドに N を指定し、term_watch_time オペランドに 0 を指定した場合、実行形式プログラム名に指定誤りがあると、サービス実行時に KFCA00713-E メッセージを繰り返し出力します。これを防ぐため、term_watch_time オペランドには 0 以外の値を指定することをお勧めします。

オペランドの異常終了回数のカウント条件を、次の表に示します。

表 3-17 オペランドの異常終了回数のカウント条件

異常終了要因	異常終了回数のカウント
UAP の問題による異常終了	カウントする
ipc_sockctl_watchtime 満了による異常終了	カウントする
trn_cpu_time 満了による異常終了	カウントする
watch_next_chain_time 満了による異常終了	カウントする
xat_trn_expiration_time 満了による異常終了	カウントする
service_expiration_time 満了による異常終了	カウントしない
trn_expiration_time 満了による異常終了	カウントしない
trn_completion_limit_time 満了による異常終了	カウントしない
mcfaalcap -v ntmetim 満了による異常終了	カウントしない
prckill, dcsvstop -fd コマンドによる異常終了	カウントしない

なお、term_watch_time オペランドの指定時間内に 3 回連続してこのユーザサーバのサーバプロセスが異常終了した場合、ソケット受信型サーバを除く SPP は、hold オペランドの指定に関係なく、このサービスグループを閉塞し、SPP のソケット受信型サーバおよび SUP は auto_restart オペランドの指定に関係なく強制停止します。

MHP の場合は、サービス開始前、term_watch_time オペランドの指定時間内に 3 回以上連続してユーザサーバのサーバプロセスが異常終了すると、このサービスグループのスケジュールを閉塞し、ユーザサーバを強制停止します。

ユーザサーバが、hold オペランドに N を、term_watch_time オペランドに 0 以外を指定した場合に異常終了したときの例を、次に示します。

1. trn_cpu_time オペランドの指定値を超えて異常終了した。

異常終了 1 回目です。閉塞しません。

2. trn_expiration_time オペランドの指定値を超えて異常終了した。

異常終了回数にはカウントしません。閉塞しません。

3. UAP の問題によって異常終了した。

異常終了 2 回目です。閉塞しません。

4. trn_cpu_time オペランドの指定値を超えて異常終了した。

異常終了 3 回目です。閉塞します。

term_watch_time オペランドで指定した連続異常終了限界経過時間が解除される契機を、次に示します。

1. term_watch_time オペランドの指定値の時間が経過したとき。

2. ユーザサーバの閉塞を解除したとき。
3. ユーザサーバを正常停止したあと、正常開始したとき。
4. OpenTP1 を正常開始、再開始したとき。

●mcf_jnl_buff_size=MCF のジャーナルバッファの大きさ

～ 〈符号なし整数〉 ((4096～131072)) (単位：バイト)

MCF の実行によって取得されるジャーナルデータを蓄えておく領域の大きさを指定します。

指定する値の計算については、「[MCF マネジャ定義](#)」の mcfmuap 定義コマンドの-j オプションの説明を参照してください。

指定する値が大きいほどジャーナルを取得するときのディスクに対する入出力回数が少なくなります。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、MCF マネジャ定義の mcfmuap 定義コマンドの-j オプションの値を仮定します。

MHP および TP1/Message Control の機能を使用する SPP だけ有効です。

SUP の場合は、指定する必要はありません。

●type=other|MHP

このサービスグループの種別を指定します。

other：キュー受信型 SPP に指定します。

MHP：MHP に指定します。

SUP の場合は、指定する必要はありません。

●balance_count=1 プロセスが処理するサービス要求数

～ 〈符号なし整数〉 ((0～512))

このユーザサーバに対応するスケジュールキューに、処理されないで残っているサービス要求を、1 プロセスで幾つ処理させるかを指定します。サービス要求が、(このオペランドの指定値) × (起動プロセス数) を超えてスケジュールキューに残っている場合には、非常駐プロセスを起動してサービス要求を処理させます。このオペランドは、parallel_count オペランドで非常駐プロセスを起動するように指定したサービスグループにだけ有効です。

次の場合には、必ずこのオペランドに 0 を指定してください。

- ・ 非常駐プロセスだけで構成される (常駐プロセス数に 0 を、最大プロセス数に 2 以上を指定する) サーバで、リカーシブコールを使用する場合

- 一つの常駐プロセスと、ほかの非常駐プロセスで構成される（常駐プロセス数に 1 を、最大プロセス数に 2 以上を指定する）サーバで、リカーシブコールを使用する場合

0 を指定すると、サービス要求のあった時点で起動されているすべてのプロセスがサービスを処理中の場合、非常駐プロセスが起動されます。

SPP のソケット受信型サーバおよび SUP の場合は、このオペランドを指定する必要はありません。

●uid=ユーザ識別子

～ 〈符号なし整数〉 ((0～4294967294))

このサービスグループのプロセスの所有者となるユーザ ID を指定します。

OS に登録されているユーザ ID を指定してください。

最大値は OS に依存するので、使用している OS のマニュアルを参照してください。

●auto_restart=Y|N

UAP が異常終了した場合の処置を指定します。

Y

UAP が異常終了した場合、プロセスを再起動します。

N

UAP が異常終了した場合、プロセスを再起動しません。

ソケット受信型サーバを除く SPP および MHP の場合は、指定する必要はありません。

●critical=Y|N

このサービスグループを実行中に UAP が異常終了した場合の OpenTP1 システムの処置を指定します。

Y

UAP が異常終了した場合、OpenTP1 システムを停止します。

N

UAP が異常終了した場合、OpenTP1 システムを続行します。

●lck_wait_priority=排他待ちの優先順位

～ 〈符号なし整数〉 ((0～127))

ユーザサーバが wait 型の排他要求を発行して、排他待ちの状態となる場合の優先順位を指定します。

指定する値が小さいほど優先順位は高くなり、優先順位の低い排他要求より先に排他待ちが解除されます。0 を指定した場合は、待ち行列の最後に登録されます。

優先順位を低く設定すると、あとから発行された優先順位の高い排他要求に追い越されることがあるので注意してください。

●mcf_psv_id=アプリケーション起動プロセス識別子

～ 〈16 進数字〉 ((01～ff))

アプリケーション起動を実行する場合の、アプリケーション起動プロセス識別子を指定します。アプリケーション起動を実行する SPP には必ず指定してください。SUP, MHP, およびアプリケーション起動を実行しない SPP には、指定する必要はありません。

●trn_cpu_time=トランザクションブランチ CPU 監視時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

トランザクションブランチが同期点処理までに使用できる CPU 時間を指定します。0 を指定した場合は、監視しません。指定時間を超えた場合は、該当するトランザクションブランチのプロセスを異常終了し、ロールバックします。ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

UAP が異常終了した場合に閉塞するかどうかは hold オペランド, term_watch_time オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

●service_hold=Y|N

このサービスグループ下のサービスについて、サービス単位の閉塞管理をするかどうかを指定します。

Y

サービス単位の閉塞管理します。

N

サービス単位の閉塞管理はしません。

このオペランドに Y を指定した場合、message_cell_size オペランドの指定が有効になります。

サービス単位の閉塞管理をすることによって、サービス単位にサーバ異常終了時の閉塞や、運用コマンドでの閉塞ができます。ただし、サーバ異常終了時のサービス単位の閉塞は、hold オペランドに Y を指定、または service_term_watch_time オペランドを指定したときだけでできます。

対象となるサービスのスケジュール要求を受けた場合、サービス要求をサービス要求元にエラーリターンします。

SPP のソケット受信型サーバ, SUP, および MHP の場合は、指定する必要はありません。

●service_priority_control=Y|N

サービス要求単位に指定された優先順位に従ってスケジューリングするかどうかを指定します。

Y

サービス要求単位に指定された優先順位に従ってスケジューリングします。

N

サービス要求単位に指定された優先順位に従ってスケジューリングしません。

このオペランドに Y を指定した場合、message_cell_size オペランドの指定が有効になります。SPP のソケット受信型サーバ、SUP、および MHP の場合は、指定する必要はありません。

●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)

次の場合は、message_store_buflen オペランドで指定したスケジュールメッセージ格納バッファプールをセル分割してメッセージを格納します。そのときのセルサイズをここで指定します。

- service_priority_control オペランドに Y を指定した場合
- service_hold オペランドに Y を指定した場合
- scdsvcddef 定義コマンドを指定した場合

スケジュールメッセージ格納バッファプールをセル分割した各セルには、セル管理テーブル (16 バイト) を作成するため、そのサイズを考慮する必要があります。

また、このオペランドの指定値には、8 の整数倍となる値を指定してください。指定した値が 8 の整数倍でない場合は、その値を 8 の整数倍に切り上げた値が指定されます。

message_store_buflen オペランドで指定した値が、この指定値の整数倍になっていない場合、message_store_buflen オペランドの値をこの指定値の整数倍に切り上げて、スケジュールメッセージ格納バッファプールサイズとして使用します。

service_priority_control オペランド、および service_hold オペランドに N を指定し、かつ scdsvcddef 定義コマンドを省略した場合は、ここで指定する必要はありません。

また、SPP のソケット受信型サーバ、SUP、MHP の場合にも、指定をする必要はありません。

●max_socket_msg=ソケット受信型サーバ受信メッセージ数の最大値

～〈符号なし整数〉((1～500))

ソケット受信型サーバが受信するメッセージの最大数を指定します。

ソケット受信型サーバは、サービス受け付け状態ではないとき (トランザクションのコミット、ロールバック指示待ち、連鎖 RPC 中、または RPC ネストコールの応答待ちなど)、新たなサービス要求を即時処理しないで、サービス受け付け状態に戻るまで保存します。

サーバが受信するメッセージ数がこのオペランドの指定を超えた場合、dc_rpc_call 関数は、DCRPCER_SERVER_BUSY でエラーリターンします。

ソケット受信型サーバを除く SPP, SUP, および MHP の場合は、指定する必要はありません。

●max_socket_msglen=ソケット受信型サーバ受信メッセージ長の最大値

～ 〈符号なし整数〉 ((1～30270)) (単位：キロバイト)

ソケット受信型サーバが受信するメッセージ長の合計の最大値を指定します。

サーバが受信するメッセージ長の合計がこのオペランドの指定値を超えた場合、dc_rpc_call 関数は、DCRPCER_SERVER_BUSY でエラーリターンします。

指定値は、サーバへ送信するメッセージの最大値以上にする必要があります。

また、指定値には、メッセージに付ける制御情報の大きさ（約 0.3 キロバイト）を加える必要があります。

ソケット受信型サーバを除く SPP, SUP, および MHP の場合は、指定する必要はありません。

●trf_put=Y|N

該当するサーバで起動されたトランザクションが出力したジャーナルをトランザクションリカバリジャーナルに出力するかどうかを指定します。

Y

トランザクションリカバリジャーナルファイルに出力します。

N

トランザクションリカバリジャーナルファイルに出力しません。

トランザクションリカバリジャーナルファイルを作成すると、長時間トランザクションによって発生するジャーナル障害を防止でき、またリラン時間を短縮できます。ただし、この機能は、性能、およびメモリへのオーバヘッドがあるので、長時間トランザクションを使用するサーバの場合に指定してください。

系切り替え機能を使用する OpenTP1 システムでは、この機能を使用できないため、ユーザサービス定義および rap リスナーサービス定義の trf_put オペランドには N を指定してください。

●mcf_mgrid=アプリケーション起動プロセスの属する MCF マネジャ識別子

～ 〈識別子〉 ((A～Z, a～z))

アプリケーション起動を実行する場合、アプリケーション起動プロセスが属している MCF マネジャの MCF マネジャプロセス識別子を指定します。

SUP, MHP, およびアプリケーションを起動しない SPP の場合は、指定する必要はありません。

●mcf_service_max_count=MCF 通信関数発行回数上限値

～ 〈符号なし整数〉 ((0～65535))

UAP の暴走を監視するために、UAP が発行するすべての MCF 通信関数の発行回数の上限値を指定します。

指定回数を超えて MCF 通信関数を発行した場合、UAP を異常終了させます。

0 を指定した場合は、MCF 通信関数の発行回数を監視しません。

MHP の場合、共に指定を省略すると、MCF マネージャ定義の mcfmuap コマンドの -d オプションに指定した値を仮定します。

SPP の場合、共に指定を省略すると監視しません。

MHP および MCF の機能を使用する SPP だけ有効です。

SUP の場合は、指定する必要はありません。

●trn_statistics_item=統計情報項目〔, 統計情報項目〕...

トランザクションブランチの統計情報を取得する項目を指定します。

nothing

トランザクションブランチの統計情報を取得しません。

base

基本情報として、次のトランザクションブランチの情報を取得します。

- トランザクションブランチの識別子
- トランザクションブランチの決着結果
- トランザクションブランチの実行プロセス種別
- トランザクションブランチの実行サーバ名
- トランザクションブランチの実行サービス名

executiontime

基本情報とトランザクションブランチの実行時間情報を取得します。

cputime

基本情報とトランザクションブランチの CPU 時間情報を取得します。

nothing の指定は、一つしかできません。また、nothing とほかの統計情報項目を同時に指定した場合、nothing の指定は無効になります。

トランザクションに関する統計情報を取得するときは次のどれかを指定してください。

- トランザクションサービス定義の trn_tran_statistics オペランドに Y を指定
- trnstics コマンドで -s オプションを指定

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクション定義の値を仮定します。

統計情報を取得する場合、取得情報の種類が多くなるほど、トランザクション性能が悪くなります。このため、統計情報は、システムチューニング時やトラブル調査時など、トランザクションの性能が悪くなくてもかまわない場合に取得してください。

●node_down_restart=Y|N

OpenTP1 再開始時、該当するユーザサーバを自動起動するかどうかを指定します。

該当するユーザサーバが dcstop コマンド実行前に正常終了した場合、この指定は無効となります。なお、次に示す場合は、status_change_when_termining オペランドの指定によって、正常終了したユーザサーバを起動するかどうかを決定します。

- ・ システム正常終了中に、ユーザサーバが正常終了したあとにシステムダウンが発生した場合
- ・ システム計画終了中に、dcstop コマンドで終了処理をする前に SUP が正常終了した場合

Y

OpenTP1 再開始時、自動起動します。

N

OpenTP1 再開始時、自動起動しません。

OpenTP1 の異常終了後に再開始（リラン）で OpenTP1 を起動させた場合、node_down_restart オペランドに N を指定しているユーザサーバは、リラン時の起動対象にならないで異常終了のままとなります。異常終了したままのユーザサーバがある場合に OpenTP1 を正常停止させようとする、OpenTP1 が停止します。OpenTP1 が停止した場合は、次に示すどちらかの対処をしてください。

- ・ 該当ユーザサーバを一度正常開始または正常停止させたあとに、dcstop コマンドを実行して OpenTP1 を正常停止させる。
- ・ -n オプションを指定した dcstop コマンドを実行して OpenTP1 を強制正常停止させる。

●rpc_response_statistics=Y|N

レスポンス統計情報を取得するかどうかを指定します。

Y

レスポンス統計情報を取得します。

N

レスポンス統計情報を取得しません。

レスポンス統計は、RPC コールのレスポンスタイム、サービスの実行時間、およびサーバの CPU 時間などを取得します。

このオペランドは、dc_rpc_call 関数のクライアント・サーバで個別に指定できます。レスポンス統計情報は、同期応答型 RPC（連鎖 RPC を含む）および非同期応答型 RPC で取得します。

●server_type="betran"|"xatmi"|"xatmi_cbl"

サービス関数の呼び出し時、OpenTP1 のパラダイムに従うか、XATMI のパラダイムに従うかを指定します。

"betran"

OpenTP1 のパラダイムに従います。

サービス関数は、XATMI の関数を使用できません。

"xatmi"

XATMI のパラダイムに従います。

"xatmi_cbl"

XATMI の COBOL 言語のサービスパラダイムに従います。

SUP、および MHP の場合は、指定する必要はありません。

●trn_rm_open_close_scope=process|transaction

OpenTP1 以外が提供しているリソースマネージャに対して、XA インタフェースである二つの関数（xa_open 関数と xa_close 関数）の発行タイミングを指定します。

process

dc_rpc_open 関数発行時に xa_open 関数を発行し、dc_rpc_close 関数発行時に xa_close 関数を発行します。

transaction

トランザクション開始時に xa_open 関数を発行し、トランザクション終了時に xa_close 関数を発行します。

process を指定した場合は、dc_rpc_open 関数の発行から dc_rpc_close 関数の発行までリソースマネージャの資源を占有します。したがって、トランザクションの性能を重視する場合は process を指定してください。

transaction を指定した場合は、トランザクションの開始終了ごとに xa_open 関数、xa_close 関数を発行します。したがって、リソースマネージャの資源を有効利用したい場合は、transaction を指定してください。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●trn_optimum_item=トランザクション最適化項目 [, トランザクション最適化項目] …

複数のユーザサーバで構成されるグローバルトランザクションの性能を向上させるための最適化項目を、次の文字列で指定します。

base

同期点処理全体（プリペア処理、コミット処理、およびロールバック処理）を最適化します。OpenTP1 のトランザクション制御は 2 相コミット方式で行っているため、二つのトランザクションブランチ間のコミット制御には、4 回のプロセス間通信が必要となります。

次の条件をすべて満たす場合、親トランザクションブランチが子トランザクションブランチのコミット処理を代わりに行うことで、コミット制御に必要な 4 回のプロセス間通信を削減します。

1. 親トランザクションブランチと、子トランザクションブランチが同一 OpenTP1 下にあること。
2. 親トランザクションブランチが、子トランザクションブランチを同期応答型 RPC で呼び出していること。
3. 子トランザクションブランチでアクセスしたリソースマネージャの XA インタフェース用オブジェクトが、親トランザクションブランチにもリンクされていること。

asyncprepare

base の指定条件を満たしていないため同期点処理全体の最適化ができない場合に、プリペア処理を最適化します。

次の条件をすべて満たす場合、親トランザクションブランチから発行された RPC によって子トランザクションブランチがサービス要求を実行したときに、RPC のリターン前にプリペア処理を実行することで、2 回のプロセス間通信を削減します。

1. base を指定した最適化ができないこと。
2. 親トランザクションブランチが、子トランザクションブランチを同期応答型 RPC で呼び出していること。

ただし、この最適化を実行した場合、親トランザクションブランチが発行した同期応答型 RPC の応答時間が遅くなります。また、子トランザクションブランチは、プリペア処理からコミット処理までの間隔（親トランザクションブランチからの指示がないとトランザクションを決着できない状態）が大きくなります。そのため、親トランザクションブランチの OpenTP1 がシステムダウンし、トランザクションブランチ間の連絡ができなくなると、ジャーナルファイルのスワップやチェックポイントダンプファイルの有効化が遅れ、子トランザクションブランチの OpenTP1 もシステムダウンする場合があります。

トランザクション最適化項目は、重複して指定できます。ただし、優先順位は次のようになります（1.> 2.）。

1. base
2. asyncprepare

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●`purge_msgget=Y|N`

非常駐サーバについて、サーバプロセスがない場合に、OpenTP1 がこのサービスグループに割り当てた OS のメッセージキュー（`msgget` システムコールでの割り当て）を解放するかどうかを指定します。

Y

サーバプロセス消滅時、OS のメッセージキューを解放します。

N

サーバプロセス消滅時、OS のメッセージキューを解放しません。

このオペランドで Y を指定すると、OS のメッセージキューをむだに占有しなくなります。解放は、サーバプロセス消滅後、約 10 秒間サービス要求が発生しない場合に行います。OS のメッセージキュー解放後にサービス要求が発生した場合、OS のメッセージキューを再度割り当ててサービス要求を処理します。

このオペランドで Y を指定する場合は、ほかのサービスグループでも Y を指定してください。これは、OS のメッセージキューが、一つのサービスグループに対して一つあるためです。そのため、ユーザサービスデフォルト定義の `purge_msgget` オペランドにも、Y を指定してください。

このオペランドは、ユーザサービス定義の `receive_from` オペランドで `queue` を指定し、かつ `parallel_count` オペランドの常駐プロセス数に 0 を指定したサービスグループに対して有効です。それ以外のサービスグループは無視されます。

なお、OS のメッセージキューの再割り当てには数 10 ミリ秒掛かるため、このレベルで性能を重視するシステムでは注意してください。

●`cancel_normal_terminate=Y|N`

このユーザサーバで、`dcsvstop` コマンドでの正常終了を抑止するかどうかを指定します。

Y

このユーザサーバで `dcsvstop` コマンドでの正常終了を抑止します。

N

このユーザサーバで `dcsvstop` コマンドでの正常終了を抑止しません。

Y を指定した場合、このユーザサーバは `dcsvstop` コマンドでの強制停止、および `dcstop` コマンドでの停止しか受け付けません。

●`prc_abort_signal=アボート用シグナル番号`

～ 〈符号なし整数〉 ((1～128))

OpenTP1 下のサーバをアボートさせる際に使用する、シグナル番号を指定します。アボート用シグナル番号には、必ずコアファイル出力の動作をするシグナル番号を指定してください。コアファイル出力の動作をするシグナル番号を指定しないと、`dcstop` コマンドおよび `dcsvstop` コマンドでの強制終了時に、`-d` オプションを指定していてもコアファイルが生成されません。

`prc_abort_signal` オペランドで指定したシグナル番号は、次の表に示すとおりサーバに送信されます。

表 3-18 シグナル番号が送信されるタイミングとサーバ

シグナル番号が送信されるタイミング	シグナル番号が送信されるサーバ
次に示す時間の満了時 ・ トランザクションブランチ限界経過時間 (trn_expiration_time オペランドで指定) ・ サービス関数開始から終了までの実行監視時間 (service_expiration_time オペランドで指定) ・ トランザクション完了限界時間 (trn_completion_limit_time オペランドで指定) ・ 非トランザクション MHP 限界経過時間 (mcfaalcap コマンドの-v オプションの ntmetim オペランドで指定)	時間監視が満了したサーバ
次に示す方法でサーバを強制停止した時 ・ prckill コマンドを実行 ・ -df オプションを指定した dcsvstop コマンドを実行 ・ -fd オプションを指定した dcstop コマンドを実行	強制停止させるサーバ

●**rpc_service_retry_count=サービスのリトライ回数の最大値**
～ 〈符号なし整数〉 ((0～65535))

サービスリトライ機能で、サービス関数をリトライする最大回数を指定します。

0 を指定すると、サービスリトライ機能を使用しません。そのため、dc_rpc_service_retry 関数はエラーリターンし、サービス関数はリトライされません。

0 以外を指定すると、指定された回数だけ連続してサービス関数がリトライされます。指定された回数を超えて呼び出された dc_rpc_service_retry 関数はエラーリターンし、サービス関数はリトライされません。

なお、サービスリトライ機能を使用できるのは、SPP だけになります。

●**rpc_extend_function=RPC サービスの機能拡張レベル**
～ 〈16 進数字〉 ((00000000～0000000F))

RPC サービスの機能の拡張レベルを、次の中から指定します。

次の指定値で示す機能の拡張レベルを複数指定する場合、それぞれの指定値の論理和を指定してください。

00000000

RPC サービスの機能を拡張しません。

00000001

サービス要求実行中の SPP が異常終了した場合に、dc_rpc_call 関数、dc_rpc_call_to 関数および dc_rpc_poll_any_replies 関数で、リターン値「DCRPCER_SERVICE_TERMINATED (00378)」を返します。

00000002

トランザクション内で開始した非トランザクション (flags に DCRPC_TPNOTRAN を指定した dc_rpc_call 関数、および dc_rpc_call_to 関数) の連鎖 RPC を、同期点処理で終了しないで、flags

に DCNOFLAGS を指定した dc_rpc_call 関数、および dc_rpc_call_to 関数で明示的に終了するまで継続します。

00000004

非同期応答型 RPC の応答メッセージを受け取っていない状態で、トランザクションの同期点処理をした場合に、非トランザクション属性の非同期応答型 RPC の応答メッセージは破棄しないで、トランザクション属性の非同期応答型 RPC の応答メッセージだけを削除します。

00000008

SPP プロセスで、dc_rpc_call 関数、および dc_rpc_call_to 関数の呼び出し元の応答受信タイムアウトを検出したため、処理を中断してサービス要求を破棄した場合に、KFCA00339-W のメッセージを出力します。

●max_socket_descriptors=ソケット用ファイル記述子の最大数

～〈符号なし整数〉((32～2032))

OpenTP1 制御下のプロセス※で、ソケット用に使用するファイル記述子の最大値を指定します。

OpenTP1 制御下のプロセス※では、システムサーバやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数および通信する他ノードの数によって、ソケット用のファイル記述子の最大数を変更する必要があります。

このオペランドには、次の条件を満たす値を指定してください。

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

条件を満たさない値を指定した場合は、このオペランドの指定値は次に示すように強制的に補正されます。

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

注※

MCF サービス (MCF マネジャサービス, MCF 通信サービス, およびアプリケーション起動サービス) 以外の OpenTP1 プロセスが対象です。MCF サービスの場合は、「[システムサービス情報定義](#)」および「[システムサービス共通情報定義](#)」を参照してください。

ソケット用ファイル記述子の最大数の計算式を、次に示します。

$$\uparrow (\text{このユーザサーバが通信する UAP プロセス数}^{\ast 1} + \text{システムサービスプロセス数}^{\ast 2}) / 0.8 \uparrow$$

↑↑：小数点以下を切り上げます。

注※1

このユーザサーバが通信する UAP プロセス数は、次に示す値の合計です。

- このユーザサーバが通信する自 OpenTP1 内の UAP プロセス数
- このユーザサーバが通信する他ノード内の UAP プロセス数

注※2

システムサービスプロセス数とは、自 OpenTP1 内のシステムサービスプロセスの数です。自 OpenTP1 内のシステムサービスプロセスは、rpcstat コマンドで表示されるサーバ名をカウントすることで求められます。rpcstat コマンドで表示されるサーバ名のうち、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造に記載されているシステムサービスプロセスをカウントしてください。

このオペランドの指定値が小さいと、OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため、プロセスが KFC A00307-E メッセージを出力して異常終了します。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●max_open_fds=UAP プロセスでアクセスするファイルおよびパイプの最大数

～〈符号なし整数〉((16～2016))

ユーザサーバのプロセスでアクセスする、ファイルおよびパイプの最大数を指定します。

ファイルおよびパイプのアクセスには、ファイル記述子が使われます。

ユーザサーバのプロセスで多くのファイルをアクセスすると、システムサーバやユーザサーバとの間でプロセス間の情報交換に使用するソケット用のファイル記述子が不足します。そのため、アクセスするファイルおよびパイプの数を事前に設定しておく必要があります。

ユーザサーバのプロセスでアクセスするファイルおよびパイプの最大数の計算式を、次に示します。

(DAM サービス, TAM サービス, ジャーナルサービス, およびステータスサービスの各システム定義で指定した全 OpenTP1 ファイルシステム数※¹) + (ISAM のキー数) + 20※² + (ユーザファイル数※³) + (自ノード内で RPC 先となる schedule_method に namedpipe を指定したユーザサーバ数※⁴)

注※1

オンラインで使用する、OpenTP1 ファイルシステム領域数

注※2

リモート API 機能を使用してサービスを要求する場合は、固定数 20 にさらに 16 を加えてください。

注※3

OpenTP1 の制御下でない、ユーザが独自に使用するファイル

注※4

このユーザサーバから、自ノード内でRPC 先となるユーザサーバのうち schedule_method に namedpipe を指定しているユーザサーバの数

このオペランドの指定値が小さいと、OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため、プロセスが KFCA00307-E メッセージを出力して異常終了します。

max_open_fds オペランドに指定した値は、dc_rpc_open 関数を発行することで UAP の動作環境に設定されます。OpenTP1 は、UAP プロセス実行中に UAP プロセスでオープンするファイルディスクリプタの最大数の超過を検出すると、UAP プロセスを異常終了させます。

dc_rpc_open 関数発行後に UAP プロセスでオープンするファイルディスクリプタの最大数を OS に対して再設定（変更）した場合、OpenTP1 の認識する最大数と一致なくなり、UAP プロセスがオープンするファイルディスクリプタの最大数の超過を検出できないため、OpenTP1 がエラーを処理できません。このため、dc_rpc_open 関数を呼び出したあとに UAP プロセスでオープンするファイルディスクリプタの最大数を変更しないでください。変更した場合の動作は保証できません。

●service_term_watch_time=連続サービス異常終了限界経過時間

～〈符号なし整数〉((0～32767)) (単位：分)

このユーザサーバのサーバプロセスが、同一サービス実行中に異常終了した回数を監視する限界経過時間を指定します。0 を指定した場合は、このオペランドを指定しなかったと見なし、時間監視をしません。

このオペランドを指定すると、指定した時間内にサーバプロセスが同一サービス実行中（dc_rpc_mainloop 関数内）で 3 回異常終了した場合は、そのサービスを閉塞します。また、サーバプロセスがサービス実行中以外（dc_rpc_mainloop 関数外）で 3 回異常終了した場合は、サービスグループを閉塞します。

このオペランドは、サーバプロセスが異常終了しても、サービスグループを閉塞しないでサービス単位で閉塞管理する場合（hold オペランドに N を指定し、service_hold オペランドに Y を指定した場合）に有効となります。また、SPP のソケット受信型サーバ、SUP、および MHP では、指定しても無視されます。

このオペランドに 0 以外の値を指定すると、サービス単位で異常終了した回数を監視するため、term_watch_time オペランドは、指定しても無視されます。

service_term_watch_time オペランドの指定と、hold オペランド、service_hold オペランド、および term_watch_time オペランドの指定の関係を、次に示します。

各オペランドの指定値				サーバプロセスが異常終了した回数と OpenTP1 の処理		
hold	service_hold	term_watch_time	service_term_watch_time	1 回目	2 回目	3 回目
Y	Y	×	×	サービス閉塞	—	—
	N	×	×	サービスグループ閉塞	—	—

各オペランドの指定値				サーバプロセスが異常終了した回数と OpenTP1 の処理		
hold	service _hold	term _watch _time	service _term _watch _time	1 回目	2 回目	3 回目
N	Y	0	指定なし	プロセス再起動	プロセス再起動	プロセス再起動
		0 以外	指定なし	プロセス再起動	プロセス再起動	サービスグループ 閉塞
		0	0	プロセス再起動	プロセス再起動	プロセス再起動
		0 以外	0	プロセス再起動	プロセス再起動	サービスグループ 閉塞
		×	0 以外	プロセス再起動	プロセス再起動	サービス閉塞※
	N	0	×	プロセス再起動	プロセス再起動	プロセス再起動
		0 以外	×	プロセス再起動	プロセス再起動	サービスグループ 閉塞

(凡例)

×：指定できません（指定しても無視されます）。

－：何もしません。

注※

サーバプロセスがサービス実行中以外（dc_rpc_mainloop 関数外）で 3 回異常終了した場合は、サービスグループを閉塞します。

●**termed_after_service=Y|N**

このユーザサーバで、サービス終了時に負荷が減少している場合に、**非常駐プロセスを終了させるかどうか**を指定します。

Y

サービス終了時に負荷が減少している場合に、非常駐プロセスを終了させます。

N

スケジューラサービスが、一定の間隔で不要な非常駐プロセスを終了させます。

通常 OpenTP1 では、一定の間隔（10 秒）でプロセスの状態をチェックして、不要な非常駐プロセスを終了させています。しかし、サーバが非常に多いシステムでは一度に終了させるプロセス数が増え、プロセスの終了処理と同時に発生したサービス要求の処理性能が悪くなることがあります。

このオペランドに Y を指定すると、サービス終了時にプロセスの状態をチェックします。そのため、一度に終了させるプロセス数が減り、プロセスの終了処理と同時に発生したサービス要求の処理性能が悪くなるのを防げます。また、システムで起動しているプロセス数を減らすことで、システムへの負荷を減らせます。

ただし、サービスが終了するたびにプロセスの状態をチェックするため、1 回ごとのサービス要求の処理性能は悪くなります。また、システム形態によっては、非常駐プロセスの起動・終了が頻繁に発生するようになり、サービス要求の処理性能が悪くなることがあります。

SPP のソケット受信型サーバ、および SUP の場合は、指定しても無視されます。

●xat_trn_expiration_time=同期点処理の限界経過監視時間

～ 〈符号なし整数〉 ((1～2147483647)) (単位：秒)

TP1/NET/OSI-TP-Extended を使った、XATMI インタフェースでの OSI TP 通信を実行する場合に、他システムとの同期点処理の限界経過監視時間を指定します。

UAP が異常終了した場合に閉塞するかどうかは hold オペランド、term_watch_time オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

TP1/NET/OSI-TP-Extended の仕様については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended 編」を参照してください。

MHP の場合は、指定しても無視されます。

●xat_osi_usr=Y|N

このサーバで、TP1/NET/OSI-TP-Extended を使った、XATMI インタフェースでの OSI TP 通信用のスタブを使用するかどうかを指定します。

Y

このサーバで XATMI インタフェースでの OSI TP 通信用のスタブを使用します。

N

このサーバで XATMI インタフェースでの OSI TP 通信用のスタブを使用しません。

N を指定すると、TP1/NET/OSI-TP-Extended を使った、XATMI インタフェースでの OSI TP 通信ができません。

TP1/NET/OSI-TP-Extended の仕様については、マニュアル「OpenTP1 プロトコル TP1/NET/OSI-TP-Extended 編」を参照してください。

MHP の場合は、指定しても無視されます。

●rpc_trace=Y|N

RPC トレースを取得するかどうかを指定します。

Y

RPC トレースを取得します。

N

RPC トレースを取得しません。

RPC トレースを取得した場合、処理速度が低下し、RPC がタイムアウトでエラーリターンすることがあります。その場合は、状況に応じて次の 1.または 2.の最大応答待ち時間（デフォルト値：180 秒）を十分な値まで増やして対処してください。

1. watch_time オペランド（システム共通定義、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で指定）に指定した最大応答待ち時間
2. クライアントから rap サーバに引き継いだ最大応答待ち時間

クライアントから rap サーバに最大応答待ち時間を引き継ぐかどうかは、クライアント環境定義の DCWATCHTIMINHERIT オペランド（TP1/Client/W, TP1/Client/P の場合）、または TP1/Client/J 環境定義の dcwatchtiminherit オペランド（TP1/Client/J の場合）で指定します。

DCWATCHTIMINHERIT オペランドまたは dcwatchtiminherit オペランドに Y が指定されている場合は、上記 2.の最大応答待ち時間を使用してください。

DCWATCHTIMINHERIT オペランドまたは dcwatchtiminherit オペランドに N が指定されているか、指定が省略されている場合は、上記 1.の最大応答待ち時間を使用してください。

リモート API 機能を使用する場合、このオペランドに Y を指定しても、rap クライアントの RPC トレースは取得されません。

このオペランドは、ユーザサービスデフォルト定義、およびシステム共通定義でも指定できます。

指定値の優先順位は次のとおりです（1.> 2.> 3.）。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. システム共通定義

●rpc_trace_name="RPC トレースを取得するファイル名"

～ 〈パス名〉

RPC トレースを取得するファイルのパス名を指定します。

パス名のうち、RPC トレースを取得するファイル名（デフォルトは rpctr）の最大長は、13 文字です。

パス名に環境変数を指定する場合、パス名の先頭に環境変数を指定してください（指定例：\$DCDIR/tmp/ファイル名）。

●rpc_trace_size=RPC トレースを取得するファイルのサイズ

～ 〈符号なし整数〉（(1024～2147483648)）（単位：バイト）

RPC トレースを取得するファイルのサイズを指定します。

このオペランドに 4096 以下を指定していても、送受信された電文長がこの指定値を上回る場合、このオペランドで指定した値よりも大きなトレースファイルが作成されることがあります。

●trn_rollback_information_put=no|self|remote|all

トランザクションブランチがロールバックした場合に、ロールバック要因に関する情報をログに取得するかどうかを指定します。

no

ロールバック情報を取得しません。

self

ロールバック要因が発生したトランザクションブランチでだけ、ログにロールバック情報を取得します。

remote

self に加え、他ノードのトランザクションブランチからロールバック要求されたトランザクションブランチでも、ログにロールバック情報を取得します。

all

remote に加え、自ノードのトランザクションブランチからロールバック要求されたトランザクションブランチでも、ログにロールバック情報を取得します。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●schedule_method=msgqueue|namedpipe

ユーザサーバのスケジューリング方式を指定します。

msgqueue

OS のメッセージ機能を使用して、サービス要求のスケジューリングをする方式です。

namedpipe

OS の名前付きパイプを使用して、サービス要求のスケジューリングをする方式です。

ユーザサービス定義の service_wait_time オペランドを指定する場合に、namedpipe を指定してください。

このオペランドで namedpipe を指定した場合の注意事項を次に示します。

- msgqueue を指定した場合に比べ、処理性能が劣化します。
- スケジュールサービスで使用するファイルディスクリプタの数が、このオペランドを指定したユーザサーバ数だけ増加します。

このオペランドは、SPP のソケット受信型サーバおよび SUP の場合は指定する必要はありません。

●service_wait_time=ユーザサーバの非常駐プロセスのサービス要求待ち時間

～ 〈符号なし整数〉 ((1～4096)) (単位：秒)

ユーザサーバの非常駐プロセスのサービス要求待ち時間を指定します。

このオペランドを指定したユーザサーバの非常駐プロセスは、このオペランドで指定した時間だけサービス要求を待ち、サービス要求がない場合に終了させます。このオペランドを省略した場合、一定間隔でスケジュールサービスが、サーバの負荷状態を監視して非常駐プロセスを終了させます。

このオペランドは、ユーザサービス定義の schedule_method オペランドで namedpipe を指定した場合に有効です。また、このオペランドを指定すると、ユーザサービス定義の termed_after_service オペランドの指定が無効になります。

このオペランドは、SPP のソケット受信型サーバおよび SUP の場合は指定する必要はありません。

●mcf_spp_oj=Y|N

SPP からの分岐メッセージ送信、アプリケーション起動、およびメッセージ再送要求時に、履歴情報（ジャーナル：OJ）を取得するかどうかを指定します。

Y

履歴情報を取得します。

N

履歴情報を取得しません。

●adm_message_option=メッセージ出力指定

～ 〈1 けたの 16 進数〉

次のメッセージを出力するかどうかをビットマスクの論理和で指定します。ビットマスクがオンのメッセージは出力され、ビットマスクがオフのメッセージは出力されません。

メッセージ ID	ビットマスク (16 進数)	メッセージ内容
KFCA01811-I	1	サーバ開始中メッセージ
KFCA01813-I	2	サーバオンラインメッセージ
KFCA01842-I	4	サーバ終了中メッセージ
KFCA01843-I	8	サーバ停止メッセージ

(例)

KFCA01813-I と KFCA01843-I を出力したい場合は、「set adm_message_option=A」と指定します。

●trn_watch_time=トランザクション同期点処理時の最大通信待ち時間

～ 〈符号なし整数〉 ((1～65535)) (単位：秒)

トランザクションの同期点処理で、トランザクションブランチ間で行う通信（プリペア、コミット、ロールバック指示、または応答など）の受信待ち時間の最大値を指定します。

指定時間を過ぎても指示または応答がない場合は、該当するトランザクションブランチが2相コミットの1相目完了前であればロールバックさせ、1相目完了後であればトランザクションサービスのシステムプロセスでトランザクション決着処理を再試行します。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。さらに、トランザクションサービス定義でも省略した場合、トランザクションを行ったユーザサーバの watch_time の値を仮定します。ただし、watch_time に 0 を指定した場合は 120 秒を仮定します。

●trn_limit_time=トランザクションブランチ最大実行可能時間

～〈符号なし整数〉((0～65535)) (単位：秒)

トランザクションブランチの最大実行可能時間を指定します。

トランザクションブランチを開始してから同期点処理が終了するまでの時間が、このオペランド指定時間を超えないように dc_rpc_call 関数、dc_rpc_poll_any_replies 関数および同期点処理内で行う通信のタイムアウト時間を、次のように自動設定します。

- dc_rpc_call 関数および dc_rpc_poll_any_replies 関数のタイムアウト時間

「 $K \geq$ このオペランド指定時間」の場合は、要求処理を実行しないで、タイムアウトでエラーリターンします。

「 $K < \text{このオペランド指定時間}$ 」でかつ「 $(\text{このオペランド指定時間} - K) \geq W$ 」の場合は、 W をタイムアウト時間とします。

「 $K < \text{このオペランド指定時間}$ 」でかつ「 $(\text{このオペランド指定時間} - K) < W$ 」の場合は、 $(\text{このオペランド指定時間} - K)$ をタイムアウト時間とします。

K

現時刻－トランザクションブランチ開始時刻

W

dc_rpc_call 関数の場合は watch_time オペランド指定時間

dc_rpc_poll_any_replies 関数の場合は引数 timeout 指定時間

- 同期点処理内で行う通信のタイムアウト時間

「 $K \geq$ このオペランド指定時間」の場合は、タイムアウト時間を 1 秒とします。

「 $K < \text{このオペランド指定時間}$ 」でかつ「 $(\text{このオペランド指定時間} - K) \geq W$ 」の場合は、 W をタイムアウト時間とします。

「 $K < \text{このオペランド指定時間}$ 」でかつ「 $(\text{このオペランド指定時間} - K) < W$ 」の場合は、 $(\text{このオペランド指定時間} - K)$ をタイムアウト時間とします。

K

現時刻－トランザクションブランチ開始時刻

W

trn_watch_time オペランド指定時間

trn_watch_time オペランドを省略した場合は watch_time オペランド指定時間

上記の受信待ち以外の処理で時間が掛かった場合は、このオペランド指定時間以内にトランザクションブランチが終了しないことがあります。

同期点処理開始前にこのオペランド指定時間が経過した場合、そのトランザクションはロールバックします。

0 を指定した場合は、残り時間の監視をしません。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●trn_rollback_response_receive=Y|N

RPC 先トランザクションブランチにロールバック指示を送信したあと、**ロールバック完了通知を受信**するかどうかを指定します。N を指定した場合、RPC 先トランザクションブランチからのロールバック完了通知を受信しないで（RPC 先トランザクションブランチのロールバック処理の完了を待たないで）自トランザクションブランチを終了します。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●trn_partial_recovery_type=type1|type2|type3

UAP 障害時のトランザクション同期点処理方式を指定します。

RPC がタイムアウトし、RPC 発行先プロセスのアドレスが未解決の場合やトランザクション実行中の UAP がダウンした場合に、トランザクションブランチ間の連絡がスムーズにできないで、トランザクションの決着に時間が掛かることがあります。

このオペランドでは、次に示す障害が発生した場合のトランザクション同期点処理方式を、指定値に示す三つの方式から選択して指定します。

(障害 1) RPC がタイムアウトした場合

この場合、RPC 発行元トランザクションブランチは、サービス要求がどのプロセスで実行されているかがわからないため、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信できません。そのため、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチはトランザクション同期点メッセージ待ちとなり、トランザクションの決着に時間が掛かります。

(障害 2) RPC 発行元 UAP が RPC の応答受信前にダウンした場合

この場合、RPC 発行元トランザクションブランチは、サービス要求がどのプロセスで実行されているかがわからないため、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信できません。そのため、RPC 発行先トランザクションブランチはトランザクション同期点メッセージ待ちとなり、トランザクションの決着に時間が掛かります。

(障害 3) RPC 発行先 UAP からの応答受信後に RPC 発行元 UAP と RPC 発行先 UAP がほぼ同時にダウンした場合

この場合、それぞれのトランザクションブランチを引き継いだトランザクション回復プロセスは、相手 UAP プロセスのダウンを知らないため、すでに存在しない UAP プロセスにトランザクション同期点メッセージを送信してしまい、トランザクションの決着に時間が掛かることがあります。

type1

(障害 1) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

(障害 2) が発生した場合、RPC 発行元トランザクションブランチは、RPC 発行先トランザクションブランチにトランザクション同期点メッセージを送信しないでトランザクションを決着します。RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

(障害 3) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

type2

(障害 1) が発生してトランザクションをコミットする場合は type1 と同じです。

(障害 1) が発生してトランザクションをロールバックする場合、または (障害 2) が発生した場合は、RPC 発行元トランザクションブランチは、RPC 発行先トランザクションブランチが存在するノードのトランザクションサービスプロセスにトランザクション同期点メッセージを送信後、トランザクションを決着します。トランザクション同期点メッセージを受信したトランザクションサービスプロセスは、該当するトランザクションブランチを処理中のプロセスに、トランザクション同期点指示を送信します。

(障害 3) が発生した場合、RPC 発行元トランザクションブランチおよび RPC 発行先トランザクションブランチは、トランザクション同期点メッセージ受信処理がタイムアウトすることによって、トランザクションを決着します。

type3

(障害 1) が発生してトランザクションをコミットする場合は、type1 と同じです。

(障害 1) が発生してトランザクションをロールバックする場合、(障害 2) が発生した場合、または (障害 3) が発生した場合、相手トランザクションブランチが存在するノードのトランザクションサービスプロセスに、トランザクション同期点メッセージを送信します。トランザクション同期点メッセージを受信したトランザクションサービスプロセスは、該当するトランザクションブランチを処理中のプロセスに、トランザクション同期点指示を送信します。

次に示す場合、このオペランドに type2 または type3 を指定しても、トランザクションの決着に時間が掛かることがあります。

1. RPC 実行中に、RPC 発行先 UAP の状態が変更となり (負荷増加, UAP 終了, UAP 閉塞など)、ほかのノードの同一 UAP にサービス要求が再転送された場合
2. 相手先の OpenTP1 がこのオプションをサポートしていないバージョンの場合

3. 相手先トランザクションブランチがトランザクション同期点メッセージ受信処理以外で時間が掛かっている場合

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

●rpc_destination_mode=namdonly|namd|definition

dc_rpc_call 関数で要求したサービスの送信先を決定する際に、ネームサービスから得た情報と、ユーザサービスネットワーク定義の指定値のどちらを優先させるかを指定します。

namdonly

従来どおり、ネームサービスの情報から選択したあて先へサービスを送信し、ユーザサービスネットワーク定義の指定値の検索は一切しません。

namd

ネームサービスの情報から選択したあて先へサービスを送信します。

ネームサービスの情報内に該当するサービスが見つからない場合だけ、ユーザサービスネットワーク定義の指定値を検索します。

definition

ユーザサービスネットワーク定義の指定値から検索したあて先へサービスを送信します。ユーザサービスネットワーク定義に該当するサービスがない場合だけ、ネームサービスにあて先検索を要求します。

このオペランドは、dc_rpc_call 関数を呼び出す SUP, SPP, MHP の場合に指定します。

このオペランドに namd または definition を指定した場合、ユーザサービスネットワーク定義の指定値を検索して、サービス要求を送信します。ユーザサービスネットワーク定義に、システム共通定義の all_node オペランドに指定していないノードのサービス情報を定義している場合、このノードの OpenTP1 がダウンして再開始したあとに要求したサービスが、タイムアウトエラーになることがあります。

●rpc_rap_auto_connect=Y|N

リモート API 機能の常設コネクションを使用してサービスを要求する場合に、UAP とリモート API 制御プロセスとの間のコネクションを、OpenTP1 が自動的に管理するかどうかを指定します。

Y

UAP とリモート API 制御プロセスとの間のコネクションを、OpenTP1 が自動的に管理します（オートコネクトモード）。

N

UAP とリモート API 制御プロセスとの間のコネクションを、dc_rap_connect 関数または dc_rap_disconnect 関数を使ってユーザが管理します（非オートコネクトモード）。

●rpc_rap_inquire_time=リモート API 機能を使用して要求するサービスの問い合わせ間隔最大時間

～ 〈符号なし整数〉 ((0～1048575)) (単位：秒)

リモート API 機能を使用してサービスを要求する場合に、リモート API 制御プロセスが UAP からのサービス要求を処理してから、次のサービス要求を処理するまでの間隔の、最大待ち時間を指定します。この指定値は、リモート API 制御プロセス側で監視するタイマであり、指定時間を超えても次のサービス要求がない場合、リモート API 制御プロセスは UAP がダウンしたものと見なして処理します。

このオペランドに 0 を指定した場合、リモート API 制御プロセス側の定義で指定した値が有効になります。

このオペランドは、リモート API 機能を使用して dc_rpc_call 関数を呼び出す SUP、SPP、MHP の場合に指定します。

常駐 SPP からリモート API 機能を使用している場合、このオペランドまたは rap リスナーサービス定義の rap_inquire_time オペランドに 0 を指定しないでください。指定した場合、rap リスナーが存在するノードの OpenTP1 システムを終了させようとしても、rap リスナーが終了できないため、常駐 SPP が終了するまで無限に待ち続けます。

SPP または SUP のユーザーサービス定義でこのオペランドの指定を省略した場合、rap リスナーサービス定義の rap_inquire_time オペランドで指定した値が最大待ち時間となります。どちらも省略した場合は、180 秒を仮定します。

●rpc_request_cancel_for_timeout=Y|N

クライアント UAP で設定した、サービス要求を送信してからサービスの応答が返るまでの待ち時間を、サーバ側に引き継ぎ、サーバにクライアントが待ち状態であることを意識させるかどうかを指定します。

クライアント UAP の応答待ち時間をサーバに引き継ぐと、クライアント UAP がタイムアウトしても、サーバ側でサービスを実行したり、同期点処理の実行を待ち続けたりすることを防止できます。

Y

クライアント UAP の応答待ち時間をサーバに意識させます。

N

クライアント UAP の応答待ち時間をサーバに意識させません。

ただし、サーバ側の TP1/Server Base のバージョンが 03-02 で、かつドメイン指定の同期型 dc_rpc_call 関数の場合は、RPC がエラーとなりますので、N を指定してください。

●status_change_when_termining=Y|N

システム正常終了中にユーザサーバが正常終了したあとにシステムダウンが発生した時、またはシステム計画終了中に dcstop コマンドによる終了処理前に SUP が正常終了した時などのサーバの状態の変化を、次回再開始時に反映させるかどうかを指定します。

Y

最終的な状態の変化を反映させます。

正常終了したユーザサーバは、次の OpenTP1 の再開始時に再起動されません。

N

最終的な状態の変化を反映させません。

正常終了したユーザサーバは、次の OpenTP1 の再開始時に再起動されます。

●service_expiration_time=サービス関数開始から終了までの実行監視時間

～〈符号なし整数〉((0～65535)) (単位：秒)

SPP プロセスでの、サービス関数開始から終了までの実行監視時間を指定します。この時間を満了してもサービス関数がリターンしない場合、このプロセスを強制停止します。

0 を指定した場合、時間監視はしません。

UAP が異常終了した場合に閉塞するかどうかは hold オペランド、term_watch_time オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

注

サービス関数開始から終了までの実行監視時間の精度は秒単位です。そのため、タイミングによっては、このオペランドに指定した値よりも短い時間でプロセスを強制停止することがあります。サービス関数開始から終了までの実行監視時間が小さくなるほど、誤差の影響を受けやすくなりますので、このオペランドには 3 (単位：秒) 以上の値を指定することを推奨します。

●multi_schedule=Y|N

サービス要求をマルチスケジューラ機能を使用してスケジューリングするかどうかを指定します。

このオペランドは、マルチスケジューラ機能を使用する場合、RPC 送信側ユーザサーバで指定します。

Y

サービス要求をマルチスケジューラ機能を使用してスケジューリングします。

N

サービス要求をマルチスケジューラ機能を使用してスケジューリングしません。

マルチスケジューラ機能を使用する場合に、OpenTP1 システム内の同一サービスグループに、マルチスケジューラ機能を使用しているユーザサーバと、マルチスケジューラ機能を使用していないユーザサーバが混在していると、マルチスケジューラ機能を使用しているユーザサーバに優先して負荷分散されます。マルチスケジューラ機能を使用しているユーザサーバが起動されていなかったり、スケジュールできない状態（閉塞、メッセージ格納バッファ不足など）になっていたりする場合は、マルチスケジューラ機能を使用していないユーザサーバに負荷分散されます。

マルチスケジューラ機能を使用する場合、このオペランドとともに、スケジュールサービス定義およびユーザサービス定義の scdmulti 定義コマンドを、RPC 受信側で指定する必要があります。

なお、この機能は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

●make_queue_on_starting=Y|N

OpenTP1 が非常駐のサービスグループに割り当てるスケジュールキューを、サーバ起動時に割り当てるかどうかを指定します。

Y

サーバ起動時にスケジュールキューを割り当てます。

N

サーバ起動時にスケジュールキューを割り当てません。

このオペランドを指定すると、非常駐サーバがシステムの資源不足によってスケジューリングできない状態をサーバ起動時に検出できます。

ただし、スケジュールキューの割り当てに失敗した場合でも、サーバの起動処理は続行します。

このオペランドは、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の `purge_msgget` オペランドで Y を指定した場合、および `parallel_count` オペランドの常駐プロセス数に 0 以外を指定した場合は無視されます。

●loadcheck_interval=負荷監視インタバル時間

～〈符号なし整数〉((0～65535)) (単位：秒)

このサービスグループの負荷レベルを監視するインタバル時間を指定します。負荷監視時に負荷レベルに変更がある場合は、各ノードのネームサービスにサーバ情報が通知されます。そのため、最悪のケースでは、負荷監視インタバルごとにサーバ情報がネットワーク上に送信されるおそれがあるので、必要以上に短い値を指定しないでください。なお、スケジュールサービス定義に `scd_announce_server_status=N` を指定した場合は、このオペランドを指定しても無視されます。また、0 を指定した場合は、負荷レベルの監視をしません。

このオペランドを指定しない場合、負荷監視インタバルは 30 秒となります。また、負荷監視の可否のチェックなどについては 10 秒のインタバルで実行されます。つまり、負荷監視の可否のチェックなどの 3 回目には、負荷監視が実行されます。

しかし、このオペランドを指定する場合、負荷監視インタバルはオペランドの指定値となり、負荷監視の可否のチェックなどについては、10 と各ユーザサーバのこのオペランドの指定値との最大公約数から求めたインタバルで実行されます。例えば、SPP1 の `loadcheck_interval` オペランドに 3 を、SPP2 の `loadcheck_interval` オペランドに 5 を指定する場合、10 と 3 と 5 の最大公約数の 1 (秒) が負荷監視の可否のチェックなどを実行するインタバルとなります。負荷監視の可否のチェックなどの 3 回目には、SPP1 の負荷監視が実行されます。5 回目には、SPP2 の負荷監視が実行されます。

したがって、システムに与える影響を少なくするために、`loadcheck_interval` オペランドに指定する値は、5 の倍数にすることをお勧めします。

MHP、ソケット受信型 SPP、および SUP の場合は、このオペランドを指定する必要はありません。指定しても無視されます。

なお、この機能は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

●**levelup_queue_count=U1,U2, および leveledown_queue_count=D0,D1**

～〈符号なし整数〉((0～32767))

このサービスグループの負荷レベルを判断するサービス要求滞留数を指定します。

U1

サーバの負荷レベルが LEVEL1 に上がったと判断するサービス要求滞留数

U2

サーバの負荷レベルが LEVEL2 に上がったと判断するサービス要求滞留数

D0

サーバの負荷レベルが LEVEL0 に下がったと判断するサービス要求滞留数

D1

サーバの負荷レベルが LEVEL1 に下がったと判断するサービス要求滞留数

levelup_queue_count オペランド, leveledown_queue_count オペランドの各指定値は、次の条件を満たすように指定する必要があります。

(条件) $0 \leq D0 < U1 \leq D1 < U2$

指定値が条件を満たしていない場合、サーバ起動時に定義エラーとなり起動が失敗となります。

なお、スケジューラサービス定義に scd_announce_server_status=N を指定した場合は、このオペランドを指定しても無視されます。また、leveledown_queue_count オペランドは、levelup_queue_count オペランドが指定されている場合だけ有効とし、levelup_queue_count オペランドを指定しないで指定した場合は無視されます。逆に、levelup_queue_count オペランドを指定した場合でも leveledown_queue_count オペランドは省略できます。この場合、leveledown_queue_count オペランドの各指定値は、次のように決定されます。

$$D0 = U1 / 2$$

$$D1 = U1 + (U2 - U1) / 2 \quad (D0, D1 \text{ 共に小数点以下は切り捨て})$$

MHP, ソケット受信型 SPP, および SUP の場合は、このオペランドを指定する必要はありません。指定しても無視されます。

なお、この機能は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

●**ipc_sockctl_highwater=ソケットの一時クローズ処理開始数パーセンテージ [ソケットの一時クローズ処理非対象数パーセンテージ]**

～〈符号なし整数〉((0～100))

max_socket_descriptors オペランドの指定値に対して、ソケットの一時クローズ処理開始数パーセンテージを指定します。

OpenTP1 は、プロセス内のソケット用に使用しているファイル記述子の数が、次の値を超えた時点で、一時クローズ処理を開始します。

max_socket_descriptors オペランドの指定値
× (ソケットの一時クローズ開始数パーセンテージ / 100)

ソケットの一時クローズ処理開始数パーセンテージに 0 を指定した場合、コネクションを確立するたびに一時クローズ処理が実行されます。一時クローズ処理については、マニュアル「OpenTP1 解説」を参照してください。

また、一時クローズ処理の対象外とするコネクション数のパーセンテージも指定できます。一時クローズ処理の対象外とするコネクション数は、次の値です。

max_socket_descriptors オペランドの指定値
× (ソケットの一時クローズ非対象数パーセンテージ / 100)

ソケットの一時クローズ非対象数パーセンテージには、ソケットの一時クローズ処理開始数パーセンテージ以下の値を指定してください。ソケットの一時クローズ処理開始数パーセンテージより大きい値を指定した場合は、ソケットの一時クローズ処理開始数パーセンテージと同じ値が指定されたものとして動作します。

OpenTP1 は、プロセス内で確立したコネクションを、確立した順に管理しています。ソケットの一時クローズ処理非対象数パーセンテージを指定した場合、最も古く確立されたコネクションから順に、一時クローズ処理要求が送信されます。

max_socket_descriptors オペランドの指定値が大きくなり、かつソケットの一時クローズ処理開始数パーセンテージに小さい値が指定されていると、一時クローズ要求が多発し、性能に影響を与えたり、通信障害になったりすることがあります。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_sockctl_watchtime=ソケット再利用可能監視時間

～ 〈符号なし整数〉 ((0~65535)) (単位：秒)

プロセス内のソケット用に使用しているファイル記述子の数が、max_socket_descriptors オペランドの指定値になった時点から、一時クローズ処理によってソケットが再利用できるようになるまでの監視時間(秒)を指定します。

UAP が異常終了した場合に閉塞するかどうかは hold オペランド、term_watch_time オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

一時クローズ処理は、コネクションを確立したプロセス間の合意によってコネクションを切断するため、一時クローズ処理の要求を送信したプロセスはその応答を受けるまではコネクションを切断できません。応答を受信したあと、コネクションは切断され、ソケットが再利用できるようになります。

ipc_sockctl_watchtime オペランドの指定値を経過しても、どのプロセスからも一時クローズ処理要求に対する応答が返らない場合、プロセスは強制停止されます。ipc_sockctl_watchtime オペランドに 0 を指定した場合、無限に待ちます。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_conn_interval=コネクション確立監視時間

～〈符号なし整数〉((8～65535)) (単位：秒)

データ送信時のコネクション確立監視時間を秒単位で指定します。

ノンブロッキングモードで呼び出した connect() システムコールに対する応答を受信するまでの監視時間を指定します。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_send_interval=データ送信監視間隔

～〈符号なし整数〉((5～32767)) (単位：秒)

データ送信時のデータ送信監視間隔を指定します。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_send_count=データ送信監視回数

～〈符号なし整数〉((1～32767))

データ送信時の送信完了監視回数を指定します。

OpenTP1 のデータ送信監視時間は、次に示す秒数で監視します。

ipc_send_intervalオペランドの指定値 × ipc_send_countオペランドの指定値

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_header_rcv_time=通信制御データの受信監視時間

～〈符号なし整数〉((5～32767)) (単位：秒)

OpenTP1 が、データの受信が始まったことを TCP/IP から通知されてから、通信制御データを受信するまでの監視時間を指定します。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●**rpc_send_retry_count**=TCP/IP コネクションの接続時にエラーが発生した場合のリトライ回数

～ 〈符号なし整数〉 ((0～65535))

サーバの応答送信時の TCP/IP コネクションの接続で、ECONNREFUSED (239), EHOSTUNREACH (242), ENETUNREACH (229) のどれかのエラーが発生した場合、このオペランドでリトライ回数を指定することによって、エラーを回避できることがあります。

0 を指定した場合は、TCP/IP コネクションの接続時に上記のエラーが発生しても接続をリトライしません。

このオペランドに 1～65535 を指定し、TCP/IP コネクションの接続時に上記のエラーが発生した場合、rpc_send_retry_interval オペランドに指定した時間を待ったあと、接続をリトライします。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●**rpc_send_retry_interval**=TCP/IP コネクションの接続時にエラーが発生した場合のリトライ間隔

～ 〈符号なし整数〉 ((0～300000)) (単位：ミリ秒)

サーバの応答送信時の TCP/IP コネクションの接続で、ECONNREFUSED (239), EHOSTUNREACH (242), ENETUNREACH (229) のどれかのエラーが発生した場合、このオペランドでリトライ間隔をミリ秒単位で指定することによって、エラーを回避できることがあります。

0 を指定した場合は、間隔を空けずに TCP/IP コネクションの接続をリトライします。1～19 は指定できません。1～19 を指定した場合は、20 ミリ秒間隔を空けてから TCP/IP コネクションの接続をリトライします。

このオペランドは、rpc_send_retry_count オペランドに 1～65535 を指定した場合に有効になります。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●**ipc_listen_sockbufset**=Y|N

～ 〈N〉

OpenTP1 がプロセス間通信で使用する listen ソケットの生成時に、ipc_sendbuf_size オペランド、および ipc_recvbuf_size オペランドに指定した TCP/IP の送受信バッファサイズを listen ソケットに設定するかどうかを指定します。

Y

listen ソケットに、TCP/IP の送受信バッファサイズを設定します。

N

listen ソケットに、TCP/IP の送受信バッファサイズを設定しません。

ipc_sendbuf_size オペランド、および ipc_recvbuf_size オペランドに指定した TCP/IP の送受信バッファサイズは、OpenTP1 プロセスがコネクション確立要求を受信し、コネクションを確立したあとに設定します。

コネクションを確立したあとに TCP/IP の送受信バッファサイズを変更するため、コネクションの接続元と接続先の間で TCP/IP のバッファサイズの値が異なり、通信の遅延が発生することがあります。

同一ノード内に TP1/Server Base と TP1/Client があり、TP1/Client から同一ノード内の TP1/Server Base に対して通信を行う場合は、このオペランドに Y を指定することをお勧めします。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●ipc_recvbuf_size=TCP/IP の受信バッファサイズ

～〈符号なし整数〉((0, 8192～1048576)) (単位：バイト)

コネクションごとに確保されるソケットの受信バッファのサイズを指定します。高速な通信媒体や MTU の大きな通信媒体を使用している場合、この値を大きくすることによって性能向上を図れます。

0 を指定した場合は、OS の受信バッファサイズを適用します。受信データサイズが 8192 バイトを超える場合、このオペランドに 0 を指定することで受信バッファのサイズが拡張され、性能向上を図れることがあります。

なお、0 を指定する場合、このノードと通信するすべてのノードで同じ値を指定してください。同じ値を指定しない場合、通信するノードとバッファサイズに差異が生じ、通信性能が劣化するおそれがあります。また、1～8191 は指定できません。1～8191 を指定した場合、0 を仮定します。

ipc_recvbuf_size オペランドに指定した値よりも小さいデータをやり取りし合う通信処理の場合、ipc_tcpnodelay オペランドに Y を指定することを推奨します。

注

TCP は、受信したデータに対し、送達確認(ACK)パケットを返信します。

受信バッファのサイズに対し、受信したデータが小さいと、データを受信しても直ちに ACK を返信しません(遅延 ACK)。

このオペランドに大きな値を指定し、小さいデータをやり取りし合うような通信処理の場合、遅延 ACK の影響によって性能が悪くなるおそれがあります。遅延 ACK の詳細については、TCP/IP の文献を参照してください。

このオペランドの値は、OS で使用できる TCP/IP の受信バッファのサイズ以下の値を指定してください。

ユーザサーバの場合、ここで指定を省略し、ユーザサービスデフォルト定義でも省略したときは、システム共通定義の値を仮定します。

●`ipc_sendbuf_size`=TCP/IP の送信バッファサイズ

～〈符号なし整数〉((0, 8192～1048576)) (単位：バイト)

コネクションごとに確保されるソケットの送信バッファのサイズを指定します。高速な通信媒体や MTU の大きな通信媒体を使用している場合、この値を大きくすることによって性能向上を図れます。

0 を指定した場合は、OS の送信バッファサイズを適用します。送信データサイズが 8192 バイトを超える場合、このオペランドに 0 を指定することで送信バッファのサイズが拡張され、性能向上を図れることがあります。

0 を指定することを推奨する場合については、「3. システム共通定義」の `ipc_sendbuf_size` オペランドの説明を参照してください。

なお、0 を指定する場合、このノードと通信するすべてのノードで同じ値を指定してください。同じ値を指定しない場合、通信するノードとバッファサイズに差異が生じ、通信性能が劣化するおそれがあります。また、1～8191 は指定できません。1～8191 を指定した場合、0 を仮定します。

`ipc_sendbuf_size` オペランドに指定した値よりも小さいデータをやり取りし合う通信処理の場合、`ipc_tcpnodelay` オペランドに Y を指定することを推奨します。

注

このオペランドの値は、OS で使用できる TCP/IP の送信バッファのサイズ以下の値を指定してください。

ユーザサーバの場合、ここで指定を省略し、ユーザサービスデフォルト定義でも省略したときは、システム共通定義の値を仮定します。

●`polling_control_data`=Y|N

OpenTP1 制御下の SPP, MHP で、サービス要求の受信待ちの状態のときに、定期的に待ち状態に割り込んで、一時クローズ処理要求が到着していないかどうかを検査することを指示します。

Y

サービス要求の受信待ち状態に定期的に割り込み、一時クローズ処理要求が到着していないかどうかを検査します。

N

サービス要求の受信待ち状態に割り込まないで、サービス要求が到着するまで待ち状態を継続します。

クライアントからのサービス要求が長時間発生しない常駐 UAP には、このオペランドに Y を指定する必要があります。

また、時間帯によって業務トラフィックにばらつきのあるシステムでは、常駐 UAP のユーザサービス定義のこのオペランドに Y を指定する必要があります。

●thread_yield_interval=ソケットの再利用指示を受信できる契機を与えるインタバル時間

～ 〈符号なし整数〉 ((1～86400)) (単位：秒)

OpenTP1 制御下の SPP, MHP で、サービス要求の受信待ちの状態のときに、定期的に待ち状態に割り込んで、一時クローズ処理要求が到着していないかどうかを検査する場合の検査インタバル時間を秒単位で指定します。

また、サービス要求を待つことなく次々にサービスを受け付けている場合には、連続してサービス要求を受け付けた時間が一定時間を超えたところで、一時クローズ処理要求が到着していないかどうかを検査します。このオペランドは、このインタバル値としても使用されます（連続してサービス要求を受け付けた時間がこのオペランド指定値を超えたときの、一時クローズ処理要求到着検査には、シグナルによる割り込みは発生しません）。

このオペランドの指定値は、一時クローズ処理要求を送信したプロセスが応答を待つ時間（180 秒：ipc_socketctl_watchtime オペランドのデフォルト値）よりも小さい値を設定します。

サービス要求待ち状態に割り込んで、一時クローズ処理要求受信を検査する機能は、polling_control_data オペランドに Y を指定した場合にだけ動作します。連続してサービス要求を受け付けたときに一時クローズ処理要求受信を検査する機能は、polling_control_data オペランドの指定に関係なく、動作します。

このオペランドに最大値を指定した場合は、polling_control_data オペランドの指定に関係なく、一時クローズ処理要求が到着していないかどうかの検査を実行しません。

●groups=グループ識別子 [グループ識別子]

～ 〈符号なし整数〉 ((0～4294967294))

このサービスグループのグループアクセスリストを設定します。

OS に登録されているグループ ID を指定してください。このオペランドには 16 個まで指定できます。

このオペランドの指定に関係なく、OpenTP1 管理者のグループ ID は setgid() システムコールによって自動的に設定されます。

最大値は OS に依存するので、使用している OS のマニュアルを参照してください。

●loadlevel_message=Y|N|A

サーバの負荷レベルを変更した場合に、KFCA00849-W の負荷レベル通知メッセージを出力するかどうかを指定します。

Y

サーバの負荷レベルが高負荷状態（LEVEL2）に上がった場合、および通常状態（LEVEL0）に下がった場合に、通知メッセージを出力します。

N

サーバの負荷レベルを変更しても通知メッセージを出力しません。

A

サーバの負荷レベルを変更した場合に、通知メッセージを出力します。

通知メッセージの出力タイミングは、loadcheck_interval オペランドによって指定する負荷監視インタバルの値によって異なります。負荷監視インタバルを指定していない場合は、30 秒を仮定します。

●ipc_backlog_count=コネクション確立要求を格納するキューの長さ

～〈符号なし整数〉((0～4096))

コネクション確立要求を格納するキューの長さ (listen システムコールのバックログ数) を指定します。

0 (デフォルト値) を指定した場合、listen システムコールのバックログ数に設定する実数値は OS によって異なります。デフォルト値を指定した場合の実数値については、「リリースノート」を参照してください。

実際のキューの長さは、指定した値より長くなる場合があります。

キューの長さの上限値、下限値は、各 OS によって異なります。OS によって、キューの長さの上限値、下限値が制限されている場合、設定した値が有効にならないことがあります。コネクション確立要求を格納するキューについての詳細は、OS のマニュアル、または、TCP/IP の文献を参照してください。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、システム共通定義の値を仮定します。

●rpc_buffer_pool_max=プーリングするバッファ数

～〈符号なし整数〉((1～64))

RPC によるメッセージ受信処理で、メッセージ組み立てなどでプーリングするバッファの数を指定します。

メッセージ受信処理で使用するバッファは、プロセス固有領域に確保し、バッファが不要となった時点でプーリングします。このとき、バッファプール内のプーリング数が、このオペランドで指定した数を超える場合は、新たにプーリングするバッファも含め、サイズのいちばん小さいバッファを解放します。

このオペランドにデフォルト値より小さな値を指定すると、メッセージ受信処理で使用するバッファの数が、このオペランドで指定した数を超えることになります。その場合の注意事項を次に示します。

- ・ 不足分のバッファの確保／解放を行うため、メッセージ受信処理性能が低下するおそれがあります。
- ・ 不足分のバッファの確保時にエラーが発生し、メッセージ受信処理が失敗するおそれがあります。

このオペランドは、ユーザサービスデフォルト定義でも指定できます。

ここで指定を省略した場合、ユーザサービスデフォルト定義の値を仮定します。

●schedule_delay_limit=スケジュール遅延限界経過時間

～〈符号なし整数〉((0～32767)) (単位：秒)

スケジュールキューによるスケジューリングの遅延限界経過時間を指定します。

スケジュールキューにサービス要求が登録されているにもかかわらず、サービス要求が取り出されない状態が、このオペランドに指定した時間を超えた場合、該当サーバごとに KFCA00838-W メッセージを出力します。この際、schedule_delay_abort オペランドで Y を指定している場合には、SCD デーモンが異常終了して OpenTP1 がシステムダウンします。

このオペランドを省略、または 0 を指定した場合、スケジュール遅延を監視しません。

スケジュールサービスは、10 秒のインタバルでスケジュール遅延を監視するため、スケジュール遅延を検知するのに最大 10 秒掛かることがあります。

このオペランドには、該当サーバの起動処理時間およびサービス処理時間よりも大きい値を指定してください。

このオペランドは、MHP、ソケット受信型 SPP、SUP に指定しても無視されます。

●schedule_delay_abort=Y|N

スケジュール遅延時にシステムダウンするかどうかを指定します。

Y

スケジュール遅延時にシステムダウンします。

N

スケジュール遅延時にシステムダウンしません。

スケジュールキューにサービス要求が登録されているにもかかわらず、サービス要求が取り出されない状態が、schedule_delay_limit オペランドに指定した時間を超えた場合、このオペランドで Y を指定すると、KFCA00839-E メッセージを出力したあと、SCD デーモンが異常終了してシステムダウンします。

schedule_delay_limit オペランドを省略、または 0 を指定した場合、このオペランドの指定は無視されます。

このオペランドは、MHP、ソケット受信型 SPP、SUP に指定しても無視されます。

●rap_autoconnect_con_error_msg=Y|N

rap クライアントでオートコネクトモードを使用していて、一度 rap サーバとのコネクションが確立されていても、以降の API の代理実行要求時、rap サーバとのコネクションの切断を検知した場合に、エラーメッセージを出力するかどうかを指定します。

このオペランドで Y を指定するか、省略した場合は、メッセージを出力します。

Y

API の代理実行要求時、rap サーバとのコネクションが切断されていた場合に、エラーメッセージを出力します。

N

API の代理実行要求時、rap サーバとの接続が切断されていた場合に、エラーメッセージを出力しません。

なお、このオペランドで抑止されるメッセージは次のとおりです。

- KFCA26971-E メッセージの理由コード=36

●core_shm_suppress=Y|N

コアファイルへの共用メモリダンプの出力を抑止するかどうかを指定します。

Y

コアファイルへの OpenTP1 共用メモリダンプの出力を抑止します。

N

コアファイルへの OpenTP1 共用メモリダンプの出力を抑止しません。

KFCA00105-E メッセージを出力して、プロセスが異常終了した場合、OS によっては、コアファイルにアタッチ中の共用メモリダンプを出力します。

このオペランドに Y を指定すると、OS によるコアファイルへの OpenTP1 共用メモリダンプの出力を抑止します。これによって、次の問題を回避できます。

- コアファイルのサイズ増大によるディスク容量の圧迫
- コアファイル出力時の I/O 占有によるマシン負荷の増大

このオペランドが有効になる前に、KFCA00105-E メッセージを出力して、プロセスが異常終了した場合は、コアファイルに、アタッチ中の OpenTP1 共用メモリダンプが出力されます（コアファイルに、共用メモリダンプの出力が行われる OS だけが該当）。

また、このオペランドに Y を指定した場合でも、CPU 時間監視が満了し、KFCA00105-E メッセージを出力して、プロセスが異常終了したときは、コアファイルに、アタッチ中の OpenTP1 共用メモリダンプが出力されます（コアファイルに、共用メモリダンプが出力される OS だけが該当）。

注

OpenTP1 の共用メモリダンプは、システムダウン時にだけ出力されます。そのため、UAP が異常を検知し、終了した場合、共用メモリダンプは出力されません。OS がコアファイルに共用メモリダンプを出力する場合、コアファイルに出力された共用メモリダンプが有効な資料になります。このオペランドに Y を指定すると、共用メモリダンプが出力されないため、トラブルシュートが困難になることがあります。

●xat_connect_resp_time=通信イベント処理用 SPP のアソシエーション確立の最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

プロトコルに OSI TP を使用する通信で、通信イベント処理用 SPP 内のアソシエーション確立要求 (dc_xat_connect 関数) の最大応答待ち時間を指定します。0 を指定した場合は、応答を無限に待ちます。

putenv 形式の XAT_CONNECT_RESP_TIME オペランドでも同じ内容を指定できます。このオペランドと putenv 形式の XAT_CONNECT_RESP_TIME オペランドの関係を次の表に示します。

表 3-19 set 形式の xat_connect_resp_time オペランドと putenv 形式の XAT_CONNECT_RESP_TIME オペランドの関係

set 形式の xat_connect_resp_time オペランドの指定	putenv 形式の XAT_CONNECT_RESP_TIME オペランドの指定	有効になる値 (単位：秒)
○	×	set 形式の xat_connect_resp_time オペランドの指定値
×	○	putenv 形式の XAT_CONNECT_RESP_TIME オペランドの指定値
○	○	set 形式の xat_connect_resp_time オペランドの指定値
×	×	180※

(凡例)

- ：指定します。
- ×

注※

set 形式の xat_connect_resp_time オペランド、および putenv 形式の XAT_CONNECT_RESP_TIME オペランドの両方の指定を省略した場合は、180 秒が仮定されます。

●scd_poolfull_check_interval=KFCA00853-E メッセージ出力インターバル時間
～ 〈符号なし整数〉 ((0～32767)) (単位：秒)

メッセージ格納バッファプールのメモリ不足が連続して発生した場合、KFCA00853-E メッセージを出力するインターバル時間を指定します。

メッセージ格納バッファプールのメモリ不足が scd_poolfull_check_interval オペランドで指定したインターバル時間中に scd_poolfull_check_count オペランドで指定した回数に達すると、KFCA00853-E メッセージが 1 回出力されます。

このオペランドで指定したインターバルは、メッセージ格納バッファプールのメモリ不足が発生した時点で開始されます。また、メッセージ格納バッファプールのメモリ不足がこのオペランドの指定値以上経過して発生した時点で再設定されます。

このオペランドに 0 を指定した場合は、メッセージ KFCA00853-E の出力を抑止します。

このオペランドは、キュー受信型 SPP にだけ指定できます。キュー受信型 SPP 以外のユーザサーバに指定しても無視されます。

●scd_poolfull_check_count=KFCA00853-E メッセージ出力判断値

～ 〈符号なし整数〉 ((1～32767)) (単位：回)

メッセージ格納バッファプールのメモリ不足が連続して発生した場合、KFCA00853-E メッセージを出力する判断値を指定します。

メッセージ格納バッファプールのメモリ不足が scd_poolfull_check_interval オペランドで指定したインタバル時間中に scd_poolfull_check_count オペランドで指定した回数に達すると、KFCA00853-E メッセージが出力されます。

このオペランドは、キュー受信型 SPP にだけ指定できます。キュー受信型 SPP 以外のユーザサーバに指定しても無視されます。

●scd_pool_warning_use_rate=警告メッセージの出力判断値になるメッセージ格納バッファプール使用率の上限値

～ 〈符号なし整数〉 ((0～99)) (単位：%)

警告メッセージを出力する判断値になるメッセージ格納バッファプールの使用率の上限値を指定します。

ユーザサービス定義の message_store_buflen オペランド，または scdbufgrp 定義コマンドで指定したメッセージ格納バッファプール長に占めるメモリの使用率を指定します。

警告メッセージ (KFCA00829-W) はサービス要求をスケジュールキューにキューイングした時点のメッセージ格納バッファプールの使用率を基に出力されます。一度表示されると使用率がこのオペランドの指定値を下回り，再びこのオペランドの指定値を上回らないかぎり出力されません。

このオペランドに 0 を指定した場合，または省略した場合には警告メッセージは出力されません。

スケジュールバッファグループによってメッセージ格納バッファプールを複数のユーザサーバで共用している場合は，共用している全ユーザサーバの使用サイズで使用率が計算されます。したがって，ある特定のユーザサーバの使用サイズがこのオペランドで指定した使用率を超えていない場合にも警告メッセージ (KFCA00829-W) が出力されることがあります。

このオペランドの指定に関係なく，メッセージ格納バッファプールが不足した場合は，警告メッセージ (KFCA00829-W) は出力されないで，エラーメッセージ (KFCA00854-E) が出力されます。

このオペランドは，キュー受信型 SPP にだけ指定できます。キュー受信型 SPP 以外のユーザサーバに指定しても無視されます。

●scd_pool_warning_interval=メッセージ格納バッファプール使用率超過時の警告メッセージ出力インタバル時間

～ 〈符号なし整数〉 ((0～32767)) (単位：秒)

メッセージ格納バッファプール使用率 (scd_pool_warning_use_rate オペランドで指定) が超過したときの警告メッセージ (KFCA00829-W) を出力するインタバル時間を指定します。

このオペランドを指定した場合、警告メッセージ (KFCA00829-W) が出力された時点からこのオペランドの指定時間が経過するまでの間は、メッセージ格納バッファプール使用率が、scd_pool_warning_use_rate オペランドの指定値を超えても警告メッセージ (KFCA00829-W) は出力されません。

●ipc_tcpnodelay=Y|N

OpenTP1 がノード間で使用する通信ソケット (INET ドメイン) に、TCP_NODELAY オプションを使用するかどうかを指定します。

TCP_NODELAY オプションを使用すると (このオペランドに Y を指定すると)、Nagle アルゴリズムが無効になるので、送信済みデータの応答待ちの状態でも遅延させることなくデータを送信できます。ただし、TCP_NODELAY オプションを使用することで、INET ドメイン通信時の送信効率が低下し、ネットワークの負荷が大きくなる場合があります。この場合は、ネットワークの帯域などを考慮して ipc_sendbuf_size オペランド、ipc_recvbuf_size オペランドのチューニングを検討するか、このオペランドに N を指定して TCP_NODELAY オプションを無効にするかを検討してください。

●stay_watch_queue_count=スケジュールキューの滞留監視判定を開始する際の判断になるサービス要求滞留数

～ 〈符号なし整数〉 ((0～32767))

スケジュールキューの滞留監視判定を開始する際の判断になるサービス要求滞留数を指定します。

このオペランドの指定値以上のサービス要求がスケジュールキューに滞留している場合、スケジュールキューの滞留監視判定が始まります。スケジュールキューの滞留監視を開始するサービス要求滞留数をチェックする間隔は、stay_watch_start_interval オペランドで指定します。スケジュールキューの滞留監視判定の開始以降は、stay_watch_check_interval オペランドで指定した間隔でサービス要求の滞留監視をします。サービス要求滞留数がこのオペランドの指定値よりも少なくなった場合、いったんスケジュールキューの滞留監視判定は終了します。サービス要求滞留数がこのオペランドで指定した値以上になると、再びスケジュールキューの滞留監視判定が始まります。

このオペランドに 0 を指定した場合、スケジュールキューの滞留監視をしません。また、このオペランドを、RAP、MHP、ソケット受信型 SPP、および SUP に指定しても無視されます。

スケジュールキューの滞留監視機能の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

●stay_watch_check_rate=スケジュールキューの滞留監視判定処理で使用するサービス要求の処理率

～ 〈符号なし整数〉 ((1～100)) (単位: %)

スケジュールキューの滞留監視判定で使用する、サーバが処理できるサービス要求の処理率を指定します。

スケジュールキューの滞留監視判定で、次の判定式を満たす場合には、該当サーバごとに KFCA00833-W メッセージを出力します。

スケジュールキューの滞留監視判定式

サービス要求の処理件数 < このオペランドの指定値 × スケジュールキューに滞留しているサービス要求数

スケジュールキューの滞留監視判定式を満たし、かつ stay_watch_abort オペランドで Y を指定している場合は、SCD デーモンが異常終了（アボートコード hclen001 を出力）して OpenTP1 が停止します。

stay_watch_queue_count オペランドを省略、または 0 を指定した場合、このオペランドの指定は無視されます。

このオペランドを、RAP、MHP、ソケット受信型 SPP、および SUP に指定しても無視されます。

スケジュールキューの滞留監視機能の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

●stay_watch_abort=Y|N

スケジュールキューの滞留監視判定式を満たした場合に、OpenTP1 システムをダウンさせるかどうかを指定します。

Y

スケジュールキュー滞留監視判定式を満たした場合に、KFCA00833-W メッセージ、および KFCA00834-E メッセージを出力します。そして、該当するユーザサーバを強制終了し、SCD デーモンを強制終了（アボートコード hclen001 を出力）させて OpenTP1 を停止させます。

N

スケジュールキュー滞留監視判定式を満たした場合に、KFCA00833-W メッセージを出力します。OpenTP1 システムはダウンしません。

stay_watch_queue_count オペランドを省略、または 0 を指定した場合、このオペランドの指定は無視されます。また、このオペランドを、RAP、MHP、ソケット受信型 SPP、および SUP に指定しても無視されます。

スケジュールキューの滞留監視機能の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

●stay_watch_start_interval=スケジュールキューの滞留監視インタバル時間

～ 〈符号なし整数〉 ((1～32767)) (単位：秒)

スケジュールキューに滞留しているサービス要求数を監視するインタバル時間を指定します。

stay_watch_queue_count オペランドを省略、または 0 を指定した場合、このオペランドの指定は無視されます。また、このオペランドを、RAP、MHP、ソケット受信型 SPP、および SUP に指定しても無視されます。

通常、SCD デーモンは 10 秒間隔で負荷監視をしています。このオペランドを指定した場合、SCD デーモンは、10、各ユーザサーバの `stay_watch_start_interval` オペランドの指定値、および各ユーザサーバの `stay_watch_check_interval` オペランドの指定値の最大公約数で負荷監視をします。OpenTP1 システムに与える影響を少なくするために、このオペランドには 5 の倍数を指定することをお勧めします。

スケジュールキューの滞留監視機能の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

●`stay_watch_check_interval`=スケジュールキューの滞留監視判定インタバル時間

～ 〈符号なし整数〉 ((1～65534)) (単位：秒)

スケジュール滞留監視判定式を基に判定処理を行うインタバル時間を指定します。

スケジュール滞留監視判定式による判定処理は、`stay_watch_queue_count` オペランドの指定値以上のサービス要求数がスケジュールキューに滞留した場合に開始されます。

`stay_watch_queue_count` オペランドを省略、または 0 を指定した場合、このオペランドの指定は無視されます。また、このオペランドを、RAP、MHP、ソケット受信型 SPP、および SUP に指定しても無視されます。

通常、SCD デーモンは 10 秒間隔で負荷監視をしています。このオペランドを指定した場合、SCD デーモンは、10、各ユーザサーバの `stay_watch_start_interval` オペランドの指定値、および各ユーザサーバの `stay_watch_check_interval` オペランドの指定値の最大公約数で負荷監視をします。OpenTP1 システムに与える影響を少なくするために、このオペランドには 5 の倍数を指定することをお勧めします。

スケジュールキューの滞留監視機能の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。

●`trn_completion_limit_time`=トランザクション完了限界時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

トランザクションブランチの開始から終了までの最大実行時間を指定します。指定時間を超えた場合、このトランザクションブランチのプロセスが異常終了したあとに、トランザクションブランチが回復プロセスによってコミットまたはロールバックのどちらかに決着して終了します。0 を指定した場合は、トランザクションブランチの最大実行時間を監視しません。

UAP が異常終了した場合に閉塞するかどうかは `hold` オペランド、`term_watch_time` オペランドの指定によります。詳細については、ユーザサービス定義の [hold オペランド](#) および [term_watch_time オペランド](#) の説明を参照してください。

このオペランドの監視対象区間は、`dc_trn_begin` 関数などの呼び出しやサービス関数の開始によるトランザクションの開始から、トランザクションの同期点処理終了情報 (TJ) の取得後であるトランザクションブランチの終了までです。ただし、トランザクションの最適化が行われた場合、サーバ側のトランザクションブランチに対する監視が終了するのは、クライアント側に応答を返したあとです。このオペランドの監

視対象区間の詳細および各種タイマ監視との関係については、「[付録 A.2 トランザクションの時間監視](#)」を参照してください。

ここで指定を省略し、ユーザサービスデフォルト定義でも省略した場合、トランザクションサービス定義の値を仮定します。

注

トランザクション完了限界時間の精度は秒単位です。そのため、タイミングによっては、このオペランドに指定した値よりも短い時間でプロセスを強制停止することがあります。トランザクション完了限界時間が小さくなるほど、誤差の影響を受けやすくなりますので、このオペランドには3（単位：秒）以上の値の指定を推奨します。

●rap_message_id_change_level=メッセージ ID の変更レベル

～ 〈符号なし整数〉 ((0～2))

リモート API 機能を使用する場合に出力される可能性の高いエラーメッセージでは、メッセージの種類の E を W に変更するために、メッセージ ID を変更できます。変更する内容によってレベルを指定してください。

メッセージログを監視し、メッセージの種類が E であるか W であるかによって管理方法を区別したい場合にこのオペランドを使用してください。

このオペランドによって、変更されるのはメッセージ ID だけであり、メッセージの内容は変更されません。

0

メッセージ ID を変更しないで、従来どおりのメッセージ ID で出力します。

1

特定の理由コードが出力される条件で、メッセージの種類 E を W に変更するために、メッセージ ID を変更できます。

2

1 を指定した場合に加え、メッセージの種類 E を W に変更するために、メッセージ ID を変更できます。

このオペランドの指定値と出力されるメッセージの対応については rap リスナーサービス定義の [rap_message_id_change_level](#) オペランドを参照してください。

●log_audit_out_suppress=Y|N

～ 〈N〉

このユーザサーバから出力される監査ログを抑止する場合に指定します。

Y

このユーザサーバから出力される監査ログを抑止します。

N

このユーザサーバから出力される監査ログを抑止しません。このオペランドはログサービス定義の log_audit_out オペランドに Y を指定した場合に有効です。

●log_audit_message=監査ログ取得メッセージ ID [、監査ログ取得メッセージ ID] …

～ 〈符号なし整数〉 ((33400～99999))

監査ログを取得する項目のメッセージ ID の番号を 33400～99999 の範囲で指定します。指定できるメッセージ ID は最大 2048 個です。

次に示す監査ログのメッセージ ID を指定できます。

- OpenTP1 が取得する監査ログのうちユーザサービス定義に指定できる監査ログのメッセージ ID
- UAP で任意に取得する監査ログのメッセージ ID

このオペランドで指定できるメッセージ ID については、「[付録 C 監査イベントを取得する定義](#)」を参照してください。

ここで指定を省略し、ユーザサービスデフォルト定義でも指定を省略した場合、ログサービス定義の値を仮定します。

このオペランドは、ログサービス定義の log_audit_out オペランドに Y を指定し、ユーザサービス定義の log_audit_out_suppress オペランドに N を指定した場合に有効です。

●mcf_prf_trace=Y|N

～ 〈Y〉

ユーザサーバごとに、MCF 性能検証用トレース情報を取得するかどうかを指定します。このオペランドの指定値を有効にするには、システムサービス共通情報定義の mcf_prf_trace_level オペランドに 00000001 を指定してください。

Y

MCF 性能検証用トレース情報を取得します。

N

MCF 性能検証用トレース情報を取得しません。

このオペランドは、ユーザサービスデフォルト定義でも指定できます。

指定値の優先度は、次のとおりです (1.>2.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義

オペランドの指定、または指定値に誤りがある場合は、ユーザサーバの開始処理中にユーザサーバが異常終了します。

ユーザサーバでの MCF 性能検証用トレース情報取得有無とオペランドの指定値の関係を、次の表に示します。

表 3-20 ユーザサーバでの MCF 性能検証用トレース情報取得有無とオペランドの指定値の関係

システムサービス共通情報定義 mcf_prf_trace_level オペランドの指定値	ユーザサービス定義 mcf_prf_trace オペランドの指定値	
	Y	N
00000000	取得しない	取得しない
00000001	取得する	取得しない

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

●scd_refresh_process=Y|N

このサービスグループのプロセスを、一つのサービス要求を処理するごとに起動し直すかどうか（非常駐 UAP プロセスのリフレッシュ機能を使用するかどうか）を指定します。

Y

一つのサービス要求を処理するごとに新たなプロセスを起動します。既存のプロセスは終了します。このオペランドに Y を指定すると、1 回のサービス要求で 1 つのプロセスを使用します。そのため、リエントラント構造ではない UAP でもサービスを要求できます。ただし、複数のサービス要求を処理する場合にプロセスの起動と停止がサービス要求数分発生するため、システム形態によってはサービス要求の処理性能が低下することがあります。

また、ユーザサービス定義の termed_after_service オペランド、および service_wait_time オペランドの指定は無視されます。

N

スケジュールキューにサービス要求が存在する場合は、同一プロセスで続けてサービス要求を処理します。

次の場合、このオペランドの指定は無視されます。

- 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 parallel_count オペランドの常駐プロセス数に 0 以外を指定した場合
- 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 balance_count オペランドに 0 以外を指定した場合
- SPP のソケット受信型サーバ、および SUP の場合

●rap_extend_function=リモート API 機能の機能拡張レベル

～ 〈16 進数字〉 ((00000000～00000001)) 《00000000》

リモート API 機能の機能拡張レベルを指定します。機能の拡張レベルを複数指定する場合、それぞれの指定値の論理和を指定してください。

00000000

リモート API 機能を機能拡張しません。

00000001

rap クライアントと rap サーバ間の dc_rpc_call 関数の処理でタイムアウト事象が発生した場合の dc_rpc_call 関数のリターン値を、DCRPCER_NET_DOWN (-306) から DCRPCER_TIMED_OUT (-307) に変更します。

リターン値が変更となるのは次の場合です。

- rap クライアントに指定された watch_time オペランドの指定時間内に、サービス要求先サーバからの応答が返らなかった場合
- rap クライアントからトランザクショナル RPC を実行した時点で rap クライアントに指定された trn_limit_time オペランドの指定値を満たした場合

●prc_coredump_filter=core ファイルに共有メモリを含めるかの判断値

～ 〈符号なし整数〉 ((1, 3))

core ファイルに共有メモリを含めるかどうかを設定します。

1

含めません

3

含めます

このオペランドはシステム共通定義、ユーザサービスデフォルト定義、およびユーザサービス定義に指定できます。OpenTP1 のユーザサーバ、システムサーバで有効にするためには次のように設定します。

有効にしたいサーバ	システム共通定義	ユーザサービスデフォルト定義	ユーザサービス定義
システムサーバ	○	×	×
ユーザサーバ	○	○	○

(凡例)

- ：含めます
- ×：含めません※

注※

システムサーバはサーバ単位に指定できません。また、ユーザサーバはサーバ単位に指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義

2. ユーザサービスデフォルト定義

3. システム共通定義

システム共通定義内のこのオペランドを変更した場合は、`dcreset` コマンドを実行してください。

このオペランドの指定値に従い、プロセス生成直後に OpenTP1 が `/proc/self/coredump_filter` を設定します。

`/proc/self/coredump_filter` の設定中にエラーが発生した場合は、KFCA00797-E メッセージが出力されます。

このオペランドを省略した場合は、OpenTP1 のプロセスに対して有効になっている OS の設定に従います。また、このオペランドは、Red Hat Enterprise Linux 5.2 以降に対して有効になります。対応 OS 以外の環境で指定した場合は無視されます。

OpenTP1 が提供する次のサービスでは、ユーザサービスデフォルト定義に指定した定義内容が有効になります。

- MQC ゲートウェイサービス
- リポジトリ管理サーバ
- リアルタイム統計情報サービス
- rap クライアントマネージャ、rap リスナー、および rap サーバ

注意事項

例えば、巨大な共有メモリを使用している環境で、core ファイルのサイズを極力抑えたい場合、このオペランドに 1 を指定することで core ファイルに共有メモリが含まれなくなり、core ファイルの大きさを抑えることができます。ただし、core ファイルに含まれている共有メモリは、その core を出力したプロセスがその瞬間参照していた共有メモリであって、トラブルシュートに大変有益な情報です。core を出力する状態自体が何か問題がある場合であり、この core に含まれている共有メモリ情報はその問題を解決するのに必要な情報です。

このオペランドに 1 を設定することで、そのトラブルシュートに必要な情報が一部失われ、問題解決に時間が掛かるおそれがあります。特に、システムサーバが core ファイルを出力する状態は、システム的に異常な状態になっているおそれが高く、その瞬間の共有メモリがトラブルシュートに不可欠なケースがあります。

これらのことを踏まえた上で、このオペランドの指定値を決めてください。

●watch_time=最大応答待ち時間

～〈符号なし整数〉((0～65535)) (単位：秒)

RPC によってプロセス間で通信する場合、サービス要求を送信してからサービスの応答が返るまでの待ち時間の最大値を指定します。

OpenTP1 の終了処理で、このオペランドで指定した時間だけ終了処理を待ち合わせる場合があります。したがって、大きな値を指定した場合、OpenTP1 の終了処理に時間が掛かる場合があります。

指定時間を過ぎても応答がない場合は、RPC は送受信タイムアウトとしてエラーリターンします。

0 を指定した場合は、応答を受信するまで待ち続けます。0 を指定した場合、OpenTP1 が終了しない場合があります。

このオペランドは、システム共通定義の watch_time オペランドのデフォルト値で使用してください。

特別なチューニングを必要とする場合以外は、このオペランドの内容を変更しないことをお勧めします。

システム共通定義の watch_time オペランドのデフォルト値よりも、極端に小さな値または大きな値を指定すると、OpenTP1 ダウンに至る障害が発生する場合がありますので、ご注意ください。

コマンド形式

次ページ以降に記述しています。

putenv 形式

●環境変数名 環境変数値

～〈文字列〉

このサービスグループのプロセスで、指定した環境変数に値を設定します。

COBOL の動作環境を OpenTP1 が起動するため、COBOL 環境の設定などに使用します。UAP の実行形式プログラムごとに、任意の環境変数を与えることができます。標準 C ライブラリ 'putenv' を参照してください。

ここで、PATH を設定するとプロセスサービス定義の prcsvpath オペランドおよび運用コマンド prcpath の指定は、このサービスグループに限り無効になり、PATH の値となります。PATH を設定する場合は、このサービスグループのロードモジュールが置かれているディレクトリも含めて設定してください。

なお、OpenTP1 では、'dc' で始まる環境変数名は使用しないでください。

●DCFPL_CONNECT_RETRY_COUNT コネクション確立処理をリトライする回数

～〈符号なし整数〉((8～2147483647))

リモート API 機能使用時の通信制御部のコネクション確立処理で、ECONNREFUSED エラーが発生した場合、コネクション確立処理をリトライする回数を指定します。

このオペランドに 8 以上の値を指定し、通信制御部のコネクション確立処理で ECONNREFUSED エラーが発生した場合、DCFPL_CONNECT_RETRY_INTERVAL オペランドに指定した時間を待ったあと、コネクション確立処理をリトライします。

なお、このオペランドの指定を省略した場合、または、8 より小さい値を指定した場合は、8 を仮定します。

●DCFPL_CONNECT_RETRY_INTERVAL コネクション確立処理のリトライ間隔

～ 〈符号なし整数〉 ((10～999)) (単位：ミリ秒)

リモート API 機能使用時の通信制御部のコネクション確立処理で、ECONNREFUSED エラーが発生した場合、コネクション確立処理のリトライ間隔をミリ秒単位で指定します。

なお、このオペランドの指定を省略した場合、または、10～999 の範囲外の値を指定した場合は、100 を仮定します。

●XAT_CONNECT_RESP_TIME 通信イベント処理用 SPP のアソシエーション確立の最大応答待ち時間

～ 〈符号なし整数〉 ((0～65535)) (単位：秒)

プロトコルに OSI TP を使用する通信で、通信イベント処理用 SPP 内のアソシエーション確立要求 (dc_xat_connect 関数) の最大応答待ち時間を指定します。0 を指定した場合は、応答を無限に待ちます。

set 形式の xat_connect_resp_time オペランドでも同じ内容を指定できます。このオペランドと set 形式の xat_connect_resp_time オペランドの関係については、表 3-14 を参照してください。

dcputenv 形式

●環境変数名 環境変数値

～ 〈文字列〉

このサービスグループのプロセスで、指定した環境変数に値を設定します。さらに、値に環境変数名を指定した場合、この環境変数の値も取得します。

なお、OpenTP1 では、'dc'で始まる環境変数名は使用しないでください。

名称

リソースマネージャ拡張子の指定

形式

```
[trnrmid -n リソースマネージャ名  
-i リソースマネージャ拡張子 [, リソースマネージャ拡張子] ...  
[-k] ]
```

機能

一つのリソースマネージャに対して、複数のリソースマネージャ拡張子を割り当てて、複数の制御単位としてアクセスする場合（トランザクションサービス定義の該当するリソースマネージャに対する trnstring 定義コマンドで -i オプションを指定した場合）、該当するユーザサーバからアクセスするリソースマネージャの拡張子を定義します。

トランザクションサービス定義の trnstring 定義コマンドで -i オプションを指定した場合、OpenTP1 は該当するリソースマネージャ名を「リソースマネージャ名+リソースマネージャ拡張子」に修飾して管理します。そのため、リソースマネージャ名だけでのアクセスはできません。必ずこのコマンドを定義してください。

OpenTP1 のトランザクションサービスでは、X/Open の XA インタフェースでリソースマネージャと連携してトランザクションを実行します。XA インタフェースではリソースマネージャが規定した xa_open 関数用、xa_close 関数用文字列を通知しなければなりません。

ユーザサーバが、OpenTP1 下のトランザクション内でリソースマネージャをアクセスする場合（トランザクション制御用オブジェクトをユーザサーバにリンクした場合）、そのリソースマネージャの xa_open 関数用文字列、および xa_close 関数用文字列を指定する必要があります。xa_open 関数、xa_close 関数は、トランザクションサービス定義の trnstring 定義コマンドで指定された該当するリソースマネージャ拡張子の文字列を引数として、ユーザサーバ起動・終了時、およびトランザクション回復処理時に OpenTP1 のトランザクションサービスが発行します。

-i オプションに指定するリソースマネージャ拡張子は、トランザクションサービス定義で指定した拡張子と同一の拡張子を指定してください。

一つのリソースマネージャに対して、複数のリソースマネージャ拡張子を割り当てて、複数の制御単位としてアクセスする必要がない場合（トランザクションサービス定義の該当するリソースマネージャに対する trnstring 定義コマンドで、-i オプションを省略した場合）は、このコマンドを定義する必要はありません。

一つの定義ファイル内に同一リソースマネージャ名で、同一リソースマネージャ拡張子の trnrmid 定義コマンドを複数指定した場合、最後に指定した定義が有効となります。

この定義は、OpenTP1 が提供するリソースの場合、OpenTP1_MCF だけ指定してください。

オプション

●-n リソースマネージャ名

～ 〈1～31 文字の識別子〉

このユーザサーバがアクセスするリソースマネージャ名（トランザクションサービス定義の `trnstring` 定義コマンドで指定したリソースマネージャ名）を指定します。OpenTP1 が提供するリソースの場合、OpenTP1_MCF だけ指定が必要です。ただし、OpenTP1_MCF でも次の場合には指定を省略できます。

- MCF マネージャ環境定義 `mcfmenv` の `id` オペランドを省略した場合、または `A` を指定した場合

●-i リソースマネージャ拡張子

～ 〈1～2 文字の識別子〉

このユーザサーバがアクセスするリソースマネージャ拡張子（トランザクションサービス定義の `trnstring` 定義コマンドで指定したリソースマネージャ拡張子）を指定します。

複数の拡張子を指定する場合は、拡張子と拡張子の間をコンマ（,）で区切ります。

トランザクションに関係する MCF を管理するために、MCF に対してのリソースマネージャ拡張子を指定しなければなりません。このオプションに指定する MCF のリソースマネージャ拡張子は、該当するユーザサーバがアクセスできる MCF のトランザクションサービス定義で指定した MCF のリソースマネージャ拡張子と同じ拡張子を指定してください。

●-k

-n オプションと -i オプションで指定したリソースマネージャ（リソースマネージャ名+リソースマネージャ拡張子）をリソースマネージャ接続先選択機能の対象とする場合に指定します。

このオプションを指定した場合は、トランザクションを開始する前に `dc_trn_rm_select` 関数を使用して接続対象となるリソースマネージャ（リソースマネージャ名+リソースマネージャ拡張子）を決定してください。`dc_trn_rm_select` 関数で接続対象のリソースマネージャを決定しないままトランザクションを開始すると、このオプションを指定したリソースマネージャへの接続はできません。

このオプションは SUP または SPP にだけ指定できます。MHP では指定できません。MHP に指定した場合は、該当するリソースマネージャに接続できなくなるため、注意してください。また、このオプションは OpenTP1 が提供するリソースには指定できません。指定した場合の動作は保証できません。

一つのユーザサーバの同一リソースマネージャでは、-k オプションの指定の有無を混在させないでください。

注意事項

ユーザサービス定義とユーザサービスデフォルト定義の両方で、`trnrmid` 定義コマンドに -i オプションを指定する場合は注意してください。例えば、次のように指定した場合、ユーザサービス定義とユーザサービスデフォルト定義の両方の指定が有効となります。

- ユーザサービス定義の指定：`trnrmid -n RM 名称 -i sl`

- ユーザサービスデフォルト定義の指定：trnrmid -n RM 名称 -i s2

名称

スケジュールバッファグループの指定

形式

```
scdbufgrp -g スケジュールバッファグループ名  
           { [-s メッセージ格納バッファ使用制限サイズ] |  
             [-p メッセージ格納バッファ使用制限率] }
```

機能

メッセージ格納バッファプールを共用するユーザサーバの集合をスケジュールバッファグループといいます。ユーザサーバは、スケジュールバッファグループごとにメッセージ格納バッファプールを共用します。

ユーザサービス定義では、このユーザサーバを所属させるスケジュールバッファグループを指定します。

また、このユーザサーバがメッセージ格納バッファプール内で使用するバッファサイズの上限を設定できます。

なお、ソケット受信型のユーザサーバでは、このコマンドを指定しても無視されます。また、キュー受信型のユーザサーバでは、ユーザサービス定義の message_store_buflen オペランド、および message_cell_size オペランドを指定しても無視されます。

オプション

●-g スケジュールバッファグループ名

～ 〈1～8 文字の識別子〉

このユーザサーバを所属させる、スケジュールバッファグループの名称を指定します。

スケジュールバッファグループ名は、スケジュールサービス定義の scdbufgrp 定義コマンドで指定した名称を指定してください。

●-s メッセージ格納バッファ使用制限サイズ

～ 〈符号なし整数〉 ((512～1610612736)) (単位：バイト)

このユーザサーバが使用するメッセージ格納バッファの使用制限サイズを指定します。-p オプションとは同時に指定できません。MHP の場合、この指定は無視されます。

この指定値を超えてメッセージをメッセージ格納バッファに格納しようとした場合、dc_rpc_call 関数は DCRPCER_NO_BUFS (バッファ不足) でエラーリターンします。

スケジュールバッファグループのメッセージ格納バッファサイズを超過した値は指定しないでください。超過した値を指定した場合、または指定可能範囲外の値を指定した場合、ユーザサービス起動時にエラーとなります。

スケジュールバッファグループに属するユーザサーバの、メッセージ格納バッファ使用制限サイズの総計が、スケジュールバッファグループのメッセージ格納バッファサイズを超えても問題ありません。

このオペランドの指定値には、スケジュールサービス定義の `scdbufgrp` 定義コマンドの `-l` オプションで指定するメッセージ格納バッファセル長（省略時は 512 バイト）の整数倍となる値を指定します。指定した値が `-l` オプションでの指定値の整数倍でない場合は、`-l` オプションでの指定値の整数倍に切り下げた値が仮定されます。切り下げた結果、値が 0 になった場合はユーザサーバの起動を中止します。

`-s` オプションで定義エラーが発生した場合、KFCA00830-E メッセージを出力し、ユーザサーバの起動を中止します。

●-p メッセージ格納バッファ使用制限率

～〈符号なし整数〉((1～100)) (単位：%)

このユーザサーバが使用するメッセージ格納バッファ使用制限率を指定します。このユーザサーバが所属する、スケジュールバッファグループのメッセージ格納バッファプール全体のサイズのうち、何%まで使用できるかを指定します。

`-s` オプションとは同時に指定できません。MHP の場合、この指定は無視されます。

この指定値を超えてメッセージをメッセージ格納バッファに格納しようとした場合、`dc_rpc_call` 関数は `DCRPCER_NO_BUFS`（バッファ不足）でエラーリターンします。

指定範囲外の値を指定した場合、ユーザサービス起動時にエラーとなります。

このオペランドの指定値により計算されたメッセージ格納バッファ使用制限サイズの値が、所属するスケジュールバッファグループのメッセージ格納バッファセル長の整数倍でない場合は、メッセージ格納バッファ使用制限サイズは整数倍に切り下げた値になります。切り下げた結果、値が 0 になった場合はユーザサーバの起動を中止します。

`-p` オプションで定義エラーが発生した場合、KFCA00830-E メッセージを出力し、ユーザサーバの起動を中止します。

注意事項

- `-s` オプションおよび `-p` オプションの指定値によって、メッセージ格納バッファにサービス要求が登録できない場合には、KFCA00879-W の警告メッセージを出力してほかの TP1 ノードへの再スケジュールを試みます。再スケジュール先が存在しない場合は、RPC 発行元に `DCRPCER_NO_BUFS(-304)` がリターンされます。
- ユーザサービスデフォルト定義に `scdbufgrp` 定義コマンドを指定した場合、すべてのユーザサービス定義の `message_store_bufllen` オペランドの指定値（メッセージ格納バッファプール長）が無視されます。

ユーザサービス定義の message_store_buflen オペランドとユーザサービスデフォルト定義の指定とメッセージ格納バッファプール長の関係を示します。

表 3-21 ユーザサービス定義の message_store_buflen オペランドとユーザサービスデフォルト定義の指定とメッセージ格納バッファプール長の関係

ユーザサービス定義		ユーザサービスデフォルト定義		メッセージ格納バッファプール長
message_store_buflen	scdbufgrp	message_store_buflen	scdbufgrp	
○	—	○	—	ユーザサービス定義の message_store_buflen オペランドの指定値
		—	○	ユーザサービスデフォルト定義に指定した、scdbufgrp 定義コマンドの共用バッファグループのメッセージ格納バッファプール長
—	○	○	—	ユーザサービス定義に指定した、scdbufgrp 定義コマンドの共用バッファグループのメッセージ格納バッファプール長
		—	○	
—	—	—	—	ユーザサービス定義の message_store_buflen オペランドのデフォルト値

(凡例)

- ：指定あり
- ：指定なし

名称

マルチスケジューラ機能の指定

形式

```
scdmulti [-g マルチスケジューラグループ名]
```

機能

マルチスケジューラ機能を使用する OpenTP1 システムで、該当するユーザサーバが使用するマルチスケジューラに関する情報を指定します。

マルチスケジューラ機能を使用する場合、この定義コマンドとともに次の定義コマンドおよびオペランドを指定する必要があります。

RPC 受信側：スケジューラサービス定義の scdmulti 定義コマンド

RPC 送信側：ユーザサービス定義の multi_schedule オペランド

オプション

●-g マルチスケジューラグループ名

～ 〈1～8 文字の識別子〉《scdmiltgp》

該当するユーザサーバが使用するマルチスケジューラグループの名称を指定します。

スケジューラサービス定義の scdmulti 定義コマンドの-g オプションに指定したマルチスケジューラグループ名を指定します。指定された名称がスケジューラサービス定義の scdmulti 定義コマンドの-g オプションで指定されていない場合は、ユーザサーバの起動を中止します。

このオプションを省略した場合、およびユーザサービスデフォルト定義の scdmulti 定義コマンドの-g オプションの指定を省略した場合、"scdmiltgp"が指定されたものとします。したがって、"scdmiltgp"はマルチスケジューラグループ名として使用しないでください。

注意事項

TP1/Client からマルチスケジューラ機能を使用してサービスを要求する場合については、マニュアル「OpenTP1 クライアント使用の手引」を参照してください。

この定義コマンドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

名称

サービス単位にスケジュールサービスの動作を指定

形式

```
[scdsvcdef  [-c サービス名]
             [-p 同時実行可能なサービス数]
             [-n キューイング可能なサービス要求数]
             [-l キューイング可能なメッセージ格納バッファプール長] ]
```

機能

SPP がスケジュールキューにサービス要求を登録、または取り出す場合の動作をサービス単位に指定できます。

SPP のソケット受信型サーバ、SUP、および MHP の場合は指定する必要はありません。指定しても無視されます。

オプション

●-c サービス名

～ 〈1～31 文字の識別子〉

サービス単位での動作を指定するサービス名称を指定します。

ユーザサービス定義の service オペランドに指定したサービス名を指定してください。service オペランドに指定されていないサービス名を指定した場合は、サービス単位のスケジュール制御は行いません。

このオプションを省略した場合は、すべてのサービスに対してサービス単位での動作を指定します。なお、サービス名称を指定した定義オペランドと重複して指定された場合は、サービス名称を指定した定義オペランドが優先されます。このオプションで指定したサービス名称が重複している場合は、最後に指定された定義オペランドが優先されます。

●-p 同時実行可能なサービス数

～ 〈符号なし整数〉 ((1～1024))

-c オプションで指定したサービス名に対するサービス要求について、同時実行可能なサービス数の上限値を指定します。

このオプションは、SPP の最大プロセス数（ユーザサービス定義の parallel_count オペランドまたは scdchprc コマンドで指定）よりも小さい値の場合に有効になります。このオプションを省略した場合、同時実行可能なサービス数の上限値はチェックしません。

●-n キューイング可能なサービス要求数

～〈符号なし整数〉((1～65535))

-c オプションで指定したサービス名に対するサービス要求について、スケジュールキューにキューイング可能なサービス要求数を指定します。

このオプションは、SPP のメッセージ格納バッファプールにキューイング可能なサービス要求数よりも小さい値の場合に有効になります。このオプションを省略した場合、キューイング可能なサービス要求数はチェックしません。

●-l キューイング可能なメッセージ格納バッファプール長

～〈符号なし整数〉((512～31457280)) (単位：バイト)

-c オプションで指定したサービス名に対するサービス要求について、スケジュールキューにキューイング可能なメッセージ格納バッファプール長を指定します。

このオプションは、SPP のメッセージ格納バッファプール長（ユーザサービス定義の message_store_bufilen オペランドまたは scdbufgrp 定義コマンドで指定）よりも小さい値の場合に有効になります。このオプションを省略した場合、キューイング可能なメッセージ格納バッファプール長はチェックしません。

注意事項

-n オプションおよび-l オプションの指定値によって、スケジュールキューにサービス要求が登録できない場合には、KFCA00831-W のエラーメッセージを出力してほかの TP1 ノードへの再スケジュールを試みます。再スケジュール先が存在しない場合は、RPC 発行元に DCRPCER_NO_BUFS(-304)がリターンされます。

ユーザサービス定義の指定値と UAP との関係

ユーザサービス定義の指定値と SPP, SUP, および MHP との関係を次の表に示します。

表 3-22 ユーザサービス定義の指定値と SPP, SUP, および MHP との関係

オペランド	SPP	SUP	MHP
service_group	○	×	○
module	○	○	○
service	○	×	○
nice	○	○	○
parallel_count	○※1	×	○
hold	○※1	×	×
hold_recovery	○※1	×	×
deadlock_priority	○	○	○
schedule_priority	○※1	×	○
message_bufilen	○※1	×	○
message_store_bufilen	○※1	×	○
trn_expiration_time	○	○	○
trn_expiration_time_suspend	○	○	○
watch_next_chain_time	○	○	○
atomic_update	○	○	Y
receive_from	queue※2	none	queue
uap_trace_max	○	○	○
uap_trace_file_put	○	○	○
term_watch_time	○※3	○※4	○
mcf_jnl_buff_size	○※5	×	○
type	other	×	MHP
balance_count	○※1	×	○
uid	○	○	○
auto_restart	×※6	○	×
critical	○	○	○
lck_wait_priority	○	○	○

オペランド	SPP	SUP	MHP
mcf_psv_id	○※7	×	×
trn_cpu_time	○	○	○
service_hold	○※1	×	×
service_priority_control	○※1	×	×
message_cell_size	○※1	×	×
max_socket_msg	×	×	×
max_socket_msglen	×	×	×
trf_put	○	○	○
mcf_mgrid	○※7	×	×
mcf_service_max_count	○※5	×	○
trn_statistics_item	○	○	○
node_down_restart	○	○	○
rpc_response_statistics	○	○	○
server_type	○	×	×
trn_rm_open_close_scope	○	○	○
trn_optimum_item	○	○	○
purge_msgget	○※8	×	○※8
cancel_normal_terminate	○	○	○
prc_abort_signal	○	○	○
rpc_service_retry_count	○	×	×
rpc_extend_function	○	○	○
max_socket_descriptors	○	○	○
max_open_fds	○	○	○
service_term_watch_time	○※1	×	×
termed_after_service	○※1	×	○
xat_trn_expiration_time	○	○	×
xat_osi_usr	○	○	×
rpc_trace	○	○	○
rpc_trace_name	○	○	○
rpc_trace_size	○	○	○

オペランド	SPP	SUP	MHP
trn_rollback_information_put	○	○	○
schedule_method	○※1	×	○
service_wait_time	○※1	×	○
mcf_spp_oj	○	×	×
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_timedout	○	○	○
status_change_when_terming	○	○	○
service_expiration_time	○	×	×
multi_schedule	○	○	×
make_queue_on_starting	○※8	×	○※8
loadcheck_interval	○※1	×	×
levelup_queue_count	○※1	×	×
leveldown_queue_count	○※1	×	×
ipc_sockctl_highwater	○	○	○
ipc_sockctl_watchtime	○	○	○
ipc_conn_interval	○	○	○
ipc_send_interval	○	○	○
ipc_send_count	○	○	○
ipc_header_rcv_time	○	○	○
rpc_send_retry_count	○	○	○
rpc_send_retry_interval	○	○	○
ipc_rcvbuf_size	○	○	○
ipc_sendbuf_size	○	○	○

オペランド	SPP	SUP	MHP
ipc_listen_sockbufset	○	○	○
polling_control_data	○	○	○
thread_yield_interval	○	○	○
groups	○	○	○
loadlevel_message	○※1	×	×
ipc_backlog_count	○	○	○
rpc_buffer_pool_max	○	○	○
schedule_delay_limit	○	×	×
schedule_delay_abort	○	×	×
rap_autoconnect_con_error_msg	○	○	×
core_shm_suppress	○	○	○
xat_connect_resp_time	○※9	×	×
scd_poolfull_check_interval	○※1	×	×
scd_poolfull_check_count	○※1	×	×
scd_pool_warning_use_rate	○※1	×	×
scd_pool_warning_interval	○※1	×	×
ipc_tcpnodelay	○	○	○
stay_watch_queue_count	○※1	×	×
stay_watch_check_rate	○※1	×	×
stay_watch_abort	○※1	×	×
stay_watch_start_interval	○※1	×	×
stay_watch_check_interval	○※1	×	×
trn_completion_limit_time	○	○	○
rap_message_id_change_level	○	○	×
log_audit_out_suppress	○	○	○
log_audit_message	○	○	○
mcf_prf_trace	○※5	×	○
scd_refresh_process	○	×	○
rap_extend_function	○	○	○
prc_coredump_filter	○	○	○

オペランド	SPP	SUP	MHP
watch_time	○	○	○
putenv	○	○	○
putenv DCFPL_CONNECT_RETRY_COUNT	○	○	×
putenv DCFPL_CONNECT_RETRY_INTERVAL	○	○	×
XAT_CONNECT_RESP_TIME	○※9	×	×
dcputenv	○	○	○
trnrmid	○※10	○※10	○
scdbufgrp	○※11	×	○
scdmulti	○	×	×
scdsvcdef	○※1	×	×

(凡例)

○：指定できます。

×：指定する必要はありません。

Y：Y を指定します。

queue：queue を指定します。

none：none を指定します。

other：other を指定します。

MHP：MHP を指定します。

注※1

SPP のソケット受信型サーバでは、指定する必要はありません。

注※2

SPP のソケット受信型サーバでは、socket を指定します。

注※3

hold オペランドに N を指定した場合だけ有効です。

SPP のソケット受信型サーバでは、hold オペランドの指定には関係なく、auto_restart オペランドに Y を指定した場合だけ有効です。

注※4

auto_restart オペランドに Y を指定した場合だけ有効です。

注※5

TP1/Message Control の機能を使用する SPP にだけ有効です。

注※6

SPP のソケット受信型サーバでは、指定できます。

注※7

アプリケーション起動機能を使用する SPP にだけ有効です。

注※8

parallel_count オペランドの常駐プロセス数に 0 を指定した場合だけ有効です。

注※9

通信イベント処理用 SPP にだけ有効です。

注※10

-k オプションは SUP または SPP にだけ有効です。

注※11

-s オプション，および-p オプションは SPP にだけ有効です。

4

ネットワークコミュニケーション定義の概要

ネットワークコミュニケーション定義の概要と定義の種類について説明します。

4.1 概要

4.1.1 ネットワークコミュニケーション定義とファイル名

ネットワークコミュニケーション定義は、テキストエディタを使用して定義ソースファイルを作成し、定義ソースファイルから定義オブジェクトファイルを生成します。この定義オブジェクトファイルを OpenTP1 システムに登録します。

定義オブジェクトファイルの名称は、表 4-1 および表 4-2 の形式に従って指定します。そのファイル名を、表 4-1 および表 4-2 の定義コマンドであらかじめ登録しておきます。

定義ソースファイルの詳細については、「[5. ネットワークコミュニケーション定義の詳細](#)」を参照してください。

定義オブジェクトファイルの生成については、「[4.3 定義オブジェクトファイルの生成](#)」を参照してください。

表 4-1 ネットワークコミュニケーション定義に登録するファイル名（その 1）

定義名	定義ソースファイルの数	定義オブジェクトファイル名※1	指定する定義コマンド名（定義名）
MCF マネージャ定義	各定義につき一つ	\$DCCONFPATH/_mu.....※2	dcsvstart -m（システムサービス構成定義）
MCF 通信構成定義	各定義につき二つ	\$DCCONFPATH/_mu.....※2	mcfmcname -s（MCF マネージャ定義）
MCF アプリケーション定義	各定義につき一つ	\$DCCONFPATH/任意	mcftenv -a（MCF 通信構成定義）

注※1

ノード内で一意となる名称を指定してください。

注※2

先頭 3 文字が'_mu'で始まるファイル名を指定してください。

表 4-2 ネットワークコミュニケーション定義に登録するファイル名（その 2）

定義名	定義ファイル名（完全パス名）	指定する定義コマンド名（定義名）
システムサービス情報定義	\$DCDIR/lib/sysconf/定義ファイル名※1	mcfmcname -s（MCF マネージャ定義）
システムサービス共通情報定義	\$DCDIR/lib/sysconf/mcf※2	指定する定義コマンドはありません。
MCF 性能検証用トレース定義	\$DCCONFPATH/_mc	指定する定義コマンドはありません。

注※1

定義ファイル名には、システムサービス情報定義の module オペランドで指定する実行形式プログラム名（先頭 4 文字が'mcfu'で始まる定義ファイル名）を指定してください。

注※2

標準値が定義されているファイルが、あらかじめ用意されています。必要に応じて定義値を変更してください。

4.1.2 MCF サービスと定義との関係

OpenTP1 から見た場合、MCF サービスは一つだけです。

一方、MCF サービス内では、MCF サービスは次の複数のプロセスから構成されます。

MCF マネジャプロセス（一つ）：

共用メモリの初期化、および 1 オンラインシステム内で生成されるすべての MCF 通信プロセスとアプリケーション起動プロセスの管理・運用をします。

MCF 通信プロセス（一つまたは複数またはなし※）：

プロトコルに依存するメッセージを制御します。プロトコル対応に起動します。

アプリケーション起動プロセス（一つまたは複数またはなし※）：

プロトコルに依存しないメッセージを制御します。次の場合に起動します。

- アプリケーション起動機能を使用する場合
- エラーイベント処理用 MHP を起動する場合
- リトライ指示のロールバック要求をする場合

MCF オンラインコマンドプロセス（一つまたはなし）：

560 端末や XP 端末から入力されたオンラインコマンドを処理します。オンラインコマンドを使用する場合に起動します。

注※

MCF プロセスか、アプリケーション起動プロセスのどちらか一方だけの構成もできます。

各ネットワークコミュニケーション定義は、次のような対応で作成します。

MCF マネジャ定義

MCF マネジャプロセスに対して一つ

MCF 通信構成定義

一つの MCF 通信プロセスに対して一つ

一つのアプリケーション起動プロセスに対して一つ

MCF アプリケーション定義

複数の MCF 通信プロセスに対して一つ

または、一つの MCF 通信プロセスに対して一つ

複数のアプリケーション起動プロセスに対して一つ
または、一つのアプリケーション起動プロセスに対して一つ

MCF 通信プロセスおよびアプリケーション起動プロセスを起動したい場合は、対応する MCF 通信構成定義を作成しなければなりません。起動したいプロセスに対応する、MCF 通信構成定義を作成してください。

なお、各ネットワークコミュニケーション定義とその出力オブジェクトファイル名を次の表に示します。

表 4-3 各ネットワークコミュニケーション定義とその出力オブジェクトファイル名

定義名	出力オブジェクトファイル名
MCF マネージャ定義	MCF サービス名
MCF 通信構成定義	MCF 通信サービス名 アプリケーション起動サービス名

4.1.3 MCF の運用形態と定義との関係

MCF の運用を円滑にするために、MCF アプリケーション定義は、各 MCF 通信構成定義とは別のファイルに作成します。

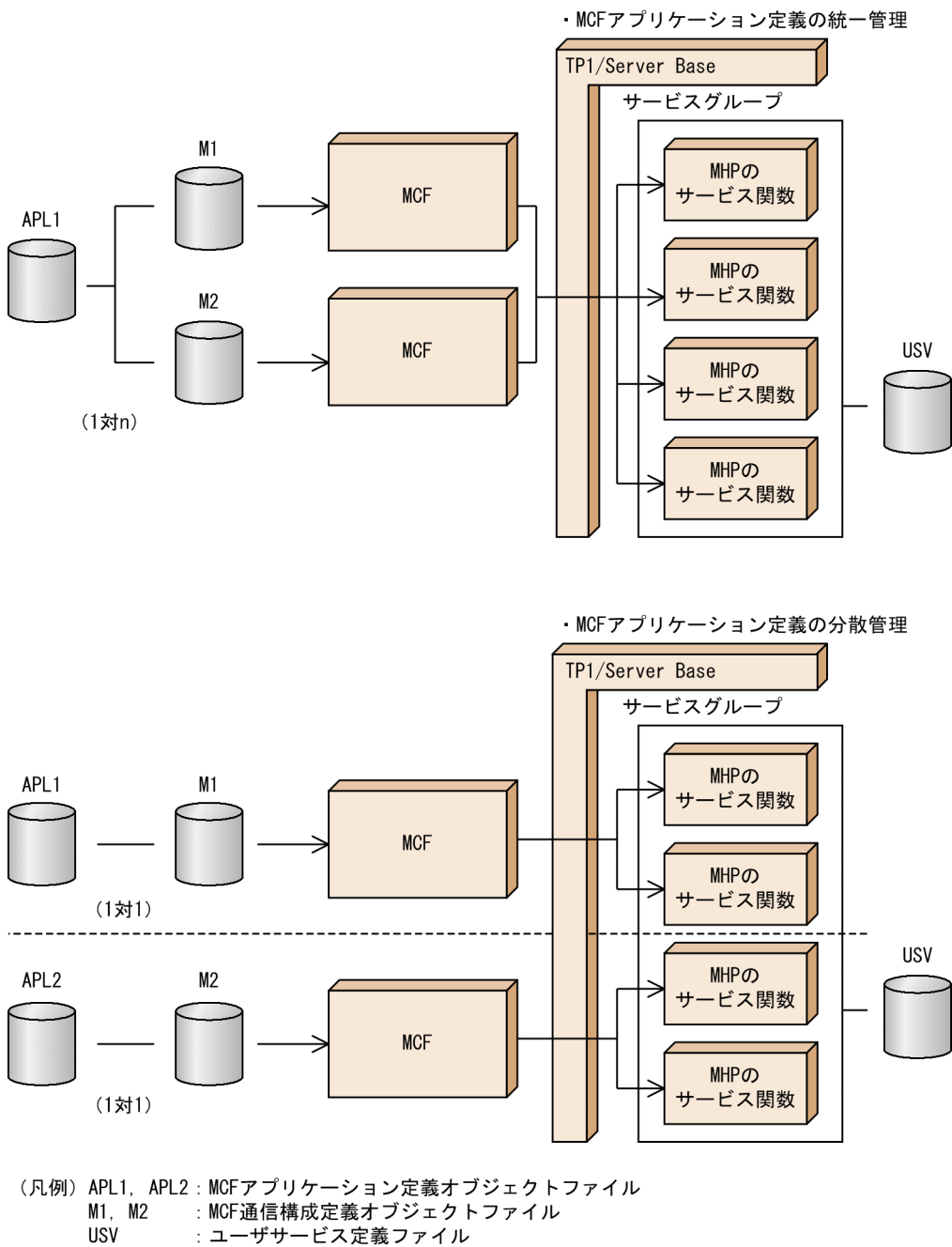
MCF の運用形態では、MCF アプリケーション定義と各 MCF 通信構成定義とは 1 対 n（n は 1 以上の整数）の関係で定義します。

MCF の運用形態と定義オブジェクトファイルとの関係を図 4-1 に示します。

図 4-1 の上図のような運用形態の場合は、MCF アプリケーション定義ファイルは統一して管理してください。

図 4-1 の下図のような運用形態の場合は、MCF アプリケーション定義ファイルは MCF ごとに分散して管理してください。

図 4-1 MCF の運用形態と定義ファイルとの関係



4.1.4 ネットワークコミュニケーション定義とシステムサービス定義との関連

(1) システム環境定義

全 MCF プロセスで使用する共用メモリの大きさを算出し、システム環境定義であらかじめ指定しておきます。

(2) メッセージキューサービス定義

MCF の入出力キュー定義コマンドのキューグループ ID は、メッセージキューサービス定義のキューグループ ID と同じ指定をします。

(3) ユーザサービス定義

MCF のアプリケーション属性定義コマンドのサービスグループ名およびサービス名は、ユーザサービス定義のサービスグループ名およびサービス名と同じ指定をします。

(4) システムサービス構成定義

MCF サービス名は、OpenTP1 が管理する MCF 識別名です。

MCF サービス名は、MCF 定義オブジェクト生成ユーティリティ (mcfmng) で生成する MCF マネージャ定義の出力オブジェクトファイル名です。

システムサービス構成定義の MCF サービス名にこの名称を指定します。

(5) ネームサービス定義

ネームサービスに登録できるサービス数を、論理端末数、UAP 数などを考慮して、ネームサービス定義であらかじめ指定しておきます。

4.1.5 アプリケーション名とサービス名との関係

MCF は、入力メッセージに入っているアプリケーション名を基に、MCF アプリケーション定義で指定した「サービスグループ名+サービス名」への変換を行います。

アプリケーション名は、業務を識別するコードに相当します。

サービスグループ名は、同系統の仕事の単位（サービス）をまとめたものです。

サービス名は、UAP の入り口点名です。

アプリケーション名は、ノード内で一意となる名称を指定します。

一つのアプリケーション名に対して、複数のサービス名を指定できません。逆に、一つのサービス名に対して、複数のアプリケーション名を指定できます。

4.1.6 ネットワークコミュニケーション定義のコマンド

ネットワークコミュニケーション定義のコマンド一覧と指定数を、次の表に示します。

表 4-4 ネットワークコミュニケーション定義のコマンド一覧と指定数

定義名		コマンド名	指定数
MCF マネジャ定義		mcfmenv (MCF マネジャ環境定義)	1
		mcfmcomn (MCF マネジャ共通定義)	1
		mcfmcname (通信サービス定義)	1～239
		mcfmuap (UAP 共通定義)	1
		mcfmqgid (入出力キュー定義)	0～4096
		mcfmexp (拡張予約定義)	1 または 0※1
		mcfmsts (状態引き継ぎ定義)	0～1
		mcfmsmsg (ログメッセージ出力抑止定義)	0～1
		mcfmsvg (サービスグループ属性定義)	0～8192
MCF 通信構成定義	共通定義	mcftenv (MCF 環境定義)	1
		mcftcomn (MCF 通信構成共通定義)	1
		mcfttred (最大処理多重度定義)	0～1
		mcfttim (タイマ定義)	0～1
		mcfttrc (トレース環境定義)	1
		mcftsts (状態引き継ぎ定義)	0～1
		mcftbuf (バッファグループ定義)	1～512 または 0※2
	プロトコル固有定義	省略※3	省略※3
	アプリケーション起動定義	mcftpsvr (アプリケーション起動環境定義の開始)	1
		mcftalcle (アプリケーション起動用論理端末定義)	0～2048
		mcftped (アプリケーション起動環境定義の終了)	1
MCF アプリケーション定義		mcfaenv (アプリケーション環境定義)	1
		mcfaalcap (アプリケーション属性定義)	1～8192

注※1

mcfmqgid の定義がある場合は 1, ない場合は 0 になります。

注※2

MCF 通信プロセスの場合は 1～512, アプリケーション起動プロセスの場合は 0 になります。

注※3

プロトコル固有定義については, マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

4.2 定義の種類

4.2.1 MCF マネージャ定義の内容

MCF マネージャ定義の内容を、次の表に示します。

表 4-5 MCF マネージャ定義の内容

形式	コマンド	オプション	オペランド	定義内容	指定値
コマンド	mcfmenv	-m	id	MCF マネージャプロセス識別子	〈英字〉 ((A~Z, a~z)) 《A》
			name	MCF マネージャ名	〈1~8 文字の識別子〉
	mcfmcomn	-n		出力通番を使用する論理端末数	〈符号なし整数〉 ((0~2048)) 《0》
		-p		MCF 作業領域長	〈符号なし整数〉 ((100~1945599)) (単位: キロバイト)
		-i		MCF 作業領域を追加確保した場合に, KFCA10242-I メッセージを出力するかどうかを指定	inc 《msg》
		-j		MCF マネージャプロセスのジャーナルバッファの大きさ	〈符号なし整数〉 ((4096~4000000 の 4 の倍数)) 《4096》 (単位: バイト)
		-o	cmdsename	MCF オンラインコマンドサービス名	〈1~8 文字の英数字〉
		-w	stats	MCF 稼働統計情報を取得するかどうかを指定	yes 《no》
	mcfmcname	-s	mcfsename	MCF 通信サービス名	〈1~8 文字の英数字〉
			syssename	システムサービス情報定義ファイル名	〈1~8 文字の識別子〉
	mcfmuap	-d		通信関数の発行回数の上限值	〈符号なし整数〉 ((0~65535)) 《0》 (単位: 回)
		-t	sndtim	同期型送信監視時間	〈符号なし整数〉 ((0~65535)) 《0》 (単位: 秒)
			sndrcvtim	同期型送受信監視時間	〈符号なし整数〉

形式	コマンド	オプション	オペランド	定義内容	指定値
コマンド	mcfmuap	-t	sndrcvtim	同期型送受信監視時間	((0~65535))《0》 (単位：秒)
			recvtim	同期型受信監視時間	〈符号なし整数〉 ((0~65535))《0》 (単位：秒)
		-j		ユーザサービスのジャーナルバッファの大きさ	〈符号なし整数〉 ((4096~4000000 の 4 の倍数))《32768》 (単位：バイト)
		-e	segsz	エラーイベント処理用 MHP 起動時、またはアプリケーション起動機能使用時の最大セグメント長	〈符号なし整数〉 ((512~2147483647)) 《32768》 (単位：バイト)
		-l	initseq	出力通番の初期値	〈符号なし整数〉 ((0~2147483647))《1》
			maxseq	ラップ時の出力通番の最大値	〈符号なし整数〉 ((0~2147483647)) 《65535》
			minseq	ラップ後の出力通番の開始値	〈符号なし整数〉 ((0~1))《1》
		-u	ntmetim	非トランザクション MHP 限界経過時間	〈符号なし整数〉 ((0~65535))《0》 (単位：秒)
		-a	delaytim	アプリケーション起動時の遅延許容時間	〈符号なし整数〉 ((0~360))《0》 (単位：分)
		-c	order	メッセージ送信順序および再送順序ならびにアプリケーション起動順序の選択	function 《commit》
			noansreply	非応答型の MHP からの問い合わせ応答をするかどうかを指定	《yes》 no
			commitdml	データ操作言語の COMMIT 文を使用するかどうかを指定	yes 《no》
			otqinhold	出力キューの入力保留中の動作を指定	《commit》 function
			errevt_recovery	OpenTP1 再開始時のエラーイベントを引き継ぐかどうかを指定	《yes》 no

形式	コマンド	オプション	オペランド	定義内容	指定値
コマンド	mcfmuap	-r	reschedulecnt	アプリケーション異常終了時の再スケジュール回数	〈符号なし整数〉 ((0～3)) 《0》 (単位：回)
			rescheduleint	アプリケーション異常終了時の再スケジュール間隔	〈符号なし整数〉 ((0～3)) 《0》 (単位：秒)
			reschedulelog	異常終了したアプリケーションを再スケジュールするときに、ログメッセージを出力するかどうかを指定	yes 《no》
	mcfmqgid	-q	quekind	キュー種別	itq otq
			quegrpid	キューグループ ID	〈1～8 文字の識別子〉
	mcfmexp	-g		サービスグループの登録数	〈符号なし整数〉 ((1～8192))
		-l		論理端末の登録数	〈符号なし整数〉 ((1～16380))
		-i		入力用ディスクキューが使用できないときに縮退運転するかどうかを指定	dg 《ndg》
		-o		出力用ディスクキューが使用できないときに縮退運転するかどうかを指定	dg 《ndg》
	mcfmsts	-g		サービスグループ数上限値	〈符号なし整数〉 ((1～8192))
		-v		サービス数上限値	〈符号なし整数〉 ((1～1044480))
	mcfmsmsg	-m		ログファイルへの出力を抑止するメッセージ ID	〈符号なし整数〉 ((10000～19999, 28000～29999))
	mcfmsvg	-g	servgrpn	サービスグループ名	〈1～31 文字の識別子〉
		-w	watchcnt	入力キューの滞留監視数	〈符号なし整数〉 ((0～65535)) 《0》
			watchint	入力キューの滞留監視インタバル時間	〈符号なし整数〉 ((5～3600)) 《10》 (単位：秒)
			expectcnt	MHP に期待するサービス要求の処理数	〈符号なし整数〉 ((1～65535))

形式	コマンド	オプション	オペランド	定義内容	指定値
コマンド	mcfmsvg	-w	abort	MHP の処理能力不足時の OpenTP1 システムのダウン可否	yes 《no》

4.2.2 MCF 通信構成定義の内容

MCF 通信構成定義の内容を、次の表に示します。

表 4-6 MCF 通信構成定義の内容

形式	コマンド	オプション	オペランド	定義内容	指定値
コマンド	mcftenv	-s		MCF 通信プロセス識別子、またはアプリケーション起動プロセス識別子	〈数字 (0~9), a~f〉 ((01~ef))
		-m		MCF マネジャ識別子	〈英字〉 ((A~Z, a~z)) 《A》
		-a		MCF アプリケーション定義オブジェクトファイル名	〈1~8 文字の識別子〉
		-q	diskitq	入力キューにディスクキューを使用するかどうかを指定	《yes》 no
	mcftcomn	-j		MCF 通信プロセス、またはアプリケーション起動プロセスのジャーナルバッファの大きさ	〈符号なし整数〉 ((4096~4000000 の 4 の倍数)) 《32768》 (単位: バイト)
	mcfttred	-m		最大処理多重度	〈符号なし整数〉 ((1~1000)) 《10》
	mcfttim	-t	btim	時間監視間隔	〈符号なし整数〉 ((1~60)) 《1》 (単位: 秒)
			mtim	未処理送信メッセージ滞留時間	〈符号なし整数〉 ((60~65535)) 《180》 (単位: 秒)
			rmtim	未処理受信メッセージ滞留時間	〈符号なし整数〉 ((0~65535)) 《0》 (単位: 秒)
		-p	usertime	ユーザタイム監視機能を使用するかどうかを指定	yes 《no》
			timereqno	最大タイム監視要求数	〈符号なし整数〉

形式	コマンド	オプション	オペランド	定義内容	指定値
コマンド	mcfttim	-p	timereqno	最大タイマ監視要求数	((1～10000))《16》
			msgsize	最大メッセージ長	〈符号なし整数〉 ((0～256))《0》 (単位：バイト)
			msgout	メッセージを出力するかどうかを指定	yes 《no》
	mcfttrc	-t	size	MCF トレースバッファの大きさ	〈符号なし整数〉 ((4096～15728640 の 4 の 倍数))《204800》 (単位：バイト)
			disk	MCF トレースディスク出力機能 を使用するかどうかを指定	《yes》 no
			bufcnt	MCF トレースバッファの数	〈符号なし整数〉 ((10～2147483647)) 《100》
			trccnt	MCF トレースファイルの数	〈符号なし整数〉 ((3～99))《3》
			msgsize	トレースとして取得する送受信 メッセージの最大サイズ	〈符号なし整数〉 ((0～1073741824))《128》 (単位：バイト)
		-m		MCF トレースファイルの数を超 えたときの処置	《del》 off
	mcftsts	-a		アプリケーション数上限値	〈符号なし整数〉 ((1～8192))
		-l		論理端末数上限値	〈符号なし整数〉 ((1～2048))
	mcftbuf	-g	groupno	メッセージ送受信用, およびメッ セージ編集用バッファグループ 番号	〈符号なし整数〉 ((1～512))
			length	メッセージ送受信用, およびメッ セージ編集用バッファ長	〈符号なし整数〉 ((512～1073741824)) (単位：バイト)
			count	メッセージ送受信用, およびメッ セージ編集用バッファ数	〈符号なし整数〉 ((1～65535))
			extend	拡張するバッファ数	〈符号なし整数〉 ((0～65535))《0》
	mcftpsvr	-c		内部通信路名	〈1～8 文字の識別子〉

形式	コマンド	オプション	オペランド	定義内容	指定値
コマンド	mcftpsvr	-o	reruntm	リラン時のタイマ起動を引き継ぐかどうかを指定	yes 《no》
	mcftalcle	-l		論理端末名称	〈1～8 文字の識別子〉
		-t		論理端末の端末タイプ	send request
		-m	mmsgcnt	メモリ出力メッセージ最大格納数	〈符号なし整数〉 ((0～65535)) 《0》
			dmsgcnt	ディスク出力メッセージ最大格納数	〈符号なし整数〉 ((0～65535)) 《0》
		-k	quekind	出力メッセージの割り当て先	《memory》 disk
			quegrpid	キューグループ ID	〈1～8 文字の識別子〉
	mcftped			アプリケーション起動環境定義の終了	なし

4.2.3 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、次の表に示します。

表 4-7 MCF アプリケーション定義の内容

形式	コマンド	オプション	オペランド	定義内容	指定値
コマンド	mcfaenv	-a		MCF アプリケーション定義識別子	〈1～8 文字の識別子〉
		-p		アプリケーション起動プロセス識別子	〈数字 (0～9), a～f〉 ((01～ef))
	mcfaalcap	-n	name	アプリケーション名, または MCF イベント名	〈1～8 文字の識別子〉
			kind	アプリケーション種別	《user》 mcf
			type	アプリケーションの型	ans 《noans》 cont
			aplihold	アプリケーション異常終了時のアプリケーションの処置	《m》 a s
			msgcnt	入力メッセージ最大格納数	〈符号なし整数〉 ((0～65535)) 《0》
			lname	論理端末名称	〈1～8 文字の識別子〉
			cname	内部通信路名	〈1～8 文字の識別子〉

形式	コマンド	オプション	オペランド	定義内容	指定値
コマンド	mcfaalcap	-n	tempsize	継続問い合わせ応答用一時記憶データ格納用領域サイズ	〈符号なし整数〉 ((1~32000)) 《4096》 (単位: バイト)
			trnmode	アプリケーションのトランザクション属性	《trn》 nontrn
			errevt	通信イベント障害時のエラーイベントを通知するかどうかを指定	yes 《no》
		-N	modelname	モデルアプリケーション名, またはモデル MCF イベント名	〈1~8 文字の識別子〉
		-g	servgrp	アプリケーション名に対応するサービスグループ名	〈1~31 文字の識別子〉
			quegrpid	キューグループ ID	〈1~8 文字の識別子〉
			quekind	入力メッセージの割り当て先	《memory》 disk
			srvghold	アプリケーション異常終了時のサービスグループの処置	《m》 s
			recvmsg	異常終了した UAP の受信メッセージの扱い	《e》 r
		-v	servname	アプリケーション名に対応するサービス名	〈1~31 文字の識別子〉
			servhold	アプリケーション異常終了時のサービスの処置	《m》 a s
			ntmetim	非トランザクション MHP 限界経過時間	〈符号なし整数〉 ((0~65535)) (単位: 秒)
		-d	holdlimit	アプリケーション異常終了限界回数	〈符号なし整数〉 ((1~65535)) 《1》 (単位: 回)
			holdlmtyp	アプリケーション異常終了回数のカウント方法	sum 《cont》
			reschedulecnt	アプリケーション異常終了時の再スケジュール回数	〈符号なし整数〉 ((0~3)) (単位: 回)
			rescheduleint	アプリケーション異常終了時の再スケジュール間隔	〈符号なし整数〉 ((0~3)) (単位: 秒)
			reschedulelog	異常終了したアプリケーションを再スケジュールするときに, ログ	yes no

形式	コマンド	オプション	オペランド	定義内容	指定値
コマンド	mcfaalcap	-d	reschedulelog	メッセージを出力するかどうかを指定	yes no
		-j	ij	他システムから受信したメッセージの入力キュー登録時に、履歴情報を取得するかどうかを指定	yes 《no》
			oj	UAP からのメッセージ送信要求時に、履歴情報を取得するかどうかを指定	yes 《no》
			gj	UAP からのメッセージ受信要求時に、履歴情報を取得するかどうかを指定	yes 《no》
		-e	evtlogout	ERREVT1～ERREVT4 を起動するとき、ログメッセージを出力するかどうかを指定	yes 《no》

4.2.4 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、次の表に示します。

表 4-8 MCF 性能検証用トレース定義の内容

形式	オペランド	定義内容	指定値
set	prf_file_size	MCF 性能検証用トレース情報ファイルのサイズ	〈符号なし整数〉 ((1024～1048576)) 《1024》 (単位：キロバイト)
	prf_file_count	MCF 性能検証用トレース情報ファイルの世代数	〈符号なし整数〉 ((3～256)) 《3》

4.2.5 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、次の表に示します。

表 4-9 システムサービス情報定義の内容

形式	オペランド	定義内容	指定値
set	module	MCF 通信サービスを起動するための MCF 実行形式プログラム名	〈1～8 文字の識別子〉
	mcf_prf_trace	MCF 通信サービスごとに MCF 性能検証用トレース情報を取得するかどうかを指定	《Y》 N

4.2.6 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、次の表に示します。

表 4-10 システムサービス共通情報定義の内容

形式	オペランド	定義内容	指定値
set	max_socket_descriptors	ソケット用ファイル記述子の最大数	〈符号なし整数〉 ((64～3596))
	max_open_fds	MCF 通信プロセスでアクセスするファイルの最大数	〈符号なし整数〉 ((500～4032)) 《500》
	thdlock_sleep_time	スレッド間で排他が競合した場合のスレッドの待ち時間	〈符号なし整数〉 ((1～32767)) (単位：ミリ秒)
	mcf_prf_trace_level	MCF 性能検証用トレース情報の取得レベル	((00000000～00000001)) 《00000001》
	mcf_start_watch_interval	MCF 開始時プロセス間監視時間間隔	〈符号なし整数〉 ((10～3000)) 《3000》 (単位：ミリ秒)
	thread_stack_size	OpenTP1 内部で使用するスレッドスタック領域（内部通信用）のサイズ	〈符号なし整数〉 ((49152～524288)) 《49152*》 (単位：バイト)
	mcf_stack_size	OpenTP1 内部で使用するスレッドスタック領域（メッセージ制御用）のサイズ	〈符号なし整数〉 ((49152～524288)) 《49152*》 (単位：バイト)
putenv	DCMCFQUEBAK	MCF 構成変更準備停止時のバックアップファイル名	〈パス名〉 《\$DCDIR/spool/mcf/mcfquebak》

注※

AIX 版および HP-UX (IPF) 版の uCosminexus TP1/Message Control の場合、デフォルト値は 65536 となります。

HP-UX (IPF) 版の uCosminexus TP1/Message Control(64)の場合、デフォルト値は 65536 となります。

AIX 版の uCosminexus TP1/Message Control(64)の場合、デフォルト値は 131072 となります。

4.3 定義オブジェクトファイルの生成

ネットワークコミュニケーション定義の各ソースファイルから定義オブジェクトファイルを生成する方法について説明します。

定義オブジェクトファイルの生成には、次の二つのユティリティを使用します。

- MCF 定義オブジェクト生成ユティリティ
- MCF 定義結合ユティリティ

MCF 定義オブジェクト生成ユティリティは、ソースファイルからオブジェクトファイルを生成するときに使用します。MCF 定義オブジェクト生成ユティリティを起動するコマンドは、定義の種類によって異なります。

MCF 定義結合ユティリティは、二つのオブジェクトファイルを結合するときに使用します。MCF 定義結合ユティリティは、MCF 通信構成定義にだけ使用します。

MCF のバージョンアップに伴って、定義オブジェクトの形式が異なることがあります。定義オブジェクトのバージョンが異なると、開始処理が続行できません。この場合は、MCF と同じバージョンのユティリティで、定義オブジェクトを生成し直してください。

ユティリティ起動コマンドを次の表に示します。

表 4-11 定義オブジェクトファイル生成に使用するユティリティ起動コマンド

定義ソースファイル			ユティリティ起動コマンド		定義オブジェクトファイル名	
MCF マネージャ定義			mcfmngr		\$DCCONFPATH/_mu……	
MCF 通信構成定義	共通定義		mcfcomn		mcflink	\$DCCONFPATH/_mu……
	データコミュニケーション定義	プロトコル固有定義 1	mcf××××※1	一つ選択※2		
		プロトコル固有定義 2	mcf××××※1			
		：	：			
	：	：				
	：	：				
	：	：				
	アプリケーション起動定義	mcfpsvr				
MCF アプリケーション定義			mcfapli		\$DCCONFPATH/任意	

注※1

プロトコル種別に対応したユティリティ起動コマンドがあります。コマンド名については、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

注※2

MCF 通信サービスの通信方式によって、使用するユティリティ起動コマンドが決まります。

各定義では、次のようにオブジェクトファイルを生成します。

- MCF マネジャ定義
MCF 定義オブジェクト生成ユティリティ（mcfmngnr コマンド）を使用します。
- MCF 通信構成定義
MCF 定義オブジェクト生成ユティリティと MCF 定義結合ユティリティを使用します。
次の手順でオブジェクトファイルを生成します。
 - 1.MCF 定義オブジェクト生成ユティリティを使用して、共通定義オブジェクトファイルとデータコミュニケーション定義オブジェクトファイルを、それぞれ生成します。ユティリティ起動コマンドについては、表 4-11、およびマニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。
 - 2.MCF 定義結合ユティリティ（mcflink コマンド）を使用して、1.で生成した二つの定義オブジェクトを結合し、MCF 通信構成定義のオブジェクトファイルを出力します。
- MCF アプリケーション定義
MCF 定義オブジェクト生成ユティリティ（mcfapli コマンド）を使用します。

なお、対応する定義ソースが不明となった MCF 定義オブジェクトファイルの内容を知りたい場合は、[「4.3.3 mcf××××（MCF 定義オブジェクト解析コマンド）」](#)を参照してください。

4.3.1 mcf××××（MCF 定義オブジェクト生成ユティリティ起動コマンド）

(1) 形式

mcf××××	-i	【パス名】 入力ファイル名
	-o	【パス名】 出力ファイル名
	[-r	{no rep}]

定義名	起動コマンド
MCF マネジャ定義	mcfmngnr
MCF 通信構成定義（共通定義）	mcfcomn
MCF 通信構成定義（プロトコル固有定義）	mcf××××※
MCF 通信構成定義（アプリケーション起動定義）	mcfpsvr

定義名	起動コマンド
MCF アプリケーション定義	mcfapli

注

各定義ごとに、専用の起動コマンド（mcf××××）があります。上表の起動コマンドを使用してください。

注※

マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

(2) 機能

定義ソースファイルを入力して、構文をチェックし、定義オブジェクトファイルを生成します。

ただし、開始から再開始の間に定義オブジェクトファイルを変更しないでください。変更した場合、再開始時に正常に動作しないことがありますのでご注意ください。

(3) オプション

- -i [パス名] 入力ファイル名 ~ 〈パス名〉〈1~8 文字の識別子〉

定義のソースファイル名を指定します。

- -o [パス名] 出力ファイル名 ~ 〈パス名〉〈1~8 文字の英数字〉

定義のオブジェクトファイル名を指定します。

MCF マネージャ定義の場合、先頭 3 文字が'_mu'で始まるファイル名を指定してください。

システム定義ファイルの指定値をチェックする dcdefchk コマンドのチェック対象から MCF アプリケーション定義を外す場合は、先頭 1 文字が '_'で始まるファイル名を指定してください。

- -r {no | rep} ~ 《no》

定義オブジェクトファイルの出力先に読み取り権限を持つファイルがすでに存在する場合、定義オブジェクトファイルを上書きするかどうかを指定します。

no：定義オブジェクトファイルを上書きしないで、KFCA10332-E メッセージを出力します。

rep：定義オブジェクトファイルを上書きします。

4.3.2 mcflink (MCF 定義結合ユーティリティ起動コマンド)

(1) 形式

```
mcflink  -i  [パス名] 入力ファイル名  [パス名] 入力ファイル名
          -o  [パス名] 出力ファイル名
          [-r  {no | rep} ]
```

(2) 機能

共通定義とデータコミュニケーション定義のオブジェクトファイルを入力し、定義コマンド間の関連をチェックし、一つのオブジェクトファイルに結合します。

ただし、開始から再開始の間にオブジェクトファイルを変更しないでください。変更した場合、再開始時に正常に動作しないことがありますのでご注意ください。

(3) オプション

- -i [パス名] 入力ファイル名 [パス名] 入力ファイル名
 ~ <パス名> <1~8 文字の英数字> <パス名> <1~8 文字の英数字>
入力オブジェクトファイル名を指定します。次のファイルの一つずつ指定してください。どちらを先に指定してもかまいません。
 - 共通定義のオブジェクトファイル
 - データコミュニケーション定義のオブジェクトファイル
- -o [パス名] 出力ファイル名 ~ <パス名> <1~8 文字の英数字>
出力オブジェクトファイル名を指定します。先頭 3 文字が'_mu'で始まるファイル名を指定してください。
- -r {no | rep} ~ <no>
オブジェクトファイルの出力先に読み取り権限を持つファイルがすでに存在する場合、オブジェクトファイルを上書きするかどうかを指定します。
no：オブジェクトファイルを上書きしないで、KFCA10332-E メッセージを出力します。
rep：オブジェクトファイルを上書きします。

4.3.3 mcf×××× (MCF 定義オブジェクト解析コマンド)

(1) 形式

```
mcf×××× -i    [パス名] 解析対象オブジェクトファイル名
```

定義オブジェクト名	解析コマンド
MCF マネジャ定義オブジェクト	mcfmngrr
MCF アプリケーション定義オブジェクト	mcfaplr
MCF 通信共通定義（または、これとプロトコル固有定義を結合したオブジェクト）	mcfcomnr
MCF アプリケーション起動定義オブジェクト（または、これと共通定義を結合したオブジェクト）	mcfpsvrr
TP1/NET/TCP/IP の MCF 通信共通定義オブジェクト（または、これと共通定義を結合したオブジェクト）	mcftcpr

定義オブジェクト名	解析コマンド
TP1/NET/XMAP3 の MCF 通信共通定義オブジェクト（または、これと共通定義を結合したオブジェクト）	mcfxpr

注

定義オブジェクトごとに、専用の解析コマンド（mcf××××）があります。上表の解析コマンドを使用してください。

(2) 機能

MCF の定義オブジェクトファイルである MCF マネジャ定義オブジェクトファイル，MCF アプリケーション定義オブジェクトファイル，MCF 通信共通定義オブジェクトファイル，および MCF アプリケーション起動定義オブジェクトファイルの定義オブジェクトを解析し，定義ソースの形式で標準出力します。

(3) オプション

- ・ -i [パス名] 解析対象オブジェクトファイル名 ～ 〈1～8 文字の英数字〉
定義オブジェクトが格納されているファイル名を指定します。

(4) 出力形式

定義オブジェクトの解析後の出力例を次に示します。

```
#####
MCF manager definition

#####
OBJECT FILE NAME : xxxxxxxx
VV-RR           : vv-rr
DATE            : yyyy-mm-dd hh:mm:ss
#####

mcfmenv
-m id           = A
-m name         = mng01

mcfmcomn
-n              = 10
-p              = 300
-j              = 4096
*-t delayed    = no
-c              = 255
-w stats       = no
@-i             = msg
*-d mcfdump     = yes
*-d dumpcount   = 3
*-d pdebug      = 00000000
-l              = 0
*-J pj_put      = yes
*-J mj_select   = no
```

```

*-J mcf_buffer = yes

mcfmcname
-s mcfsvname = _muua01
-s syssvname = mcfu01

mcfmcname
-s mcfsvname = _mups01
-s syssvname = mcfu02

.
.
.

##### End Of File #####

```

(凡例)

xxxxxxx: 解析する定義オブジェクトファイル名

vv-rr: 解析する定義オブジェクトのバージョン・リビジョン番号

yyyy-mm-dd hh:mm:ss: 解析する定義オブジェクトのファイル生成日時

(5) 解析結果

定義オブジェクト解析コマンドは、その解析結果を定義ソースの形式で出力します。しかし、それは解析結果であり記述形式は元の定義ソースの記述形式とは一致しません。定義ソースと定義オブジェクト解析結果の差異を次に示します。

項目	定義ソース	定義オブジェクト解析結果
注釈文	書き込みできる	出力しない
省略値の扱い	省略できる	- 限定公開部分も含めて、省略値を出力する - システム環境定義の default_value_option オペランドの指定値で省略値が変更となるオペランドの場合、行の先頭に "@" を付与する
限定公開部分の表記方法	一般公開部分と差異なし	OpenTP1 Version 7 での限定公開機能の行の先頭に "*" を付与する
定義コマンド名とオプションの表記方法	1 行に表記できる (例) mcfmcomn -n 10	定義コマンド名を表記後、改行する。また、オプションに "=" を付記する (例) mcfmcomn -n = 10
1 定義コマンドが複数の行にわたる場合	継続記号 "¥" を付与する (例) mcfmcomn -n 10 ¥ -p 300	継続記号は出力しない (例) mcfmcomn -n = 10 -p = 300

項目	定義ソース	定義オブジェクト解析結果
1 定義オプションに複数のオペランドを指定する場合	複数のオペランドをまとめて二重引用符(")で囲む (例) mcfmenv -m "id=A name=mng01"	個々のオペランドに対してオプションを付記する (例) mcfmenv -m id=A -m name = mng01
その他	なし	• ファイル名などを記したタイトルが出力される • 定義オブジェクト生成時の補正によって、実際の指定値とは異なる内容が出力される場合がある • 定義ソースと該当コマンドのバージョンの差異によって、解析結果にサポート内容の過不足がある場合がある

(6) 注意事項

解析対象が不正であった場合の動作は保証しません。

5

ネットワークコミュニケーション定義の詳細

ネットワークコミュニケーション定義の詳細について、それぞれの定義ごとに説明します。

MCF マネージャ定義

MCF マネージャ定義は、MCF マネージャプロセスに対して、実行環境を設定するものです。定義ソースファイルを一つ作成してください。

MCF サービスで使用する通信サービスの種類によっては、通信サービス固有の環境情報を設定する必要があります。この場合、このマニュアルで記述する定義のほかに、通信サービス固有の定義を加えてください。その定義コマンドのオプションなどについては、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

MCF マネージャ定義コマンドの指定順序を次の図に示します。

図 5-1 MCF マネージャ定義コマンドの指定順序

mcfmenv	(MCF マネージャ環境定義)
mcfmcomm	(MCF マネージャ共通定義)
{{ mcfmcname }} 繰り返し可	
(通信サービス定義)	
mcfmuap	(UAP 共通定義)
{{ mcfmqgid }} 繰り返し可	
mcfmexp	(入出力キュー定義)
(拡張予約定義)	
[mcfmsts]	(状態引き継ぎ定義)
[mcfmsmsg]	(ログメッセージ出力抑止定義)
[mcfmsvg]	(サービスグループ属性定義)

名称

MCF マネジャ環境定義

形式

```
mcfmenv  -m  " [id=MCFマネジャプロセス識別子]
               name=MCFマネジャ名"
```

機能

MCF マネジャに関する環境を定義します。

オプション

●-m

(オペランド)

id=MCF マネジャプロセス識別子 ～ 〈英字〉 ((A～Z, a～z)) 《A》

MCF マネジャ識別子を指定します。

MCF マネジャプロセス識別子と MCF 通信構成定義の MCF 通信プロセス識別子とを組み合わせる 3 文字の文字列が作成されます。この文字列が運用コマンドや MCF が出力するメッセージログなどで MCF 通信プロセスを特定するためのキーとなります。

このオプションで指定した MCF マネジャプロセス識別子は、次に示すコマンドの -i オプション（リソースマネジャ拡張子）で指定してください。

- トランザクションサービス定義の trnstring コマンド
- ユーザサービスデフォルト定義の trnrmid コマンド
- ユーザサービス定義の trnrmid コマンド

name=MCF マネジャ名 ～ 〈1～8 文字の識別子〉

MCF マネジャ名を指定します。

MCF マネジャ名は、MCF 起動時に表示されるメッセージで MCF マネジャ定義を識別するために使用します。

名称

MCF マネジャ共通定義

形式

```
mcfmcomn  [-n 出力通番使用論理端末数]
           [-p MCF作業領域長]
           [-i inc | msg]
           [-j MCFマネジャプロセスのジャーナルバッファの大きさ]
           [-o " [cmdsvname=MCFオンラインコマンドサービス名] "]
           [-w " [stats=yes | no] "]
```

機能

MCF マネジャに共通する環境を定義します。

オプション

●-n 出力通番使用論理端末数

～ 〈符号なし整数〉 ((0～2048)) 《0》

一つの MCF システムでメッセージ出力通番を使用する論理端末の最大数を指定します。

0 を指定した場合は、メッセージ出力通番を使用しません。

指定値に応じて静的共用メモリ、ステータスファイル、およびチェックポイントダンプファイルの容量を見積もる必要があります。静的共用メモリの見積もりは、[「付録 B.2 MCF サービス用の共用メモリの見積もり式」](#)を参照してください。ステータスファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「ステータスファイルのサイズの見積もり式」の説明を参照してください。チェックポイントダンプファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「チェックポイントダンプファイルのサイズの見積もり式」の説明を参照してください。

●-p MCF 作業領域長

～ 〈符号なし整数〉 ((100～1945599)) (単位：キロバイト)

MCF プロセス間で使用する共用メモリの大きさを指定します。静的に確保されるテーブルおよびバッファプールの共用メモリの大きさを指定してください。詳細については、[「付録 B.2 MCF サービス用の共用メモリの見積もり式」](#)を参照してください。

ここで設定した MCF 作業領域長をシステム環境定義 (\$DCDIR/conf/env) の static_shmpool_size に加えてください。MCF 作業領域長の設定値が足りないと、static_shmpool_size の値が十分でもメモリ不足が発生する場合があります。

MCF 静的共用メモリが不足した場合、静的共用メモリ（システム環境定義の static_shmpool_size オペランドの指定値）の未使用領域から、このオプションで指定したサイズの 1/2 のサイズ分を自動的に追加確保します。メモリの自動確保は最大 254 回行います。それでもメモリが不足した場合は、ログメッセージとして KFCA10230-E を出力し、障害情報を出力します。

メモリの自動確保回数が 254 回以内であっても、静的共用メモリ不足を検出した場合は、ログメッセージとして KFCA10240-E を出力して、障害情報を出力します。メモリ追加確保時に、ログメッセージとして KFCA10242-I を出力するかどうかは、-i オプションで指定します。静的共用メモリ不足の発生を検知したい場合は、-i オプションに msg を指定し、メッセージを出力するように指定してください。

●-i inc | msg

～《msg》

-p オプションで指定する MCF 作業領域長の静的共用メモリが不足し、自動的に追加確保したときに、ログメッセージとして KFCA10242-I を出力するかどうかを指定します。

inc

静的共用メモリの追加確保時に、ログメッセージを出力しません。

msg

静的共用メモリの追加確保時に、ログメッセージとして KFCA10242-I を出力します。

●-j MCF マネジャプロセスのジャーナルバッファの大きさ

～〈符号なし整数〉((4096～4000000 の 4 の倍数))《4096》(単位：バイト)

プロセス回復用のジャーナルデータを蓄えておく領域の大きさを指定します。

MCF マネジャプロセスが、ジャーナルサービスに対して次の要求を出すまで、この領域にジャーナルデータを蓄えておきます。

- ジャーナルファイルへの出力要求
- ジャーナルサービスのバッファへの格納要求

ジャーナルバッファの大きさには、CJ のデータ長のうち最大値以上の値を指定してください。

CJ (メッセージキューサーバ) のデータ長

ITQ, OTQ にディスクキューを使用した場合に取得します。

次に示す二つの計算式のうち、最大の値をデータ長としてください。

計算式 1：

●32 ビット版のとき

$260 + (24 \times m)$

●64 ビット版のとき

$\uparrow \{260 + (24 \times m)\} / 8 \uparrow \times 8$

計算式 2 :

●32 ビット版のとき

$$\uparrow \{88 + (24 \times (\uparrow \text{msg} / \text{ql} \uparrow + \uparrow 960 / \text{ql} \uparrow)) + \text{qio}\} / 4 \uparrow \times 4$$

●64 ビット版のとき

$$\uparrow \{88 + (24 \times (\uparrow \text{msg} / \text{ql} \uparrow + \uparrow 1072 / \text{ql} \uparrow)) + \text{qio}\} / 8 \uparrow \times 8$$

(凡例)

$\uparrow \uparrow$: 小数点以下を切り上げます。

m : 保持メッセージ数 (メッセージキューサービス定義の quegrp 定義コマンドの -m オプションで指定した値), すべての論理端末, またはサービスグループのうちの構成変更準備停止時における滞留メッセージ数の最大値 (構成変更準備停止を行う直前にそれぞれ -m オプションを指定した mcftlsle コマンド, または mcftlssg コマンドを実行することで滞留メッセージの最大値を取得できます)

msg : ディスクキューを使用した受信メッセージ長または送信メッセージ長

ql : キューファイル物理レコード長

qio : メッセージキューサービス定義の que_io_maxrecsize オペランドの指定値

ユーザが指定した値によって, MCF は次の処理をします。

1. 指定した値が MCF で定めた下限値より小さい場合, その値を MCF で定めた下限値に補正します。
2. 指定した値が MCF で定めた上限値より大きい場合, その値を MCF で定めた上限値に補正します。
3. 指定した値が 4 の倍数でない場合, その値を 4 の倍数になるように切り上げます。

●-o cmdsvname=MCF オンラインコマンドサービス名

～ 〈1～8 文字の英数字〉

MCF オンラインコマンドサービス名を指定します。

ここで指定する値は, 先頭が '_mcs' で始まる 8 文字以内の英数字を指定します。

なお, ご使用の OS が Windows の場合, このオプションは指定できません。

●-w

(オペランド)

stats=yes | no ～ 《no》

MCF 稼働統計情報を取得するかどうかを指定します。

なお, ご使用の OS が Windows の場合, このオプションは指定できません。

yes

静的共用メモリ上に MCF 稼働統計情報を取得します。

取得した MCF 稼働統計情報を参照するには, mcfstats コマンドおよび mcfreport コマンドを実行します。

no

MCF 稼働統計情報を取得しません。

名称

通信サービス定義

形式

```
{{ mcfmcname -s "mcfsvname=MCF通信サービス名  
syssvname=システムサービス情報定義ファイル名"}}
```

機能

MCF 通信サービスに関する環境を定義します。

指定数に応じて静的共用メモリ、およびステータスファイルの容量を見積もる必要があります。静的共用メモリの見積もりは、「付録 B.2 MCF サービス用の共用メモリの見積もり式」を参照してください。ステータスファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「ステータスファイルのサイズの見積もり式」の説明を参照してください。

オプション

●-s

(オペランド)

mcfsvname=MCF 通信サービス名 ～ 〈1～8 文字の英数字〉

MCF 通信サービス名を指定します。

MCF 通信サービス名には、先頭 3 文字が'_mu'で始まる名称を指定します。

この MCF 通信サービス名は、MCF 通信構成定義のオブジェクトファイル名です。

なお、MCF 通信サービス名は、ほかの mcfmcname コマンドで指定する MCF 通信サービス名と通信プロセス単位で重複して指定できません。

syssvname=システムサービス情報定義ファイル名 ～ 〈1～8 文字の識別子〉

システムサービス情報定義ファイルの名称を指定します。

システムサービス情報定義ファイル名称には、先頭 4 文字が'mcfu'で始まる名称を指定します。

このシステムサービス情報定義ファイルは、MCF 通信サービスを起動するために使用する実行形式プログラムを指定するファイルです。

名称

UAP 共通定義

形式

```
mcfmuap  [-d 通信関数発行回数上限値]
          [-t "[sndtim=同期型送信監視時間]
              [sndrcvtim=同期型送受信監視時間]
              [recvtim=同期型受信監視時間]" ]
          [-j ユーザーバのジャーナルバッファの大きさ]
          [-e "segsize=最大セグメント長"]
          [-l "[initseq=出力通番の初期値]
              [maxseq=ラップ時の出力通番最大値]
              [minseq=ラップ後の出力通番開始値]" ]
          [-u "[ntmetim=非トランザクションMHP限界経過時間]" ]
          [-a "delaytim=アプリケーション起動時の遅延許容時間"]
          [-c "[order=function | commit]
              [noansreply=yes | no]
              [commitdml=yes | no]
              [otqinhold=commit | function]
              [errevt_recovery=yes | no]" ]
          [-r "[reschedulecnt=アプリケーション異常終了時の
              再スケジュール回数]
              [rescheduleint=アプリケーション異常終了時の
              再スケジュール間隔]
              [reschedulelog=yes | no]" ]
```

機能

UAP に共通する環境を定義します。

オプション

●-d 通信関数発行回数上限値

～ 〈符号なし整数〉 ((0～65535)) 《0》 (単位：回)

MHP の暴走を監視するために、MHP がトランザクション処理中に発行する総通信関数の発行回数の上限値を指定します。

指定回数を超えて通信関数を発行した場合、MHP を異常終了させます。

ユーザーサービスデフォルト定義および該当するサービスのユーザーサービス定義の MCF 通信関数発行上限値の指定を省略したときに、ここで 0 を指定していた場合は、該当サービスの通信関数の発行回数を監視しません。

●-t

(オペランド)

sndtim=同期型送信監視時間 ～〈符号なし整数〉((0～65535))《0》(単位：秒)

同期型メッセージ送信の仕掛け開始時から、仕掛け終了時までの限界経過時間を指定します。

相手システムの処理完了報告までの時間を監視します。

0 を指定した場合は、時間監視をしません。

sndrcvtim=同期型送受信監視時間 ～〈符号なし整数〉((0～65535))《0》(単位：秒)

同期型メッセージ送受信の仕掛け開始時から、仕掛け終了時までの限界経過時間を指定します。

相手システムからの応答時間を監視します。

0 を指定した場合は、時間監視をしません。

rcvtim=同期型受信監視時間 ～〈符号なし整数〉((0～65535))《0》(単位：秒)

同期型メッセージ受信の仕掛け開始時から、仕掛け終了時までの限界経過時間を指定します。

相手システムからの処理完了報告までの時間を監視します。

0 を指定した場合は、時間監視をしません。

●-j ユーザーバのジャーナルバッファの大きさ

～〈符号なし整数〉((4096～4000000 の 4 の倍数))《32768》(単位：バイト)

ジャーナルデータを蓄えておく領域の大きさを指定します。

UAP プロセスが、ジャーナルに対して次の要求を出すまで、この領域にジャーナルデータを蓄えておきます。

- ・ ジャーナルファイルへの出力要求
- ・ ジャーナルサービスのバッファへの格納要求

ジャーナルバッファの大きさには、GJ, OJ, CJ のデータ長のうち最大値以上の値を指定してください。

1. GJ のデータ長

MCF アプリケーション定義の mcfaalcap 定義コマンドの-j オプションの gj オペランドに yes を指定したときに取得します。

●32 ビット版のとき

$$\uparrow (204 + \text{seg}) / 4 \uparrow \times 4$$

●64 ビット版のとき

$$\uparrow (204 + \text{seg}) / 8 \uparrow \times 8$$

2. OJ のデータ長

MHP からメッセージ送信をした場合は、MCF アプリケーション定義の mcfaalcap の-j オプションの oj オペランドに yes を指定したときに取得します。SPP からメッセージ送信をした場合は、ユーザーサービス定義の mcf_spp_oj オペランドに Y を指定したときに取得します。

●32 ビット版のとき

$$\uparrow (204 + \text{seg}) / 4 \uparrow \times 4$$

●64 ビット版のとき

$$\uparrow (204 + \text{seg}) / 8 \uparrow \times 8$$

3. CJ (メッセージキューサーバ) のデータ長

ITQ, OTQ にディスクキューを使用した場合に取得します。

●32 ビット版のとき

$$\uparrow \{260 + \Sigma \{(24 \times (\uparrow \text{msg} / \text{ql} \uparrow + \uparrow 960 / \text{ql} \uparrow))\} + \text{qio}\} / 4 \uparrow \times 4$$

●64 ビット版のとき

$$\uparrow \{260 + \Sigma \{(24 \times (\uparrow \text{msg} / \text{ql} \uparrow + \uparrow 1072 / \text{ql} \uparrow))\} + \text{qio}\} / 8 \uparrow \times 8$$

(凡例)

$\uparrow \uparrow$: 小数点以下を切り上げます。

seg : セグメント長

msg : ディスクキューを使用した受信メッセージ長または送信メッセージ長

ql : キューファイル物理レコード長

qio : メッセージキューサービス定義の que_io_maxrecsize オペランドの指定値

Σ : 該当するトランザクションで、受信または送信したメッセージに対する見積もりの総和

ユーザが指定した値によって、MCF は次の処理をします。

1. 指定した値が MCF で定めた下限値より小さい場合、その値を MCF で定めた下限値に補正します。
2. 指定した値が MCF で定めた上限値より大きい場合、その値を MCF で定めた上限値に補正します。
3. 指定した値が 4 の倍数でない場合、その値を 4 の倍数になるように切り上げます。

なお、このオペランドは、ユーザサービス定義、およびユーザサービスデフォルト定義の mcf_jnl_buff_size オペランドでも指定できます。

指定値の優先順位は次のとおりです (1.> 2.> 3.)。

1. ユーザサービス定義
2. ユーザサービスデフォルト定義
3. MCF マネージャ定義

ただし、ジャーナルのデータ長が 131072 を超える場合は、データ長を MCF マネージャ定義に指定し、ユーザサービス定義、およびユーザサービスデフォルト定義の mcf_jnl_buff_size オペランドは省略してください。

●-e segsize=最大セグメント長

～ 〈符号なし整数〉 ((512～2147483647)) 《32768》 (単位 : バイト)

次の場合に送受信するメッセージの最大セグメント長を指定します。最も大きい方の値を、最大セグメント長として指定してください。

- ・ エラーイベント処理用 MHP を起動する場合（エラーイベント処理用 MHP が送受信するメッセージの最大セグメント長）
- ・ アプリケーション起動機能を使用する場合（dc_mcf_execap 関数で起動するアプリケーションに渡すメッセージの最大セグメント長）
- ・ メッセージ再送機能を使用する場合
- ・ mcftdmpqu コマンドで入出力キューの内容を複写する場合（内容を複写するキュー内のメッセージの最大セグメント長）

●-l

（オペランド）

initseq=出力通番の初期値 ～〈符号なし整数〉((0～2147483647))《1》

メッセージ出力通番の初期値を指定します。

maxseq=ラップ時の出力通番最大値 ～〈符号なし整数〉((0～2147483647))《65535》

出力通番をラップさせる時の、最大値を指定します。

指定した値の次は、開始値になります。

指定した値が小さいと、出力通番のラップアラウンドが発生したあとに、同一キューファイル内に同一の出力通番が存在して、メッセージの出力通番のユニーク性が保証されない場合があります。

このため、次の計算式を満たす値を指定してください。

ラップ時の出力通番最大値 > $Q \times (1 + R)$

Q：ディスクメッセージ最大格納数

R：ロールバック率

ただし、ロールバック率は最悪のケースを考慮した値を指定してください。

minseq=ラップ後の出力通番開始値 ～〈符号なし整数〉((0～1))《1》

出力通番をラップさせる時の、ラップ後の開始値を指定します。

ラップ後の出力通番開始値は、接続相手のオンラインコントロールプログラムによって異なります。

●-u ntmetim=非トランザクション MHP 限界経過時間

～〈符号なし整数〉((0～65535))《0》（単位：秒）

アプリケーション属性定義（mcfaalcap）の-v オプションに指定する ntmetim オペランドの省略値を指定します。

非トランザクション MHP の処理のループ、およびハングアップを監視するために、非トランザクション MHP の処理に掛かる限界経過時間を指定します。

0 を指定した場合は、時間監視しません。

指定時間を超えても非トランザクション MHP の処理が終了しない場合、非トランザクション MHP を異常終了します。

●-a delaytim=アプリケーション起動時の遅延許容時間

～ 〈符号なし整数〉 ((0～360)) 《0》 (単位：分)

起動時刻指定のアプリケーション起動で、指定時刻が現在時刻の前の場合の遅延許容時間を指定します。

指定した時刻が現在時刻より前で、遅延許容時間の範囲内の時間の場合、MCF はすぐに該当するアプリケーションを起動します。遅延許容時間外の時刻の場合、翌日の指定時刻に起動します。

●-c

(オペランド)

order=function | commit ～ 《commit》

複数の UAP による次に示す処理順序を指定します。

- ・ 同一の論理端末 (mcftalcle -l) を使用するメッセージの送信順序および再送順序
- ・ 同一の論理端末 (mcfaalcap -n lname) または内部通信路 (mcfaalcap -n cname) を使用する UAP の起動順序

function

UAP からの関数※の呼び出し順に処理します。

commit

UAP によるトランザクションのコミット順に処理します。

論理端末の出力キュー（ディスクキュー）にメッセージが滞留している状態で、OpenTP1 を再開した場合でも、トランザクションのコミット順に処理します。

注※

関数とは、メッセージ送信関数、メッセージ再送関数およびアプリケーション起動関数を示します。

noansreply=yes | no ～ 《yes》

非応答型の MHP からの問い合わせ応答をするかどうかを指定します。

yes

非応答型の MHP からの問い合わせ応答をします。

非応答型 MHP から応答メッセージを送信した場合、一般一方送信メッセージとして扱います。

なお、出力通番は付けられません。

no

非応答型の MHP からの問い合わせ応答をしません。

非応答型 MHP から応答メッセージを送信した場合、エラーになります。

commitdml=yes | no ～ 《no》

データ操作言語の COMMIT 文を発行するかどうかを指定します。

yes

COMMIT 文を使用します。

データ操作言語の COMMIT 文を発行してエラーが発生した場合、KFCA11199-E メッセージを出力し、MHP が異常終了します。

no

COMMIT 文を使用しません。

データ操作言語の COMMIT 文を発行してエラーが発生しても、UAP でエラーを検出できません。

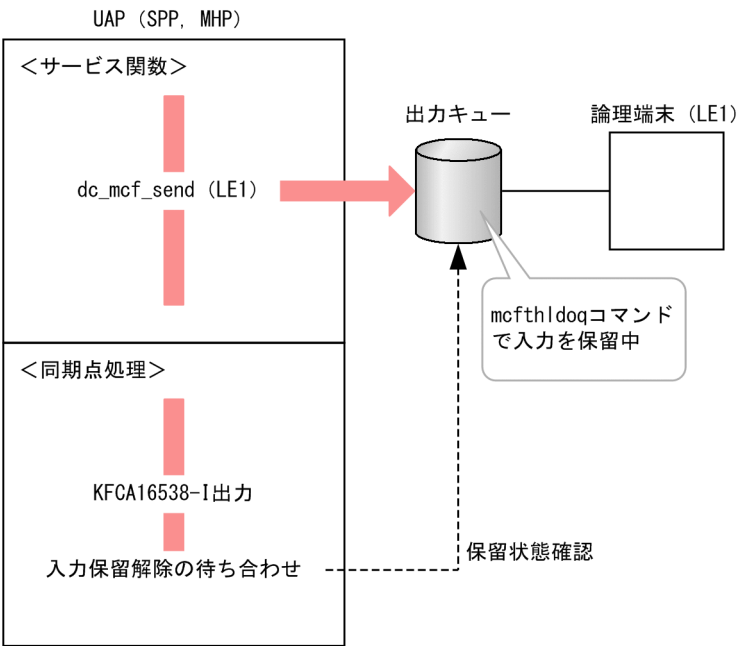
otqinhold=commit | function ~ 《commit》

出力キューの入力保留中に UAP がメッセージを送信したときの動作を指定します。

commit

トランザクションの同期点処理で UAP の処理を待ち合わせます。

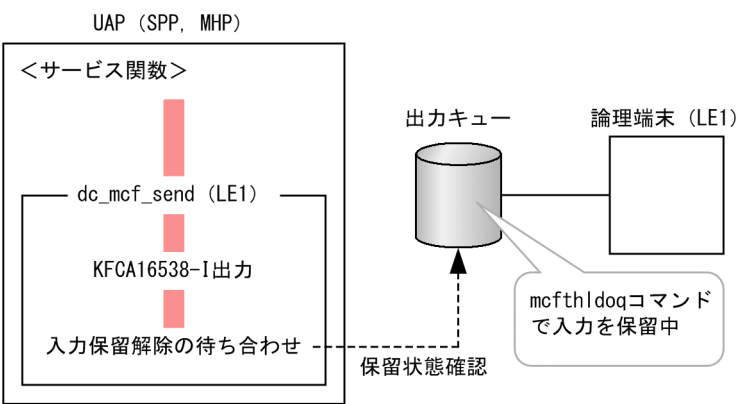
同期点処理で待ち合わせる場合の例を次に示します。



function

メッセージの送信関数で UAP の処理を待ち合わせます。

メッセージの送信関数で待ち合わせる場合の例を次に示します。



出力キューの入力保留中に UAP の処理を待ち合わせる関数を次の表に示します。

言語		関数名称
C 言語		dc_mcf_send 関数 dc_mcf_resend 関数 dc_mcf_reply 関数
COBOL 言語	CALL インタフェース	CBLDCMCF('SEND ') CBLDCMCF('RESEND ') CBLDCMCF('REPLY ')
	DML インタフェース	SEND - メッセージの送信※

注※

非同期型のメッセージ送信（SYNCHRONOUS MODE 句を指定していない、または ASYNC, '0'もしくは空白が設定されているデータ項目を指定）の場合だけです。

注意事項

同期点処理で待ち合わせる場合とメッセージの送信関数で待ち合わせる場合の動作の違いを次の表に示します。

UAP 共通定義 (mcfmuap -c) の otqinhold オペランド の指定値	dcsvstop -f コマ ンドによる強制 停止	service_expiration_ti me オペランドによる実 時間監視タイムアウト※	trn_expiration_time オペラ ンド, mcfmuap -u ntmetim オペランド, mcfaalcap -v ntmetim オ ペランドの監視対象該否	UAP 強制停止前に 送信したメッセージ の扱い
commit または省略	入力保留を解除するまで待ち合わせる	入力保留を解除するまで UAP の異常終了を待ち合わせる	該当しない (監視区間の範囲外)	有効 (出力キューに残留)
function	即時に受け付ける	即時に UAP が異常終了する	該当する (監視区間の範囲内)	無効 (出力キューから削除)

注※

SPP だけ有効です。MHP では監視しません。

errevt_recovery=yes | no ~ 《yes》

OpenTP1 再開始時のエラーイベント（ERREVT1, ERREVT2, ERREVT3, ERREVT4, ERREVT5）を引き継ぐかどうかを指定します。

yes

OpenTP1 再開始時のエラーイベントを引き継ぎます。

no

OpenTP1 再開始時のエラーイベントを引き継ぎません。すべてのエラーイベントを破棄します。

注意事項

- 引き継ぎ対象となるのは、ディスクキューに割り当てられている（アプリケーション属性定義（mcfaalcap -g）の quekind オペランドに disk を指定）エラーイベントです。メモリキューに割り当てられているエラーイベントはこのオペランドの指定値に関係なく、破棄します。

- ・ エラーイベントが発生したメッセージが OpenTP1 再開始時に破棄されるメッセージの場合、このオペランドの指定値に関係なく、OpenTP1 の再開始時にエラーイベントを破棄します。

●-r

(オペランド)

reschedulecnt=アプリケーション異常終了時の再スケジュール回数 ～ 〈符号なし整数〉 ((0～3)) 《0》
(単位：回)

MHP が異常終了した場合の、再スケジュール回数を指定します。

アプリケーション属性定義 (mcfaalcap) に指定する-d オプションの reschedulecnt オペランドの省略時解釈値となります。

このオペランドに 1～3 の値を指定すると、受信したメッセージを該当する入力キューの最後に格納し、異常終了した MHP を自動的にスケジュールし直します。

0 を指定した場合は、再スケジュールしません。

指定した回数を超えて MHP が異常終了した場合、再スケジュールしません。この場合、KFCA11163-E メッセージを出力し、エラーイベントを通知します。

rescheduleint=アプリケーション異常終了時の再スケジュール間隔 ～ 〈符号なし整数〉 ((0～3)) 《0》
(単位：秒)

異常終了した MHP を自動的に再スケジュールするまでの所要時間※に加算する時間を、秒単位で指定します。

アプリケーション属性定義 (mcfaalcap) に指定する-d オプションの rescheduleint オペランドの省略時解釈値となります。

この指定は、reschedulecnt オペランドに 1～3 の値を指定した場合に有効となります。

注※

異常終了したプロセス以外に空きプロセスが存在する場合、間隔を空けないで再スケジュールします。

異常終了したプロセス以外に空きプロセスが存在しない、かつ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 hold オペランドに Y を指定した場合、再スケジュールするまでに最大 10 秒掛かります。

異常終了したプロセス以外に空きプロセスが存在しない、かつ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 hold オペランドに N を指定した場合、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 term_watch_time オペランドの指定時間内で、異常終了した MHP の 3 回目の再スケジュールに最大 10 秒掛かります。2 回目までは間隔を空けないで再スケジュールします。

reschedulelog=yes | no ～ 《no》

異常終了した MHP を自動的に再スケジュールするときに、ログメッセージを出力するかどうかを指定します。

アプリケーション属性定義 (mcfaalcap) に指定する-d オプションの reschedulelog オペランドの省略時解釈値となります。

この指定は、reschedulecnt オペランドに 1～3 の値を指定した場合に有効となります。

yes

KFCA11810-I メッセージを出力します。

no

KFCA11810-I メッセージを出力しません。

名称

入出力キュー定義

形式

```
[{{mcfmqgid -q "quekind=itq|otq  
quegrpid=キューグループID"}}]
```

機能

入出力キューに関する環境を定義します。

ディスクキューを指定する場合は必ず定義してください。メモリキューの場合は定義する必要はありません。

入力キューおよび出力キューに対応する物理ファイルをそれぞれ複数個作成できます。

入力キューおよび出力キューに対応する物理ファイルは MCF 通信サービス間で共用できます。

ただし、入力キューおよび出力キューを同一の物理ファイルに割り当てることはできません。

入力キューは、サービスグループに対応します。また、出力キューは論理端末に対応します。

指定数に応じて静的共用メモリ、およびステータスファイルの容量を見積もる必要があります。静的共用メモリの見積もりは、「付録 B.2 MCF サービス用の共用メモリの見積もり式」を参照してください。ステータスファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「ステータスファイルのサイズの見積もり式」の説明を参照してください。

オプション

●-q

(オペランド)

quekind=itq|otq

キュー（ディスク）の種別を指定します。

itq

入力キュー用

otq

出力キュー用

quegrpid=キューグループ ID ～ 〈1～8 文字の識別子〉

quekind オペランドで指定したディスクキューを使用するキューグループ ID を指定します。

この MCF マネジャ定義下の MCF で使用するキューグループ ID をすべて指定します。キューグループ ID はメッセージキューサービス定義で指定したキューグループ ID を指定します。

このキューグループ ID はほかの `mcfmqgid` コマンドで指定するキューグループ ID と通信プロセス単位で重複して指定できません。

名称

拡張予約定義

形式

```
[mcfmexp  -g サービスグループの登録数
          -l 論理端末の登録数
          [-i dg|ndg]
          [-o dg|ndg] ]
```

機能

拡張機能に関する環境を定義します。mcfmqgid コマンドを定義したとき、必ずこの mcfmexp コマンドを定義してください。また、mcfmqgid コマンドを省略したときは、mcfmexp コマンドを定義できません。

オプション

●-g サービスグループの登録数

～〈符号なし整数〉((1～8192))

この MCF マネージャ管理下の MCF で使用する MCF アプリケーション定義中のサービスグループの総数を含む予約最大数を指定します。

MCF アプリケーション定義の mcfaalcap 定義コマンドの-g オプションの quekind オペランドに、disk を指定したアプリケーションが属する MHP のサービスグループの総数を指定してください。なお、条件に該当するサービスグループの総数が 0 になった場合は、1 を指定してください。

●-l 論理端末の登録数

～〈符号なし整数〉((1～16380))

この MCF マネージャ管理下の MCF で使用する論理端末の総数を含む予約最大数を指定します。

指定する論理端末数には、MCF 通信プロセスまたはアプリケーション起動プロセスごとに、それぞれ次の条件を満たす論理端末の総数を指定してください。なお、条件に該当する論理端末の総数が 0 になった場合は、1 を指定してください。

- MCF 環境定義 (mcftenv) の-q オプションの diskitq オペランドに yes を指定、または指定を省略している場合

論理端末定義 (mcftalcle) の-k オプションの quekind オペランドの指定に関係なく、この MCF 通信プロセスまたはアプリケーション起動プロセスに定義している論理端末の総数を指定してください。

- MCF 環境定義 (mcftenv) の-q オプションの diskitq オペランドに no を指定している場合

論理端末定義（mcftalcle）の-k オプションの quekind オペランドに disk を指定した論理端末の総数を指定してください。

注意事項

- mcfmexp 定義コマンドの-g オプションと mcfmexp 定義コマンドの-l オプションには次の条件を満たす値を指定してください。

$$32767 \geq \text{サービスグループの登録数 (mcfmexp定義コマンドの-gオプションの指定値)} \\ + \text{論理端末の登録数 (mcfmexp定義コマンドの-lオプションの指定値)} \times 2 \\ + \text{MCF通信サービスの数 (mcfmcname定義コマンド指定数)} \times 5$$

条件を満たさない値を指定した場合、OpenTP1 開始時にメッセージ KFCA11033-W を出力します。

- mcfmexp 定義コマンドの-g オプションおよび mcfmexp 定義コマンドの-l オプションの指定値に応じて、静的共用メモリおよびステータスファイルの容量を見積もる必要があります。

静的共用メモリの見積もりは、「付録 B.1 TP1/Server Base 用の共用メモリの見積もり式」を参照してください。ステータスファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「ステータスファイルのサイズの見積もり式」の説明を参照してください。

- mcfmexp 定義コマンドの-g オプションおよび mcfmexp 定義コマンドの-l オプションの指定値は、オンラインの開始および終了の所要時間に影響しますので、必要以上に大きな値を指定しないでください。

●-i dg|ndg

～《ndg》

システム開始時に入力用ディスクキューが使用できない場合、メモリキューを代用して縮退運転をするかどうかを指定します。

dg：縮退運転をします。

ndg：縮退運転をしません。

●-o dg|ndg

～《ndg》

システム開始時に出力用ディスクキューが使用できない場合、メモリキューを代用して縮退運転をするかどうかを指定します。

dg：縮退運転をします。

ndg：縮退運転をしません。

名称

状態引き継ぎ定義

形式

```
mcfmsts  [-g サービスグループ数上限値]
          [-v サービス数上限値]
```

機能

再開始時の状態引き継ぎについて定義します。

状態を引き継ぐリソース数の上限値を指定します。この MCF サービスで使用するリソースに関して指定します。

オプション

●-g サービスグループ数上限値

～ 〈符号なし整数〉 ((1～8192))

状態を引き継ぐサービスグループの数を指定します。引き継ぐ対象は、mcftdctsg コマンドによる閉塞の状態、および mcftthldiq コマンドによる保留の状態です。アプリケーションの異常終了に伴う閉塞状態は引き継がれません。この MCF サービスで使用するサービスグループの総数以上を指定してください。

サービスグループの状態を引き継がない場合、このオプションを省略します。

指定値に応じてステータスファイルの容量を見積もる必要があります。ステータスファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「ステータスファイルのサイズの見積もり式」の説明を参照してください。

●-v サービス数上限値

～ 〈符号なし整数〉 ((1～1044480))

状態を引き継ぐサービスの数を指定します。引き継ぐ対象は、mcftdctsv コマンドによる閉塞の状態です。アプリケーションの異常終了に伴う閉塞状態は引き継がれません。この MCF サービスで使用するサービスの総数以上を指定してください。

サービスの状態を引き継がない場合、このオプションを省略します。

指定値に応じてステータスファイルの容量を見積もる必要があります。ステータスファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「ステータスファイルのサイズの見積もり式」の説明を参照してください。

名称

ログメッセージ出力抑止定義

形式

```
mcfmsmsg [-m " [抑止メッセージID [抑止メッセージID] ...] "
```

機能

ログメッセージの出力を抑止するときに定義します。

-m オプションで指定したメッセージ ID のログメッセージ出力（出力先：標準出力，標準エラー出力，およびメッセージログファイル）を抑止します。ログメッセージ出力要求が頻繁に発生すると，ログサービスに負担が掛かり，業務処理が遅延することがあります。このような場合に-m オプションを指定してログサービスへの負担を軽減してください。

オプション

●-m " [抑止メッセージ ID [抑止メッセージ ID] ...] "

～ 〈符号なし整数〉 ((10000～19999, 28000～29999))

ログファイルへの出力を抑止するメッセージの ID を指定します。

指定する抑止メッセージ ID はメッセージ通番（整数 5 けた）だけとします。

抑止メッセージ ID は最大 50 個指定できます。

なお，抑止対象外のメッセージ ID を指定した場合，そのメッセージは抑止できません。抑止対象外のメッセージを次に示します。

- MCF 以外が出力しているメッセージ
- MCF 開始中のメッセージ
- メッセージ出力時ログファイルに出力されないメッセージ
- 定義コマンド，および運用コマンドが出力するメッセージ

名称

サービスグループ属性定義

形式

```
[{{mcfmsvg -g "servgrp=サービスグループ名"  
  [-w " [watchcnt=入力キューの滞留監視数]  
        [watchint=入力キューの滞留監視インタバル時間]  
        [expectcnt=MHPに期待するサービス要求の処理数]  
        [abort=yes | no] " ] }}]
```

機能

サービスグループの属性について定義します。

オプション

●-g

(オペランド)

servgrp=サービスグループ名 ～ 〈1～31 文字の識別子〉

入力キューの滞留監視を行うサービスグループ名を指定します。

サービスグループ名は、ユーザサービス定義に指定してあるサービスグループ名を指定します。

●-w

(オペランド)

watchcnt=入力キューの滞留監視数 ～ 〈符号なし整数〉 ((0～65535)) 《0》

滞留数監視区間から処理能力判定区間に遷移する際の判断になる入力キューのメッセージ滞留数（しきい値）を指定します。

このオペランドの指定値以上のメッセージが入力キューに滞留している場合、処理能力判定区間に遷移します（watchint オペランドで指定した間隔で MHP の処理能力の判定を行います）。

このオペランドを省略した場合、または 0 を指定した場合は、入力キューの滞留監視は行いません。

watchint=入力キューの滞留監視インタバル時間 ～ 〈符号なし整数〉 ((5～3600)) 《10》 (単位：秒)

入力キューに滞留しているメッセージ数および MHP の処理能力を監視するインタバル時間を指定します。

watchcnt オペランドを省略した場合、または 0 を指定した場合は、このオペランドの指定値は無効です。

expectcnt=MHP に期待するサービス要求の処理数 ～ 〈符号なし整数〉 ((1～65535))

次の MHP の処理能力判定時まで処理完了を期待するサービス要求の処理数を指定します。

watchcnt オペランドに 1 以上を指定した場合に、このオペランドを省略したときは定義エラーになります。watchcnt オペランドを省略した場合、または 0 を指定した場合は、このオペランドの指定値は無効です。

このオペランドで指定する値は、次に示す算出式で見積もってください。

$$\text{MHPに期待するサービス要求の処理数} = \text{入力キューの滞留監視インタバル時間} \div \text{サービス要求1件当たりのMHPの処理時間} \times \text{利用率}$$

abort=yes | no ~ 《no》

MHP の処理能力の不足を検出した場合、OpenTP1 システムをダウンさせるかどうかを指定します。watchcnt オペランドを省略した場合、または 0 を指定した場合は、このオペランドの指定値は無効です。

yes

MHP の処理能力の不足を検出した場合に、KFCA11821-E メッセージを出力し、MHP および MCF マネジャプロセスを強制停止します。

no

MHP の処理能力の不足を検出した場合に、KFCA11820-W メッセージを出力し、処理を続行します。

MCF 通信構成定義

MCF 通信構成定義は、MCF 通信サービスに対して、実行環境を設定するものです。また、MCF 通信構成定義を作成することによって、MCF 通信サービスの起動を宣言します。複数の通信サービスを使用する場合、その数だけ MCF 通信構成定義を作成してください。

MCF 通信構成定義は、次の二つの定義から構成されます。

- 共通定義
- データコミュニケーション定義

共通定義とデータコミュニケーション定義は、別々のファイルに作成します。最終的には二つの定義を結合して、一つの定義オブジェクトファイルを生成します。定義オブジェクトファイルの生成については、[「4.3 定義オブジェクトファイルの生成」](#)を参照してください。

定義内容の概略は次のとおりです。

共通定義

各 MCF 通信サービスで共通して必要な環境情報を定義します。共通の定義コマンドを使用します。ただし、識別子などの指定値は、定義間で重複して指定できません。通信サービスごとに、別々の共通定義ソースファイルを作成してください。ただし、通信サービスの種類によっては、固有のオプションがあります。これについては、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

データコミュニケーション定義

各 MCF 通信サービスに固有の環境情報を定義します。通信サービスに固有の定義コマンドを使用します。

データコミュニケーション定義は、通信サービスの性格によって次の 2 種類に分けられます。

- プロトコル固有定義
- アプリケーション起動定義

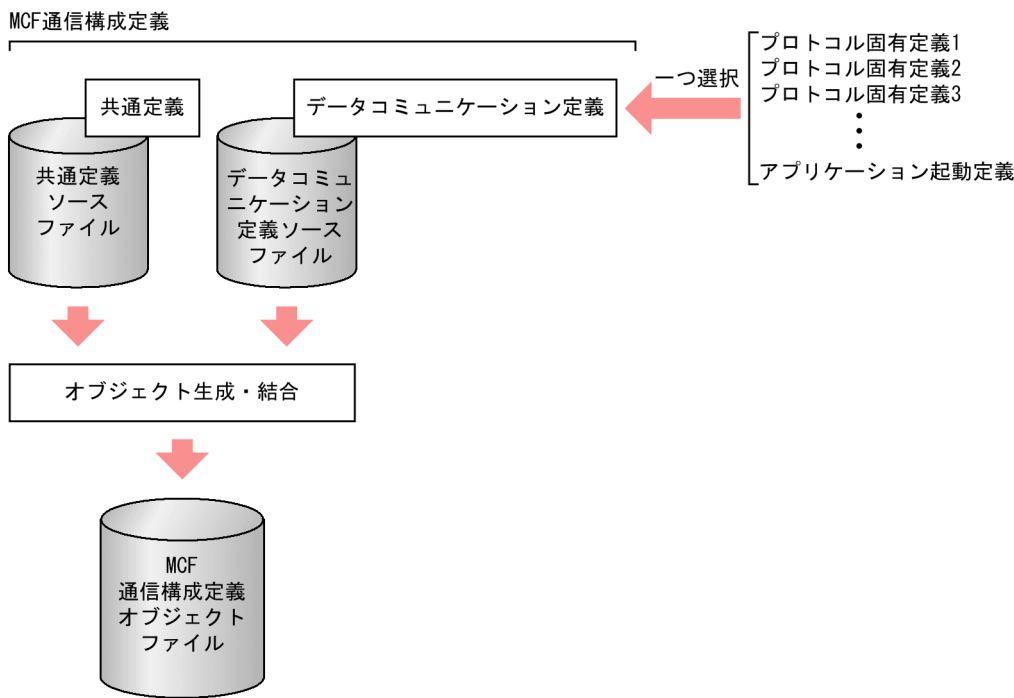
外部通信のための通信サービス（MCF 通信プロセス）に対して MCF 通信構成定義を作成する場合、プロトコル固有定義コマンドを使用します。プロトコル固有定義コマンドの詳細については、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

ノード内通信のための通信サービス（アプリケーション起動プロセス）に対して MCF 通信構成定義を作成する場合、アプリケーション起動定義コマンドを使用します。

各データコミュニケーション定義で共通する定義コマンドがありますが、指定するオプションなどが多少異なります。適切な通信サービスの定義コマンドを参照してください。

次の図に、MCF 通信構成定義のファイル構成を示します。

図 5-2 MCF 通信構成定義のファイル構成



共通定義コマンドの指定順序

共通定義コマンドの指定順序を次の図に示します。

図 5-3 共通定義コマンドの指定順序

mcftenv	(MCF環境定義)
mcftcomn	(MCF通信構成共通定義)
[mcfttred]	(最大処理多重度定義)
[mcfttim]	(タイマ定義)
mcfttrc	(トレース環境定義)
[mcftsts]	(状態引き継ぎ定義)
[{{ mcftbuf }} 繰り返し可] (バッファグループ定義) ※	

注※ アプリケーション起動プロセスでは、このバッファを使用しません。
対応するMCF通信構成定義では、このmcftbufコマンドは省略します。

アプリケーション起動定義コマンドの指定順序

アプリケーション起動定義コマンドの指定順序を次の図に示します。

図 5-4 アプリケーション起動定義コマンドの指定順序

mcftpsvr	(アプリケーション起動環境定義の開始)
{{ mcftalcle }} 繰り返し可	(論理端末定義)
mcftped	(アプリケーション起動環境定義の終了)

名称

MCF 環境定義

形式

```
mcftenv -s MCF通信プロセス識別子|アプリケーション起動プロセス識別子
        [-m MCFマネジャプロセス識別子]
        -a MCFアプリケーション定義オブジェクトファイル名
        -q [diskitq=yes|no]
```

機能

MCF に共通する環境を定義します。

オプション

●-s MCF 通信プロセス識別子|アプリケーション起動プロセス識別子

～ 〈数字 (0～9), a～f) ((01～ef))

MCF 通信プロセスの識別子, またはアプリケーション起動プロセスの識別子を指定します。ここで指定する識別子は, ほかの mcftenv コマンドで指定する識別子と重複してはいけません。

MCF 通信プロセス識別子

この MCF 通信構成定義が MCF 通信プロセスに対応するとき, MCF 通信プロセス識別子を指定します。MCF 通信プロセス識別子は, MCF が出力するメッセージログなどで, MCF 通信プロセスを特定する場合に使用します。また, 運用コマンドを入力するときの, MCF 通信プロセスの指定に使用します。

アプリケーション起動プロセス識別子

この MCF 通信構成定義がアプリケーション起動プロセスに対応するとき, アプリケーション起動プロセス識別子を指定します。ここで指定したアプリケーション起動プロセス識別子を, 対応する MCF アプリケーション定義の mcfaenv コマンドの-p オプションで指定します。

●-m MCF マネジャプロセス識別子

～ 〈英字〉 ((A～Z, a～z)) 《A》

該当する MCF 通信プロセスを管理する MCF マネジャプロセスの識別子を指定します。MCF マネジャプロセス識別子は, 該当する MCF マネジャ定義内の mcfmenv コマンドで指定した値を指定します。

●-a MCF アプリケーション定義オブジェクトファイル名

～ 〈1～8 文字の英数字〉

この MCF 通信構成定義に対応する MCF アプリケーション定義オブジェクトファイル名を指定します。

この MCF アプリケーション定義オブジェクトファイル名は、MCF 定義オブジェクト生成ユーティリティ (mcfapli) で生成する MCF アプリケーション定義の出力オブジェクトファイル名を指定します。

●-q

(オペランド)

diskitq=yes|no ～《yes》

該当する MCF 通信サーバから起動されるアプリケーションが、入力キューにディスクキューを使用するかどうかを指定します。

入力キューにディスクキューを使用しない (diskitq=no) と指定した場合、MCF の開始処理が速くなります。

なお、ご使用の OS が Windows の場合、このオプションは指定できません。

yes

一部またはすべてのアプリケーションが入力キューにディスクキューを使用します (MCF アプリケーション定義 mcfaalcap の -g オプションの quekind オペランドの指定値に従います)。

no

すべてのアプリケーションが入力キューにディスクキューを使用しません。

ただし、このオペランドで no を指定した MCF 通信サーバが、MCF アプリケーション定義の mcfaalcap コマンドの -g オプションで quekind=disk を指定したアプリケーションを起動する場合、この指定値を無視して、入力キューにメモリキューを使用します。

名称

MCF 通信構成共通定義

形式

mcftcomn	[-j MCF通信プロセスまたはアプリケーション起動プロセスのジャーナルバッファの大きさ]

機能

MCF 通信プロセスまたはアプリケーション起動プロセスのジャーナルバッファの大きさを定義します。

オプション

- j MCF 通信プロセスまたはアプリケーション起動プロセスのジャーナルバッファの大きさ
～ 〈符号なし整数〉 ((4096～4000000 の 4 の倍数)) 《32768》 (単位：バイト)

ジャーナルデータを蓄えておく領域の大きさを指定します。

MCF 通信プロセスまたはアプリケーション起動プロセスが、ジャーナルサービスに対して次の要求を出すまで、この領域にジャーナルデータを蓄えておきます。

- ・ ジャーナルファイルへの出力要求
- ・ ジャーナルサービスのバッファへの格納要求

ジャーナルバッファの大きさには、IJ, MJ, CJ のデータ長のうち、最大値以上の値を指定してください。

1. IJ のデータ長

MCF アプリケーション定義の mcfaalcap 定義コマンドの-j オプションの ij オペランドに yes を指定したときに取得します。

- 32 ビット版のとき

$$\uparrow (172 + \text{seg}) / 4 \uparrow \times 4$$

- 64 ビット版のとき

$$\uparrow (172 + \text{seg}) / 8 \uparrow \times 8$$

2. MJ のデータ長

mcftactmj コマンドを実行したときに取得します。

- 32 ビット版のとき

$$\uparrow (180 + \text{seg}) / 4 \uparrow \times 4$$

- 64 ビット版のとき

$$\uparrow (180 + \text{seg}) / 8 \uparrow \times 8$$

3. CJ (メッセージキューサーバ) のデータ長

ITQ, OTQ にディスクキューを使用した場合に取得します。

次に示す二つの計算式のうち最大の値をデータ長としてください。

計算式 1 :

●32 ビット版のとき

$$\uparrow \{88 + (24 \times (\uparrow \text{msg} / \text{ql} \uparrow + \uparrow 960 / \text{ql} \uparrow)) + \text{qio}\} / 4 \uparrow \times 4$$

●64 ビット版のとき

$$\uparrow \{88 + (24 \times (\uparrow \text{msg} / \text{ql} \uparrow + \uparrow 1072 / \text{ql} \uparrow)) + \text{qio}\} / 8 \uparrow \times 8$$

計算式 2 :

●32 ビット版のとき

$$260 + (24 \times m)$$

●64 ビット版のとき

$$\uparrow \{260 + (24 \times m)\} / 8 \uparrow \times 8$$

(凡例)

$\uparrow \uparrow$: 小数点以下を切り上げます。

seg : セグメント長

msg : ディスクキューを使用した受信メッセージ長または送信メッセージ長

ql : キューファイル物理レコード長

qio : メッセージキューサービス定義の que_io_maxrecsize オペランドの指定値

m : 保持メッセージ数 (メッセージキューサービス定義の quegrp 定義コマンドの -m オプションで指定した値)

ユーザが指定した値によって, MCF は次の処理をします。

1. 指定した値が MCF で定めた下限値より小さい場合, その値を MCF で定めた下限値に補正します。
2. 指定した値が MCF で定めた上限値より大きい場合, その値を MCF で定めた上限値に補正します。
3. 指定した値が 4 の倍数でない場合, その値を 4 の倍数になるように切り上げます。

名称

最大処理多重度定義

形式

```
[mcfttred [-m 最大処理多重度] ]
```

機能

MCF 通信プロセスの最大処理多重度を定義します。アプリケーション起動プロセスの場合は、この定義コマンドは省略してください。

オプション

●-m 最大処理多重度

～ 〈符号なし整数〉 ((1～1000)) 《10》

MCF 通信プロセスで、並行してメッセージ受信およびメッセージ送信を処理するコネクション数の最大値を指定します。

MCF 通信プロセスは、複数のコネクションから同時期に受信したメッセージおよび UAP から、複数のコネクションに対して同時期に送信要求されたメッセージを、並行して処理できます。並行して処理できる多重度の上限値は、このオプションで指定します。ただし、高トラフィック状態でも、システム的环境によっては、定義で指定した値まで並行処理しない場合があります。この場合、むだにローカルメモリを使用していることになり、かえって性能に悪影響を与えていることがあります。最大処理多重度定義の指定値を 1 増すごとに、MCF 通信プロセスが使用するローカルメモリが約 32 キロバイト（AIX 5L の場合は約 64 キロバイト）増加します。

このオプションのチューニングは、メッセージ多重処理状況の表示コマンド（mcftlstrd）を実行した結果、または MCF 稼働統計情報を取得した結果から、現状の多重処理状況を把握してから行ってください。

なお、最大処理多重度定義の指定値よりコネクション定義の指定数が小さい場合、最大処理多重度は、コネクション定義の指定数になります。

名称

タイマ定義

形式

```
[mcfttim  [-t " [btim=時間監視間隔]
            [mtim=未処理送信メッセージ滞留時間]
            [rmtim=未処理受信メッセージ滞留時間] ")
          [-p " [usertime=yes|no]
            [timereqno=最大タイマ監視要求数]
            [msgsize=最大メッセージ長]
            [msgout=yes|no] ") ]
```

機能

MCF の時間監視に関する環境を定義します。

オプション

●-t

(オペランド)

btim=時間監視間隔 ～ 〈符号なし整数〉 ((1～60)) 《1》 (単位：秒)

MCF 通信サービスで使用する時間監視間隔を指定します。

MCF で時間監視をする場合、指定した時間が経過したかどうか、この間隔ごとに調べます。ほかのオペランドで指定した監視時間値の誤差は、時間監視間隔の大きさに影響して大きくなります。

mtim=未処理送信メッセージ滞留時間 ～ 〈符号なし整数〉 ((60～65535)) 《180》 (単位：秒)

未処理送信メッセージの滞留時間を指定します。

MCF の終了処理時に閉塞解除されている論理端末の出力キューに残っている未処理送信メッセージの滞留時間を監視します。終了処理が長時間終了しないのを防止できます。ただし、正常終了のときだけ有効です。

指定時間を超えても未処理送信メッセージが残っている場合は、送信されたものと見なして、終了処理を続行します。この場合、未処理送信メッセージを破棄して、**未処理送信メッセージ廃棄通知イベント (ERREVTa)** が通知されます。

ただし、dc_mcf_execap 関数によるタイマ起動要求メッセージや閉塞されている論理端末の出力キューに残っている未処理送信メッセージは、未処理送信メッセージ滞留時間の監視対象とはなりません。

OpenTP1 の正常終了コマンド実行時にメッセージが残っていた場合は、すぐにメッセージが破棄され、ERREVTa が通知されます。

rmtim=未処理受信メッセージ滞留時間 ～ 〈符号なし整数〉 ((0～65535)) 《0》 (単位：秒)

未処理受信メッセージの滞留時間を指定します。

MCF の終了処理時に入力キューに残っている未処理受信メッセージの滞留時間を監視します。終了処理が長時間終了しないのを防止できます。ただし、正常終了、計画停止 A のときだけ有効です。

指定時間を超えても未処理受信メッセージが残っている場合、MCF は異常終了します。

ここで 0 を指定した場合、時間を監視しません。入力キューに残っている未処理受信メッセージがなくなるまで待ちます。

●-p

(オペランド)

usertime=yes|no ～《no》

ユーザタイム監視機能を使用するかどうかを指定します。

yes

ユーザタイム監視機能を使用します。

no

ユーザタイム監視機能を使用しません。

timereqno=最大タイム監視要求数 ～〈符号なし整数〉((1～10000))《16》

ユーザタイム監視の要求数の最大値を指定します。

指定値に応じて静的共用メモリの容量を見積もる必要があります。静的共用メモリの見積もりは、「[付録 B.2 MCF サービス用の共用メモリの見積もり式](#)」を参照してください。

msgsize=最大メッセージ長 ～〈符号なし整数〉((0～256))《0》(単位：バイト)

指定した時間を超えた場合に、起動する MHP に渡すメッセージのセグメントの最大長を指定します。

指定値に応じて静的共用メモリの容量を見積もる必要があります。静的共用メモリの見積もりは、「[付録 B.2 MCF サービス用の共用メモリの見積もり式](#)」を参照してください。

msgout=yes|no ～《no》

次に示す場合のメッセージを出力するかどうかを指定します。

- ・ ユーザタイム監視の取り消し関数が、DCMCFER_PARAM_TIM_ID (C 言語)、または 70910 (COBOL 言語) でエラーリターンした場合に KFCA16518-I を出力します。
- ・ タイムアウト発生時に KFCA16519-I を出力します。

yes

メッセージを出力します。

no

メッセージを出力しません。

名称

トレース環境定義

形式

```
mcfttrc  [-t " [size=トレースバッファの大きさ]
           [disk=yes|no]
           [bufcnt=トレースバッファの数]
           [trccnt=トレースファイルの数]
           [msgsize=トレースとして取得する送受信メッセージ
                 の最大サイズ] " ]
          [-m del|off]
```

機能

MCF のトレースに関する環境を定義します。

MCF トレースは、MCF に関する情報を取得し、トラブル解析用に使用します。

オプション

●-t

(オペランド)

size=トレースバッファの大きさ ～〈符号なし整数〉((4096～15728640 の 4 の倍数))《204800》(単位：バイト)

MCF トレースを格納するためのバッファの大きさを指定します。

通常は省略値を使ってください。

ユーザが指定した値によって、MCF は次の処理をします。

1. 指定した値が MCF で定めた下限値より小さい場合、その値を MCF で定めた下限値に補正します。
2. 指定した値が MCF で定めた上限値より大きい場合、その値を MCF で定めた上限値に補正します。
3. 指定した値が 4 の倍数でない場合、その値を 4 の倍数になるように切り上げます。

指定値に応じて静的共用メモリの容量を見積もる必要があります。静的共用メモリの見積もりは、「[付録 B.2 MCF サービス用の共用メモリの見積もり式](#)」を参照してください。

disk=yes|no ～《yes》

MCF トレースのディスク出力機能を使用するかどうかを指定します。

yes

ディスク出力機能を使用します。MCF トレース情報でトレースバッファが満杯になった時に、バッファの内容をディスクに出力（スワップ）します。

MCF トレースファイルは、\$DCDIR/spool/ディレクトリの下に、mcftXXXnn (XXX は MCF 識別子、nn は通し番号) というファイル名で出力されます。

MCF トレースファイルの詳細については、マニュアル「OpenTP1 運用と操作」の「MCF トレースに関する運用」および「OpenTP1 が出力するファイル一覧」の説明を参照してください。

なお、MCF 通信プロセスおよびアプリケーション起動プロセスが異常終了したとき、最新トレース情報を mcftXXX0 というファイルに出力する場合があります。このファイルは、trccnt オペランドで指定するトレースファイルの数に含まれません。

no

ディスク出力機能を使用しません。メモリにだけ出力します。

disk オペランドを省略した場合は、MCF トレースファイルを生成しません。メモリ上の 2 面バッファを交互に使用します。

また、disk オペランドを省略した場合は、次の bufcnt オペランド、trccnt オペランド、および-m オプションの指定は無効になります。

bufcnt=トレースバッファの数 ～〈符号なし整数〉((10～2147483647))《100》

トレースバッファの数を指定します。

MCF トレースファイルにトレースバッファを何面分格納するかを指定します。

なお、MCF トレースファイルの大きさが次の条件を満たすようにトレースバッファの大きさ (mcfttrc -t size の指定値) とトレースバッファの数 (mcfttrc -t bufcnt の指定値) を指定してください。

MCF トレースファイルの大きさ=トレースバッファの数 (mcfttrc -t bufcnt の指定値) ×トレースバッファの大きさ (mcfttrc -t size の指定値) < 2147483648 バイト

trccnt=トレースファイルの数 ～〈符号なし整数〉((3～99))《3》

MCF トレースファイルの数を指定します。

スワップ済みトレースファイル (bufcnt オペランドで指定した数のスワップを終了したトレースファイル) とスワップ用トレースファイルの合計を指定します。

msgsize=トレースとして取得する送受信メッセージの最大サイズ ～〈符号なし整数〉((0～1073741824))《128》(単位: バイト)

MCF はメッセージ受信時およびメッセージ送信時に、そのメッセージの一部またはすべてをトレースとして取得します。メッセージの先頭からどれだけの内容をトレースに取得するか、その最大サイズを指定します。

0 を指定した場合、メッセージの内容をトレースに取得しません。メッセージのサイズがこの指定値より小さい場合は、メッセージのすべての内容をトレースとして取得します。

このオペランドは、TP1/NET/TCP/IP を使用した MCF 通信サービスに対して有効です。

●-m del|off

～《del》

スワップ済みトレースファイルの数と、trccnt オペランドの指定値との関係を指定します。

del

スワップ済みトレースファイルの数と、trccnt オペランドの指定値が一致した場合、いちばん古いスワップ済みトレースファイルを削除し、そのファイル名でスワップ用トレースファイルを作成します。

off

trccnt オペランドの指定値を無視します。スワップ済みトレースファイルの数が 99 になった場合、いちばん古いスワップ済みトレースファイルを削除し、そのファイル名でスワップ用トレースファイルを作成します。

ディスクの空き容量を圧迫しますので、スワップ済みトレースファイルは、運用者の責任で処理したあと、削除してください。

名称

状態引き継ぎ定義

形式

```
[mcftsts [-a アプリケーション数上限値]
          [-l 論理端末数上限値] ]
```

機能

再開始時の状態引き継ぎについて定義します。

状態を引き継ぐリソース数の上限値を指定します。この MCF 通信サービスで使用するリソースに関して指定します。

オプション

●-a アプリケーション数上限値

～ 〈符号なし整数〉 ((1～8192))

状態を引き継ぐアプリケーションの数を指定します。引き継ぐ対象は、mcfadctap コマンドによる閉塞の状態です。アプリケーションの異常終了に伴う閉塞状態は引き継がれません。この MCF 通信サービスで使用するアプリケーションの総数以上を指定してください。

アプリケーションの状態を引き継がない場合、このオプションを省略します。

指定値に応じてステータスファイルの容量を見積もる必要があります。ステータスファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「ステータスファイルのサイズの見積もり式」の説明を参照してください。

●-l 論理端末数上限値

～ 〈符号なし整数〉 ((1～2048))

状態を引き継ぐ論理端末の数を指定します。引き継ぐ対象は、出力キューの入力、および出力の保留／保留解除状態です。この MCF 通信サービスで使用する論理端末の総数以上を指定してください。

論理端末の状態を引き継がない場合、このオプションを省略します。

指定値に応じてステータスファイルの容量を見積もる必要があります。ステータスファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「ステータスファイルのサイズの見積もり式」の説明を参照してください。

名称

バッファグループ定義

形式

```
{ {mcftbuf  -g  "groupno=バッファグループ番号  
                length=バッファ長  
                count=バッファ数  
                [extend=拡張バッファ数]  " } }
```

機能

メッセージ送受信用、およびメッセージ編集用のバッファに関する環境を定義します。

アプリケーション起動プロセスに対応する MCF 通信構成定義では、この mcftbuf コマンドは省略します。

オプション

●-g

(オペランド)

groupno=バッファグループ番号 ～ 〈符号なし整数〉 ((1～512))

メッセージ送受信用、およびメッセージ編集用のバッファグループ番号を指定します。

バッファグループ番号は、ほかの mcftbuf コマンドの groupno オペランドで指定するバッファグループ番号と通信プロセス単位で重複して指定できません。

length=バッファ長 ～ 〈符号なし整数〉 ((512～1073741824)) (単位：バイト)

メッセージ送受信用、およびメッセージ編集用のバッファ長を指定します。

詳細は、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編の、mcftalccn 定義コマンド (TP1/NET/UDP の場合は mcftalcle 定義コマンド) の注意事項を参照してください。

count=バッファ数 ～ 〈符号なし整数〉 ((1～65535))

メッセージ送受信用、およびメッセージ編集用のバッファ数を指定します。

詳細は、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編の、mcftalccn 定義コマンド (TP1/NET/UDP の場合は mcftalcle 定義コマンド) の注意事項を参照してください。

なお、mcftlsbuf コマンドに -m オプションを指定して実行すると、最大バッファ使用数を確認できます。mcftlsbuf 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。

extend=拡張バッファ数 ～ 〈符号なし整数〉 ((0～65535)) 《0》

count オペランドで指定したバッファがいっぱいになった場合に拡張するバッファの数を指定します。

名称

アプリケーション起動環境定義の開始

形式

```
mcftpsvr -c 内部通信路名  
          [-o "reruntm=yes|no"]
```

機能

アプリケーションプログラムの起動に関する環境を定義します。

オプション

●-c 内部通信路名

～ 〈1～8 文字の識別子〉

アプリケーション起動プロセスで使用する内部通信路名を指定します。

内部通信路とは、同一ノード内のアプリケーション間でメッセージを受け渡すための、仮想的な通信路のことです。

ほかのアプリケーション起動プロセスで使用する内部通信路名と重複しないように指定してください。重複している場合は、プロセスが異常終了します。

●-o

(オペランド)

reruntm=yes|no ～ 《no》

リラン時のタイマ起動を引き継ぐかどうかを指定します。

タイムアウトが発生する前（指定した時間が過ぎる前）のタイマ起動要求が残った状態で OpenTP1 のリランが発生した場合、タイマ起動要求を引き継ぐかどうかを指定してください。

タイムアウトが発生したあとのタイマ起動要求の場合、このオプション指定に該当しません。タイムアウトが発生したあとのタイマ起動要求は、このオプションの指定に関係なく引き継がれます。

yes

リラン時のタイマ起動を引き継ぎます。また、アプリケーション起動サービスのメイン関数でタイマ起動引き継ぎ決定 UOC を組み込んだ場合、yes を指定してください。

なお、引き継ぎ対象となるのは、次の条件を満たしているアプリケーションです。

- ・アプリケーション属性定義（mcfaalcap -g quekind）で disk を指定している

- 内部通信路 (mcfaalcap -n cname), または内部通信路上の論理端末 (mcfaalcap -n lname) に対応するアプリケーション起動用論理端末定義 (mcftalcle -k quekind) で disk を指定している

no

リラン時のタイマ起動を引き継ぎません。すべてのタイマ起動を破棄します。

アプリケーション起動サービスのメイン関数で、タイマ起動引き継ぎ決定 UOC を組み込んでも、no を指定した場合は、タイマ起動引き継ぎ決定 UOC をコールしません。

名称

アプリケーション起動用論理端末定義

形式

```
{ {mcftalcle  -l  論理端末名称
               -t  send|request
               [-m  " [mmsgcnt=メモリ出力メッセージ最大格納数]
                   [dmsgcnt=ディスク出力メッセージ最大格納数] " ]
               [-k  " [quekind=memory|disk]
                   [quegrpID=キューグループID] " ] } }
```

機能

アプリケーション起動用の論理端末に関する環境を定義します。

エラーイベント処理用 MHP およびリトライ指示のロールバック要求のための論理端末は、MCF で自動的に生成します。ここで定義する必要はありません。アプリケーション起動機能を使用する場合は、論理端末を一つ以上定義する必要があります。

指定数に応じて静的共用メモリ、およびステータスファイルの容量を見積もる必要があります。静的共用メモリの見積もりは、「付録 B.2 MCF サービス用の共用メモリの見積もり式」を参照してください。ステータスファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「ステータスファイルのサイズの見積もり式」の説明を参照してください。

オプション

●-l 論理端末名称

～ 〈1～8 文字の識別子〉

論理端末名称を指定します。

OpenTP1 システム内で一意となる論理端末名称を指定してください。

●-t send|request

この論理端末の端末タイプを指定します。

send

一方送信型論理端末

非応答型のアプリケーションを起動するとき、この端末タイプを使用します。

request

問い合わせ型論理端末

応答型のアプリケーションを起動するとき、この端末タイプを使用します。

●-m

(オペランド)

mmsgcnt=メモリ出力メッセージ最大格納数 ～ 〈符号なし整数〉 ((0～65535)) 《0》

メモリで待ち合わせをする出力メッセージの最大格納数を指定します。

出力メッセージの待ち合わせ数が指定した最大数になると、それ以後 UAP からの送信要求 (SEND) はエラーリターンとなります。

0 を指定、または指定を省略した場合、メモリで待ち合わせをする出力メッセージ数は指定可能な最大数 (65535) になります。

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

ディスクで待ち合わせをする出力メッセージの最大格納数を指定します。

出力メッセージの待ち合わせ数が指定した最大数になると、それ以後 UAP からの送信要求 (SEND) はエラーリターンとなります。

0 を指定、または指定を省略した場合、ディスクで待ち合わせをする出力メッセージ数は指定可能な最大数 (65535) になります。

●-k

(オペランド)

quekind=memory|disk ～ 《memory》

出力メッセージの割り当て先 (メモリキューまたはディスクキュー) を指定します。

memory

メモリキューだけに割り当てます。

disk

ディスクキューおよびメモリキューに割り当てます。

disk を指定した場合、必ず quegrpID オペランドを指定してください。

quegrpID=キューグループ ID ～ 〈1～8 文字の識別子〉

ディスクで待ち合わせをする出力メッセージに使用するキューグループ ID を指定します。MCF マネージャ定義の mcfmqgid コマンドで指定するキューグループ ID (キューグループ種別は otq) のどれかを指定します。

この quegrpID オペランドは、quekind オペランドで disk を指定した場合だけ指定します。

mcftped

名称

アプリケーション起動環境定義の終了

形式

mcftped

機能

アプリケーション起動環境定義の終了を示します。

オプション

なし。

MCF アプリケーション定義

MCF アプリケーション定義は、アプリケーションを定義し、その実行環境を設定するものです。一つの MCF アプリケーション定義に対し、一つのソースファイルを作成してください。

アプリケーションとは、一つの業務のことです。同じサービスでも、使用する通信サービスの違いによって、別々のアプリケーションとして定義できます。

MCF アプリケーション定義コマンドの指定順序を次の図に示します。

図 5-5 MCF アプリケーション定義コマンドの指定順序

```
mcfaenv          (アプリケーション環境定義)
{{ mcfaalcap }} 繰り返し可 (アプリケーション属性定義)
```

アプリケーション属性定義を指定するとき、mcfaalcap 定義コマンドの-N オプションの modelname オペランドを使用すると、モデルとする定義の指定内容を流用できます。

このとき、モデルとするアプリケーション属性定義の一部だけを流用して、新しい定義を作成することもできます。例えば、アプリケーション名称「ex01」の属性定義の一部を流用してアプリケーション名称「ex02」の属性定義をする場合、次の図に示すように定義できます。

図 5-6 モデルアプリケーション属性定義の指定例

●モデルとするアプリケーション属性定義
(アプリケーション名称: ex01)

mcfaalcap	-n " name = ex01	¥
	kind = user	¥
	type = cont	¥
	msgcnt = 600	¥
	lname = copsvr01 "	¥
	-g " servgrp = uap02	¥
	quekind = memory "	¥
	-v " servname = serv032	¥
	servhold = m "	

●新しく作成するアプリケーション属性定義
(アプリケーション名称: ex02)

mcfaalcap	-n " name = ex02	¥
	-N " modelname = ex01 "	¥
	-v " servname = serv033 "	



●ex02に定義される内容

mcfaalcap	-n " name = ex02	¥
	kind = user	¥
	type = cont	¥
	msgcnt = 600	¥
	lname = copsvr01 "	¥
	-N " modelname = ex01 "	¥
	-g " servgrp = uap02	¥
	quekind = memory "	¥
	-v " servname = serv033 "	¥
	servhold = m "	

ex02の指定
内容が適用
される

ex01の指定
内容が流用
される

名称

アプリケーション環境定義

形式

```
mcfaenv  -a MCFアプリケーション定義識別子  
          [-p アプリケーション起動プロセス識別子]
```

機能

アプリケーションに関する環境を定義します。

オプション

●-a MCF アプリケーション定義識別子

～ 〈1～8 文字の識別子〉

この MCF アプリケーション定義の識別子を指定します。

MCF アプリケーション定義識別子は、MCF 起動時に表示される MCF アプリケーション定義の識別子として使用します。

●-p アプリケーション起動プロセス識別子

～ 〈数字 (0～9), a～f) ((01～ef))

この MCF アプリケーション定義に対応するアプリケーション起動プロセス識別子を指定します。ただし、次の場合に指定します。

- アプリケーション起動機能を使用する場合
- エラーイベントを起動する場合
- リトライ指示のロールバック要求をする場合

アプリケーション起動プロセス識別子は、MCF 通信構成定義の mcftenv コマンドの-s オプションで指定したものを指定します。

注意事項

このオプションに、MCF 通信プロセス識別子を指定しないでください。指定した場合、エラーが発生します。

名称

アプリケーション属性定義

形式

```
mcfaalcap -n " name=アプリケーション名|MCFイベント名
               {kind=user|mcf}
               {type=ans|noans|cont}
               {aplihold=m|a|s}
               {msgcnt=入力メッセージ最大格納数}
               {lname=論理端末名称}
               {cname=内部通信路名}
               {tempsize=継続問い合わせ応答用一時記憶データ格納用
                       領域サイズ}
               {trnmode=trn|nontrn}
               {errevt=yes|no} "
[-N " modelname=モデルアプリケーション名|モデルMCFイベント名"]
-g " servgrp=サービスグループ名
     {quegrpid=キューグループID}
     {quekind=memory|disk}
     {srvghold=m|s}
     {recvmmsg=e|r} "
-v " servname=サービス名
     {servhold=m|a|s}
     {ntmetim=非トランザクションMHP限界経過時間} "
[-d " {holdlimit=アプリケーション異常終了限界回数}
     {holdlmtyp=sum|cont}
     {reschedulecnt=アプリケーション異常終了時の
                     再スケジュール回数}
     {rescheduleint=アプリケーション異常終了時の
                     再スケジュール間隔}
     {reschedulelog=yes|no} " ]
[-j " {ij=yes|no}
     {oj=yes|no}
     {gj=yes|no} " ]
[-e " evtlogout=yes|no"]
```

機能

アプリケーションに関する属性を定義します。

指定数に応じて静的共用メモリの容量を見積もる必要があります。静的共用メモリの見積もりは、「[付録 B.2 MCF サービス用の共用メモリの見積もり式](#)」を参照してください。

オプション

●-n

(オペランド)

name=アプリケーション名|MCF イベント名 ～ 〈1～8 文字の識別子〉

使用するアプリケーション名、または MCF イベント名を指定します。

name オペランドの指定値は kind オペランドで指定するアプリケーションの種別内で、ほかの mcfaalcap コマンドの name オペランドの指定値とは異なる指定をします。

mcfuevt コマンドで直接 MHP を起動したい場合は、UCMDEV T を指定してください。

ERREVT2 の一部、および ERREVT3 を起動する場合、アプリケーション起動プロセスが必要です。アプリケーション起動プロセス用の MCF 通信構成定義を作成してください。

kind=user|mcf ～ 《user》

アプリケーションの種別を指定します。

user

ユーザのアプリケーション

mcf

MCF イベント

name オペランドに UCMDEV T を指定した場合は、user を指定するか、指定を省略してください。

誤って指定すると、MCF 定義オブジェクト生成ユーティリティ起動コマンド (mcfapli) は、KFCA11513-E メッセージを表示し、異常終了します。

type=ans|noans|cont ～ 《noans》

アプリケーションの型を指定します。

ans：応答型

noans：非応答型

cont：継続問い合わせ応答型

name オペランドで MCF イベント名を指定した場合、この type オペランドの指定は無効になります。MCF は MCF イベントに対して次のように自動的に設定します。

1. 次の MCF イベントの場合、要求元となった論理端末の端末タイプに応じて、reply 型論理端末の場合は ans を、reply 型以外の論理端末の場合は noans を設定します。

- ・不正アプリケーション名検出通知イベント (ERREVT1)

2. 次の MCF イベントの場合、それを起動する要因となったアプリケーションの型をそのまま引き継ぎます。

- ・メッセージ廃棄通知イベント (ERREVT2)

- ・UAP 異常終了通知イベント (ERREVT3)

- ・タイマ起動メッセージ廃棄通知イベント (ERREVT4)

3. 上記以外の MCF イベントの場合は、noans を設定します。

異常が発生した場合、起動元通信プロセスに対応する MCF イベントが起動されます。アプリケーション起動機能を使用して複数の MCF アプリケーション定義を経由した場合も、最初の起動元通信プロセスに対応する MCF イベントが起動されます。

SPP からアプリケーションを起動し、起動元通信プロセスがない場合は、アプリケーション起動プロセスに対応する MCF イベントが起動されます。

mcfuevt コマンドで直接 MHP を起動したい場合は、noans を指定するか、指定を省略してください。誤って指定すると、MCF 定義オブジェクト生成ユーティリティ起動コマンド (mcfapli) は、KFCA11513-E メッセージを表示し、異常終了します。

aplihold=m|a|s ～ 《m》

アプリケーションが異常終了した場合のアプリケーションの処置を指定します。

m

アプリケーションを閉塞しません。

a

アプリケーションを閉塞します。入力およびスケジュールを禁止します。

s

アプリケーションを閉塞します。スケジュールを禁止します。

次の場合には、この定義の指定値に関係なく、アプリケーションを閉塞しません。

- dc_mcf_rollback 関数に DCMCFRTRY を指定 (CBLDCMCF('ROLLBACK')に RTRY を指定) してロールバックした場合
- アプリケーション属性定義 (mcfaalcap) または UAP 共通定義 (mcfmuap) の reschedulecnt オペランドの指定値が 1 以上の場合に、MHP が異常終了したとき
- アプリケーションに対応するサービス開始前にユーザサービスが異常終了した場合

msgcnt=入力メッセージ最大格納数 ～ 〈符号なし整数〉 ((0～65535)) 《0》

入力するメッセージの最大格納数を指定します。

0 を指定した場合、メモリまたはディスクに入力するメッセージの数は無制限になります。

lname=論理端末名称 ～ 〈1～8 文字の識別子〉

アプリケーション起動機能を使用するときの、内部通信路上の論理端末名称を指定します。-n オプションの type オペランドで noans を指定した場合だけ指定できます。

この論理端末名称は、アプリケーション起動プロセスに対応する MCF 通信構成定義の、mcftalcle コマンドで登録したものを指定します。ここでは、一方送信型論理端末を指定します。

ほかのアプリケーションからこのアプリケーションを起動するとき、ここで指定した論理端末を使用してメッセージが受け渡されます。

アプリケーション起動機能使用時の定義方法については、cname オペランドの説明を参照してください。

cname=内部通信路名 ～ 〈1～8 文字の識別子〉

アプリケーション起動機能を使用するときの、内部通信路名を指定します。ただし、type オペランドで ans、または cont を指定したときだけ指定できます。

この内部通信路名は、MCF 通信構成定義の mcftpsvr コマンドで登録したものを指定します。

ほかのアプリケーションからこのアプリケーションを起動するとき、ここで指定した内部通信路上の適当な問い合わせ型論理端末を使用して、メッセージが受け渡されます。論理端末定義で、一つ以上、問い合わせ型論理端末を定義しておく必要があります。

アプリケーション起動機能使用時の定義方法を次の表に示します。

表 5-1 アプリケーション起動機能使用時の定義方法

起動元のアプリケーションプログラム	起動先のアプリケーションプログラム		
	noans 型 MHP	ans 型 MHP	cont 型 MHP
noans 型 MHP	LNAME	×	×
ans 型 MHP	LNAME	CNAME	×
cont 型 MHP	LNAME	×	CNAME
SPP	LNAME	×	×

(凡例)

LNAME：起動先の mcfaalcap 定義コマンドの lname オペランドに、内部通信路上の send 型論理端末名称を指定します。

CNAME：起動先の mcfaalcap 定義コマンドの cname オペランドに、request 型論理端末が一つ以上定義されている内部通信路名を指定します。

×：アプリケーション起動機能を使用できないため、指定できません。

tempsize=継続問い合わせ応答用一時記憶データ格納用領域サイズ ～〈符号なし整数〉((1～32000))
《4096》(単位：バイト)

継続問い合わせ応答で使用する一時記憶データ格納用領域サイズを指定します。ただし、type オペランドで cont を指定した場合だけ有効です。

指定値に応じて動的共用メモリの容量を見積もる必要があります。動的共用メモリの見積もりは、「[付録 B.2 MCF サービス用の共用メモリの見積もり式](#)」を参照してください。

trnmode=trn|nontrn ～《trn》

アプリケーションのトランザクション属性を指定します。

trn

MHP の処理をトランザクションとして管理します。

trn を指定した場合、該当 MHP のユーザサービス定義の atomic_update オペランドに Y を指定してください。

nontrn

MHP の処理をトランザクションとして管理しません。

送受信にはメモリキューが使用されます。

errevt=yes|no ～《no》

通信イベントの受信メッセージを破棄した場合、および通信イベント処理用 MHP が異常終了した場合の、エラーイベント通知（通信イベント障害時のエラーイベント通知機能の使用）の有無を指定します。

yes

エラーイベントを通知します。

no

エラーイベントを通知しません。

このオペランドは、name オペランドで通信イベントを指定した場合に有効になります。

name オペランドにエラーイベント (ERREVT1, ERREVT2, ERREVT3, ERREVT4, ERREVT5) を指定した場合、このオペランドの指定は無効になります。また、-n オプションの kind オペランドを省略または user を指定した場合は指定できません。

●-N

(オペランド)

modelname=モデルアプリケーション名|モデル MCF イベント名 ～〈1～8 文字の識別子〉

このアプリケーション属性定義で使用する定義情報を持つ（モデルとする）アプリケーション属性定義のアプリケーション名、または MCF イベント名を指定します。ただし、指定したアプリケーション名、または MCF イベント名に対する、アプリケーション属性定義が先に定義されていなければなりません。

このオペランドを指定したときは、-n オプションの name オペランド以外のオプション、またはオペランドを省略できます。

-n オプションの name オペランド以外のオプション、またはオペランドを省略した場合、modelname オペランドに指定したアプリケーション名、または MCF イベント名の定義指定値がすべて流用されます。

アプリケーション属性定義に-N オプション以外のオプション、またはオペランドを指定した場合は、指定したオプション、またはオペランドの指定値が適用されます。

このオペランドは 2 回以上指定できません。また、モデルとしたアプリケーション名、または MCF イベント名と、-N オプション以外に指定したオペランドの組み合わせによっては、相関チェックによってエラーになる場合があります。そのため、次に示すオペランドには、モデル定義と異なる指定をしないでください。

- -n オプションの type オペランド
- -g オプションの quekind オペランド

●-g

(オペランド)

servgrpname=サービスグループ名 ～〈1～31 文字の識別子〉

アプリケーション名に対応するサービスグループ名を指定します。

サービスグループ名は、ユーザサービス定義に指定してあるサービスグループ名を指定します。

指定したサービスグループの数に応じて静的共用メモリ、およびステータスファイルの容量を見積もる必要があります。静的共用メモリの見積もりは、「付録 B.2 MCF サービス用の共用メモリの見積もり式」を参照してください。ステータスファイルの見積もりは、マニュアル「OpenTP1 運用と操作」の「ステータスファイルのサイズの見積もり式」の説明を参照してください。

quegrpid=キューグループ ID ～〈1～8 文字の識別子〉

入力メッセージのキューグループ ID を指定します。MCF マネージャ定義の mcfmqgid コマンドで指定するキューグループ ID (キュー種別は itq) のどれかを指定します。

このオペランドは、quekind オペランドで disk を指定した場合だけ指定します。

注

一つのサービスグループに対して、異なるキューグループを定義することはできません。異なるキューグループを定義した場合、KFCA11008-W メッセージが出力されます。一つのサービスグループに対して、異なるキューグループを定義しても、MCF 開始時にすでに定義されたキューグループで登録し直します。

quekind=memory|disk ～《memory》

入力メッセージの割り当て先 (メモリキューまたはディスクキュー) を指定します。

memory

メモリキューに割り当てます。

disk

ディスクキューに割り当てます。

disk を指定した場合、必ず quegrpid オペランドを指定してください。

mcfuevt コマンドで直接 MHP を起動する場合に、disk を指定すると、リラン後に UCMDEVT が再スケジュールされることがあります。

-n オプションの trnmode オペランドで nontrn を指定した場合、disk を指定できません。

srvghold=m|s ～《m》

アプリケーションが異常終了した場合のサービスグループの処置を指定します。

m

サービスグループを閉塞しません。

s

サービスグループを閉塞します。スケジュールを禁止します。

次の場合には、この定義の指定値に関係なく、サービスグループを閉塞しません。

- dc_mcf_rollback 関数に DCMCFRTRY を指定 (CBLDCMCF('ROLLBACK')に RTRY を指定) してロールバックした場合
- アプリケーション属性定義 (mcfaalcap) または UAP 共通定義 (mcfmuap) の reschedulecnt オペランドの指定値が 1 以上の場合に、MHP が異常終了したとき

アプリケーションに対応するサービス開始前にユーザサービスが異常終了した場合、この定義の指定値に関係なく、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 hold オペランドおよび term_watch_time オペランドの指定値によって、サービスグループを閉塞するかどうかが決まります。

recvmsg=e|r ～《e》

異常終了した UAP の受信メッセージの扱いを指定します。

このオペランドは、`srvghold` オペランドで `s` を指定したサービスグループが閉塞されたときに有効となります。

e

処理途中のメッセージをエラーイベントに切り替えます。

r

自動閉塞時に処理途中のメッセージを、スケジュールキューの先頭に再登録し、スケジュール待ち状態にします。ただし、`-g` オプションの `quekind` オペランドで `disk` を指定したときだけ有効です。`memory` を指定した場合は、閉塞時に発生する `ERREVT2` としてスケジュールします。

なお、このオペランドはスケジュールの順序性を保証するものではありません。

●-v

(オペランド)

servname=サービス名 ～〈1～31 文字の識別子〉

アプリケーション名に対応するサービス名を指定します。

サービス名は、ユーザサービス定義に指定してあるサービス名を指定します。

指定したサービスの数に応じて静的共用メモリの容量を見積もる必要があります。静的共用メモリの見積もりは、「[付録 B.2 MCF サービス用の共用メモリの見積もり式](#)」を参照してください。

servhold=m|a|s ～〈m〉

アプリケーションが異常終了した場合のサービスの処置を指定します。

m

サービスを閉塞しません。

a

サービスを閉塞します。入力およびスケジュールを禁止します。

s

サービスを閉塞します。スケジュールを禁止します。

次の場合には、この定義の指定値に関係なく、サービスを閉塞しません。

- `dc_mcf_rollback` 関数に `DCMCFRTRY` を指定 (`CBLDCMCF('ROLLBACK')` に `RTRY` を指定) してロールバックした場合
- アプリケーション属性定義 (`mcfaalcap`) または UAP 共通定義 (`mcfmuap`) の `reschedulecnt` オペランドの指定値が 1 以上の場合に、MHP が異常終了したとき
- アプリケーションに対応するサービス開始前にユーザサービスが異常終了した場合

ntmetim=非トランザクション MHP 限界経過時間 ～〈符号なし整数〉((0～65535)) (単位：秒)

非トランザクション MHP の処理のループ、およびハングアップを監視するために、非トランザクション MHP の処理に掛かる限界経過時間を指定します。

0 を指定した場合は、時間監視をしません。

指定時間を超えても非トランザクション MHP の処理が終了しない場合、非トランザクション MHP を異常終了します。

指定を省略した場合は、UAP 共通定義 (mcfmuap) の -u オプションに指定した ntmetim オペランドの値を仮定します。

このオペランドは、-n オプションの trnmode オペランドで nontrn を指定したときに有効となります。

●-d

(オペランド)

holdlimit=アプリケーション異常終了限界回数 ~ 〈符号なし整数〉((1~65535)) 《1》(単位: 回)

アプリケーションの異常終了限界回数を指定します。ここで指定した回数分の異常終了が起こると、そのアプリケーション、サービス、またはサービスグループを閉塞します。ただし、-n オプションの aplihold オペランドまたは -v オプションの servhold オペランドで、a または s を指定した場合、および -g オプションの srvghold オペランドで s を指定した場合だけ有効です。

-d オプションに reschedulecnt オペランドを指定している場合、reschedulecnt オペランドの指定値を超えるまでの異常終了回数は、ここで指定した回数に含まれません。また、dc_mcf_rollback 関数に DCMCFRTRY を指定 (CBLDCMCF('ROLLBACK')に RTRY を指定) して異常終了した場合も、アプリケーション異常終了限界回数にカウントしません。

holdlmtyp=sum|cont ~ 《cont》

holdlimit オペランドを指定した場合の**アプリケーション異常終了回数**のカウント方法を指定します。

sum

アプリケーションが異常終了した累積回数をカウントします。

次に示すとき、カウントリセットします※。

- mcfaactap コマンドを実行したとき
- mcfacldcap コマンドを実行したとき
- OpenTP1 の終了/開始時

cont

アプリケーションが連続して異常終了した場合にカウントします。

次に示すとき、カウントリセットします※。

- mcfaactap コマンドを実行したとき
- mcfacldcap コマンドを実行したとき
- MHP サービス関数が正常終了したとき
- OpenTP1 の終了/開始時

注※

dcsvstop コマンド、dcsvstart コマンド、mcftactsg コマンド、または mcftactsv コマンドではカウント値はリセットされません。

reschedulecnt=アプリケーション異常終了時の再スケジュール回数 ～〈符号なし整数〉((0～3)) (単位: 回)

MHP が異常終了した場合の、再スケジュール回数を指定します。

このオペランドに 1～3 の値を指定すると、受信したメッセージを該当する入力キューの最後に格納し、異常終了した MHP を自動的にスケジュールし直します。

0 を指定した場合は、再スケジュールしません。

指定を省略した場合は、UAP 共通定義 (mcfmuap) の -r オプションに指定した reschedulecnt オペランドの値を仮定します。

指定した回数を超過して MHP が異常終了した場合、再スケジュールしません。この場合、KFCA11163-E メッセージを出力し、エラーイベントを通知します。

なお、異常終了した MHP を再スケジュールするには、アプリケーション起動プロセスが必要です。アプリケーション起動プロセス用の MCF 通信構成定義を作成してください。

また、指定した回数を超過するまでは、-d オプションの holdlimit オペランドで指定したアプリケーション異常終了回数をカウントしません。このため、アプリケーション、サービス、およびサービスグループは閉塞されません。

rescheduleint=アプリケーション異常終了時の再スケジュール間隔 ～〈符号なし整数〉((0～3)) (単位: 秒)

異常終了した MHP を自動的に再スケジュールするまでの所要時間※に加算する時間を、秒単位で指定します。

指定を省略した場合は、UAP 共通定義 (mcfmuap) の -r オプションに指定した rescheduleint オペランドの値を仮定します。

この指定は、reschedulecnt オペランドに 1～3 の値を指定した場合に有効となります。

注※

異常終了したプロセス以外に空きプロセスが存在する場合、間隔を空けないで再スケジュールします。

異常終了したプロセス以外に空きプロセスが存在しない、かつ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 hold オペランドに Y を指定した場合、再スケジュールするまでに最大 10 秒掛かります。

異常終了したプロセス以外に空きプロセスが存在しない、かつ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 hold オペランドに N を指定した場合、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 term_watch_time オペランドの指定時間内で、異常終了した MHP の 3 回目の再スケジュールに最大 10 秒掛かります。2 回目までは間隔を空けないで再スケジュールします。

reschedulelog=yes | no

異常終了した MHP を自動的に再スケジュールするときに、ログメッセージを出力するかどうかを指定します。

指定を省略した場合は、UAP 共通定義 (mcfmuap) の -r オプションに指定した reschedulelog オペランドの値を仮定します。

この指定は、reschedulecnt オペランドに 1～3 の値を指定した場合に有効となります。

yes

KFCA11810-I メッセージを出力します。

no

KFCA11810-I メッセージを出力しません。

●-j

(オペランド)

ij=yes|no ~ 《no》

他システムから受信したメッセージを入力キューに登録する場合に、履歴情報 (IJ) を取得するかどうかを指定します。

-n オプションの kind オペランドで mcf を指定した場合、ij オペランドの指定は無効になります。

入力キューを経由しない同期送受信関数 (dc_mcf_sendrecv 関数) による受信メッセージは、履歴情報の取得対象外になります。

yes

取得します。

no

取得しません。

oj=yes|no ~ 《no》

UAP からのメッセージ送信要求時に、履歴情報 (OJ) を取得するかどうかを指定します。

-n オプションの kind オペランドで mcf を指定した場合、oj オペランドの指定は無効になります。

出力キューを経由しない同期送受信関数 (dc_mcf_sendrecv 関数) による送信メッセージは、履歴情報の取得対象外になります。

yes

取得します。

no

取得しません。

gj=yes|no ~ 《no》

UAP からのメッセージ受信要求時に、履歴情報 (GJ) を取得するかどうかを指定します。

-n オプションの kind オペランドで mcf を指定した場合、gj オペランドの指定は無効になります。

yes

取得します。

no

取得しません。

●-e

(オペランド)

evtlogout=yes|no ～《no》

ERREVT1～ERREVT4 を起動するとき、ログメッセージを出力するかどうかを指定します。

このオペランドは、-n オプションの kind オペランドで mcf を指定し、-n オプションの name オペランドで ERREVT1～ERREVT4 を指定した場合だけ有効です。

yes

ログメッセージを出力します。

ログメッセージとして、KFCA11194-I メッセージとともに、次の内容が出力されます。

- ・ MCF 識別子
- ・ エラーイベント名
- ・ 理由コード
- ・ エラーイベント起動の原因となったアプリケーション名

KFCA11194-I メッセージの詳細については、マニュアル「OpenTP1 メッセージ」を参照してください。

なお、trnmode オペランドに trn を指定した場合は、ユーザサービス定義の trn_rollback_information_put オペランドでロールバック要因のログを取得するように指定することをお勧めします。ERREVT3 の発生要因を調査する際に必要になることがあります。

no

ログメッセージを出力しません。

MCF 性能検証用トレース定義

形式

set 形式

```
[set prf_file_size=MCF性能検証用トレース情報ファイルのサイズ]  
[set prf_file_count=MCF性能検証用トレース情報ファイルの世代数]
```

機能

MCF 性能検証用トレース情報を取得する場合の実行環境を定義します。

説明

set 形式のオペランド

●prf_file_size=MCF 性能検証用トレース情報ファイルのサイズ

～<符合なし整数>((1024～1048576)) 《1024》 (単位：キロバイト)

MCF 性能検証用トレース情報ファイルのサイズを指定します。

トレース情報は、通常ファイルに出力します。このファイルは MCF 性能検証用トレース定義の prf_file_count オペランドの指定値分の世代を用意し、すべての世代を使用したらオーバーラップして上書きします。このため、ある程度時間が経つと古いトレース情報がなくなります。このオペランドの指定値を大きくすることで、ファイルを上書きする時間を延長できます。

トレースファイルの 1 ファイルサイズの算出式を、次に示します (単位：バイト)。

1 ファイルサイズ=128 + (1 トランザクションに必要なトレースデータ長※×実行トランザクション数)

注※

1 トランザクションに必要なトレースデータ長は、マニュアル「OpenTP1 運用と操作」を参照して計算してください。

●prf_file_count=MCF 性能検証用トレース情報ファイルの世代数

～<符合なし整数>((3～256)) 《3》

MCF 性能検証用トレース情報ファイルの世代数を指定します。

値を小さくすると、新しい世代数の範囲で制御するため、範囲外のトレースファイルが使用されないで残ることになります。その場合は、必要に応じて範囲外のファイルを削除してください。

注意事項

この定義の使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できませんので、ご了承ください。

システムサービス情報定義

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

システムサービス情報定義では、MCF 通信サービスおよびアプリケーション起動サービスを起動するための環境を定義します。ユーザが MCF サービスを作成するときに定義する必要があります。

形式

set 形式

```
set module="MCF実行形式プログラム名"  
[set mcf_prf_trace=Y|N]
```

機能

プロセスサービスが MCF 通信サービスを起動するための環境を定義します。

各 MCF 通信サービスに対して一つ、システムサービス情報定義を作成できます。また、複数の MCF 通信サービスで一つのシステムサービス情報定義を共用することもできます。

説明

set 形式のオペランド

●module="MCF 実行形式プログラム名"

～ 〈1～8 文字の識別子〉

MCF 通信サービスを起動するための実行形式プログラム名を指定します。

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

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

MCF 実行形式プログラム名には、先頭 4 文字が'mcfu'で始まる最大 8 文字の名称を指定します。

●mcf_prf_trace=Y|N

～ 〈Y〉

MCF 通信サービスごとに、MCF 性能検証用トレース情報を取得するかどうかを指定します。このオペランドの指定値を有効にするには、システムサービス共通情報定義の mcf_prf_trace_level オペランドに 00000001 を指定してください。

Y

MCF 性能検証用トレース情報を取得します。

N

MCF 性能検証用トレース情報を取得しません。

MCF 通信サービスでの MCF 性能検証用トレース情報取得有無とオペランドの指定値の関係を、次の表に示します。

表 5-2 MCF 通信サービスでの MCF 性能検証用トレース情報の取得有無とオペランドの指定値の関係

システムサービス共通情報定義 mcf_prf_trace_level オペランドの指定値	システムサービス情報定義 mcf_prf_trace オペランドの指定値	
	Y	N
00000000	取得しない	取得しない
00000001	取得する	取得しない

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

システムサービス共通情報定義

システム構成によっては、システムサービス共通情報定義を指定する必要があります。システムサービス共通情報定義では、複数の MCF 通信サービスに共通する情報を定義します。この定義ファイルは、標準値を定義した状態で製品に含まれています。必要に応じて、テキストエディタを使用して定義値を変更してください。

形式

set 形式

```
set max_socket_descriptors=ソケット用ファイル記述子の最大数
set max_open_fds=MCF通信プロセスでアクセスするファイルの最大数
[set thdlock_sleep_time=スレッド間で排他が競合した場合のスレッドの待ち時間]
[set mcf_prf_trace_level=MCF性能検証用トレース情報の取得レベル]
[set mcf_start_watch_interval=MCF開始時プロセス間監視時間間隔]
[set thread_stack_size=OpenTP1内部で使用するスレッドスタック領域（内部通信用）のサイズ]
[set mcf_stack_size=OpenTP1内部で使用するスレッドスタック領域（メッセージ制御用）のサイズ]
```

putenv 形式

```
[putenv DCMCFQUEBAK MCF構成変更準備停止時のバックアップファイル名]
```

機能

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

説明

set 形式のオペランド

●max_socket_descriptors=ソケット用ファイル記述子の最大数

～ 〈符号なし整数〉 ((64～3596))

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

OpenTP1 制御下のプロセスでは、システムサーバやユーザサーバとの間で、ソケットを使用した TCP/IP 通信でプロセス間の情報交換をしています。そのため、同時に稼働する UAP プロセスの数などによって、ソケット用のファイル記述子の最大数を変更する必要があります。

各 MCF 通信プロセスまたはアプリケーション起動プロセスが使用するソケット用ファイル記述子の最大数を求める計算式を次に示します。

↑（このMCF通信プロセスに対してメッセージ送信要求を行うUAPプロセス数※¹
+システムサービスプロセス数※²+このMCF通信プロセスまたはアプリケーション起動プロセスに対し
て同時に処理要求を行う運用コマンド数）／0.8↑

（凡例）

↑↑：小数点以下を切り上げます。

注※1

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

注※2

システムサービスプロセス数とは、自 OpenTP1 内のシステムサービスプロセス数です。自 OpenTP1 内のシステムサービスプロセスは、rpcstat コマンドで表示されるサーバ名をカウントすることで求められます。rpcstat コマンドで表示されるサーバ名のうち、マニュアル「OpenTP1 解説」の OpenTP1 のプロセス構造に記載されているシステムサービスプロセスをカウントしてください。

自 OpenTP1 内の各 MCF 通信プロセスおよびアプリケーション起動プロセスごとに計算し、その結果の中で最大値が 64 より大きい場合は、その値を指定します。64 以下の場合は、64 を指定します。

このオペランドの指定値が小さいと、OpenTP1 制御下の他プロセスとのコネクションが設定できなくなるため、プロセスが KFCA00307-E メッセージを出力して異常終了します。

マッピングサービス、およびオンラインコマンドサービスでは、このオペランドの指定に関係なくソケット用ファイル記述子の最大数は 64 です。

●max_open_fds=MCF 通信プロセスでアクセスするファイルの最大数

～〈符号なし整数〉((500～4032))《500》

各 MCF 通信プロセスでアクセスするファイル数の中の最大値を指定します。

MCF 通信プロセスが行うメッセージの送受信にもファイル記述子が使われます。この数が不足すると、コネクションの確立ができないなどの障害が発生するため、事前に必要となるファイル記述子の数を設定しておく必要があります。

各 MCF 通信プロセスが使用するファイル記述子の最大数を求める計算式を次に示します。

（プロトコル制御で使用するファイル記述子数※¹）
+MCFメイン関数でユーザが使用するファイル記述子数
+30※²

注※1

使用するプロトコル制御によって値が異なります。マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

注※2

MCF 通信プロセスが扱う定義ファイルなどの数の最大値です。

自 OpenTP1 内の MCF 通信プロセスごとに計算し、その結果の中で最大値が 500 より大きい場合は、その値を指定します。500 以下の場合は、500 を指定します。指定値を超えてファイルのアクセスが発生した場合、その超過分はソケット用ファイル記述子使用数として扱われます。この場合、「max_socket_descriptors オペランドの指定値-max_open_fds オペランドの指定値の超過分」が、実際のソケット用ファイル記述子の最大数になりますので、ご注意ください。

max_socket_descriptors オペランドと max_open_fds オペランドには次の条件を満たす値を指定してください。

(「max_socket_descriptorsオペランドの指定値」 + 「max_open_fdsオペランドの指定値」) ≤4096

TP1/NET/TCP/IP の MCF 通信プロセス以外のプロセスで、max_socket_descriptors オペランドと max_open_fds オペランドの和が 1 プロセスで利用できるファイル記述子の最大数を超えている場合、各プロセスで利用できるファイル記述子数は、4096 よりも小さい値に強制的に補正されます。

1 プロセスで利用できるファイル記述子の最大数を次の表に示します。

プロセス	OS	1 プロセスで利用できるファイル記述子の最大数
TP1/NET/TCP/IP の MCF 通信プロセス	すべて	4096
上記以外	AIX, HP-UX, Windows	2048
	Linux, Solaris	1024

max_socket_descriptors オペランドと max_open_fds オペランドの和が 1 プロセス当たりでオープンできるファイル数の物理限界値（ハードリミット）を超えていたとき、MCF の開始を中断します。

●thdlock_sleep_time=スレッド間で排他が競合した場合のスレッドの待ち時間

～〈符号なし整数〉((1～32767)) (単位：ミリ秒)

MCF 通信プロセスが内部で用いる排他制御関数が、スレッド間で排他の競合が発生したため排他を獲得できなかった場合のスレッドの待ち時間を、ミリ秒単位で指定します。

指定を省略した場合は、システム共通定義の thdlock_sleep_time オペランドの値を仮定します。

このオペランドの詳細については、システム共通定義の thdlock_sleep_time オペランドの説明を参照してください。

●mcf_prf_trace_level=MCF 性能検証用トレース情報の取得レベル

～((00000000～00000001))《00000001》

MCF 性能検証用トレース情報の取得レベルを指定します。MCF 性能検証用トレースを取得する場合は、システム共通定義の prf_trace オペランドに Y を指定するか、または省略してください。

00000000

MCF 性能検証用トレース情報を取得しません。

00000001

MCF 性能検証用トレース情報（イベント ID：0xa000～0xa0ff）を取得します。イベント ID の詳細については、マニュアル「OpenTP1 運用と操作」を参照してください。また、プロトコル固有の出力情報や取得タイミングについては、マニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照してください。

オペランドの指定に誤りがある場合は、OpenTP1 開始処理中に OpenTP1 が異常終了します。

このオペランドの使用は、TP1/Extension 1 をインストールしていることが前提です。TP1/Extension 1 をインストールしていない場合の動作は保証できません。

●mcf_start_watch_interval=MCF 開始時プロセス間監視時間間隔

～〈符号なし整数〉((10～3000))《3000》(単位：ミリ秒)

OpenTP1 の正常開始または再開で、MCF のプロセス間で互いの状態を監視する際の時間間隔を指定します。

注意事項

- このオペランドの指定値を小さくすると、MCF の各プロセスの状態の変化を早期に検出できるため、OpenTP1 の正常開始または再開の所要時間を短縮できることがあります。
- このオペランドの指定値が小さ過ぎると、OpenTP1 の正常開始または再開時に CPU 使用率が高騰することがあります。その場合はこのオペランドの指定値を大きくしてください。

●thread_stack_size=OpenTP1 内部で使用するスレッドスタック領域（内部通信用）のサイズ

～〈符号なし整数〉((49152～524288))《49152※》(単位：バイト)

注※

AIX 版および HP-UX (IPF) 版の uCosminexus TP1/Message Control の場合、デフォルト値は 65536 となります。

HP-UX (IPF) 版の uCosminexus TP1/Message Control(64)の場合、デフォルト値は 65536 となります。

AIX 版の uCosminexus TP1/Message Control(64)の場合、デフォルト値は 131072 となります。

OpenTP1 の内部で使用するスレッドスタック領域（内部通信用）のサイズを指定します。

通常、このオペランドは省略します。保守員から依頼があった場合に変更してください。

●mcf_stack_size=OpenTP1 内部で使用するスレッドスタック領域（メッセージ制御用）のサイズ

～〈符号なし整数〉((49152～524288))《49152※》(単位：バイト)

注※

AIX 版および HP-UX (IPF) 版の uCosminexus TP1/Message Control の場合、デフォルト値は 65536 となります。

HP-UX (IPF) 版の uCosminexus TP1/Message Control(64)の場合、デフォルト値は 65536 となります。

AIX 版の uCosminexus TP1/Message Control(64)の場合、デフォルト値は 131072 となります。

OpenTP1 の内部で使用するスレッドスタック領域 (メッセージ制御用) のサイズを指定します。

通常、このオペランドは省略します。保守員から依頼があった場合に変更してください。

putenv 形式のオペランド

●DCMCFQUEBAK MCF 構成変更準備停止時のバックアップファイル名

～ 〈パス名〉《\$DCDIR/spool/mcf/mcfquebak》

MCF 構成変更準備停止による終了中に作成するバックアップファイルの、絶対パス名を 255 バイト以内で指定します。

このオペランドは、再開始時に変更できます。MCF 構成変更準備停止を実行する前にこのオペランドを変更しても、MCF 構成変更準備停止時のバックアップファイルの格納先は変わりません。

このオペランドに \$DCDIR/spool/mcf/mcfquebak 以外の値を指定する場合、\$DCDIR/spool 配下を指定しないでください。指定した場合、dccspool コマンドによって、MCF 構成変更準備停止時のバックアップファイルが削除されることがあります。

6

マルチ OpenTP1 システムの定義

マルチ OpenTP1 システムの定義について説明します。

6.1 定義のしかた

一つのマシン上で複数の OpenTP1 システムが、互いに独立して動作するシステムをマルチ OpenTP1 システムといいます。

1. マルチ OpenTP1 システムでは、異なる OpenTP1 システムごとに異なるパーティション、または異なるファイルを OpenTP1 ファイルシステム用に割り当てます。
2. 定義ファイル格納ディレクトリを、OpenTP1 システムごとに異なったディレクトリとして作成します。
3. それぞれのディレクトリ下に各 OpenTP1 システムの定義ファイルを作成します。

ただし、OpenTP1 システム間で、定義内容を重複して指定できません。

OpenTP1 システムごとに異なる必要がある定義内容を次の表に示します。

ディレクトリなどの詳細は、マニュアル「OpenTP1 運用と操作」を参照してください。

表 6-1 OpenTP1 システムごとに異なる必要がある定義内容

項番	定義ファイル名	定義名	内容
1	システム共通定義	name_port	ネームサービスのウェルノウンポートの番号
		system_id	OpenTP1 識別子
		all_node	OpenTP1 システムのすべてのノード名およびネームサービスのウェルノウンポート番号※
		prc_port	マルチノード連携制御機能で使うポート番号
2	ステータスサービス定義	sts_file_name	ステータスファイルの物理ファイルのパス名
3	ジャーナル関係定義 • システム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 • アーカイブジャーナルサービス定義	jnladdpf	ジャーナル関係ファイルの物理ファイルのパス名
4	DAM サービス定義	damfile	DAM ファイルの物理ファイルのパス名
5	TAM サービス定義	tamtable	TAM テーブルの物理ファイルのパス名
6	メッセージキューサービス定義	quegrp	キューファイルの物理ファイルのパス名

注※

ノード名に、自ホストを示すホスト名（または IP アドレス）を指定する場合に、ポート番号が重複しないようにしてください。

マルチ OpenTP1 システム環境で、各 OpenTP1 間でリモートプロシジャコールを行う場合には、各 OpenTP1 ノードのシステム共通定義で、all_node オペランドに自ノード以外の OpenTP1 ノードのホスト名を定義した上で、rpc_multi_tp1_in_same_host オペランドに Y を指定してください。

7

OpenTP1 システムの定義の変更

OpenTP1 システムの定義の変更方法について説明します。

7.1 定義の変更手順

定義内容を変更する場合は、OpenTP1 システムを正常終了させてから行ってください。定義内容の変更後は、OpenTP1 の正常開始から始めます。

ただし、ユーザサービス定義については、OpenTP1 システムを正常終了させなくても、ユーザサーバを停止することで定義内容を変更できます。ユーザサービスデフォルト定義については、OpenTP1 システムを正常終了させてから定義内容を変更してください。

また、システム共通定義を変更する場合は、注意が必要です。次に示す手順で定義内容を変更してください。

1. OpenTP1 を正常終了させます。

2. 定義内容を変更します。

3. 次に示すどれかの方法で、システム共通定義の変更を反映します。※

- dcreset コマンドを入力します。
- dcsetup -d コマンドを入力してから、再度 dcsetup コマンドを入力して、OpenTP1 を OS に登録します。
- マシンをリブートします。

注※

Windows 版 OpenTP1 では、この手順は不要です。

各コマンドの使用方法については、マニュアル「OpenTP1 運用と操作」を参照してください。

4. OpenTP1 を正常開始させます。

定義内容を変更する場合の注意事項を次に示します。

- システム環境定義の mode_conf オペランドに AUTO が指定されている場合、dcreset コマンドを実行すると OpenTP1 が自動開始します。
- OpenTP1 システムが再開始する場合でも、変更できるシステムサービス定義があります。システムの再開始時に変更できる定義については、「[7.2 再開始時に変更できる定義](#)」を参照してください。
- 定義内容は、OpenTP1 システムへの影響について十分検討してから変更してください。
- ネットワークコミュニケーション定義の定義内容を変更する場合には、ユティリティを使用して定義オブジェクトファイルを再生成する必要があります。定義オブジェクトファイルの生成方法については、「[4.3 定義オブジェクトファイルの生成](#)」を参照してください。

7.2 再開始時に変更できる定義

OpenTP1 システム再開始時に変更できるシステムサービス定義の一覧を次の表に示します。そのほかの定義の指定値は変更できません。

表 7-1 再開始時に変更できるシステムサービス定義の一覧

項番	定義名	オペランド名または定義コマンド名	変更方法
1	システム環境定義	mode_conf	○
		static_shmpool_size	○※1
		dynamic_shmpool_size	○※1
		shmpool_attribute	○
		user_command	○
		dcstart_wakeup_retry_count	○（dcstart 実行前の場合）
		dcstart_wakeup_retry_interval	○（dcstart 実行前の場合）
		putenv DCCONFPATH	○
		putenv DCUAPCONFPATH	○
		dcputenv DCCONFPATH	○
		dcputenv DCUAPCONFPATH	○
2	システム共通定義	all_node	■
		nam_prf_trace_level	○
		fil_prf_trace_option	○
		fil_prf_trace_delay_time	○
		jnl_prf_event_trace_level	○
		name_service_mode	○
		name_manager_node	■
		name_remove_down_node	○
		name_node_add_policy	○
3	ロックサービス定義	lck_prf_trace_level	○
4	ネームサービス定義	name_total_size	●
		name_cache_size	○
		name_nodeid_check_message	○
		name_cache_validity_time	○
		name_start_watch_time	○

項番	定義名	オペランド名または定義コマンド名	変更方法
4	ネームサービス定義	name_start_retry_count	○
		name_start_retry_interval	○
		name_start_error	○
		name_sync_ready_time	○
		namnlfil	○
5	プロセスサービス定義	prc_process_count	○
		prc_recovery_resident	○
		prc_take_over_sypath	○
		prc_take_over_dlpath	○
		prcsvpath	○
6	スケジュールサービス定義	scd_hold_recovery_count	○（ただし、0 からの変更、または 0 への変更の場合だけです）
		scd_hold_recovery	○
7	ステータスサービス定義	sts_initial_error_switch	○
		sts_single_operation_switch	○
		sts_last_active_file	○
		sts_last_active_side	○
8	ジャーナルサービス定義	jnl_watch_time	○
		jnl_message_id_change_level	○
9	システムジャーナルサービス定義	jnl_cdinterval	○
		jnl_rerun_swap	○
		jnl_arc_terminate_check	○
		jnl_auto_unload	○
		jnl_auto_unload_path	○
		jnladdfg	△
		jnladdpf	△
10	チェックポイントダンプサービス定義	cpd_message_id_change_level	○
11	ログサービス定義	log_filesize	○
		log_msg_console	○
		log_msg_allno	○

項番	定義名	オペランド名または定義コマンド名	変更方法
11	ログサービス定義	log_msg_prcid	○
		log_msg_prcno	○
		log_msg_sysid	○
		log_msg_date	○
		log_msg_time	○
		log_msg_hostname	○
		log_msg_pgmid	○
		log_netm_out	○
		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_audit_out	○
		log_audit_path	○
		log_audit_size	○
		log_audit_count	○
		log_audit_message	○
		putenv TZ	○
12	トランザクションサービス定義	trn_rm_open_close_scope	○
		trn_optimum_item	○
		trn_processing_in_rm_error	○
		trn_recovery_list_remove	○
		trn_recovery_list_remove_level	○
		trn_watch_time	○
		trn_rollback_information_put	○
		trn_limit_time	○
		trn_rollback_response_receive	○

項番	定義名	オペランド名または定義コマンド名	変更方法
12	トランザクションサービス定義	trn_partial_recovery_type	○
		max_socket_descriptors	○
		trn_recovery_failmsg_interval	○
		trn_wait_rm_open	○
		trn_retry_interval_rm_open	○
		trn_retry_count_rm_open	○
		thread_stack_size	○
		polling_control_data	○
		groups	○
		trn_xa_commit_error	○
		trn_prf_event_trace_level	○
		trn_prf_event_trace_condition	○
		watch_time	○
		thread_yield_interval	○
		trn_start_recovery_mode	○
		trn_start_recovery_watch_time	○
		trn_start_recovery_interval	○
		trn_extend_function	○
		trnstring -m	○
		trnstring -r	○
		putenv	○※2
		dcputenv	○※2
13	XA リソースサービス定義	xar_eventtrace_level	○
		xar_eventtrace_record	○
		xar_session_time	○
		xar_prf_trace_level	○
		xarfile	○
14	グローバルアーカイブジャーナルサービス定義	jnl_watch_time	○
15	アーカイブジャーナルサービス定義	jnl_rerun_swap	○
		jnladdfg	△

項番	定義名	オペランド名または定義コマンド名	変更方法
15	アーカイブジャーナルサービス 定義	jnladdpf	△
16	ユーザサービスネットワーク 定義	dcsvgdef	○
17	rap リスナーサービス定義	rap_listen_port	○
		rap_parallel_server	○
		rap_watch_time	○
		rap_inquire_time	○
		nice	○
		uap_trace_max	○
		uid	○
		rpc_response_statistics	○
		rpc_trace	○
		rpc_trace_name	○
		rpc_trace_size	○
		trn_expiration_time	○
		trn_expiration_time_suspend	○
		trn_cpu_time	○
		trf_put	○
		trn_statistics_item	○
		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_assign_type	○
		rap_max_client	○
		rap_notify	○
		rap_client_manager_node	○

項番	定義名	オペランド名または定義コマンド名	変更方法
17	rap リスナーサービス定義	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	○
		rap_msg_output_interval	○
		rap_recovery_server	○
		rap_connect_interval	○
		rpc_extend_function	○
		max_socket_descriptors	○
		trn_completion_limit_time	○
		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_sockctl_highwater	○
		ipc_sockctl_watchtime	○
		watch_time	○
18	rap クライアントマネージャサービス定義	rap_client_manager_port	○
		rap_listen_inf	○
		uid	○
		log_audit_out_suppress	○
		log_audit_message	○
		rap_watch_time	○
19	性能検証用トレース定義	prf_file_size	○
		prf_information_level	○
		prf_file_count	○※3

項番	定義名	オペランド名または定義コマンド名	変更方法
19	性能検証用トレース定義	prf_trace_backup	○
20	XAR 性能検証用トレース定義	prf_file_size	○
		prf_information_level	○
		prf_file_count	○※3
21	JNL 性能検証用トレース定義	prf_file_size	○
		prf_file_count	○※3
		prf_trace_backup	○
22	LCK 性能検証用トレース定義	prf_file_size	○
		prf_information_level	○
		prf_file_count	○※3
23	リアルタイム統計情報サービス定義	rts_trcput_interval	○
		rts_service_max	○※4
		rts_item_max	○※4
		rts_log_file	○
		rts_log_file_name	○
		rts_log_file_size	○
		rts_log_file_count	○
		rts_log_file_backup	○
		rts_swap_message	○
		rtspout	○
24	ユーザサービスデフォルト定義	watch_next_chain_time	○
		max_socket_msg	○
		max_socket_msglen	○
		rpc_response_statistics	○
		rpc_service_retry_count	○
		rpc_extend_function	○
		max_socket_descriptors	○
		max_open_fds	○
		watch_time	○
		rpc_destination_mode	○

項番	定義名	オペランド名または定義コマンド名	変更方法
24	ユーザサービスデフォルト定義	rpc_rap_auto_connect	○
		rpc_rap_inquire_time	○
		rpc_request_cancel_for_timedout	○
		service_expiration_time	○
		ipc_sockctl_highwater	○
		ipc_sockctl_watchtime	○
		ipc_conn_interval	○
		ipc_send_interval	○
		ipc_send_count	○
		ipc_header_rcv_time	○
		rpc_send_retry_count	○
		rpc_send_retry_interval	○
		ipc_rcvbuf_size	○
		ipc_sendbuf_size	○
		thread_yield_interval	○
		ipc_backlog_count	○
		rpc_buffer_pool_max	○
		message_buflen	○
		message_store_buflen	○
		trn_expiration_time	○
		trn_limit_time	○
		trn_cpu_time	○
		trn_completion_limit_time	○
		rap_autoconnect_con_error_msg	○
		rap_message_id_change_level	○
		log_audit_out_suppress	○
		log_audit_message	○
		mcf_prf_trace	○
		scdsvcdef	○
25	ユーザサービス定義	watch_next_chain_time	○
		uid	○

項番	定義名	オペランド名または定義コマンド名	変更方法
25	ユーザサービス定義	max_socket_msg	○
		max_socket_msglen	○
		rpc_response_statistics	○
		rpc_service_retry_count	○
		rpc_extend_function	○
		max_socket_descriptors	○
		max_open_fds	○
		watch_time	○
		rpc_destination_mode	○
		rpc_rap_auto_connect	○
		rpc_rap_inquire_time	○
		rpc_request_cancel_for_timeout	○
		service_expiration_tim	○
		ipc_sockctl_highwater	○
		ipc_sockctl_watchtime	○
		ipc_conn_interval	○
		ipc_send_interval	○
		ipc_send_count	○
		ipc_header_rcv_time	○
		rpc_send_retry_count	○
		rpc_send_retry_interval	○
		ipc_rcvbuf_size	○
		ipc_sendbuf_size	○
		thread_yield_interval	○
		ipc_backlog_count	○
		rpc_buffer_pool_max	○
		message_bufen	○
		message_store_bufen	○
		trn_expiration_time	○
		trn_limit_time	○
		trn_cpu_time	○

項番	定義名	オペランド名または定義コマンド名	変更方法
25	ユーザサービス定義	trn_completion_limit_time	○
		rap_autoconnect_con_error_msg	○
		rap_message_id_change_level	○
		log_audit_out_suppress	○
		log_audit_message	○
		mcf_prf_trace	○
		scdsvcdef	○
		putenv	○
		dcputenv	○
26	MCF 性能検証用トレース定義	prf_file_size	○
		prf_file_count	○
27	システムサービス情報定義	mcf_prf_trace	○
28	システムサービス共通情報定義	mcf_prf_trace_level	○
		mcf_start_watch_interval	○
		thread_stack_size	○
		mcf_stack_size	○

(凡例)

○：値を変更できます。

△：追加だけです。変更、削除はできません。

■：値を変更できます。ただし、再開始時に実際にある値を指定してください。

●：値を変更できます。ただし、指定する値はシステムサービスとユーザサーバで登録するサービス情報の数を基に、計算して求めてください。

注※1

メモリサイズを小さくすると、再起動できない場合があります。

注※2

変更できる環境変数名および環境変数値については、リソースマネージャの仕様書を参照してください。

注※3

値を小さくすると、新しい世代数の範囲で制御するため、範囲外のトレースファイルが使用されないで残ることになります。その場合は、必要に応じて範囲外のファイルを削除してください。

注※4

値を大きくすると、リアルタイム統計情報サービスが開始できない場合があります。

7.3 OpenTP1 システムの変更に影響する定義

OpenTP1 システムの変更に伴って見直す必要のある定義および OpenTP1 ファイルについて説明します。

7.3.1 ユーザサーバの追加

ユーザサーバを追加する場合、ユーザサービス定義を新規に追加してください。

ユーザサーバを追加する場合に見直す必要のある定義の一覧、および再見積もりが発生する条件を次の表に示します。

表 7-2 ユーザサーバを追加する場合に見直しが必要な定義の一覧

定義ファイル名	定義	再見積もりが発生する条件
• betranrc • jnl • nam • prc • scd • trn	max_socket_descriptors	無条件に再見積もりが必要
cltsrv	cup_parallel_count	TP1/Client/P または TP1/Client/W から常設コネクション経由の RPC を受信する場合
	parallel_count	TP1/Client/P または TP1/Client/W からトランザクショナル RPC を受信する場合
dam	dam_cache_size	DAM ファイルにアクセスする場合
	dam_cache_size_fix	DAM ファイルにアクセスする場合
	dam_tran_process_count	DAM ファイルにアクセスする場合
env	dynamic_shmpool_size	次に示すどちらかの条件の場合、再見積もりが必要 • DAM ファイルまたは TP1/Message Queue もしくは MCF のキューファイルにアクセスする場合 • MCF 通信プロセスに対してメッセージ送信または受信を行う UAP 数が増える場合
	server_count	無条件に再見積もりが必要
	static_shmpool_size	無条件に再見積もりが必要
lck	lck_limit_fordam	DAM ファイルにアクセスする場合
	lck_limit_formqa	TP1/Message Queue ファイルにアクセスする場合
	lck_limit_fortam	TAM ファイルにアクセスする場合

定義ファイル名	定義	再見積もりが発生する条件
lck	lck_limit_foruser	dc_lck_get 関数を使用する場合
nam	name_cache_size	次に示すどちらかの条件の場合、再見積もりが必要 - parallel_count オペランドの変更を行うユーザサーバが SPP の場合 - all_node オペランドに他ノードを設定している場合は、他ノードの name_cache_size オペランドの再見積もりが必要
	name_total_size	parallel_count オペランドの変更を行うユーザサーバが SPP の場合
prc	prc_process_count	無条件に再見積もりが必要
scd	scd_server_count	SPP または MHP の場合
	scd_hold_recovery_count	追加するユーザサーバについて閉塞状態を引き継ぐ場合
	scdbufgrp	スケジュールバッファグループを利用する場合
	scdmulti	マルチスケジューラ経由の RPC コールを受信する場合
tam	tam_max_trnnum	TAM ファイルにアクセスする場合
tim	tim_watch_count	次に示すどれかの条件の場合、再見積もりが必要 - service_expiration_time オペランドの監視対象 - trn_completion_limit_time オペランドの監視対象 - 非トランザクション MHP 限界経過時間が監視対象 - trn_expiration_time オペランドの監視対象
trn	trn_max_subordinate_count	トランザクション処理を行う場合
	trn_tran_process_count	トランザクション処理を行う場合
- usrrc - ユーザサービス定義	message_store_buflen	追加するユーザサーバが RPC コールをする場合、RPC コール先のユーザサーバの message_store_buflen オペランドの再見積もりが必要
ユーザサービス定義	max_socket_descriptors	追加するユーザサーバが他ノードのユーザサーバと RPC コールで通信する場合、他ノードのユーザサーバの max_socket_descriptors オペランドの再見積もりが必要
\$DCDIR/lib/sysconf/mcf	max_socket_descriptors	MCF 通信プロセスに対してメッセージ送信要求を行う UAP 数、および同時に処理要求を行う運用コマンド数が増える場合

定義ファイル名	定義	再見積もりが発生する条件
MCF マネージャ共通定義	mcfmcomn -p	MHP を追加する場合、再見積もりが必要 増分はシステム環境定義 (env) の static_shmpool_size オペランドに加算
拡張予約定義	mcfmexp -g	拡張予約定義を定義していて新たな MHP の サービスグループを追加する場合
状態引き継ぎ定義	mcfmsts -g	状態を引き継ぐサービスグループが増える場合
	mcfmsts -v	状態を引き継ぐサービスが増える場合
	mcftsts -a	状態を引き継ぐ MHP が増える場合
アプリケーション起動用論理端末 定義	mcftalcle -m	アプリケーション起動を行うユーザサーバを追 加する場合

見直す必要のある OpenTP1 ファイルの一覧、および再見積もりが発生する条件を次の表に示します。

表 7-3 ユーザサーバを追加する場合に見直しが必要な OpenTP1 ファイルの一覧

OpenTP1 ファイル	再見積もりが発生する条件
ステータスファイル	無条件に再見積もりが必要
システムジャーナルファイル	次に示すどちらかの条件の場合、再見積もりが必要 ・ トランザクション処理を行う場合 ・ 統計情報を取得する場合
チェックポイントダンプファイル	トランザクション処理を行う場合
XAR ファイル	トランザクション処理を行い、かつ XA リソースサービス機能を使用 している場合

7.3.2 ユーザサーバの多重度の変更

ユーザサーバの多重度を変更する場合、ユーザサービス定義の parallel_count オペランドの指定の変更が必要です。

この場合に見直す必要のある定義の一覧、および再見積もりが発生する条件を次の表に示します。

表 7-4 ユーザサーバの多重度を変更する場合に見直しが必要な定義の一覧

定義ファイル名	定義	再見積もりが発生する条件
・ betranrc ・ jnl ・ nam ・ prc ・ scd	max_socket_descriptors	無条件に再見積もりが必要

定義ファイル名	定義	再見積もりが発生する条件
• trn	max_socket_descriptors	無条件に再見積もりが必要
cltsrv	cup_parallel_count	クライアントとなるユーザサーバを増やし、TP1/Client/P または TP1/Client/W から常設コネクション経由の RPC を受信する場合
	parallel_count	クライアントとなるユーザサーバを増やし、TP1/Client/P または TP1/Client/W からトランザクショナル RPC を受信する場合
rap リスナーサービス定義	rap_parallel_server	クライアントとなるユーザサーバを増やし、rap クライアントからの要求を代理実行する場合
dam	dam_cache_size	DAM ファイルにアクセスする場合
	dam_cache_size_fix	DAM ファイルにアクセスする場合
	dam_tran_process_count	DAM ファイルにアクセスする場合
env	dynamic_shmpool_size	次に示すどちらかの条件の場合、再見積もりが必要 • DAM ファイルまたは TP1/Message Queue もしくは MCF のキューファイルにアクセスする場合 • MCF 通信プロセスに対してメッセージ送信または受信を行う UAP 数が増える場合
	static_shmpool_size	無条件に再見積もりが必要
lck	lck_limit_fordam	DAM ファイルにアクセスする場合
	lck_limit_formqa	TP1/Message Queue ファイルにアクセスする場合
	lck_limit_fortam	TAM ファイルにアクセスする場合
	lck_limit_foruser	dc_lck_get 関数を使用する場合
prc	prc_process_count	無条件に再見積もりが必要
scd	scd_hold_recovery_count	多重度を変更するユーザサーバについて閉塞状態を引き継ぐ場合
	scdbufgrp	スケジュールバッファグループを利用する場合
	scdmulti	マルチスケジューラ経由の RPC コールを受信する場合
	scdsvcdef -p	scdsvcdef 定義コマンドの-p オプションを指定している場合
tam	tam_max_trnnum	TAM ファイルにアクセスする場合
tim	tim_watch_count	次に示すどれかの条件の場合、再見積もりが必要

定義ファイル名	定義	再見積もりが発生する条件
tim	tim_watch_count	• service_expiration_time オペランドの監視対象 • trn_completion_limit_time オペランドの監視対象 • 非トランザクション MHP 限界経過時間が監視対象 • trn_expiration_time オペランドの監視対象
trn	trn_max_subordinate_count	トランザクション処理を行う場合
	trn_tran_process_count	トランザクション処理を行う場合
• usrrc • ユーザサービス定義	message_store_buflen	追加するユーザサーバが RPC コールをする場合、RPC コール先のユーザサーバの message_store_buflen オペランドの再見積もりが必要
ユーザサービス定義	max_socket_descriptors	追加するユーザサーバが他ノードのユーザサーバと RPC コールで通信する場合、他ノードのユーザサーバの max_socket_descriptors オペランドの再見積もりが必要
\$DCDIR/lib/sysconf/mcf	max_socket_descriptors	MCF 通信プロセスに対してメッセージ送信要求を行う UAP 数、および同時に処理要求を行う運用コマンド数が増える場合
MCF マネージャ共通定義	mcfmcomn -p	出力キューに滞留するメッセージが増える場合、再見積もりが必要 増分はシステム環境定義 (env) の static_shmpool_size オペランドに加算する

見直す必要のある OpenTP1 ファイルの一覧、および再見積もりが発生する条件を次の表に示します。

表 7-5 ユーザサーバの多重度を変更する場合に見直しが必要な OpenTP1 ファイルの一覧

OpenTP1 ファイル	再見積もりが発生する条件
システムジャーナルファイル	次に示すどちらかの条件の場合、再見積もりが必要 • トランザクション処理を行う場合 • 統計情報を取得する場合
チェックポイントダンプファイル	トランザクション処理を行う場合
XAR ファイル	トランザクション処理を行い、かつ XA リソースサービス機能を使用している場合

7.3.3 ノードの追加

ノードを追加する場合、次の作業が必要です。

- システム共通定義の all_node オペランドの変更
ただし、ノード自動追加機能を使用している場合は不要です。
- ドメイン定義ファイルの変更

この場合に見直す必要のある定義の一覧、および再見積もりが発生する条件を次の表に示します。

表 7-6 ノードを追加する場合に見直しが必要な定義の一覧

定義ファイル名	定義	再見積もりが発生する条件
betranrc	all_node_extend_number	無条件に再見積もりが必要
env	static_shmpool_size	無条件に再見積もりが必要
ist	ist_node	IST テーブルにアクセスする場合、見直しが必要
jnl	max_socket_descriptors	グローバルアーカイブジャーナル機能を使用している場合
• nam • scd • tm • ユーザサービス定義	max_socket_descriptors	無条件に再見積もりが必要
nam	name_cache_size	無条件に再見積もりが必要
	name_total_size	無条件に再見積もりが必要

7.3.4 ジャーナルファイルグループの追加

ジャーナルファイルグループを追加する場合、次の作業が必要です。

- システムジャーナルサービス定義ファイルの jnladdfg 定義コマンドの変更
- システムジャーナルサービス定義ファイルの jnladdpf 定義コマンドの変更

この場合に見直す必要のある定義の一覧、および再見積もりが発生する条件を次の表に示します。

表 7-7 ジャーナルファイルグループを追加する場合に見直しが必要な定義の一覧

定義ファイル名	定義	再見積もりが発生する条件
env	static_shmpool_size	無条件に再見積もりが必要

見直す必要のある OpenTP1 ファイルの一覧、および再見積もりが発生する条件を次の表に示します。

表 7-8 ジャーナルファイルグループを追加する場合に見直しが必要な OpenTP1 ファイルの一覧

OpenTP1 ファイル	再見積もりが発生する条件
ステータスファイル	無条件に再見積もりが必要

7.3.5 ホスト名または IP アドレスの変更

ホスト名または IP アドレスを変更する場合に、見直しが必要な定義と、変更手順について説明します。

(1) ホスト名または IP アドレスを変更する場合に見直しが必要な定義

ホスト名または IP アドレスを変更する場合、見直す必要のある定義の一覧と発生する条件を次の表に示します。

注意事項

自ノードおよび他ノードともに変更前のホスト名称、IP アドレスを指定している個所があればすべて見直しが必要です。

表 7-9 ホスト名または IP アドレスを変更する場合に見直しが必要な定義の一覧

定義ファイル名	定義	見直しが必要な条件
betranrc	all_node	ノード自動追加機能を使用していない場合、見直しが必要
	all_node_ex	無条件に見直しが必要
	dcbindht -h	無条件に見直しが必要
	my_host	無条件に見直しが必要
	name_manager_node	ノード自動追加機能を使用している場合、見直しが必要
nodeaddr	dcprcport -h	TP1/Multi を使用している場合、見直しが必要
usrnet	dcsvgdef -h	無条件に見直しが必要
rap クライアントマネージャサービス定義	rap_listen_inf	無条件に見直しが必要
rap リスナーサービス定義	rap_client_manager_node	無条件に見直しが必要
ドメイン定義ファイル	—	name_domain_file_use オペランドに Y を指定している場合、見直しが必要

(凡例)

—：該当しません。

TP1/Message Queue または MCF を使用している場合は、それぞれマニュアル「TP1/Message Queue 使用の手引」またはマニュアル「OpenTP1 プロトコル」の該当するプロトコル編を参照して見直す必要のある定義を確認してください。

(2) ホスト名または IP アドレスの変更手順

ホスト名または IP アドレスは、次の手順で変更してください。

1. OpenTP1 を正常停止します。
2. \$DCDIR/conf, および \$DCCONFPATH 配下について、変更前のホスト名または IP アドレスを検索します。
OS が UNIX の場合は grep コマンド, Windows の場合は findstr コマンドを使用して検索します。
3. 検索の結果、変更前のホスト名または IP アドレスが見つかった場合には、変更します。
4. システム共通定義ファイル (\$DCCONFPATH/betranrc) を変更している場合には、dcreset コマンドを実行します。

8

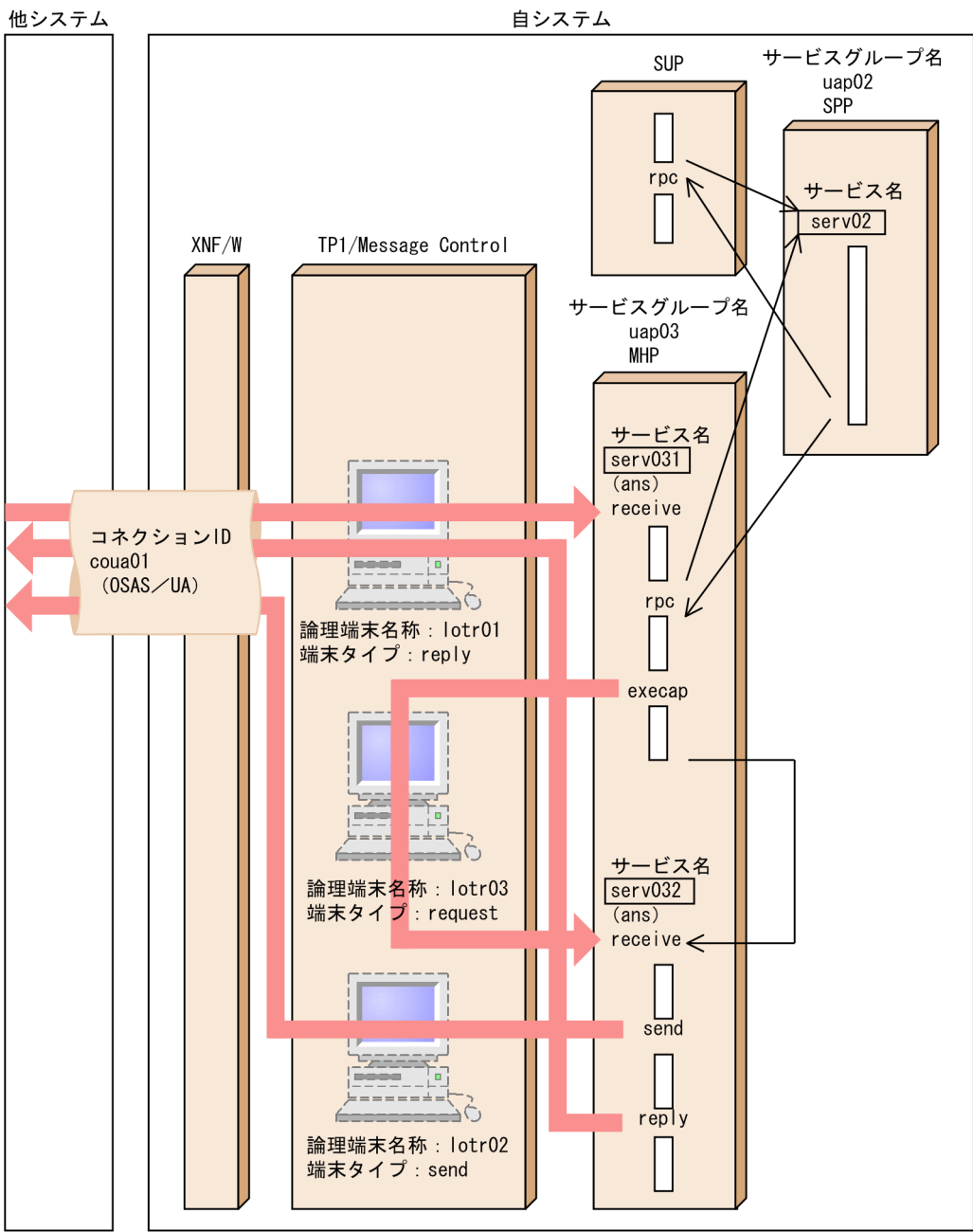
定義例

OpenTP1 システム定義の作成例を示します。

8.1 OpenTP1 システムを構成する定義例

定義例の通信環境と UAP 構成を次の図に示します。

図 8-1 定義例の通信環境と UAP 構成



定義のコーディング例を次に示します。

```
#####
# システム環境定義
# ファイル名:env
#
#####
```

```

set mode_conf          = AUTO    # システムの開始方法
set static_shmpool_size = 6000    # 静的共用メモリの総量
set dynamic_shmpool_size = 500    # 動的共用メモリの最大使用時の総量
set shmpool_attribute   = free    # 共用メモリプールのメモリ固定の要否
putenv DCCONFPATH /qa3d/qa4g10/makai3g # 定義ファイルの格納ディレクトリ
putenv SHELL /bin/sh          # ユーザ環境設定コマンド起動時のシェル

```

システム環境定義の終わり

```

#####
# システムサービス構成定義
# ファイル名:sysconf
#
#####

```

```

set uap_conf = Y          # ユーザサーバを実行するかどうか
set dam_conf = Y          # DAMサービスを使用するかどうか
set que_conf = Y          # メッセージキューサービスを使用するかどうか
set tam_conf = Y          # TAMサービスを使用するかどうか

```

```

#-----システムサービスの起動-----#
dcsvstart -m _mutest1      # 起動するMCFサービスの名称

```

システムサービス構成定義の終わり

```

#####
# ユーザサービス構成定義
# ファイル名:usrconf
#
#####

```

```

#-----ユーザサーバの起動-----#
dcsvstart -u exuap02       # 起動するシステムサービス名

dcsvstart -u exuap03       # 起動するシステムサービス名

```

ユーザサービス構成定義の終わり#####

```

#####
# システム共通定義
# ファイル名:betranrc
#
#####

```

```

set watch_time = 240      # RPCの応答に対する最大応答待ち時間
set name_port   = 10004    # ネームサービスのポート番号
set system_id   = o1       # OpenTP1識別子
set all_node    = ft6101,ft6102 # ネームサービスが存在する全ノード名
set node_id     = n d01    # ノード識別子
putenv LANG ja_JP.SJIS    # LANG設定値

```

システム共通定義の終わり

```

#####
# ロックサービス定義
# ファイル名:lck
#
#####

```

```

set lck_limit_foruser = 64          # ユーザサーバの最大同時排他要求数
set lck_limit_fordam = 64          # DAMサービスの最大同時排他要求数
set lck_limit_fortam = 64          # TAMサービスの最大同時排他要求数
set lck_wait_timeout = 30          # 排他待ち限界経過時間
set lck_deadlock_info = Y          # デッドロック情報を出力するかどうか

### ロックサービス定義の終わり #####

#####
# タイマサービス定義
# ファイル名:tim
#
#####

set tim_watch_count = 128          # 最大時間監視サービス数

### タイマサービス定義の終わり #####

#####
# ネームサービス定義
# ファイル名:nam
#
#####

set name_total_size = 64          # サービス情報領域の大きさ
set name_cache_size = 16          # サービス情報キャッシュ領域の大きさ

### ネームサービス定義の終わり #####

#####
# プロセスサービス定義
# ファイル名:prc
#
#####

set prc_process_count = 100        # 最大同時起動サーバプロセス数

#-----ユーザサーバパスの指定-----#
prcsvpath /BeTRAN/aplib           # ユーザサーバおよびコマンドのパス名

### プロセスサービス定義の終わり #####

#####
# スケジュールサービス定義
# ファイル名:scd
#
#####

set scd_server_count = 32          # 最大ユーザサーバ数

### スケジュールサービス定義の終わり #####

#####
# トランザクションサービス定義
# ファイル名:trn

```

```

#
#####

set trn_tran_process_count = 20 # 同時起動トランザクションブランチ数
set trn_recovery_process_count = 2
                                # 並行回復プロセス数
set trn_expiration_time     = 60 # トランザクションブランチ限界経過時間
set trn_tran_statistics     = N  # 統計情報を取得するかどうか

### トランザクションサービス定義の終わり ####

#####
# ステータスサービス定義
# ファイル名:sts
#
#####

set sts_file_name_1 = "stsfil01","/dev/rdisk/rhd111/stsfil01", ¥
                        "/dev/rdisk/rhd112/stsfil02"
    # 論理ファイル名,A系ステータスファイル名,B系ステータスファイル名
set sts_file_name_2 = "stsfil02","/dev/rdisk/rhd111/stsfil03", ¥
                        "/dev/rdisk/rhd112/stsfil04"
    # 論理ファイル名,A系ステータスファイル名,B系ステータスファイル名
set sts_initial_error_switch = stop # ステータスサービスの障害時の処置

### ステータスサービス定義の終わり #####

#####
# ジャーナルサービス定義
# ファイル名:jnl
#
#####

#-----ジャーナル関係のファイルの指定-----#
jnl dfsv -r sjs -c cdtjl,cdmcf
    # -r システムジャーナルサービス定義のファイル名
    # -c チェックポイントダンプサービス定義のファイル名

### ジャーナルサービス定義の終わり #####

#####
# システムジャーナルサービス定義
# ファイル名:sjs
#
#####

set jnl_max_datasize     = 32000          # 最大レコードデータ長
set jnl_cdinterval       = 1000          # ジャーナルブロック数
set jnl_rerun_swap       = N
    # システム再開始時にジャーナルをスワップするかどうか
set jnl_dual             = Y             # ジャーナルファイルを二重化するかどうか
set jnl_singleoperation = N             # 片系使用できないときスワップするかどうか
set jnl_rerun_reserved_file_open = N
    # 全面回復時に予約ファイルをオープンするかどうか

#-ジャーナル関係のファイルグループの指定(1)--#
jnl addfg -g jnlgrp01 ONL

```



```

# システムジャーナルを構成するファイルグループ名

#---ジャーナル関係の物理ファイルの指定(1)---#
jnladdpf -g jnlgrp01 ¥
          -a /dev/rdisk/rhd111/jnlf011 ¥
          -b /dev/rdisk/rhd112/jnlf012
          # -g 対応するファイルグループ名
          # -a 物理ファイル名
          # -b 物理ファイル名

#-ジャーナル関係のファイルグループの指定(2)-#
jnladdfg -g jnlgrp02 ONL

#--ジャーナル関係の物理ファイルの指定(2)---#
jnladdpf -g jnlgrp02 ¥
          -a /dev/rdisk/rhd111/jnlf021 ¥
          -b /dev/rdisk/rhd112/jnlf022

#-ジャーナル関係のファイルグループの指定(3)-#
jnladdfg -g jnlgrp03 ONL

#--ジャーナル関係の物理ファイルの指定(3)---#
jnladdpf -g jnlgrp03 ¥
          -a /dev/rdisk/rhd111/jnlf031 ¥
          -b /dev/rdisk/rhd112/jnlf032

#-ジャーナル関係のファイルグループの指定(4)-#
jnladdfg -g jnlgrp04 ONL

#--ジャーナル関係の物理ファイルの指定(4)---#
jnladdpf -g jnlgrp04 ¥
          -a /dev/rdisk/rhd111/jnlf041 ¥
          -b /dev/rdisk/rhd112/jnlf042

### システムジャーナルサービス定義の終わり ##

#####
# チェックポイントダンプサービス定義
# ファイル名:cdtjl
#
#####

set jnl_objservername = "_tjl" # 対象となるシステムサービス名
set jnl_max_datasize = 32768 # チェックポイントダンプ取得用バッファ長
set assurance_count = 2 # 有効保証世代数を2とする

#-ジャーナル関係のファイルグループの指定(1)-#
jnladdfg -g cpdgrp01 ONL
          # チェックポイントダンプを構成するファイルグループ名
#--ジャーナル関係の物理ファイルの指定(1)---#
jnladdpf -g cpdgrp01 ¥
          -a /dev/rdisk/rhd111/cpdf01
          # -g 対応するファイルグループ名
          # -a 物理ファイル名

#-ジャーナル関係のファイルグループの指定(2)-#
jnladdfg -g cpdgrp02 ONL

```

```

#--ジャーナル関係の物理ファイルの指定(2)----#
jnladdpf -g cpdgrp02 ¥
        -a /dev/rdisk/rhd111/cpdf02

#-ジャーナル関係のファイルグループの指定(3)-#
jnladdfg -g cpdgrp03 ONL

#--ジャーナル関係の物理ファイルの指定(3)----#
jnladdpf -g cpdgrp03 ¥
        -a /dev/rdisk/rhd111/cpdf03

## チェックポイントダンプサービス定義の終わり

#####
# チェックポイントダンプサービス定義
# ファイル名:cdmcf
#
#####

set jnl_objservername = "_mutest1"      # 対象となるシステムサービス名
set jnl_max_datasize  = 32768          # チェックポイントダンプ取得用バッファ長
set assurance_count   = 2              # 有効保証世代数を2とする

#-ジャーナル関係のファイルグループの指定(1)--#
jnladdfg -g cpdgrp04 ONL
        # チェックポイントダンプを構成するファイルグループ名

#--ジャーナル関係の物理ファイルの指定(1)----#
jnladdpf -g cpdgrp04 ¥
        -a /dev/rdisk/rhd111/cpdf04
                                # -g 対応するファイルグループ名
                                # -a 物理ファイル名

#-ジャーナル関係のファイルグループの指定(2)--#
jnladdfg -g cpdgrp05 ONL

#--ジャーナル関係の物理ファイルの指定(2)----#
jnladdpf -g cpdgrp05 ¥
        -a /dev/rdisk/rhd111/cpdf05

#-ジャーナル関係のファイルグループの指定(3)-#
jnladdfg -g cpdgrp06 ONL

#--ジャーナル関係の物理ファイルの指定(3)----#
jnladdpf -g cpdgrp06 ¥
        -a /dev/rdisk/rhd111/cpdf06

## チェックポイントダンプサービス定義の終わり

#####
# ログサービス定義
# ファイル名:log
#
#####

set log_filesize      = 100      # メッセージログファイルの最大容量
set log_msg_console   = Y        # リアルタイム出力機能を使用するかどうか
set log_msg_allno     = N        # システム内の通番を付加するかどうか

```

```

set log_msg_prcid      = N      # 要求元のプロセスIDを付加するかどうか
set log_msg_prcno      = N      # プロセス内の通番を付加するかどうか
set log_msg_sysid      = Y      # OpenTP1識別子を付加するかどうか
set log_msg_date       = Y      # 出力要求時の日付を付加するかどうか
set log_msg_time       = Y      # 出力要求時の時刻を付加するかどうか
set log_msg_hostname   = Y      # 要求元のホスト名を付加するかどうか
set log_msg_pgmid      = Y      # 要求元プログラムIDを付加するかどうか
set log_netm_out       = N      # メッセージログをNETMに出力するかどうか
set log_netm_allno     = N      # システム内通番を付加するかどうか
set log_netm_prcid     = N      # 要求元のプロセスIDを付加するかどうか
set log_netm_prcno     = N      # プロセス内の通番を付加するかどうか
set log_netm_sysid     = Y      # OpenTP1識別子を付加するかどうか
set log_netm_date      = Y      # 出力要求時の日付を付加するかどうか
set log_netm_time      = Y      # 出力要求時の時刻を付加するかどうか
set log_netm_hostname  = Y      # 要求元のホスト名を付加するかどうか
set log_netm_pgmid     = Y      # 要求元プログラムIDを付加するかどうか
putenv TZ JST-9        # タイムゾーン指定

### ログサービス定義の終わり #####

#####
# マルチノード構成定義
# ファイル名:nodeconf
#
#####

#--マルチノードエリアの指定-----#
dcmareas -m area1 -w nd01,nd02 # マルチノードエリアに属するノードを指定する
dcmareas -m area1 -w nd03,nd04

#--マルチノードサブエリアの指定-----#
dcmareas -g sub1 -w nd01,nd02
# マルチノードサブエリアに属するノードを指定する

### マルチノード構成定義の終わり #####

#####
# マルチノード物理定義
# ファイル名:nodeaddr
#
#####

dcprcport -w nd01 -h ft6101 -p 20000 # 各OpenTP1ノードについてOpenTP1
dcprcport -w nd02 -h ft6102 -p 20000 # ノードがあるホスト名とマルチ
dcprcport -w nd03 -h ft6103 -p 20000 # ノード連携制御機能が使用する
dcprcport -w nd04 -h ft6104 -p 20000 # ポート番号を指定する

### マルチノード物理定義の終わり #####

#####
# DAMサービス定義
# ファイル名:dam
#
#####

set dam_update_block = 32      # 更新する最大ブロック数
set dam_added_file   = 8      # オンライン中に追加する最大論理ファイル数

```

```

#-----論理ファイルの指定-----#
damfile DAMFILE /dev/rdsk/rhd112/damfile0
      # オンライン中にアクセスする論理ファイル名と物理ファイル名

### DAMサービス定義の終わり #####

#####
# TAMサービス定義
# テーブル名:tam
#
#####

set tam_max_tblnum = 100      # オンライン中に使用するテーブルの最大数
set tam_max_filesize = 50000 # オンライン中に使用する最大テーブルサイズ
set tam_max_recsz = 200      # TAMテーブル最大レコード長
set tam_jnl_err_flag = STOP  # ジャーナル障害時の処理形態
set tam_pool_attri = fixed   # 共用メモリプールを固定するかどうか

#-----TAMファイルの属性の指定-----#

tamtable TAMTABLE /dev/rdsk/rhd112/tamfile0
      # TAMテーブル名 物理ファイル名

### TAMサービス定義の終わり #####

#####
# ISTサービス定義
# ファイル名:ist
#
#####

set ist_node =node1,node2,node3      # テーブル操作をするノード名

istdef isttbl1 32 100      # アクセスするテーブル名,レコード長,レコード数
istdef isttbl2 128 10      # アクセスするテーブル名,レコード長,レコード数
istdef isttbl3 1024 10     # アクセスするテーブル名,レコード長,レコード数

### ISTサービス定義の終わり #####

#####
# メッセージキューサービス定義
# ファイル名:que
#
#####

set que_xidnum = 100      # 同時に実行するトランザクションの最大数

#-----メッセージキューファイルの指定-----#
quegrp -g otqgrp01 -f /dev/rdsk/rhd112/quef01 -n 128 -m 10 -w 80
      # -g 物理ファイルに割り当てるキューグループID
      # -f 物理ファイル名
      # -n バッファキャッシュ用の入出力バッファ数
      # -m キューファイルの保持メッセージ数
      # -w 物理ファイルの使用容量警告率

quegrp -g itqgrp01 -f /dev/rdsk/rhd112/quef02 -n 128 -m 10 -w 80

```

```

### メッセージキューサービス定義の終わり ###

#####
# ユーザサービスデフォルト定義
# ファイル名:usrrc
#
#####

set nice                = 10      # プロセスの実行優先度
set parallel_count      = 1      # 常駐プロセス数
set hold                = Y      # UAP異常終了時、閉塞するかどうか
set hold_recovery       = N      # 全面回復時、閉塞状態を引き継ぐか
set deadlock_priority   = 64     # UAPのデッドロックの優先順位
set schedule_priority   = 8      # スケジュールの優先順位
set message_buflen      = 4096   # メッセージ長
set message_store_buflen = 12288 # メッセージ格納プール長
set atomic_update       = Y      # トランザクションを発生させるかどうか
set receive_from        = queue  # スケジュールキューを使用するかどうか
set uap_trace_max       = 32     # UAPトレース格納最大数
set term_watch_time     = 30     # 連続異常終了限界経過時間
set type                = other  # サービスグループの種別
set balance_count       = 3      # サービス要求滞留値
set auto_restart        = N      # UAP異常終了時のサービスグループの処置
set critical            = N      # UAP異常終了時のシステムの処置
set lck_wait_priority    = 0      # 排他待ちの優先順位

### ユーザサービスデフォルト定義の終わり #####

#####
# ユーザサービス定義1
# ファイル名:exuap01
# クライアントUAP(SUP)の定義
#
#####

set module              = "upout1"      # 実行形式プログラム名
set receive_from        = none          # メッセージ送受信にスケジュールキューを使うかどうか

### ユーザサービス定義の終わり #####

#####
# ユーザサービス定義2
# ファイル名:exuap02
# サーバUAP(SPP)の定義
#
#####

set service_group       = "uap02"      # サービスグループ名
set module              = "upout2"     # 実行形式ファイル名
set parallel_count      = 2 , 4        # 常駐プロセス数と非常駐プロセス数
set service             = "serv02=serv02" # サービス名 = エントリポイント名

### ユーザサービス定義の終わり #####

#####
# ユーザサービス定義3

```

```

# ファイル名:exuap03
# メッセージ処理プログラム(MHP)の定義
#
#####

set service_group = "uap03"          # サービスグループ名
set module        = "upout3"         # 実行形式ファイル名
set parallel_count = 1, 2             # 常駐プロセス数と非常駐プロセス数
set hold          = N                 # サービスグループを閉塞するかどうか
set service       = "serv031=serv031", ¥
                  "serv032=serv032"
                                     # サービス名 = エントリポイント名
set type          = MHP               # サービスグループの種別

### ユーザサービス定義の終わり #####

#####
# MCFマネージャ定義
# ファイル名:abc_mngr
# オブジェクトファイル名:_mutest1
# 対応するMCF通信構成定義オブジェクトファイル名:_muua01,_mups01
#
#####

#-----MCFマネージャ環境定義-----#
mcfmenv -m "name = mng01"            # MCFマネージャ名

#-----MCFマネージャ共通定義-----#
mcfmcomn -n 10                       # 出力通番使用論理端末数 # ¥
          -p 300                     # MCF作業領域長 # ¥
          -j 4096                    # MCFマネージャプロセスのジャーナルバッファの大きさ

#-----通信サービス定義-----#
mcfmcname -s "mcfsvname = _muua01    # MCF通信サービス名 # ¥
              syssvname = mcfu01"    # システムサービス情報定義ファイル名

mcfmcname -s "mcfsvname = _mups01    # MCF通信サービス名 # ¥
              syssvname = mcfu02"    # システムサービス情報定義ファイル名

#-----UAP共通定義-----#
mcfmuap -d 65535                     # 通信関数発行回数上限値 # ¥
         -j 32768                    # ユーザサーバのジャーナルバッファの大きさ # ¥
         -l "initseq = 1             # 出力通番の初期値 # ¥
              maxseq = 9999          # ラップ時の出力通番最大値 # ¥
              minseq = 1"            # ラップ後の出力通番開始値

#-----入出力キュー定義-----#
mcfmqgid -q "quekind = otq           # キューの種別 # ¥
            quegrp = otqgrp01"       # キューグループID

mcfmqgid -q "quekind = itq           # キューの種別 # ¥
            quegrp = itqgrp01"       # キューグループID

#-----拡張予約定義-----#
mcfmexp -g 100                       # サービスグループの登録数 # ¥
         -l 30                       # 論理端末の登録数 # ¥
         -i dg # 入力キューが使用できないとき縮退運転するかどうか # ¥
         -o dg # 出力キューが使用できないとき縮退運転するかどうか

```

```

#-----状態引き継ぎ定義-----#
mcfmsts    -g 20          # サービスグループ数上限値 # ￥
           -v 100         # サービス数上限値

#####MCF マネジャ 定義の終わり#####

#####
# MCF通信構成定義(ua) 共通定義
# ファイル名:abc_ua_c
# オブジェクトファイル名:obj_ua_c
# 対応するデータコミュニケーション定義オブジェクトファイル名:obj_ua_d
# MCF通信構成定義オブジェクトファイル名:_muua01
# 対応するMCFアプリケーション定義オブジェクトファイル名:aplipt01
#
#
#####

#-----MCF通信環境定義-----#
mcftenv    -s 01          # MCF通信プロセス識別子 # ￥
           -a aplipt01    # MCFアプリケーション定義オブジェクトファイル名

#-----MCF通信構成共通定義-----#
mcftcomn   -j 32768       # MCF通信プロセスのジャーナルバッファの大きさ # ￥
           -x "termrls = no" # (UAプロトコル固有)

#-----最大処理多重度定義-----#
mcfttred   -m 5           # 最大処理多重度

#-----タイマ定義-----#
mcfttim    -t "btim      = 5      # 時間監視間隔 # ￥
           mtim      = 180      # 未処理送信メッセージ滞留時間 # ￥
           rmtim     = 180"     # 未処理受信メッセージ滞留時間

#-----トレース環境定義-----#
mcfttrc    -t "size      = 20480   # トレースバッファの大きさ # ￥
           disk      = yes        # ディスク出力機能を使用するかどうか # ￥
           bufcnt    = 50         # トレースバッファの数 # ￥
           trccnt    = 3"        # トレースファイルの数 # ￥
           -m del          # トレースファイルの数を越えたときの処置

#-----状態引き継ぎ定義-----#
mcftsts    -a 10          # アプリケーション数上限値

#-----バッファグループ定義-----#
###(送信用)
mcftbuf    -g "groupno = 1        # バッファグループ番号 # ￥
           length    = 4096      # バッファ長 # ￥
           count     = 256       # バッファ数 # ￥
           extend    = 256"     # 拡張バッファ数

###(受信用)
mcftbuf    -g "groupno = 2        # バッファグループ番号 # ￥
           length    = 4096      # バッファ長 # ￥
           count     = 256       # バッファ数 # ￥
           extend    = 256"     # 拡張バッファ数

###(編集用)
mcftbuf    -g "groupno = 3        # バッファグループ番号 # ￥

```



```

length = 4096    # バッファ長 # ¥
count  = 256     # バッファ数 # ¥
extend = 256"    # 拡張バッファ数

```

MCF通信構成定義(共通定義)の終わり

```

#####
# MCF通信構成定義(ua)
#   データコミュニケーション定義(プロトコル固有定義)
#   ファイル名:abc_ua_d
#   オブジェクトファイル名:obj_ua_d
#   対応する共通定義オブジェクトファイル名:obj_ua_c
#   MCF通信構成定義オブジェクトファイル名:_muua01
#   対応するMCFアプリケーション定義オブジェクトファイル名:aplipt01
#
#   OSAS/UAプロトコル固有定義のため
#   詳細は プロトコル TP1/NET/User Agent編 参照のこと
#
#####

```

```

#-----コネクション定義-----#
#-----コネクション定義の開始-----#
mcftalccn -c coua01          # コネクションID # ¥
          -p ua              # プロトコルの種別 # ¥
          -n x'0a81008202001283020012' ¥
                                # 自システムのPSAPアドレス # ¥
          -g "sndbuf = 1      # メッセージ送信用バッファグループ番号 # ¥
              rcvbuf = 2"    # メッセージ受信用バッファグループ番号 # ¥
          -e "msgbuf = 3      # メッセージ編集用バッファグループ番号 # ¥
              count = 5"     # メッセージ編集用バッファ数 # ¥
          -m "mode = xnfw"    # 通信管理の接続モード # ¥
          -i auto            # 再開始時にコネクションを自動確立するかどうか # ¥
          -o old             # OSAS/UA プロトコルの種別 # ¥
          -u ht              # 通信相手システムの種別 # ¥
          -y e'T1000'        # 制御UAの端末識別子 # ¥
          -q x'15810082008302ffff840b490001420103000005ffe01' ¥
                                # 通信相手システムのPSAPアドレス # ¥
          -z "slot = 1"      # 自システムで使用するスロット番号

```

```

#-----論理端末定義/UA定義(1)-----#
mcftalcle -l lotr01          # 論理端末名称 # ¥
          -t reply           # 端末タイプ # ¥
          -m "mmsgcnt = 20    # メモリ出力メッセージ最大格納数 # ¥
              dmsgcnt = 10"  # ディスク出力メッセージ最大格納数 # ¥
          -k "quekind = disk  # 出力キューの媒体の種類 # ¥
              quegrp01"      # キューグループID
mcftalcua -u 1               # UA番号 # ¥
          -y e'T1001'        # UA端末識別子

```

```

#-----論理端末定義/UA定義(2)-----#
mcftalcle -l lotr02          # 論理端末名称 # ¥
          -t send            # 端末タイプ # ¥
          -m "mmsgcnt = 20    # メモリ出力メッセージ最大格納数 # ¥
              dmsgcnt = 10"  # ディスク出力メッセージ最大格納数 # ¥
          -k "quekind = disk  # 出力キューの媒体の種類 # ¥
              quegrp02"      # キューグループID
mcftalcua -u 2               # UA番号 # ¥
          -y e'T1002'        # UA端末識別子

```



```

#-----コネクション定義の終了-----#
mcftalced                                # 引数なし

## MCF通信構成定義(データコミュニケーション定義)の終わり

#####
# MCF通信構成定義(psvr) 共通定義
# ファイル名:psvr_cmn
# オブジェクトファイル名:psvr_c_o
# 対応するデータコミュニケーション定義オブジェクトファイル名:psvr_d_o
# MCF通信構成定義オブジェクトファイル名:_mups01
# 対応するMCFアプリケーション定義オブジェクトファイル名:aplipt01
#
#
#####

#-----MCF通信環境定義-----#
mcftenv  -s 02                # アプリケーション起動プロセス識別子 # ¥
          -a aplipt01         # MCFアプリケーション定義オブジェクトファイル名

#-----MCF通信構成共通定義-----#
mcftcomn  -j 32768
           # アプリケーション起動プロセスのジャーナルバッファの大きさ

#-----最大処理多重度定義-----#
mcfttred  -m 1                # 最大処理多重度

#-----タイマ定義-----#
mcfttim   -t "btim = 5        # 時間監視間隔 # ¥
           mtim = 180         # 未処理送信メッセージ滞留時間 # ¥
           rmtim = 180"       # 未処理受信メッセージ滞留時間

#-----トレース環境定義-----#
mcfttrc   -t "size = 20480    # トレースバッファの大きさ # ¥
           disk = yes         # ディスク出力機能を使用するかどうか # ¥
           bufcnt = 50        # トレースバッファの数 # ¥
           trccnt = 3"        # トレースファイルの数 # ¥
           -m del             # トレースファイルの数を越えたときの処置

#-----状態引き継ぎ定義-----#
mcftsts   -a 10               # アプリケーション数上限値

##### MCF通信構成定義(共通定義)の終わり #####

#####
# MCF通信構成定義(psvr)
# データコミュニケーション定義(アプリケーション起動定義)
# ファイル名:psvr_dta
# オブジェクトファイル名:psvr_d_o
# 対応する共通定義オブジェクトファイル名:psvr_c_o
# MCF通信構成定義オブジェクトファイル名:_mups01
# 対応するMCFアプリケーション定義オブジェクトファイル名:aplipt01
#
#
#####

#----アプリケーション起動環境定義の開始----#

```

```

mcftpsvr -c copsvr01                                # 内部通信路名

#-----論理端末定義-----#
mcftalcle -l lotr03                                # 論理端末名称 # ¥
-t request                                          # 端末タイプ # ¥
-m "mmsgcnt = 20"                                  # メモリ出力メッセージ最大格納数 # ¥
  dmsgcnt = 10"                                    # ディスク出力メッセージ最大格納数 # ¥
-k "quekind = disk"                                # 出力キューの媒体の種類 # ¥
  quegrpid = otqgrp01"                             # キューグループID

#----アプリケーション起動環境定義の終了----#
mcftped                                             # 引数なし
## MCF通信構成定義(データコミュニケーション定義)の終わり

#####
# MCFアプリケーション定義
# ファイル名:abc_apli
# オブジェクトファイル名:aplipt01
#
#####
#-----MCFアプリケーション環境定義-----#
mcfaenv -a apli01                                  # MCFアプリケーション定義識別子 # ¥
-p 02                                              # アプリケーション起動プロセス識別子

#-----MCFアプリケーション属性定義(1)-----#
mcfaalcap -n "name = WORK1"                        # アプリケーション名 # ¥
  kind = user                                       # アプリケーションの種別 # ¥
  type = ans                                        # アプリケーションの型 # ¥
  aplihold = a # アプリケーションを閉塞するかどうか # ¥
  msgcnt = 600"                                    # 入力メッセージ最大格納数 # ¥
-g "servgrpn = uap03"                              # サービスグループ名 # ¥
  quegrpid = itqgrp01 # キューグループID # ¥
  quekind = disk" # 入力キューの媒体の種類 # ¥
-v "servname = serv031"                            # 対応するサービス名 # ¥
  servhold = m" # 異常終了時のサービスの処置 # ¥
-d "holdlimit = 1" # アプリケーション異常終了限界回数 # ¥
  holdlmtyp = cont" # 異常終了回数のカウント法 # ¥
-j "ij = yes" # 履歴情報を取得するかどうか # ¥
  # (入力メッセージ登録時) # ¥
  oj = yes # 履歴情報を取得するかどうか # ¥
  # (メッセージ送信要求時) # ¥
  gj = yes" # 履歴情報を取得するかどうか # ¥
  # (メッセージ受信要求時)

#-----MCFアプリケーション属性定義(2)-----#
mcfaalcap -n "name = WORK2"                        # アプリケーション名 # ¥
  kind = user                                       # アプリケーションの種別 # ¥
  type = ans                                        # アプリケーションの型 # ¥
  aplihold = a # アプリケーションを閉塞するかどうか # ¥
  msgcnt = 600 # 入力メッセージ最大格納数 # ¥
  cname = copsvr01" # 内部通信路名 # ¥
-g "servgrpn = uap03"                              # サービスグループ名 # ¥
  quegrpid = itqgrp01 # キューグループID # ¥
  quekind = disk" # 入力キューの媒体の種類 # ¥
-v "servname = serv032"                            # 対応するサービス名 # ¥
  servhold = m" # 異常終了時のサービスの処置 # ¥
-d "holdlimit = 1" # アプリケーション異常終了限界回数 # ¥
  holdlmtyp = cont" # 異常終了回数のカウント法 # ¥

```

```

        -j "ij"      = yes      # 履歴情報を取得するかどうか # ¥
                                #      (入力メッセージ登録時) # ¥
        oj          = yes      # 履歴情報を取得するかどうか # ¥
                                #      (メッセージ送信要求時) # ¥
        gj          = yes"     # 履歴情報を取得するかどうか
                                #      (メッセージ受信要求時)
##### MCFアプリケーション定義の終わり #####

#####
# システムサービス情報定義(ua)
# ファイル名:mcfu01
#
#####

set module = "mcfu01"
# ユーザが作成したMCFのメイン関数の実行形式プログラム名(=定義ファイル名)
set receive_from = socket      # 受信種別
set fixpriority = 52           # 固定プロセス実行優先度

### システムサービス情報定義の終わり #####

#####
# システムサービス情報定義(psvr)
# ファイル名:mcfu02
#
#####

set module = "mcfu02"
# ユーザが作成したMCFのメイン関数の実行形式プログラム名(=定義ファイル名)
set fixpriority = 52           # 固定プロセス実行優先度

### システムサービス情報定義の終わり #####

```

8.2 リアルタイム統計情報の取得項目の定義例

リアルタイム統計情報サービスが取得する統計情報の取得対象および取得項目は、rtspud 定義コマンドの指定方法によって変更できます。

ここでは、リアルタイム統計情報サービス定義の定義例と、その場合に取得される情報を示します。さらに、rts_service_max オペランド（最大取得サービス数）およびrts_item_max オペランド（最大取得項目数）の指定値による取得情報の違いもあわせて示します。

8.2.1 リアルタイム統計情報サービス定義の定義例

```
#####
# リアルタイム統計情報サービス定義
# ファイル名:rts
#
#####

set rts_service_max = 7
set rts_item_max    = 6

rtspud -u sys -f File1                ※1
rtspud -u srv -s supA -e 項目4, 項目5 -f File1    ※2
rtspud -u srv -s sppA -f File1            ※3
rtspud -u svc -s sppA -v update -e 項目4, 項目5, 項目6 ※4
rtspud -u svc -s sppB -v refer -e 項目4, 項目5    ※5
rtspud -u obj -o ポート番号 -b IPアドレス -e 項目6 ※6

### リアルタイム統計情報サービス定義の終わり #####

#####
# リアルタイム取得項目定義
# ファイル名:File1
#
#####

set 項目1 = Y
set 項目2 = Y
set 項目3 = Y

### リアルタイム取得項目定義の終わり #####

#####
# ユーザサービス定義1
# ファイル名:supA
#
#####

set module = "supA"
set receive_from = none

### ユーザサービス定義1の終わり#####
```

```
#####
# ユーザサービス定義2
# ファイル名:sppA
#
#####

set service_group = "sppA_svg"
set module = "sppA"
set service = "refer=refer","update=update"

### ユーザサービス定義2の終わり#####

#####
# ユーザ サービス定義3
# ファイル名:sppB
#
#####

set service_group = "sppB_svg"
set module = "sppB"
set service = "refer=refer","update=update"

### ユーザサービス定義3の終わり#####
```

上記のように rtsput 定義コマンドを指定する場合、rts_service_max オペランドには 7 以上の値を指定し、rts_item_max オペランドには 6 以上の値を指定する必要があります。

この定義例に基づいてリアルタイム統計情報サービスを開始した場合の取得対象および取得項目を次の表に示します。

表 8-1 定義例に基づいて取得できる内容

項番	取得対象		取得項目					
1	_SYSTEM	△△△△	項目 1	項目 2	項目 3	—	—	—
2	supA	△△△△	項目 1	項目 2	項目 3	項目 4	項目 5	—
3	sppA	△△△△	項目 1	項目 2	項目 3	—	—	—
4	sppA	****	項目 1	項目 2	項目 3	—	—	—
5	sppA	refer	項目 1	項目 2	項目 3	—	—	—
6	sppA	update	項目 1	項目 2	項目 3	項目 4	項目 5	項目 6
7	sppB	refer	項目 4	項目 5	—	—	—	—
8	ポート番号	IP アドレス	項目 6	—	—	—	—	—

(凡例)

△△△△：サーバ単位で取得したリアルタイム統計情報

****：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

—：該当なし

注※1

rts_service_max オペランドに指定した値に関係なく、表 8-1 の項番 1 のようにシステム全体の統計情報を取得対象として登録します。

注※2

-u オプションに srv を指定し、ユーザサービス定義内に service オペランドを定義していない場合、表 8-1 の項番 2 のようにサービス全体の統計情報を取得対象として登録します。

注※3

-u オプションに srv を指定し、ユーザサービス定義内に service オペランドを定義した場合、表 8-1 の項番 3～項番 6 のように、サーバ単位およびサービス以外の処理単位の統計情報、ならびに service オペランドに定義したサービスを統計情報の取得対象として登録します。

注※4

すでに登録されている取得対象に対して rtspat 定義コマンドで定義した場合、表 8-1 の項番 6 のように取得項目をまとめて登録します。

注※5

-s オプションに指定したサーバ名と -v オプションに指定したサービス名を組み合わせると表 8-1 の項番 7 のように一つの取得対象として登録します。

注※6

-o オプションおよび -b オプションの引数に指定した値を組み合わせると表 8-1 の項番 8 のように一つの取得対象として登録します。

8.2.2 オペランドの指定値による取得情報の違い

リアルタイム統計情報サービスは、最大で「rts_service_max オペランドの指定値+1」（システム全体の統計情報）分の取得対象の統計情報を取得できます。各取得対象は、rts_item_max オペランドの指定値分の取得項目を取得できます。

また、rtspat 定義コマンドに、rts_service_max オペランドおよび rts_item_max オペランドの指定値より多くの取得対象および取得項目を指定した場合、リアルタイム統計情報サービスは、登録できる分だけを登録して、統計情報を取得します。

次に、rts_service_max オペランドおよび rts_item_max オペランドの指定値より多くの取得対象および取得項目を rtspat 定義コマンドに指定した場合の定義例を示します。

```
#####
# リアルタイム統計情報サービス定義
# ファイル名: rts
#
#####

set rts_service_max = 6
set rts_item_max    = 5

rtspat -u sys -f File1                ※1
rtspat -u srv -s supA -e 項目4, 項目5 -f File1    ※2
rtspat -u srv -s sppA -f File1            ※3
rtspat -u svc -s sppA -v update -e 項目4, 項目5, 項目6  ※4
rtspat -u svc -s sppB -v refer -e 項目4, 項目5      ※5
```

```

rtsput -u obj -o ポート番号 -b IPアドレス -e 項目6 ※6

### リアルタイム統計情報サービス定義の終わり #####

#####
# リアルタイム取得項目定義
# ファイル名:File1
#
#####

set 項目1 = Y
set 項目2 = Y
set 項目3 = Y

### リアルタイム取得項目定義の終わり #####

#####
# ユーザ サービス定義1
# ファイル名:supA
#
#####

set module = "supA"
set receive_from = none

### ユーザサービス定義1の終わり#####

#####
# ユーザ サービス定義2
# ファイル名:sppA
#
#####

set service_group = "sppA_svg"
set module = "sppA"
set service = "refer=refer","update=update"

### ユーザサービス定義2の終わり#####

#####
# ユーザ サービス定義3
# ファイル名:sppB
#
#####

set service_group = "sppB_svg"
set module = "sppB"
set service = "refer=refer","update=update"

### ユーザサービス定義3の終わり#####

```

「[8.2.1 リアルタイム統計情報サービス定義の定義例](#)」で示した定義例では rts_service_max オペランドに 7 を、rts_item_max オペランドに 6 を指定しているのに対し、この定義例では rts_service_max オペランドに 6 を、rts_item_max オペランドに 5 を指定しています。この指定値の違いによって、取得対象および取得項目は、次の表のように「[8.2.1 リアルタイム統計情報サービス定義の定義例](#)」の場合とは異なります。

表 8-2 指定値より多くの取得対象および取得項目を指定した場合に取得できる内容

項番	取得対象		取得項目				
1	_SYSTEM	△△△△	項目 1	項目 2	項目 3	—	—
2	supA	△△△△	項目 1	項目 2	項目 3	項目 4	項目 5
3	sppA	△△△△	項目 1	項目 2	項目 3	—	—
4	sppA	****	項目 1	項目 2	項目 3	—	—
5	sppA	refer	項目 1	項目 2	項目 3	—	—
6	sppA	update	項目 1	項目 2	項目 3	項目 4	項目 5
7	sppB	refer	項目 5	項目 6	—	—	—

(凡例)

△△△△：サーバ単位で取得したリアルタイム統計情報

****：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

—：該当なし

注※1

rts_service_max オペランドに指定した値に関係なく、表 8-2 の項番 1 のようにシステム全体の統計情報を取得対象として登録します。

注※2

-u オプションに srv を指定し、ユーザサービス定義内に service オペランドを定義していない場合、表 8-2 の項番 2 のようにサービス全体の統計情報を取得対象として登録します。

注※3

-u オプションに srv を指定し、ユーザサービス定義内に service オペランドを定義した場合、表 8-2 の項番 3～項番 6 のように、サーバ単位およびサービス以外の処理単位の統計情報、ならびに service オペランドに定義したサービスを統計情報の取得対象として登録します。

注※4

取得項目を三つ登録した場合に-e オプションに指定した項目のうち二つを登録すると、rts_item_max オペランドの指定値に達するため、表 8-2 の項番 6 のように項目 6 を取得項目として登録しません。

注※5

-s オプションに指定したサーバ名と-v オプションに指定したサービス名を組み合わせると表 8-2 の項番 7 のように一つの取得対象として登録します。

注※6

表 8-2 の項番 1 を除いた取得対象の登録数が rts_service_max オペランドの指定値に達しているため、取得対象として登録しません。

付録

付録 A アプリケーションプログラムとシステム環境設定の関連

UAP とシステム環境設定の関係について説明します。システム環境設定の方法は、OpenTP1 の基本機能によって異なります。

- TP1/Server Base の場合
テキストエディタで、必要なシステム定義ファイルを作成します。
- TP1/LiNK の場合
OS が UNIX の場合は、対話形式のコマンドで実行環境を設定します。OS が Windows の場合は、GUI 画面で実行環境を設定します。
TP1/Server Base の場合には、システム定義が必要です。

付録 A.1 アプリケーションプログラムに関連する OpenTP1 のシステム定義

UAP に関連する OpenTP1 のシステム定義について説明します。システム定義は、OpenTP1 の基本機能が TP1/Server Base の場合にだけ必要です。以降の記述は、TP1/Server Base の場合に該当します。

(1) UAP に必要な定義

UAP に必要なシステム定義は次のとおりです。

(a) システム共通定義

OpenTP1 システム共通の実行環境を決定する定義です。RPC の最大応答待ち時間などを指定します。

(b) ユーザサービス構成定義

OpenTP1 システムの開始時に、一緒に開始する UAP を決定する定義です。

(c) ユーザサービス定義

オンラインで起動された UAP の、実行環境を決定する定義です。UAP ごとに作成します。

(d) ユーザサービスデフォルト定義

ユーザサービス定義の項目のうち、省略した項目のデフォルト値を指定する定義です。

(e) MCF アプリケーション定義

メッセージ制御機能（MCF）を使うときに、UAP のアプリケーション名を決定する定義です。ユーザサービス定義で実行環境を設定した UAP を、MCF で認識するために定義します。

UAP に関連する OpenTP1 のシステム定義を次の表に示します。

表 A-1 UAP に関連する OpenTP1 のシステム定義

区分	システム定義				
	システム共通定義	ユーザサービス構成定義	ユーザサービス定義	ユーザサービスデフォルト定義	MCF アプリケーション定義
定義を作成する単位	一つのノード単位		サービスグループ単位	一つのノード単位	複数の MCF 通信プロセスで一つ、または MCF 通信プロセス単位
使用する時期	SUP, SPP, MHP の起動時	システム正常開始時	SUP, SPP, MHP の起動時	SUP, SPP, MHP の起動時	MHP の起動時
定義の変更方法 (set 形式の場合※)	定義を変更して、変更前のノードの OpenTP1 を終了させて、新しい定義でノードの OpenTP1 を起動します。		定義を変更して変更前の定義のプロセスを終了させて、新しい定義で再び UAP を起動します。		

注※

コマンド形式の定義項目は、オンライン中に変更できます。

(2) UAP に関連する OpenTP1 のシステム定義のファイル名

OpenTP1 のシステム定義ファイルは、\$DCCONFPATH の下に作成されます。UAP に関連する OpenTP1 のシステム定義のファイル名を次の表に示します。

表 A-2 UAP に関連する OpenTP1 のシステム定義のファイル名

UAP に関連するシステム定義	定義ファイル名 (完全パス名)
システム共通定義	\$DCCONFPATH/betranrc
ユーザサービス構成定義	\$DCCONFPATH/usrconf
ユーザサービス定義	\$DCCONFPATH/ユーザサービス構成定義で指定するユーザサーバ名または、サーバの開始コマンド (dcsvstart) で指定するユーザサーバ名
ユーザサービスデフォルト定義	\$DCCONFPATH/usrrc
MCF アプリケーション定義	\$DCCONFPATH/ノード内で一意となる任意の名称

(3) サービス名とアプリケーション名 (set service, mcfaalcap)

MHP のサービスは、MCF で受信したメッセージの先頭セグメントの先頭 8 文字のアプリケーション名を基にスケジュールされます。システム定義では、ユーザサービス定義で指定した MHP のサービスグループ名とサービス名を、MCF アプリケーション定義にも指定して、二つの定義を対応付けます。MCF アプリケーション定義のアプリケーション属性定義のサービスグループ名とサービス名は、ユーザサービス定義で指定されている名称を指定してください。

付録 A.2 トランザクションの時間監視

トランザクションの開始から同期点の取得までの間の限界経過時間に、他プロセスで実行するトランザクションブランチの処理時間（同期応答型 RPC の処理時間）を含めることも含めないこともできます。

トランザクションの限界経過時間の監視方法は、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 `trn_expiration_time_suspend` で指定します。

タイマ監視機能の運用について、`trn_expiration_time_suspend` オペランドの指定と各種タイマ値との関係を説明します。

(1) `trn_expiration_time_suspend` オペランドと各種タイマ監視値の関係

`trn_expiration_time_suspend` オペランドの指定値と各種タイマ監視値の関係を図 A-1 に示します。

`trn_expiration_time_suspend` オペランドは図 A-1 の SPP1 に指定します。このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義、またはトランザクションサービス定義に指定できます。指定値の優先順位は次のとおりです（1.> 2.> 3.）。

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

このオペランドに Y を指定した場合と、N または F を指定した場合とでは、トランザクションブランチ限界経過時間に RPC の処理時間を含めるかどうか（図 A-1 の t3 で表す監視対象区間）が異なります。

図 A-1 trn_expiration_time_suspend オペランドと各種タイマ値の関係

図中の t1～t10 について次に説明します。

- SUP のユーザサービス定義の watch_time オペランドの指定値
- SUP のユーザサービス定義で watch_time オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の watch_time オペランドの指定値
- SUP のユーザサービス定義およびユーザサービスデフォルト定義で watch_time オペランドの指定を省略した場合は、システム共通定義の watch_time オペランドの指定値
- すべての定義で watch_time オペランドの指定を省略した場合は 180 秒

RPC を受け取ってサービス関数を開始してから、サービス関数終了までの時間を監視します。

タイムアウト時には、KFCA00502-I メッセージが種別コード=3 で出力され、SPP1 がダウンします。
次に示すどれかの値が監視時間の最大値となります。

- SPP1 のユーザサービス定義の service_expiration_time オペランドの指定値
- SPP1 のユーザサービス定義で service_expiration_time オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の service_expiration_time オペランドの指定値
- すべての定義で service_expiration_time オペランドの指定を省略した場合は 0（時間監視しない）

t3: trn_expiration_time（トランザクションブランチ限界経過時間）

トランザクション開始から、同期点処理を開始するまでの時間を監視します。

SPP1 の trn_expiration_time_suspend オペランドの指定によって、監視対象区間が異なります。監視区間は次のとおりです。

trn_expiration_time_suspend オペランドの指定が Y の場合

t3(Y)で示す区間

trn_expiration_time_suspend オペランドの指定が N または F の場合

t3(N1)と t3(N2)で示す区間の合計

タイムアウト時には、KFCA00502-I メッセージが種別コード=1 で出力され、SPP1 がダウンします。
次に示すどれかの値が監視時間の最大値となります。

- SPP1 のユーザサービス定義の trn_expiration_time オペランドの指定値
- SPP1 のユーザサービス定義で trn_expiration_time オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の trn_expiration_time オペランドの指定値
- SPP1 のユーザサービス定義およびユーザサービスデフォルト定義で trn_expiration_time オペランドの指定を省略した場合は、トランザクションサービス定義の trn_expiration_time オペランドの指定値
- すべての定義で trn_expiration_time オペランドの指定を省略した場合は 0（時間監視しない）

t4: watch_time（最大応答待ち時間）

RPC を発行してから応答が返ってくるまでの時間を監視します。

タイムアウト時には dc_rpc_call()が DCRPCER_TIMED_OUT（-307）でエラーリターンします。

次に示すどれかの値が監視時間の最大値となります。

- SPP1 のユーザサービス定義の watch_time オペランドの指定値
- SPP1 のユーザサービス定義で watch_time オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の watch_time オペランドの指定値
- SPP1 のユーザサービス定義およびユーザサービスデフォルト定義で watch_time オペランドの指定を省略した場合は、システム共通定義の watch_time オペランドの指定値
- すべての定義で watch_time オペランドの指定を省略した場合は 180 秒

t5: service_expiration_time（サービス関数開始から終了までの実行監視時間）

RPC を受け取ってサービス関数を開始してから、サービス関数終了までの時間を監視します。

タイムアウト時には、KFCA00502-I メッセージが種別コード=3 で出力され、SPP2 がダウンします。
次に示すどれかの値が監視時間の最大値となります。

- SPP2 のユーザサービス定義の service_expiration_time オペランドの指定値
- SPP2 のユーザサービス定義で service_expiration_time オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の service_expiration_time オペランドの指定値
- すべての定義で service_expiration_time オペランドの指定を省略した場合は 0（時間監視しない）

t6：trn_expiration_time（トランザクションブランチ限界経過時間）

トランザクション開始から、同期点処理を開始するまでの時間を監視します。

タイムアウト時には、KFCA00502-I メッセージが種別コード=1 で出力され、SPP2 がダウンします。
次に示すどれかの値が監視時間の最大値となります。

- SPP2 のユーザサービス定義の trn_expiration_time オペランドの指定値
- SPP2 のユーザサービス定義で trn_expiration_time オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の trn_expiration_time オペランドの指定値
- SPP2 のユーザサービス定義およびユーザサービスデフォルト定義で trn_expiration_time オペランドの指定を省略した場合は、トランザクションサービス定義の指定値
- すべての定義で trn_expiration_time オペランドの指定を省略した場合は 0（時間監視しない）

t7：trn_watch_time（トランザクション同期点処理時の最大通信待ち時間）

トランザクションの同期点処理で、トランザクションブランチ間の通信（プリペア、コミット、ロールバック指示、または応答など）の受信待ち時間です。

次に示すどれかの値が待ち時間の最大値となります。

- SPP1 のユーザサービス定義の trn_watch_time オペランドの指定値
- SPP1 のユーザサービス定義で trn_watch_time オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の trn_watch_time オペランドの指定値
- SPP1 のユーザサービス定義およびユーザサービスデフォルト定義で trn_watch_time オペランドの指定を省略した場合は、トランザクションサービス定義の trn_watch_time オペランドの指定
- SPP1 のすべての定義で trn_watch_time オペランドの指定を省略した場合は、SPP1 の watch_time オペランドの指定値

t8：trn_watch_time（トランザクション同期点処理時の最大通信待ち時間）

トランザクションの同期点処理で、トランザクションブランチ間の通信（プリペア、コミット、ロールバック指示、または応答など）の受信待ち時間です。

次に示すどれかの値が待ち時間の最大値となります。

- SPP2 のユーザサービス定義の trn_watch_time オペランドの指定値
- SPP2 のユーザサービス定義で trn_watch_time オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の trn_watch_time オペランドの指定値

- SPP2 のユーザサービス定義およびユーザサービスデフォルト定義で `trn_watch_time` オペランドの指定を省略した場合は、トランザクションサービス定義の `trn_watch_time` オペランドの指定値
- SPP2 のすべての定義で `trn_watch_time` オペランドの指定を省略した場合は、SPP2 の `watch_time` オペランドの指定値

t9: `trn_completion_limit_time` (トランザクション完了限界時間)

トランザクション開始から、終了までの時間を監視します。

タイムアウト時には、KFCA00502-I メッセージが種別コード=4 で出力され、SPP1 がダウンします。次に示すどれかの値が監視時間の最大値となります。

- SPP1 のユーザサービス定義の `trn_completion_limit_time` オペランドの指定値
- SPP1 のユーザサービス定義で `trn_completion_limit_time` オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の `trn_completion_limit_time` オペランドの指定値
- SPP1 のユーザサービス定義およびユーザサービスデフォルト定義で `trn_completion_limit_time` オペランドの指定を省略した場合は、トランザクションサービス定義の `trn_completion_limit_time` オペランドの指定値
- SPP1 のすべての定義で `trn_completion_limit_time` オペランドの指定を省略した場合は 0 (時間監視しない)

t10: `trn_completion_limit_time` (トランザクション完了限界時間)

トランザクション開始から、終了までの時間を監視します。

タイムアウト時には、KFCA00502-I メッセージが種別コード=4 で出力され、SPP2 がダウンします。次に示すどれかの値が監視時間の最大値となります。

- SPP2 のユーザサービス定義の `trn_completion_limit_time` オペランドの指定値
- SPP2 のユーザサービス定義で `trn_completion_limit_time` オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の `trn_completion_limit_time` オペランドの指定値
- SPP2 のユーザサービス定義およびユーザサービスデフォルト定義で `trn_completion_limit_time` オペランドの指定を省略した場合は、トランザクションサービス定義の `trn_completion_limit_time` オペランドの指定値
- SPP2 のすべての定義で `trn_completion_limit_time` オペランドの指定を省略した場合は 0 (時間監視しない)

(2) `trn_expiration_time_suspend` オペランドと各種タイマ監視値の関係 (連鎖 RPC を使用した場合)

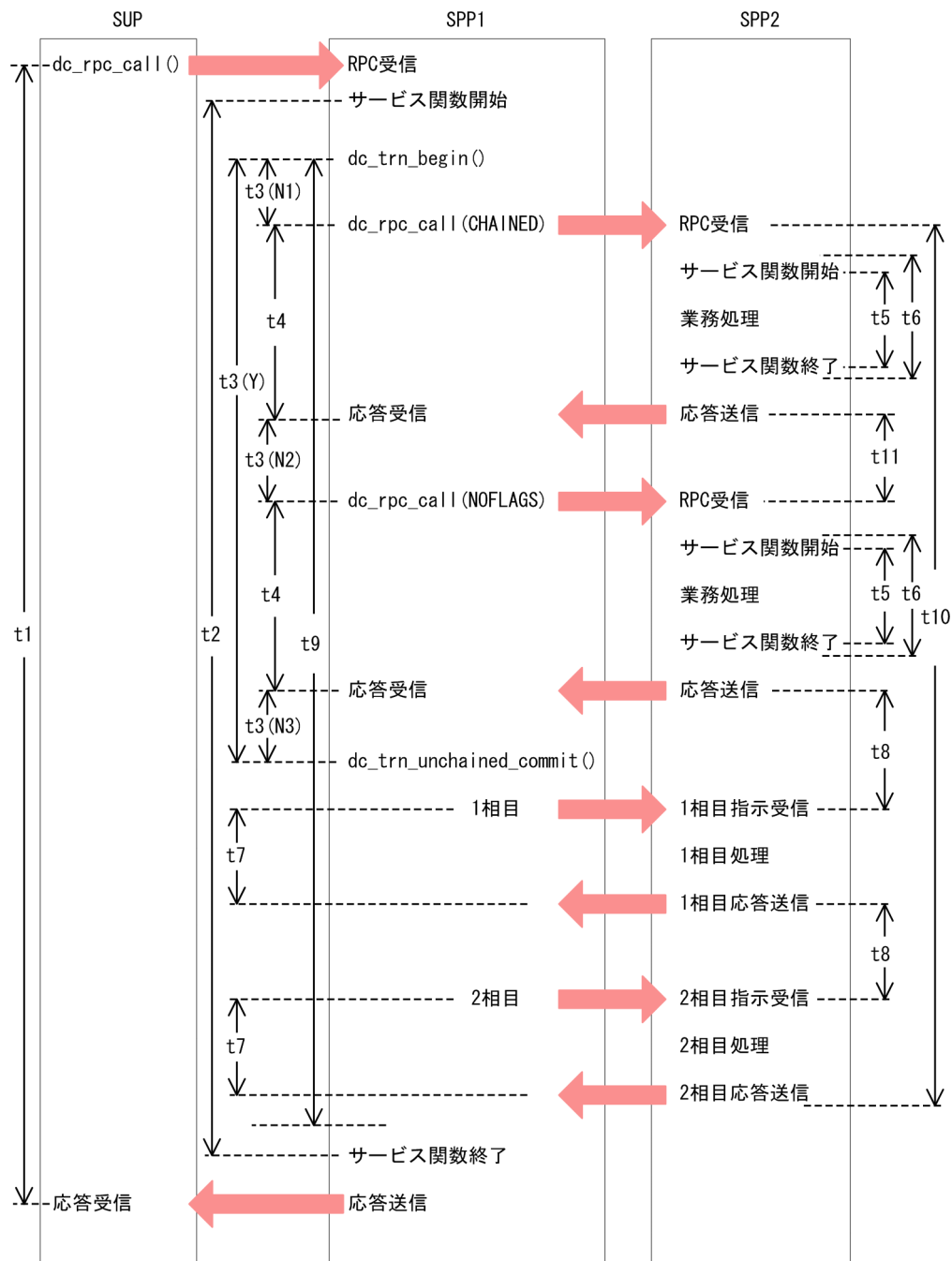
連鎖 RPC を使用した場合の、`trn_expiration_time_suspend` オペランドの指定値と各種タイマ監視値の関係を、図 A-2 に示します。

`trn_expiration_time_suspend` オペランドは図 A-2 の SPP1 に指定します。このオペランドは、ユーザサービス定義、ユーザサービスデフォルト定義、またはトランザクションサービス定義に指定できます。指定値の優先順位は次のとおりです (1.> 2.> 3.)。

- ### 3. トランザクションサービス定義

このオペランドに Y を指定した場合と、N または F を指定した場合とでは、トランザクションブランチ限界経過時間に RPC の処理時間を含めるかどうか（図 A-2 の t3 で表す監視対象区間）が異なります。

図 A-2 連鎖 RPC を使用した場合の各種タイマ値の関係 (連鎖 RPC を使用した場合)



図中の t1, t2, t4～t10 については、図 A-1 の説明を参照してください。t3 および t11 について、次に説明します。

t3: trn_expiration_time (トランザクションブランチ限界経過時間)

トランザクション開始から、コミットされるまでの時間を監視します。

SPP1 の trn_expiration_time_suspend オペランドの指定によって、監視対象区間が異なります。監視区間は次のとおりです。

trn_expiration_time_suspend オペランドの指定が Y の場合

t3(Y)で示す区間

trn_expiration_time_suspend オペランドの指定が N または F の場合

t3(N1), t3(N2)および t3(N3)で示す区間の合計

タイムアウト時には、KFCA00502-I メッセージが種別コード=1 で出力され、SPP1 がダウンします。次に示すどれかの値が監視時間の最大値となります。

- SPP1 のユーザサービス定義の trn_expiration_time オペランドの指定値
- SPP1 のユーザサービス定義で trn_expiration_time オペランドの指定を省略した場合は、ユーザサービスデフォルト定義の trn_expiration_time オペランドの指定値
- SPP1 のユーザサービス定義およびユーザサービスデフォルト定義で trn_expiration_time オペランドの指定を省略した場合は、トランザクションサービス定義の trn_expiration_time オペランドの指定値
- すべての定義で trn_expiration_time オペランドの指定を省略した場合は 0 (時間監視しない)

tl1: watch_next_chain_time (連鎖 RPC 間隔監視時間)

連鎖 RPC の応答を返してから、次の要求が来るまでの時間を監視します。

タイムアウト時には KFCA00315-E メッセージを出力し、SPP2 がダウンします。

次に示すどれかの値が監視時間の最大値となります。

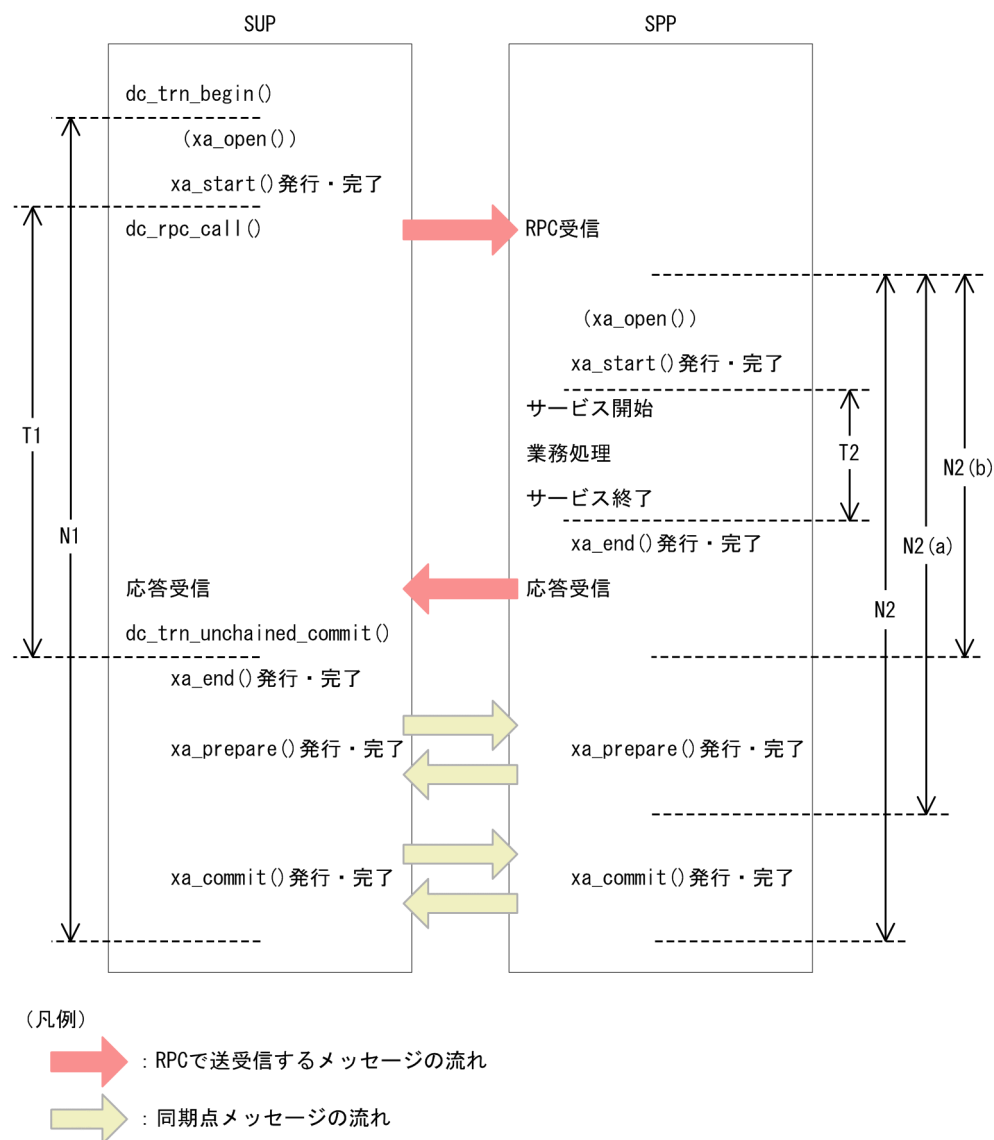
- SPP2 のユーザサービス定義の watch_next_chain_time オペランドの指定値
- SPP2 のユーザサービス定義で watch_next_chain_time オペランドを省略した場合は、ユーザサービスデフォルト定義の watch_next_chain_time オペランドの指定値
- すべての定義で watch_next_chain_time オペランドの指定を省略した場合は 180 秒

(3) trn_completion_limit_time オペランドの監視区間

trn_completion_limit_time オペランドの監視区間には、XA インタフェース処理、サーバ UAP 呼び出し処理、およびすべてのユーザ処理が監視対象として含まれます。ただし、トランザクションの最適化 (コミット最適化、およびプリペア最適化) が行われた場合は、SPP プロセスで実行中のトランザクション処理が終了した時点で時間監視を終了します。トランザクションの最適化の詳細については、マニュアル「OpenTP1 プログラム作成の手引」を参照してください。

trn_expiration_time オペランドと trn_completion_limit_time オペランドの監視区間の差異を次の図に示します。

図 A-3 trn_expiration_time オペランドと trn_completion_limit_time オペランドの監視区間の差異



図中の T および N について次に説明します。

T1 : SUP での trn_expiration_time オペランドの監視区間 (トランザクションブランチ限界経過時間)

T2 : SPP での trn_expiration_time オペランドの監視区間 (トランザクションブランチ限界経過時間)

N1 : SUP での trn_completion_limit_time オペランドの監視区間 (トランザクション完了限界時間)

N2 : SPP での trn_completion_limit_time オペランドの監視区間 (トランザクション完了限界時間)

SPP プロセスのサービス関数が開始してから、コミット完了のメッセージを送信したあとの時間を監視します。

N2(a)：コミット最適化が行われた場合の `trn_completion_limit_time` オペランドの監視区間（トランザクション完了限界時間）

SPP プロセスのサービス関数が開始してから、プリペア完了のメッセージを送信したあとまでの時間を監視します。

N2(b)：プリペア最適化が行われた場合 `trn_completion_limit_time` オペランドの監視区間（トランザクション完了限界時間）

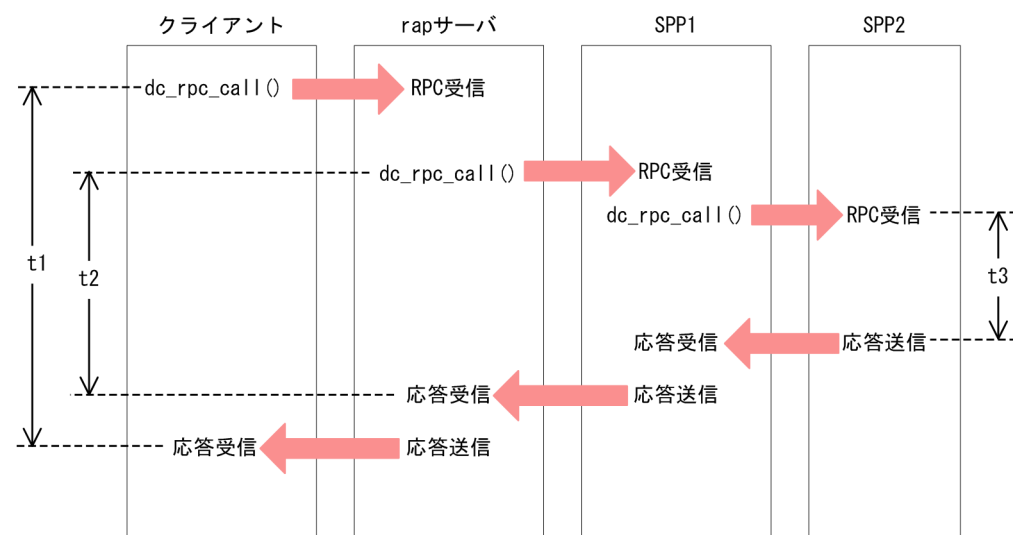
SPP プロセスのサービス関数が開始してから、サービス関数完了の応答を送信したあとまでの時間を監視します。

付録 A.3 サービス応答の待ち時間

クライアントがサービス要求を送ってから、サービスの応答が返るまでの待ち時間について、図 A-4 に示します。

rap サーバが `dc_rpc_call()` を発行してから、サービスの応答が返るまでの待ち時間の最大値は、クライアント環境定義の `DCWATCHTIMINHERIT` オペランドの指定によって異なります。

図 A-4 サービス応答の待ち時間



図中の $t1 \sim t3$ について次に説明します。

$t1$

クライアントが `dc_rpc_call()` を発行してサービス要求してから、クライアントにサービスの応答が返るまでの待ち時間です。クライアント環境定義の `dcwatchtim` オペランドの指定値が待ち時間の最大値となります。

$t2$

rap サーバが `dc_rpc_call()` を発行してサービス要求してから、rap サーバにサービスの応答が返るまでの待ち時間です。クライアント環境定義の `DCWATCHTIMINHERIT` オペランドの指定によって最大値が異なります。

DCWATCHTIMINHERIT=Y の場合

クライアント環境定義の dcwatchtim オペランドの指定値からクライアント環境定義の dccltdelay オペランドの指定値を引いた値が待ち時間の最大値となります。

DCWATCHTIMINHERIT=N の場合

rap リスナーサービス定義の watch_time オペランドの指定値が待ち時間の最大値となります。rap リスナーサービス定義で watch_time オペランドを省略した場合は、ユーザサービスデフォルト定義の watch_time オペランドの指定値が待ち時間の最大値となります。

t3

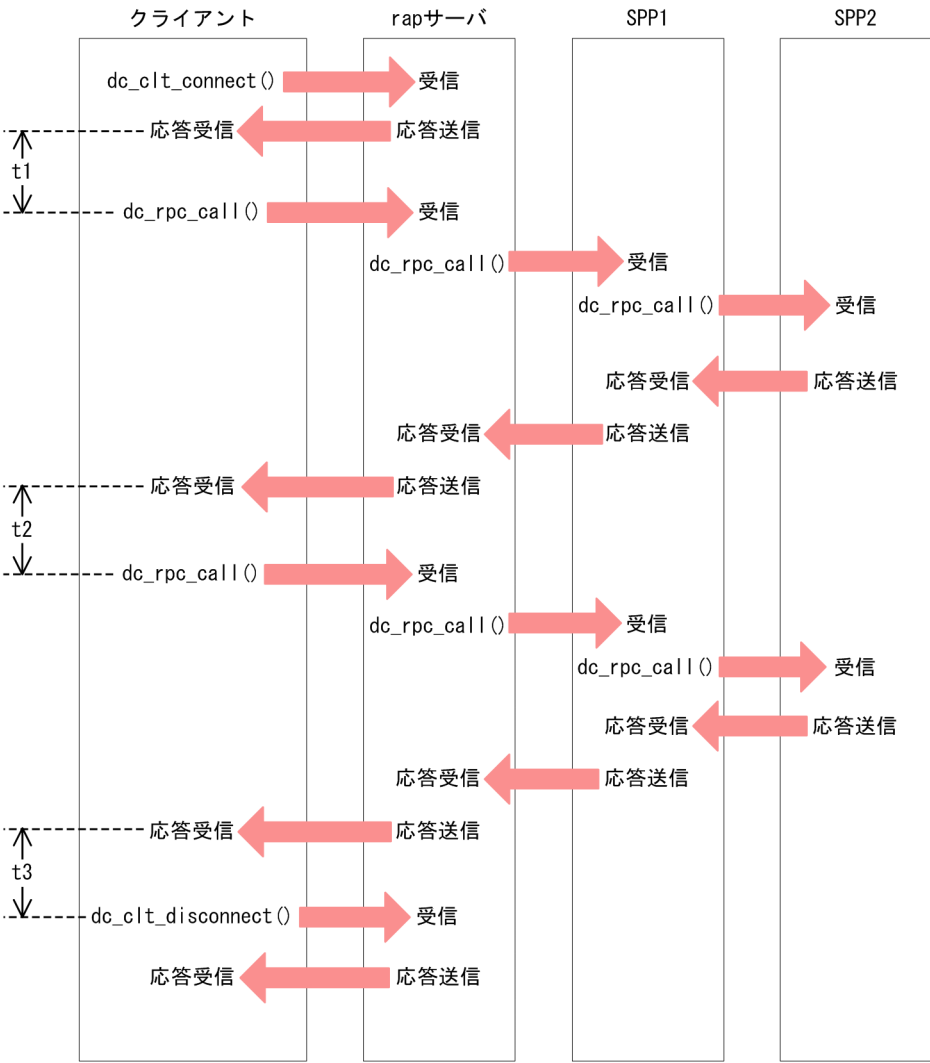
SPP1 が dc_rpc_call() を発行してサービス要求してから、SPP1 にサービスの応答が返るまでの待ち時間です。SPP1 のユーザサービス定義の watch_time オペランドの指定値が待ち時間の最大値となります。ユーザサービス定義で watch_time オペランドを省略した場合は、ユーザサービスデフォルト定義の watch_time オペランドの指定値が待ち時間の最大値となります。

付録 A.4 常設コネクション問い合わせ間隔最大時間の適用範囲

常設コネクション問い合わせ間隔最大時間は、クライアント環境定義の dccltinquiretime オペランドで指定します。クライアント環境定義で dccltinquiretime オペランドを省略した場合は、rap リスナーサービス定義の rap_inquire_time オペランドの指定値が常設コネクション問い合わせ間隔最大時間となります。

常設コネクション問い合わせ間隔最大時間とは、CUP が rap サーバに対して問い合わせしてから、次の問い合わせをするまでの間隔の最大値です。常設コネクション問い合わせ間隔最大時間は、CUP 実行プロセスまたは rap サーバで監視します。

図 A-5 常設コネクション問い合わせ間隔最大時間の適用範囲



図中の t1～t3 は、常設コネクション問い合わせ間隔最大時間が適用される範囲での rap サーバの監視時間です。

付録 A.5 アプリケーションプログラムが閉塞する場合

UAP が異常終了すると、OpenTP1 は UAP のプロセスを自動的に再起動させます。ただし、異常終了と再起動の繰り返しでシステムの処理がループしないように、OpenTP1 は指定した回数を超えて異常終了した UAP を閉塞します。また、異常終了した UAP を閉塞することで、クライアントからのサービス要求に意図的にエラーを返して不要な再実行を防ぐようにしています。

閉塞の対象になる UAP は、SPP（ただし、キュー受信型サーバの SPP）です。SUP、オフラインの業務をする UAP、およびソケット受信型サーバの SPP は、閉塞の対象になりません。

以降の説明は、SPP についてのものです。MHP については該当しません。

(1) 閉塞と監視時間の関係

異常終了した UAP をいつ閉塞するかは、監視時間と異常終了した回数で決まります。

TP1/Server Base の場合、システム定義に指定した監視時間内に UAP が 3 回異常終了すると、OpenTP1 は 3 回目の異常終了時に UAP を閉塞させます。TP1/LiNK の場合は、1 回目の異常終了時に UAP を閉塞させます。閉塞の時間監視はしません。

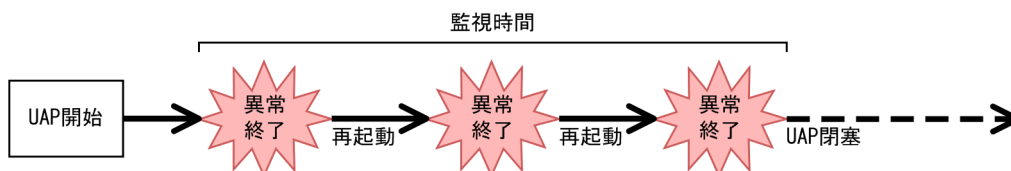
監視時間内の異常終了回数が 2 回以内の場合は、UAP のプロセスを OpenTP1 が再起動します。

時間の監視は、1 回目の異常終了から開始します。異常終了回数が 2 回以内の場合でも、2 回目の異常終了からさらに異常終了回数の監視が始まっています。

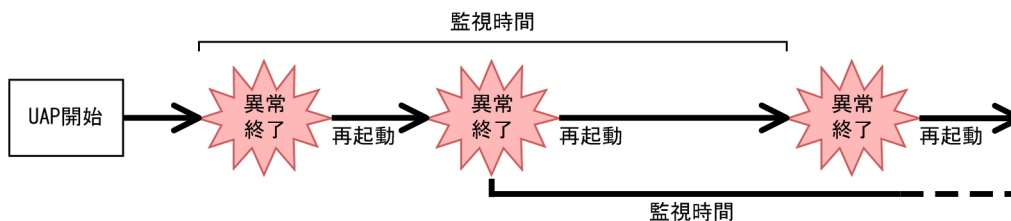
監視時間と異常終了回数の関係を次の図に示します。

図 A-6 監視時間と異常終了回数の関係

●閉塞する場合



●閉塞しない場合



(2) サービスグループ単位の閉塞とサービス単位の閉塞

UAP を閉塞する単位では、サービスグループ単位の閉塞とサービス単位の閉塞があります。サービス単位の閉塞を指定しておく、監視時間内に UAP が 3 回異常終了しても、該当するサービスだけを閉塞します。そのため、同じサービスグループのほかのサービスは処理を続けられます。

サービス単位の閉塞を指定できるのは、キュー受信型サーバの SPP だけです。また、サービスグループ単位の閉塞を指定できるのは、OpenTP1 の基本機能が TP1/Server Base の場合だけです。TP1/LiNK の SPP の場合は、サービス単位の閉塞だけです。

(3) アプリケーションプログラムの閉塞に関するユーザサービス定義

UAP の閉塞に関するユーザサービス定義を次に示します。ユーザサービス定義は、OpenTP1 の基本機能が TP1/Server Base の場合だけ指定できます。TP1/LiNK の SPP の場合は、UAP が異常終了した時点でサービス単位にすぐに閉塞します。

- **hold オペランド (SPP, MHP で有効)**

UAP が異常終了した場合、すぐに閉塞するか、時間監視して閉塞するかを指定します。すぐに閉塞する場合は Y を、時間監視して閉塞する場合は N を指定します。

- **service_hold オペランド (SPP で有効)**

閉塞する単位をサービスグループ単位にするかサービス単位にするかを指定します。サービスグループ単位にする場合は N を、サービス単位にする場合は Y を指定します。

- **term_watch_time オペランド (SPP, MHP で有効)**

時間監視して閉塞する場合の、サービスグループ単位の閉塞の監視時間を指定します。term_watch_time オペランドの値は、service_term_watch_time オペランドを指定した場合は無視されます。

- **service_term_watch_time オペランド (SPP で有効)**

時間監視して閉塞する場合の、サービス単位の閉塞の監視時間を指定します。service_term_watch_time オペランドを指定した場合、term_watch_time オペランドの値は無視されます。

閉塞に関するオペランドは、それぞれに指定した値の組み合わせによって、UAP が異常終了したあとに再起動するかまたは閉塞するかが決まります。オペランドの組み合わせと閉塞の有無の関係については、それぞれのオペランドの説明を参照してください。

付録 B 共用メモリの見積もり式

TP1/Server Base 用および MCF サービス用の共用メモリの見積もり式について説明します。

付録 B.1 TP1/Server Base 用の共用メモリの見積もり式

TP1/Server Base 用の共用メモリの見積もり式について説明します。ここで説明する見積もり式の単位はバイトです。

(1) 静的共用メモリの算出式（アーカイブジャーナルノードでないノードの場合）

システム環境定義の `static_shmpool_size` オペランドに指定する静的共用メモリの算出式について説明します。ここで説明する算出式に従って静的共用メモリを算出してください。

(a) システムマネージャ

■ 32 ビットの場合

$$128 \times (\text{システム環境定義で指定した最大サーバ数} + 3) + 6024 + \text{dcstats コマンドで統計情報のジャーナル出力を指定するユーザサーバの最大数} \times 1024$$

■ 64 ビットの場合

$$128 \times (\text{システム環境定義で指定した最大サーバ数} + 3) + 8000 + \text{dcstats コマンドで統計情報のジャーナル出力を指定するユーザサーバの最大数} \times 1280$$

(b) プロセスサーバ

$$944 \times \text{prc\_process\_count オペランドの指定値} + 60624$$

(c) タイマサーバ

$$32 \times \text{tim\_watch\_count オペランドの指定値} + 1440$$

(d) スケジューラ

$$\begin{aligned} & 48160 + ((\text{scd\_server\_count オペランドの指定値} + 3) \times 1216) \\ & + (\text{scd\_hold\_recovery\_count オペランドの指定値} \times 160) \\ & + \sum_{i=1}^i (\text{スケジューラバッファグループ } i \text{ の共用メッセージ格納バッファプールサイズ} + 320) \\ & + \sum_{j=1}^j (\text{サービスグループ } j \text{ の message\_store\_buflen オペランドの指定値} + 128) \\ & + \sum_{k=1}^k (\text{サービスグループ } k \text{ の サービス数} \times 64 + 192) + (\text{サービスグループ } L \times 128) \\ & + (\text{マルチスケジューラグループ } M \times 128) + (\text{マルチスケジューラデーモン } N \times 128) \\ & + \sum_{o=1}^o (\text{サービスグループ } o \text{ の サービス数} \times 128 + 64) \end{aligned}$$

(凡例)

i: スケジューラバッファグループの定義数

j: scdbufgrp 定義コマンドを指定しないサービスグループの定義数

k: ユーザサービス定義の service_hold オペランドに Y を指定したサービスグループの定義数

L: ユーザサービス定義の schedule_method オペランドに namedpipe を指定したサービスグループの定義数

M: マルチスケジューラグループ数

N: マルチスケジューラデーモン数

o: ユーザサービス定義に scdsvcdef 定義コマンドを指定したサービスグループの定義数 (キュー受信型サーバの SPP が対象になります)

(e) ロックサーバ

■ 32 ビットの場合

$$\begin{aligned} & (\text{lck\_limit\_foruser オペランドの指定値} + \text{lck\_limit\_fordam オペランドの指定値} \\ & + \text{lck\_limit\_fortam オペランドの指定値} + \text{lck\_limit\_formqa オペランドの指定値}) \\ & \times 544 + 62016 + 128 \end{aligned}$$

■ 64 ビットの場合

$$\begin{aligned} & (\text{lck\_limit\_foruser オペランドの指定値} + \text{lck\_limit\_fordam オペランドの指定値} \\ & + \text{lck\_limit\_fortam オペランドの指定値} + \text{lck\_limit\_formqa オペランドの指定値}) \\ & \times 544 + 62432 + 128 \end{aligned}$$

(f) トランザクションマネージャ

■ 32 ビットの場合

$$\begin{aligned} & 2048 + 1280 \times (\text{RM 名} + \text{RM 拡張子}) \text{ 数} + (816 + 128 \times (\text{RM 名} + \text{RM 拡張子}) \text{ 数} \\ & + 128 \times \text{trn\_max\_subordinate\_count オペランドの指定値} \\ & + \downarrow (63 + 4 \times (\text{RM 名} + \text{RM 拡張子}) \text{ 数}) / 32 \downarrow \times 32) \\ & \times \text{trn\_tran\_process\_count オペランドの指定値} \end{aligned}$$

$$+ (C \times (\text{trn\_max\_crm\_subordinate\_count オペランドの指定値} + 1) \\ \times \text{trn\_tran\_process\_count オペランドの指定値} \times 1216) + 128$$

(凡例)

↓↓：小数点以下を切り捨てます。

C：trn_crm_use オペランドに Y を指定した場合は 1，N を指定した場合は 0 になります。

■ 64 ビットの場合

$$2048 + 1280 \times (\text{RM 名} + \text{RM 拡張子}) \text{ 数} + (816 + 128 \times (\text{RM 名} + \text{RM 拡張子}) \text{ 数} \\ + 128 \times \text{trn\_max\_subordinate\_count オペランドの指定値} \\ + \downarrow (63 + 4 \times (\text{RM 名} + \text{RM 拡張子}) \text{ 数}) / 32 \downarrow \times 32) \\ \times \text{trn\_tran\_process\_count オペランドの指定値} \\ + (C \times (\text{trn\_max\_crm\_subordinate\_count オペランドの指定値} + 1) \\ \times \text{trn\_tran\_process\_count オペランドの指定値} \times 1248) + 128$$

(凡例)

↓↓：小数点以下を切り捨てます。

C：trn_crm_use オペランドに Y を指定した場合は 1，N を指定した場合は 0 になります。

(g) ジャーナルサーバ

■ アーカイブジャーナルノードにはアーカイブされないノードの場合

$$4864 + 256 \times \text{jnladdfg 定義コマンドの定義数} \\ + 480 \times m \times \text{jnladdfg 定義コマンドの定義数} \\ + 64 \times m \\ + 128 \times (m \times 2 + 1) \\ + 4096 \times m \times i \\ + \uparrow (128 \times (34 + 2 \times \text{jnladdfg 定義コマンドの定義数} \times 2 \times m) / 8064) \uparrow \times 8192 \\ + \uparrow (\text{jnl\_max\_datasize オペランドの指定値} + 336) / 4096 \uparrow \times 4096 \times (m \times 2 + 1) \\ + \uparrow (\text{jnl\_max\_datasize オペランドの指定値} + 336) / 4096 \uparrow \times 4096 \times 16 \times m \\ + \uparrow (388 + 192 \times m) / 4096 \uparrow \times 4096 \\ + n$$

(凡例)

↑↑：小数点以下を切り上げます。

i：システムジャーナルサービス定義の jnl_dual オペランドに Y を指定した場合は 2，N を指定した場合は 1 になります。

m：システムジャーナルサービス定義の jnl_max_file_dispersion オペランドの指定値です。

n：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 を指定した場合は 128，N を指定した場合は 0 になります。

■ アーカイブジャーナルノードによってアーカイブされるノードの場合

$$4864 + 256 \times \text{jnladdfg 定義コマンドの定義数} \\ + 480 \times m \times \text{jnladdfg 定義コマンドの定義数} \\ + 64 \times m \\ + 128 \times (m \times 2 + 1)$$

$$\begin{aligned}
&+4096 \times m \times i \\
&+ \uparrow (128 \times (34 + 2 \times \text{jnladdfg 定義コマンドの定義数} \times 2 \times m) / 8064) \uparrow \times 8192 \\
&+ \uparrow (\text{jnl\_max\_datasize オペランドの指定値} + 336) / 4096 \uparrow \times 4096 \times (m \times 2 + 1) \\
&+ \uparrow (\text{jnl\_max\_datasize オペランドの指定値} + 336) / 4096 \uparrow \times 4096 \times 16 \times m \\
&+ \uparrow (388 + 192 \times m) / 4096 \uparrow \times 4096 \\
&+ n \\
&+ 512 \\
&+ 128 \times \downarrow (j \times 1024 \times 1024 / (k \times 1024)) \downarrow \\
&+ k \times 1024 \times \downarrow (j \times 1024 \times 1024 / (k \times 1024)) \downarrow
\end{aligned}$$

(凡例)

↑↑：小数点以下を切り上げます。

↓↓：小数点以下を切り捨てます。

i：システムジャーナルサービス定義の jnl_dual オペランドに Y を指定した場合は 2，N を指定した場合は 1 になります。

m：システムジャーナルサービス定義の jnl_max_file_dispersion オペランドの指定値です。

n：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 を指定した場合は 128，N を指定した場合は 1 になります。

j：jnl_arc_buff_size オペランドに指定した値です。

k：jnl_arc_max_datasize オペランドに指定した値です。

■ ジャーナルファイルレス機能を使用するノードの場合

共用メモリの所要量は、10496 バイトです。

(h) チェックポイントダンプ

■ 32 ビットの場合

$$\begin{aligned}
&17872 + \text{チェックポイントダンプサービス定義の定義数} \times 1024 \\
&+ (\text{チェックポイントダンプサービス定義の jnladdfg オペランドの定義数}) \times 400
\end{aligned}$$

■ 64 ビットの場合

$$\begin{aligned}
&18640 + \text{チェックポイントダンプサービス定義の定義数} \times 1024 \\
&+ (\text{チェックポイントダンプサービス定義の jnladdfg オペランドの定義数}) \times 400
\end{aligned}$$

■ ジャーナルファイルレス機能を使用するノードの場合

共用メモリの所要量は、0 バイトです。

(i) サーバリカバリジャーナル

■ 32 ビットの場合

$$1376 + 6272 \times n + (\uparrow (a + 336) / 4096 \uparrow + 1) \times 4096 \times n$$

(凡例)

↑↑：小数点以下を切り上げます。

n：TP1/Message Queue を使用しない場合は，OpenTP1 が提供する RM 数+ 2 になります。TP1/Message Queue を使用する場合は，内部 RM 数+ 3 になります。

a：システムジャーナルサービス定義で指定する最大レコードデータ長になります。

■ 64 ビットの場合

$$4448 + 6856 \times n + (\uparrow (a + 336) / 4096 \uparrow + 1) \times 4096 \times n$$

(凡例)

↑↑：小数点以下を切り上げます。

n：TP1/Message Queue を使用しない場合は，OpenTP1 が提供する RM 数+ 2 になります。TP1/Message Queue を使用する場合は，内部 RM 数+ 3 になります。

a：システムジャーナルサービス定義で指定する最大レコードデータ長になります。

(j) トランザクションジャーナル

■ ジャーナルファイルレス機能を使用しないノードの場合

$$\begin{aligned} &13472 + 1200 \times \text{trn\_tran\_process\_count オペランドの指定値} \\ &+ 64 \times (\text{trn\_tran\_process\_count オペランドの指定値} \\ &\quad \times \text{trn\_max\_subordinate\_count オペランドの指定値}) \\ &+ (\uparrow (J + 336) / 4096 \uparrow) \times 4096 \\ &+ \text{DAM} + \text{TAM} + \text{MCF} + \text{MQ} \end{aligned}$$

(凡例)

↑↑

小数点以下を切り上げます。

J

システムジャーナルサービス定義で指定した jnl_max_datasize オペランドの指定値になります。

DAM

TP1/FS/Direct Access を使用しない場合は，0 になります。

TP1/FS/Direct Access を使用する場合は，次の算出式で求めた値になります。

$$\begin{aligned} &128 + 128 \times \text{trn\_tran\_process\_count オペランドの指定値} \\ &+ (4160 + (\uparrow (J + 336) / 4096 \uparrow \times 4096)) \\ &\times (\text{trn\_recovery\_process\_count オペランドの指定値}) \\ &+ 56 \times (\text{trn\_tran\_process\_count オペランドの指定値} \times 2 + 2) \end{aligned}$$

TAM

TP1/FS/Table Access を使用しない場合は，0 になります。

TP1/FS/Table Access を使用する場合は，次の算出式で求めた値になります。

$4288 + 240 \times \text{trn\_tran\_process\_count}$ オペランドの指定値
+ ($\uparrow (J+336) / 4096 \uparrow$) $\times 4096$

MCF

TP1/Message Control を使用しない場合は、0 になります。

TP1/Message Control を使用する場合は、次の算出式で求めた値になります。

$128 + 240 \times \text{trn\_tran\_process\_count}$ オペランドの指定値
+ ($4160 + (\uparrow (J+336) / 4096 \uparrow \times 4096)$)
 $\times (\text{trn\_recovery\_process\_count}$ オペランドの指定値)

MQ

TP1/Message Queue を使用しない場合は、0 になります。

TP1/Message Queue を使用する場合は、次の算出式で求めた値になります。

$4288 + 240 \times \text{trn\_tran\_process\_count}$ オペランドの指定値
+ ($\uparrow (J+336) / 4096 \uparrow$) $\times 4096$

■ ジャーナルファイルレス機能を使用するノードの場合

共用メモリの所要量は、0 バイトです。

(k) ステータスサーバ

共用メモリの所要量は、64 バイトです。

(l) ネームサーバ

(name_total_size オペランドの指定値 + name_cache_size オペランドの指定値)
 $\times 1024$

name_total_size 、および name_cache_size の算出式を次に示します。

■ ノード自動追加機能を使用しない場合

$\text{name\_total\_size} = \uparrow (7616 + (\text{all\_node}$ のノード数 $\times 384$
+ all_node_ex のノード数 $\times 284$
+ $\text{all\_node\_extend\_number}$ オペランドの指定値 $\times 384$
+ $\text{all\_node\_ex\_extend\_number}$ オペランドの指定値 $\times 284$)
+ ((SPP 数^{※1} + rap サーバ数^{※2} + XATMI 数^{※3}) $\times 264$)
+ MCF ^{※4} + DAM ^{※5} + TAM ^{※6} + CLT ^{※7}) $/ 1024 \uparrow$

$\text{name\_cache\_size} = \uparrow (\text{all\_node}$ オペランドおよび all_node_ex オペランド
に指定したノード^{※8}で起動する SPP 数^{※1}の総計 $\times 224$) $/ 1024 \uparrow$

(凡例)

$\uparrow \uparrow$: 小数点以下を切り上げます。

注※1

サービスグループの数になります。

注※2

rap サーバを起動する場合は、rap サービス数を指定します。

注※3

XATMI-API を使用する場合は、XATMI の SPP で定義したサービスの総数になります。

注※4

TP1/Message Control を使用する場合は、次の算出式で求めた値になります。

$$(1 + \text{MCF通信サーバの数} + \text{MHP数}) \times 264$$

注※5

TP1/FS/Direct Access を使用する場合は、936 になります。

注※6

TP1/FS/Table Access を使用する場合は、1424 になります。

注※7

クライアント拡張サービス機能を使用する場合は、936 になります。

注※8

システム共通定義の all_node オペランドおよび all_node_ex オペランドに指定したノードか、またはドメイン定義ファイルに指定したノードです。

■ ノード自動追加機能を使用する場合

```
name_total_size= ↑ (7936 + (all_node_exのノード数 × 284)
                    + all_node_ex_extend_numberオペランドの指定値 × 284
                    + 163840
                    + ((SPP数※1 + rapサーバ数※2 + XATMI数※3) × 264)
                    + MCF※4 + DAM※5 + TAM※6 + CLT※7) / 1024 ↑

name_cache_size= ↑ (ノードリストに登録されるノード
                    およびall_node_exオペランドに指定したノード※8で起動する
                    SPP数※1の総計 × 224) / 1024 ↑
```

(凡例)

↑ ↑ : 小数点以下を切り上げます。

注※1

サービスグループの数になります。

注※2

rap サーバを起動する場合は、rap サービス数を指定します。

注※3

XATMI-API を使用する場合は、XATMI の SPP で定義したサービスの総数になります。

注※4

TP1/Message Control を使用する場合は、次の算出式で求めた値になります。

$$(1 + \text{MCF通信サーバの数} + \text{MHP数}) \times 264$$

注※5

TP1/FS/Direct Access を使用する場合は、936 になります。

注※6

TP1/FS/Table Access を使用する場合は、1424 になります。

注※7

クライアント拡張サービス機能を使用する場合は、936 になります。

注※8

システム共通定義の all_node オペランドおよび all_node_ex オペランドに指定したノードか、またはドメイン定義ファイルに指定したノードです。

(m) キューサーバ

ディスクキューを使用する場合、次に示す算出式に従って算出します（メモリキューだけの構成のときは見積もり不要です）。

■ 32 ビットの場合

$$\begin{aligned}
 & 384 + \uparrow (169 \times (\text{mcfmexp 定義コマンドの -g オプションの指定値} \\
 & + \text{mcfmexp 定義コマンドの -l オプションの指定値} \times 2 + \text{mcfmcname 定義コマンドの定義数} \times 5)) / 16 \uparrow \times 16 \\
 & + (272 \times \text{que\_xidnum オペランドの指定値}) \\
 & + \uparrow (185 \times \text{quegrp 定義コマンドの定義数}) / 8 \uparrow \times 8 \\
 & + \uparrow (161 \times (\text{mcfmexp 定義コマンドの -g オプションの指定値} \\
 & + \text{mcfmexp 定義コマンドの -l オプションの指定値})) / 16 \uparrow \times 16 \\
 & + (32 \times \text{全物理ファイルのレコード数}^{※1}) \\
 & + \sum_{i=1}^n (\uparrow (704 + 20 \times \text{物理ファイルのレコード数}^{※2}) \\
 & + 4 \times (2 \times \text{quegrp 定義コマンドの -n オプションの指定値} + 1) \\
 & + (\text{物理ファイルのレコード長}^{※3} + 16) \\
 & \times \text{quegrp 定義コマンドの -n オプションの指定値}) / 32 \uparrow \times 32) \\
 & + 96 + 4192
 \end{aligned}$$

(凡例)

↑↑：小数点以下を切り上げます。

n：メッセージキューサービス定義の quegrp 定義コマンドの定義数

注※1

quegrp 定義コマンドで定義したそれぞれの物理ファイルのレコード数の合計値。物理ファイルのレコード数は、queinit コマンドの -n オプションで指定します。

注※2

queinit コマンドの -n オプションに指定したレコード数。

注※3

queinit コマンドの -s オプションに指定したレコード長。

■ 64 ビットの場合

$$\begin{aligned} & 480 + \uparrow (249 \times (\text{mcfmexp 定義コマンドの } -g \text{ オプションの指定値} \\ & + \text{mcfmexp 定義コマンドの } -l \text{ オプションの指定値} \times 2 + \text{mcfmcname 定義コマンドの定義数} \times 5)) / 16 \uparrow \times 16 \\ & + (352 \times \text{que\_xidnum オペランドの指定値}) \\ & + \uparrow (217 \times \text{quegrp 定義コマンドの定義数}) / 8 \uparrow \times 8 \\ & + \uparrow (177 \times (\text{mcfmexp 定義コマンドの } -g \text{ オプションの指定値} \\ & + \text{mcfmexp 定義コマンドの } -l \text{ オプションの指定値})) \\ & / 16 \uparrow \times 16 \\ & + (32 \times \text{全物理ファイルのレコード数}^{※1}) \\ & + \sum_{i=1}^n (\uparrow (720 + 20 \times \text{物理ファイルのレコード数}^{※2} \\ & + 4 \times (2 \times \text{quegrp 定義コマンドの } -n \text{ オプションの指定値} + 1) \\ & + (\text{物理ファイルのレコード長}^{※3} + 16) \\ & \times \text{quegrp 定義コマンドの } -n \text{ オプションの指定値}) / 32 \uparrow \times 32) \\ & + 96 + 4208 \end{aligned}$$

(凡例)

$\uparrow \uparrow$: 小数点以下を切り上げます。

n : メッセージキューサービス定義の quegrp 定義コマンドの定義数

注※1

quegrp 定義コマンドで定義したそれぞれの物理ファイルのレコード数の合計値。物理ファイルのレコード数は、queinit コマンドの $-n$ オプションで指定します。

注※2

queinit コマンドの $-n$ オプションに指定したレコード数。

注※3

queinit コマンドの $-s$ オプションに指定したレコード長。

(n) 性能検証用トレース取得サービス

■ prf_trace オペランドに N を指定した場合

共用メモリの所要量は、1024 バイトです。

■ prf_trace オペランドに Y を指定した場合

共用メモリの所要量は、1553440 + prf_buff_size オペランドの指定値 $\times$ 1024 バイトです。

(o) rap リスナーおよび rap サーバ

$1024 + 148 \times A + \uparrow (A / 8) \uparrow + 260 \times B$

(凡例)

$\uparrow \uparrow$: 小数点以下を切り上げます。

A : rap_parallel_server オペランドの指定値になります。

B : rap_max_client オペランドの指定値になります。

(p) XA リソースサービス

$672 + A \times \text{trn\_tran\_process\_count}$ オペランドの指定値

(凡例)

A: xar_msdtc_use オペランドに N を指定するか、または指定を省略した場合は、288 です。

xar_msdtc_use オペランドに Y を指定した場合は、xarfile 定義コマンドで指定した XAR ファイルのレコード長です。

(q) DAM

■ 32 ビットの場合

$288 \times (\text{damfile 定義コマンドの定義数} + \text{dam\_added\_file オペランドの指定値}) + 288 + 512$

■ 64 ビットの場合

$320 \times (\text{damfile 定義コマンドの定義数} + \text{dam\_added\_file オペランドの指定値}) + 304 + 512$

(r) IST

$2336 + (48 \times \text{IST サービス定義に定義したテーブル数}) + 512$

(s) メッセージキュー (Windows 版)

$16 + (72 + 12 \times n1) \times n2$

(凡例)

n1: すべてのユーザーサービス定義の中で次の算出式が最大となる値です。

$\uparrow \text{message\_store\_buflen オペランドの指定値} / \text{message\_cell\_size オペランドの指定値} \uparrow$

rap リスナーサービス定義の場合は、rap_parallel_server オペランドの指定値が 3 以下のとき、8 になります。

rap_parallel_server オペランドの指定値が 4 以上のときは次の算出式になります。

$\uparrow \text{rap\_parallel\_server オペランドの指定値} \times 2.1 \uparrow$

n2: スケジュールサービス定義の scd_server_count オペランドの指定値 + 5 になります。

↑↑: 小数点以下を切り上げます。

(t) リアルタイム統計情報サービス

共用メモリの所要量は、64 バイトです。

(u) クライアント拡張サービス

M×N

(凡例)

M：クライアントサービス定義の message_store_buflen オペランドの指定値
N：トランザクション RPC 実行プロセス，および CUP 実行プロセスをあわせて使用する場合は 2 です。トランザクション RPC 実行プロセス，または CUP 実行プロセスのどちらかを使用する場合は 1 です。

(v) XATMI 通信サービス

共用メモリの所要量は，32 バイトです。

(2) 静的共用メモリの算出式（アーカイブジャーナルノードの場合）

システム環境定義の static_shmpool_size オペランドに指定する静的共用メモリの算出式について説明します。ここで説明する算出式に従って静的共用メモリを算出してください。

(a) システムマネージャ

128×（システム環境定義で指定した最大サーバ数+3）+5487
+dcstatsコマンドで統計情報のジャーナル出力を指定するユーザサーバの最大数×1024

(b) プロセスサーバ

共用メモリの所要量は，251648 バイトです。

(c) タイマサーバ

32×tim_watch_countオペランドの指定値+1440

(d) ジャーナルサーバ

2304+ $\sum_{i=1}^r$ (10880
+ (256+480×m) × リソースiで指定したjnladdfg定義コマンドの定義数
+ (64+4096×d) × m+ (4177920×m)
+1044608 × リソースiに接続されるサーバ数)
+ (r+ $\sum_{i=1}^r$ (↑ (128× ((1+2×m) × リソースiで指定したjnladdfg定義コマンドの定義数+5)
⌈8064)) ↑)
×8192

(凡例)

↑↑：小数点以下を切り上げます。

r：グローバルアーカイブジャーナルサービス定義の jnldfsv -a で指定したリソースグループ数になります。

d：リソースグループ i の二重化係数になります。

- ・ jnl_dual オペランドが N の場合：1
- ・ jnl_dual オペランドが Y の場合：2

m：リソース i で指定した jnl_max_file_dispersion オペランドの指定値になります。

(e) ネームサーバ

■ ノード自動追加機能を使用しない場合

$$(\text{name\_total\_size オペランドの指定値} + \text{name\_cache\_size オペランドの指定値}) \times 1024$$

name_total_size, および name_cache_size の算出式を次に示します。

$$\begin{aligned} \text{name\_total\_size} &= \uparrow (7616 + \text{all\_node オペランドのノード数} \times 384 \\ &\quad + \text{all\_node\_ex オペランドのノード数} \times 284 \\ &\quad + \text{all\_node\_extend\_number オペランドの指定値} \times 384 \\ &\quad + \text{all\_node\_ex\_extend\_number オペランドの指定値} \times 284) \\ &\quad \nearrow 1024 \uparrow \\ \text{name\_cache\_size} &= \uparrow (\text{該当アーカイブジャーナルノードに接続する被アーカイブ} \\ &\quad \text{ジャーナルノード数} \times 224) \nearrow 1024 \uparrow \end{aligned}$$

■ ノード自動追加機能を使用する場合

$$(\text{name\_total\_size オペランドの指定値} + \text{name\_cache\_size オペランドの指定値}) \times 1024$$

name_total_size, および name_cache_size の算出式を次に示します。

$$\begin{aligned} \text{name\_total\_size} &= \uparrow (7936 + \text{all\_node\_ex オペランドのノード数} \times 284 \\ &\quad + \text{all\_node\_ex\_extend\_number オペランドの指定値} \times 284 \\ &\quad + 163840) \nearrow 1024 \uparrow \\ \text{name\_cache\_size} &= \uparrow (\text{該当アーカイブジャーナルノードに接続する被アーカイブ} \\ &\quad \text{ジャーナルノード数} \times 224) \nearrow 1024 \uparrow \end{aligned}$$

(凡例)

↑ ↑：小数点以下を切り上げます。

(f) 性能検証用トレース取得サービス

■ prf_trace オペランドに N を指定した場合

共用メモリの所要量は、99328 バイトです。

■ prf_trace オペランドに Y を指定した場合

共用メモリの所要量は、1553440 + prf_buff_size オペランドの指定値 × 1024 バイトです。

(3) 動的共用メモリの算出式

システム環境定義の `dynamic_shmpool_size` オペランドに指定する動的共用メモリの算出式について説明します。ここで説明する算出式に従って動的共用メモリを算出してください。

なお、ここで説明する算出式は、アーカイブジャーナルノードでないノードの場合です。アーカイブジャーナルノードの場合は動的共用メモリの値に 0 を指定してください。

(a) トランザクションジャーナル

■ ジャーナルファイルレス機能を使用しないノードの場合

$$\left(\left(1 + \left(\uparrow \left(\text{システムジャーナルサービス定義で指定した } jnl\_max\_datasize \text{ オペランドの指定値} + 512 \right) / 4096 \uparrow \right) \right) \times 4096 \right) + 32 \times \text{trn\_tran\_process\_count オペランドの指定値}$$

(凡例)

↑↑：小数点以下を切り上げます。

このノードに、`trf_put=Y` を指定したユーザサーバがない場合、共用メモリの所要量は、0 バイトです。

■ ジャーナルファイルレス機能を使用するノードの場合

共用メモリの所要量は、0 バイトです。

(b) キューサーバ

ディスクキューを使用する場合、次に示す算出式に従って算出します（メモリキューだけの構成のときは見積もり不要です）。

■ 32 ビットの場合

$$\uparrow \left(\text{複数のトランザクションで同期点処理までに、同時に要求するメッセージ数} \times 48 + 32 \right) / 8192 \uparrow \times (8192 + 512)$$

(凡例)

↑↑：小数点以下を切り上げます。

■ 64 ビットの場合

$$\uparrow \left(\text{複数のトランザクションで同期点処理までに、同時に要求するメッセージ数} \times 64 + 56 \right) / 8192 \uparrow \times (8192 + 512)$$

(凡例)

↑↑：小数点以下を切り上げます。

(c) DAM

■ 32 ビットの場合

トランザクションアクセス種別がブランチアクセスの場合

$$(484 \times \uparrow 1 \text{ トランザクションブランチでオープンしているファイル数} / 10 \uparrow + 72 \times \text{dam\_update\_block オペランドの指定値} + 692) \times \text{同時実行トランザクションブランチ数}$$

トランザクションアクセス種別がグローバルアクセスの場合

$$(484 \times \uparrow 1 \text{ トランザクションブランチでオープンしているファイル数} / 10 \uparrow + 72 \times \text{dam\_update\_block オペランドの指定値} + 692) \times \text{同時実行トランザクションブランチ数} \\ + (128 + 52 \times \uparrow 1 \text{ グローバルトランザクションで発生するトランザクションブランチ数} / 10 \uparrow) \times \text{同時実行グローバルトランザクション数}$$

(凡例)

↑↑：小数点以下を切り上げます。

■ 64 ビットの場合

トランザクションアクセス種別がブランチアクセスの場合

$$(488 \times \uparrow 1 \text{ トランザクションブランチでオープンしているファイル数} / 10 \uparrow + 72 \times \text{dam\_update\_block オペランドの指定値} + 728) \times \text{同時実行トランザクションブランチ数}$$

トランザクションアクセス種別がグローバルアクセスの場合

$$(488 \times \uparrow 1 \text{ トランザクションブランチでオープンしているファイル数} / 10 \uparrow + 72 \times \text{dam\_update\_block オペランドの指定値} + 728) \times \text{同時実行トランザクションブランチ数} \\ + (160 + 96 \times \uparrow 1 \text{ グローバルトランザクションで発生するトランザクションブランチ数} / 10 \uparrow) \times \text{同時実行グローバルトランザクション数}$$

(凡例)

↑↑：小数点以下を切り上げます。

(d) TAM

■ 32 ビットの場合

$$\uparrow (128 + 32 \times \text{tam\_max\_tblnum オペランドの指定値} + 24) / 512 \uparrow \times 512$$

■ 64 ビットの場合

$$\uparrow (192 + 48 \times \text{tam\_max\_tbl\_num オペランドの指定値} + 40) / 512 \uparrow \times 512$$

(凡例)

↑↑：小数点以下を切り上げます。

(4) DAM サービス用の共用メモリの算出式

DAM ブロックを退避するためのバッファ領域用の共用メモリの算出式について説明します。次に示す D1 または D2 のうち、どちらか大きい値に 1024 を加えた値が算出結果になります。

D1=DAMサービス定義のdam_cache_sizeオペランドで指定したバッファ領域長
D2=A+B

(凡例)

A : $(Mb/64 + 1) \times 128 \times n \times Tr$

B : $\{\downarrow (A/576) \downarrow\} \times 32 + 64$

↓↓ : 小数点以下を切り捨てます。

Mb : DAM サービス定義で定義した論理ファイルの中で最大ブロック長を持つファイルのブロック長 + 8 (セクタ長の倍数)

n : DAM サービス定義の dam_update_block オペランドで指定した最大更新ブロック数

Tr : DAM サービス定義の dam_tran_process_count オペランドで定義したトランザクションブランチ数

DAM サービス用の共用メモリとして確保されるバッファ領域長を DAM サービス定義の dam_cache_size オペランドまたは dam_cache_size_fix オペランドに指定した場合、自動的にバイトに換算されます。そのときに、1 キロバイト当たり 1000 バイトで換算するか、1024 バイトで換算するかを dam_kb_size オペランドで指定できます。詳細については、DAM サービス定義の [dam_kb_size](#) オペランドを参照してください。

(5) TAM サービス用の共用メモリの算出式

オンライン中に使用する TAM テーブルはすべて共用メモリ上にロードされるため、共用メモリ所要量は TAM ファイルの容量に依存します。TAM テーブル用の共用メモリは次に示す算出式に従って算出してください。

(a) 32 ビットの場合

$2000 + Fd \times Fe + Fg + Fc \times 320$

$+ \left(\sum_{i=1}^n (Fs + 32) + (Fc - n) \times (Fm + 32) \right) \times 2 < \text{MIN (2ギガバイト, 動作環境上の制限値)}$

(b) 64 ビットの場合

$2000 + Fd \times Fe + Fg + Fc \times 336$

$+ \left(\sum_{i=1}^n (Fs + 32) + (Fc - n) \times (Fm + 32) \right) \times 2 < \text{MIN (2ギガバイト, 動作環境上の制限値)}$

(凡例)

n : TAM サービス定義で指定した tamtable 定義コマンドの数 (tamtable 数)

Fa : TAM サービス定義で定義したトランザクション内最大アクセステーブル数 (tam_max_trnfilnum)
 Fb : TAM サービス定義で定義した同時走行最大トランザクションブランチ数 (tam_max_trnnum)
 Fc : TAM サービス定義で定義したオンライン中に使用する TAM ファイルの最大数 (tam_max_tblnum)
 $Fd : \uparrow (320 + 256 \times Fb) \div Fe \uparrow \times 2 + Fb \times 2$
 Fe : $(72 + 128 \times Fa)$ を 32 の整数倍に切り上げた値
 Fg : $(210 + Fd \times 4)$ を 16 の整数倍に切り上げた値
 Fm : TAM サービス定義で定義したオンライン中に使用する TAM ファイルの最大容量 (tam_max_filesize) を 16 の整数倍に切り上げた値
 Fs : tamtable 定義コマンドで指定した TAM ファイル個々のファイル容量 (TAM ファイルの容量は、マニュアル「OpenTP1 運用と操作」を参照)

TAM テーブル用の共用メモリ量は、使用する TAM ファイルの合計サイズの 2 倍以上必要です。なお、TAM テーブル用の共用メモリとして確保できるサイズは、全体として 2 ギガバイト未満です。ただし、2 ギガバイト未満であっても、OS の制限値など動作環境によっては確保できない場合があります。

(6) IST サービス用の共用メモリの算出式

IST テーブル用の共用メモリは次に示す算出式に従って算出してください。

$$(S + 8280) \times 2 + 176$$

(凡例)

$$S : \left(\sum_{i=1}^n ((L + 16) \times N) + 88 \right) \div 64 \times 64$$

n : IST サービス定義に定義したテーブル数

L : 各テーブルのレコード長

N : 各テーブルのレコード数

(7) リアルタイム統計情報サービス用の共用メモリの算出式

リアルタイム統計情報サービス用の共用メモリは次に示す算出式に従って算出してください。

$$304 + (\uparrow (4 \times rts\_service\_max \text{ オペランドの指定値} \div 8 \uparrow \times 8) + (104 \times (rts\_service\_max \text{ オペランドの指定値} + 1))) + (144 \times rts\_item\_max \text{ オペランドの指定値} \times (rts\_service\_max \text{ オペランドの指定値} + 1))$$

(8) 共用メモリプールのサイズの算出式

共用メモリプールのサイズは次に示す算出式に従って算出してください。

$$\begin{aligned} & \text{OpenTP1 制御用共用メモリ (約 10240 キロバイト)} \\ & + \text{システム環境定義の static\_shm\_pool\_size オペランドの指定値} \\ & + \text{システム環境定義の dynamic\_shm\_pool\_size オペランドの指定値} \end{aligned}$$

付録 B.2 MCF サービス用の共用メモリの見積もり式

MCF サービス用の共用メモリの算出式について説明します。ここで説明するサイズの単位はバイトです。

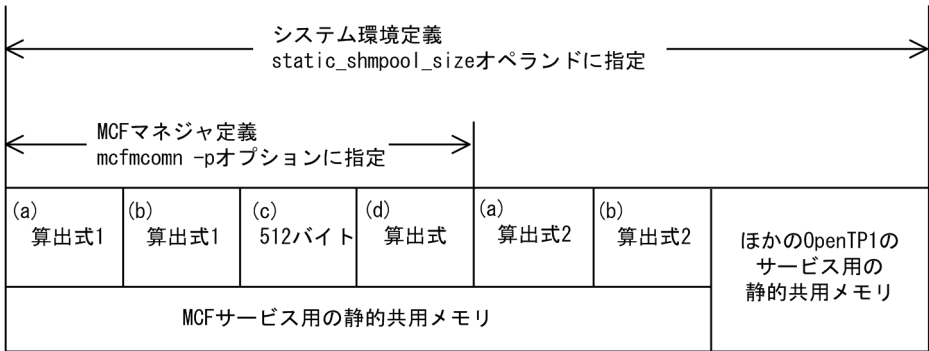
(1) 静的共用メモリ

ここで説明する算出結果の合計値が MCF サービスで使用する静的共用メモリのサイズです。

ほかの OpenTP1 のサービスが使用する静的共用メモリに MCF サービスで使用する静的共用メモリを加え、システム環境定義の static_shmpool_size オペランドに指定してください。また、「付録 B.2(1)(a) MCF マネジャプロセス」の算出式 2、および「付録 B.2(1)(b) MCF 通信プロセスおよびアプリケーション起動プロセス」の算出式 2 を除いた算出結果の合計値を MCF マネジャ定義の mcfmcomn 定義コマンドの -p オプションに指定してください。

MCF サービス用の静的共用メモリの算出式と定義指定値の関係を、次に示します。

図 B-1 MCF サービス用の静的共用メモリの算出式と定義指定値の関係



以降、図中の(a)から(d)について説明します。

(a) MCF マネジャプロセス

MCF マネジャプロセスで使用する共用メモリのサイズは、それぞれ次に示す二つの算出式の結果の合計値です。MCF マネジャプロセスは、OpenTP1 システム内で一つのプロセスを使用します。

算出式 1 :

- 32 ビットの場合

21000+2000×A+600×B+400×C+1000×D+400×E+400×F+500×G+400×H+400×I

- 64 ビットの場合

29000+2800×A+800×B+400×C+1300×D+500×E+500×F+700×G+400×H+500×I

(凡例)

- A : MCF 通信プロセスおよびアプリケーション起動プロセス数 (mcfmcname 定義コマンドの指定数)
B : 論理端末総数 (すべての mcftalcle 定義コマンドの指定数)

- C：アプリケーション総数（すべての mcfaalcap 定義コマンドの指定数）
D：MHP サービスグループ数
E：MHP サービス総数（各 MHP サービスグループ内のサービス数の合計）
F：キューグループ数（mcfmqgid 定義コマンドの指定数）
G：メッセージ送信を行う UAP プロセス数
H：MCF 構成変更再開始機能を使用しない場合は，0 になります。MCF 構成変更再開始機能を使用する場合は，1 になります。
I：同時に実行するタイマ起動によるアプリケーション起動要求数

算出式 2：

106000+200×A+B

(凡例)

- A：メッセージ出力通番使用論理端末数（mcfmcomn 定義コマンドの-n オプションの指定値）
B：MCF メッセージ回復用作業領域長（mcfmcomn 定義コマンドの-l オプションの指定値）

(b) MCF 通信プロセスおよびアプリケーション起動プロセス

MCF 通信プロセスおよびアプリケーション起動プロセスで使用する共用メモリのサイズは，次に示す二つの算出式の結果の合計値です。MCF 通信プロセスおよびアプリケーション起動プロセスは，OpenTP1 システム内で mcfmcname 定義コマンドの定義数分のプロセスを使用します。

算出式 1：

- 32 ビットの場合

4000+A×B+C×D+800×E+800×F+（100+G）×H+600×I

- 64 ビットの場合

5500+A×B+C×D+900×E+900×F+（100+G）×H+800×I

(凡例)

A，C：プロセスおよび使用する通信プロトコル対応製品によって次のように値が異なります。

プロセス	通信プロトコル対応製品	A	C
アプリケーション起動 プロセス	—	• 32 ビットの場合 1200 • 64 ビットの場合 1500	• 32 ビットの場合 3300 • 64 ビットの場合 4000
MCF 通信プロセス	TP1/NET/NCSB TP1/NET/X25-Extended	2500	1000
	TP1/NET/OSAS-NIF	1000	4500

プロセス	通信プロトコル対応製品	A	C
MCF 通信プロセス	TP1/NET/UDP	—	• 32 ビットの場合 3500 • 64 ビットの場合 4200
	上記以外	• 32 ビットの場合 500 • 64 ビットの場合 600	• 32 ビットの場合 3000 • 64 ビットの場合 3600

(凡例)

—：該当しない

B：コネクション数（mcftalccn 定義コマンドの指定数）

アプリケーション起動プロセスの場合は 1 です。TP1/NET/UDP の MCF 通信プロセスの場合は 0 です。

D：論理端末数（mcftalcle 定義コマンドの指定数）

E：出力キューにたまる送信メッセージの最大数※（ディスクキューおよびメモリキューの両方を含みます）

F：アプリケーション数（mcfaalcap 定義コマンドの指定数）

G：ユーザタイム監視機能の最大メッセージ長（mcfttim 定義コマンドの msgsize オペランドの指定値）

H：最大タイム監視要求数（mcfttim 定義コマンドの timereqno オペランドの指定値）

I：コネクショングループ数（mcftgrpcn 定義コマンドの指定数）

注※

システムとして想定する論理端末ごとの滞留数の総和を指定してください。なお、少なくとも同時に送信する論理端末の最大数を指定してください。

指定した数が正しいかどうかは、mcftlsle -m コマンドを実行し、最大未送信メッセージ数を確認してください。

算出式 2：

$$21000 + 2 \times A$$

(凡例)

A：MCF のトレースバッファサイズ（mcfttrc 定義コマンドの size オペランドの指定値）

(c) MCF オンラインコマンドプロセス

MCF オンラインコマンドプロセスで使用する共用メモリのサイズは、512 バイトです。MCF オンラインコマンドプロセスは、TP1/NET/XMAP3 を使用する場合に、一つのプロセスを使用します。TP1/NET/XMAP3 を使用しない場合、プロセスは発生しません。

(d) MCF マッピングプロセス

MCF マッピングプロセスで使用する共用メモリのサイズは、次に示す算出式の算出結果です。MCF マッピングプロセスは、TP1/NET/XMAP3 を使用する場合に、一つのプロセスを使用します。TP1/NET/XMAP3 を使用しない場合、プロセスは発生しません。

$$1300 + 10000 \times A + \Sigma (32 \times B + 300 \times C) + D + E + F \times G$$

(凡例)

Σ () : 各マッピング属性定義について括弧内を計算した結果のすべての定義総計

A : マッピングサービス識別子 (マッピングサービス定義に記述したマッピングサービス識別子の数)

B : 常駐管理をする物理マップ数 (マッピングサービス属性定義の MAPNAME の指定数)

C : LRU 管理をする物理マップ数 (マッピングサービス属性定義の MAPCNT の指定数)

D : 全常駐させる物理マップサイズの合計

E : LRU 管理をする 64 キロバイト以下の全非常駐物理マップのサイズの合計

F : LRU 管理をする 64 キロバイトを超える全非常駐物理マップ中、サイズの最大値

G : LRU 管理をする 64 キロバイトを超える全非常駐物理マップ数

(2) 動的共用メモリ

ここで説明する算出結果の合計値が MCF サービスで使用する動的共用メモリのサイズです。ほかの OpenTP1 のサービスが使用する動的共用メモリに MCF サービスで使用する動的共用メモリを加え、システム環境定義の dynamic_shmpool_size オペランドに指定してください。

(a) MCF マネジャプロセス

MCF マネジャプロセスで使用する共用メモリは、次に示す算出式の算出結果です。MCF マネジャプロセスは、OpenTP1 システム内で一つのプロセスを使用します。

$$((600 + A) \times B + 1600) \times C + 2000 \times D + 1600 \times E + (600 + F) \times G \times H + 2 \times (1600 \times H) + 1600 \times I + 2 \times (600 + J) \times K$$

(凡例)

A : 最大受信メッセージ長

B : 最大受信セグメント数

C : すべての入力キュー (メモリキュー) にたまる受信メッセージの最大数^{※1} (ディスクキューの場合は 0 です)

D : メッセージの受信、または送信を行う UAP プロセス数

E : すべての入力キューにたまる受信メッセージの最大数^{※1} (ディスクキューおよびメモリキューの両方を含みます)

F : 最大送信メッセージ長

G : 最大送信セグメント数

H：すべての出力キュー（メモリキュー）にたまる送信メッセージの最大数※2（ディスクキューの場合は0です）

I：すべての出力キューにたまる送信メッセージの最大数※2（ディスクキューおよびメモリキューの両方を含みます）

J：継続問い合わせ応答用一時記憶データ格納用領域サイズの最大値（すべての cont 型のアプリケーションの mcfaalcap 定義コマンドの tempsize オペランドの指定値の最大値）

K：継続問い合わせ応答形態の通信を行う論理端末数

注※1

システムとして想定するサービスグループごとの滞留数の総和を指定してください。

指定した数が正しいかどうかは、mcftlssg -m コマンドを実行し、最大未処理受信メッセージ数を確認してください。

注※2

システムとして想定する論理端末ごとの滞留数の総和を指定してください。なお、少なくとも同時に送信する論理端末の最大数を指定してください。

指定した数が正しいかどうかは、mcftlsle -m コマンドを実行し、最大未送信メッセージ数を確認してください。

(b) MCF オンラインコマンドプロセス

MCF オンラインコマンドプロセスで使用する共用メモリは、14000 バイトです。MCF オンラインコマンドプロセスは、OpenTP1 システム内にプロセスは発生しません。TP1/NET/XMAP3 を使用する場合に、一つのプロセスを使用します。

付録 C 監査イベントを取得する定義

監査イベントに対応するメッセージ ID と、それぞれのメッセージを出力するために指定する定義を次の表に示します。

表 C-1 監査ログのメッセージ ID と指定する定義

監査イベント	log_audit_message オペランドを指定する定義				メッセージ ID
	ログサービス定義	ユーザサービス定義	rap リスナーサービス定義	rap クライアントマネージャサービス定義	
OpenTP1 開始	○	—	—	—	KFCA334 00-I
OpenTP1 待機状態	○	—	—	—	KFCA334 01-I
OpenTP1 正常終了	○	—	—	—	KFCA334 02-I
OpenTP1 異常終了	○	—	—	—	KFCA334 03-E
プロセスサービスの重大なエラー	○	—	—	—	KFCA334 04-E
ユーザサーバ開始	○	—	—	—	KFCA334 05-I
ユーザサーバ正常終了	○	—	—	—	KFCA334 06-I
ユーザサーバ異常終了	○	—	—	—	KFCA334 07-E
ユーザサーバ閉塞	○	—	—	—	KFCA334 08-I
ユーザサーバのサービス閉塞	○	—	—	—	KFCA334 09-I
クライアントユーザ認証成功	○	—	—	—	KFCA334 10-I
クライアントユーザ認証失敗	○	—	—	—	KFCA334 11-W
サービス関数の実行開始	○	○	○	—	KFCA334 12-I

監査イベント	log_audit_message オペランドを指定する定義				メッセージ ID
	ログサービス定義	ユーザサービス定義	rap リスナーサービス定義	rap クライアントマネージャサービス定義	
サービス関数の実行完了	○	○	○	—	KFCA33413-I
不正電文の破棄	○	○	○	—	KFCA33414-W
RPC 呼び出し完了	○	○	○	—	KFCA33415-I
RPC 応答の受信 (dc_rpc_poll_any_replies 関数の使用時)	○	○	—	—	KFCA33416-I
rap の不正電文の破棄	○	—	○	○	KFCA33417-W
OpenTP1 ファイルシステムに対するアクセスエラー	○	○	○	—	KFCA33418-W
コマンドの実行	○	—	—	—	KFCA33419-I
OpenTP1 サービスの開始	○	—	—	—	KFCA33420-I※ ¹
OpenTP1 サービスの停止	○	—	—	—	KFCA33421-I※ ¹
UAP からユーザが任意に取得する監査ログの取得	○	○	—	—	KFCA34000-x～ KFCA34999-x※ ²

(凡例)

○：log_audit_message オペランドにメッセージ ID を指定することで監査イベントが発生したときに該当メッセージが出力されます。

—：log_audit_message オペランドにメッセージ ID を指定しても無効です（該当メッセージは出力されません）。

注※1

Windows 版だけで出力されるメッセージ ID です。

注※2

KFCA34000-x~KFCA34999-x は、UAP で任意の監査ログを取得する場合に、監査ログに対して割り当てられるメッセージ ID です。x には dc_log_audit_print 関数で指定したメッセージの種類 (E, W または I) が入ります。

付録 D 定義チェックの詳細

定義チェック時に出力されるメッセージおよび問題識別コードと、それらに対応するチェック対象の定義ファイル名、オペランド名、定義コマンド名などを表 D-1 に示します。

表 D-1 のヘッダで使用する用語の意味を次に示します。

- ・「コード」：問題識別コード
- ・「ID」：出力されるメッセージ ID
- ・「区分」：メッセージ区分
メッセージ区分の記号の意味を次に示します。
C：CHECK（オペランド値の妥当性について確認を促す場合に出力されます）
E：ERROR（OpenTP1 の起動および停止ができない状態、または動作不完全となる問題を検出した場合に出力されます）
W：WARNG（推奨しない値が指定されていることを検出した場合に出力されます）
- ・「ファイル名」：チェック対象の定義ファイル名（定義ファイル名を任意に指定できる場合は定義名）
- ・「定義」：チェック対象のオペランド名、定義コマンド名など
- ・「関連ファイル名」：チェック時に比較対象となる定義ファイル名
- ・「関連定義」：チェック時に比較対象となるオペランド名、定義コマンド名など
- ・「条件式または判断根拠」：チェックの条件、またはチェック結果に至る判断根拠

表 D-1 定義チェックの詳細

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
1	ADM-0001	KFCA00268-W	E	env	user_command	—	—	ファイルが存在しない	ADM#1 ADM#2
2	ADM-0002	KFCA00268-W	E	env	user_command	—	—	ファイルにアクセスできない	ADM#1 ADM#2
3	ADM-0003	KFCA00268-W	E	env	user_command	—	—	ファイルパスが長過ぎる	ADM#1 ADM#2
4	ADM-0004	KFCA00268-W	E	env	user_command	—	—	実行可能なファイルではない	ADM#1 ADM#2 ADM#3
5	ADM-0005	KFCA00268-W	E	env	user_command	—	—	ファイルではない	ADM#1 ADM#2

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
6	ADM-0006	KFCA00268-W	E	env	user_command_online	—	—	ファイルが存在しない	ADM#1 ADM#2
7	ADM-0007	KFCA00268-W	E	env	user_command_online	—	—	ファイルにアクセスできない	ADM#1 ADM#2
8	ADM-0008	KFCA00268-W	E	env	user_command_online	—	—	ファイルパスが長過ぎる	ADM#1 ADM#2
9	ADM-0009	KFCA00268-W	E	env	user_command_online	—	—	実行可能なファイルではない	ADM#1 ADM#2 ADM#3
10	ADM-0010	KFCA00268-W	E	env	user_command_online	—	—	ファイルではない	ADM#1 ADM#2
11	ADM-0011	KFCA00270-W	C	env	user_command_online_tplmgr_id	env	user_command_online	(user_command_online_tplmgr_id 指定あり) && (user_command_online 指定なし)	OS が Windows 以外の場合だけチェック対象です。
12	ADM-0012	KFCA00276-W	C	env	user_server_ha	sysconf	ha_conf	(user_server_ha == Y) && (ha_conf != Y)	—
13	ADM-0013	KFCA00269-W	C	betranrc	dcstart_wakeup_retry_interval	betranrc	dcstart_wakeup_retry_count	(dcstart_wakeup_retry_interval 指定あり) && (dcstart_wakeup_retry_count == 0)	—
14	ADM-0014	KFCA00270-W	C	betranrc	dcstart_wakeup_retry_interval	betranrc	dcstart_wakeup_retry_count	(dcstart_wakeup_retry_interval 指定あり) && (dcstart_wakeup_retry_count 指定なし)	—
15	ADM-0015	KFCA00286-W	E	betranrc	node_id	—	—	betranrc 定義ファイルが存在しない	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
16	ADM-0016	KFCA00285-W	E	betranrc	node_id	—	—	node_id 指定なし	—
17	ADM-0017	KFCA01880-W	W	betranrc	dcbindht	sysconf	ha_conf	(ha_conf == Y) && (dcbindht 指定なし)	—
18	ADM-0018	KFCA00266-W	C	betranrc	prc_port	—	—	prc_port 指定あり	—
19	ADM-0019	KFCA00282-W	C	betranrc	multi_node_option	—	—	multi_node_option 指定あり	OS が Windows または Linux の場合、チェック対象の定義はサポートしていません。
20	ADM-0020	KFCA33301-E	E	sysconf	dam_conf tam_conf mqa_conf jar_conf dcsvstart -m	betranrc	jnl_fileless_option	(jnl_fileless_option == Y && dam_conf == Y) (jnl_fileless_option == Y && tam_conf == Y) (jnl_fileless_option == Y && mqa_conf == Y) (jnl_fileless_option == Y && jar_conf == Y) (jnl_fileless_option == Y && dcsvstart -m 指定あり)	—
21	—	KFCA01868-E	—	sysconf	dcsvstart	—	—	先頭が _mu ではない	—
22	—	KFCA01866-E	—	sysconf	dcsvstart	—	—	重複したサーバ名が存在する	—
23	—	KFCA01868-E	—	usrconf	dcsvstart	—	—	先頭が _ である	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
24	—	KFCA01866-E	—	usrconf	dcsvstart	—	—	重複したサーバ名が存在する	—
25	CLT-0001	KFCA25160-W	C	cltsrv	clt_trn_conf clt_cup_conf	—	—	(clt_trn_conf == N && clt_cup_conf == N) (clt_trn_conf == N && clt_cup_conf 指定なし)	クライアントサービス定義の論理チェックは行いません (clt_port を除く)。
26	CLT-0002	KFCA00278-W	C	cltsrv	parallel_count	cltsrv	clt_trn_conf	(parallel_count 指定あり) && (clt_trn_conf == N)	—
27	CLT-0003	KFCA25161-W	E	cltsrv	parallel_count	—	—	(clt_trn_conf == Y) && (parallel_count の常駐プロセス数 > parallel_count の最大プロセス数)	—
28	CLT-0004	KFCA00262-W	E	cltsrv	parallel_count	prc	prc_process_count	(clt_trn_conf == Y) && (parallel_count の常駐プロセス数 >= prc_process_count 指定値)	—
29	CLT-0005	KFCA00262-W	W	cltsrv	parallel_count	trn	trn_tran_process_count	(clt_trn_conf == Y) && (parallel_count の最大プロセス数 > trn_tran_process_count 指定値)	—
30	CLT-0006	KFCA00278-W	C	cltsrv	balance_count	cltsrv	clt_trn_conf	((balance_count 指定あり) && (clt_trn_conf == N))	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
31	CLT-0007	KFCA25162-W	C	cltsrv	balance_count	cltsrv	parallel_count	(clt_trn_conf == Y) && (parallel_count の常駐プロセス数 == parallel_count の最大プロセス数) && (balance_count 指定あり)	—
32	CLT-0008	KFCA00262-W	C	cltsrv	trn_expiration_time	cltsrv	trn_completion_limit_time	(trn_expiration_time > 0) && (trn_completion_limit_time > 0) && (trn_expiration_time > trn_completion_limit_time)	—
33	CLT-0009	KFCA00282-W	C	cltsrv	trn_statistics_item	—	—	trn_statistics_item に cputime 指定あり	OS が Windows の場合、チェック対象の定義はサポートしていません。
34	CLT-0010	KFCA00278-W	C	cltsrv	trn_expiration_time_suspend	cltsrv	trn_expiration_time	(trn_expiration_time == 0) && (trn_expiration_time_suspend 指定あり)	trn_expiration_time（省略時：0）
35	CLT-0011	KFCA00282-W	C	cltsrv	trn_cpu_time	—	—	trn_cpu_time 指定あり ただし、0 を指定した場合だけ	OS が Windows の場合、チェック対象の定義はサポートしていません。
36	CLT-0012	KFCA00264-W	C	cltsrv	clt_inquire_time	cltsrv	trn_expiration_time	(clt_cup_conf == Y)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
36	CLT-0012	KFCA00264-W	C	cltsrv	clt_inquire_time	cltsrv	trn_expiration_time	&&(trn_expiration_time > 0) &&(clt_inquire_time > trn_expiration_time)	—
37	CLT-0013	KFCA00264-W	C	cltsrv	clt_inquire_time	cltsrv	trn_limit_time	(clt_cup_conf == Y) &&(trn_limit_time > 0) &&(clt_inquire_time > trn_limit_time)	—
38	CLT-0014	KFCA00264-W	C	cltsrv	clt_inquire_time	cltsrv	trn_completion_limit_time	(clt_cup_conf == Y) &&(trn_completion_limit_time > 0) &&(clt_inquire_time > trn_completion_limit_time)	—
39	CLT-0015	KFCA00266-W	C	cltsrv	clt_port	—	—	clt_port 指定あり	clt_port の指定がある場合は clt_trn_conf や clt_cup_conf の指定に関係なく論理チェックを実施します。
40	CLT-0016	KFCA00278-W	C	cltsrv	cup_parallel_count	cltsrv	clt_cup_conf	(cup_parallel_count 指定あり) &&(clt_cup_conf == N)	—
41	CLT-0017	KFCA00279-W	C	cltsrv	cup_parallel_count	cltsrv	clt_cup_conf	(cup_parallel_count 指定あり) &&(clt_cup_conf 指定なし)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
42	CLT-0018	KFCA00262-W	E	cltsrv	cup_parallel_count	prc	prc_process_count	(clt_cup_conf == Y) && (cup_parallel_count の常駐プロセス数 >= prc_process_count 指定値)	—
43	CLT-0019	KFCA00278-W	C	cltsrv	cup_balance_count	cltsrv	clt_cup_conf	(cup_balance_count 指定あり) && (clt_cup_conf == N)	—
44	CLT-0020	KFCA00279-W	C	cltsrv	cup_balance_count	cltsrv	clt_cup_conf	(cup_balance_count 指定あり) && (clt_cup_conf 指定なし)	—
45	CLT-0021	KFCA25162-W	C	cltsrv	cup_balance_count	cltsrv	cup_parallel_count	(clt_cup_conf == Y) && (cup_balance_count 指定あり) && (cup_parallel_count の常駐プロセス数 == cup_parallel_count の最大プロセス数)	—
46	CLT-0022	KFCA00278-W	C	cltsrv	clttrn_port	cltsrv	clt_trn_conf	(clttrn_port 指定あり) && (clt_trn_conf == N)	—
47	CLT-0023	KFCA00266-W	C	cltsrv	clttrn_port	—	—	(clt_trn_conf == Y) && (clttrn_port 指定あり)	—
48	CLT-0024	KFCA25163-W	C	cltsrv	clttrn_port	cltsrv	parallel_count	(clt_trn_conf == Y) && (clttrn_port + parallel_count の最大プロセス数 - 1) > 65535	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
49	CLT-0025	KFCA00278-W	C	cltsrv	cltcon_port	cltsrv	clt_cup_conf	(cltcon_port 指定あり) && (clt_cup_conf == N)	—
50	CLT-0026	KFCA00279-W	C	cltsrv	cltcon_port	cltsrv	clt_cup_conf	(cltcon_port 指定あり) && (clt_cup_conf 指定なし)	—
51	CLT-0027	KFCA00266-W	C	cltsrv	cltcon_port	—	—	(clt_cup_conf == Y) && (cltcon_port の指定あり)	—
52	CLT-0028	KFCA25163-W	C	cltsrv	cltcon_port	cltsrv	cup_parallel_count	(clt_cup_conf == Y) && (cltcon_port + cup_parallel_count の最大プロセス数 - 1) > 65535	—
53	CLT-0029	KFCA00272-W	C	cltsrv	trn_optimum_item	—	—	trn_optimum_item != base	—
54	CLT-0030	KFCA00262-W	C	cltsrv	trn_watch_time	cltsrv	trn_limit_time	(trn_limit_time > 0) && (trn_watch_time > trn_limit_time)	—
55	CLT-0031	KFCA00262-W	C	cltsrv	trn_watch_time	cltsrv	trn_completion_limit_time	(trn_completion_limit_time > 0) && (trn_watch_time > trn_completion_limit_time)	—
56	CLT-0032	KFCA00272-W	C	cltsrv	trn_rollback_information_put	—	—	trn_rollback_information_put != all	—
57	CLT-0033	KFCA00265-W	C	cltsrv	message_store_buflen	—	—	message_store_buflen 指定あり	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
58	CLT-0034	KFCA00262-W	C	cltsrv	parallel_count	prc	prc_process_count	(clt_trn_conf == Y) && (parallel_count の常駐プロセス数 < prc_process_count 指定値) && (parallel_count の最大プロセス数 >= prc_process_count 指定値)	—
59	CLT-0035	KFCA00262-W	C	cltsrv	cup_parallel_count	prc	prc_process_count	(clt_cup_conf == Y) && (cup_parallel_count の常駐プロセス数 < prc_process_count 指定値) && (cup_parallel_count の最大プロセス数 >= prc_process_count 指定値)	—
60	CLT-0036	KFCA25161-W	E	cltsrv	cup_parallel_count	—	—	(clt_cup_conf == Y) && (cup_parallel_count の常駐プロセス数 > cup_parallel_count の最大プロセス数)	—
61	CLT-0037	KFCA00261-W	C	cltsrv	trn_expiration_time	cltsrv	trn_cpu_time	(trn_expiration_time > 0) && (trn_cpu_time > 0) && (trn_expiration_time < trn_cpu_time)	OS が Windows の場合はチェックしません。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
62	CLT-0038	KFCA00262-W	C	cltsrv	trn_cpu_time	cltsrv	trn_completion_limit_time	(trn_cpu_time > 0) && (trn_completion_limit_time > 0) && (trn_cpu_time > trn_completion_limit_time)	OS が Windows の場合はチェックしません。
63	CPD-0001	KFCA00286-W	E	チェックポイントダンプサービス定義	全体	jnl	jnldfs	jnldfs の-c オプションに指定したチェックポイントダンプサービス定義ファイルが存在しない	JNL#1 CPD#1
64	CPD-0002	KFCA00285-W	E	チェックポイントダンプサービス定義	jnl_objservername	—	—	jnl_objservername の指定がない	JNL#1 CPD#1
65	CPD-0003	KFCA26030-W	E	チェックポイントダンプサービス定義	jnl_objservername	—	—	サーバ名が「_tjl」, 「_mqa」, または先頭3文字が「_mu」で始まるファイル名以外である	JNL#1 CPD#1
66	CPD-0004	KFCA00272-W	C	チェックポイントダンプサービス定義	jnl_reduced_mode	—	—	jnl_reduced_mode != 0	JNL#1 CPD#1
67	CPD-0005	KFCA26031-I	C	チェックポイントダンプサービス定義	jnl_reserved_file_auto_open	チェックポイントダンプサービス定義	jnladdfg	(jnl_reserved_file_auto_open == Y) && (すべての jnladdfg に ONL 指定あり)	JNL#1 CPD#1
68	CPD-0006	KFCA00278-W	C	チェックポイントダンプサービス定義	jnl_singleoperation	チェックポイントダンプサービス定義	jnl_dual	(jnl_singleoperation == Y) && (jnl_dual == N)	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
69	CPD-0007	KFCA00279-W	C	チェックポイントダンプサービス定義	jnl_singleoperation	チェックポイントダンプサービス定義	jnl_dual	(jnl_singleoperation == Y) && (jnl_dual 指定なし)	JNL#1 CPD#1
70	CPD-0008	KFCA26032-W	E	チェックポイントダンプサービス定義	jnladdfg	—	—	jnladdfg -j srf が複数指定されている	JNL#1 CPD#1
71	CPD-0009	KFCA26033-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	指定したファイルシステムがキャラクタ型スペシャルファイルではない、またはこのファイルシステムに対応する装置がない	JNL#1 CPD#1
72	CPD-0010	KFCA26034-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	指定したファイルが filmkfs コマンドで OpenTP1 ファイルシステム用に初期化されていない	JNL#1 CPD#1
73	CPD-0011	KFCA26035-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	チェックポイントダンプファイルがない	JNL#1 CPD#1
74	CPD-0012	KFCA26036-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	OpenTP1 ファイルシステムのバージョン不一致	JNL#1 CPD#1
75	CPD-0013	KFCA26037-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	チェックポイントダンプファイルのオープンで上限値オーバが発生	JNL#1 CPD#1
76	CPD-0014	KFCA26038-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	該当するスペシャルファイルのアクセス権限なし	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
77	CPD-0015	KFCA26039-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	該当するチェックポイントダンプファイルのアクセス権限なし	JNL#1 CPD#1
78	CPD-0016	KFCA26040-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	チェックポイントダンプファイルに対して I/O エラーが発生	JNL#1 CPD#1
79	CPD-0017	KFCA26041-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	チェックポイントダンプファイルのオープンでメモリ不足が発生	JNL#1 CPD#1
80	CPD-0018	KFCA26042-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	指定されたファイルはチェックポイントダンプファイルとして使用できない	JNL#1 CPD#1
81	CPD-0019	KFCA26043-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	チェックポイントダンプファイルの読み込みエラーが発生	JNL#1 CPD#1
82	CPD-0020	KFCA26032-W	E	チェックポイントダンプサービス定義	jnladdpf	—	—	jnladdfg -j srf が複数指定されている	JNL#1 CPD#1
83	CPD-0021	KFCA00265-W	C	チェックポイントダンプサービス定義	jnl_cdskip_limit	チェックポイントダンプサービス定義	jnl_objservername	jnl_objservername == _tjl && jnl_cdskip_limit の指定あり	JNL#1 CPD#1
84	CPD-0022	KFCA00280-W	C	チェックポイントダンプサービス定義	jnl_cdskip_limit	チェックポイントダンプサービス定義	jnl_objservername	jnl_objservername != _tjl && jnl_cdskip_limit の指定あり	JNL#1 CPD#1
85	CPD-0023	KFCA00280-W	C	チェックポイントダンプサービス定義	jnl_cdskip_msg	チェックポイントダンプサービス定義	jnl_objservername	jnl_objservername != _tjl && jnl_cdskip_msg の指定あり	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
86	—	KFCA02175-E	—	チェックポイントダンプサービス定義	jnl_objservername	ほかのチェックポイントダンプサービス定義	jnl_objservername	ほかのチェックポイントダンプサービス定義に指定した jnl_objservername と同じサーバ名を指定している	JNL#1 CPD#1
87	—	KFCA02135-E	—	チェックポイントダンプサービス定義	jnladdfg	—	—	ほかのチェックポイントダンプサービス定義の jnladdfg で指定したファイルグループ名と重複している	JNL#1 CPD#1
88	—	KFCA02149-E	—	チェックポイントダンプサービス定義	jnladdfg	—	—	jnladdfg(ONL 指定あり && -jsrf 指定なし)の指定が (assurance_count 指定値 + 1) より少ない	JNL#1 CPD#1
89	—	KFCA02153-E	—	チェックポイントダンプサービス定義	jnladdfg	—	—	jnladdfg の指定が最大指定可能数の 60 個を超えている	JNL#1 CPD#1
90	—	KFCA02156-E	—	チェックポイントダンプサービス定義	jnladdfg	—	—	-g オプションに指定されたファイルグループ用の jnladdpf がない	JNL#1 CPD#1
91	—	KFCA02190-E	—	チェックポイントダンプサービス定義	jnladdfg	—	—	jnladdfg(ONL 指定あり && -jsrf 指定なし)の指定が最大指定可能数の 30 個を超えている	JNL#1 CPD#1
92	—	KFCA02141-E	—	チェックポイントダンプサービス定義	jnladdfg	—	—	ほかの jnladdfg で指定したファイルグループ名と重複している	JNL#1 CPD#1
93	—	KFCA02137-E	—	チェックポイントダンプ	jnladdpf	—	—	ほかのチェックポイントダンプサービス定義の	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
93	—	KFCA02137-E	—	プサービス定義	jnladdpf	—	—	jnladdpf で指定した物理ファイル名と重複している	JNL#1 CPD#1
94	—	KFCA02148-E	—	チェックポイントダン プサービス定義	jnladdpf	チェックポイント ダンプサービス定義	jnladdfg	-g オプションに指定されたファイルグループ用の jnladdfg 定義がない	JNL#1 CPD#1
95	—	KFCA02154-E	—	チェックポイントダン プサービス定義	jnladdpf	—	—	ほかの jnladdpf で指定したファイルグループ名と重複している	JNL#1 CPD#1
96	—	KFCA02155-E	—	チェックポイントダン プサービス定義	jnladdpf	—	—	jnladdpf の指定が最大指定可能数の 60 個を超えている	JNL#1 CPD#1
97	—	KFCA26004-E	—	チェックポイントダン プサービス定義	jnladdpf	チェックポイント ダンプサービス定義	jnl_dual	(jnl_dual == Y 指定あり) && ((jnladdpf の-a オプション指定なし) (jnladdpf の-b オプション指定なし))	JNL#1 CPD#1
98	—	KFCA26005-E	—	チェックポイントダン プサービス定義	jnladdpf	チェックポイント ダンプサービス定義	jnl_dual	(jnladdpf の-b オプション指定あり) && (jnl_dual == N)	JNL#1 CPD#1
99	—	KFCA26005-E	—	チェックポイントダン プサービス定義	jnladdpf	チェックポイント ダンプサービス定義	jnl_dual	(jnladdpf の-b オプション指定あり) && (jnl_dual 指定なし)	JNL#1 CPD#1
100	DAM-0001	KFCA02751-W	W	dam	dam_update_block	—	—	dam_update_block > 5000	—
101	DAM-0002	KFCA02752-W	C	dam	dam_message_level	—	—	(dam_message_level 指定なし) (dam_message_level == 1)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
102	DAM-0003	KFCA00264-W	C	dam	dam_tran_process_count	trn	trn_tran_process_count	(dam_tran_process_count 指定あり) && (dam_tran_process_count > trn_tran_process_count)	—
103	DAM-0004	KFCA02753-W	C	dam	dam_cache_size	—	—	(dam_cache_size_fix 指定なし) && (dam_cache_size < DAM 起動時計算式)	—
104	DAM-0005	KFCA00277-W	C	dam	dam_cache_size	dam	dam_cache_size_fix	(dam_cache_size_fix 指定あり) && (dam_cache_size 指定あり)	—
105	DAM-0006	KFCA02754-W	C	dam	dam_cache_size_fix	—	—	(dam_cache_size_fix 指定あり)	—
106	DAM-0007	KFCA00282-W	C	dam	dam_cache_attribute	—	—	(OS == AIX OS == Linux) && (dam_cache_attribute == fixed)	OS が AIX または Linux の場合だけチェック対象です。
107	DAM-0008	KFCA00272-W	C	dam	dam_cache_attribute	—	—	(OS == HP-UX OS == Solaris) && (dam_cache_attribute == fixed)	OS が HP-UX または Solaris の場合だけチェック対象です。
108	DAM-0009	KFCA02755-W	W	dam	dam_io_error_occure	—	—	dam_io_error_occure == stop	—
109	DAM-0010	KFCA00271-W	W	dam	dam_cache_reuse_from	—	—	dam_cache_reuse_from == first	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
110	DAM-0011	KFCA02756-W	W	dam	damchmt コマンド	—	—	(DAM ファイルブロック数 > 5000) && (damchmt 指定なし)	—
111	DAM-0012	KFCA02757-W	W	dam	damfile コマンド	—	—	(damfile コマンド指定数 + dam_added_file 指定値) > 3600	—
112	—	KFCA02566-E	—	dam	damcache コマンド	—	—	damfile で DAM ファイルの指定なし	—
113	—	KFCA02566-E	—	dam	damchmt コマンド	—	—	damfile で DAM ファイルの指定なし	—
114	—	KFCA01636-E	—	dam	damfile コマンド	—	—	DAM ファイルパス名が正しく指定されていない	—
115	—	KFCA01636-E	—	dam	damfile コマンド	—	—	OpenTP1 ファイルシステムのある装置が未初期化	—
116	—	KFCA01636-E	—	dam	damfile コマンド	—	—	DAM ファイルがない	—
117	—	KFCA01636-E	—	dam	damfile コマンド	—	—	オープンファイル数がシステムの上限值を超過	—
118	—	KFCA01636-E	—	dam	damfile コマンド	—	—	OpenTP1 ファイルシステムへのアクセス権がない	—
119	—	KFCA01636-E	—	dam	damfile コマンド	—	—	OpenTP1 ファイルシステムへのアクセス権がない	—
120	—	KFCA01636-E	—	dam	damfile コマンド	—	—	DAM ファイルのバージョンが正しくない	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
121	—	KFCA01646-E	—	dam	damfile コマンド	—	—	DAM ファイル に対して I/O エ ラーが発生	—
122	—	KFCA01627-E	—	dam	damfile コマンド	—	—	メモリ不足が 発生	—
123	—	KFCA01637-E	—	dam	damfile コマンド	—	—	指定した DAM ファイルが正し い DAM ファイ ルではない	—
124	—	KFCA01617-E	—	システム ジャーナル サービス 定義	jnl_max_ datasi ze	—	—	(dam_update_ block_over == flush) && (jnl_max_datas ize < 152 + 最 大ブロック長× 2) (dam_update_ block_over == error) && (jnl_max_datas ize < 152 + 最 大ブロック長)	JNL#1 CPD#1
125	JNL-00 01	KFCA00262-W	W	jnl	jnl_arc_ termina te_time out	env	system_ termina te_watc h_time	!(OS == Windows OS == Linux) && system_termin ate_watch_tim e <= jnl_arc_termin ate_timeout	JNL#1 CPD#1
126	JNL-00 05	KFCA32800-W	E	jnl	jnldfs	—	—	-r オプションの 指定なし	JNL#1 CPD#1
127	JNL-00 06	KFCA32801-W	E	jnl	jnldfs	—	—	-r オプションの 引数の指定なし	JNL#1 CPD#1
128	JNL-00 07	KFCA32802-W	W	jnl	jnldfs	—	—	jnldfs の定義数 > 1	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
129	JNL-0010	KFCA00272-W	C	システムジャーナルサービス定義	jnl_cdiinterval	—	—	jnl_cdiinterval < 1000	JNL#1 CPD#1
130	JNL-0011	KFCA00272-W	C	システムジャーナルサービス定義	jnl_cdiinterval	—	—	jnl_cdiinterval > 1000	JNL#1 CPD#1
131	JNL-0014	KFCA00269-W	C	システムジャーナルサービス定義	jnl_singleoperation	—	jnl_dual	(jnl_dual == N) && (jnl_singleoperation == Y)	JNL#1 CPD#1
132	JNL-0015	KFCA00270-W	C	システムジャーナルサービス定義	jnl_arc_buff_size	—	jnl_arc_name	!(OS == Windows OS == Linux) && jnl_arc_name 指定なし && jnl_arc_buff_size 指定あり	JNL#1 CPD#1
133	JNL-0016	KFCA00273-W	E	システムジャーナルサービス定義	jnl_arc_buff_size	—	jnl_arc_max_datasize	!(OS == Windows OS == Linux) && (jnl_arc_buff_size < ↑ (jnl_arc_max_ddatasize × 1024) / 1048576 ↑ × 3)	JNL#1 CPD#1
134	JNL-0017	KFCA00270-W	C	システムジャーナルサービス定義	jnl_arc_max_ddatasize	—	jnl_arc_name	!(OS == Windows OS == Linux) && jnl_arc_name 指定なし && jnl_arc_max_ddatasize 指定あり	JNL#1 CPD#1
135	JNL-0018	KFCA00273-W	E	システムジャーナルサービス定義	jnl_arc_max_ddatasize	—	jnl_max_datasize	!(OS == Windows OS == Linux) && (jnl_arc_max_ddatasize < ↑ (↑	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
135	JNL-0018	KFCA00273-W	E	システムジャーナルサービス定義	jnl_arc_max_data_size	—	jnl_max_data_size	(jnl_max_data_size + 336) / 4096 ↑ × 4096) / 1024 ↑)	JNL#1 CPD#1
136	JNL-0019	KFCA00274-W	E	システムジャーナルサービス定義	jnl_arc_max_data_size	betranrc	rpc_max_message_size	!(OS == Windows OS == Linux) && (rpc_max_message_size < ↑ (jnl_arc_max_data_size × 1024 + 4096) / 1048576 ↑)	JNL#1 CPD#1
137	JNL-0020	KFCA00270-W	C	システムジャーナルサービス定義	jnl_arc_terminate_check	—	jnl_arc_name	!(OS == Windows OS == Linux) && jnl_arc_name 指定なし && jnl_arc_terminate_check 指定あり	JNL#1 CPD#1
138	JNL-0021	KFCA00270-W	C	システムジャーナルサービス定義	jnl_arc_rec_kind	—	jnl_arc_name	!(OS == Windows OS == Linux) && jnl_arc_name 指定なしの場合, jnl_arc_rec_kind 指定値は有効にならない	JNL#1 CPD#1
139	JNL-0022	KFCA00270-W	C	システムジャーナルサービス定義	jnl_arc_uj_code	—	jnl_arc_name	!(OS == Windows OS == Linux) && jnl_arc_name 指定なしの場合, jnl_arc_uj_code 指定値は有効にならない	JNL#1 CPD#1
140	JNL-0023	KFCA00270-W	C	システムジャーナルサービス定義	jnl_arc_check_level	—	jnl_arc_name	!(OS == Windows OS == Linux) &&	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
140	JNL-0023	KFCA00270-W	C	システムジャーナルサービス定義	jnl_arc_check_level	—	jnl_arc_name	jnl_arc_name 指定なしの場合, jnl_arc_check_level 指定値は有効にならない	JNL#1 CPD#1
141	JNL-0024	KFCA00270-W	C	システムジャーナルサービス定義	jnl_arc_trn_stat	—	jnl_arc_name	!(OS == Windows OS == Linux) && jnl_arc_name 指定なしの場合, jnl_arc_trn_stat 指定値は有効にならない	JNL#1 CPD#1
142	JNL-0025	KFCA00269-W	C	システムジャーナルサービス定義	jnl_auto_unload	—	jnl_unload_check	jnl_unload_check=N を指定している場合, jnl_auto_unload=Y の指定は無効	JNL#1 CPD#1
143	JNL-0026	KFCA00280-W	C	システムジャーナルサービス定義	jnl_auto_unload_path	—	jnl_auto_unload	jnl_auto_unload=N を指定している場合, jnl_auto_unload_path の指定は無効	JNL#1 CPD#1
144	JNL-0027	KFCA32804-W	W	システムジャーナルサービス定義	jnl_auto_unload_path	—	—	ディレクトリへのアクセス権がない	JNL#1 CPD#1
145	JNL-0028	KFCA32804-W	W	システムジャーナルサービス定義	jnl_auto_unload_path	—	—	ディレクトリが存在しない	JNL#1 CPD#1
146	JNL-0029	KFCA32804-W	W	システムジャーナルサービス定義	jnl_auto_unload_path	—	—	指定パスがディレクトリではない	JNL#1 CPD#1
147	JNL-0030	KFCA32805-W	W	システムジャーナルサービス定義	jnl_auto_unload_path	—	—	指定ディレクトリにアンロードジャーナルファイルが存在する	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
148	JNL-0031	KFCA00264-W	C	システムジャーナルサービス定義	jnl_max_file_dispersion	—	jnl_min_file_dispersion	jnl_min_file_dispersion > jnl_max_file_dispersion	JNL#1 CPD#1
149	JNL-0050	KFCA32810-W	E	システムジャーナルサービス定義	jnladdfg	—	—	-g オプションの指定なし	JNL#1 CPD#1
150	JNL-0051	KFCA32811-W	E	システムジャーナルサービス定義	jnladdfg	—	—	-g オプションの引数の指定なし	JNL#1 CPD#1
151	JNL-0052	KFCA32812-W	E	システムジャーナルサービス定義	jnladdfg	—	—	jnladdfg の指定数 > 256	JNL#1 CPD#1
152	JNL-0053	KFCA32813-W	E	システムジャーナルサービス定義	jnladdfg	—	—	jnladdfg の指定数 < 2	JNL#1 CPD#1
153	JNL-0054	KFCA32814-W	E	システムジャーナルサービス定義	jnladdfg	—	—	ONL 指定の jnladdfg の指定数 < 2	JNL#1 CPD#1
154	JNL-0055	KFCA32815-W	W	システムジャーナルサービス定義	jnladdfg	—	—	同一ファイルグループ名の jnladdfg の指定数 > 1	JNL#1 CPD#1
155	JNL-0056	KFCA32816-W	E	システムジャーナルサービス定義	jnladdfg	—	jnladdpf	jnladdpf が定義されていないファイルグループがある	JNL#1 CPD#1
156	JNL-0057	KFCA32820-W	E	システムジャーナルサービス定義	jnladdpf	—	—	-g オプションの指定なし	JNL#1 CPD#1
157	JNL-0058	KFCA32821-W	E	システムジャーナルサービス定義	jnladdpf	—	—	-g オプションの引数の指定なし	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
158	JNL-0059	KFCA32822-W	E	システムジャーナルサービス定義	jnladdpf	—	—	-a オプションの指定なし	JNL#1 CPD#1
159	JNL-0060	KFCA32823-W	E	システムジャーナルサービス定義	jnladdpf	—	—	-a オプションまたは-b オプションの引数の指定なし	JNL#1 CPD#1
160	JNL-0061	KFCA32824-W	E	システムジャーナルサービス定義	jnladdpf	—	jnl_dual	(jnl_dual == Y) && (jladdpf の-b オプションの指定なし)	JNL#1 CPD#1
161	JNL-0062	KFCA32825-W	C	システムジャーナルサービス定義	jnladdpf	—	jnl_dual	(jnl_dual == N) && (jnladdpf の-b オプションの指定あり)	JNL#1 CPD#1
162	JNL-0063	KFCA32826-W	E	システムジャーナルサービス定義	jnladdpf	—	jnladdfg	jnladdfg に定義されていないファイルグループ名が jnladdpf に定義されている	JNL#1 CPD#1
163	JNL-0064	KFCA32827-W	W	システムジャーナルサービス定義	jnladdpf	—	jnl_max_file_dispersion	jnl_max_file_dispersion = 1 && 同一ファイルグループ名の jnladdpf 定義コマンドの指定数 > jnl_max_file_dispersion	JNL#1 CPD#1
164	JNL-0065	KFCA32828-W	E	システムジャーナルサービス定義	jnladdpf	—	—	ジャーナルファイル名の重複	JNL#1 CPD#1
165	JNL-0066	KFCA32829-W	E	システムジャーナルサービス定義	jnladdpf	—	—	物理ファイルはジャーナルファイルではない	JNL#1 CPD#1
166	JNL-0067	KFCA32830-W	W	システムジャーナルサービス定義	jnladdpf	—	—	ジャーナルファイルが他ノードのジャーナルファイルである	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
167	JNL-0068	KFCA32831-W	E	システムジャーナルサービス定義	jnladdpf	—	—	ジャーナルファイルのレコード数 < ↑ (jnl_max_data size 値) + 336) ／ 4096 ↑ + 12	JNL#1 CPD#1
168	JNL-0069	KFCA32832-W	E	システムジャーナルサービス定義	jnladdpf	—	—	次に示すどちらかの条件 指定したファイルシステムが キャラクタ型スペシャルファイルではない 指定したファイルシステムに対応する装置がない	JNL#1 CPD#1
169	JNL-0070	KFCA32833-W	E	システムジャーナルサービス定義	jnladdpf	—	—	OpenTP1 ファイルシステムのある装置が未初期化	JNL#1 CPD#1
170	JNL-0071	KFCA32834-W	E	システムジャーナルサービス定義	jnladdpf	—	—	OpenTP1 ファイルシステムのバージョン不一致	JNL#1 CPD#1
171	JNL-0072	KFCA32835-W	E	システムジャーナルサービス定義	jnladdpf	—	—	メモリ不足の発生	JNL#1 CPD#1
172	JNL-0073	KFCA32836-W	E	システムジャーナルサービス定義	jnladdpf	—	—	オープンファイル数がシステム の上限値を超過	JNL#1 CPD#1
173	JNL-0074	KFCA32837-W	E	システムジャーナルサービス定義	jnladdpf	—	—	指定した OpenTP1 ファイルシステムまたはジャーナル ファイルに対するアクセス権限がない	JNL#1 CPD#1
174	JNL-0075	KFCA32838-W	E	システムジャーナルサービス定義	jnladdpf	—	—	ファイルが存在しない	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
175	JNL-0076	KFCA32839-W	E	システムジャーナルサービス定義	jnladdpf	—	—	ロックセグメント不足が発生	JNL#1 CPD#1
176	JNL-0077	KFCA32840-W	E	システムジャーナルサービス定義	jnladdpf	—	—	指定されたファイルはジャーナルファイルとして使用できない	JNL#1 CPD#1
177	JNL-0078	KFCA32841-W	E	システムジャーナルサービス定義	jnladdpf	—	—	次に示すどちらかの条件 物理ファイルの管理情報の読み込み失敗 管理情報が破壊されている	JNL#1 CPD#1
178	JNL-0079	KFCA32842-W	W	システムジャーナルサービス定義	jnladdpf	—	—	jnl_unload_check=Y の場合、ファイルグループがアンロードされていない	JNL#1 CPD#1
179	JNL-0080	KFCA32843-W	C	システムジャーナルサービス定義	jnladdpf	—	—	物理ファイル上の状態が現用である	JNL#1 CPD#1
180	JNL-0081	KFCA32844-W	C	システムジャーナルサービス定義	jnladdpf	—	jnl_max_file_dispersion	同一ファイルグループに指定する要素ファイル数 > jnl_max_file_dispersion	JNL#1 CPD#1
181	JNL-0082	KFCA32845-W	C	システムジャーナルサービス定義	jnladdpf	—	jnl_min_file_dispersion	同一ファイルグループに指定する要素ファイル数 < jnl_min_file_dispersion	JNL#1 CPD#1
182	JNL-0083	KFCA32846-W	C	システムジャーナルサービス定義	jnladdpf	—	jnl_max_file_dispersion	jnl_max_file_dispersion > 1 && jnladdpf の-e オプション指定なし	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
183	JNL-0084	KFCA32847-W	W	システムジャーナルサービス定義	jnladdpf	—	—	同一要素ファイル名の jnladdpf 定義コマンドの指定数 > 1	JNL#1 CPD#1
184	JNL-0086	KFCA32849-W	E	システムジャーナルサービス定義	jnladdpf	—	—	-e オプション引数が指定されていない。	JNL#1 CPD#1
185	JNL-0090	KFCA00282-W	C	jnl	jnl_arc_terminate_timeout	—	—	(OS == Windows OS == Linux) && jnl_arc_terminate_timeout 指定あり	JNL#1 CPD#1
186	JNL-0090	KFCA00282-W	C	jnl	jnl_arc_ipc_buffer_size	—	—	(OS == Windows OS == Linux) && jnl_arc_ipc_buffer_size 指定あり	JNL#1 CPD#1
187	JNL-0090	KFCA00282-W	C	システムジャーナルサービス定義	jnl_arc_name	—	—	(OS == Windows OS == Linux) && jnl_arc_name 指定あり	JNL#1 CPD#1
188	JNL-0090	KFCA00282-W	C	システムジャーナルサービス定義	jnl_arc_buffer_size	—	—	(OS == Windows OS == Linux) && jnl_arc_buffer_size 指定あり	JNL#1 CPD#1
189	JNL-0090	KFCA00282-W	C	システムジャーナルサービス定義	jnl_arc_max_datasize	—	—	(OS == Windows OS == Linux) && jnl_arc_max_datasize 指定あり	JNL#1 CPD#1
190	JNL-0090	KFCA00282-W	C	システムジャーナル	jnl_arc_termina	—	—	(OS == Windows OS == Linux)	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
190	JNL-0090	KFCA00282-W	C	サービス定義	te_check	—	—	&&jnl_arc_terminate_check 指定あり	JNL#1 CPD#1
191	JNL-0090	KFCA00282-W	C	システムジャーナルサービス定義	jnl_arc_rec_kind	—	—	OS が Windows または Linux の場合で指定されている	JNL#1 CPD#1
192	JNL-0090	KFCA00282-W	C	システムジャーナルサービス定義	jnl_arc_uj_code	—	—	OS が Windows または Linux の場合で指定されている	JNL#1 CPD#1
193	JNL-0090	KFCA00282-W	C	システムジャーナルサービス定義	jnl_arc_check_level	—	—	OS が Windows または Linux の場合で指定されている	JNL#1 CPD#1
194	JNL-0090	KFCA00282-W	C	システムジャーナルサービス定義	jnl_arc_trn_stat	—	—	OS が Windows または Linux の場合で指定されている	JNL#1 CPD#1
195	JNL-0100	KFCA00285-W	E	jnl	jnldfs	—	—	jnldfs 指定なし	JNL#1 CPD#1
196	JNL-0101	KFCA00285-W	E	システムジャーナルサービス定義	jnladdfg	—	—	jnladdfg 指定なし	JNL#1 CPD#1
197	JNL-0102	KFCA00285-W	E	システムジャーナルサービス定義	jnladdpf	—	—	jnladdpf 指定なし	JNL#1 CPD#1
198	JNL-0103	KFCA00286-W	E	jnl	—	betranrc	jnl_fileless_option	jnl_fileless_option=N &&ジャーナルサービス定義ファイル(jnl)が存在しない	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
199	JNL-0104	KFCA00286-W	E	システムジャーナルサービス定義	—	betranrc	jnl_fileless_option	jnl_fileless_option=N && システムジャーナルサービス定義ファイルが存在しない	JNL#1 CPD#1
200	JNL-0110	KFCA32850-W	E	システムジャーナルサービス定義	jnl_arc_name	—	—	!(OS == Windows OS == Linux) && (リソースグループ名の文字数 < 1 リソースグループ名の文字数 > 8) (ノード識別子の文字数 != 4)	JNL#1 CPD#1
201	JNL-0111	KFCA32851-W	E	システムジャーナルサービス定義	jnl_arc_rec_kind	—	—	!(OS == Windows OS == Linux) && (!(a) !(c) !(f) !(g) !(i) !(m) !(o) !(s) !(u))	JNL#1 CPD#1
202	JNL-0112	KFCA32852-W	E	システムジャーナルサービス定義	jnl_arc_uj_code	—	—	!(OS == Windows OS == Linux) && (uj コード < 0) (uj コード > 255)	JNL#1 CPD#1
203	JNL-0120	KFCA32806-W	W	システムジャーナルサービス定義	jnl_auto_unload_path	—	—	指定ディレクトリのパスが絶対パスではない	JNL#1 CPD#1
204	JNL-0121	KFCA32807-W	W	システムジャーナルサービス定義	jnl_auto_unload_path	—	—	指定ディレクトリのパス文字列 > 80 バイト	JNL#1 CPD#1
205	JNL-0122	KFCA32808-W	W	システムジャーナルサービス定義	jnl_auto_unload_path	—	—	すべての指定ディレクトリが使用不可能	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
206	JNL-0123	KFCA00272-W	C	betranrc	jnl_prf_event_trace_level	—	—	(jnl_prf_event_trace_level & 00000001) != 00000001	JNL#1
207	JNL-0124	KFCA26780-W	C	_jl	prf_file_size	—	—	prf_file_size == 1024	JNL#1
208	JNL-0125	KFCA00272-W	C	_jl	prf_trace_backup	—	—	prf_trace_backup == N	JNL#1
209	LCK-0001	KFCA00421-W	C	lck	全体	lck	lck_limit_for*	(lck_limit_foruser 指定値 + lck_limit_fordam 指定値 + lck_limit_fortam 指定値 + lck_limit_formqa 指定値) == 0 ただし、 dam_conf==N の場合は lck_limit_fordam=0 tam_conf==N の場合は lck_limit_fortam=0	ロックサービス定義の論理チェックは行いません。
210	LCK-0002	KFCA00265-W	C	lck	lck_limit_foruser	—	—	lck_limit_foruser 指定あり	—
211	LCK-0003	KFCA00272-W	C	lck	lck_limit_fordam	sysconf	dam_conf	((dam_conf == N) (dam_conf 指定なし)) && ((lck_limit_fordam > 0) (lck_limit_fordam 指定なし))	—
212	LCK-0004	KFCA00265-W	C	lck	lck_limit_fordam	sysconf	dam_conf	(dam_conf == Y)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
212	LCK-0004	KFCA00265-W	C	lck	lck_limit_fordam	sysconf	dam_conf	&& (lck_limit_fordam 指定あり)	—
213	LCK-0005	KFCA00272-W	C	lck	lck_limit_fortam	sysconf	tam_conf	((tam_conf == N) (tam_conf 指定なし)) && ((lck_limit_fortam > 0) (lck_limit_fortam 指定なし))	—
214	LCK-0006	KFCA00265-W	C	lck	lck_limit_fortam	sysconf	tam_conf	(tam_conf == Y) && (lck_limit_fortam 指定あり)	—
215	LCK-0007	KFCA00278-W	C	lck	lck_limit_formqa	sysconf	mqa_conf	((mqa_conf == N) (mqa_conf 指定なし)) && (lck_limit_formqa > 0)	—
216	LCK-0008	KFCA00265-W	C	lck	lck_limit_formqa	sysconf	mqa_conf	mqa_conf == Y	—
217	LCK-0009	KFCA00265-W	C	lck	lck_wait_timeout	—	—	lck_wait_timeout 指定あり	—
218	LCK-0010	KFCA00272-W	C	lck	lck_deadlock_info	env	default_value_option	(lck_deadlock_info == N) ((default_value_option == 0) && (lck_deadlock_info 指定なし))	—
219	LCK-0011	KFCA00278-W	C	lck	lck_deadlock_info_remove	lck	lck_deadlock_info	(lck_deadlock_info_remove 指定あり)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
219	LCK-0011	KFCA00278-W	C	lck	lck_deadlock_info_remove	lck	lck_deadlock_info	&&(lck_deadlock_info == N)	—
220	LCK-0012	KFCA00279-W	C	lck	lck_deadlock_info_remove	lck env	lck_deadlock_info default_value_option	(lck_deadlock_info_remove 指定あり) &&((default_value_option == 0) &&(lck_deadlock_info 指定なし))	—
221	LCK-0013	KFCA00272-W	C	lck	lck_deadlock_info_remove	—	—	(lck_deadlock_info == Y) &&((lck_deadlock_info_remove == no) (lck_deadlock_info_remove 指定なし))	—
222	LCK-0014	KFCA00278-W	C	lck	lck_deadlock_info_remove_level	lck	lck_deadlock_info	(lck_deadlock_info_remove_level 指定あり) &&(lck_deadlock_info == N)	—
223	LCK-0015	KFCA00279-W	C	lck	lck_deadlock_info_remove_level	lck env	lck_deadlock_info default_value_option	(lck_deadlock_info_remove_level 指定あり) &&((default_value_option == 0) && (lck_deadlock_info 指定なし))	—
224	LCK-0016	KFCA00278-W	C	lck	lck_deadlock_info_remove_level	lck	lck_deadlock_info_remove	(lck_deadlock_info_remove_level 指定あり) &&(lck_deadlock_info_remove == no)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
225	LCK-0017	KFCA00279-W	C	lck	lck_deadlock_info_remove_level	lck	lck_deadlock_info_remove	(lck_deadlock_info_remove_level 指定あり) && (lck_deadlock_info_remove 指定なし)	—
226	LCK-0018	KFCA00272-W	W	lck	lck_release_detect	—	—	(lck_release_detect == interval)	—
227	LCK-0019	KFCA00278-W	C	lck	lck_release_detect_interval	lck	lck_release_detect	(lck_release_detect_interval 指定あり) && (lck_release_detect == pipe)	—
228	LCK-0020	KFCA00279-W	C	lck	lck_release_detect_interval	lck	lck_release_detect	(lck_release_detect_interval 指定あり) && (lck_release_detect 指定なし)	—
229	LCK-0021	KFCA00422-W	C	ユーザサービス定義	deadlock_priority	lck	lck_limit_foruser lck_limit_forform lck_limit_forform lck_limit_forforma lck_limit_forforma	(lck_limit_foruser 指定値 + lck_limit_forform 指定値 + lck_limit_forform 指定値 + lck_limit_forforma 指定値) == 0 ただし, dam_conf==N の場合は, lck_limit_forform=0 tam_conf==N の場合は, lck_limit_forform=0	—
230	LCK-0022	KFCA00422-W	C	ユーザサービス定義	lck_wait_priority	lck	lck_limit_foruser	(lck_limit_foruser 指定値 + lck_limit_forform 指定値 +	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
230	LCK-0022	KFCA00422-W	C	ユーザー ビス定義	lck_wai t_priorit y	lck	lck_limit_fordam lck_limit_fordam lck_limit_fordam lck_limit_fordam	lck_limit_fordam 指定値 + lck_limit_fordam 指定値) == 0 ただし、 dam_conf==N の場合は lck_limit_fordam=0 tam_conf==N の場合は lck_limit_fordam=0	—
231	LCK-0023	KFCA00272-W	C	lck	lck_prf_ trace_level	—	—	(lck_prf_trace_level & 00000001) != 00000001	—
232	LCK-0024	KFCA00423-W	C	_lk	prf_file_ size	—	—	prf_file_size < デフォルト値 (5120)	—
233	LOG-0001	KFCA00272-W	C	log	log_msg_ console	log	—	log_msg_console == N	—
234	LOG-0002	KFCA00278-W	C	log	log_msg_ allno	log	log_msg_ console	(log_msg_console == N) && (log_msg_allno 指定あり) && (log_msg_allno == Y)	—
235	LOG-0002	KFCA00278-W	C	log	log_msg_ prcid	log	log_msg_ console	(log_msg_console == N) && (log_msg_prcid 指定あり) && (log_msg_prcid == Y)	—
236	LOG-0002	KFCA00278-W	C	log	log_msg_ prcno	log	log_msg_ console	(log_msg_console == N) && (log_msg_prcno 指定あり)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
236	LOG-0002	KFCA00278-W	C	log	log_msg_prcno	log	log_msg_console	&& (log_msg_prcno == Y)	—
237	LOG-0002	KFCA00278-W	C	log	log_msg_sysid	log	log_msg_console	(log_msg_console == N) && (log_msg_sysid 指定あり) && (log_msg_sysid == Y)	—
238	LOG-0002	KFCA00278-W	C	log	log_msg_date	log	log_msg_console	(log_msg_console == N) && (log_msg_date 指定あり) && (log_msg_date == Y)	—
239	LOG-0002	KFCA00278-W	C	log	log_msg_time	log	log_msg_console	(log_msg_console == N) && (log_msg_time 指定あり) && (log_msg_time == Y)	—
240	LOG-0002	KFCA00278-W	C	log	log_msg_hostname	log	log_msg_console	(log_msg_console == N) && (log_msg_hostname) && (log_msg_hostname == Y)	—
241	LOG-0002	KFCA00278-W	C	log	log_msg_pgmid	log	log_msg_console	(log_msg_console == N) && (log_msg_pgmid 指定あり) && (log_msg_pgmid == Y)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
242	LOG-0003	KFCA00272-W	C	log	log_msg_date	log	log_msg_console	(log_msg_console == Y) && (log_msg_date 指定あり) && (log_msg_date == N)	—
243	LOG-0003	KFCA00272-W	C	log	log_msg_time	log	log_msg_console	(log_msg_console == Y) && (log_msg_time 指定あり) && (log_msg_time == N)	—
244	LOG-0004	KFCA00282-W	C	log	log_netm_out	—	—	log_netm_out 指定あり	OS が Windows または Linux の場合だけチェック対象です。
245	LOG-0004	KFCA00282-W	C	log	log_netm_allno	—	—	log_netm_allno 指定あり	OS が Windows または Linux の場合だけチェック対象です。
246	LOG-0004	KFCA00282-W	C	log	log_netm_prcid	—	—	log_netm_prcid 指定あり	OS が Windows または Linux の場合だけチェック対象です。
247	LOG-0004	KFCA00282-W	C	log	log_netm_prcno	—	—	log_netm_prcno 指定あり	OS が Windows または Linux の場合だけチェック対象です。
248	LOG-0004	KFCA00282-W	C	log	log_netm_sysid	—	—	log_netm_sysid 指定あり	OS が Windows または Linux の場合だけチェック対象です。
249	LOG-0004	KFCA00282-W	C	log	log_netm_date	—	—	log_netm_date 指定あり	OS が Windows または Linux の場合

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
249	LOG-0004	KFCA00282-W	C	log	log_netm_date	—	—	log_netm_date 指定あり	だけチェック対象です。
250	LOG-0004	KFCA00282-W	C	log	log_netm_time	—	—	log_netm_time 指定あり	OS が Windows または Linux の場合だけチェック対象です。
251	LOG-0004	KFCA00282-W	C	log	log_netm_hostname	—	—	log_netm_hostname 指定あり	OS が Windows または Linux の場合だけチェック対象です。
252	LOG-0004	KFCA00282-W	C	log	log_netm_pgmiid	—	—	log_netm_pgmiid 指定あり	OS が Windows または Linux の場合だけチェック対象です。
253	LOG-0005	KFCA00280-W	C	log	log_netm_allno	log	log_netm_out	(log_netm_out == N) && (log_netm_allno 指定あり) && (log_netm_allno == Y)	OS が Windows または Linux の場合はチェック対象外です。
254	LOG-0005	KFCA00280-W	C	log	log_netm_prcid	log	log_netm_out	(log_netm_out == N) && (log_netm_prcid 指定あり) && (log_netm_prcid == Y)	OS が Windows または Linux の場合はチェック対象外です。
255	LOG-0005	KFCA00280-W	C	log	log_netm_prcno	log	log_netm_out	(log_netm_out == N) && (log_netm_prcno 指定あり) && (log_netm_prcno == Y)	OS が Windows または Linux の場合はチェック対象外です。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
256	LOG-005	KFCA00280-W	C	log	log_netm_sysid	log	log_netm_out	(log_netm_out == N) && (log_netm_sysid 指定あり) && (log_netm_sysid == Y)	OS が Windows または Linux の場合はチェック対象外です。
257	LOG-005	KFCA00280-W	C	log	log_netm_date	log	log_netm_out	(log_netm_out == N) && (log_netm_date 指定あり) && (log_netm_date == Y)	OS が Windows または Linux の場合はチェック対象外です。
258	LOG-005	KFCA00280-W	C	log	log_netm_time	log	log_netm_out	(log_netm_out == N) && (log_netm_time 指定あり) && (log_netm_time == Y)	OS が Windows または Linux の場合はチェック対象外です。
259	LOG-005	KFCA00280-W	C	log	log_netm_hostname	log	log_netm_out	(log_netm_out == N) && (log_netm_hostname 指定あり) && (log_netm_hostname == Y)	OS が Windows または Linux の場合はチェック対象外です。
260	LOG-005	KFCA00280-W	C	log	log_netm_pgmid	log	log_netm_out	(log_netm_out == N) && (log_netm_pgmid 指定あり) && (log_netm_pgmid == Y)	OS が Windows または Linux の場合はチェック対象外です。
261	LOG-006	KFCA00282-W	C	log	log_jpl_allno	—	—	log_jpl_allno 指定あり	OS が Windows の場合だけチェック対象です。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
262	LOG-0006	KFCA00282-W	C	log	log_jpl_prcid	—	—	log_jpl_prcid 指定あり	OS が Windows の場合だけチェック対象です。
263	LOG-0006	KFCA00282-W	C	log	log_jpl_prcno	—	—	log_jpl_prcno 指定あり	OS が Windows の場合だけチェック対象です。
264	LOG-0006	KFCA00282-W	C	log	log_jpl_sysid	—	—	log_jpl_sysid 指定あり	OS が Windows の場合だけチェック対象です。
265	LOG-0006	KFCA00282-W	C	log	log_jpl_date	—	—	log_jpl_date 指定あり	OS が Windows の場合だけチェック対象です。
266	LOG-0006	KFCA00282-W	C	log	log_jpl_time	—	—	log_jpl_time 指定あり	OS が Windows の場合だけチェック対象です。
267	LOG-0006	KFCA00282-W	C	log	log_jpl_hostname	—	—	log_jpl_hostname 指定あり	OS が Windows の場合だけチェック対象です。
268	LOG-0006	KFCA00282-W	C	log	log_jpl_pgmid	—	—	log_jpl_pgmid 指定あり	OS が Windows の場合だけチェック対象です。
269	LOG-0006	KFCA00282-W	C	log	log_jerr_rint	—	—	log_jerr_rint 指定あり	OS が Windows の場合だけチェック対象です。
270	LOG-0007	KFCA00280-W	C	log	log_jpl_allno	betranrc	jpl_use	(jpl_use == N) && (log_jpl_allno 指定あり) && (log_jpl_allno == Y)	OS が Windows の場合はチェック対象外です。
271	LOG-0007	KFCA00280-W	C	log	log_jpl_prcid	betranrc	jpl_use	(jpl_use == N)	OS が Windows の場合はチェック対象外です。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
271	LOG-007	KFCA00280-W	C	log	log_jpl_prcid	betranrc	jpl_use	&&(log_jpl_prcid 指定あり) &&(log_jpl_prcid == Y)	OS が Windows の場合はチェック対象外です。
272	LOG-007	KFCA00280-W	C	log	log_jpl_prcno	betranrc	jpl_use	(jpl_use == N) &&(log_jpl_prcno) &&(log_jpl_prcno == Y)	OS が Windows の場合はチェック対象外です。
273	LOG-007	KFCA00280-W	C	log	log_jpl_sysid	betranrc	jpl_use	(jpl_use == N) &&(log_jpl_sysid 指定あり) &&(log_jpl_sysid == Y)	OS が Windows の場合はチェック対象外です。
274	LOG-007	KFCA00280-W	C	log	log_jpl_date	betranrc	jpl_use	(jpl_use == N) &&(log_jpl_date 指定あり) &&(log_jpl_date == Y)	OS が Windows の場合はチェック対象外です。
275	LOG-007	KFCA00280-W	C	log	log_jpl_time	betranrc	jpl_use	(jpl_use == N) &&(log_jpl_time 指定あり) &&(log_jpl_time == Y)	OS が Windows の場合はチェック対象外です。
276	LOG-007	KFCA00280-W	C	log	log_jpl_hostname	betranrc	jpl_use	(jpl_use == N) &&(log_jpl_hostname 指定あり) &&(log_jpl_hostname == Y)	OS が Windows の場合はチェック対象外です。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
277	LOG-0007	KFCA00280-W	C	log	log_jpl_pgmid	betranrc	jpl_use	(jpl_use == N) && (log_jpl_pgmid 指定あり) && (log_jpl_pgmid == Y)	OS が Windows の場合はチェック対象外です。
278	LOG-0007	KFCA00280-W	C	log	log_jerr_rint	betranrc	jpl_use	(jpl_use == N) && (log_jerr_rint 指定あり) && (log_jerr_rint >= 1)	OS が Windows の場合はチェック対象外です。
279	LOG-0008	KFCA00280-W	C	log	log_notify_allno	log	log_notify_out	(log_notify_out == N) && (log_notify_allno 指定あり) && (log_notify_allno == Y)	OS が Linux の場合はチェック対象外です。
280	LOG-0008	KFCA00280-W	C	log	log_notify_prcid	log	log_notify_out	(log_notify_out == N) && (log_notify_prcid 指定あり) && (log_notify_prcid == Y)	OS が Linux の場合はチェック対象外です。
281	LOG-0008	KFCA00280-W	C	log	log_notify_prcno	log	log_notify_out	(log_notify_out == N) && (log_notify_prcno 指定あり) && (log_notify_prcno == Y)	OS が Linux の場合はチェック対象外です。
282	LOG-0008	KFCA00280-W	C	log	log_notify_sysid	log	log_notify_out	(log_notify_out == N) && (log_notify_sysid 指定あり)	OS が Linux の場合はチェック対象外です。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
282	LOG-0008	KFCA00280-W	C	log	log_notify_sysid	log	log_notify_out	&& (log_notify_sysid == Y)	OS が Linux の場合はチェック対象外です。
283	LOG-0008	KFCA00280-W	C	log	log_notify_date	log	log_notify_out	(log_notify_out == N) && (log_notify_date) && (log_notify_date == Y)	OS が Linux の場合はチェック対象外です。
284	LOG-0008	KFCA00280-W	C	log	log_notify_time	log	log_notify_out	(log_notify_out == N) && (log_notify_time 指定あり) && (log_notify_time == Y)	OS が Linux の場合はチェック対象外です。
285	LOG-0008	KFCA00280-W	C	log	log_notify_hostname	log	log_notify_out	(log_notify_out == N) && (log_notify_hostname 指定あり) && (log_notify_hostname == Y)	OS が Linux の場合はチェック対象外です。
286	LOG-0008	KFCA00280-W	C	log	log_notify_pgmid	log	log_notify_out	(log_notify_out == N) && (log_notify_pgmid 指定あり) && (log_notify_pgmid == Y)	OS が Linux の場合はチェック対象外です。
287	LOG-0009	KFCA00282-W	C	log	log_syslog_out	—	—	log_syslog_out 指定あり	OS が Windows の場合だけチェック対象です。
288	LOG-0009	KFCA00282-W	C	log	log_syslog_allno	—	—	log_syslog_allno 指定あり	OS が Windows の場合だけチェック対象です。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
289	LOG-0009	KFCA00282-W	C	log	log_syslog_prcid	—	—	log_syslog_prcid 指定あり	OS が Windows の場合だけチェック対象です。
290	LOG-0009	KFCA00282-W	C	log	log_syslog_prcno	—	—	log_syslog_prcno 指定あり	OS が Windows の場合だけチェック対象です。
291	LOG-0009	KFCA00282-W	C	log	log_syslog_sysid	—	—	log_syslog_sysid 指定あり	OS が Windows の場合だけチェック対象です。
292	LOG-0009	KFCA00282-W	C	log	log_syslog_date	—	—	log_syslog_date 指定あり	OS が Windows の場合だけチェック対象です。
293	LOG-0009	KFCA00282-W	C	log	log_syslog_time	—	—	log_syslog_time 指定あり	OS が Windows の場合だけチェック対象です。
294	LOG-0009	KFCA00282-W	C	log	log_syslog_hostname	—	—	log_syslog_hostname 指定あり	OS が Windows の場合だけチェック対象です。
295	LOG-0009	KFCA00282-W	C	log	log_syslog_pgmid	—	—	log_syslog_pgmid 指定あり	OS が Windows の場合だけチェック対象です。
296	LOG-0009	KFCA00282-W	C	log	log_syslog_append_nodeid	—	—	log_syslog_append_nodeid 指定あり	OS が Windows の場合だけチェック対象です。
297	LOG-0009	KFCA00282-W	C	log	log_syslog_synchro	—	—	log_syslog_synchro 指定あり	OS が Windows の場合だけチェック対象です。
298	LOG-0009	KFCA00282-W	C	log	log_syslog_elist	—	—	log_syslog_elist 指定あり	OS が Windows の場合だけチェック対象です。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
299	LOG-0009	KFCA00282-W	C	log	log_syslog_elist_rint	—	—	log_syslog_elist_rint 指定あり	OS が Windows の場合だけチェック対象です。
300	LOG-0010	KFCA00272-W	C	log	log_syslog_out	log	—	log_syslog_out == 0	OS が Windows の場合はチェック対象外です。
301	LOG-0011	KFCA00280-W	C	log	log_syslog_out	log	DCSYSLOGOUT	(log_syslog_out >= 1) && (DCSYSLOGOUT == 1 指定なし)	OS が Windows の場合はチェック対象外です。
302	LOG-0011	KFCA00278-W	C	log	log_syslog_allno	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_allno == Y)	OS が Windows の場合はチェック対象外です。
303	LOG-0011	KFCA00278-W	C	log	log_syslog_prcid	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_prcid == Y)	OS が Windows の場合はチェック対象外です。
304	LOG-0011	KFCA00278-W	C	log	log_syslog_prcno	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_prcno == Y)	OS が Windows の場合はチェック対象外です。
305	LOG-0011	KFCA00278-W	C	log	log_syslog_sysid	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_sysid == Y)	OS が Windows の場合はチェック対象外です。
306	LOG-0011	KFCA00278-W	C	log	log_syslog_date	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_date == Y)	OS が Windows の場合はチェック対象外です。
307	LOG-0011	KFCA00278-W	C	log	log_syslog_time	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_time == Y)	OS が Windows の場合はチェック対象外です。
308	LOG-0011	KFCA00278-W	C	log	log_syslog_hostname	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_hostname == Y)	OS が Windows の場合はチェック対象外です。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
309	LOG-0011	KFCA00278-W	C	log	log_syslog_pgm_id	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_pgm_id == Y)	OS が Windows の場合はチェック対象外です。
310	LOG-0011	KFCA00278-W	C	log	log_syslog_append_nodeid	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_append_nodeid == Y)	OS が Windows の場合はチェック対象外です。
311	LOG-0011	KFCA00278-W	C	log	log_syslog_synchro	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_synchro == Y)	OS が Windows の場合はチェック対象外です。
312	LOG-0011	KFCA00278-W	C	log	log_syslog_elist	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_elist > 0)	OS が AIX または Linux の場合だけチェック対象です。
313	LOG-0011	KFCA00278-W	C	log	log_syslog_elist_rint	log	log_syslog_elist_rint	(log_syslog_elist_rint == 0) && (log_syslog_elist > 0)	OS が AIX または Linux の場合だけチェック対象です。
314	LOG-0011	KFCA00278-W	C	log	log_syslog_elist_rint	log	log_syslog_out	(log_syslog_out == 0) && (log_syslog_elist_rint > 0)	OS が AIX または Linux の場合だけチェック対象です。
315	LOG-0011	KFCA00278-W	C	log	log_syslog_elist_rint	log	log_syslog_elist	(log_syslog_elist == 0) && (log_syslog_elist_rint > 0)	OS が AIX または Linux の場合だけチェック対象です。
316	LOG-0011	KFCA00278-W	C	log	DCSYSLOGCTYPE	log	log_syslog_out	(log_syslog_out == 0) && (DCSYSLOGCTYPE == sjis)	OS が Linux の場合だけチェック対象です。
317	LOG-0012	KFCA00282-W	C	log	log_notify_out	—	—	log_notify_out 指定あり	OS が Linux の場合だけチェック対象です。
318	LOG-0012	KFCA00282-W	C	log	log_notify_allno	—	—	log_notify_allno 指定あり	OS が Linux の場合だけチェック対象です。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
319	LOG-0012	KFCA00282-W	C	log	log_notify_prcid	—	—	log_notify_prcid 指定あり	OS が Linux の場合だけチェック対象です。
320	LOG-0012	KFCA00282-W	C	log	log_notify_prcno	—	—	log_notify_prcno 指定あり	OS が Linux の場合だけチェック対象です。
321	LOG-0012	KFCA00282-W	C	log	log_notify_sysid	—	—	log_notify_sysid 指定あり	OS が Linux の場合だけチェック対象です。
322	LOG-0012	KFCA00282-W	C	log	log_notify_date	—	—	log_notify_date 指定あり	OS が Linux の場合だけチェック対象です。
323	LOG-0012	KFCA00282-W	C	log	log_notify_time	—	—	log_notify_time 指定あり	OS が Linux の場合だけチェック対象です。
324	LOG-0012	KFCA00282-W	C	log	log_notify_hostname	—	—	log_notify_hostname 指定あり	OS が Linux の場合だけチェック対象です。
325	LOG-0012	KFCA00282-W	C	log	log_notify_pgmid	—	—	log_notify_pgmid 指定あり	OS が Linux の場合だけチェック対象です。
326	LOG-0013	KFCA00280-W	C	log	log_audit_path	log	log_audit_out	(log_audit_out == N) && (log_audit_path 指定あり)	—
327	LOG-0013	KFCA00280-W	C	log	log_audit_size	log	log_audit_out	(log_audit_out == N) && (log_audit_size 指定あり)	—
328	LOG-0013	KFCA00280-W	C	log	log_audit_count	log	log_audit_out	(log_audit_out == N) && (log_audit_count 指定あり)	—
329	LOG-0013	KFCA00280-W	C	log usrsrc ユーザサービス定義 rap リスナーサービス定義 rap クライアントマネ	log_audit_message	log	log_audit_out	(log_audit_out == N) && (log_audit_message 指定あり)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
329	LOG-0013	KFCA00280-W	C	ジャサービス定義	log_audit_message	log	log_audit_out	(log_audit_out == N) && (log_audit_message 指定あり)	—
330	LOG-0014	KFCA00267-W	W	log	log_audit_path	log	log_audit_out	(log_audit_out == Y) && (log_audit_path 指定あり) && (log_audit_path に指定したディレクトリが存在しない)	—
331	LOG-0014	KFCA00267-W	W	log	log_audit_path	log	log_audit_out	(log_audit_out == Y) && (log_audit_path 指定あり) && (log_audit_path に指定したパスがディレクトリではない)	—
332	LOG-0014	KFCA00267-W	W	log	log_audit_path	log	log_audit_out	(log_audit_out == Y) && (log_audit_path 指定あり) && (log_audit_path に指定したディレクトリのアクセス権限が不正)	dcauditsetup コマンドで作成するアクセス権の属性と異なっています。
333	NAM-0001	KFCA00266-W	C	betranrc	name_port	—	—	name_port 指定あり	—
334	NAM-0002	KFCA00692-W	C	betranrc	name_port	—	—	name_port 指定なし	—
335	NAM-0003	KFCA00699-W	E	betranrc	all_node	—	—	(IP アドレス == 127.x.x.x) (ホスト名を変換した IP アドレス == 127.x.x.x)	x は 0～255 の任意の値です。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
336	NAM-0004	KFCA00691-W	W	betranrc	all_node	—	—	((all_node 指定 ホスト名 == 自 ホスト名) (all_node 指 定 IP == 自 IP)) && (all_node 指定ポート番号 == name_port)	—
337	NAM-0005	KFCA00690-W	E	betranrc	all_node	betranrc	rpc_multi_tpl_in_same_host	(OS != Windows) && ((rpc_multi_tpl _in_same_host != Y) && ((自ホスト 名) (自 IP アド レス)))	OS が Windows の場 合はチェック対 象外です。
338	NAM-0006	KFCA00278-W	C	betranrc	all_node	betranrc	name_domain_file_use	((name_domain_f ile_use == Y) && all_node 指定あり	—
339	NAM-0008	KFCA00693-W	C	betranrc	name_notify	—	—	name_notify == Y	—
340	NAM-0009	KFCA00694-W	E	betranrc	name_notify	—	—	(name_notify == Y) && (all_node, all_node_ex に 同一 IP が複数指 定)	—
341	NAM-0010	KFCA00699-W	E	betranrc	all_node_ex	—	—	(IP アドレス == 127.x.x.x) (ホスト名を変換 した IP == 127.x.x.x)	x は 0～255 の 任意の値です。
342	NAM-0011	KFCA00691-W	W	betranrc	all_node_ex	—	—	((all_node_ex 指定ホスト名 == 自ホスト名) (all_node_ex 指定 IP == 自 IP)) && (all_node_ex 指	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
342	NAM-0011	KFCA00691-W	W	betranrc	all_node_ex	—	—	定ポート番号 == name_port)	—
343	NAM-0012	KFCA00690-W	E	betranrc	all_node_ex	betranrc	rpc_multi_tpl_in_same_host	(OS != Windows) && ((rpc_multi_tpl_in_same_host != Y) && ((自ホスト名) (自 IP アドレス)))	OS が Windows の場合はチェック対象外です。
344	NAM-0013	KFCA00278-W	C	betranrc	all_node_ex	betranrc	name_domain_file_use	(name_domain_file_use == Y) && all_node_ex 指定あり	—
345	NAM-0014	KFCA00265-W	C	nam	name_total_size	—	—	name_total_size 指定あり	—
346	NAM-0015	KFCA00265-W	C	nam	name_cache_size	—	—	name_cache_size 指定あり	—
347	NAM-0016	KFCA00695-W	C	nam	name_global_lookup	—	—	name_global_lookup == Y	—
348	NAM-0017	KFCA00696-W	W	nam	name_service_extend	betranrc	all_node	((name_service_extend 指定なし) (name_service_extend == 0)) && (all_node 数 > 128)	—
349	NAM-0018	KFCA00694-W	E	nam	name_audit_conf	—	—	((name_audit_conf == 1) (name_audit_conf == 2)) && (all_node, all_node_ex に同一 IP が複数指定)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
350	NAM-0019	KFCA00261-W	W	nam	name_audit_interval	nambetranrc	name_audit_watch_time ipc_connection_interval	(name_audit_conf == 1) && (ipc_connection_interval > name_audit_interval) (name_audit_conf == 2) && (name_audit_watch_time > name_audit_interval)	—
351	NAM-0020	KFCA00276-W	C	nam	name_audit_interval	nam	name_audit_conf	((name_audit_conf 指定なし) (name_audit_conf == 0)) && name_audit_interval 指定あり	—
352	NAM-0021	KFCA00276-W	C	nam	name_audit_watch_time	nam	name_audit_conf	(name_audit_conf != 2) && name_audit_watch_time 指定あり	—
353	NAM-0022	KFCA00697-W	W	nam	name_rpc_control_list	nam	name_audit_conf	(name_rpc_control_list == N) && (name_audit_conf == 0) (name_audit_conf 指定なし))	—
354	NAM-0023	KFCA00698-W	C	nam	name_rpc_control_list	nam	name_audit_interval	((name_rpc_control_list 指定なし) (name_rpc_control_list != N)) && (name_audit_conf != 0) && (name_audit_interval <= 180))	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
355	NAM-0024	KFCA00689-W	E	ドメイン定義ファイル	dcnamnd	—	—	(name_domain_file_use == Y) && (IP アドレス == 127.x.x.x) (ホスト名を変換した IP == 127.x.x.x)	x は 0~255 の任意の値です。
356	NAM-0025	KFCA00688-W	W	ドメイン定義ファイル	dcnamnd	—	—	(name_domain_file_use == Y) && ((指定ホスト名 == 自ホスト名) (指定 IP == 自 IP)) && (指定ポート番号 == name_port)	—
357	NAM-0026	KFCA00687-W	E	ドメイン定義ファイル	dcnamnd	betranrc	rpc_multi_tpl_in_same_host	(name_domain_file_use == Y) && ((OS != Windows) && ((rpc_multi_tpl_in_same_host != Y) && ((自ホスト名) (自 IP アドレス)))))	OS が Windows の場合はチェック対象外です。
358	NAM-0027	KFCA00689-W	E	ドメイン定義ファイル	dcnamndex	—	—	(name_domain_file_use == Y) && (IP アドレス == 127.x.x.x) (ホスト名を変換した IP == 127.x.x.x)	x は 0~255 の任意の値です。
359	NAM-0028	KFCA00688-W	W	ドメイン定義ファイル	dcnamndex	—	—	(name_domain_file_use == Y) && ((指定ホスト名 == 自ホスト名) (指定 IP == 自 IP)) && (指定ポート番号 == name_port)	—
360	NAM-0029	KFCA00687-W	E	ドメイン定義ファイル	dcnamndex	betranrc	rpc_multi_tpl_i	(name_domain_file_use == Y) && ((OS !=	OS が Windows の場

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
360	NAM-0029	KFCA00687-W	E	ドメイン定義ファイル	dcnamindex	betranrc	n_same_host	Windows) && ((rpc_multi_tpl_in_same_host != Y) && ((自ホスト名) (自 IP アドレス))))	合はチェック対象外です。
361	NAM-0030	KFCA00282-W	C	betranrc	domain_master_s_addr	—	—	(OS == Windws) && (domain_masters_addr 指定あり)	OS が Windows の場合、チェック対象の定義はサポートしていません。
362	NAM-0031	KFCA00282-W	C	betranrc	domain_master_s_port	—	—	(OS == Windws) && (domain_masters_port 指定あり)	OS が Windows の場合、チェック対象の定義はサポートしていません。
363	NAM-0032	KFCA00282-W	C	betranrc	domain_use_dns	—	—	(OS == Windws) && (domain_use_dns 指定あり)	OS が Windows の場合、チェック対象の定義はサポートしていません。
364	NAM-0033	KFCA00272-W	W	nam	name_nodeid_check_message	—	—	name_nodeid_check_message == N	—
365	NAM-0034	KFCA00272-W	C	betranrc	nam_prf_trace_level	—	—	((nam_prf_trace_level & 00000001) != 00000001) && ((nam_prf_trace_level & 00000002) != 00000002)	—
366	NAM-0035	KFCA00686-W	W	優先選択ノードの定義ファイル	dcnampr	—	—	(name_domain_file_use==Y) && (all_node に指定のないノードが優先選択ノード定義ファイルに指定されている)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
367	NAM-0036	KFCA00278-W	C	nam	namnlfi l	nam	name_service_mode	(name_service_mode != manager) && (name_service_mode != agent) && namnlfil 指定あり	—
368	NAM-0037	KFCA00279-W	C	nam	namnlfi l	nam	name_service_mode	(name_savice_mode 指定なし) && (namlfil 指定あり)	—
369	—	KFCA33623-E	E	nam	namnlfi l	—	—	(引数指定誤り) (ノードリスト ファイル名指定誤り)	—
370	—	KFCA00218-E	E	nam	namnlfi l	—	—	(!= -n) && (!= -r)	—
371	—	KFCA33615-E	—	nam	namnlfi l	—	—	パスの指定に誤りがある	—
372	—	KFCA33615-E	—	nam	namnlfi l	—	—	ノードリスト ファイルが存在しない	—
373	—	KFCA33615-E	—	nam	namnlfi l	—	—	OpenTP1 ファイルシステム用に初期化されていない	—
374	—	KFCA33615-E	—	nam	namnlfi l	—	—	OpenTP1 ファイルシステムのバージョンが一致しない	—
375	—	KFCA33615-E	—	nam	namnlfi l	—	—	指定したファイルはノードリストファイルとして使用できない	—
376	—	KFCA33615-E	—	nam	namnlfi l	—	—	ノード情報格納数が共用メモリとノードリストファイルで一致しない	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
377	—	KFCA33615-E	—	nam	namnflil	—	—	ノードリストファイルのバージョンが一致しない	—
378	—	KFCA33615-E	—	nam	namnflil	—	—	ファイルに対するアクセス権がない	—
379	—	KFCA33615-E	—	nam	namnflil	—	—	メモリ不足が発生	—
380	—	KFCA33615-E	—	nam	namnflil	—	—	入出力エラーが発生	—
381	—	KFCA33615-E	—	nam	namnflil	—	—	ロックセグメント不足が発生	—
382	—	KFCA33615-E	—	nam	namnflil	—	—	OpenTP1 ファイルシステムのオープンの上限値を超えた	—
383	—	KFCA00216-E	E	betranrc	name_service_mode	—	—	(!= manager) && (!= agent) && (!= normal)	—
384	NAM-0038	KFCA00691-W	E	betranrc	name_manager_node	—	—	((name_manager_node 指定ホスト名 = 自ホスト名) (name_manager_node 指定 IP = 自 IP)) && (name_manager_node 指定ポート番号 = neme_port)	—
385	NAM-0039	KFCA00690-W	E	betranrc	name_manager_node	betranrc	rpc_multi_tpl_in_same_host	(OS != Windows) && (rpc_multi_tpl_in_same_host = Y) && ((自ホスト名) (自 IP アドレス))	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
386	NAM-0040	KFCA00699-W	E	betranrc	name_manager_node	—	—	(IP アドレス = 127.0.0.1) (ホスト名を変換した IP = 127.0.0.1)	—
387	NAM-0041	KFCA00278-W	C	betranrc	name_remove_down_node	betranrc	name_service_mode	(name_remove_down_node 指定あり) && (name_service_mode != manager)	—
388	—	KFCA00216-E	E	betranrc	name_remove_down_node	—	—	(!= Y) && (!= N)	—
389	NAM-0042	KFCA00279-W	C	betranrc	name_remove_down_node	betranrc	name_service_mode	(name_remove_down_node 指定あり) && (name_service_mode 指定なし)	—
390	NAM-0043	KFCA00278-W	C	betranrc	name_node_add_policy	betranrc	name_service_mode	(name_node_add_policy 指定あり) && (name_service_mode != agent)	—
391	—	KFCA00216-E	E	betranrc	name_node_add_policy	—	—	(!= using_only) && (!= all)	—
392	NAM-0044	KFCA00279-W	C	betranrc	name_node_add_policy	betranrc	name_service_mode	(name_node_add_policy 指定あり) && (name_service_mode 指定なし)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
393	NAM-0045	KFCA00278-W	C	betranrc	name_manage_r_node	betranrc	name_service_mode	(name_manager_node 指定あり) && (name_service_mode != agent)	—
394	—	KFCA00606-E	E	betranrc	name_manage_r_node	—	—	(!ホスト名) (!IP アドレス)	—
395	—	KFCA00607-E	E	betranrc	name_manage_r_node	—	—	(指定ポート < 5001) && (指定ポート > 65535)	—
396	NAM-0046	KFCA00279-W	C	betranrc	name_manage_r_node	betranrc	name_service_mode	(name_manager_node 指定あり) && (name_service_mode 指定なし)	—
397	NAM-0047	KFCA00278-W	C	nam	name_start_watch_time	nam	name_service_mode	(name_start_watch_time 指定あり) && (name_service_mode != agent)	—
398	—	KFCA00216-E	E	nam	name_start_watch_time	—	—	(<0) (>1800)	—
399	NAM-0048	KFCA00279-W	C	nam	name_start_watch_time	nam	name_service_mode	(name_start_watch_time 指定あり) && (name_service_mode 指定なし)	—
400	NAM-0049	KFCA00278-E	C	nam	name_start_retry_count	nam	name_service_mode	(name_start_retry_count 指定あり)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
400	NAM-0049	KFCA00278-E	C	nam	name_start_retry_count	nam	name_service_mode	&& (name_service_mode != agent)	—
401	—	KFCA00216-E	E	nam	name_start_retry_count	—	—	(<0) (>60)	—
402	NAM-0050	KFCA00279-E	C	nam	name_start_retry_count	nam	name_service_mode	(name_start_retry_count 指定あり) && (name_service_mode 指定なし)	—
403	NAM-0051	KFCA00278-E	C	nam	name_start_retry_interval	nam	name_service_mode	(name_start_retry_interval 指定あり) && (name_service_mode != agent)	—
404	—	KFCA00216-E	E	nam	name_start_retry_interval	—	—	(<1) (>60)	—
405	NAM-0052	KFCA00279-E	C	nam	name_start_retry_interval	nam	name_service_mode	(name_start_retry_interval 指定あり) && (name_service_mode 指定なし)	—
406	NAM-0053	KFCA00278-W	C	nam	name_start_error	nam	name_service_mode	(name_start_error 指定あり) && (name_service_mode != agent)	—
407	—	KFCA00216-E	E	nam	name_start_error	—	—	(!= continue) && (!= stop)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
408	NAM-0054	KFCA00279-W	C	nam	name_start_error	nam	name_service_mode	(name_start_error 指定あり) && (name_service_mode 指定なし)	—
409	NAM-0055	KFCA00278-W	C	nam	name_sync_ready_time	nam	name_service_mode	(name_sync_ready_time 指定あり) && (name_service_mode != manager)	—
410	—	KFCA00216-E	E	nam	name_sync_ready_time	—	—	(<0) (>65535)	—
411	NAM-0056	KFCA00279-W	C	nam	name_sync_ready_time	nam	name_service_mode	(name_sync_ready_time 指定あり) && (name_service_mode 指定なし)	—
412	NTS-0001	KFCA00277-W	W	env	redirect_file	betranrc	rpc_port_base prc_port	(redirect_file == Y) && (rpc_port_base 指定あり prc_port 指定あり)	OS が Windows の場合だけ、チェック対象の定義をサポートしています。
413	NTS-0002	KFCA00280-W	C	env	redirect_file_size	env	redirect_file	(redirect_file != Y) && (redirect_file_size 指定あり)	OS が Windows の場合だけ、チェック対象の定義をサポートしています。
414	NTS-0003	KFCA26531-W	C	env	redirect_file_size	—	—	redirect_file_size < 1024	OS が Windows の場合だけ、チェック対象の定義をサポートしています。
415	NTS-0004	KFCA26532-W	C	env	redirect_file_size	—	—	redirect_file_size = 0	OS が Windows の場合だけ、チェッ

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
415	NTS-0004	KFCA26532-W	C	env	redirect_file_size	—	—	redirect_file_size = 0	ク対象の定義をサポートしています。
416	NTS-0005	KFCA00280-W	C	env	redirect_file_name	env	redirect_file	(redirect_file != Y) && (redirect_file_name 指定あり)	OS が Windows の場合だけ、チェック対象の定義をサポートしています。
417	NTS-0006	KFCA26533-W	C	env	redirect_file_name	—	—	無効なファイル名	OS が Windows の場合だけ、チェック対象の定義をサポートしています。
418	NTS-0007	KFCA00269-W	W	env	console_output	env	redirect_file	(redirect_file == Y) && (console_output == Y) && (rpc_port_base 指定なし) && (prc_port 指定なし)	OS が Windows の場合だけ、チェック対象の定義をサポートしています。
419	NTS-0008	KFCA26534-W	W	env	console_output	—	—	(console_output == Y) && (OpenTP1 のサービスログオンアカウントがユーザアカウントである) && (OS が Windows Server 2003 以前)	OS が Windows の場合だけ、チェック対象の定義をサポートしています。 問題識別コードが NTS-0007 の条件を満たす場合は出力されません。
420	NTS-0009	KFCA26535-W	W	env	console_output	—	—	(console_output == Y) && (OS が Windows Vista 以降)	OS が Windows の場合だけ、チェック対象の定義をサポートしています。 問題識別コードが NTS-0007 の条件を満たす場合は出力されません。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
421	NTS-0010	KFCA26536-W	W	env	console_output	—	—	(console_output == Y) && (「デスクトップとの対話をサービ스에許可する」チェックボックスが OFF) && (OS が Windows Server 2003 以前)	OS が Windows の場合だけ、チェック対象の定義をサポートしています。 問題識別コードが NTS-0007 の条件を満たす場合は出力されません。
422	NTS-0020	KFCA26537-W	W	ユーザーサービス定義	process_privilege_name	—	—	有効な特権名称ではない	OS が Windows の場合だけ、チェック対象の定義をサポートしています。
423	NTS-0021	KFCA00280-E	C	ユーザーサービス定義	process_privilege_name	—	process_privilege_restrict	(process_privilege_restrict != Y) && (process_privilege_name 指定あり)	OS が Windows の場合だけ、チェック対象の定義をサポートしています。
424	OSL-0001	KFCA00265-W	C	env	static_shmpool_size	env	—	条件なし	—
425	OSL-0002	KFCA00265-W	C	env	dynamic_shmpool_size	env	—	条件なし	—
426	OSL-0003	KFCA00272-W	W	env	shmpool_attribute	—	—	((OS == HP-UX) (OS == Solaris)) && (shmpool_attribute != free)	OS が HP-UX または Solaris の場合だけチェック対象です。
427	OSL-0004	KFCA00272-W	W	betranrc	thdlock_sleep_time	—	—	thdlock_sleep_time != 15	—
428	OSL-0005	KFCA00282-W	C	usrsrc ユーザーサービス定義	core_shm_suppress	—	—	((OS == WINDOWS) (OS == HP-UX)) && (core_shm_suppress 指定あり)	OS が Windows または HP-UX の場合、チェック対象の定義はサポートしていません。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
429	OSL-0006	KFCA00272-W	C	usrsrc ユーザー ビス定義	core_sh m_suppress	—	—	((OS != WINDOWS) && (OS != HP- UX)) && (core_shm_sup press != N)	OS が Windows また は HP-UX の場 合以外だけ チェック対象で す。
430	—	KFCA00122-E	—	env	dynami c_shmpool_size	env	static_s hmpool_size	(1992294400 — (static_shmpool_size×1024)) < (dynamic_shm pool_size×1024)	—
431	PRC-0001	KFCA00272-W	C	prc	prc_rec overy_r esident	—	—	prc_recovery_r esident == N	—
432	PRC-0002	KFCA00269-W	C	prc	term_w atch_t ime	prc	term_w atch_c ount	(term_watch_t ime 指定あり) && (term_watch_c ount == 1 or 2)	—
433	PRC-0003	KFCA00278-W	C	prc	prc_prf _trace	betranr c	prf_trac e	(prc_prf_trace ==Y && prf_trace==N) (prc_prf_trace 指定なし && prf_trace==N)	—
434	PRC-0004	KFCA00272-W	C	prc	prc_prf _trace	—	—	prc_prf_trace == N	—
435	PRC-0005	KFCA00267-W	W	prc	prc_cor esave_p ath	—	—	パスの先頭が '/'ではない	PRC#1
436	PRC-0006	KFCA00267-W	W	prc	prc_cor esave_p ath	—	—	ディレクトリで はない	—
437	PRC-0007	KFCA00267-W	W	prc	prc_cor esave_p ath	—	—	ディレクトリが 存在しない	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
438	PRC-0008	KFCA00267-W	W	prc	prc_cor esave_p ath	—	—	アクセスできない	—
439	PRC-0009	KFCA00267-W	W	prc	prc_cor esave_p ath	—	—	OpenTP1 管理者に書き込み権限がない	—
440	PRC-0010	KFCA00771-W	W	prc	prcsvpa th	—	—	prcsvpath に \$DCDIR/bin と \$DCDIR/aplib が含まれていない	PRC#3
441	PRC-0011	KFCA00282-W	C	trn	groups	—	—	groups 指定あり	OS が Windows の場合、チェック対象の定義はサポートしていません。
442	PRC-0012	KFCA00267-W	W	betranrc	prc_cur rent_w ork_pat h	—	—	パスの先頭が '/' ではない	PRC#1
443	PRC-0013	KFCA00267-W	W	betranrc	prc_cur rent_w ork_pat h	—	—	ディレクトリが存在しない	—
444	PRC-0014	KFCA00267-W	W	betranrc	prc_cur rent_w ork_pat h	—	—	ディレクトリではない	—
445	PRC-0015	KFCA00267-W	W	betranrc	prc_cur rent_w ork_pat h	—	—	アクセスできない	—
446	PRC-0016	KFCA00267-W	W	betranrc	prc_cur rent_w ork_pat h	—	—	OpenTP1 管理者に書き込み権限がない	—
447	PRC-0017	KFCA00267-W	W	betranrc	prc_cur rent_w ork_pat h	—	—	prc_current_w ork_path==ほ かの OpenTP1 環境の OpenTP1 デイ レクトリ	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
448	PRC-0018	KFCA00267-W	W	betranrc	prc_current_work_path	—	—	prc_current_work_path==ほかの OpenTP1 環境の prc_current_work_path に指定したディレクトリ	—
449	PRC-0019	KFCA00285-W	E	ユーザサービス定義	module	—	—	module 指定なし	—
450	PRC-0020	KFCA00268-W	W	ユーザサービス定義	module	prc	prcsvpath	prcsvpath に指定された各パス上で module に指定されたファイルを検索したところ、ファイルが存在しない	PRC#4
451	PRC-0021	KFCA00268-W	W	ユーザサービス定義	module	prc	prcsvpath	prcsvpath に指定された各パス上で module に指定されたファイルを検索したところ、ファイルパスが長過ぎる	PRC#4
452	PRC-0022	KFCA00268-W	W	ユーザサービス定義	module	prc	prcsvpath	prcsvpath に指定された各パス上で module に指定されたファイルを検索したところ、実行可能ファイルではない	PRC#4
453	PRC-0023	KFCA00272-W	C	ユーザサービス定義	nice	—	—	nice != 0	—
454	PRC-0024	KFCA00282-W	C	ユーザサービス定義	uid	—	—	uid 指定あり	OS が Windows の場合、チェック対象の定義はサポートしていません。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
455	PRC-0025	KFCA00282-W	C	ユーザサービス定義	groups	—	—	groups 指定あり	OS が Windows の場合、チェック対象の定義はサポートしていません。
456	PRC-0026	KFCA00267-W	W	prc	prc_cor esave_p ath	—	—	ファイルパスが長過ぎる	—
457	PRC-0027	KFCA00267-W	W	betranrc	prc_cur rent_w ork_pat h	—	—	ファイルパスが長過ぎる	—
458	—	KFCA00756-E KFCA00708-E	—	prc	prcsvpa th	—	—	(パス名 < 1) (255 < パス名)	—
459	—	KFCA00756-E KFCA00708-E	—	prc	prcsvpa th	—	—	指定パス名の先頭が':'から始まっているか、最後が':'で終わっていない	PRC#2
460	—	KFCA00757-E KFCA00708-E	—	prc	prcsvpa th	—	—	ディレクトリが存在しない	—
461	—	KFCA00758-E KFCA00708-E	—	prc	prcsvpa th	—	—	ディレクトリではない	—
462	PRF-0001	KFCA26780-W	C	prf	prf_file_ size	env	default _value_ option	(default_value_option == 0) && (prf_file_size == 1024) (default_value_option == 1) && (prf_file_size == 10240)	—
463	PRF-0002	KFCA00272-W	W	betranrc	prf_trac e	—	—	prf_trace == N	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
464	PRF-0004	KFCA00272-W	C	prf	prf_trace_backup	—	—	prf_trace_backup == N	—
465	PRF-0005	KFCA26781-W	C	prf	prf_file_size	env	default_value_option	(default_value_option == 1) && (prf_file_size < 10240)	—
466	QUE-0001	KFCA00286-W	C	que	—	sysconf	que_conf	(que_conf == Y) && (que 定義ファイルなし)	—
467	QUE-0002	KFCA00264-W	C	que	que_xidnum	trn	trn_tran_processes_count	que_xidnum > trn_tran_processes_count	—
468	QUE-0003	KFCA00264-W	C	que	quegrp	que	quegrp の-w オプション	警告解除使用率 > 使用容量警告率	—
469	—	KFCA01303-E	—	que	quegrp	—	—	キューグループ ID が重複	—
470	—	KFCA01304-E	—	que	quegrp	—	—	物理ファイルパス名が重複	—
471	—	KFCA01300-E	—	que	quegrp	—	—	有効な quegrp が存在しない, または quegrp の指定なし	—
472	—	KFCA01301-E	—	que	quegrp	—	—	次に示すどれかの条件に該当 物理ファイルが存在しない キュー物理ファイルではない キュー物理ファイルに対するアクセス権がない キュー物理ファイルのバージョンと OpenTP1 のバージョンが不一致	—
473	RPC-0001	KFCA00370-W	C	betranrc jnl scd	max_socket_de	—	—	システムサーバの論理チェック, およびユーザ	JNL#1 CPD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
473	RPC-0001	KFCA00370-W	C	trn nam prc usrcc ユーザーサービス定義	scriptors	—	—	サーバの論理チェックによって無条件で一回出力される	出力メッセージの「定義ファイル名」に次に示す文字列が出力されます。 betranrc：システムサーバの論理チェック実行時 usrcc：ユーザーサーバの論理チェック実行時
474	RPC-0002	KFCA00265-W	C	betranrc jnl scd trn nam prc usrcc ユーザーサービス定義	max_socket_descriptors	—	—	max_socket_descriptors 指定あり	JNL#1 CPD#1
475	RPC-0003	KFCA00262-W	C	betranrc usrcc ユーザーサービス定義	ipc_socketctl_highwater	—	—	ipc_socketctl_highwater = a, b 指定の場合, b > a	—
476	RPC-0004	KFCA00377-W	E	betranrc	my_host	—	—	hostname コマンドによるホスト名の取得ができない hosts ファイル, DNS などによるホスト名と IP アドレスのマッピングができない	hostname コマンドでホスト名が取得できない場合, 出力メッセージの「指定されたホスト名」に"*****"が出力されることがあります。
477	RPC-0005	KFCA00372-W	E	betranrc	my_host	—	—	ホスト名を変換した IP == 127.x.x.x	x は 0～255 の任意の値です。
478	RPC-0007	KFCA00373-W	E	betranrc	rpc_netmask	—	—	指定値がネットワークアドレス形式に変換できない	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
479	RPC-0008	KFCA00374-W	C	betranrc	rpc_netmask	betranrc	dcbindht	(dcbindht 指定あり) && (rpc_netmask 指定なし)	—
480	RPC-0009	KFCA00371-W	C	betranrc	rpc_port_base	prc	prc_process_count	(rpc_port_base + prc_process_count + 128) > 65535	—
481	RPC-0010	KFCA00266-W	C	betranrc	rpc_port_base	prc	prc_process_count	(rpc_port_base + prc_process_count + 128) <= 65535	—
482	RPC-0011	KFCA00280-W	C	betranrc	rpc_retry_count	betranrc	rpc_retry	(rpc_retry == N) && (rpc_retry_count 指定あり)	—
483	RPC-0012	KFCA00280-W	C	betranrc	rpc_retry_interval	betranrc	rpc_retry	(rpc_retry == N) && (rpc_retry_interval 指定あり)	—
484	RPC-0013	KFCA00376-W	C	betranrc	rpc_router_retry_interval	—	—	(0 < rpc_router_retry_interval) && (rpc_router_retry_interval < 10)	—
485	RPC-0014	KFCA00278-W	C	betranrc	rpc_router_retry_interval	betranrc	rpc_router_retry_count	(rpc_router_retry_count == 0) && (rpc_router_retry_interval 指定あり)	—
486	RPC-0015	KFCA00376-W	C	betranrc usrrc ユーザサービス定義	rpc_send_retry_interval	—	—	(0 < rpc_send_retry_interval) && (rpc_send_retry_interval < 20)	—
487	RPC-0016	KFCA00278-W	C	betranrc usrrc	rpc_send_retry_interval	betranrc usrrc	rpc_send_retry_count	(rpc_send_retry_count == 0) && (rpc_send_retry	RPC#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
487	RPC-0016	KFCA00278-W	C	ユーザサービス定義	rpc_send_retry_interval	ユーザサービス定義	rpc_send_retry_count	y_interval 指定あり)	RPC#1
488	RPC-0017	KFCA00370-W	C	usrsrc ユーザサービス定義	max_open_fds	—	—	ユーザサーバの論理チェックで無条件に一回出力される	ユーザサーバの論理チェックが実行された場合、出力メッセージの「定義ファイル名」に usrsrc が出力されます。
489	RPC-0018	KFCA00265-W	C	usrsrc ユーザサービス定義	max_open_fds	—	—	max_open_fds 指定あり	—
490	RPC-0019	KFCA00371-W	E	usrsrc ユーザサービス定義	max_open_fds	betranrc usrsrc ユーザサービス定義	max_socket_descriptors	((OS != Solaris) && (OS != Linux)) && ((max_socket_descriptors + max_open_fds) > 2048) ((OS == Solaris) (OS == Linux)) && ((max_socket_descriptors + max_open_fds) > 1024)	RPC#1
491	RPC-0020	KFCA00375-W	E	ユーザサービス定義	service	usrsrc ユーザサービス定義	receive_from	((receive_from == queue) (receive_from == socket)) && (service 指定なし)	RPC#1
492	RPC-0021	KFCA00278-W	C	ユーザサービス定義	service	usrsrc ユーザサービス定義	receive_from	(receive_from == none) && (service 指定あり)	—
493	RPC-0022	KFCA00275-W	E	ユーザサービス定義	service	—	—	次に示す条件のどれかに該当する場合	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
493	RPC-0022	KFCA00275-W	E	ユーザー ビス定義	service	—	—	UAP 共用ライブラリ名に空白文字の指定あり UAP 共用ライブラリ名にタブコードの指定あり UAP 共用ライブラリ名のパス名に環境変数を指定する場合で、環境変数をパス名の先頭以外に指定している	—
494	RPC-0024	KFCA00372-W	E	betranrc	dcbindht	—	—	-h オプションに指定されたホスト名を変換した IP == 127.x.x.x	x は 0～255 の任意の値です。
495	RPC-0025	KFCA00378-W	C	ユーザー ビス定義	service	—	—	UAP 共用ライブラリ名指定あり	—
496	RPC-0026	KFCA00269-W	C	usrsrc ユーザー ビス定義	atomic_update	betranrc	jnl_fileless_option	(jnl_fileless_option == Y) && (atomic_update == Y)	RPC#1 atomic_update (省略時は Y)
497	RPC-0027	KFCA00376-W	C	betranrc jnl scd trn nam prc usrsrc ユーザー ビス定義	ipc_sendbuf_size	—	—	(1 <= ipc_sendbuf_size <= 8191)	—
498	RPC-0028	KFCA00376-W	C	betranrc jnl scd trn nam prc usrsrc ユーザー ビス定義	ipc_recvbuf_size	—	—	(1 <= ipc_recvbuf_size <= 8191)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
499	—	KFCA00323-E	—	betranrc	dcbind ht	—	—	-h オプションに指定されたホスト名が hosts ファイル, DNS などホスト名と IP アドレスのマッピングができない	—
500	—	KFCA00324-E	—	betranrc	dcbind ht	—	—	-n オプションに指定されたネットワーク名が networks ファイル, NIS などネットワーク名とネットワーク番号をマッピングできない	—
501	—	KFCA00340-W (エラー要因コード: 3)	—	usrnet	dcsvgd ef	—	—	-h オプションに指定されたホスト名が hosts ファイル, DNS などホスト名と IP アドレスのマッピングができない	—
502	—	KFCA00340-W (エラー要因コード: 7)	—	usrnet	dcsvgd ef	—	—	(-w オプション指定あり) && (-t オプション指定あり)	—
503	—	KFCA00340-W (エラー要因コード: 7)	—	usrnet	dcsvgd ef	—	—	(-h オプション指定ホスト 1 個) && (-t オプション指定あり)	—
504	RTS-0001	KFCA00267-W	E	rts	rts_log_ file_ name	—	—	(rts_log_file == Y) && (RTS ログファイルの出力先ディレクトリが存在しない)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
505	RTS-0002	KFCA00272-W	C	rts	rts_log_file_count	—	—	(rts_log_file == Y) && (rts_log_file_count < 3)	—
506	RTS-0003	KFCA00280-W	C	rts	rts_log_file_name	rts	rts_log_file	(rts_log_file == N) && (rts_log_file_name 指定あり)	—
507	RTS-0004	KFCA00280-W	C	rts	rts_log_file_size	rts	rts_log_file	(rts_log_file == N) && (rts_log_file_size 指定あり)	—
508	RTS-0005	KFCA00280-W	C	rts	rts_log_file_count	rts	rts_log_file	(rts_log_file == N) && (rts_log_file_count 指定あり)	—
509	RTS-0006	KFCA00280-W	C	rts	rts_swap_message	rts	rts_log_file	(rts_log_file == N) && (rts_swap_message == Y)	—
510	RTS-0007	KFCA32765-W	C	rts	rtspout	rts	rts_service_max	取得対象の登録数 > rts_service_max	—
511	RTS-0008	KFCA00275-W	W	ユーザサービス定義	type	—	—	((type == RTS) && (定義ファイル名が RTSSUP ではない)) ((type == RTS) && (定義ファイル名が RTSSPP ではない))	—
512	RTS-0009	KFCA00275-W	W	ユーザサービス定義	module	—	—	((type != RTS) && (module == rtssup) && (定義ファイル名が RTSSUP ではない)) ((type != RTS) && (module ==	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
512	RTS-0009	KFCA00275-W	W	ユーザサービス定義	module	—	—	rtsspp) && (定義ファイル名が RTSSPP ではない))	—
513	RTS-0010	KFCA32763-W	W	ユーザサービス定義 (RTSSUP または RTSSPP)	type	ユーザサービス定義	module	((type != RTS) && (module == rtssup) && (定義ファイル名が RTSSUP)) ((type != RTS) && (module == rtsspp) && (定義ファイル名が RTSSPP))	—
514	RTS-0011	KFCA32764-W	W	ユーザサービス定義 (RTSSUP または RTSSPP)	—	—	—	RTSSUP または RTSSPP の内容が, rtsssetup コマンドで作成したときと異なっている	—
515	RTS-0012	KFCA00267-W	E	rts	rts_log_file_name	—	—	(rts_log_file == Y) && ((RTS ログファイルの出力先ディレクトリに OpenTP1 管理者の書き込み権限がない) (RTS ログファイルの出力先ディレクトリに OpenTP1 管理者の実行権限がない))	—
516	RTS-0013	KFCA00268-W	E	rts	rts_log_file_name	—	—	(rts_log_file == Y) && (rts_log_file_name に指定した RTS ログファイルに OpenTP1 管理者の書き込み権限がない)	RTS#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
517	RTS-0014	KFCA00268-W	E	rts	rts_log_file_name	—	—	(rts_log_file == Y) && (rts_log_file_name に指定した RTS ログファイルがファイルではない)	RTS#1
518	RTS-0015	KFCA00272-W	C	rts	rts_log_file	—	—	rts_log_file == N	—
519	RTS-0016	KFCA00280-W	C	rts	rts_log_file_backup	—	rts_log_file	(rts_log_file == N) && (rts_log_file_backup == Y)	—
520	RTS-0017	KFCA00268-W	C	rts	rts_log_file_backup	—	rts_log_file rts_log_file_name	(rts_log_file == Y) && (rts_log_file_backup == Y) && (バックアップファイルと同じ名称のディレクトリが存在する)	RTS#2
521	—	KFCA00242-E	—	rts	rtspout	—	—	((-u srv) && (-s 存在しないユーザサービス定義ファイル名)) (-f 存在しないリアルタイム取得項目定義ファイル名)	—
522	—	KFCA32710-W	—	rts	rtspout	rts	rts_item_max	取得する項目の登録数 > rts_item_max	—
523	—	KFCA32709-W	—	rts	rtspout	rts	rts_service_max	取得対象の登録数 > rts_service_max	—
524	SCD-0001	KFCA00262-W	C	scd	scd_server_count	env	server_count	scd_server_count > server_count	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
525	SCD-0002	KFCA00279-W	C	scd	scd_hold_recovery	env	start_scheduling_timing	(scd_hold_recovery 指定あり) && (start_scheduling_timing 指定なし)	—
526	SCD-0003	KFCA00278-W	C	scd	scd_hold_recovery	env	start_scheduling_timing	(scd_hold_recovery 指定あり) && (start_scheduling_timing == AFTER)	—
527	SCD-0004	KFCA00278-W	C	scd	scd_hold_recovery	scd	scd_hold_recovery_count	(scd_hold_recovery 指定あり) && (scd_hold_recovery_count == 0))	—
528	SCD-0005	KFCA33200-W	E	scd	scd_hold_recovery_count	betranrc	rpc_max_message_size	↑ (scd_hold_recovery_count / 7280) ↑ > (rpc_max_message_size 指定値 (省略値：1))	—
529	SCD-0006	KFCA00266-W	C	scd	scd_port	—	—	scd_port 指定あり	—
530	SCD-0007	KFCA00272-W	C	scd	scd_announce_server_status	—	—	scd_announce_server_status == N	—
531	SCD-0008	KFCA00278-W	C	scd	scd_advertise_control	env	start_scheduling_timing	(scd_advertise_control == AFTER) && (start_scheduling_timing == BEFORE)	—
532	SCD-0009	KFCA00272-W	C	scd	scd_message_level	—	—	scd_message_level == 1	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
533	SCD-0010	KFCA00259-W	E	scd	scdbufgrp	scd	scdbufgrp	ほかの scdbufgrp に指定したスケジュールバッファグループ名と重複	—
534	SCD-0011	KFCA00274-W	E	scd	scdbufgrp	—	—	(scdbufgrp の-n オプション指定値 × scdbufgrp の-l オプション指定値) > 31457280 推奨値： 31457280 / -n オプション指定値	—
535	SCD-0012	KFCA00259-W	E	scd	scdmulti	scd	scdmulti	ほかの scdmulti に指定したマルチスケジューラグループ名と重複 (-g オプション指定なしの scdmulti が複数指定された場合)	—
536	SCD-0013	KFCA33201-W	E	scd	scdmulti	prc	prc_process_count	(scdmulti の-m オプション指定値の総和) > (prc_process_count 指定値)	—
537	SCD-0014	KFCA33201-W	E	scd	scdmulti	env	server_count	(scdmulti の-m オプション指定値の総和) > (server_count 指定値)	—
538	SCD-0015	KFCA33202-W	E	scd	scdmulti	scd	scd_port	(scd_port 指定なし) && (最初に指定した scdmulti に-p オプション指定なし)	—
539	SCD-0016	KFCA33203-W	E	scd	scdmulti	—	—	(scdmulti の-p オプション指定値)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
539	SCD-0016	KFCA33203-W	E	scd	scdmulti	—	—	+ scdmulti の-m オプション指定値(省略値は 1) - 1) > 65535)	—
540	SCD-0017	KFCA33203-W	E	scd	scdmulti	scd	scd_port	(scdmulti の-p オプション指定なし) && ((scdmulti の-m オプション指定値(省略値は 1) + scd_port 指定値) > 65535)	—
541	SCD-0018	KFCA00266-W	C	scd	scdmulti	—	—	ほかの scdmulti の論理チェックがエラーではない場合	—
542	SCD-0019	KFCA00285-W	E	scd	scdbufgrp	scd	scdbufgrp	scdbufgrp のスケジュールバッファグループ名を省略	—
543	SCD-0027	KFCA00274-W	E	scd	scdbufgrp	—	—	↑ (scdbufgrp の-e オプション指定値 / scdbufgrp の-l オプション指定値) ↑ × scdbufgrp の-l オプション指定値 > 1610612736 推奨値：-l オプション指定値の整数倍	—
544	SCD-0028	KFCA00262-W	E	scd	scdbufgrp	—	—	(scdbufgrp 指定あり) && (scdbufgrp -s 指定あり) && (scdbufgrp の-s オプション指定値 > (scdbufgrp の-l オプション × -n オプション指定値))	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
545	SCD-0029	KFCA00262-W	E	scd	scdbufgrp	—	—	(scdbufgrp 指定あり) && (scdbufgrp -s 指定あり) && (scdbufgrp の-s オプション指定値 > scdbufgrp の-e オプション指定値)	—
546	SCD-0030	KFCA00263-W	E	scd	scdbufgrp	—	—	(scdbufgrp 指定あり) && (scdbufgrp -s 指定あり) && (scdbufgrp の-s オプション指定値 < scdbufgrp の-l オプション指定値)	—
547	SCD-0031	KFCA33210-W	E	scd	scdbufgrp	—	—	(scdbufgrp 指定あり) && (scdbufgrp -p 指定あり) && (セル数 * scdbufgrp の-p オプション指定値 ÷ 100 < 1) セル数： scdbufgrp -n オプション指定値 または scdbufgrp -e オプション指定値 ÷ -l オプション指定値	—
548	SCD-1001	KFCA00285-W	E	ユーザサービス定義	service_group	ユーザサービス定義	receive_from	(receive_from == socket) && (service_group 指定なし)	—
549	SCD-1002	KFCA00285-W	E	ユーザサービス定義	service_group	ユーザサービス定義	receive_from	(receive_from == queue) && (service_group 指定なし)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
550	SCD-1003	KFCA33204-W	E	ユーザサービス定義	parallel_count	—	—	(receive_from == queue) && (parallel_count の常駐プロセス数が == 0) && (parallel_count の最大プロセス数が == 0)	SCD#1 parallel_count が次に示す指定値の場合 ・ set parallel_count = 0, 0 ・ set parallel_count = 0
551	SCD-1004	KFCA33205-W	E	ユーザサービス定義	parallel_count	—	—	(receive_from == queue) && (parallel_count の常駐プロセス数 > parallel_count の最大プロセス数)	SCD#1
552	SCD-1005	KFCA00262-W	E	ユーザサービス定義	parallel_count	prc	prc_process_count	(receive_from == queue) && (parallel_count の常駐プロセス数 >= prc_process_count)	SCD#1
553	SCD-1006	KFCA00262-W	C	ユーザサービス定義	parallel_count	prc	prc_process_count	(receive_from == queue) && (parallel_count の常駐プロセス数 < prc_process_count) && (parallel_count の最大プロセス数 >= prc_process_count)	SCD#1
554	SCD-1007	KFCA00278-W	C	ユーザサービス定義	hold_recovery	scd	scd_hold_recovery_count	(receive_from == queue) && (type == other) && (hold_recovery 指定あり)	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
554	SCD-1007	KFCA00278-W	C	ユーザサービス定義	hold_recovery	scd	scd_hold_recovery_count	&&(hold_recovery == Y) &&(scd_hold_recovery_count == 0)	SCD#1
555	SCD-1008	KFCA00280-W	C	ユーザサービス定義	hold_recovery	env scd	start_scheduling_timing scd_hold_recovery	(receive_from == queue) &&(type == other) &&(hold_recovery 指定あり) &&(hold_recovery == Y) &&(start_scheduling_timing == BEFORE) &&(scd_hold_recovery != F)	SCD#1
556	SCD-1009	KFCA00281-W	C	ユーザサービス定義	message_store_buflen	ユーザサービス定義	scdbufgrp	(receive_from == queue) &&(message_store_buflen 指定あり) &&(scdbufgrp 指定あり)	SCD#1
557	SCD-1010	KFCA00265-W	C	ユーザサービス定義	message_store_buflen	ユーザサービス定義	type	(receive_from == queue) &&(type == other) &&(scdbufgrp 指定なし)	SCD#1
558	SCD-1011	KFCA00273-W KFCA00265-W	C	ユーザサービス定義	message_store_buflen	ユーザサービス定義	type	(receive_from == queue) &&(type == MHP) &&((parallel_coun	SCD#1 SCD#2

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
558	SCD-1011	KFCA00273-W KFCA00265-W	C	ユーザサービス定義	message_store_buflen	ユーザサービス定義	type	t の常駐プロセス数 == 最大プロセス数) balance_count == 0) && (message_store_buflen の指定値 < (512 × parallel_count の最大プロセス数))	SCD#1 SCD#2
559	SCD-1012	KFCA00273-W KFCA00265-W	C	ユーザサービス定義	message_store_buflen	ユーザサービス定義	type	(receive_from == queue) && (type == MHP) && ((parallel_count の常駐プロセス数 != 最大プロセス数) && balance_count > 0) && (message_store_buflen の指定値 < (512 × parallel_count の最大プロセス数 × balance_count の指定値))	SCD#1 SCD#2
560	SCD-1013	KFCA33209-W	C	ユーザサービス定義	balance_count	ユーザサービス定義	parallel_count	(receive_from == queue) && (balance_count 指定あり) && (parallel_count の常駐プロセス数 == 最大プロセス数)	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
561	SCD-1014	KFCA00281-W	C	ユーザサービス定義	message_cell_size	ユーザサービス定義	scdbufgrp	(receive_from == queue) && (message_cell_size 指定あり) && (scdbufgrp 指定あり)	SCD#1
562	SCD-1015	KFCA33206-W	C	ユーザサービス定義	message_cell_size	ユーザサービス定義	service_priority_control service_hold	(receive_from == queue) && (type == other) && (message_cell_size 指定あり) && ((service_priority_control == N) && (service_hold == N))	SCD#1
563	SCD-1016	KFCA33207-W	C	ユーザサービス定義	purge_msgget	ユーザサービス定義	parallel_count	(receive_from == queue) && (purge_msgget 指定あり) && (parallel_count の常駐プロセス数 > 0)	SCD#1
564	SCD-1017	KFCA00278-W	C	ユーザサービス定義	service_term_watch_time	ユーザサービス定義	hold	(receive_from == queue) && (type == other) && (service_term_watch_time > 0) && (hold == Y)	SCD#1
565	SCD-1018	KFCA00279-W	C	ユーザサービス定義	service_term_watch_time	ユーザサービス定義	hold	(receive_from == queue) && (type == other)	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
565	SCD-1018	KFCA00279-W	C	ユーザサービス定義	service_term_watch_time	ユーザサービス定義	hold	&& (service_term_watch_time > 0) && (hold 指定なし)	SCD#1
566	SCD-1019	KFCA00278-W	C	ユーザサービス定義	service_term_watch_time	ユーザサービス定義	service_hold	(receive_from == queue) && (type == other) && (service_term_watch_time > 0) && (service_hold 指定あり) && (service_hold == N)	SCD#1
567	SCD-1020	KFCA00279-W	C	ユーザサービス定義	service_term_watch_time	ユーザサービス定義	service_hold	(receive_from == queue) && (type == other) && (service_term_watch_time > 0) && (service_hold 指定なし)	SCD#1
568	SCD-1021	KFCA33209-W	C	ユーザサービス定義	termed_after_service	ユーザサービス定義	parallel_count	(receive_from == queue) && (parallel_count の常駐プロセス数 == 最大プロセス数) && (termed_after_service == Y)	SCD#1
569	SCD-1023	KFCA00282-W	C	ユーザサービス定義	schedule_method	—	—	schedule_method 指定あり	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
570	SCD-1024	KFCA00282-W	C	ユーザサービス定義	service_wait_time	—	—	service_wait_time 指定あり	SCD#1
571	SCD-1027	KFCA00278-W	C	ユーザサービス定義	make_queue_on_starting	ユーザサービス定義	purge_msgget	(receive_from == queue) && (make_queue_on_starting == Y) && (purge_msgget == Y)	SCD#1
572	SCD-1028	KFCA33207-W	C	ユーザサービス定義	make_queue_on_starting	ユーザサービス定義	parallel_count	(receive_from == queue) && (make_queue_on_starting == Y) && (parallel_count の常駐プロセス数 > 0)	SCD#1
573	SCD-1029	KFCA00278-W	C	ユーザサービス定義	scd_poolfull_check_count	ユーザサービス定義	scd_poolfull_check_interval	(receive_from == queue) && (scd_poolfull_check_count 指定あり) && (scd_poolfull_check_interval == 0)	SCD#1
574	SCD-1030	KFCA00279-W	C	ユーザサービス定義	scd_poolfull_check_count	ユーザサービス定義	scd_poolfull_check_interval	(receive_from == queue) && (scd_poolfull_check_count 指定あり) && (scd_poolfull_check_interval 指定なし)	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
575	SCD-1031	KFCA00278-W	C	ユーザサービス定義	loadcheck_interval	scd	scd_announce_server_status	(receive_from == queue) && (type == other) && (loadcheck_interval > 0) && (scd_announce_server_status == N)	SCD#1
576	SCD-1032	KFCA00278-W	C	ユーザサービス定義	levelup_queue_count	scd	scd_announce_server_status	(receive_from == queue) && (type == other) && (levelup_queue_count 指定あり) && (scd_announce_server_status == N)	SCD#1 levelup_queue_count と leveldown_queue_count の指定値が次に示す条件に当てはまるように指定します。 条件式： $0 \leq D0 < U1 \leq D1 < U2$
577	SCD-1033	KFCA00261-W	E	ユーザサービス定義	levelup_queue_count	ユーザサービス定義	leveldown_queue_count	(receive_from == queue) && (type == other) && (levelup_queue_count 指定あり) && ((scd_announce_server_status == Y) (scd_announce_server_status 指定なし)) && (U1 < 1)	SCD#1 levelup_queue_count と leveldown_queue_count の指定値が次に示す条件に当てはまるように指定します。 条件式： $0 \leq D0 < U1 \leq D1 < U2$
578	SCD-1034	KFCA00262-W	E	ユーザサービス定義	levelup_queue_count	—	—	(receive_from == queue) && (type == other)	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
578	SCD-1034	KFCA00262-W	E	ユーザサービス定義	levelup_queue_count	—	—	&& (levelup_queue_count 指定あり) && ((scd_announcement_server_status == Y) (scd_announcement_server_status 指定なし)) && (U1 > 0) && (U2 ≤ U1)	SCD#1
579	SCD-1035	KFCA00279-W	C	ユーザサービス定義	leveldown_queue_count	ユーザサービス定義	levelup_queue_count	(receive_from == queue) && (type == other) && (leveldown_queue_count 指定あり) && (levelup_queue_count 指定なし)	SCD#1 levelup_queue_count と leveldown_queue_count の指定値が次に示す条件に当てはまるように指定します。 条件式：0 ≤ D0 < U1 ≤ D1 < U2
580	SCD-1036	KFCA00262-W	E	ユーザサービス定義	leveldown_queue_count	ユーザサービス定義	levelup_queue_count	(receive_from == queue) && (type == other) && (leveldown_queue_count 指定あり) && (levelup_queue_count 指定あり) && ((scd_announcement_server_status == Y) (scd_announce	SCD#1 levelup_queue_count と leveldown_queue_count の指定値が次に示す条件に当てはまるように指定します。 条件式：0 ≤ D0 < U1 ≤ D1 < U2

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
580	SCD-1036	KFCA00262-W	E	ユーザーサービス定義	leveldown_queue_count	ユーザーサービス定義	levelup_queue_count	_server_status 指定なし)) && (D0 ≥ U1)	SCD#1 levelup_queue_count と leveldown_queue_count の指定値が次に示す条件に当てはまるように指定します。 条件式：0 ≤ D0 < U1 ≤ D1 < U2
581	SCD-1037	KFCA00262-W	E	ユーザーサービス定義	leveldown_queue_count	ユーザーサービス定義	levelup_queue_count	(receive_from == queue) && (type == other) && (leveldown_queue_count 指定あり) && (levelup_queue_count 指定あり) && ((scd_announce_server_status == Y) (scd_announce_server_status 指定なし)) && (D1 ≥ U2)	SCD#1
582	SCD-1038	KFCA00278-W	C	ユーザーサービス定義	schedule_delay_abort	ユーザーサービス定義	schedule_delay_limit	(receive_from == queue) && (type == other) && schedule_delay_abort 指定あり && (schedule_delay_limit == 0)	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
583	SCD-1039	KFCA00279-W	C	ユーザーサービス定義	schedul e_delay _abort	ユーザ サービス 定義	schedul e_delay _limit	(receive_from == queue) && (type == other) && schedule_dela y_abort 指定 あり && (schedule_dela y_limit 指定な し)	SCD#1
584	SCD-1040	KFCA00278-W	C	ユーザーサービス定義	scd_po ol_warn ing_inte rval	ユーザ サービス 定義	scd_po ol_warn ing_use _rate	(receive_from == queue) && (type == other) && (scd_pool_war ning_interval > 0) && (scd_pool_war ning_use_rate == 0)	SCD#1
585	SCD-1041	KFCA00279-W	C	ユーザーサービス定義	scd_po ol_warn ing_inte rval	ユーザ サービス 定義	scd_po ol_warn ing_use _rate	(receive_from == queue) && (type == other) && (scd_pool_war ning_interval > 0) && (scd_pool_war ning_use_rate 指定なし)	SCD#1
586	SCD-1042	KFCA00278-W	C	ユーザーサービス定義	stay_w atch_ch eck_rat e	ユーザ サービス 定義	stay_w atch_qu eue_co unt	(receive_from == queue) && (type == other) && (stay_watch_q ueue_count == 0)	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
586	SCD-1042	KFCA00278-W	C	ユーザサービス定義	stay_watch_check_rate	ユーザサービス定義	stay_watch_queue_count	&& (stay_watch_check_rate 指定あり)	SCD#1
587	SCD-1043	KFCA00279-W	C	ユーザサービス定義	stay_watch_check_rate	ユーザサービス定義	stay_watch_queue_count	(receive_from == queue) && (type == other) && (stay_watch_queue_count 指定なし) && (stay_watch_check_rate 指定あり)	SCD#1
588	SCD-1044	KFCA00278-W	C	ユーザサービス定義	stay_watch_abort	ユーザサービス定義	stay_watch_queue_count	(receive_from == queue) && (type == other) && (stay_watch_abort 指定あり) && (stay_watch_queue_count == 0)	SCD#1
589	SCD-1045	KFCA00279-W	C	ユーザサービス定義	stay_watch_abort	ユーザサービス定義	stay_watch_queue_count	(receive_from == queue) && (type == other) && (stay_watch_abort 指定あり) && (stay_watch_queue_count 指定なし)	SCD#1
590	SCD-1046	KFCA00278-W	C	ユーザサービス定義	stay_watch_start_interval	ユーザサービス定義	stay_watch_queue_count	(receive_from == queue) && (type == other) && (stay_watch_start_interval 指定あり)	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
590	SCD-1046	KFCA00278-W	C	ユーザサービス定義	stay_watch_start_interval	ユーザサービス定義	stay_watch_queue_count	art_interval 指定あり) && (stay_watch_queue_count == 0)	SCD#1
591	SCD-1047	KFCA00279-W	C	ユーザサービス定義	stay_watch_start_interval	ユーザサービス定義	stay_watch_queue_count	(receive_from == queue) && (type == other) && (stay_watch_start_interval 指定あり) && (stay_watch_queue_count 指定なし)	SCD#1
592	SCD-1048	KFCA00278-W	C	ユーザサービス定義	stay_watch_check_interval	ユーザサービス定義	stay_watch_queue_count	(receive_from == queue) && (type == other) && (stay_watch_check_interval 指定あり) && (stay_watch_queue_count == 0)	SCD#1
593	SCD-1049	KFCA00279-W	C	ユーザサービス定義	stay_watch_check_interval	ユーザサービス定義	stay_watch_queue_count	(receive_from == queue) && (type == other) && (stay_watch_check_interval 指定あり) && (stay_watch_queue_count 指定なし)	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
594	SCD-1050	KFCA00276-W	E	ユーザーサービス定義	scdbufgrp	scd	scdbufgrp	指定されたバックアップグループがスケジュールサービス定義に指定されていない	—
595	SCD-1051	KFCA33208-W	E	ユーザーサービス定義	scdmulti	ユーザーサービス定義	scdmulti	scdmulti が一つのユーザーサービス定義に複数指定されている	—
596	SCD-1052	KFCA00276-W	E	ユーザーサービス定義	scdmulti	scd	scdmulti	指定されたマルチスケジュールグループがスケジュールサービス定義の scdmulti で指定されていない	—
597	SCD-1053	KFCA00281-W	C	ユーザーサービス定義	termed_after_service	ユーザーサービス定義	service_wait_time	(receive_from == queue) && (termed_after_service == Y) && (schedule_method == namedpipe) && (service_wait_time 指定あり)	OS が Windows の場合はチェックしません。
598	SCD-1054	KFCA00278-W	C	ユーザーサービス定義	service_wait_time	ユーザーサービス定義	schedul_e_method	(receive_from == queue) && ((schedule_method == msgque) (schedule_method 指定なし)) && (service_wait_time 指定あり)	SCD#1 OS が Windows の場合はチェックしません。
599	SCD-1055	KFCA00259-W	W	ユーザーサービス定義	scdsvcd ef	ユーザーサービス定義	scdsvcd ef	次に示すどちらかの条件	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
599	SCD-1055	KFCA00259-W	W	ユーザサービス定義	scdsvcd ef	ユーザサービス定義	scdsvcd ef	指定されたサービス名がほかの scdsvcd ef 定義コマンドで指定されている サービス名を省略した scdsvcd ef 定義コマンドが重複して指定されている	—
600	SCD-1056	KFCA00276-W	W	ユーザサービス定義	scdsvcd ef	ユーザサービス定義	service	指定されたサービス名が service 定義コマンドに指定されていない	—
601	SCD-1057	KFCA00262-W	W	ユーザサービス定義	scdsvcd ef	ユーザサービス定義	message_store_buflen	(receive_from == queue) && (type == other) && (scdbufgrp 指定なし) && (scdsvcd ef -l 指定値 >= message_store_buflen)	SCD#1
602	SCD-1058	KFCA00262-W	W	ユーザサービス定義	scdsvcd ef	ユーザサービス定義	scdbufgrp	(receive_from == queue) && (type == other) && (scdbufgrp 指定あり) && (scdsvcd ef -l 指定値 >= (scdbufgrp -n 指定値 × scdbufgrp -l 指定値))	—
603	SCD-1059	KFCA00262-W	W	ユーザサービス定義	scdsvcd ef	ユーザサービス定義	parallel_count	(receive_from == queue) && (type == other) && (scdsvcd ef -p 指定値 >= parallel_count)	SCD#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
604	SCD-1060	KFCA00262-W	E	ユーザサービス定義	scdbufgrp	ユーザサービス定義	scdbufgrp	(receive_from == queue) && (type == other) && (scdbufgrp 指定あり) && (scdbufgrp -s 指定あり) && (scdbufgrp の-s オプション指定値 > (スケジュールサービス定義 scdbufgrp の-l オプション×-n オプション指定値))	—
605	SCD-1061	KFCA00262-W	E	ユーザサービス定義	scdbufgrp	ユーザサービス定義	scdbufgrp	(receive_from == queue) && (type == other) && (scdbufgrp 指定あり) && (scdbufgrp -s 指定あり) && (scdbufgrp の-s オプション指定値 > スケジュールサービス定義 scdbufgrp の-e オプション指定値)	—
606	SCD-1062	KFCA00263-W	E	ユーザサービス定義	scdbufgrp	ユーザサービス定義	scdbufgrp	(receive_from == queue) && (type == other) && (scdbufgrp 指定あり) && (scdbufgrp -s 指定あり) && (scdbufgrp の-s オプション指定値 < スケジュールサービス定義	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
606	SCD-1062	KFCA00263-W	E	ユーザサービス定義	scdbufgrp	ユーザサービス定義	scdbufgrp	scdbufgrp の-l オプション指定値)	—
607	SCD-1063	KFCA00269-W	E	ユーザサービス定義	scdbufgrp	ユーザサービス定義	scdbufgrp	(receive_from == queue) && (type == MHP) && (scdbufgrp 指定あり) && (scdbufgrp -s または -p 指定あり)	—
608	SCD-1064	KFCA33210-W	E	ユーザサービス定義	scdbufgrp	ユーザサービス定義	scdbufgrp	(receive_from == queue) && (type == other) && (scdbufgrp 指定あり) && (scdbufgrp -p 指定あり) && (セル数 * scdbufgrp の-p オプション指定値 ÷ 100 < 1) セル数：スケジューラサービス定義の scdbufgrp -n オプション指定値または scdbufgrp -e オプション指定値 ÷ -l オプション指定値	—
609	SCS-0001	KFCA27770-W	C	rap リスナー用ユーザサービス定義	rap_client_manager_node	—	—	(rap_client_manager_node == my_host 指定値) (rap_client_manager_node のホスト名 == ローカルループバックアドレス)	—
610	SCS-0002	KFCA27772-W	E	rap リスナー用ユー	rap_client_man	—	—	rap_client_manager_node != "ホスト名:port"	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
610	SCS-0002	KFCA27772-W	E	ザサービス定義	ager_node	—	—	rap_client_manager_node != "ホスト名:port"	—
611	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	module	—	—	(DCSCSLNAME 値 == 定義ファイル名) && (module != raplisnr)	—
612	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	atomic_update	—	—	atomic_update != N	—
613	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	receive_from	—	—	receive_from != none	—
614	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	auto_restart	—	—	auto_restart != Y	—
615	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	critical	—	—	critical != N	—
616	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	trf_put	—	—	trf_put != N	—
617	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	node_down_restart	—	—	node_down_restart != Y	—
618	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	term_watch_time	—	—	term_watch_time != 0	—
619	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	max_open_fds	—	—	max_open_fds != rapdfgen 生成値	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
620	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	rpc_destination_mode	—	—	rpc_destination_mode != namdonly	—
621	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	status_change_when_termining	—	—	status_change_when_termining != N	—
622	SCS-0002	KFCA27772-W	E	rap リスナー用ユーザサービス定義	DCSCSPARA (rap_parallel_server)	任意 (rap サーバ)	parallel_count	DCSCSPARA 値 != parallel_count	—
623	SCS-0003	KFCA00266-W	C	rap リスナー用ユーザサービス定義	rap_client_manager_node	—	—	rap_client_manager_node 指定あり	—
624	SCS-0004	KFCA27771-W	E	rap リスナー用ユーザサービス定義	module	—	—	(type != RAP) && (module == raplisnr)	—
625	SCS-0005	KFCA00278-W	C	rap リスナー用ユーザサービス定義	rap_client_manager_node	—	—	(rap_notify == N) && (rap_client_manager_node 指定あり)	—
626	SCS-0006	KFCA00266-W	C	rap リスナー用ユーザサービス定義	DCSCSPORT (rap_listen_port)	—	—	DCSCSPORT 指定あり	—
627	SCS-0007	KFCA27777-W	W	rap リスナー用ユーザサービス定義	max_socket_descriptors	betranrc usrrc	max_open_fds	((OS != Solaris) && (OS != Linux)) && ((max_socket_descriptors + max_open_fds) > 2048) ((OS == Solaris) (OS == Linux)) && ((max_socket_	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
627	SCS-0007	KFCA27777-W	W	rap リスナー用ユーザサービス定義	max_socket_descriptors	betranrc usrrc	max_open_fds	descriptors + max_open_fds) > 1024)	—
628	SCS-0008	KFCA27778-W	E	rap リスナー用ユーザサービス定義	—	任意 (rap サーバ)	—	rap サーバの定義ファイルなし	—
629	SCS-0009	KFCA27775-W	E	rap リスナー用ユーザサービス定義	rap_client_manager_node	—	—	ホスト名がアドレス解決できない	—
630	SCS-0010	KFCA00279-W	C	rap リスナー用ユーザサービス定義	rap_client_manager_node	—	—	(rap_notify == Y) && (rap_client_manager_node 指定なし)	—
631	SCS-0011	KFCA00262-W	C	rap リスナー用ユーザサービス定義	rap_term_disconnect_time	env	system_terminate_watch_time	(system_terminate_watch_time != 0) && ((rap_term_disconnect_time == 0) (rap_term_disconnect_time >= system_terminate_watch_time))	—
632	SCS-0012	KFCA00274-W	C	rap リスナー用ユーザサービス定義	rap_stay_watch_time	—	—	rap_stay_watch_time > 30	—
633	SCS-0013	KFCA00278-W	C	rap リスナー用ユーザサービス定義	rap_stay_warning_interval	rap リスナー用ユーザサービス定義	rap_stay_watch_time	(rap_stay_watch_time == 0) && rap_stay_warning_interval 指定あり	—
634	SCS-0101	KFCA27772-W	E	rap サーバ用ユーザサービス定義	service_group	—	—	(type == RAP) && (service_group != 定義ファイル名)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
635	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	module	—	—	(type == RAP) && (module != rapserver)	—
636	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	atomic_ update	—	—	atomic_update != Y	—
637	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	hold	—	—	hold != N	—
638	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	hold_re covery	—	—	hold_recovery != N	—
639	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	service	—	—	service != "rapexec=scs_ service_exec"	—
640	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	balance _count	—	—	balance_count != 0	—
641	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	auto_re start	—	—	auto_restart != Y	—
642	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	critical	—	—	critical != N	—
643	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	service_ hold	—	—	service_hold != N	—
644	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	service_ priority _contro l	—	—	service_priority _control != N	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
645	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	node_down_re start	—	—	node_down_re start != N	—
646	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	server_type	—	—	server_type != betran	—
647	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	term_watch_t ime	—	—	term_watch_t ime != 0	—
648	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	max_op en_fds	—	—	max_open_fds != rapdfgen 生 成値	—
649	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	server_s ecurity	—	—	server_security != N	—
650	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	messag e_store _buflen	—	—	message_store _buflen != rapdfgen 生 成値	—
651	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	schedul e_delay _limit	—	—	schedule_dela y_limit != 0	—
652	SCS-0101	KFCA27772-W	E	rap サーバ 用ユーザ サービス 定義	schedul e_delay _abort	—	—	schedule_dela y_abort != N	—
653	SCS-0102	KFCA27777-W	W	rap サーバ 用ユーザ サービス 定義	max_so cket_de scriptor s	betranr c usrrc	max_op en_fds	((OS != Solaris) && (OS != Linux)) && ((max_socket_ descriptors + max_open_fds) > 2048) ((OS == Solaris)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
653	SCS-0102	KFCA27777-W	W	rap サーバ 用ユーザ サービス 定義	max_socket_descriptors	betranrc usrsrc	max_open_fds	(OS == Linux)) && ((max_socket_descriptors + max_open_fds) > 1024)	—
654	SCS-0103	KFCA27771-W	E	rap サーバ 用ユーザ サービス 定義	module	—	—	(type != RAP) && (module == rapserver)	—
655	SCS-0201	KFCA27772-W	E	rap クライ アントマネ ージャ用ユー ザサービス 定義	module	—	—	(type == RAP) && (module != rapclman)	—
656	SCS-0201	KFCA27772-W	E	rap クライ アントマネ ージャ用ユー ザサービス 定義	atomic_update	—	—	atomic_update != N	—
657	SCS-0201	KFCA27772-W	E	rap クライ アントマネ ージャ用ユー ザサービス 定義	receive_from	—	—	receive_from != none	—
658	SCS-0201	KFCA27772-W	E	rap クライ アントマネ ージャ用ユー ザサービス 定義	auto_restart	—	—	auto_restart != Y	—
659	SCS-0201	KFCA27772-W	E	rap クライ アントマネ ージャ用ユー ザサービス 定義	critical	—	—	critical != N	—
660	SCS-0201	KFCA27772-W	E	rap クライ アントマネ ージャ用ユー ザサービス 定義	trf_put	—	—	trf_put != N	—
661	SCS-0201	KFCA27772-W	E	rap クライ アントマネ ージャ用ユー	node_down_restart	—	—	node_down_restart != Y	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
661	SCS-0201	KFCA27772-W	E	ザサービス定義	node_down_restart	—	—	node_down_restart != Y	—
662	SCS-0201	KFCA27772-W	E	rap クライアントマネージャ用ユーザサービス定義	term_watch_time	—	—	term_watch_time != 0	—
663	SCS-0201	KFCA27772-W	E	rap クライアントマネージャ用ユーザサービス定義	max_open_fds	—	—	max_open_fds != rapdfgen 生成値	—
664	SCS-0201	KFCA27772-W	E	rap クライアントマネージャ用ユーザサービス定義	rpc_destination_mode	—	—	rpc_destination_mode != namdonly	—
665	SCS-0201	KFCA27772-W	E	rap クライアントマネージャ用ユーザサービス定義	status_change_when_termining	—	—	status_change_when_termining != N	—
666	SCS-0201	KFCA27772-W	E	rap クライアントマネージャ用ユーザサービス定義	rap_listen_inf	—	—	rap_listen_inf != "ノード名:port=ホスト名:port"	—
667	SCS-0202	KFCA00266-W	C	rap クライアントマネージャ用ユーザサービス定義	rap_client_manager_port	—	—	rap_client_manager_port 指定あり	—
668	SCS-0203	KFCA27777-W	W	rap クライアントマネージャ用ユーザサービス定義	max_socket_descriptors	betranrc usrrc	max_open_fds	((OS != Solaris) && (OS != Linux)) && ((max_socket_descriptors + max_open_fds) > 2048) ((OS == Solaris)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
668	SCS-0203	KFCA27777-W	W	rap クライアントマネージャ用ユーザサービス定義	max_socket_descriptors	betranrc usrsrc	max_open_fds	(OS == Linux)) && ((max_socket_descriptors + max_open_fds) > 1024)	—
669	SCS-0204	KFCA27771-W	E	rap クライアントマネージャ用ユーザサービス定義	module	—	—	(type!=RAP) && (module==rapclman)	—
670	SCS-0205	KFCA27770-W	C	rap クライアントマネージャ用ユーザサービス定義	rap_listen_inf	—	—	(rap_listen_inf == my_host 指定値) (rap_listen_inf == ローカルループバックアドレス)	—
671	SCS-0206	KFCA27775-W	E	rap クライアントマネージャ用ユーザサービス定義	rap_listen_inf	—	—	ホスト名がアドレス解決できない	—
672	SCS-0501	KFCA00260-W	C	usrnet	dcsvgdef	—	—	次に示すどちらかの条件 -h のホスト名 == my_host 指定値 -h のホスト名 == ローカルループバックアドレス	—
673	STS-0001	KFCA00286-W	E	sts	全体	—	—	ステータスサービス定義ファイル (sts) が存在しない	—
674	STS-0002	KFCA00285-W	E	sts	sts_file_name_1 ~ sts_file_name_7	—	—	sts_file_name_1 指定なし	—
675	STS-0003	KFCA01020-W	E	sts	sts_file_name_1 ~	—	—	次に示すどちらかの条件	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
675	STS-0003	KFCA01020-W	E	sts	sts_file_name_7	—	—	指定したファイルシステムがキャラクタ型スペシャルファイルではない このファイルシステムに対応する装置がない	—
676	STS-0004	KFCA01021-W	E	sts	sts_file_name_1 ～ sts_file_name_7	—	—	指定したファイルが filmkfs コマンドで OpenTP1 ファイルシステム用に初期化されていない	—
677	STS-0005	KFCA01022-W	E	sts	sts_file_name_1 ～ sts_file_name_7	—	—	ステータスファイルがない	—
678	STS-0006	KFCA01023-W	E	sts	sts_file_name_1 ～ sts_file_name_7	—	—	OpenTP1 ファイルシステムのバージョン不一致	—
679	STS-0007	KFCA01024-W	E	sts	sts_file_name_1 ～ sts_file_name_7	—	—	ステータスファイルのオープンで上限値オーバーが発生	—
680	STS-0008	KFCA01025-W	E	sts	sts_file_name_1 ～ sts_file_name_7	—	—	該当するスペシャルファイルのアクセス権限がない	—
681	STS-0009	KFCA01026-W	E	sts	sts_file_name_1 ～ sts_file_name_7	—	—	該当するステータスファイルのアクセス権限がない	—
682	STS-0010	KFCA01027-W	E	sts	sts_file_name_1 ～	—	—	ステータスファイルに対して I/O エラーが発生	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
682	STS-0010	KFCA01027-W	E	sts	sts_file_name_7	—	—	ステータスファイルに対して I/O エラーが発生	—
683	STS-0011	KFCA01028-W	E	sts	sts_file_name_1 ～ sts_file_name_7	—	—	ステータスファイルのオープンでメモリ不足が発生	—
684	STS-0012	KFCA01029-W	E	sts	sts_file_name_1 ～ sts_file_name_7	—	—	指定されたファイルはステータスファイルとして使用できない	—
685	STS-0013	KFCA01030-W	E	sts	sts_file_name_1 ～ sts_file_name_7	—	—	A 系の物理ファイルと B 系の物理ファイルでレコード長が異なっている	—
686	STS-0014	KFCA01031-W	E	sts	sts_file_name_1 ～ sts_file_name_7	—	—	A 系の物理ファイルと B 系の物理ファイルでレコード数が異なっている	—
687	STS-0015	KFCA01032-W	E	sts	sts_file_name_1 ～ sts_file_name_7	sts	sts_file_name_1 ～ sts_file_name_7	sts_file_name_1～sts_file_name_7 に指定された論理ファイル名と物理ファイル名が重複している	—
688	STS-0016	KFCA01033-W	W	sts	sts_last_active_file	—	—	sts_last_active_file 指定あり	—
689	STS-0017	KFCA01034-W	C	sts	sts_last_active_file	sts	sts_file_name_1 ～ sts_file_name_7	指定された論理ファイルが sts_file_name_1～sts_file_name_7 に存在しない	—
690	STS-0018	KFCA00278-W	C	sts	sts_last_active_file	sts	sts_initial_error_switch	(sts_last_active_file 指定があり)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
690	STS-0018	KFCA00278-W	C	sts	sts_last_active_file	sts	sts_initial_error_switch	&&(sts_initial_error_switch == stop)	—
691	STS-0019	KFCA00279-W	C	sts	sts_last_active_file	sts	sts_initial_error_switch	(sts_last_active_file 指定あり) &&(sts_initial_error_switch 指定なし)	—
692	STS-0020	KFCA01033-W	W	sts	sts_last_active_side	—	—	sts_last_active_side 指定あり	—
693	STS-0021	KFCA00278-W	C	sts	sts_last_active_side	sts	sts_single_operation_switch	(sts_last_active_side 指定あり) &&(sts_single_operation_switch != continue)	—
694	STS-0022	KFCA00279-W	C	sts	sts_last_active_side	sts	sts_single_operation_switch	(sts_last_active_side 指定あり) &&(sts_single_operation_switch 指定なし)	—
695	STS-0023	KFCA00278-W	C	sts	sts_last_active_side	sts	sts_initial_error_switch	(sts_last_active_side 指定あり) &&(sts_initial_error_switch == stop)	—
696	STS-0024	KFCA00279-W	C	sts	sts_last_active_side	sts	sts_initial_error_switch	(sts_last_active_side 指定あり) &&(sts_initial_error_switch 指定なし)	—
697	TAM-0001	KFCA00285-W	E	tam	tam_max_tblnum	—	—	tam_max_tblnum 指定なし	—
698	TAM-0002	KFCA26208-W	E	tam	tam_max_tblnum	tam	tamtable	tam_max_tblnum < tamtable 数	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
699	TAM-0003	KFCA00285-W	E	—	tam_max_recsize	—	—	tam_max_recsize 指定なし	—
700	TAM-0004	KFCA00285-W	E	—	tam_max_filesiz	—	—	tam_max_filesiz 指定なし	—
701	TAM-0005	KFCA00264-W	C	tam	tam_max_trnnum	trn	trn_trn_process_count	tam_max_trnnum > trn_trn_process_count	—
702	TAM-0006	KFCA00264-W	C	tam	tam_max_trnfile	tam	tam_max_tblnum	tam_max_tblnum < tam_max_trnfile	—
703	TAM-0007	KFCA00286-W	E	tam	—	—	—	(TAM がセットアップされている) && (TAM サービス定義がない)	—
704	—	KFCA01719-E	—	tam	tam_max_recsiz	システムジャーナルサービス定義	jnl_max_datasi	(tam_max_recsiz を 4 の整数倍に切り上げた値) × 2 + 96 < jnl_max_datasi	JNL#1 OS が Linux (IPF) の場合は 8 の整数倍に切り上げられます。
705	—	KFCA01734-E	—	tam	tamtable	—	—	テーブル名が重複している	—
706	—	KFCA01733-E	—	tam	tamtable	—	—	ファイル名が重複している	—
707	—	KFCA01740-E	—	tam	tamtable	—	—	定義ファイルで指定した TAM ファイルはスペシャルファイル名ではない	—
708	—	KFCA01741-E	—	tam	tamtable	—	—	定義ファイルで指定した TAM ファイルを割り当てるディスクパーティションは OpenTP1 ファイルシステムとして初期化されていない	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
709	—	KFCA01742-E	—	tam	tamtable	—	—	定義ファイルで指定したファイルが存在しない	—
710	—	KFCA01743-E	—	tam	tamtable	—	—	定義ファイルで指定した TAM ファイルはほかのプロセスで使用するため、使用できない	—
711	—	KFCA01747-E	—	tam	tamtable	—	—	定義ファイルで指定した TAM ファイルのキャラクター型スペシャルファイルのオープンでシステムによって上限値オーバのエラーが報告された	—
712	—	KFCA01744-E	—	tam	tamtable	—	—	定義ファイルで指定した TAM ファイルのスペシャルファイルに対するアクセス権がない	—
713	—	KFCA01745-E	—	tam	tamtable	—	—	定義ファイルで指定した TAM ファイルに対するアクセス権がない	—
714	—	KFCA01701-E	—	tam	tamtable	—	—	メモリ不足で処理を続行できない	—
715	—	KFCA01736-E	—	tam	tamtable	—	—	定義ファイルに指定された TAM ファイルで入出力エラーが発生	—
716	—	KFCA01746-E	—	tam	tamtable	—	—	定義ファイルで指定した TAM ファイルでファイルシステム作成時のシステムと実行時のシス	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
716	—	KFCA01746-E	—	tam	tamtable	—	—	テムのバージョンが異なっている	—
717	—	KFCA01786-E	—	tam	tamtable	—	—	TAM ファイルサイズ > 1000000000	—
718	—	KFCA01787-E	—	tam	tamtable	—	—	tamcre コマンドで作成された TAM ファイルではない	—
719	—	KFCA01764-E	—	tam	tamtable	—	—	現在使用中の TAM とは互換性のない TAM ファイルを使用している	—
720	—	KFCA01786-E	—	tam	tamtable	tam	tam_max_filesiz e	TAM ファイルサイズ > tam_max_filesi ze	—
721	—	KFCA02883-E	—	tam	tamtable	tam	tam_ma x_recsiz e	TAM ファイルのレコードサイ ズ > tam_max_recsi ze	—
722	TIM-00 01	KFCA00265- W	C	tim	tim_wat ch_cou nt	—	—	tim_watch_cou nt 指定あり	—
723	TRN-0 001	KFCA00285- W	E	trn	trn_tran _proces s_count	—	—	trn_tran_proce ss_count 指定 なし	—
724	TRN-0 002	KFCA00265- W	C	trn	trn_tran _proces s_count	trn	trn_rec overy_p rocess_ count	trn_tran_proce ss_count 指定 あり	—
725	TRN-0 003	KFCA00262- W	C	trn	trn_rec overy_p rocess_ count	trn	trn_tran _proces s_count	trn_recovery_p rocess_count 指定値 > trn_tran_proce ss_count 指定値	—
726	TRN-0 004	KFCA00262- W	E	trn	trn_rec overy_p	prc	prc_pro cess_co unt	trn_recovery_p rocess_count 指定値 >	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
726	TRN-0004	KFCA00262-W	E	trn	rocess_count	prc	prc_process_count	prc_process_count 指定値	—
727	TRN-0005	KFCA32523-W	C	trn ユーザー ビス定義	trn_expiration_time	trn	trn_completion_limit_time	(trn_expiration_time > 0) && (trn_completion_limit_time > 0) && (trn_expiration_time 指定値 > trn_completion_limit_time 指定値)	—
728	TRN-0006	KFCA00282-W	C	trn ユーザー ビス定義	trn_statistics_item	—	—	trn_statistics_item に cputime 指定あり	OS が Windows の場合、チェック対象の定義はサポートしていません。
729	TRN-0007	KFCA00278-W	C	trn ユーザー ビス定義	trn_expiration_time_suspend	trn	trn_expiration_time	(trn_expiration_time_suspend 指定あり) && (trn_expiration_time == 0)	—
730	TRN-0008	KFCA00279-W	C	trn ユーザー ビス定義	trn_expiration_time_suspend	trn	trn_expiration_time	(trn_expiration_time_suspend 指定あり) && (trn_expiration_time 指定なし)	—
731	TRN-0009	KFCA00272-W	C	trn	trn_tran_recovery_list	—	—	trn_tran_recovery_list != Y	—
732	TRN-0010	KFCA00282-W	C	trn ユーザー ビス定義	trn_cpu_time	—	—	trn_cpu_time 指定あり（ただし 0 を指定した場合だけ）	OS が Windows の場合、チェック対象の定義はサポートしていません。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
733	TRN-0011	KFCA00265-W	C	trn	trn_max_subordinate_count	—	—	trn_max_subordinate_count 指定あり	—
734	TRN-0012	KFCA00272-W	C	trn ユーザー ビス定義	trn_rm_open_close_scope	—	—	trn_rm_open_close_scope != process	—
735	TRN-0013	KFCA00272-W	C	trn ユーザー ビス定義	trn_optimum_item	—	—	trn_optimum_item に base 指定なし	—
736	TRN-0014	KFCA00272-W	C	trn	trn_processing_in_rm_error	—	—	(trn_processing_in_rm_error != down) && (trn_processing_in_rm_error != retry)	—
737	TRN-0015	KFCA00278-W	C	trn	trn_recovery_list_remove	trn	trn_tran_recovery_list	(trn_recovery_list_remove 指定あり) && (trn_tran_recovery_list == N)	—
738	TRN-0016	KFCA00279-W	C	trn	trn_recovery_list_remove	trn	trn_tran_recovery_list	(trn_recovery_list_remove 指定あり) && (trn_tran_recovery_list 指定なし)	—
739	TRN-0017	KFCA00272-W	C	trn	trn_recovery_list_remove	trn	trn_tran_recovery_list	(trn_tran_recovery_list == Y) && ((trn_recovery_list_remove == no) (trn_recovery_list_remove 指定なし))	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
740	TRN-0018	KFCA00278-W	C	trn	trn_recovery_list_remove_level	trn	trn_tran_recovery_list	(trn_recovery_list_remove_level 指定あり) && (trn_tran_recovery_list == N)	—
741	TRN-0019	KFCA00279-W	C	trn	trn_recovery_list_remove_level	trn	trn_tran_recovery_list	(trn_recovery_list_remove_level 指定あり) && (trn_tran_recovery_list 指定なし)	—
742	TRN-0020	KFCA00282-W	C	trn	trn_crm_use	—	—	trn_crm_use 指定あり	—
743	TRN-0021	KFCA00282-W	C	trn	trn_max_crm_subordinate_count	—	—	trn_max_crm_subordinate_count 指定あり	—
744	TRN-0022	KFCA00262-W	C	trn ユーザーサービス定義	trn_watch_time	trn	trn_limit_time	(trn_limit_time != 0) && (trn_watch_time 指定値 > trn_limit_time 指定値)	TRN#1 すべての定義で trn_watch_time を省略した場合は watch_time の指定値を採用します。 watch_time に 0 を指定した場合は trn_watch_time のデフォルト値として 120 を採用します。
745	TRN-0023	KFCA00262-W	C	trn ユーザーサービス定義	trn_watch_time	trn	trn_completion_limit_time	(trn_completion_limit_time != 0) && ((trn_watch_time 指定値 > trn_completion_limit_time 指定値))	TRN#1 すべての定義で trn_watch_time を省略した場合は watch_time の指定値を採用します。 watch_time に

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
745	TRN-0023	KFCA00262-W	C	trn ユーザーサービス定義	trn_watch_time	trn	trn_completion_limit_time	(trn_completion_limit_time != 0) && ((trn_watch_time 指定値 > trn_completion_limit_time 指定値))	0 を指定した場合は trn_watch_time のデフォルト値として 120 を採用します。
746	TRN-0024	KFCA00272-W	C	trn ユーザーサービス定義	trn_rollback_information_put	—	—	trn_rollback_information_put != all trn_rollback_information_put 指定なし	—
747	TRN-0025	KFCA00272-W	C	trn	trn_recovery_failmsg_interval	—	—	trn_recovery_failmsg_interval == 0	—
748	TRN-0026	KFCA00278-W	C	trn	trn_retry_interval_rm_open	trn	trn_wait_rm_open	(trn_retry_interval_rm_open 指定あり) && (trn_wait_rm_open != retry_continue) && (trn_wait_rm_open != retry_stop)	—
749	TRN-0027	KFCA00278-W	C	trn	trn_retry_count_rm_open	trn	trn_wait_rm_open	(trn_retry_count_rm_open 指定あり) && (trn_wait_rm_open != retry_continue) && (trn_wait_rm_open != retry_stop)	—
750	TRN-0028	KFCA32521-W	C	trn	thread_stack_size	—	—	thread_stack_size 指定あり	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
751	TRN-0029	KFCA00279-W	C	trn	trn_start_recovery_mode	trn	trnstring	(trn_start_recovery_mode 指定あり) && (trnstring 指定なし)	—
752	TRN-0030	KFCA00279-W	C	trn	trn_start_recovery_mode	trn	trnstring	(trn_start_recovery_mode 指定あり) && (trnstring の-m オプション 指定なし)	—
753	TRN-0031	KFCA00279-W	C	trn	trn_start_recovery_watch_time	trn	trnstring	(trn_start_recovery_watch_time 指定あり) && (trnstring 指定なし)	—
754	TRN-0032	KFCA00279-W	C	trn	trn_start_recovery_watch_time	trn	trnstring	(trn_start_recovery_watch_time 指定あり) && (trnstring の-m オプション 指定なし)	—
755	TRN-0033	KFCA00279-W	C	trn	trn_start_recovery_interval	trn	trnstring	(trn_start_recovery_interval 指定あり) && (trnstring 指定なし)	—
756	TRN-0034	KFCA00279-W	C	trn	trn_start_recovery_interval	trn	trnstring	(trn_start_recovery_interval 指定あり) && (trnstring の-m オプション 指定なし)	—
757	TRN-0035	KFCA00278-W	C	trn	trn_start_recovery_interval	trn	trn_start_recovery_watch_time	(trn_start_recovery_interval 指定あり) && (trnstring の-m オプション 指定あり) && (trn_start_recovery_watch_time == 0)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
758	TRN-0036	KFCA00272-W	C	trn	trn_xa_commit_error	—	—	trn_xa_commit_error != down	—
759	TRN-0037	KFCA00272-W	C	trn	trn_prf_event_trace_level	—	—	(trn_prf_event_trace_level & 00000007) != 00000007	—
760	TRN-0038	KFCA00272-W	C	trn	trn_prf_event_trace_condition	—	—	xafunc 指定なし	—
761	TRN-0039	KFCA00259-W	C	trn	trnstring	trn	trnstring	ほかの trnstring の -n オプション, または -n オプションと -i オプションを同時に指定したりソースマネージャ名と重複	—
762	TRN-0040	KFCA32524-W	E	ユーザーサービス定義	trnrmid	trn	trnstring	trnrmid の -n オプションと -i オプションの組み合わせが trnstring に指定されていない	—
763	TRN-0041	KFCA32522-W	C	_tr	prf_file_size	—	—	prf_file_size < デフォルト値 (10240)	—
764	TRN-0043	KFCA00286-W	E	trn	全体	—	—	トランザクションサービス定義ファイル (trn) が存在しない	—
765	TRN-0044	KFCA00261-W	C	trn ユーザーサービス定義	trn_expiration_time	trn ユーザーサービス定義	trn_cpu_time	(trn_expiration_time > 0) && (trn_cpu_time > 0) && (trn_expiration_time < trn_cpu_time)	OS が Windows の場合はチェックしません。

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
766	TRN-0045	KFCA00262-W	C	trn ユーザーサービス定義	trn_cpu_time	trn ユーザーサービス定義	trn_completion_limit_time	(trn_cpu_time > 0) && (trn_completion_limit_time > 0) && (trn_cpu_time > trn_completion_limit_time)	OS が Windows の場合はチェックしません。
767	TRN-0046	KFCA00278-W	C	trn	trn_max_crm_subordinate_count	trn	trn_crm_use	(trn_max_crm_subordinate_count 指定あり) && (trn_crm_use == N)	OS が Windows の場合はチェックしません。
768	TRN-0047	KFCA00279-W	C	trn	trn_max_crm_subordinate_count	trn	trn_crm_use	(trn_max_crm_subordinate_count 指定あり) && (trn_crm_use 省略)	OS が Windows の場合はチェックしません。
769	TRN-0048	KFCA00265-W	C	trn	trn_max_crm_subordinate_count	trn	trn_crm_use	trn_max_crm_subordinate_count 指定あり && (trn_crm_use == Y)	OS が Windows の場合はチェックしません。
770	UTL-0001	KFCA00271-W	W	betranrc usrrc ユーザーサービス定義	rpc_trace	—	—	rpc_trace == Y	—
771	UTL-0002	KFCA00268-W	W	betranrc usrrc ユーザーサービス定義	rpc_trace_name	—	—	(rpc_trace_name 指定あり) && (rpc_trace_name に指定した RPC トレースファイルがファイルではない)	UTL#1 UTL#2
772	UTL-0003	KFCA00268-W	W	betranrc usrrc	rpc_trace_name	—	—	(rpc_trace_name 指定あり) &&	UTL#1 UTL#2

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
772	UTL-0003	KFCA00268-W	W	ユーザサービス定義	rpc_trace_name	—	—	(rpc_trace_name に指定したRPC トレースファイルに書き込み権限がない)	UTL#1 UTL#2
773	UTL-0004	KFCA00268-W	W	betranrc usrsrc ユーザサービス定義	rpc_trace_name	—	—	(OS == Windows) && ((rpc_trace_name 指定あり) && (rpc_trace_name 指定値 > 259)) (OS != Windows) && ((rpc_trace_name 指定あり) && (rpc_trace_name 指定値 > 1023))	UTL#1 UTL#3
774	UTL-0005	KFCA00267-W	W	betranrc usrsrc ユーザサービス定義	rpc_trace_name	—	—	(rpc_trace_name 指定あり) && (RPC トレースファイルの出力先ディレクトリが存在しない)	UTL#1
775	UTL-0006	KFCA00267-W	W	betranrc usrsrc ユーザサービス定義	rpc_trace_name	—	—	(rpc_trace_name 指定あり) && (RPC トレースファイルの出力先ディレクトリがディレクトリではない)	UTL#1
776	UTL-0007	KFCA00267-W	W	betranrc usrsrc ユーザサービス定義	rpc_trace_name	—	—	(rpc_trace_name 指定あり) && ((RPC トレースファイルの出力先ディレクトリの書き込み権限がない)	UTL#1

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
776	UTL-0007	KFCA00267-W	W	betranrc usrsrc ユーザサービス定義	rpc_trace_name	—	—	 (RPC トレースファイルの出力先ディレクトリの実行権限がない))	UTL#1
777	UTL-0008	KFCA00287-W	W	betranrc usrsrc ユーザサービス定義	rpc_trace_name	—	—	(rpc_trace_name 指定あり) && (rpc_tarce_name 指定値のファイル名長 > 13byte)	メッセージに出力する rpc_trace_name に指定したファイル名長が長い場合、371 バイト以降は省略されます。
778	UTL-0009	KFCA00276-W	C	betranrc usrsrc ユーザサービス定義	rpc_trace_name	ユーザサービス定義	rpc_trace	(rpc_trace != Y) && (rpc_trace_name 指定あり)	UTL#4 メッセージに出力する rpc_trace_name に指定したファイルパス名長が長い場合、346 バイト以降は省略されます。
779	UTL-0010	KFCA00276-W	C	betranrc usrsrc ユーザサービス定義	rpc_trace_size	ユーザサービス定義	rpc_trace	(rpc_trace != Y) && (rpc_trace_size 指定あり)	UTL#4
780	UTL-0011	KFCA00272-W	W	Usrrc ユーザサービス定義	uap_trace_max	—	—	(uap_trace_max < 32)	—
781	UTL-0012	KFCA00288-W	W	betranrc usrsrc ユーザサービス定義	rpc_trace_name	—	—	(rpc_trace_name 指定値内に '\$'がある) && (先頭ではない位置に '\$'が指定されている)	—
782	XAR-0001	KFCA00271-W	W	Xar	xar_eventtrace_level	—	—	xar_eventtrace_level != ERR	—
783	XAR-0002	KFCA32164-W	W	Xar	xarfile	xar	xar_msdtc_use	(xar_msdtc_use == Y)	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
783	XAR-0002	KFCA32164-W	W	Xar	xarfile	xar	xar_msdtc_use	&& (XAR ファイルのレコード長 < 1024)	—
784	XAR-0003	KFCA00286-W	E	Xar	—	trn	trn_xar_use	(trn_xar_use == Y) && (XA リソースサービス定義がない)	—
785	XAR-0004	KFCA32172-W	E	—	—	trn betranr c	trn_xar_use jnl_fileless_option	(trn_xar_use == Y) && (jnl_fileless_option == Y)	—
786	XAR-0005	KFCA00272-W	C	Xar	xar_prf_trace_level	—	—	(xar_prf_trace_level & 00000003) != 00000003	—
787	XAR-0006	KFCA32163-W	C	_xr	prf_file_size	—	—	prf_file_size < デフォルト値 (10240)	—
788	—	KFCA32019-E	—	Xar	xarfile	—	—	xarfile -t online -a XAR ファイル, xarfile -t backup -a XAR ファイルの組み合わせで指定していない	—
789	—	KFCA32025-E	—	Xar	xarfile	—	—	XAR ファイルパス名が正しくない	—
790	—	KFCA32026-E	—	Xar	xarfile	—	—	OpenTP1 ファイルシステムへのアクセス権がない	—
791	—	KFCA32027-E	—	Xar	xarfile	—	—	オープンファイル数がシステムの上限值を超過	—
792	—	KFCA32028-E	—	Xar	xarfile	—	—	XAR ファイルに対して I/O エラーが発生	—
793	—	KFCA32029-E	—	Xar	xarfile	—	—	メモリ不足が発生	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
794	—	KFCA3 2030-E	—	Xar	xarfile	—	—	XAR ファイルのバージョンが正しくない	—
795	—	KFCA3 2031-E	—	Xar	xarfile	—	—	OpenTP1 ファイルシステムのある装置が未初期化	—
796	—	KFCA3 2032-E	—	Xar	xarfile	—	—	XAR ファイル名が正しくない	—
797	—	KFCA3 2036-E	—	Xar	xarfile	—	—	XAR ファイルが存在しない	—
798	—	KFCA3 2038-E	—	Xar	xarfile	—	—	XAR ファイルに対するアクセス権がない	—
799	—	KFCA3 2111-E	—	Xar	xarfile	—	—	定義に指定したファイルが XAR ファイルではない	—
800	—	KFCA3 2044-E	—	Xar	xarfile	—	—	定義で指定した XAR ファイルのバージョンが不正	—
801	—	KFCA3 2021-E KFCA3 2024-E	—	Xar	xarfile	trn	trn_tran _proces s_count	XAR ファイルのレコード数 < trn_tran_proce ss_count	—
802	—	KFCA3 2014- W	—	Xar	xarfile	—	—	定義で指定した XAR ファイルをクローズできない	—
803	—	KFCA3 2022-E KFCA3 2024-E	—	Xar	xarfile	—	—	オンライン系 XAR ファイルのレコード数 != バックアップ系 XAR ファイルのレコード数	—
804	—	KFCA3 2023-E KFCA3 2024-E	—	Xar	xarfile	—	—	オンライン系 XAR ファイルのレコード長 != バックアップ系	—

項番	コード	ID	区分	ファイル名	定義	関連ファイル名	関連定義	条件式または判断根拠	備考
804	—	KFCA3 2023-E KFCA3 2024-E	—	Xar	xarfile	—	—	XAR ファイルの レコード長	—
805	—	KFCA3 2048-E	—	Xar	xarfile	—	—	オンライン系 XAR ファイル名 == バックアップ系 XAR ファ イル名	—

(凡例)

—：該当しません。

↑↑：小数点以下を切り上げます。

ADM#1：絶対パスで指定された場合だけチェックします。

ADM#2：OS が Windows の場合で、拡張子が指定されていないときは、".bat", ".exe", ".com"の三つの拡張子を自動的に付与してチェックします。

ADM#3：ファイルそのものが実行可能なファイルであるかどうかのチェックを試みます。

CPD#1：システム共通定義の jnl_fileless_option オペランドに Y を指定した場合はチェックしません。

JNL#1：システム共通定義の jnl_fileless_option オペランドに Y を指定した場合はチェックしません。

PRC#1：OS が Windows の場合はドライブ文字を含めて'¥'であるかどうかをチェックします。

PRC#2：OS が Windows の場合は';'でチェックします。

PRC#3：OS が Windows の場合は\$DCDIR/bin はチェックしません。

PRC#4：OS が Windows の場合は指定された名称に".exe"の拡張子を付与してチェックします。また、prcsvpath の指定に関係なく自動的に\$DCDIR/bin に関してもチェックします。

RPC#1：指定可能な定義ファイルがすべて省略されている場合は、出力メッセージの「定義ファイル名」に"*****"が出力されることがあります。

RTS#1：rts_log_file_name の指定値に 1 から rts_log_file_count の指定値までの値を付与したファイルパス名でチェックします。

RTS#2：RTS ログファイルのパス名に".bk"の拡張子を付与したファイルパス名でチェックします。

SCD#1：指定可能な定義ファイルがすべて省略されている場合は、出力メッセージの「定義ファイル名」に"*****"が出力されることがあります。

SCD#2：KFCA00265-W メッセージが出力された場合は、MHP として起動するすべてのユーザサービス定義を見直してください。

TRN#1：指定可能な定義ファイルがすべて省略されている場合は、出力メッセージの「定義ファイル名」に"*****"が出力されることがあります。

UTL#1：絶対パスで指定された場合だけチェックします。

UTL#2：rpc_trace_name で指定したファイルパス名称に 1, 2 および l を付与してチェックします。

UTL#3：rpc_trace_name の指定値の長さをチェックします。

UTL#4：指定可能な定義ファイルがすべて省略されている場合は、出力メッセージの「定義ファイル名」に"*****"が出力されることがあります。

注

出力メッセージの「定義ファイル名」に"*****"が出力された場合は、チェック対象の定義が指定できるすべての定義ファイルを見直してください。

付録 E 旧製品からの移行に関する注意事項

ここでは、バージョン 5 以前からバージョン 7 に移行する場合の注意事項について説明します。

付録 E.1 バージョン 5 以前からの移行

バージョン 5 以前からバージョン 7 に移行する場合の注意事項について説明します。

バージョン 5 以前とバージョン 7 では、オペランドのデフォルト値が変更されています。オペランドのデフォルト値の変更点を、次の表に示します。

表 E-1 オペランドのデフォルト値の変更点

定義名／ 定義コマンド名	オプション	オペランド名／ コマンド引数	バージョン 5 以 前のデフォル ト値	バージョン 7 の デフォルト値
システム共通定義	—	thdlock_sleep_time	300 (単位：ミリ 秒)	15 (単位：ミリ 秒)
ロックサービス定義	—	lck_release_detect	interval	pipe
mcfttim	-t	btim	5 (単位：秒)	1 (単位：秒)
mcfttrc	-t	size	20480 (単位： バイト)	204800 (単位： バイト)
システムサービス共通情報定義	—	max_open_fds	100	500
	—	thdlock_sleep_time	300 (単位：ミリ 秒)	15 (単位：ミリ 秒)

(凡例)
—：該当しません。

また、バージョン 5 以前とバージョン 7 では、オペランドと指定値の対応に変更があります。対応するオペランドの変更点を、次の表に示します。

表 E-2 オペランドと指定値の対応の変更点

定義名／ 定義コマンド名	オペランド名／ コマンド引数	バージョン 7 の指定値	バージョン 5 以前の指定値
トランザクションサービ ス定義	trn_optimum_item	asyncprepare	rpc_prepare_in_advance オペランドに Y を指定
		base	rpc_prepare_in_advance オペランドに N を指定

バージョン 5 以前からバージョン 7 に移行する場合は、バージョン 6 からバージョン 7 での変更点について
も注意する必要があります。

バージョン 6 からバージョン 7 での変更点については、マニュアル「OpenTP1 解説」のバージョンアップ時の変更点について説明している個所を参照してください。

索引

数字

1 プロセスが処理するサービス要求数 380, 511, 545

A

abort 655
adm_message_option 516, 563
all_node 152
all_node_ex 168
all_node_ex_extend_number 177
all_node_extend_number 177
API の代理実行要求時, rap サーバとのコネクションの切断を検知した場合に, エラーメッセージを出力 579
aplihold 680
assurance_count 307
atomic_update 510, 540
auto_restart 511, 546
A 系ステータスファイル名 281

B

balance_count 380, 511, 545
btim 663
bufcnt 666
B 系ステータスファイル名 281

C

cancel_normal_terminate 514, 554
client_uid_check 162
clt_conf 142
clt_cup_conf 383
clt_inquire_time 382
clt_port 383
clt_trn_conf 383
cltcon_port 385
clttrn_port 384
cname 680
COBOL API の排他レベル 374

commitdml 643
core_shm_suppress 521, 580
core_suppress_watch_time 163
core ファイルに共有メモリを含めるかの判断値 184, 523, 589
count 669
cpd_message_id_change_level 309
critical 511, 546
CRM 252
CRM を使用 252
cup_balance_count 384
cup_parallel_count 383
CUP 実行プロセスのポート番号 385
CUP 実行プロセスを起動 383

D

dam_added_file 356
dam_cache_attribute 361
dam_cache_reuse_from 363
dam_cache_size 358
dam_cache_size_fix 358
dam_conf 140
dam_default_cache_num 363
dam_ex_refer_read 364
dam_io_error_occur 362
dam_io_interval 361
dam_kb_size 366
dam_max_block_size 365
dam_message_level 357
dam_tran_process_count 357
dam_transaction_access 361
dam_update_block 353
dam_update_block_over 357
damcache 367
damchlimt 369
damfile 370

DAM キャッシュブロックの確保回数イベントを取得 496

DAM キャッシュ用の共用メモリの使用率イベントを取得 496

DAM サービスが出力するメッセージのレベル 357

DAM サービス定義 353

DAM サービスの最大同時排他要求数 195

DAM サービスの処置 362

DAM サービスを使用 141

DAM 情報の read イベントを取得 495

DAM 情報の read エラーイベントを取得 495

DAM 情報の write イベントを取得 495

DAM 情報の write エラーイベントを取得 495

DCADMDEBUG 137

dcbindht 191

DCCONFPATH 137, 138

DCFPL_CONNECT_RETRY_COUNT 591

DCFPL_CONNECT_RETRY_INTERVAL 592

dcmarea 337

DCMCFQUEBAK 697

dcmstart_watch_time 335

dcmstart 打ち切り時間 335

dcmstop_watch_time 335

dcmstop 打ち切り時間 335

dcprcport 340

dcputenv 形式 44

dcreset コマンド 702

dcsetup コマンド 702

dcsetup -d コマンド 702

dcstart_wakeup_retry_count 181

dcstart_wakeup_retry_interval 181

dcsvgdef 418

dcsvstart 144, 146

dcsvstop コマンドでの正常終了を抑止 554

DCSYSLOGOUT 333

DCUAPCONFPATH 138

deadlock_priority 509, 537

default_value_option 136

delaytim 643

disk 665

diskitq 659

dmsgcnt 673

domain_masters_addr 161

domain_masters_port 161

domain_use_dns 161

dynamic_shmpool_size 131

E

ENOBUFS, ENOMEM 発生時のリトライ回数 174

ENOBUFS, ENOMEM 発生時のリトライ間隔 174

errevt 681

errevt_recovery 645

ERREVT1 679

ERREVT2 679

ERREVT3 679

ERREVT4 679

ERREVTA 663

evtlogout 688

expectcnt 654

extend 669

F

fil_prf_trace_delay_time 183

fil_prf_trace_option 182

FIL イベントトレース取得条件となるファイルアクセス処理時間のしきい値 183

FJ 出力回数イベントを取得 496

G

gj 687

groupno 669

groups 261, 520, 577

H

ha_conf 141

hold 509, 536

hold_recovery 509, 536

holdlimit 685

holdlmtyp 685

I

ij 687

initseq 642

ipc_backlog_count 174, 520, 578

ipc_conn_interval 166, 518, 573

ipc_header_rcv_time 167, 519, 573

ipc_listen_sockbufset 171, 519, 574

ipc_rcvbuf_size 172, 520, 575

ipc_send_count 166, 519, 573

ipc_send_interval 166, 519, 573

ipc_sendbuf_size 173, 520, 576

ipc_sockctl_highwater 164, 449, 518, 571

ipc_sockctl_watctime 165, 449, 518, 572

ipc_tcpnodelay 235, 522, 583

ist_conf 142

ist_node 394

ist_node_group 394

istdef 396

IST サービス定義 394

IST サービスのテーブル名の定義 396

IST サービスを使用 142

J

jar_conf 141

jnl_arc_buff_size 297

jnl_arc_check_level 299

jnl_arc_ipc_buff_size 289, 343

jnl_arc_max_datasize 297, 349

jnl_arc_name 296

jnl_arc_rec_kind 298

jnl_arc_terminate_check 297

jnl_arc_terminate_timeout 287, 342

jnl_arc_trn_stat 299

jnl_arc_uj_code 298

jnl_auto_unload 301

jnl_auto_unload_path 301

jnl_cdinterval 295

jnl_cdskip_limit 309

jnl_cdskip_msg 309

jnl_dual 296, 308, 347

jnl_fileless_option 184

jnl_max_datasize 292, 307

jnl_max_file_dispersion 301, 348

jnl_message_id_change_level 289, 344

jnl_min_file_dispersion 302, 348

jnl_objservername 306

jnl_prf_event_trace_level 183

jnl_reduced_mode 307

jnl_rerun_reserved_file_open 296

jnl_rerun_swap 295, 348

jnl_reserved_file_auto_open 308

jnl_singleoperation 296, 308, 347

jnl_tran_optimum_level 286

jnl_unload_check 300, 348

jnl_watch_time 289, 344

jnladdfg 303, 311, 350

jnladdpf 304, 313, 351

jnldfs 291, 346

JNL 性能検証用トレース情報ファイルの世代数 465

JNL 性能検証用トレース情報ファイルのバックアップ
を取得 465

JNL 性能検証用トレース定義 464

JNL 性能検証用トレースの取得レベル 183

jp1_use 157

JP1 イベントサービスへの登録機能 316

K

KFCA00356-W メッセージを出力するバンドル回数
169

KFCA00853-E メッセージ出力インタバル時間 521,
581

KFCA00853-E メッセージ出力判断値 521, 582

kind 679

L

LANG 190

LANG 設定値 190

lck_deadlock_info 196
lck_deadlock_info_remove 197
lck_deadlock_info_remove_level 197
lck_limit_fordam 195
lck_limit_formqa 196
lck_limit_fortam 196
lck_limit_foruser 195
lck_prf_trace_level 198
lck_release_detect 197
lck_release_detect_interval 198
lck_wait_priority 512, 546
lck_wait_timeout 196
LCK 性能検証用トレース関連のメッセージの表示レベル 468
LCK 性能検証用トレース情報の取得レベル 198
LCK 性能検証用トレース情報ファイルのサイズ 467
LCK 性能検証用トレース情報ファイルの世代数 468
LCK 性能検証用トレース定義 467
length 669
leveldown_queue_count 518, 571
levelup_queue_count 518, 571
lname 680
loadcheck_interval 518, 570
loadlevel_message 520, 577
log_audit_count 331
log_audit_message 332, 448, 456, 523, 587
log_audit_out 330
log_audit_out_suppress 448, 456, 523, 586
log_audit_path 330
log_audit_size 331
log_filesize 316
log_jerr_rint 325
log_jp1_allno 321
log_jp1_date 322
log_jp1_hostname 322
log_jp1_pgmid 323
log_jp1_prcid 321
log_jp1_prcno 321
log_jp1_sysid 322
log_jp1_time 322
log_msg_allno 317
log_msg_console 316
log_msg_date 318
log_msg_hostname 318
log_msg_pgmid 318
log_msg_prcid 317
log_msg_prcno 317
log_msg_sysid 317
log_msg_time 318
log_netm_allno 319
log_netm_date 320
log_netm_hostname 320
log_netm_out 319
log_netm_pgmid 321
log_netm_prcid 319
log_netm_prcno 319
log_netm_sysid 320
log_netm_time 320
log_notify_allno 323
log_notify_date 324
log_notify_hostname 325
log_notify_out 323
log_notify_pgmid 325
log_notify_prcid 323
log_notify_prcno 324
log_notify_sysid 324
log_notify_time 324
log_syslog_allno 327
log_syslog_append_nodeid 329
log_syslog_date 328
log_syslog_elist 329
log_syslog_elist_rint 329
log_syslog_hostname 328
log_syslog_out 325
log_syslog_pgmid 328
log_syslog_prcid 327
log_syslog_prcno 327
log_syslog_synchro 330

log_syslog_sysid 327

log_syslog_time 328

M

make_queue_on_starting 517, 570

max_open_fds 233, 408, 514, 557, 694

max_socket_descriptors 159, 204, 222, 232, 258, 287, 342, 408, 444, 514, 556, 693

max_socket_msg 512, 548

max_socket_msglen 512, 549

maxseq 642

MCF 31

mcf_jnl_buff_size 511, 545

mcf_mgrid 513, 549

mcf_prf_trace 523, 587, 691

mcf_prf_trace_level 695

mcf_psv_id 512, 547

mcf_service_max_count 513, 549

mcf_spp_oj 516, 563

mcf_stack_size 696

mcf_start_watch_interval 696

mcfaalcap 678

mcfacnv 677

mcfapli 626

mcfapli 627

mcfapli コマンド 625

mcfcomn 625

mcfcomnr 627

mcflink 626

mcflink コマンド 625

mcfmcname 638

mcfmcomn 634

mcfmenv 633

mcfmexp 650

mcfmngr 625

mcfmngr 627

mcfmngrr コマンド 625

mcfmqgid 648

mcfmsmsg 653

mcfmsts 652

mcfmsvg 654

mcfmuap 639

mcfpsvr 625

mcfpsvrr 627

mcfsvname 638

mcftalcle 672

mcftbuf 669

mcftcomn 660

mcftcpr 627

mcftenv 658

mcftped 674

mcftpsvr 670

mcftsts 668

mcfttim 663

mcfttrc 665

mcfttred 662

mcfxpr 628

MCF アプリケーション定義 675, 743

MCF アプリケーション定義オブジェクトファイル名 658

MCF アプリケーション定義コマンドの指定順序 675

MCF アプリケーション定義識別子 677

MCF イベント名 679

MCF オンラインコマンドサービス名 636

MCF オンラインコマンドプロセス 610

MCF 開始時プロセス間監視時間間隔 696

MCF 稼働統計情報を取得 636

MCF 環境定義 658

MCF 構成変更準備停止時のバックアップファイル名 697

MCF サービスと定義との関係 610

MCF サービス用の共用メモリの見積もり式 774

MCF 作業領域長 634

MCF 実行形式プログラム名 691

MCF 性能検証用トレース情報の取得レベル 695

MCF 性能検証用トレース定義 689

MCF 通信関数発行回数上限値 513, 549

MCF 通信構成共通定義 660

MCF 通信構成定義 656
MCF 通信構成定義のファイル構成 657
MCF 通信サービス名 638
MCF 通信プロセス 610
MCF 通信プロセス識別子 658
MCF 通信プロセスでアクセスするファイルの最大数 694
MCF 通信プロセスまたはアプリケーション起動プロセスのジャーナルバッファの大きさ 660
MCF 定義オブジェクト解析コマンド 627
MCF 定義オブジェクト生成ユーティリティ 624
MCF 定義オブジェクト生成ユーティリティ起動コマンド 625
MCF 定義結合ユーティリティ 624
MCF 定義結合ユーティリティ起動コマンド 626
MCF の運用形態 611
MCF の運用形態と定義との関係 611
MCF のジャーナルバッファの大きさ 511, 545
MCF マネジャ環境定義 633
MCF マネジャ共通定義 634
MCF マネジャ定義 632
MCF マネジャ定義コマンドの指定順序 632
MCF マネジャプロセス 610
MCF マネジャプロセス識別子 633, 658
MCF マネジャプロセスのジャーナルバッファの大きさ 635
MCF マネジャ名 633
mcf×××× 625, 627
message_bufllen 509, 537
message_cell_size 512, 548
message_store_bufllen 392, 510, 537
MHP に期待するサービス要求の処理数 654
minseq 642
mmsgcnt 673
mode_conf 129
modelname 682
module 532, 691
mqa_conf 143
MQA サービスの最大同時排他要求数 196
MQA サービスを使用 143

MQT 通信構成定義 35
MSDTC 連携機能を使用 276
msgcnt 680
msgout 664
msgsize 664, 666
mtim 663
multi_node_option 155
multi_schedule 517, 569
my_host 157

N

nam_prf_trace_level 181
name 679
name_audit_conf 208
name_audit_interval 210
name_audit_watch_time 210
name_cache_size 203
name_cache_validity_time 212
name_domain_file_use 175
name_global_lookup 206
name_manager_node 187
name_node_add_policy 188
name_nodeid_check_message 212
name_notify 167
name_port 151
name_remove_down_node 188
name_rpc_control_list 211
name_service_extend 208
name_service_mode 186
name_start_error 216
name_start_retry_count 215
name_start_retry_interval 215
name_start_watch_time 215
name_sync_ready_time 216
name_total_size 202
namnlfil 217
NAM イベントトレースの取得レベル 181
NETM 出力機能 316
NETM に出力 319

nice 429, 509, 535
noansreply 643
node_down_restart 513, 551
node_id 154
ntmetim 642, 684

O

oj 687
open_rm 382
OpenTP1 管理者のユーザ識別子 430, 455
OpenTP1 起動時および再開始（リラン）時に有効にするドメイン構成を指定 175
OpenTP1 起動時のノードリスト要求応答待ち時間 215
OpenTP1 起動時のノードリスト要求リトライ回数 215
OpenTP1 起動時のノードリスト要求リトライ間隔 215
OpenTP1 起動通知エラーリトライ回数 181
OpenTP1 起動通知エラーリトライ間隔 181
OpenTP1 識別子 152
OpenTP1 識別子を付ける 317, 320, 322, 324, 327
OpenTP1 システムごとに異なる必要がある定義内容 699
OpenTP1 システム定義の構成 37
OpenTP1 システム定義の体系 32
OpenTP1 システム内で稼働するマネジャノードのノード名 187
OpenTP1 システムの処置 546
OpenTP1 システムの定義の変更 701
OpenTP1 システムの変更に影響する定義 713
OpenTP1 通信時または系切り替え機能使用時のホスト名の指定 191
OpenTP1 提供以外のリソースマネジャの登録 403
OpenTP1 内部で使用するスレッドスタック領域（内部通信用）のサイズ 696
OpenTP1 内部で使用するスレッドスタック領域（メッセージ制御用）のサイズ 696
OpenTP1 内部で使用するスレッドスタック領域のサイズ 260
OpenTP1 のパラダイム 552

OpenTP1 のパラダイムに従うか, XATMI のパラダイムに従うかを指定 552
OpenTP1 も同時にダウン 398
order 643
OSI TP 通信のアソシエーションによって使用される FDS の最大値 408
OSI TP 通信用のスタブを使用 560
otqinhold 644

P

parallel_count 379, 509, 535
polling_control_data 260, 520, 576
prc_abort_signal 514, 554
prc_corecompress 225
prc_coredump_filter 184, 523, 589
prc_coresave_path 225
prc_current_work_path 178
prc_port 156
prc_prf_trace 224
prc_process_count 219
prc_recovery_resident 220
prc_take_over_dlpath 221
prc_take_over_svpath 220
prcsvpath 227
preend_warning_watch_time 135
prf_buff_size 460
prf_file_count 459, 463, 465, 468, 470, 689
prf_file_size 458, 462, 464, 467, 469, 689
prf_information_level 458, 462, 467, 469
prf_trace 163
prf_trace_backup 459, 465
purge_msgget 514, 553
putenv 形式 44

Q

que_conf 141
que_io_maxrecsize 410
que_xidnum 410
quegrp 412

quegrpid 648, 673, 683

quekind 648, 673, 683

R

rap_autoconnect_con_error_msg 521, 579

rap_client_manager_node 441

rap_client_manager_port 455

rap_connect_interval 443

rap_connect_retry_count 442

rap_connect_retry_interval 442

rap_connection_assign_type 440

rap_extend_function 450, 523, 588

rap_inquire_time 429

rap_inquire_timeout_message 440

rap_io_retry_interval 442

rap_listen_backlog 442

rap_listen_inf 455

rap_listen_port 428

rap_max_buff_size 441

rap_max_client 441

rap_message_id_change_level 445, 523, 586

rap_msg_output_interval 443

rap_notify 441

rap_parallel_server 428

rap_recovery_server 443

rap_sock_count 442

rap_sock_interval 442

rap_stay_warning_interval 448

rap_stay_watch_time 447

rap_term_disconnect_time 447

rap_watch_time 428, 456

rap クライアントマネージャサービス定義 455

rap クライアントマネージャに対し起動通知を行う 441

rap クライアントマネージャのポート番号 455

rap クライアントマネージャへ起動通知 441

rap サーバ数 428

rap サーバ用ユーザサービス定義 428

rap サーバ割り当て待ち要求の滞留監視時間 447

rap リスナーおよび rap サーバから出力される監査ログを抑止 448

rap リスナーサービス定義 427

rap リスナー終了時のコネクション切断待ち時間 447

rap リスナーに同時接続する最大クライアント数 441

rap リスナーポート番号 428

rap リスナー用ユーザサービス定義 428

read メッセージイベントを取得 487

receive_from 510, 541

recvmsg 683

recvtim 640

reruntm 670

reschedulecnt 646, 686

rescheduleint 646, 686

reschedulelog 646, 686

rmm_abort_command 399

rmm_check_services 397

rmm_command_gid 400

rmm_command_uid 400

rmm_command_watch_time 400

rmm_conf 142

rmm_down_with_system 397

rmm_get_pid_command 400

rmm_start_command 399

rmm_start_watch_time 400

rmm_stop_command 399

rmm_sysdown_with_rm 398

rmm_system_behavior 397

RMM サービス定義 397

RMM サービスを使用 142

rmtim 663

rpc_buffer_pool_max 520, 578

rpc_datacomp 160

rpc_delay_statistics 156

rpc_destination_mode 517, 567

rpc_extend_function 443, 514, 555

rpc_max_message_size 179

rpc_message_level 157

rpc_multi_tp1_in_same_host 158

rpc_netmask 163
rpc_port_base 162
rpc_rap_auto_connect 517, 567
rpc_rap_inquire_time 517, 567
rpc_request_cancel_for_timedout 517, 568
rpc_response_statistics 430, 513, 551
rpc_retry 154
rpc_retry_count 155
rpc_retry_interval 155
rpc_router_retry_count 174
rpc_router_retry_interval 174
rpc_send_retry_count 169, 519, 574
rpc_send_retry_interval 170, 519, 574
rpc_server_busy_count 169
rpc_service_retry_count 514, 555
rpc_trace 150, 430, 515, 560
rpc_trace_name 150, 431, 515, 561
rpc_trace_size 151, 431, 515, 561
RPC コール（同期応答型）イベントを取得 490
RPC コール（連鎖 RPC 型）イベントを取得 490
RPC サービスの機能拡張レベル 443, 514, 555
RPC 処理のリトライ回数最大値 155
RPC 処理のリトライ間隔 155
RPC 送受信電文の最大長 179
RPC タイムアウトイベントを取得 491
RPC トレースを取得 150, 430, 560
RPC トレースを取得するファイルのサイズ 151, 431, 515, 561
RPC トレースを取得するファイル名 150, 431, 515, 561
RPC の受け付けの開始時期 133
RPC の処理をリトライ 154
RPC のトラブルシュートメッセージの表示レベル 157
RPC 抑止リストに登録されたノードの監視機能を使用 211
rts_cpd_collct_cpd 481
rts_cpd_validt_cpd 481
rts_dam_cache_block 496
rts_dam_fj 496
rts_dam_read 495
rts_dam_read_err 495
rts_dam_shm_pool 496
rts_dam_trn_branch 496
rts_dam_write 495
rts_dam_write_err 495
rts_item_max 471
rts_jnl_buf_full 481
rts_jnl_io_wait 482
rts_jnl_jnl_input 483
rts_jnl_jnl_output 482
rts_jnl_read 483
rts_jnl_swap 482
rts_jnl_wait_buf 481
rts_jnl_write 482
rts_lck_deadlock 484
rts_lck_lock_acqst 483
rts_lck_lock_wait 483
rts_log_file 472
rts_log_file_backup 473
rts_log_file_count 473
rts_log_file_name 472
rts_log_file_size 472
rts_mcf_ap_scd_stay 503
rts_mcf_ap_usr_srvc 504
rts_mcf_in_msg_scd_wait 504
rts_mcf_out_msg_norm_scd_wait 505
rts_mcf_out_msg_prio_scd_wait 505
rts_mcf_out_msg_resp_scd_wait 504
rts_mcf_out_msg_sync_scd_wait 504
rts_mcf_que_scd_wait_num 505
rts_nam_global_cache_hit 484
rts_nam_local_cache_hit 484
rts_nam_lookup 484
rts_nam_node_lookup 485
rts_nam_node_lookup_responce 485
rts_osl_dynmem_acq 486
rts_osl_dynmem_pol 486
rts_osl_stamem_acq 485

rts_osl_stamem_pol 485
rts_prc_prc_genert 486
rts_prc_prc_num 487
rts_prc_prc_term 487
rts_prc_sys_abnml 487
rts_prc_uap_abnml 486
rts_que_delay_msg 489
rts_que_delay_rec 489
rts_que_delay_wrt 489
rts_que_read 487
rts_que_read_err 488
rts_que_real_read 489
rts_que_real_write 489
rts_que_wait_buf 488
rts_que_write 488
rts_que_write_err 488
rts_rpc_rpc_call 490
rts_rpc_rpc_call_chained 490
rts_rpc_rpc_ovrtim 490
rts_rpc_usr_srvc 490
rts_scd_lack_buf 491
rts_scd_parallel=Y;N 492
rts_scd_scd_stay 492
rts_scd_scd_wait 491
rts_scd_schedule 491
rts_scd_svc_scd_wait 492
rts_scd_svc_using_buf=Y;N 492
rts_scd_using_buf 491
rts_service_max 471
rts_swap_message 474
rts_tam_read 498
rts_tam_read_err 498
rts_tam_real_renew 497
rts_tam_real_renew_time 497
rts_tam_rec_refer 497
rts_tam_rec_renew 497
rts_tam_write 498
rts_tam_write_err 498
rts_trcput_interval 471

rts_trn_branch 494
rts_trn_cmt_cmd 493
rts_trn_commit 493
rts_trn_haz_cmd 494
rts_trn_mix_cmd 494
rts_trn_rbk_cmd 493
rts_trn_rollback 493
rts_trn_sync_point 494
rts_xar_call 499
rts_xar_call_err 499
rts_xar_commit 501
rts_xar_commit_err 501
rts_xar_end 500
rts_xar_end_err 500
rts_xar_forget 503
rts_xar_forget_err 503
rts_xar_prepare 500
rts_xar_prepare_err 501
rts_xar_recover 502
rts_xar_recover_err 502
rts_xar_rollback 502
rts_xar_rollback_err 502
rts_xar_start 499
rts_xar_start_err 499
rtsput 475
RTS ログファイルのサイズ 472
RTS ログファイルのスワップメッセージ
(KFCA32740-I メッセージ) を出力 474
RTS ログファイルの世代数 473
RTS ログファイルのバックアップファイルを作成 473
RTS ログファイル名 472

S

scd_advertise_control 234
scd_announce_server_status 231
scd_hold_recovery 229
scd_hold_recovery_count 230
scd_message_level 234
scd_pool_warning_interval 521, 582

scd_pool_warning_use_rate 521, 582
scd_poolfull_check_count 521, 582
scd_poolfull_check_interval 521, 581
scd_port 231
scd_refresh_process 523, 588
scd_retry_of_comm_error 234
scd_server_count 228
scd_this_node_first 231
scdbufgrp 236, 526, 596
scdmulti 239, 527, 599
scdsvcdef 528, 600
schedule_delay_abort 521, 579
schedule_delay_limit 520, 578
schedule_method 516, 562
schedule_priority 509, 537
schedule_rate 233
scs_prf_trace_level 450
segsz 641
server_count 132
server_type 513, 552
servgrp 654, 682
servhold 684
service 533
service_expiration_time 517, 569
service_group 532
service_hold 512, 547
service_priority_control 512, 547
service_term_watch_time 514, 558
service_wait_time 516, 563
servname 684
set 形式 44
shmpool_attribute 131
size 665
sndrcvtim 640
sndtim 640
srvghold 683
start_scheduling_timing 133
static_shmpool_size 130
statistics 175

stats 636
status_change_when_termining 517, 568
stay_watch_abort 522, 584
stay_watch_check_interval 522, 585
stay_watch_check_rate 522, 583
stay_watch_queue_count 522, 583
stay_watch_start_interval 522, 584
sts_file_name_1 281
sts_file_name_7 281
sts_initial_error_switch 281
sts_last_active_file 284
sts_last_active_side 284
sts_single_operation_switch 283
syslog 失敗リストのエレメント数 329
syslog 失敗リストを定期的に出力する間隔 329
syslog 出力機能 316, 327
syslog 出力レベル 325
sysvname 638
system_id 152
system_init_watch_time 134
system_terminate_watch_time 133

T

tam_cbl_level 374
tam_conf 141
tam_jnl_err_flag 373
tam_max_filesize 372
tam_max_recsz 373
tam_max_tblnum 372
tam_max_trnfilnum 375
tam_max_trnnum 374
tam_pool_attri 373
tam_tbl_lock_mode 374
tamtable 377
TAM サービス定義 372
TAM サービスの最大同時排他要求数 196
TAM サービスを使用 141
TAM サービスを停止 373
TAM 情報の read イベントを取得 498

TAM 情報の read エラーイベントを取得 498
 TAM 情報の write イベントを取得 498
 TAM 情報の write エラーイベントを取得 498
 TAM テーブルアクセス形態 374
 TAM テーブルの最大レコード長 373
 TAM テーブルの属性の指定 377
 TAM ファイル実更新イベントを取得 497
 TAM ファイル実更新時間イベントを取得 497
 TCP_NODELAY オプションを使用 235
 TCP/IP コネクションの接続時にエラーが発生した場合のリトライ回数 169, 519, 574
 TCP/IP コネクションの接続時にエラーが発生した場合のリトライ間隔 170, 519, 574
 TCP/IP の受信バッファサイズ 172, 520, 575
 TCP/IP の送受信バッファサイズ 289, 343
 TCP/IP の送受信バッファサイズを listen ソケットに設定 171, 574
 TCP/IP の送信バッファサイズ 173, 520, 576
 TCP/IP のネットワーク定義ファイルに指定したサブネットマスク値 163
 tempsize 681
 term_watch_count 223
 term_watch_time 222, 511, 543
 termed_after_service 515, 559
 thdlock_sleep_time 170, 695
 thread_stack_size 260, 696
 thread_yield_interval 261, 520, 577
 tim_watch_count 200
 timereqno 664
 TP1/NET/OSI-TP-Extended の定義 36
 TP1/Server Base 用の共用メモリの見積もり式 758
 trccnt 666
 trf_put 433, 513, 549
 trn_completion_limit_time 264, 392, 445, 522, 585
 trn_cpu_time 247, 381, 433, 512, 547
 trn_crm_use 252
 trn_expiration_time 245, 380, 432, 510, 539
 trn_expiration_time_suspend 246, 381, 432, 510, 539
 trn_extend_function 265
 trn_limit_time 254, 388, 437, 516, 564
 trn_max_crm_subordinate_count 252
 trn_max_subordinate_count 249
 trn_optimum_item 250, 386, 434, 513, 552
 trn_partial_recovery_type 256, 390, 438, 517, 565
 trn_prf_event_trace_condition 264
 trn_prf_event_trace_level 263
 trn_prf_trace_level 163
 trn_processing_in_rm_error 251
 trn_rcv_open_close_scope 266
 trn_recovery_failmsg_interval 259
 trn_recovery_list_remove 252
 trn_recovery_list_remove_level 252
 trn_recovery_process_count 245
 trn_retry_count_rm_open 260
 trn_retry_interval_rm_open 260
 trn_rm_open_close_scope 249, 513, 552
 trn_rollback_information_put 253, 388, 436, 515, 562
 trn_rollback_response_receive 255, 390, 438, 516, 565
 trn_start_recovery_interval 262
 trn_start_recovery_mode 262
 trn_start_recovery_watch_time 262
 trn_statistics_item 248, 385, 433, 513, 550
 trn_tran_process_count 244
 trn_tran_recovery_list 247
 trn_tran_statistics 246
 trn_wait_rm_open 259
 trn_watch_time 253, 387, 435, 516, 563
 trn_xa_commit_error 263
 trn_xar_use 262
 trnlnkrm 403
 trnmode 681
 trnrmid 525, 593
 trnstring 270
 TRN イベントトレース関連のメッセージの表示レベル 469

TRN イベントトレース情報ファイルのサイズ 469
TRN イベントトレース情報ファイルの世代数 470
TRN イベントトレース定義 469
TRN イベントトレースの取得レベル 263
type 511, 545, 679
TZ 333

U

uap_conf 140
uap_trace_file_put 180, 511, 542
uap_trace_max 429, 510, 541
UAP 異常終了イベントを取得 486
UAP 異常終了通知イベント 679
UAP 共通定義 639
UAP 共用ライブラリ 533
UAP 共用ライブラリのサーチパスをリラン時に引き継ぐ 221
UAP トレース格納最大数 429, 510, 541
UAP トレースのトレース情報を、ファイルに取得 180
UAP プロセスでアクセスするファイルおよびパイプの最大数 514, 557
uid 430, 455, 511, 546
UJ コード 298
user_command 131
user_command_online 134
user_command_online_tp1mgr_id 135
user_command_online オペランドに指定したシステム開始完了コマンドを OpenTP1 管理者の UID/GID で実行 135
user_server_ha 133
usertime 664

W

watch_next_chain_time 510, 540
watch_time 189, 213, 226, 235, 268, 279, 284, 290, 302, 310, 332, 336, 366, 375, 393, 451, 523, 590
watchcnt 654
watchint 654
write メッセージイベントを取得 488

X

xa_open 関数発行リトライインタバル時間 260
xa_open 関数発行リトライ回数 260
xar_eventtrace_level 275
xar_eventtrace_record 275
xar_msdtc_use 276
xar_prf_trace_level 276
xar_session_time 276
xarfile 278
XAR イベントトレース情報の出力レベル 275
XAR イベントトレース情報ファイルの最大出力レコード数 275
XAR 性能検証用トレース関連のメッセージの表示レベル 462
XAR 性能検証用トレース情報の取得レベル 276
XAR 性能検証用トレース情報ファイルのサイズ 462
XAR 性能検証用トレース情報ファイルの世代数 463
XAR 性能検証用トレース定義 462
XAR ファイルの指定 278
xat_aso_con_event_svcname 407
xat_aso_discon_event_svcname 407
xat_aso_failure_event_svcname 407
xat_conf 143
xat_connect_resp_time 521, 580
XAT_CONNECT_RESP_TIME 524, 592
xat_osi_usr 515, 560
xat_trn_expiration_time 515, 560
xatinitaeq 406
xatinitapt 406
XATMI 通信サービス定義 406
XATMI 通信サービスと UAP との通信で使用される FDS の最大値 408
XATMI 通信サービスを使用 143
XATMI のパラダイム 552
xatsrvadd 409
XA 関数 (xa_open(), xa_commit(), xa_rollback(), xa_close()) の発行順序を変更する 266
XA リソースサービス定義 275
XA リソースサービスを使用する 262

あ

アーカイブ時の転送データの最大長 297, 349
アーカイブジャーナルサービス定義 347
アーカイブジャーナルファイルのファイルグループ名と属性の定義 350
アーカイブジャーナルファイルのファイルグループを構成する物理ファイルの定義 351
アーカイブジャーナルファイルをスワップ 348
アーカイブジャーナルファイルを二重化 347
アーカイブ対象ジャーナルをすべてアーカイブ 298
アーカイブの対象とする 299
アーカイブ用バッファサイズ 297
アイドル状態のトランザクションブランチの監視時間 276
空きバッファ待ちイベントを取得 481
アクセス関数をエラーリターン 357
アソシエーションの異常解放通知を受信 407
アソシエーションの確立通知を受信 407
アソシエーションの正常解放通知を受信 407
アプリケーション異常終了回数のカウント方法 685
アプリケーション異常終了限界回数 685
アプリケーション異常終了時の再スケジュール回数 646, 686
アプリケーション異常終了時の再スケジュール間隔 646, 686
アプリケーション環境定義 677
アプリケーション起動環境定義の開始 670
アプリケーション起動環境定義の終了 674
アプリケーション起動機能 610
アプリケーション起動時の遅延許容時間 643
アプリケーション起動定義 656
アプリケーション起動定義コマンドの指定順序 657
アプリケーション起動プロセス 610
アプリケーション起動プロセス識別子 512, 547, 658, 677
アプリケーション起動プロセスの属する MCF マネジャ識別子 513, 549
アプリケーション起動用論理端末定義 672
アプリケーション数上限値 668
アプリケーション属性定義 678

アプリケーションの型 679
アプリケーションの種別 679
アプリケーションの処置 680
アプリケーションプログラムが閉塞する場合 755
アプリケーションプログラムとシステム環境設定の関連 743
アプリケーションプログラムに関連する OpenTP1 のシステム定義 743
アプリケーション名 613, 679
アプリケーション名とサービス名との関係 613
アボート用シグナル番号 514, 554
アンロード待ち状態をチェック 300, 348

い

一時クローズ処理要求が到着していない 261, 576
一般分岐型送信メッセージの処理待ち情報を取得 505
一方送信型論理端末 672
イベント登録 157
インタバルサービス定義 279

う

上書きできるファイルグループとして割り当てる 308

え

エージェントノードのノードリストに、ノーマルノードのノードの情報を登録するかどうか 189
エラーイベント処理用 MHP 610
エラーが発生したときの処置 259
エラーリターンした場合の OpenTP1 の処置 251
エントリポイント名 533

お

オートコネクトモード 567
オープン状態 303, 311, 350
オンライン中に使用する TAM テーブルの最大数 372
オンライン中に使用する TAM テーブルの最大容量 372
オンライン中に追加する最大論理ファイル数 356

か

回復対象外ファイル 370
拡張 RM 登録定義 402
拡張バッファ数 669
拡張予約定義 650
各ネットワークコミュニケーション定義とその出力オブジェクトファイル名 611
片系運転 296
片系クローズ状態のファイルグループを使用 347
カレントワーキングディレクトリ作成用のディレクトリパス 178
環境変数 LANG 190
環境変数 TZ 333
環境変数値 268, 524, 591, 592
環境変数名 268, 524, 591, 592
監査イベントを取得する定義 779
監査ログ機能 316
監査ログ機能を使用 330
監査ログ取得メッセージ ID 587
監査ログファイルの最大サイズ 331
監査ログファイルの最大数 331
監査ログファイルの出力先ディレクトリ 330
監査ログを取得する項目のメッセージ ID 332, 448, 456, 523
監視間隔時間 210
監視時間 246, 381, 432, 539
監視する rap リスナーを指定 455
監視対象 RM 開始コマンド 399
監視対象 RM 開始処理監視時間 400
監視対象 RM 強制終了コマンド 399
監視対象 RM コマンド監視時間 400
監視対象 RM 終了コマンド 399
監視対象 RM 定義 399
監視対象 RM 名 397
監視対象プロセス ID 取得コマンド 400
監視対象リソースマネージャも同時にダウン 397

き

記号の説明 48

記述形式 45

起動通知機能を使用 152, 168

起動通知を行う 167

起動プロセス数の監視イベントを取得 487

基本事項 44

キャッシュブロック再利用境界デフォルト値 363

キャッシュブロック数のしきい値の指定 369

キャッシュブロックの再利用境界値の指定 367

キャッシュブロック用領域 363

キュー（ディスク）の種別 648

キューグループ 412

キューグループ ID 412, 648, 673, 683

キューファイル 410

共通定義 656

共通定義コマンドの指定順序 657

共用メモリ上にシステム統計情報を取得 175

共用メモリプールをメモリ上に固定 131

<

クライアント UAP の応答待ち時間をサーバに意識 568

クライアント拡張サービスのポート番号 383

クライアント拡張サービスを使用 142

クライアントからの要求最大待ち時間が満了したことを検知したとき、エラーメッセージを出力 440

クライアントサービス定義 379

クライアント数表示メッセージの出力間隔 443

グローバルアーカイブジャーナルサービス定義 342
グローバルアーカイブジャーナルサービスの通信応答待ち時間 344

グローバルアーカイブジャーナルサービスのリソースグループ名 296

グローバルアーカイブジャーナルサービスのリソースグループ名の指定 346

グローバルアーカイブジャーナルサービスを実行 141

グローバルキャッシュヒットイベントを取得 484

グローバル検索機能を使用 206

グローバルドメイン 158

グローバルランザクション単位の最適化 286

け

系切り替え機能を使用 141
警告メッセージの出力判断値になるメッセージ格納バッファプール使用率の上限値 521, 582
継続問い合わせ応答用一時記憶データ格納用領域サイズ 681
現用ファイルが確定できないときの論理ファイルの状態 (continue の場合) 282
現用ファイルが確定できないときの論理ファイルの状態 (excontinue の場合) 283

こ

コアファイル格納パス 225
コアファイルの格納時に OpenTP1 で自動的に圧縮 225
更新する最大ブロック数 353
構文要素記号 50
コネクション確立監視時間 166, 518, 573
コネクション確立処理の間隔 443
コネクション確立処理のリトライ回数 442
コネクション確立処理のリトライ間隔 442, 592
コネクション確立処理をリトライする回数 591
コネクション確立要求を格納するキューの長さ 174, 520, 578
コネクションを、OpenTP1 が自動的に管理 567
コマンド形式 44
コマンド実行用グループ ID 400
コマンド実行用ユーザ ID 400
コマンドによるコミットイベントを取得 493
コマンドによるハザードイベントを取得 494
コマンドによるミックスイベントを取得 494
コマンドによるロールバックイベントを取得 494
コミット、ロールバック (レコード更新) イベントを取得 497
コミット、ロールバック (レコード参照) イベントを取得 497
コミットイベントを取得 493
コメント 46

さ

サーバの状態の変化を、次回再開時に反映させる 568
サーバリカバリジャーナルファイル 311, 313
サービス応答の待ち時間 753
サービス関数開始から終了までの実行監視時間 517, 569
サービスグループ、およびサービスの閉塞状態を引き継ぐ 536
サービスグループ、またはサービスを閉塞 536
サービスグループ数上限値 652
サービスグループ属性定義 654
サービスグループのグループアクセスリストを設定 261, 577
サービスグループの種別 545
サービスグループの処置 683
サービスグループの登録数 650
サービスグループ名 532, 613, 654, 682
サービス情報キャッシュ領域の大きさ 203
サービス情報の検索回数イベントを取得 484
サービス情報領域の大きさ 202
サービス数上限値 652
サービス単位にスケジュールサービスの動作を指定 528, 600
サービス単位のスケジュール待ちイベントを取得 492
サービス単位の閉塞管理 547
サービス単位のメッセージ格納バッファプールの使用中サイズイベントを取得 492
サービス定義ファイルまたはユーザサービスデフォルト定義ファイルの格納ディレクトリ 138
サービスの合計数 230
サービスの処置 684
サービスの送信先を決定 567
サービスのリトライ回数の最大値 514, 555
サービス名 533, 613, 684
サービス要求滞留数 518, 571
サービス要求滞留値 384
サービス要求でのエラーイベントを取得 499
サービス要求の回数イベントを取得 499

再開始時に変更できるシステムサービス定義の一覧 703
再開始時に変更できる定義 703
最小ポート番号 162
最新の現用ファイル 284
最大 CRM 経由子トランザクションブランチ数 252
最大応答待ち時間 189, 213, 226, 235, 268, 279, 284, 290, 302, 310, 332, 336, 366, 375, 393, 523, 590
最大子トランザクションブランチ数 249
最大サーバ数 132
最大時間監視サービス数 200
最大取得項目数 471
最大取得サービス数 471
最大処理多重度 662
最大処理多重度定義 662
最大セグメント長 641
最大同時起動サーバプロセス数 219
最大プロセス数 379, 383, 509, 535
最大ブロック長 365
最大メッセージ長 509, 537
最大ユーザサーバ数 228
最大レコードデータ長 292
作成手順 42

し

自 AE 修飾子 406
自 AP 名称 406
仕掛中トランザクション情報メッセージ間隔最小時間 259
時間監視間隔 663
システム開始完了コマンド 134
システム環境設定の関係 743
システム環境定義 128
システム共通定義 148, 743
システムサーバ異常終了イベントを取得 487
システムサービス共通情報定義 693
システムサービス構成定義 140
システムサービス情報定義 691
システムサービス情報定義ファイル名 638

システムサービス定義 31
システムサービス定義とファイル名 52
システムサービス定義の概要 51
システムサービス定義の構成 38
システムサービス定義の詳細 127
システムサービスの起動 144
システムジャーナルサービス定義 292
システムジャーナルファイルのファイルグループ名と属性の定義 303
システムジャーナルファイルのファイルグループを構成する物理ファイルの定義 304
システム終了監視時間 133
システム初期化の待ち時間 134
システム全体の開始処理を続行 397
システム定義 743
システム定義のデフォルト値選択オプション 136
システム定義のファイル名 744
システム内の通番を付ける 319, 321, 323, 327
システム内のメッセージ通番を付ける 317
システムの開始方法 129
実行間隔時間 361
実行形式プログラム名 532
実時間監視タイムアウトプロセスのコアダンプ出力を抑制する時間間隔 163
指定参照先ノードへのサービス検索の応答送信回数イベントを取得 485
指定参照先ノードへのサービス検索の送信回数イベントを取得 485
自動アンロード機能を使用 301
自ノードと同じノード識別子を指定したノードから通信を受信した場合に、KFCA00677-W メッセージを出力 212
自ノードのサーバの状態を、他ノードに通知 231
自ノードのサーバを優先してスケジュール 231
ジャーナル関係のファイルの指定 291
ジャーナルサービス定義 286
ジャーナルサービスの通信応答待ち時間 289
ジャーナル出力（ブロック）イベントを取得 482
ジャーナル情報の read イベントを取得 483
ジャーナル情報の write イベントを取得 482

ジャーナル入力イベントを取得 483
ジャーナルの出力方式に対する最適化レベル 286
ジャーナルファイルグループの追加 718
ジャーナルファイルレス機能を使用 184
ジャーナルファイルを二重化 296
ジャーナルブロック数 295
ジャーナルレコード種別 298
ジャーナルをスワップ 295
終了準備警告監視時間 135
縮退運転 651
縮退運転オプション 307
受信メッセージの扱い 683
出力キュー 410
出力通番使用論理端末数 634
出力通番の初期値 642
出力メッセージの割り当て先 673
出力要求時の時刻を付ける 318, 320, 322, 325, 328
出力要求時の日付を付ける 318, 320, 322, 324, 328
出力要求元のプログラム ID を付ける 318
出力要求元のホスト名を付ける 318, 320, 322, 325, 328
取得した統計情報を RTS ログファイルに出力 472
条件付きジャーナル非同期出力 287
常設コネクション問い合わせ間隔最大時間 382
常設コネクション問い合わせ間隔最大時間の適用範囲 754
状態引き継ぎ定義 652, 668
常駐プロセス数 379, 383, 509, 535

す

スケジューリングする 547
スケジュールイベントを取得 491
スケジュールキュー 541
スケジュールキューの滞留監視インタバル時間 522, 584
スケジュールキューの滞留監視判定インタバル時間 522, 585
スケジュールキューの滞留監視判定式 584
スケジュールキューの滞留監視判定式を満たした場合に、OpenTP1 システムをダウンさせる 584

スケジュールキューの滞留監視判定処理で使用するサービス要求の処理率 522, 583
スケジュールキューの滞留監視判定を開始する際の判断になるサービス要求滞留数 522, 583
スケジュールキューを、サーバ起動時に割り当てる 570
スケジュールサービス定義 228
スケジュールサービスのポート番号 231
スケジュール滞留イベントを取得 492
スケジュール遅延限界経過時間 520, 578
スケジュール遅延時にシステムダウン 579
スケジュールできなかったメッセージサイズイベントを取得 492
スケジュールの優先順位 509, 537
スケジュールバッファグループ 236, 596
スケジュールバッファグループの指定 526, 596
スケジュール比率 233
スケジュール待ちイベントを取得 491
スケジュール待ち情報を取得 503
スケジュールメッセージ格納セル長 512, 548
ステータスサービス定義 280
ステータスサービスの処置 282, 283
スレッド間で排他が競合した場合のスレッドの待ち時間 170, 695
スワップイベントを取得 482
スワップ先として割り当てる 296

せ

制限事項 46
正常だった系を指定 284
静的共用メモリの使用サイズイベントを取得 485
静的共用メモリの総量 130
静的共用メモリプールの必要最大サイズイベントを取得 485
性能検証用トレース関連のメッセージの表示レベル 458
性能検証用トレース情報ファイルのサイズ 458
性能検証用トレース情報ファイルの世代数 459
性能検証用トレース情報を取得 163
性能検証用トレース定義 458

接続待ちキューに保留できる最大要求数 442

前回の終了モード 129

そ

属性表示記号 49

ソケットウィンドウサイズ 441

ソケット再利用可能監視時間 165, 449, 518, 572

ソケット受信型サーバ受信メッセージ数の最大値
512, 548

ソケット受信型サーバ受信メッセージ長の最大値
512, 549

ソケット生成処理のリトライ回数 442

ソケット生成処理のリトライ間隔 442

ソケットの一時クローズ処理開始数パーセンテージ
164, 449, 518, 571

ソケットの一時クローズ処理非対象数パーセンテージ
164, 449, 518, 571

ソケットの再利用指示を受信できる契機を与えるイン
タバル時間 261, 520, 577

ソケット用ファイル記述子の最大数 159, 204, 222,
232, 258, 287, 342, 444, 514, 556, 693

た

待機系のユーザサーバを起動 133

対象となるシステムサービス名 306

ダイナミックコネクションスケジュール機能を使用
440

タイマ起動メッセージ廃棄通知イベント 679

タイマサービス定義 200

タイマ定義 663

タイムアウト情報ファイルの削除レベル 197

タイムアウト情報を出力 197

タイムゾーン 333

滞留警告メッセージの出力間隔 448

他システムが提供するサービス名の指定 409

他ノードのサービス情報の有効時間 212

端末タイプ 672

ち

チェックポイントダンプサービス定義 306

チェックポイントダンプ取得イベントを取得 481

チェックポイントダンプ取得契機のスキップ回数の上
限值 309

チェックポイントダンプ取得用バッファ長 307

チェックポイントダンプファイルのファイルグループ
名と属性の定義 311

チェックポイントダンプファイルのファイルグループ
を構成する物理ファイルの定義 313

チェックポイントダンプファイルを二重化 308

チェックポイントダンプ有効化イベントを取得 481

遅延書き込み（回数）イベントを取得 489

遅延書き込み最大レコード長 410

つ

通信相手のサービス情報の指定 418

通信イベント障害時のエラーイベント通知機能 681

通信イベント処理用 SPP のアソシエーション確立の最
大応答待ち時間 521, 524, 580, 592

通信関数発行回数上限値 639

通信サービス定義 638

通信制御データの受信監視時間 167, 519, 573

通信遅延時間 156

通信遅延時間統計情報を取得 156

通信リソースマネージャ 252

て

定義オブジェクトファイル 609

定義オブジェクトファイル生成に使用するユティリ
ティ起動コマンド 624

定義オブジェクトファイルの生成 624

定義ソースファイル 609

定義の規則 44

定義の構成 37

定義の作成手順 42

定義のしかた 699

定義の種類 56, 615

定義の体系 31

定義の変更手順 702

定義ファイルの格納ディレクトリ 137, 138

定義例 721

ディスク出力機能を使用 665
ディスク出力メッセージ最大格納数 673
ディファード更新機能 361
ディファード更新処理 370
データコミュニケーション定義 656
データ送信監視回数 166, 519, 573
データ送信監視間隔 166, 519, 573
テーブル排他モードを指定 374
デッドロックイベントを取得 484
デッドロック情報とタイムアウト情報を出力 196
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レベル 197
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を削除 197
デッドロックの優先順位 509, 537
デバッグ情報を取得 137
電文送受信時のリトライ間隔 442

と

問い合わせ応答型送信メッセージの処理待ち情報を取得 504
問い合わせ型論理端末 672
問い合わせ間隔最大時間 429
同一グローバルドメインとして運用 159
同期型受信監視時間 640
同期型送受信監視時間 640
同期型送信監視時間 640
同期型送信メッセージの処理待ち情報を取得 504
同期点処理全体の最適化 250, 435
同期点処理の限界経過監視時間 515, 560
統計情報 247
統計情報項目 248, 385, 433, 513, 550
統計情報取得サービスの指定 475
統計情報の取得間隔 471
同時実行 DAM トランザクションブランチ数イベントを取得 496
同時実行サービス数イベントを取得 493
同時実行トランザクション数 410
同時走行最大トランザクションブランチ数 374
同時に起動するトランザクションブランチの数 244

同時に実行するトランザクションブランチの数 357
動的共用メモリの最大使用時の総量 131
動的共用メモリの使用サイズイベントを取得 486
動的共用メモリプールの必要最大サイズイベントを取得 486
ドメイン代表スケジュールサービスのポート番号 161
ドメイン定義ファイルを使用したドメイン変更時の最大ノード数 177
ドメインネームサービスに問い合わせる 161
ドメイン変更時の最大ノード数 177
ドメイン名 161
トランザクショナル RPC 実行プロセスのポート番号 384
トランザクショナル RPC 実行プロセスを起動 383
トランザクション開始要求でのエラーイベントを取得 499
トランザクション開始要求の回数イベントを取得 499
トランザクション回復処理方式 262
トランザクション完了限界時間 264, 392, 445, 522, 585
トランザクションサービス定義 243
トランザクションサービスの機能の拡張レベル 265
トランザクション最適化項目 250, 386, 434, 513, 552
トランザクション終了要求でのエラーイベントを取得 500
トランザクション終了要求の回数イベントを取得 500
トランザクション通知要求でのエラーイベントを取得 503
トランザクション通知要求の回数イベントを取得 502
トランザクション同期点処理時の最大通信待ち時間 253, 387, 435, 516, 563
トランザクション同期点処理方式を指定 256, 390, 438, 565
トランザクション内最大アクセステーブル数 375
トランザクションのコミット処理要求でのエラーイベントを取得 501
トランザクションのコミット処理要求の回数イベントを取得 501
トランザクションの時間監視 745
トランザクションの単位を指定 362

トランザクションのプリペア処理要求でのエラーイベントを取得 501

トランザクションのプリペア処理要求の回数イベントを取得 500

トランザクションのロールバック処理要求でのエラーイベントを取得 502

トランザクションのロールバック処理要求の回数イベントを取得 502

トランザクション破棄要求でのエラーイベントを取得 503

トランザクション破棄要求の回数イベントを取得 503

トランザクションブランチ CPU 監視時間 247, 381, 433, 512, 547

トランザクションブランチ限界経過時間 245, 380, 432, 510, 539

トランザクションブランチごとの統計情報を取得 247

トランザクションブランチ最大実行可能時間 254, 388, 437, 516, 564

トランザクションブランチの最大数 249

トランザクションブランチの処理を監視するとき、次の処理時間も監視時間を含む 246, 381, 432, 539

トランザクションブランチの統計情報 248

トランザクションブランチの統計情報を取得 385

トランザクションリカバリジャーナルに出力 433, 549

トランザクションを発生 540

トレース環境定義 665

トレース取得種類 264

トレースとして取得する送受信メッセージの最大サイズ 666

トレースの取得レベル 163

トレースバッファの大きさ 665

トレースバッファの数 666

トレースファイルの数 666

な

内部通信路名 670, 680

名前付きパイプ 562

に

入出力キュー定義 648

入出力待ちイベントを取得 482

入力キュー 410

入力キューの滞留監視インタバル時間 654

入力キューの滞留監視数 654

入力キューの滞留数の情報を取得 505

入力メッセージ最大格納数 680

入力メッセージの割り当て先 683

ね

ネームサービス定義 202

ネームサービスで管理するサーバ UAP の取得数 208

ネームサービスのポート番号 151

ネットワークコミュニケーション定義 31

ネットワークコミュニケーション定義とシステムサービス定義との関連 612

ネットワークコミュニケーション定義とファイル名 609

ネットワークコミュニケーション定義の概要 608

ネットワークコミュニケーション定義の構成 40

ネットワークコミュニケーション定義のコマンド 613

ネットワークコミュニケーション定義の詳細 631

ネットワークコミュニケーション定義を登録するファイル名 609

ネットワークライブラリ定義 31

ネットワークライブラリ定義の構成 41

の

ノード監視機能を使用 208

ノードグループ名 394

ノード識別子 154, 296

ノード識別子を付ける 329

ノードの追加 717

ノードの動作モード 186

ノード名 153, 168, 394

ノードリスト同期待ち時間 216

ノードリストファイルの指定 217

は

排他あり参照指定で読み込むブロックを、トランザクション決着までキャッシュバッファ領域上に残す 364

排他解除検知インタバル最大時間 198

排他待ち限界経過時間 196
排他待ちの優先順位 512, 546
排他待ちプロセス 198
発行タイミング 249, 552
バッファグループ定義 669
バッファグループ番号 669
バッファ数 669
バッファ長 669
バッファ満杯イベントを取得 481
バッファ領域長 358
バッファ領域をメモリ上に固定 361

ひ

非オートコネクトモード 567
非常駐プロセスを終了 559
非ランザクション MHP 限界経過時間 642, 684

ふ

ファイルグループがスワップ先に割り当てられる条件 299
プーリングするバッファ数 520, 578
負荷監視インタバル時間 518, 570
負荷レベル通知メッセージを出力 577
不正アプリケーション名検出通知イベント 679
物理ファイル単位の遅延書き込み（メッセージ）イベントを取得 490
物理ファイル単位の遅延書き込み（レコード）イベントを取得 489
ブランチ実行時間イベントを取得 494
ブランチ同期点処理の実行時間イベントを取得 494
プログラム ID を付ける 321, 323, 325, 328
プロセス ID を付ける 317, 319, 321, 324, 327
プロセスがいつ排他を解除したかを調べる方法 197
プロセスサービス定義 219
プロセスサービスのイベントトレースを取得 224
プロセス終了イベントを取得 487
プロセス数の指定条件 380, 384, 535
プロセス生成イベントを取得 486
プロセス内の通番を付ける 319, 321, 324, 327
プロセス内のメッセージ通番を付ける 317

プロセスの優先順位の変更 429, 509, 535
プロセスを常駐にするか、非常駐にするか 220
プロトコル固有定義 656
文法記述記号 48

へ

並行回復プロセス数 245
閉塞 755
閉塞状態の引き継ぎが必要なサーバ、およびサービスの合計数 230
並列アクセス化する場合の最小分散数 302, 348
並列アクセス化する場合の最大分散数 301, 348

ほ

ポート番号 153, 161, 168
ホスト名 157, 161
ホスト名または IP アドレスの変更 719

ま

待ち合わせ最大時間 287, 342
マネージャノードが停止を検知したエージェントノード、およびノーマルノードのノード情報を、ノードリストから削除するかどうか 188
マルチ OpenTP1 システム 699
マルチ OpenTP1 システムの定義 698
マルチサーバ機能 535
マルチスケジューラ機能の指定 527, 599
マルチスケジューラ機能を使用してスケジューリングする 569
マルチスケジューラデーモンの情報の指定 239
マルチノードエリア／サブエリアに属するノード識別子を指定する 337
マルチノード機能を使用 156
マルチノード構成定義 335
マルチノードのホスト名とポート番号を指定する 340
マルチノード物理定義 339
マルチノード連携制御機能で使用するポート番号 156

み

未決着トランザクション情報ファイルの削除レベル 252
未決着トランザクション情報ファイルを削除 252
未決着トランザクション情報を取得 247
未処理受信メッセージ滞留時間 663
未処理送信メッセージ滞留時間 663
未処理送信メッセージ廃棄通知イベント 663

め

メッセージ ID の変更レベル 289, 344, 445, 523, 586
メッセージ格納バッファプール使用率超過時の警告
メッセージ出力インタバル時間 521, 582
メッセージ格納バッファプール長 392, 510, 537
メッセージ格納バッファプールの指定 236
メッセージ格納バッファプールの使用中サイズイベントを取得 491
メッセージキューサービス (MCF サービス) を使用 141
メッセージキューサービス定義 410
メッセージキュー情報の read エラーイベントを取得 488
メッセージキュー情報の write エラーイベントを取得 488
メッセージキュー情報の空きバッファ待ちイベントを取得 488
メッセージキュー情報の実 read イベントを取得 489
メッセージキュー情報の実 write イベントを取得 489
メッセージキュー定義 31
メッセージキュー定義の構成 41
メッセージキューファイルの指定 412
メッセージキューを解放 554
メッセージ出力指定 516, 563
メッセージ制御機能 31
メッセージ送受信最大監視時間 428, 456
メッセージ送受信用, およびメッセージ編集用のバッファ 669
メッセージ廃棄通知イベント 679
メッセージログ出力機能 316

メッセージログ出力抑止回数 325
メッセージログ通知機能 316
メッセージログ通知機能を使用 323
メッセージログファイルの最大サイズ 316
メッセージログを関連オペランド(log_syslog_xxxx)に従ったフォーマットで syslog に出力 330
メッセージを何を使って受信するかを指定 541
メモリ出力メッセージ最大格納数 673
メモリプールをメモリ上に固定 373
メンテナンス 45

も

モデルアプリケーション属性定義の指定例 676

ゆ

有効保証世代数 307
ユーザ環境設定コマンド 131
ユーザサーバおよびコマンドのパスをリラン時に引き継ぐ 221
ユーザサーバの起動 146
ユーザサーバの最大同時排他要求数 195
ユーザサーバのジャーナルバッファの大きさ 640
ユーザサーバのスケジューリング方式を指定 562
ユーザサーバの多重度の変更 715
ユーザサーバの追加 713
ユーザサーバのパスの指定 227
ユーザサーバの非常駐プロセスのサービス要求待ち時間 516, 563
ユーザサーバの閉塞状態を引き継ぐ 229
ユーザサーバを起動 140
ユーザサーバを自動起動 551
ユーザサービス構成定義 145, 743
ユーザサービス実行イベントを取得 490
ユーザサービス実行情報を取得 504
ユーザサービス定義 529, 743
ユーザサービス定義の指定値と SPP, SUP, および MHP との関係 602
ユーザサービス定義の指定値と UAP との関係 602
ユーザサービスデフォルト定義 506, 743
ユーザサービスネットワーク定義 415

ユーザ識別子 511, 546
ユーザデータを圧縮 160
ユーザ認証機能を使用 162
優先分岐型送信メッセージの処理待ち情報を取得 505
ユーティリティ起動コマンド 624

よ

抑止メッセージ ID 653
予約ファイルをオープン 296
予約ファイルを自動的にオープン 308

ら

ラップ後の出力通番開始値 642
ラップ時の出力通番最大値 642

り

リアルタイム出力機能 316
リアルタイム出力機能を使用 316
リアルタイム取得項目定義 479
リアルタイム統計情報サービス定義 471
リカバリ要求用待機 rap サーバ数 443
リソースマネージャ拡張子の指定 525, 593
リソースマネージャのアクセス用文字列または拡張子の指定 270
リトライ回数 234
リトライ間隔時間 262
リトライ経過時間 262
リトライ指示のロールバック要求 610
リモート API 機能 415, 418
リモート API 機能における性能検証用トレース取得レベル 450
リモート API 機能の機能拡張レベル 450, 523, 588
リモート API 機能を使用して要求するサービスの問い合わせ間隔最大時間 517, 567
リモート API 機能を使用する 426
リラン時のタイマ起動を引き継ぐかどうか 670
履歴情報 (GJ) 687
履歴情報 (IJ) 687
履歴情報 (OJ) 687
履歴情報を取得 563

れ

レスポンス統計情報を取得 430, 551
連鎖 RPC 間隔監視時間 510, 540
連続異常終了限界回数 223
連続異常終了限界経過時間 222, 511, 543
連続サービス異常終了限界経過時間 514, 558

ろ

ローカルキャッシュヒットイベントを取得 484
ロールバックイベントを取得 493
ロールバック完了通知を受信 255, 390, 438, 565
ロールバック情報を取得 253, 388, 436, 562
ログサービス定義 315
ログに取得 388, 436, 562
ログメッセージ出力抑止定義 653
ロックサービス定義 195
ロック取得イベントを取得 483
ロック待ちイベントを取得 483
論理端末数上限値 668
論理端末単位に受信メッセージの処理待ち情報を取得 504
論理端末の登録数 650
論理端末名称 672, 680
論理ファイルの指定 370
論理ファイル名 281, 284